Cat. No. W614-E1-01

SYSMAC CP Series

CP2E-E□□D□-□
CP2E-S□□D□-□
CP2E-N□□D□-□

CP2E CPU Unit Software

USER'S MANUAL

OMRON

SYSMAC CP Series
CP2E-E□□D□-□
CP2E-S□□D□-□
CP2E-N□□D□-□
CP2E CPU Unit Software

User's Manual

Produced September 2019

Introduction

Thank you for purchasing a SYSMAC CP-series CP2E Programmable Controller.

This manual contains information required to use the CP2E. Read this manual completely and be sure you understand the contents before attempting to use the CP2E.

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

Applicable Products

● CP-series CP2E CPU Units

- Essential Model CP2E-E□□D□-□
  A model of CPU Unit that supports connections to Programmable Terminals and basic control applications using instructions such as basic, movement, arithmetic, and comparison instructions.
- Standard Model CP2E-S□□D□-□
  A model of CPU Unit that supports connections to inverters and servo drives.
- Network Model CP2E-N□□D□-□
  A model of CPU Unit that supports Ethernet connection and enhanced positioning functions such as 4-axis linear interpolation and pulse.

The CP Series is centered around the CP1H, CP1L, CP1E and CP2E 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.

CP2E CPU Unit Manuals

Information on the CP2E CPU Units is provided in the following manuals.

Refer to the appropriate manual for the information that is required.

	CP2E CPU Unit Hardware User's Manual(Cat. No. W613)	This Manual CP2E CPU Unit Software User's Manual(Cat. No. W614)	CP1E/CP2E CPU Unit Instructions Reference Manual(Cat. No. W483)
1 Mounting and Setting Hardware	· Names and specifications of the parts of all Units · Basic system configuration for each CPU Unit · Connection methods for Expansion I/O Units and Expansion Units		
2 Wiring	· Wiring methods for the power supply · Wiring methods between external I/O devices and Expansion I/O Units or Expansion Units		
3 Connecting Online to the PLC	Connecting Cables for CX-Programmer Support Software	Procedures for connecting the CX-Programmer Support Software	
4 Software Setup		Software setting methods for the CPU Units (PLC Setup)	
5 Creating the Program		· Program types and basic information · CPU Unit operation · Internal memory · Built-in CPU functions · Settings	Detailed information on programming instructions
6 Checking and Debugging Operation		· Checking I/O wiring, setting the Auxiliary Area settings, and performing trial operation · Monitoring and debugging with the CX-Programmer	
7 Maintenance and Troubleshooting	Error codes and remedies if a problem occurs		

Manual Configuration

The CP2E CPU manuals are organized in the sections listed in the following tables. Refer to the appropriate section in the manuals as required.

CP2E CPU Unit Software User's Manual (Cat. No. W614) (This Manual)

Section	Contents
Section 1 Overview	This section gives an overview of the CP2E, describes its application procedures.
Section 2 CPU Unit Memory	This section describes the types of internal memory in a CP2E CPU Unit and the data that is stored.
Section 3 CPU Unit Operation	This section describes the operation of a CP2E CPU Unit.
Section 4 Programming Concepts	This section provides basic information on designing ladder programs for a CP2E CPU Unit.
Section 5 I/O Memory	This section describes the types of I/O memory areas in a CP2E CPU Unit and the details.
Section 6 I/O Allocation	This section describes I/O allocation used to exchange data between the CP2E CPU Unit and other units.
Section 7 PLC Setup	This section describes the PLC Setup, which are used to perform basic settings for a CP2E CPU Unit.
Section 8 Overview and Allocation of Built-in Functions	This section lists the built-in functions and describes the overall application flow and the allocation of the functions.
Section 9 Quick-response Inputs	This section describes the quick-response inputs that can be used to read signals that are shorter than the cycle time.
Section 10 Interrupts	This section describes the interrupts that can be used with CP2E PLCs, including input interrupts and scheduled interrupts.
Section 11 High-speed Counters	This section describes the high-speed counter inputs, high-speed counter interrupts, and the frequency measurement function.
Section 12 Pulse Outputs	This section describes positioning functions such as trapezoidal control, jogging, and origin searches.
Section 13 PWM Outputs	This section describes the variable-duty-factor pulse (PWM) outputs.
Section 14 Serial Communications	This section describes communications with Programmable Terminals (PTs) without using communications programming, no-protocol communications with general components, and connections with a Modbus-RTU Easy Master, Serial PLC Link, host computer and Modbus-RTU Slave.
Section 15 Ethernet	This section gives an outline of the built-in Ethernet function, explains its specification and how to make the settings required for operation.
Section 16 Other Functions	This section describes PID temperature control, clock functions, DM backup functions, security functions.
Section 17 Analog Option Board	This section describes an overview of the Analog Option Board, describes its installation and setting methods, memory allocations, startup operation, refresh time, troubleshooting and how to use the Analog Option Board.
Section 18 Operating the Programming Device	This section describes basic functions of the CX-Programmer, such as using the CX-Programmer to write ladder programs to control the CP2E CPU Unit, to transfer the programs to the CP2E CPU Unit, and to debug the programs.
Appendices	The appendices provide lists of programming instructions, the Auxiliary Area, cycle time response performance, PLC performance at power interruption, memory map and Ethernet functions.

CP2E CPU Unit Hardware User's Manual (Cat. No. W613)

Section	Contents
Section 1 Overview and Specifications	This section gives an overview of the CP2E, describes its features, and provides its specifications.
Section 2 Basic System Configuration and Devices	This section describes the basic system configuration and unit models of the CP2E.
Section 3 Part Names and Functions	This section describes the part names and functions of the CPU Unit, Expansion I/O Units, and Expansion Units in a CP2E PLC.
Section 4 Programming Device	This section describes the features of the CX-Programmer used for programming and debugging PLCs, as well as how to connect the PLC with the Programming Device by USB, Ethernet and serial port.
Section 5 Installation and Wiring	This section describes how to install and wire CP2E Units.
Section 6 Troubleshooting	This section describes how to troubleshoot problems that may occur with a CP2E PLC, including the error indications provided by the CP2E Units.
Section 7 Maintenance and Inspection	This section describes periodic inspections, the service life of the Battery, and how to replace the Battery.
Section 8 Using Expansion Units and Expansion I/O Units	This section describes application methods for Expansion Units.
Appendices	The appendices provide information on dimensions, wiring diagrams, and wiring serial communications, network installation for the CP2E and comparison between CP1E and CP2E.

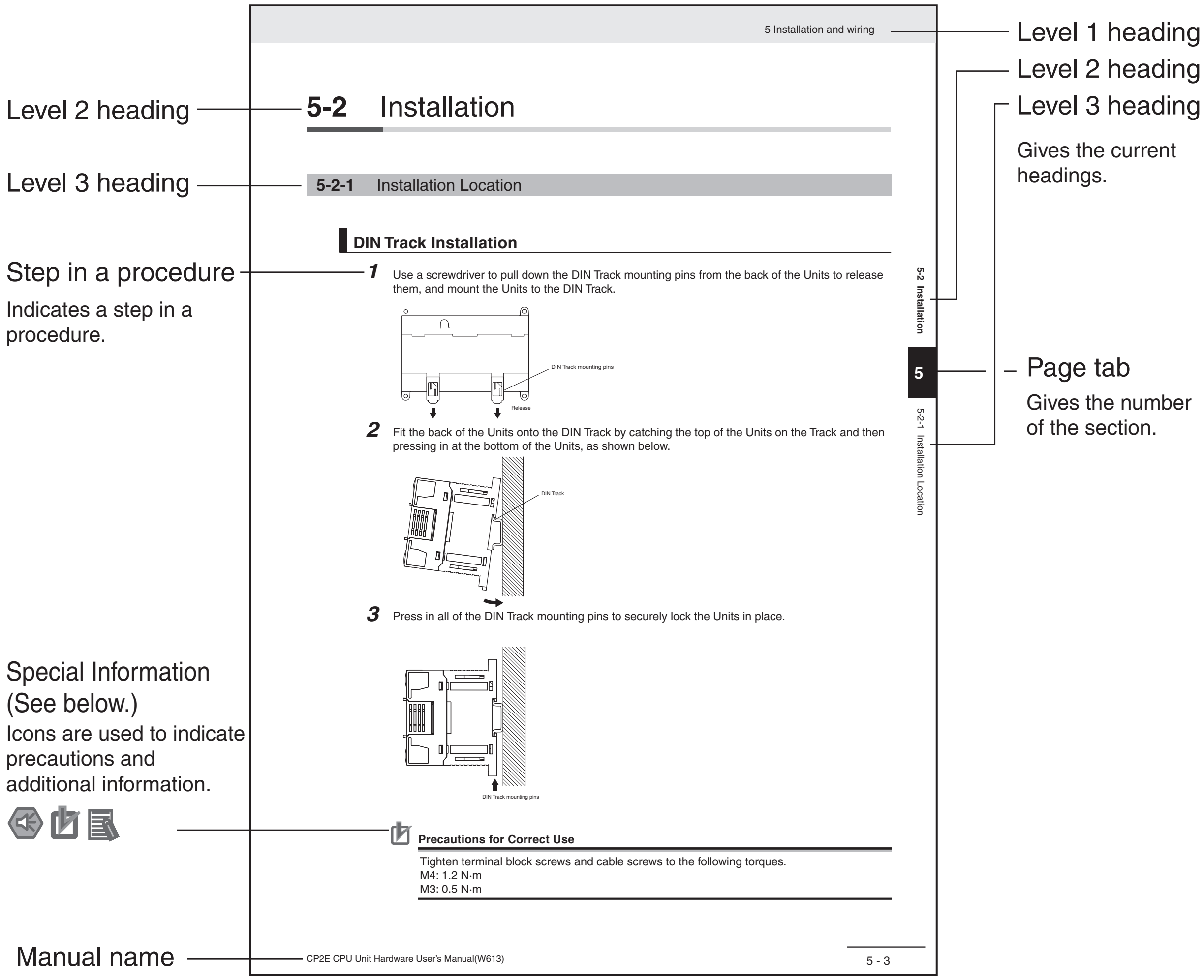
CP1E/CP2E CPU Unit Instructions Reference Manual (Cat. No. W483)

Section	Contents
Section 1 Summary of Instructions	This section provides a summary of instructions used with a CP1E/CP2E CPU Unit.
Section 2 Instruction	This section describes the functions, operands and sample programs of the instructions that are supported by a CP1E/CP2E CPU Unit.
Section 3 Instruction Execution Times and Number of Steps	This section provides the execution times for all instructions used with a CP1E/CP2E CPU Unit.
Section 4 Monitoring and Computing the Cycle Time	This section describes how to monitor and calculate the cycle time of a CP1E/CP2E CPU Unit that can be used in the programs.
Appendices	The appendices provide a list of instructions by Mnemonic and ASCII code table for the CP1E/CP2E CPU Unit.

Manual Structure

Page Structure and Icons

The following page structure and icons are used in this manual.



This illustration is provided only as a sample and may not literally appear in this manual.

Special Information

Special information in this manual is classified as follows:

Precautions for Safe Use
Precautions on what to do and what not to do to ensure using the product safely.

Precautions for Correct Use
Precautions on what to do and what not to do to ensure proper operation and performance.

Additional Information
Additional information to increase understanding or make operation easier.

Terminology and Notation

Term	Description
E-type CPU Unit	An essential model of CPU Unit that supports connections to Programmable Terminals and basic control applications using instructions such as basic, movement, arithmetic, and comparison instructions. Essential models of CPU Units are called “E☐☐-type CPU Units” or “E14/20 CPU Units” in this manual. The models of E☐☐-type CPU Units are shown below. CP2E-E☐☐D☐-☐
S-type CPU Unit	A standard model of CPU Unit that supports connections to inverters and servo drives. Standard models of CPU Units are called “S☐☐-type CPU Units” or “S30/40/60 CPU Units” in this manual. The models of S☐☐-type CPU Units are shown below. CP2E-S☐☐D☐-☐
N-type CPU Unit	A network model of CPU Unit that supports Ethernet connection and enhanced positioning functions such as 4-axis linear interpolation and pulse. Network models of CPU Units are called “N☐☐-type CPU Units” or “N30/40/60 CPU Units” in this manual. The models of N☐☐-type CPU Units are shown below. CP2E-N☐☐D☐-☐
CX-One **CX-Programmer**	A programming device that applies for programming and debugging PLCs. CP2E CPU Units are supported by CX-One version 4.51 or higher and CX-Programmer version 9.72 or higher.

Sections in this Manual

CONTENTS

Section 5 I/O Memory

Section 6 I/O Allocation

Section 7 PLC Setup

Section 8 Overview of Built-in Functions and Allocations

Section 9 Quick-response Inputs

Section 10 Interrupts

Section 11 High-speed Counters

Section 12 Pulse Outputs

Section 13 PWM Outputs

Section 14 Serial Communications

Section 15 Ethernet

Section 16 Other Functions

Section 17 Analog Input/Output Option Board

Section 18 Programming Device Operations

Section A Appendices

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

- **Exclusive Warranty**

Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

- **Limitations**

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.

Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

- **Buyer Remedy**

Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses 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. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc

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

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials 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 user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

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 part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Safety Precautions

Definition of Precautionary Information

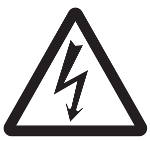
The following notation is used in this manual to provide precautions required to ensure safe usage of a CP-series PLC. The safety precautions that are provided are extremely important to safety. Always read and heed the information provided in all safety precautions.

⚠ WARNING	Indicates an imminently hazardous situation which, if not avoided, will 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.

Precautions for Safe Use
Indicates precautions on what to do and what not to do to ensure using the product safely.

Precautions for Correct Use
Indicates precautions on what to do and what not to do to ensure proper operation and performance.

Symbols

	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for electric shock.
	The circle and slash symbol indicates operations that you must not do. The specific operation is shown in the circle and explained in text.
	The filled circle symbol indicates operations that you must do. The specific operation is shown in the circle and explained in text. This example shows a general precaution for something that you must do.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a general precaution.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for hot surfaces.

⚠Caution

Be sure to sufficiently confirm the safety at the destination when you transfer the program or I/O memory or perform procedures to change the I/O memory.

Devices connected to PLC outputs may incorrectly operate regardless of the operating mode of the CPU Unit.

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.

Sufficiently check safety if I/O bit status or present values are monitored in the Ladder Section Pane or present values are monitored in the Watch Pane.

If bits are set, reset, force-set, or force-reset by inadvertently pressing a shortcut key, devices connected to PLC outputs may operate incorrectly regardless of the operating mode.

Program so that the memory area of the start address is not exceeded when using a word address or symbol for the offset.

For example, write the program so that processing is executed only when the indirect specification does not cause the final address to exceed the memory area by using an input comparison instruction or other instruction.

If an indirect specification causes the address to exceed the area of the start address, the system will access data in other area, and unexpected operation may occur.

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.

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.

Precautions for Safe Use

Observe the following precautions when using a CP-series PLC.

● Handling

- Set the Unit properly as specified in the operation manual. Improper setting of the Unit may result in malfunction.
- Check that the DIP switches and data memory (DM) are properly set before starting operation.
- To initialize the DM Area, back up the initial contents for the DM Area to the built-in Flash Memory using one of the following methods.
  - Set the number of words of the DM Area to be backed up starting with D0 in the *Number of CH of DM for backup* Box in the *Startup Data Read* Area.
  - Include programming to back up specified words in the DM Area to the built-in Flash Memory by turning ON A751.15 (DM Backup Save Start Bit).
- Check the ladder program for proper execution before actually running it on the Unit. Not checking the program may result in an unexpected operation.
- Transfer a routing table to the CPU Unit only after confirming that no adverse effects will be caused by restarting CPU Bus Units, which is automatically done to make the new tables effective.
- The ladder program and parameter area data in the CP2E CPU Units are 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 and a memory error will occur the next time the power supply is turned ON.
- With a CP2E CPU Unit, data memory can be backed up to the built-in Flash Memory. The BKUP indicator will light on the front of the CPU Unit when backup is in progress. Do not turn OFF the power supply to the CPU Unit when the BKUP indicator is lit. If the power is turned OFF during a backup, the data will not be backed up and will not be transferred to the DM Area in RAM the next time the power supply is turned ON.
- Install a battery (sold separately), if you are using clock data for the program. If the battery is not installed, the clock data will be initialized when the power is turned off, and the program may cause malfunction.
- When using a battery, set it to “Detect Low Battery” in PLC settings. If the setting is not changed, a program that uses clock data may cause malfunction, when the battery is exhausted.
- Before replacing the battery, supply power to the CPU Unit for at least 30 minutes and then complete battery replacement within 5 minutes. Memory data may be corrupted if this precaution is not observed.
- The equipment may operate unexpectedly if inappropriate parameters are set. Even if the appropriate parameters are set, confirm that equipment will not be adversely affected before transferring the parameters to the CPU Unit.
- Before starting operation, confirm that the contents of the DM Area is correct.
- After replacing the CPU Unit, make sure that the required data for the DM Area, Holding Area, and other memory areas has been transferred to the new CPU Unit before restarting operation.
- Do not attempt to disassemble, repair, or modify any Units. Any attempt to do so may result in malfunction, fire, or electric shock.
- 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.

● External Circuits

- Always configure the external circuits to turn ON power to the PLC before turning ON power to the control system. 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 terminals 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 instruction, all outputs from PLC will be turned OFF and only the internal output status in the CPU Unit will be maintained.)

Regulations and Standards

Trademarks

SYSMAC is a registered trademark for Programmable Controllers made by OMRON Corporation.

CX-One is a registered trademark for Programming Software made by OMRON Corporation.

Windows is a registered trademark of Microsoft Corporation.

Other system names and product names in this document are the trademarks or registered trademarks of their respective companies.

Software Licenses and Copyrights

This product incorporates certain third party software. The license and copyright information associated with this software is shown at the following.

Related Manuals

The following manuals are related to the CP2E. Use them together with this manual.

Manual name	Cat. No.	Model numbers	Application	Contents

SYSMAC CP Series CP2E CPU Unit Software User's Manual (this manual)	W614	CP2E-E□□D□-□
		CP2E-S□□D□-□
		CP2E-N□□D□-□
			To learn the software specifications of the CP2E PLCs	Describes the following information for CP2E PLCs.
				• CPU Unit operation
				• Internal memory
				• Programming
				• Settings
				• CPU Unit built-in functions
				  • Interrupts
				  • High-speed counter inputs
				  • Pulse outputs
				  • Serial communications
				  • Ethernet
				  • Other functions

			Use this manual together with the CP2E CPU Unit Hardware User's Manual (Cat. No. W613) and Instructions Reference Manual (Cat. No. W483).	
SYSMAC CP Series CP2E CPU Unit Hardware User's Manual	W613	CP2E-E□□D□-□ CP2E-S□□D□-□ CP2E-N□□D□-□	To learn the hardware specifications of the CP2E PLCs	Describes the following information for CP2E PLCs. • Overview and features • Basic system configuration • Part names and functions • Installation and settings • Troubleshooting
			Use this manual together with the CP2E CPU Unit Software User's Manual (Cat. No. W614) and Instructions Reference Manual (Cat. No. W483).	
SYSMAC CP Series CP1E/CP2E CPU Unit Instructions Reference Manual	W483	CP1E-E□□D□-□ CP1E-N□□D□-□ CP1E-NA□□D□-□ CP2E-E□□D□-□ CP2E-S□□D□-□ CP2E-N□□D□-□	To learn programming instructions in detail	Describes each programming instruction in detail. When programming, use this manual together with the CP2E CPU Unit Software User's Manual (Cat. No. W614).
CS/CJ/CP/NSJ Series Communications Commands Reference Manual	W342	CS1G/H-CPU□□H CS1G/H-CPU□□-V1 CS1D-CPU□□H CS1D-CPU□□S CS1W-SCU□□-V1 CS1W-SCB□□-V1 CJ1G/H-CPU□□H CJ1G-CPU□□P CJ1M-CPU□□ CJ1G-CPU□□ CJ1W-SCU□□-V1	To learn communications commands for CS/CJ/CP/NSJ-series Controllers in detail	Describes 1) C-mode commands and 2) FINS commands in detail. Read this manual for details on C-mode and FINS commands addressed to CPU Units.
			Note This manual describes commands addressed to CPU Units. It does not cover commands addressed to other Units or ports (e.g., serial communications ports on CPU Units, communications ports on Serial Communications Units/Boards, and other Communications Units).	

Manual name	Cat. No.	Model numbers	Application	Contents
CX-One FA Integrated Tool Package Setup Manual	W463	CXONE-AL☐☐D-V4	To install the software provided in the CX-One	Describes the overview of the CX-One FA Integrated Tool Package, and how to install and uninstall the CX-One.
CX-Programmer Operation Manual	W446		To learn the operation procedures for the CX-Programmer, the Programming Device for Windows computers	Describes the operation procedures for the CX-Programmer.
CX-Programmer Operation Manual (Function Blocks/ Structured Text)	W447			
CX-Simulator Operation Manual	W366		To learn the operation procedures for the CX-Simulator, the Simulation Device for Windows computers	Describes the operation procedures for the CX-Simulator.
CX-Integrator Operation Manual	W464		To set up and monitor networks	Describes the operation procedures for the CX-Integrator.

1

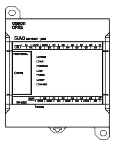
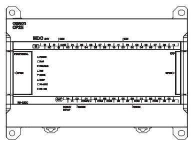
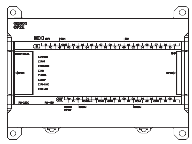
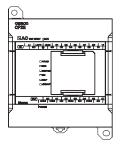
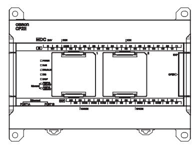
Overview

This section gives an overview of the CP2E and describes its specifications.

1-1 CP2E Overview

1-1-1 Overview of Features

The SYSMAC CP2E Programmable Controller is a package-type PLC made by OMRON that is designed for easy application. The CP2E includes E□□-type CPU Units (essential models) that support connections to Programmable Terminals and basic control applications using basic, movement, arithmetic, and comparison instructions, S□□-type CPU Units (standard models) that support connections to Inverters and Servo Drives and N□□-type CPU Units (network models) that support Ethernet connection and enhanced positioning functions such as 4-axis linear interpolation and pulse.

	Essential Models		Standard Models	Network Models	
	E□□-type CPU Units		S□□-type CPU Units	N□□-type CPU Units	
	CPU Unit with 14, 20 I/O Points	CPU Unit with 30, 40 or 60 I/O Points	CPU Unit with 30, 40 or 60 I/O Points	CPU Unit with 14, 20 I/O Points	CPU Unit with 30, 40 or 60 I/O Points
Appearance					
I/O points	14/20	30/40/60	30/40/60	14/20	30/40/60
Program capacity	4K steps		8K steps	10K steps	
FB capacity	4K steps		8K steps	10K steps	
DM Area capacity	4K words Of these 1,500 words can be written to the built-in Flash Memory.		8K words Of these 7,000 words can be written to the built-in Flash Memory.	16K words Of these 15,000 words can be written to the built-in Flash Memory.	
Mounting Expansion I/O Units and Expansion Units	Not supported.	3 Units maximum	3 Units maximum	Not supported.	3 Units maximum
Model with transistor outputs	Not supported.		Available		
Pulse outputs (Models with transistor outputs only)	Not supported.		2 axes supported.	2 axes supported. (Linear interpolation supported)	4 axes supported. (Linear interpolation supported)
Built-in serial communications port	RS-232C port provided.		RS-232C/RS-485 port provided.	Not provided. Expand to up to two ports by a Option Board.	Not provided. Expand to up to three ports by Option Boards.
Option Board	Not supported.			1 slot	2 slots
Built-in Ethernet port	Not supported.			1 port	2 ports (Switch function)
Connection port for Programming Device	USB port			Ethernet port	
Clock	Not supported.		Supported		
Using a Battery	Not supported.		Supported (CP2W-BAT02 sold separately)		
Battery-free operation	Always battery-free operation. Data in I/O Memory is retained even if no battery is attached.				

1-2 Basic Operating Procedure

In general, use the following procedure.

1. Setting Devices and Hardware

Connect the CPU Unit, Expansion I/O Units, and Expansion Units.
Set the DIP switches on the Option Board and Expansion Units as required.

Refer to *Section 3 Part Names and Functions* and *Section 5 Installation and Wiring* in the *CP2E CPU Unit Hardware User's Manual* (Cat. No. W613).

2. Wiring

Wire the power supply, I/O, and communications.

Refer to *Section 5 Installation and Wiring* in the *CP2E CPU Unit Hardware User's Manual* (Cat. No. W613).

3. Connecting Online to the PLC

Connect the personal computer online to the PLC.

Refer to *Section 4 Programming Device* in the *CP2E CPU Unit Hardware User's Manual* (Cat. No. W613).

4. I/O Allocations

Allocations for built-in I/O on the CPU Unit are predetermined and memory is allocated automatically to Expansion I/O Units and Expansion Units, so the user does not have to do anything.

Refer to Section 6 I/O Allocation in the *CP2E CPU Unit Software User's Manual* (Cat. No. W614).

5. Software Setup

Make the PLC software settings.
With a CP2E CPU Unit, all you have to do is set the PLC Setup.

Refer to *Section 7 PLC Setup* in the *CP2E CPU Unit Software User's Manual* (Cat. No. W614).

6. Writing the Programs

Write the programs using the CX-Programmer.

Refer to *Section 4 Programming Concepts* in the *CP2E CPU Unit Software User's Manual* (Cat. No. W614).

7. Checking Operation

Check the I/O wiring and the Auxiliary Area settings, and perform trial operation.
The CX-Programmer can be used for monitoring and debugging.

Refer to *Section 8 Overview and Allocation of Built-in Functions* in the *CP2E CPU Unit Software User's Manual* (Cat. No. W614).

8. Basic Program Operation

Set the operating mode to RUN mode to start operation.

2

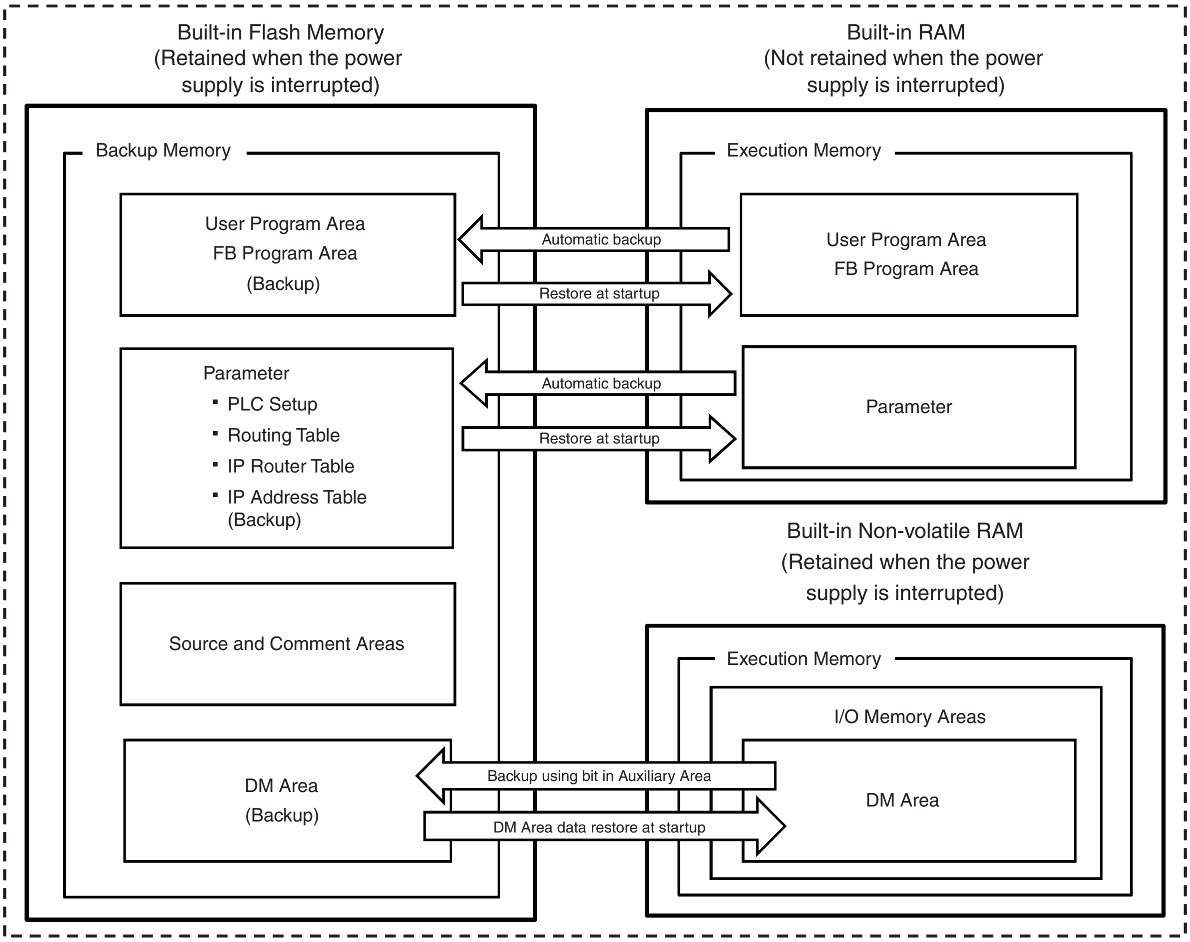
Internal Memory in the CPU Unit

This section describes the types of internal memory in a CP2E CPU Unit and the data that is stored.

2-1 Internal Memory in the CPU Unit

2-1-1 CPU Unit Memory Backup Structure

The internal memory in the CPU Unit consists of built-in RAM, built-in non-volatile RAM and built-in Flash Memory. The built-in RAM and built-in non-volatile RAM are used as execution memory and the built-in Flash Memory is used as backup memory.



Built-in RAM

The built-in RAM is the execution memory for the CPU Unit.

The user program and parameters are stored in the built-in RAM.

Data is unstable when the power supply is interrupted.

Build-in Non-volatile RAM

The built-in non-volatile RAM is the execution memory for the CPU Unit. It is used to retain the data when the power supply is interrupted.

I/O memory is stored in the built-in non-volatile RAM.

Data Memory Area (D), Holding Area (H) and Counter Area (C) are retained even if the power supply is interrupted with no battery installed.

Built-in Flash Memory

The built-in Flash Memory is the backup memory for user program, parameters, program source, comment and Data Memory backed up using control bits in the Auxiliary Area.

Data is retained even if the power supply is interrupted.

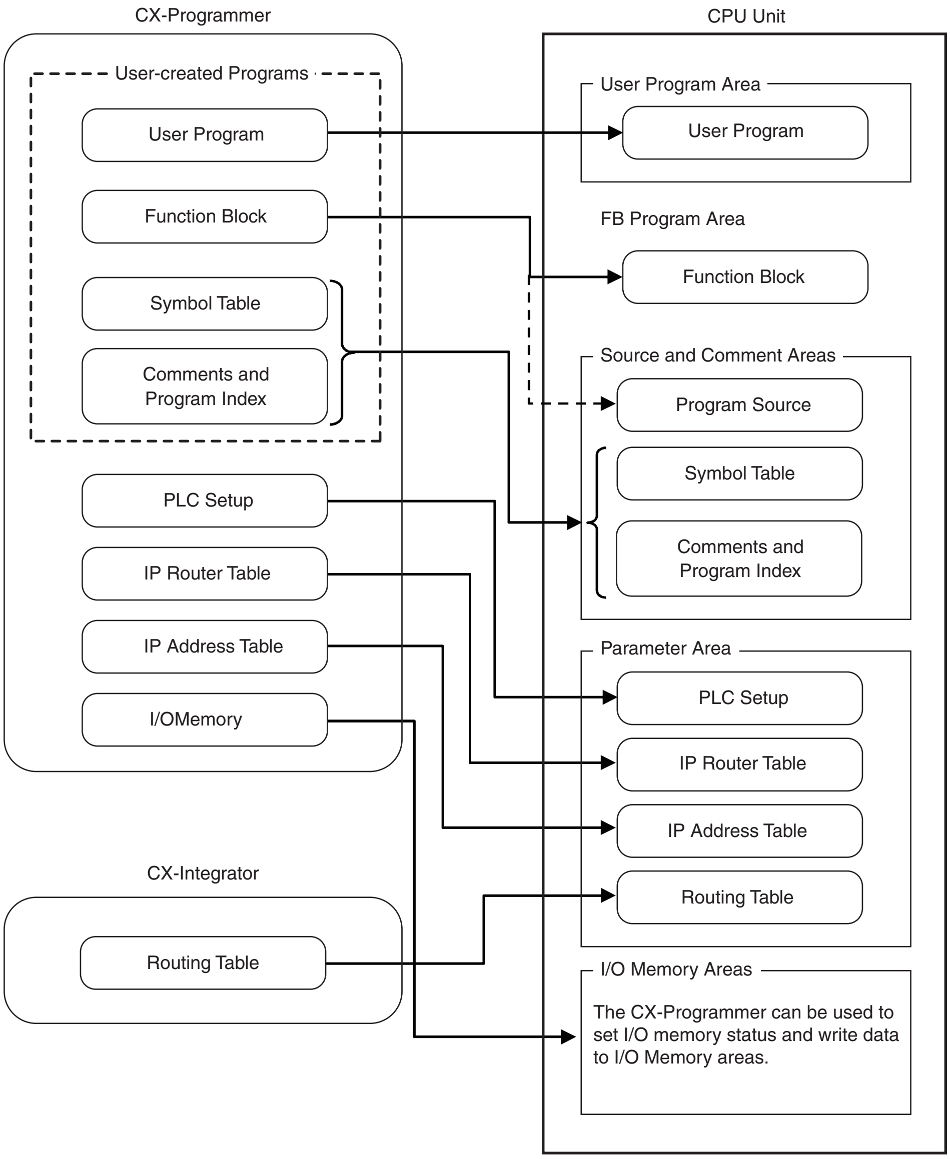
2-1-2 Memory Areas and Stored Data

The following table lists the CPU Unit memory areas and the data stored in each area.

Memory area and stored data	Details	Built-in RAM	Built-in non-volatile RAM	Built-in Flash Memory
User Program Area	The User Program Area stores the object code for executing the user program that was created using the CX-Programmer.	Stored	Not stored	Stored
FB Program Area	The FB Program Area stores the function block that was created using the CX-Programmer.	Stored	Not stored	Stored
Parameter Area		Stored	Not stored	Stored
PLC Setup	Various initial settings are made in the PLC Setup using software switches. Refer to *Section 7 PLC Setup*.			
Routing Table	Routing tables are network parameters for FINS communications. They are specified using the CX-Integrator. Refer to the *CX-Integrator Operation Manual* (Cat. No. W464)			
IP Router Table	IP router table is used to store the IP address of the relay IP router when the device communicates through the IP router.			
IP Address Table	Definition table of the relationship between FINS node address and IP address.			
Source and Comment Areas		Not stored	Not stored	Stored
Program Source	The source code of the program (ladder language, in task/function block)			
Symbol Table	The symbol table contains symbols created using the CX-Programmer (symbol names, addresses, and I/O comments).			
Comments	Comments are created using the CX-Programmer and include annotations and row comments.			
Program Index	The program index provides information on program sections created using the CX-Programmer, as well as program comments.			
I/O Memory Areas	The I/O Memory Areas are used for reading and writing from the user programs.It is partitioned into the following regions according to purpose. • Regions where data is cleared when the power supply to the CPU Unit is reset, and regions where data is automatically retained without battery even if the power supply is turned OFF. • Regions where data are exchanged with other Units, and regions that are used internally.	Not stored	Stored	Not stored
	DM Area words backed up to backup memory (built-in Flash Memory) using control bits in the Auxiliary Area.	Not stored	Stored	Stored

2-1-3 Transferring Data from a Programming Device

Data that has been created using the CX-Programmer or CX-Integrator is transferred to the internal memory in the CPU Unit as shown in the following diagram.



2-1-4 Backup

The CPU Unit will access the backup memory in the following process.

- The program, parameters, comments, program index or symbol table are transferred from the CX-Programmer and CX-Integrator.
- The program is changed during online editing.
- DM backup is operated by the Auxiliary Area.

During these processes, BKUP LED will light, indicating that the CPU Unit is being backed up.

There are the following limitations during backup.

- The operation mode cannot be switched from PROGRAM mode to MONITOR/RUN mode.
- If the power supply is interrupted when the program or parameters are being transferred or backed up, memory error may occur when the power supply is turned ON next time.
- If the power supply is interrupted when the DM area is being backed up, the reading of backed up DM area will fail when the power supply is turned ON next time.

3

CPU Unit Operation

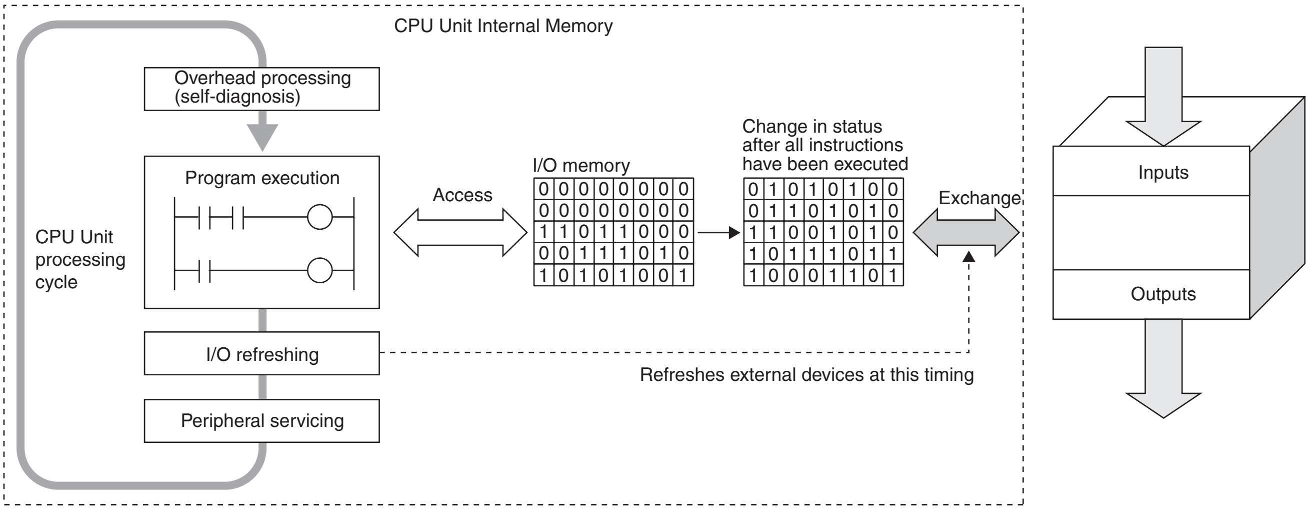
This section describes the operation of the CP2E CPU Unit. Make sure that you understand the contents of this section completely before writing ladder programs.

3-1 CPU Unit Operation

This section gives an overview of the CPU Unit operation, describes the operating modes, and explains how the Unit operates when there is a power interruption.

3-1-1 Overview of CPU Unit Operation

The CPU Unit reads and writes data to the internal I/O memory areas while executing user ladder programs by executing the instructions in order one at a time from the start to the end.



Overhead Processing (Self-diagnosis)

Self-diagnosis, such as an I/O bus check, is performed.

Ladder Program Execution

Instructions are executed from the beginning of the program and I/O memory is refreshed.

I/O Refresh

Data to and from external devices, such as sensors and switches, directly connected to the built-in I/O terminals and expansion I/O terminals, is exchanged with data in the I/O memory of the PLC. This process of data exchange is called the I/O refresh.

Peripheral Servicing

Peripheral servicing is used to communicate with devices connected to the communications port or for exchanging data with the CX-Programmer.

Cycle Time

The cycle time is the time between one I/O refresh and the next.

Additional Information

The average cycle time during operation will be displayed in the status bar on the bottom right of the Ladder Program Window on the CX-Programmer.

I/O Memory

These are the PLC memory areas that are accessed by the ladder programs. SYSMAC PLCs refer to these areas as the I/O memory. It can be accessed by specifying instruction operands. There are words in the I/O memory area where data is cleared and words where data is retained when recovering from a power interruption. There are also words that can be set to be cleared or retained. Refer to *Section 5 I/O Memory*.

3-1-2 CPU Unit Operating Modes

Overview of Operating Modes

CPU Units have the following three operating modes.

PROGRAM mode:	The programs are not executed in PROGRAM mode.This mode is used for the initial settings in PLC Setup, transferring ladder programs, checking ladder programs, and making prepartions for executing ladder programs such as force-setting/resetting bits.
MONITOR mode:	In this mode, it is possible to perform online editing, force-set/reset bits, and change I/O memory present values while the ladder programs are being executed. Adjustments during trial operation are also made in this mode.
RUN mode:	This is the mode in which the ladder program is executed. Some operations are disabled during this mode. It is the startup mode at initial value when the CPU Unit is turned ON.

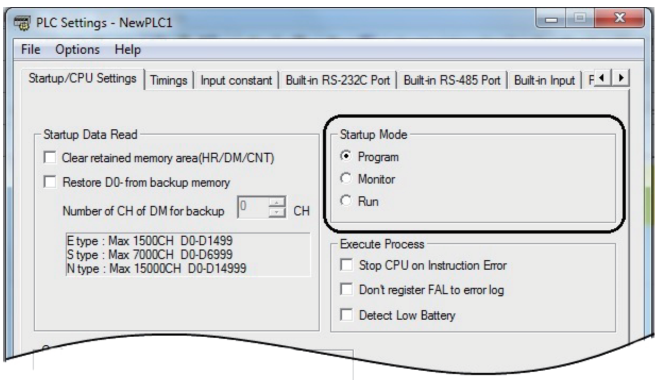
Changing the Operating Mode

The operating mode can be changed from the CX-Programmer.

● Changing the Startup Mode

The default operating mode when the CPU Unit is turned ON is RUN mode.

To change the startup mode to PROGRAM or MONITOR mode, set the desired mode in Startup Setting in PLC Setup from the CX-Programmer.



● Changing the Operating Mode after Startup

Use one of the following procedures.

- Select PROGRAM, MONITOR, or RUN from the Startup Mode Menu.
- Right-click the PLC in the project tree, and then select PROGRAM, MONITOR, or RUN from the Startup Mode Menu.

Operating Modes and Operation

The following table lists status and operations for each mode.

<table>
<tr><th colspan="3">Operating mode</th><th>PROGRAM</th><th>MONITOR</th><th>RUN</th></tr>
<tr><td colspan="3">Ladder program execution</td><td>Stopped</td><td>Executed</td><td>Executed</td></tr>
<tr><td colspan="3">I/O refresh</td><td>Executed</td><td>Executed</td><td>Executed</td></tr>
<tr><td colspan="3">External I/O status</td><td>OFF after changing to PROGRAM mode but can be turned ON from the CX-Programmer afterward.</td><td>Controlled by the ladder programs.</td><td>Controlled by the ladder programs.</td></tr>
<tr><td rowspan="2">I/O memory</td><td colspan="2">Non-retained memory</td><td>Cleared</td><td rowspan="2">Controlled by the ladder programs.</td><td rowspan="2">Controlled by the ladder programs.</td></tr>
<tr><td colspan="2">Retained memory</td><td>Retained</td></tr>
<tr><td rowspan="10">CX-Programmer operations</td><td colspan="2">I/O memory monitoring</td><td>Yes</td><td>Yes</td><td>Yes</td></tr>
<tr><td colspan="2">Ladder program monitoring</td><td>Yes</td><td>Yes</td><td>Yes</td></tr>
<tr><td rowspan="2">Ladder program transfer</td><td>From CPU Unit</td><td>Yes</td><td>Yes</td><td>Yes</td></tr>
<tr><td>To CPU Unit</td><td>Yes</td><td>No</td><td>No</td></tr>
<tr><td colspan="2">Checking programs</td><td>Yes</td><td>No</td><td>No</td></tr>
<tr><td colspan="2">Setting the PLC Setup</td><td>Yes</td><td>No</td><td>No</td></tr>
<tr><td colspan="2">Changing ladder programs</td><td>Yes</td><td>Yes</td><td>No</td></tr>
<tr><td colspan="2">Forced-set/reset operations</td><td>Yes</td><td>Yes</td><td>No</td></tr>
<tr><td colspan="2">Changing timer/counter PV</td><td>Yes</td><td>Yes</td><td>No</td></tr>
<tr><td colspan="2">Change I/O memory PV</td><td>Yes</td><td>Yes</td><td>No</td></tr>
</table>

The Retaining of I/O Memory When Changing the Operating Mode

<table>
<tr><th rowspan="2">Mode changes</th><th>Non-retained areas</th><th>Retained areas</th></tr>
<tr><td>• I/O bits
• Serial PLC Link Words
• Work bits
• Timer PV/Completion Flags
• Index Registers
• Data Registers
(Auxiliary Area bits/words are retained or not retained depending on the address.)</td><td>• Holding Area
• DM Area
• Counter PV and Completion Flags
(Auxiliary Area bits/words are retained or not retained depending on the address.)</td></tr>
<tr><td>RUN or MONITOR to PROGRAM</td><td>Cleared*</td><td>Retained</td></tr>
<tr><td>PROGRAM to RUN or MONITOR</td><td>Cleared*</td><td>Retained</td></tr>
<tr><td>RUN to MONITOR or MONITOR to RUN</td><td>Retained</td><td>Retained</td></tr>
</table>

* The data is cleared when the IOM Hold Bit is OFF. The outputs of the Output Units will be turned OFF when a fatal error is occurred, regardless of the status of the IOM Hold Bit, and the status of the output bits in CPU Unit's I/O memory is retained.

Refer to *Section 5 I/O Memory* for details on the I/O memory.

3-2 Backing Up Memory

This section describes backing up the CP2E CPU Unit memory areas.

3-2-1 CPU Unit Memory Configuration

Data backup to the CP2E CPU Unit's built-in memory describes as below.

● Ladder Programs and Parameter Area

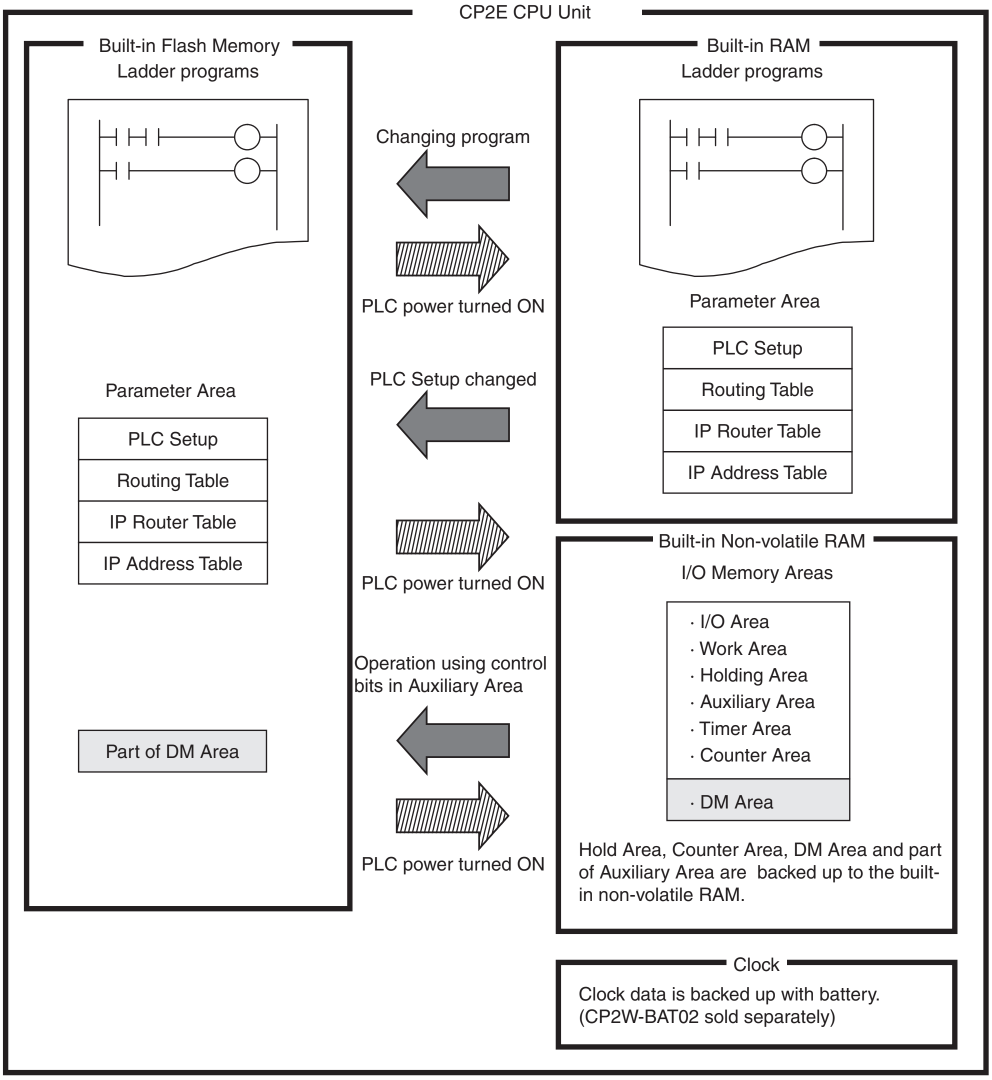
Automatically backed up to the built-in Flash Memory whenever changed.

● DM Area in the I/O Memory

Data in specified words of the DM Area can be backed up to the built-in Flash Memory by using bits in the Auxiliary Area. Other words are not backed up to the built-in Flash Memory.

● Other Areas in the I/O Memory (Including Holding Area Data, Counter PVs, and Counter Completion Flags)

Automatically backed up to the built-in non-volatile RAM with no battery installed.



3-2-2 Backing Up Ladder Programs and Parameter Area

Ladder programs and the parameter area are automatically backed up to and restored from the built-in Flash Memory.

● Backing Up Memory

Ladder programs and parameter area are backed up to the built-in Flash Memory by transferring them from the CX-Programmer or writing them using online editing.

● Restoring Memory

Ladder programs and parameter area are automatically transferred from the built-in Flash Memory to the RAM memory when power is turned ON again or at startup.

Precautions for Safe Use

The BKUP indicator on the front of the CPU Unit turns ON when data is being written to the built-in Flash Memory. Never turn OFF the power supply to the CPU Unit when the BKUP indicator is lit.

3-2-3 I/O Memory Backup

Hold Area, Counter Area, DM Area and part of Auxiliary Area are automatically backed up to the built-in non-volatile RAM. DM Area can be backed up to the built-in Flash Memory by using bits in the Auxiliary Area.

<table>
<tr><th colspan="2">Area</th><th>Backup to built-in Flash Memory</th><th>Status at startup</th></tr>
<tr><td colspan="2">CIO Area</td><td rowspan="8">Not backed up.</td><td rowspan="5">Cleared to all zeros.</td></tr>
<tr><td colspan="2">Work Area (W)</td></tr>
<tr><td colspan="2">Timer Area (T)</td></tr>
<tr><td colspan="2">Index Register Area (IR)</td></tr>
<tr><td colspan="2">Data Register Area (DR)</td></tr>
<tr><td colspan="2">Holding Area (H)</td><td rowspan="2">The values are retained immediately before power interruption.*</td></tr>
<tr><td colspan="2">Counter Area (C)</td></tr>
<tr><td colspan="2">Auxiliary Area (A)</td><td>Retained or not retained depending on the address. The values of the retained area are retained immediately before power interruption.</td></tr>
<tr><td rowspan="2">DM Area (D)</td><td>Number of words starting from D0 set in the Number of CH of DM for backup Box in the Startup Data Read Area in the PLC Settings.</td><td>The specified number of words starting from D0 is backed up by turning ON A751.15 (DM Backup Save Start Bit).</td><td>The specified number of words starting from D0 is restored from the built-in Flash Memory if the Restore D0- from backup memory Check Box is selected in the Startup Data Read Area in the PLC Settings.</td></tr>
<tr><td>Ranges not given above.</td><td>Not backed up.</td><td>The values are retained immediately before power interruption.*</td></tr>
</table>

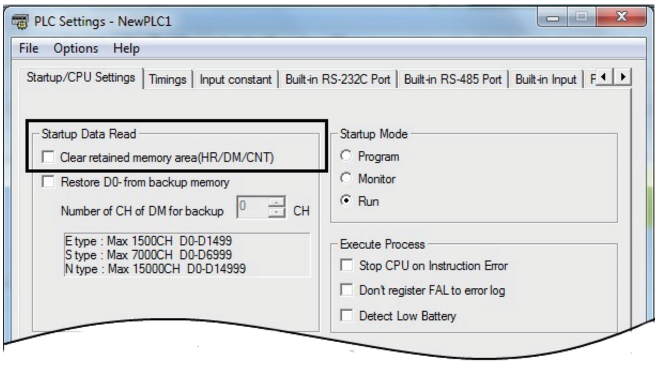
* These areas will be cleared when the power supply is turned ON if the Clear retained memory area (HR/DM/CNT) Check Box is selected in the PLC Settings.

3-2-4 Initializing I/O Memory at Startup

For CP2E CPU Units, the held areas in I/O memory (Holding Area, Counter PVs, Counter Completion Flags and DM Area) are automatically retained in the built-in non-volatile RAM with no battery installed. Use the following way to clear these areas when the power supply is turned ON.

● Clearing All Held Areas to Zero at Startup

Select the Clear retained memory area (HR/DM/CNT) Check Box in the PLC Settings.



Note If the Restore D0- from backup memory Check Box is selected, only the specified words in the DM Area will be restored from the built-in Flash Memory when the power supply is turned ON.

4

Understanding Programming

This section provides basic information on ladder programming for CP2E CPU Units.

4

4-1 Programming

4-1-1 User Programs

Structure of User Programs

User programs are created by using the CX-Programmer.

The user programs consist of the following parts.

- Programs
  A program consists of more than one instruction and ends with an END instruction.
- Tasks (Smallest Executable Unit)
  A program is assigned to an interrupt task to execute it. (In the CX-Programmer, the interrupt task number is specified in the program properties.)
  Tasks include cyclic tasks (executed with normal cyclic processing), interrupt tasks (executed when interrupt conditions have been completed) and scheduled interrupt tasks (executed at specified intervals).
  The CP2E can use only one cyclic task.
- Sections
  When creating and displaying programs with the CX-Programmer, the one program can be divided into any number of parts.
  Each part is called a section.
  Sections are created mainly to make programs easier to understand.
- Subroutines and Function Blocks
  You can create subroutines and function blocks within a program.

User Program Data

The user programs are saved in a project file (.CXP) for the CX-Programmer along with other parameters, such as the symbol table, PLC Setup data, and I/O memory data.

Programming Languages

Programs can be written using only ladder programs.

4-1-2 Program Capacity

The maximum program capacities of the CP2E CPU Units for all ladder programs (including symbol table and comments) are given in the following table.

The total number of steps must not exceed the maximum program capacity.

Unit type	Model numbers	Program capacity
E□□-type CPU Unit	CP2E-E□□D□-□	4K steps
S□□-type CPU Unit	CP2E-S□□D□-□	8K steps
N□□-type CPU Unit	CP2E-N□□D□-□	10K steps

It is possible to check the program size by selecting ***Program - Memory View*** in the CX-Programmer.

The size of a ladder instruction depends on the specific instruction and operands that are used.

4-1-3 Basics of Programming

This section describes the basics of programming for the CP2E.

Basic Concepts of Ladder Programming

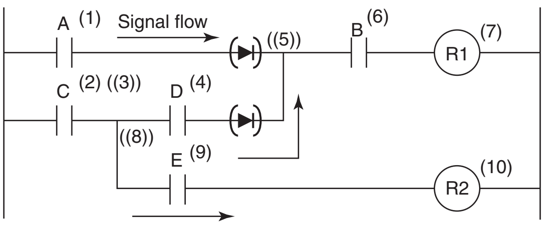
Instructions are executed in the order that they are stored in memory (i.e., in the order of the mnemonic code). Be sure you understand the concepts of ladder programming, and write the programs in the proper order.

● Basic Points in Creating Ladder Programs

Order of Ladder Program Execution

When the ladder diagram is executed by the CPU Unit, the execution condition (i.e., power flow) flows from left to right and top to bottom.
The flow is different from that for circuits that consist of hard-wired control relays.
For example, when the diagram in figure A is executed by the CPU Unit, power flows as though the diodes in brackets were inserted so that output R2 is not controlled by input condition D.
The actual order of execution is indicated on the right with mnemonics.
To achieve operation without these imaginary diodes, the diagram must be rewritten. Also, the power flow in figure B cannot be programmed directly and must be rewritten.

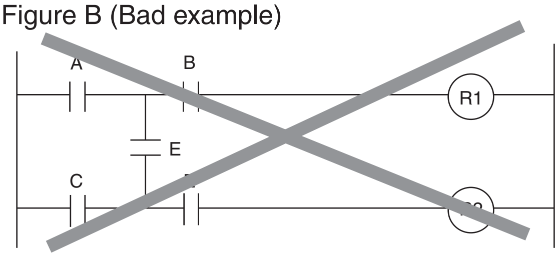
Figure A (Good example)



Order of execution (mnemonics)

(1) LD A	(6) AND B
(2) LD C	(7) OUT R1
(3) OUT TR0	(8) LD TR0
(4) AND D	(9) AND E
(5) OR LD	(10) OUT R2

Figure B (Bad example)

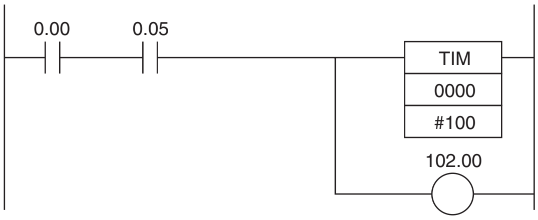


● Number of Times Bits Can be Used and Connection Method

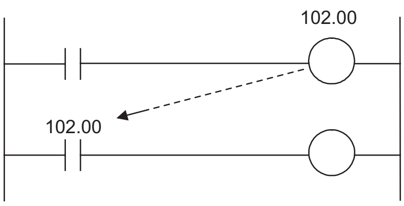
- There is no limit to the number of I/O bits, work bits, timers, and other input bits that can be used. Program structure should be kept as clear and simple as possible to make the programs easier to understand and maintain even if it means using more input bits.

- There is no limit to the number of input conditions that can be connected in series or in parallel on the rungs.

- Two or more OUT instructions can be connected in parallel.

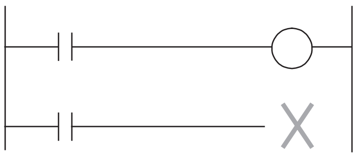


- Output bits can also be used in input conditions.

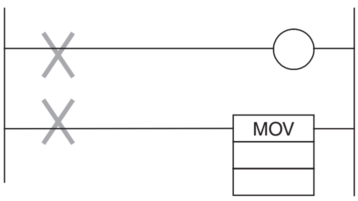


● Ladder Programming Restrictions

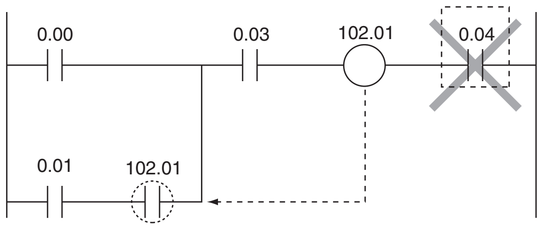
- A rung error will occur if a ladder program is not connected to both bus bars.
  The ladder program must be connected to both bus bars so that the execution condition will flow from the left bus bar to the right bus bar.
  If the rungs are not connected to both bus bars, a rung error will occur during the program check on the CX-Programmer and program transfer will be impossible.



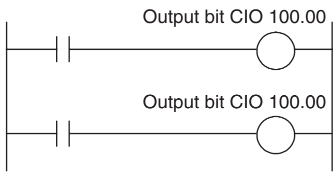
- A rung error will occur if the instruction shown below is made to directly connect to the bus bar without an input condition.
  OUT instructions, timers, counters, and other output instructions cannot be connected directly to the left bus bar.
  If one of these instructions is connected directly to the left bus bar, a rung error will occur and program transfer will be impossible.



- A location error will occur if an instruction is not connected directly to the right bus bar.
  An input condition cannot be inserted after an OUT instruction or other output instruction. The input condition must be inserted before an OUT instruction or other output instruction. If it is inserted after an output instruction, then a location error will occur during the program check in the CX-Programmer.



- A warning will occur if the same output bit is used more than once in an OUT instruction.
  One output bit can be used in one instruction only. Instructions in a ladder program are executed in order from the top rung in each cycle. The result of an OUT instruction in a lower rung will be eventually saved in the output bit. The results of any previous instructions controlling the same bit will be overwritten and not output.



4-2 Tasks, Sections, and Symbols

4-2-1 Overview of Tasks

There are basically two types of tasks.

Task settings must be made to use interrupt tasks with a CP2E CPU Unit.

Task type	Description	Applicable programming language	Execution condition
Cyclic task	Executed once per cycle	Ladder diagram	Only one for the CP2E. (Normally, the user does not have to consider this.)
Interrupt tasks	Executed when a specific condition occurs. The process being executed is interrupted.	Ladder diagram	An interrupt task is placed into READY status when the interrupt condition occurs. A condition can be set for the following interrupt tasks. • Scheduled interrupt tasks • I/O interrupt tasks

4-2-2 Overview of Sections

With the CX-Programmer, programs can be created and displayed in functional units called sections.

Any program in a task can be divided into sections.

Sections improve program legibility and simplifies editing.

4-2-3 Overview of Symbols

Symbols

I/O memory area addresses or constants can be specified by using character strings registered as symbols.

The symbols are registered in the symbol table of the CX-Programmer.

Programming with symbols enables programming with names without being aware of the addresses.

The symbol table is saved in the CX-Programmer project file (.CXP) along with other parameters, such as the user programs.

Symbol Types

There are two types of symbols that can be used in programs.

● Global Symbols

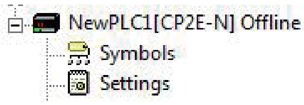
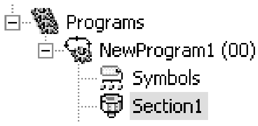
Global symbols can be accessed from all ladder programs in the PLC.

● Local Symbols

Local symbols can be accessed from only one task. They are assigned to individual tasks.

Addresses are allocated to symbols using one of the following methods.

- User Specified allocation
- Automatic allocation using the CX-Programmer
  The area of memory used for automatic allocations is set by selecting ***Memory Allocation - Automatic Address Allocation*** from the PLC Menu in the CX-Programmer.

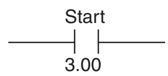
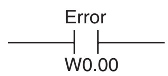
Types of symbols	Project tree in the CX-Programmer	Scope			Address and I/O comment (without a symbol name)
		Access using symbols from a network	Access from other tasks	Access from the local task	
Global symbols	PLC tree 	Not possible.	Possible.	Possible.	Supported
Local symbols	Program tree 		Not possible.	Possible.	Not supported

Note "Global" and "local" indicate only the applicable scope of the symbol.
They have nothing to do with the applicable scope of memory addresses.
Therefore, a warning but not an error will occur in the following cases, and it will be possible to transfer the user program.

- The same addresses is used for two different local symbols.
- The same addresses is used for a global symbol and a local symbol.

Additional Information

In programs in the CX-Programmer, global symbols and local symbols can be identified by the following character colors and symbol icons.

Classification	Display color	Example (default color)
Global symbols	Black (default)	
Local symbols	Blue (default)	

Select *Tools - Options*, and select *Local Symbols* or *Global Symbols in Appearance* to change the color.

4-3 Function Blocks

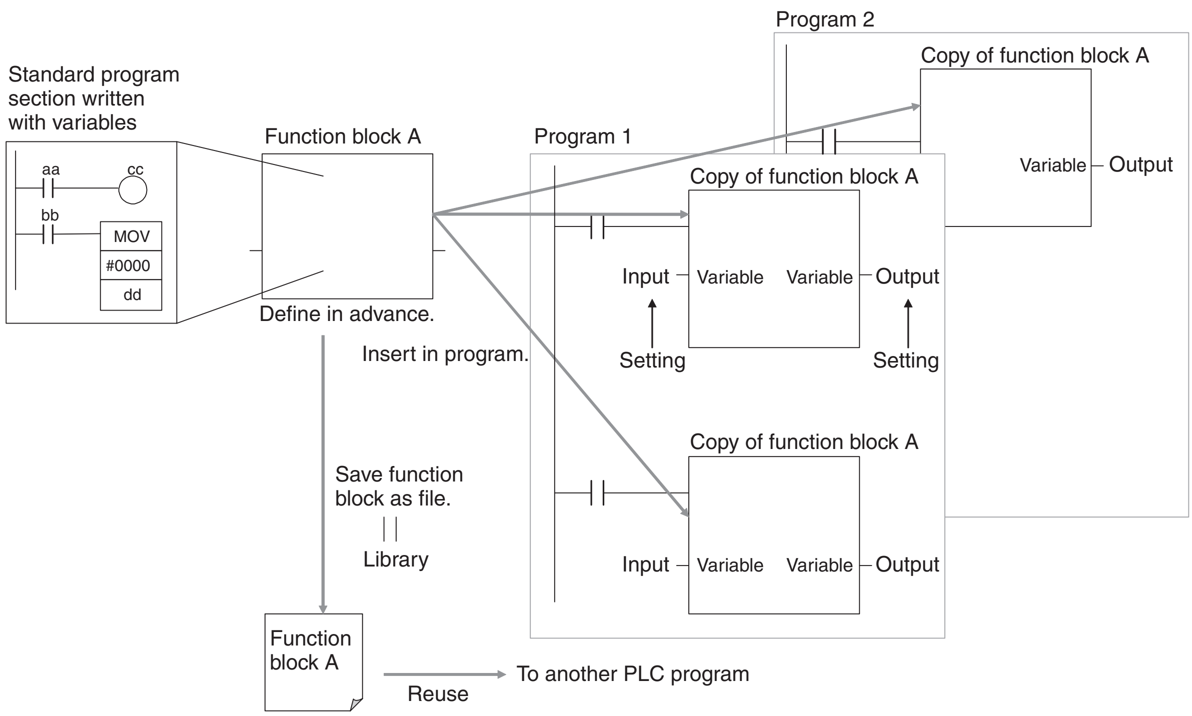
Function blocks can be used in programming SYSMAC CP-series PLCs.

4-3-1 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 I/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.



4-3-2 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

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

● Easy-to-read “Block Box” Design

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

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.

● Reduced Coding Errors

Coding mistakes can be reduced, because blocks that have already been debugged can be reused.

● Data Protection

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.)

● Improved Reusability through Programming with Variables

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.

Creating Libraries

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.

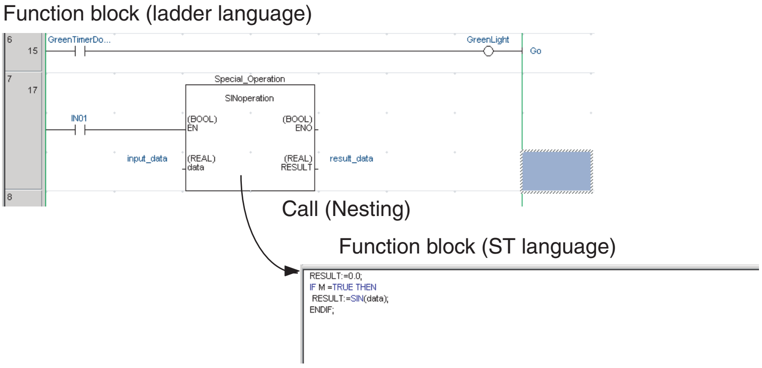
The function blocks are created with local variable names that are not tied to physical addresses, so new programs can be developed easily just by reading the definitions from the file and placing them in a new program.

Nesting Multiple Languages

Mathematical expressions can be entered in structured text (ST) language.

Nesting function blocks is supported. 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).

4-3-3 Function Block Specifications

Item		Specifications	
Defining and creating function blocks	Number of function block definitions	64 max.	
	Function block names	32 characters max.	
	Variables	Variable names	15,000 characters max.
		Variable types	Input variables (Inputs), output variables (Outputs), internal variables (Internals), and external variables (Externals)
		Number of variables used in a function block (not including internal variables, external variables, EN, and EN0)	Maximum number of variables per function block definition • Input-output variables: 16 max. • Input variables + input-output variables: 64 max. • Output variables + input-output variables: 64 max.
		Allocation of addresses used by variables	Automatic allocation (The allocation range can be set by the user.)
		Actual address specification	Supported
		Array specifications	Supported (one-dimensional arrays only and only for internal variables)
	Language	Function blocks can be created in ladder programming language or structured text.	
Creating instances	Number of instances	128 max.	
	Instance names	15,000 characters max.	
FB Program Area		E□□-type: 4K steps, S□□-type: 8K steps, N□□-type: 10K steps	
Storing function blocks as files	Project files	The project file (.cxp/cxt) Includes function block definitions and instances.	
	Program symbol files	The file memory program file (*.obj) includes function block definitions and instances.	
	Function block library files	Each function block definition can be stored as a single file (.cxf) for reuse in other projects.	

Data Types that Can be Used in Function Blocks

Data type	Content	Size	Availability in CP2E				
			Inputs	Outputs	In Out	Internals	Externals
BOOL	Bit data	1	Yes	Yes	No	Yes	Yes
INT	Integer	16	Yes	Yes	No	Yes	Yes
DINT	Double integer	32	Yes	Yes	No	Yes	Yes
LINT	Long (4-word) integer	64	Yes	Yes	No	Yes	Yes
UINT	Unsigned integer	16	Yes	Yes	No	Yes	Yes
UDINT	Unsigned double integer	32	Yes	Yes	No	Yes	Yes
ULINT	Unsigned long (4-word) integer	64	Yes	Yes	No	Yes	Yes
REAL	Real number	32	Yes	Yes	No	Yes	Yes
LREAL	Long real number	64	No	No	No	No	No
WORD	16-bit data	16	Yes	Yes	No	Yes	Yes
DWORD	32-bit data	32	Yes	Yes	No	Yes	Yes
LWORD	64-bit data	64	Yes	Yes	No	Yes	Yes
STRING	Text string data	Variable	No	No	No	No	No
TIMER	Timer	Flag: 1 PV: 16	No	No	No	Yes	Yes
COUNTER	Counter	Flag: 1 PV: 16	No	No	No	Yes	Yes
Array	-	-	No	No	No	No	No
Structure	-	-	No	No	No	No	No

Number of Steps Used by Function Blocks

When function blocks are used, memory is required for the following two items:

1. Function block definitions
2. Instance generation processing when function blocks are pasted into the user program as function block instances

Therefore, the number of steps used in memory will increase with the number of instances of function blocks created in the program (item 2).

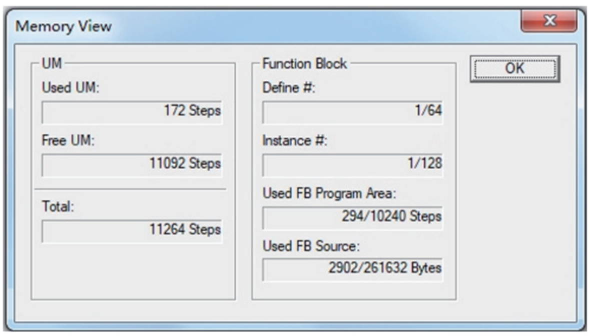
Creating instances of function blocks (i.e., calling function block definitions and transferring parameters) uses user program memory.

Function block definitions use memory in the FB Program Area, a special area for function blocks. If the capacity of the FB Program Area is exceeded, the user program area is used.

Checking Function Block Memory Usage

Select ***View*** – ***Memory View*** from CX-Programmer.

The following Memory View Dialog Box will be displayed.



4-3-4 ST Language

CP2E can use the ST (Structured Text) language within function blocks.
The standard control statements, operators, and functions make the ST language ideal for mathematical processing that is difficult to write in ladder programming.

Additional Information

For details on ST programming specifications, notation, and input procedures, refer to the *CX-Programmer Operation Manual: Function Blocks and Structured Text* (Cat. No. W447).

● Features of ST Programming

- There are many control statements available, such as loop statements and IF-THEN-ELSE statements, many operators such as arithmetic operators, comparison operators, and AND/OR operators, as well as many mathematical functions, string extract and merge functions, Memory Card processing functions, string transfer functions, and trigonometric functions.
- Programs can be written like high-level languages such as C, and comments can be included to make the program easy to read.
- ST programs can be uploaded and downloaded just like ordinary programs, but ST program can only be used in function blocks.
- One-dimensional array variables are supported for easier data handling in applications.

Control Statements, Operators and Functions that Can be Used in ST Programs

The control statements, operators and functions that can be used in CP2E are shown as follows. A part of operators and functions are not available in CP2E.

The data types used in ST programs are the same as that can be used in function blocks.

● Control Statements

Control statements	Availability in CP2E
IF, THEN, ELSEIF, ELSE, END_IF	Yes
CASE, ELSE, END_CASE	Yes
FOR, TO, BY, DO, END_FOR	Yes
WHILE, DO, END_WHILE	Yes
REPEAT, UNTIL, END_REPEAT	Yes
EXIT	Yes
RETURN	Yes

● Operators

Operators	Symbol	Availability in CP2E
Parentheses and brackets	(expression), array[index]	Yes
Function evaluation	identifier	Yes
Exponential	**	No
Complement	NOT	Yes
Multiplication	*	Yes
Division	/	Yes
Addition	+	Yes
Subtraction	-	Yes
Comparisons	<, >, <=, >=	Yes
Equality	=	Yes
Non-equality	<>	Yes
Boolean AND	%	Yes
Boolean AND	AND	Yes
Boolean exclusive OR	XOR	Yes
Boolean OR	OR	Yes

● Standard Functions

Function type	Availability in CP2E
Numerical Functions	No*
Arithmetic Functions	No
Data Type Conversion Functions	Yes
Number-String Conversion Functions	No
Data Shift Functions	Yes
Data Control Functions	Yes
Data Selection Functions	Yes

* Only MOD Function is available.

● OMRON Expansion Functions

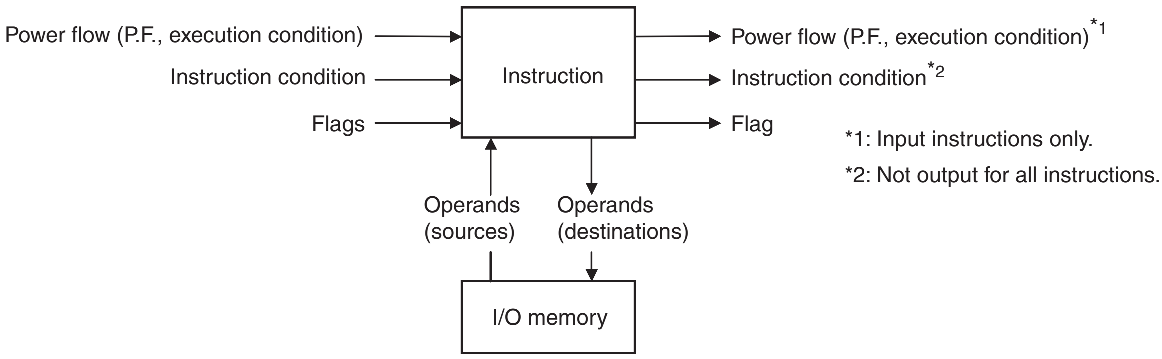
Function type	Availability in CP2E
Memory Card Functions	No
Communications Functions	No
Angle Conversion Functions	No
Timer/Counter Functions	No

4-4 Programming Instructions

4-4-1 Basic Understanding of Instructions

Structure of Instructions

Programs consist of instructions. The conceptual structure of the inputs to and outputs from an instruction is shown in the following diagram.

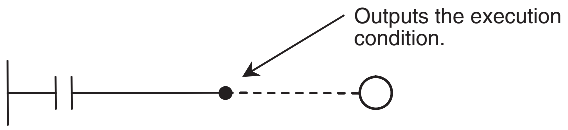


- **Power Flow**

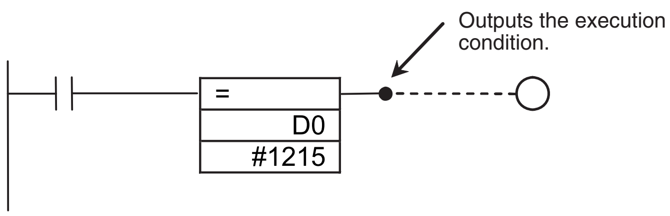
The power flow is the execution condition that is used to control the execution and instructions when programs are executing normally. In a ladder program, power flow represents the status of the execution condition.

Input Instructions

- Load instructions indicate a logical start and outputs the execution condition.

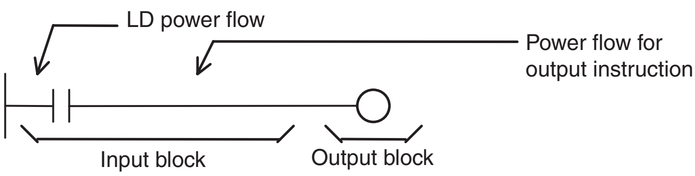


- Intermediate instructions input the power flow as an execution condition and output the power flow to an intermediate or output instruction.



Output Instructions

Output instructions execute all functions, using the power flow as an execution condition.



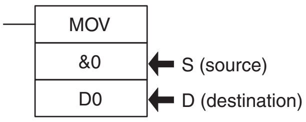
4-4-2 Operands

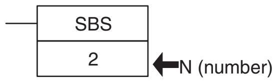
Operands specify preset instruction parameters that are used to specify I/O memory area contents or constants. Operands are given in boxes in the ladder programs.

Addresses and constants are entered for the operands to enable executing the instructions.

Operands are classified as source, destination, or number operands.

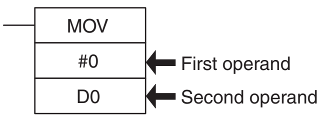
Example:





Operand type		Operand symbol	Description	
Source operand	Specifies the address of the data to be read or a constant.	S	Source operand	Source operand other than control data (C)
		C	Control data	Compound data in a source operand that has different meanings depending on bit status.
Destination operand (results)	Specifies the address where data will be written.	D	–	
Number	Specifies a particular number used in the instruction, such as a subroutine number.	N	With numbers, it is not possible to specify an address for indirect specification (except for jump instruction numbers).	

Operands are also called the first operand, second operand, and so on, starting from the top of the instruction.

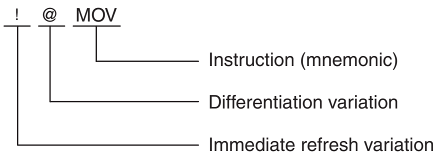


4-4-3 Instruction Variations

The following variations are available for instructions to differentiate executing conditions and to refresh data when the instruction is executed (immediate refreshing).

Variation		Symbol	Description
No variation used.		–	These instructions are executed once every cycle while the execution condition is satisfied.
Differentiation variations	ON	@	These instructions are executed only once when the execution condition turns ON.
	OFF	%	These instructions are executed only once when the execution condition turns OFF.
Immediate refreshing		!	Data in the built-in I/O area specified by the operands is refreshed when the instruction is executed.

Example:



4-4-4 Execution Conditions

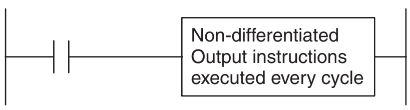
The following two types of basic and special instructions can be used.

- Non-differentiated instructions: Executed every cycle
- Differentiated instructions: Executed only once

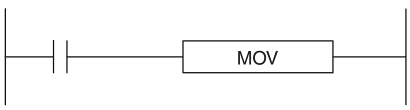
Non-differentiated Instructions

● Output Instructions (Instructions That Require Input Conditions)

These instructions are executed once every cycle while the execution condition is satisfied (ON or OFF).

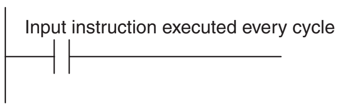


Example:

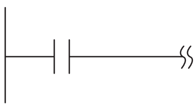


● Input Instructions (Logical Starts and Intermediate Instructions)

These instructions read bit status, make comparisons, test bits, or perform other types of processing every cycle. If the results are ON, the input condition is output (i.e., the execution condition is turned ON).



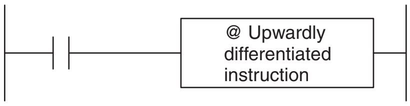
Example:



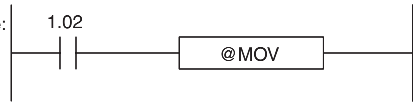
Input-differentiated Instructions

● Upwardly Differentiated Instructions (Instructions Preceded by @)

- Output Instructions
  The instruction is executed only during the cycle in which the execution condition changes from OFF to ON.
  The instruction is not executed in the following cycle.

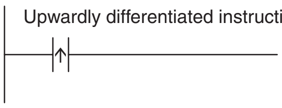


Example:

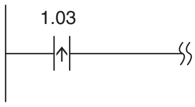


Executes the MOV instruction once when CIO 1.02 turns ON.

- Input Instructions (Logical Starts and Intermediate Instructions)
  The instruction reads bit status, makes comparisons, tests bits, or performs other types of processing every cycle and will output an ON execution condition (power flow) when the result changes from OFF to ON.
  The execution condition will turn OFF the next cycle.



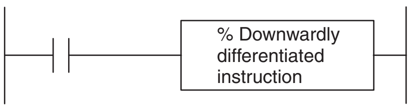
Example:



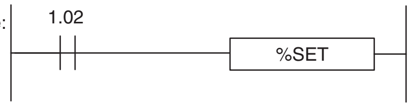
ON execution condition created for one cycle when CIO 1.03 turns ON.

● Downwardly Differentiated Instructions (Instruction Preceded by %)

- Output Instructions
  The instruction is executed only during the cycle in which the execution condition changes from ON to OFF.
  The instruction is not executed in the following cycle.

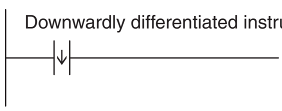


Example:

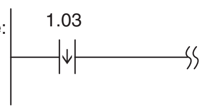


Executes the SET instruction once when CIO 1.02 turns OFF.

- Input Instructions (Logical Starts and Intermediate Instructions)
  The instruction reads bit status, makes comparisons, tests bits, or performs other types of processing every cycle and will output an ON execution condition (power flow) when the result changes from ON to OFF.
  The execution condition will turn OFF the next cycle.



Example:

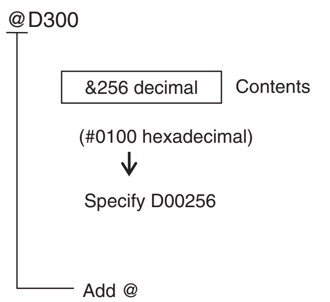
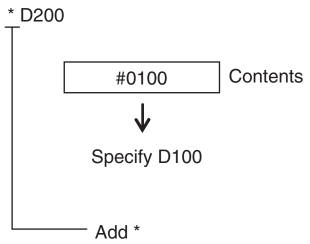


ON execution condition created for one cycle when CIO 1.03 turns ON.

4-4-5 Specifying Data in Operands

Specifying Addresses

Operand	Description	Example	Application examples
Specifying bit addresses	The word address and bit number are specified directly to specify a bit. □□□□ . □□ Bit number (00 to 15) Word address	1 . 02 Bit number 02 Word address CIO 1	1.02
Specifying word addresses	The word address is specified directly to specify a 16-bit word. □□□□ Word address	3 Word address CIO 3 D200 Word address D200	MOV 3 D200
Specifying offsets for bit addresses	In brackets, specify the number of bits to offset the specified starting bit address. □□□□.□□ □□ Offset Constant 0 to 15 or word address in I/O memory Starting bit address A symbol can also be specified for the starting bit address. Only Holding, Work, and DM Area addresses can be used regardless of whether a physical address or symbol is used. A constant or word address in I/O memory can be used for the offset. If a word address is specified, the contents of the word is used as the offset.	10.00[2] Number of bits to offset the address →Specify 10.02 Starting bit address 10.00 [W0] Number of bits to offset the address When W0 = &2→Specify 10.02 Starting bit address	10.00[2]
Specifying offsets for word addresses	In brackets, specify the number of words to offset the specified starting bit address. □□□□ [□□] Offset Constant of 0 or higher or word address in I/O memory Starting word address A symbol can also be specified for the starting word address. Only Holding, Work, and DM Area addresses can be used regardless of whether a physical address or symbol is used. A constant or word address in I/O memory can be used for the offset. If a word address is specified, the contents of the word is used as the offset.	D0[2] Number of words to offset the address →Specify D2 Starting word address D0 [W0] Number of bits to offset the address When W0 = &2→Specify D2 Starting word address	MOV 3 D0[200]

Operand	Description	Example	Application examples
Specifying indirect DM addresses in Binary Mode	An offset from the beginning of the DM Area is specified. The contents of the address will be treated as binary data (E☐☐-type CPU Unit 0000 to 4095, S☐☐-type CPU Unit 0000 to 8191, N☐☐-type CPU Unit 0000 to 16383) to specify the word address in DM Area. Add the @ symbol at the front to specify an indirect address in Binary Mode.		MOV #0001 @D300
Specifying indirect DM Addresses in BCD Mode	An offset from the beginning of the DM Area is specified. The contents of the address will be treated as BCD data (E☐☐-type CPU Unit 0000 to 4095, S☐☐-type CPU Unit 0000 to 8191, N☐☐-type CPU Unit 0000 to 16383) to specify the word address in the DM Area. Add an asterisk (*) at the front to specify an indirect address in BCD Mode.		MOV #0001 *D200

Note For Timer Completion Flags and Counter Completion Flags, there is no distinction between word addresses and bit addresses.

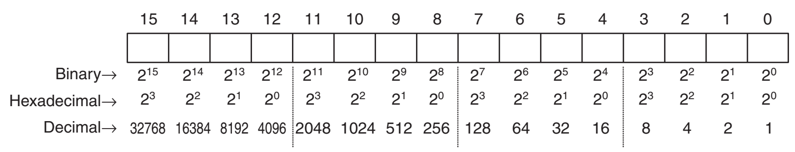
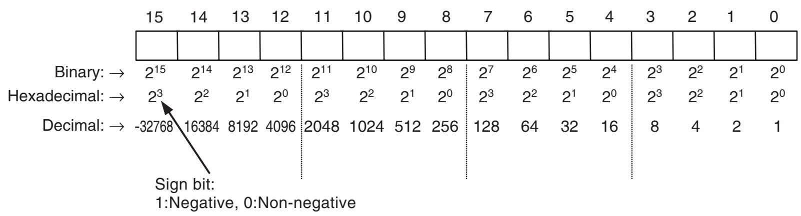
Operand	Description		Notation	Application examples
Specifying a register directly (See note.)	An index register (IR) or a data register (DR) is specified directly by specifying IR☐ (☐: 0 to 15) or DR☐ (☐: 0 to 15).		IR0	MOVR 1.02 IR0 Stores the PLC memory address for CIO 0010 in IR0.
			IR1	MOVR 10 IR1 Stores the PLC memory address for CIO 0010 in IR1.
Specifying an indirect address using a register (See note.)	Indirect address (No offset)	The bit or word with the PLC memory address contained in IR☐ will be specified. Specify ,IR☐ to specify bits and words for instruction operands.	,IR0	LD ,IR0 Loads the bit with the PLC memory address in IR0.
			,IR1	MOV #0001 ,IR1 Stores #0001 in the word with the PLC memory in IR1.
	Constant offset	The bit or word with the PLC memory address in IR☐ + or - the constant is specified. Specify +/- constant ,IR☐. Constant offsets range from -2048 to +2047 (decimal). The offset is converted to binary data when the instruction is executed.	+5,IR0	LD +5,IR0 Loads the bit with the PLC memory address in IR0 + 5.
			+31,IR1	MOV #0001 +31,IR1 Stores #0001 in the word with the PLC memory address in IR1 + 31.
	DR offset	The bit or word with the PLC memory address in IR☐ + the contents of DR☐ is specified. Specify DR☐ ,IR☐. DR (data register) contents are treated as signed-binary data. The contents of IR☐ will be given a negative offset if the signed binary value is negative.	DR0 ,IR0	LD DR0 ,IR0 Loads the bit with the PLC memory address in IR0 + the value in DR0.
			DR0 ,IR1	MOV #0001 DR0 ,IR1 Stores #0001 in the word with the PLC memory address in IR1 + the value in DR0.

Operand	Description		Notation	Application examples
	Auto-increment	The contents of IR□ is incremented by +1 or +2 after referencing the value as an PLC memory address. +1: Specify ,IR□+ +2: Specify ,IR□+ +	,IR0++	LD ,IR0 ++ Increments the contents of IR0 by 2 after the bit with the PLC memory address in IR0 is loaded.
			,IR1+	MOV #0001 ,IR1 + Increments the contents of IR1 by 1 after #0001 is stored in the word with the PLC memory address in IR1.
	Auto-decrement	The contents of IR□ is decremented by -1 or -2 after referencing the value as an PLC memory address. -1: Specify ,-IR□ -2: Specify ,--IR□	,--IR0	LD ,--IR0 After decrementing the contents of IR0 by 2, the bit with the PLC memory address in IR0 is loaded.
			,-IR1	MOV #0001 ,-IR1 After decrementing the contents of IR1 by 1, #0001 is stored in the word with the PLC memory address in IR1.

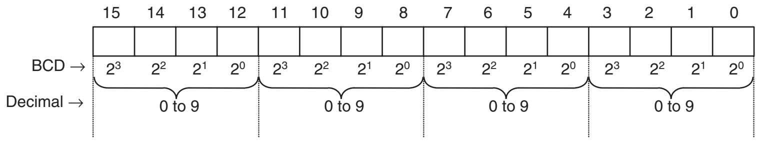
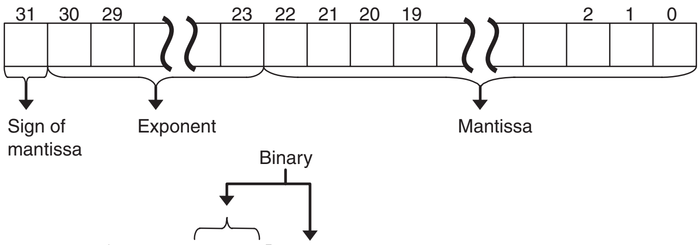
Note For specific application methods, refer to *4-6 Index Registers*.

4-4-6 Data Formats

The following table shows the data formats that the CP2E CPU Units can handle.

<table>
<tr><th>Type</th><th>Data format</th><th>Decimal equivalent</th><th>4-digit hexadecimal</th></tr>
<tr><td>Unsigned binary</td><td>
</td><td>&0 to
&65535</td><td>#0000 to
#FFFF</td></tr>
<tr><td rowspan="2">Signed binary</td><td rowspan="2">

The data is treated as 16-bit signed binary data using the leftmost bit as the sign bit. The value is expressed in 4-digit hexadecimal.
Positive numbers: If the leftmost bit is OFF, it indicates a non-negative value. For 4-digit hexadecimal, the value will be 0000 to 7FFF hex.
Negative numbers: If the leftmost bit is ON, it indicates a negative value. For 4-digit hexadecimal, the value be 8000 to FFFF hex. It will be expressed as the 2’s complement of the absolute value of the negative value (decimal). </td><td>Negative:
-1 to
- 32768</td><td>Negative:
#8000 to
#FFFF</td></tr>
<tr><td>Positive:
0 to 32767</td><td>Positive:
#0000 to
#7FFF</td></tr>
</table>

<table>
<tr><th>Type</th><th>Data format</th><th>Decimal equivalent</th><th>4-digit hexadeci-mal</th></tr>
<tr><td>BCD (binary coded deci-mal)</td><td>
</td><td>#0 to #9999</td><td>#0000 to #9999</td></tr>
<tr><td>Single-preci-sion floating-point decimal</td><td>

Value = $(-1)^{sign} \times 1.[\text{Mantissa}] \times 2^{\text{Exponent}}$
· Sign bit (bit 31): 1: Negative, 0: Positive
· Mantissa: The 23 bits from bit 00 to bit 22 contain the mantissa, i.e., the portion below the decimal point in 1. □□□□.....,in binary.
Indicates this value.
· The 8 bits from bit 23 to bit 30 contain the exponent. The exponent is expressed in binary as the n in 2^n. The actual value is 2^{n-127}.

This format conforms to the IEEE 754 standard for single-precision floating-point data. It is used only with instructions that convert or calculate floating-point data.
• Input using operands in the CX-Programmer as signed decimal or 32-bit hexadecimal with the # symbol.
• When inputting operands in the I/O Memory Edit/Monitor Window of the CX-Programmer as signed decimal values with seven digits or less, the value will be automatically converted to scientific notation (mantissa$\times 10^{\text{Exponent}}$) for setting and monitoring. Inputs must be made using scientific notation for values with eight or more digits.
Example: When -1234.00 is input, it will become -1.234000e+003 in scientific notation. For the mantissa$\times 10^{\text{Exponent}}$, the value before the e is the mantissa and the value after the e is the signed exponent.</td><td>*</td><td>–</td></tr>
</table>

* Data range for single-precision floating-point decimal: $-3.402823 \times 10^{38} \leq$ Value $\leq -1.175494 \times 10^{-38}$, 0, $+1.175494 \times 10^{-38} \leq$ Value $\leq 3.402823 \times 10^{38}$

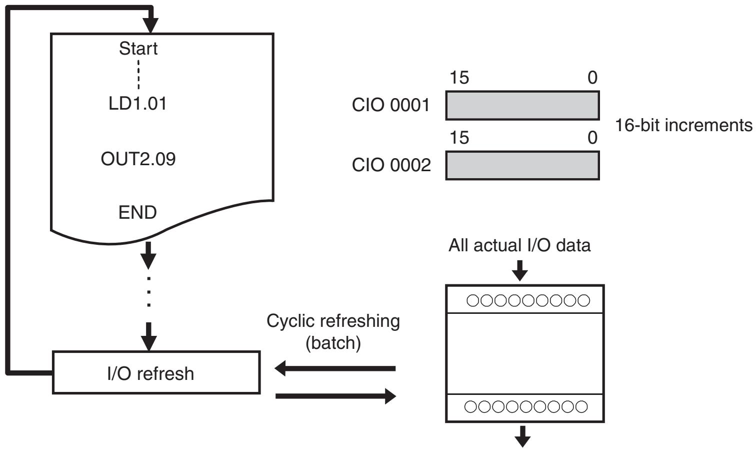
4-4-7 I/O Refresh Timing

The following methods are used to refresh external I/O.

- Cyclic refreshing
- Immediate refreshing (instructions with the ! variation and IORF)

Cyclic Refreshing

I/O is all refreshed after ladder programs are executed.



Execute an instruction with the immediate refresh variation or an IORF instruction to perform I/O refreshing while ladder programming is being executed.

Immediate Refresh

The method of specifying immediate refreshing depends on whether the object to be refreshed is built-in I/O or an Expansion Unit.

- To specify immediate refreshing for the CPU Unit's built-in I/O, specify the immediate refresh variation (!) of the instruction.
- To specify immediate refreshing for Expansion I/O or an Expansion Unit, use the IORF instruction.

● Instructions with Refresh Variation (!)

Add an exclamation mark (!) in front of the instruction to specify immediate refreshing.

I/O will be refreshed as shown below when an instruction is executing if a real I/O bit in the CPU Unit's built-in I/O is specified as an operand.

- Bit Operands: I/O refreshing for the bit will be performed.
- Word Operands: I/O refreshing for the 16 specified bits will be performed.
- Input or Source Operands: Inputs are refreshed immediately before the instruction is executed.
- Output or Destination Operands: Outputs are refreshed immediately after the instruction is executed.

● IORF(097) Instruction

An I/O refresh (IORF) instruction is supported as a special instruction to refresh actual I/O data in the specified word range. By using this instruction, it is possible to refresh all data or data in a specified range of actual I/O in CP-series Expansion I/O and Expansion Unit during the cycle.

Precautions for Correct Use

It is not possible to use the immediate refresh variation (!) for the actual I/O of Expansion I/O or an Expansion Unit. Use the IORF instruction.

4-5 Constants

Overview

Constants are numeric values expressed in 16 or 32 bits and can be specified as instruction operands.

The following types of constants are supported.

- Bit Strings or Numeric Values (Integers)
  Decimal values (with & symbol), hexadecimal values (with # symbol), BCD values (with # symbol), or signed decimal values (with + or - symbol)
- Operands Specifying Numbers
  Decimal Notation (No Symbol)
- Floating Point (Real Number) Notation
  Signed decimal notation (with + or - symbol and decimal point)

Notation and Ranges

● Using Operands for Bit Strings or Numeric Values (Integers)

Unsigned Binary

Data type		Decimal values	Hexadecimal values
Notation		With & symbol & 10 10: Decimal value (integer) &: Decimal symbol	With # symbol # 000A 000A: Hexadecimal value using 0 to F #: Hexadecimal symbol
Application example		MOV &10 D0 Stores 10 decimal (#000A hex) in D0.	MOV #000A D0 Stores #000A hex (&10 decimal) in D0.
Precautions for correct use		• An error will occur and the left bus bar will be displayed in red if a hexadecimal value including A to F is input with & from the CX-Programmer. • The input will be treated as an address in the CIO Area and the contents of that address will be specified if a decimal value without & is input from the CX-Programmer.	• An error will occur and the left bus bar will be displayed in red if a hexadecimal value including A to F is input without # from the CX-Programmer. • The input will be treated as an address in the CIO Area and the contents of that address will be specified if a decimal value without # is input from the CX-Programmer.
Range	**16 bits**	&0 to 65535	#0000 to #FFFF
	32 bits	&0 to 4294967295	#00000000 to #FFFFFFFF

Signed Binary

Data type		Decimal values	Hexadecimal values
Notation		Signed + or - - 10 Decimal value (integer) + or - sign	With # symbol # FFF6 Hexadecimal value using 0 to F Hexadecimal symbol
Application example		MOV -10 D0 Stores 10 decimal (#FFF6 hex) in D0.	MOV # FFF6 D0 Stores #FFF6 hex (10 decimal) in D0.
Precautions for correct use		The input will be treated as an address in the CIO Area and the contents of that address will be specified if a decimal value without + or - is input from the CX-Programmer.	• An error will occur and the left bus bar will be displayed in red if a hexadecimal value including A to F is input without # from the CX-Programmer. • The input will be treated as an address in the CIO Area and the contents of that address will be specified if a decimal value without # is input from the CX-Programmer.
Range	**16 bits**	Negative: -32768 to -1	Negative: #8000 to #FFFF
		Positive: 0 to +32767	Positive: #0000 to #7FFF
	32 bits	Negative: -2147483648 to -1	Negative: #80000000 to #FFFFFFFF
		Positive: 0 to +2147483647	Positive: #00000000 to #7FFFFFFF

Unsigned BCD

Data type		Decimal values	BCD values
Notation		None	# 0010 Decimal value using 0 to 9 BCD symbol
Application example			+B #0010 D0 D1 Adds #0010 and the contents of D0 as BCD data and stores the result in D1.
Precautions for correct use			The input will be treated as an address in the CIO Area and the contents of that address will be specified if a decimal value without # is input from the CX-Programmer.
Range	**16 bits**	None	#0000 to #9999
	32 bits		#0000 0000 to #99999999

● Using Operands to Specify Numbers

Data type	Decimal values	Hexadecimal values or BCD values
Notation	No symbol (value only) 10 Number only	Not possible.
Application example	SBS 0 Jumps to subroutine 0.	
Precautions for correct use	An error will occur and the left bus bar will be displayed in red if a decimal value is input with & from the CX-Programmer.	

● Using Floating-point (Real Number) Notation for Operands

Data type	Decimal values	Hexadecimal values
Notation	With + or - + 0.10 Decimal value (real number) + or - sign	With # symbol (for single-precision data) # 3DCCCCCD Hexadecimal value using 0 to F Hexadecimal symbol
Application example	FIX +0.10 D0 Converts floating point +0.10 into 16-bit signed binary data and stores the integer portion in D0.	FIX #3DCCCCCD D0 Converts floating point #3DCCCCCD (+0.10 decimal) into 16-bit signed binary data and stores the integer portion in D0.
Precautions for correct use	The input will be treated as an address in the CIO Area, an error will occur, and the left bus bar will be displayed in red if a decimal value with a decimal point is input without + from the CX-Programmer.	The input will be treated as an address in the CIO Area, an error will occur, and the left bus bar will be displayed in red if a hexadecimal value including A to F is input without # from the CX-Programmer.

Additional Information

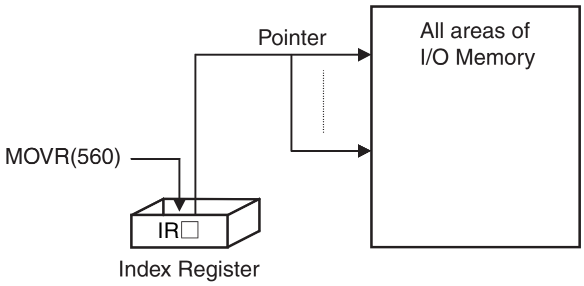
- Zero suppression can be used when inputting any data type.
  For example, “&2” and “&02”, “#000F” and “#F” are treated as the same.
- “BIN” indicates binary data.
- BCD data is binary coded decimal.

4-6 Index Registers

4-6-1 What are Index Registers?

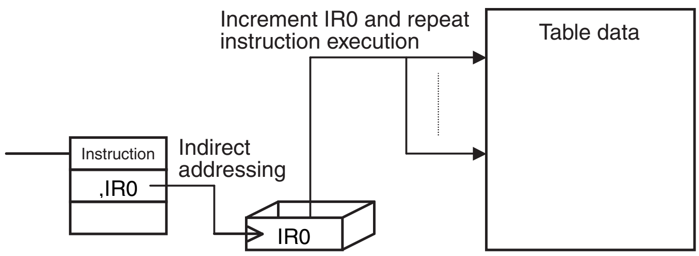
Index Registers function as pointers to specify PLC memory addresses, which are absolute memory addresses in I/O memory. After storing a PLC memory address in an Index Register with MOVR or MOVRW, input the Index Register as an operand in other instructions to indirectly address the stored PLC memory address.

The advantage of Index Registers is that they can specify any bit or word in I/O memory, including timer and counter PVs.



4-6-2 Using Index Registers

Index Registers can be a powerful tool when combined with loops such as FOR-NEXT loops. The contents of Index Registers can be incremented, decremented, and offset very easily, so a few instructions in a loop can process tables of consecutive data very efficiently.



Basically, Index Registers are used with the following steps:

1 Use MOVR to store the PLC memory address of the desired bit or word in an Index Register.

2 Specify the Index Register as the operand in almost any instruction to indirectly address the desired bit or word.

3 Offset or increment the original PLC memory address (see below) to redirect the pointer to another address.

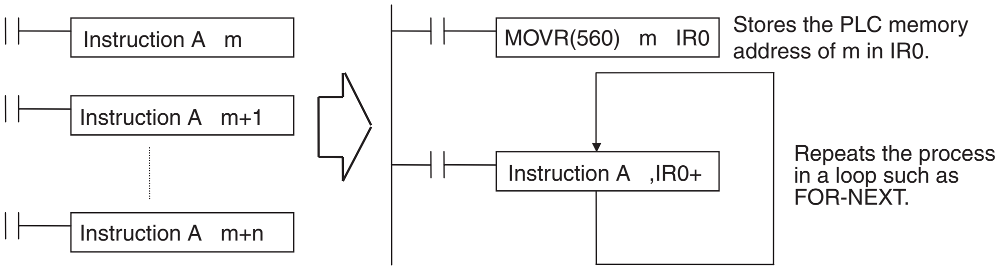
4 Continue steps 2 and 3 to execute the instruction on any number of addresses.

Note Use any of the following methods to offset or increment an Index Register.

- Indirect Addressing of Index Registers:
  Indirect addressing with auto-incrementing (,IR□+ or ,IR□++), indirect addressing with auto-decrementing (,–IR□ or ,––IR□), indirect referencing with a constant offset (constant,IR□), indirect addressing with a DR offset (DR□,IR□)

- Instructions for Direct Addressing of Index Registers:
  BINARY ADD (+L), BINARY SUBTRACT (–L), DOUBLE INCREMENT BINARY (++L), DOUBLE DECREMENT BINARY (––L)

Example:



The example given above shows how an Index Register in a program loop can replace a long series of instructions. In this case, instruction A is repeated n+1 times to perform some operation such as reading and comparing a table of values.

Precautions for Correct Use

The following instructions are executed even when the input conditions are OFF. Therefore, when indirect memory addresses are specified using auto-incrementing or auto-decrementing (,IR+ or ,IR-) in an operand of any of these instructions, the value in the Index Register (IR) is refreshed each cycle regardless of the input condition (increases or decreases one every cycle). This must be considered when writing a program.

Classification	Instructions
Sequence input instructions	LD, LD NOT, AND, AND NOT, OR, OR NOT, LD TST, LD TSTN, AND TST, AND TSTN, OR TST, OR TSTN
Sequence output instructions	OUT, OUT NOT, DIFU, DIFD
Sequence control instructions	JMP, FOR
Timer and counter instructions	TIM/TIMX(550), TIMH(015)/TIMHX(551), TMHH(540)/TMHHX(552), TTIM(087)/TTIMX(555), TIML(542)/TIMLX(553), MTIM(533)/MTIMX(554), CNT/CNTX(546), CNTR(012)/CNTRX(548)
Comparison instructions	Symbol comparison instructions (LD, AND, OR =, etc.)
Single-precision floating-point math instructions	Single-precision floating-point data comparison (LD, AND, OR = F, etc.)

The following ladder programming examples show how the index registers are treated.

Example 1

Ladder Program:
LD P_Off
OUT, IR0+

Operation: When the PLC memory address CIO 0.13 is stored in IR0.

The input condition is OFF (P_Off is the Always OFF Flag), so the OUT instruction sets CIO 0.13, which is indirectly addressed by IR0, to OFF. The OUT instruction is executed, so IR0 is incremented. As a result, the PLC memory address CIO 0.14, which was incremented by +1 in the IR0, is stored. Therefore, in the following cycle the OUT instruction turns OFF CIO 0.14.

Example 2

Ladder Program:
LD P_Off
SET, IR0+

Operation: When the PLC memory address CIO 0.13 is stored in IR0.

The input condition is OFF (P_Off is the Always OFF Flag), so the SET instruction is not executed. Therefore, IR0 is not incremented and the value stored in IR0 remains PLC memory address CIO 0.13.

- The following instructions are executed even when the interlock is active. Therefore, when indirect memory addresses are specified using auto-incrementing or auto-decrementing (,IR+ or ,IR-) in an operand of any of these instructions, the value in the Index Register (IR) is refreshed each cycle regardless of the input condition (increases or decreases one every cycle). This must be considered when writing a program.

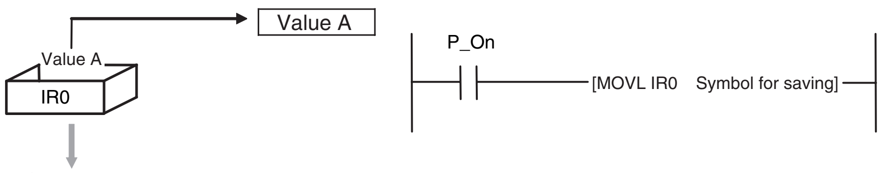
Instruction group	Instructions
Sequence output instructions	OUT, OUT NOT
Timer and counter instructions	TIM/TIMX, TIMH/TIMHX, TMHH/TMHHX, and TIML/TIMLX

Precautions for Correct Use

- When Index Registers IR0 to IR15 are used within function blocks, using the same Index Register within other function blocks or in the program outside of function blocks will create competition between the two instances and the program will not execute properly. Therefore, when using Index Registers (IR0 to IR15), always save the value of the Index Register at the point when the function block starts (or before the Index Register is used), and when the function block is completed (or after the Index Register has been used), incorporate processing in the program to return the Index Register to the saved value.

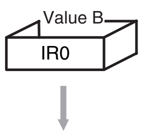
Example: Starting function block (or before using Index Register):

1. Save the value of IR (e.g., A).



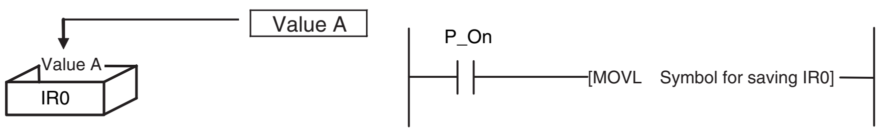
Within function block:

2. Use IR.



At start of function block (or before Index Register is used):

3. Return IR to saved value (e.g., A).



- Always set the value before using Index Registers. Operation will not be stable if Index Registers are used without the values being set.

Precautions for Correct Use

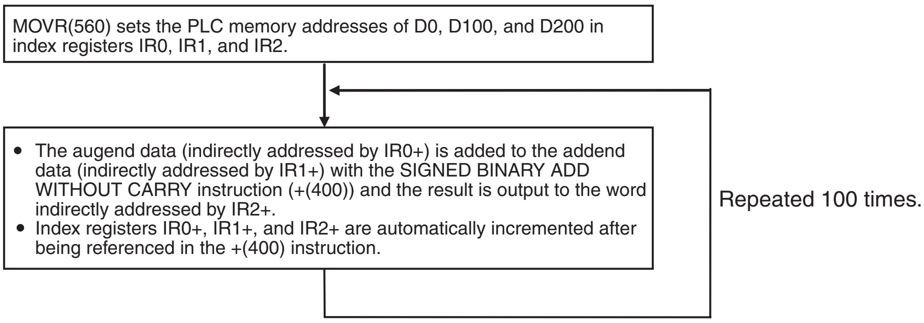
Precaution for Indirect Register Addressing with Index Registers

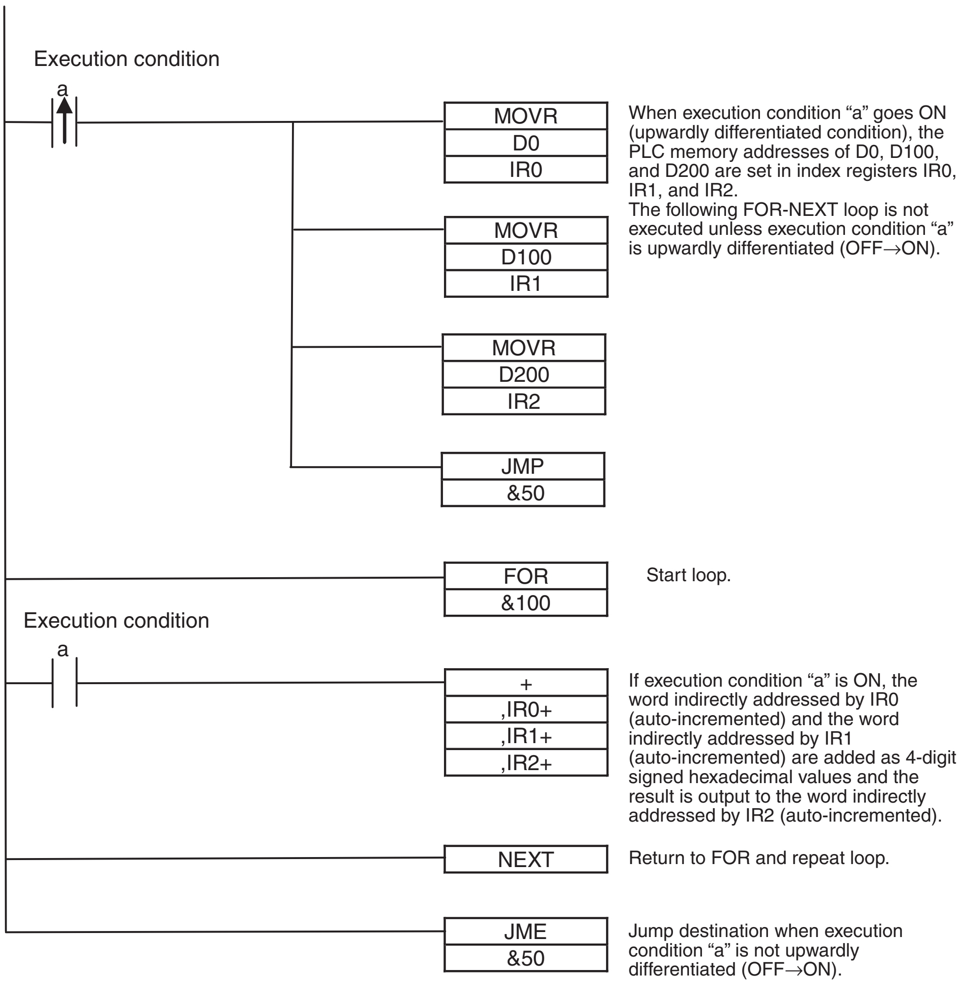
Do not use indirect register addressing to access areas outside of the I/O memory or areas reserved by the system. An Illegal Access Error will occur if access is attempted.

Refer to *A-4 Memory Map of PLC Memory Addresses* for details on the limits of PLC memory addresses.

Application Example for Index Registers

The data in D0 to D99 (augend data) is added to the data in D100 to D199 (addend data) and the addition results are output to D200 to D299. The operands of a single addition instruction are specified by index registers and the addition operations are performed by incrementing the index registers and repeatedly executing the addition instruction.





Additional Information

Index Registers can be directly addressed only in the instructions shown in the following table.

● Direct Addressing of Index Registers

The size of an index registers is two words per register for Index Registers IR0 to IR15, so use a double-word instruction (with an "L" in the mnemonic).

Instruction group	Instruction name	Mnemonic	Primary function
Data Movement Instructions	MOVE TO REGISTER	MOVR(560)	Stores the PLC memory address of a bit or word in an Index Register.
	MOVE TIMER/COUNTER PV TO REGISTER	MOVRW(561)	
	DOUBLE MOVE	MOVL(498)	Transfers between Index Registers. Used for exchanges and comparisons.
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)	
Increment/Decrement Instructions	DOUBLE INCREMENT BINARY	++L(591)	Changes the PLC memory address in the Index Register by incrementing, decrementing, or offsetting its content.
	DOUBLE DECREMENT BINARY	—L(593)	
Symbol Math Instructions	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L(401)	
	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	–L(411)	

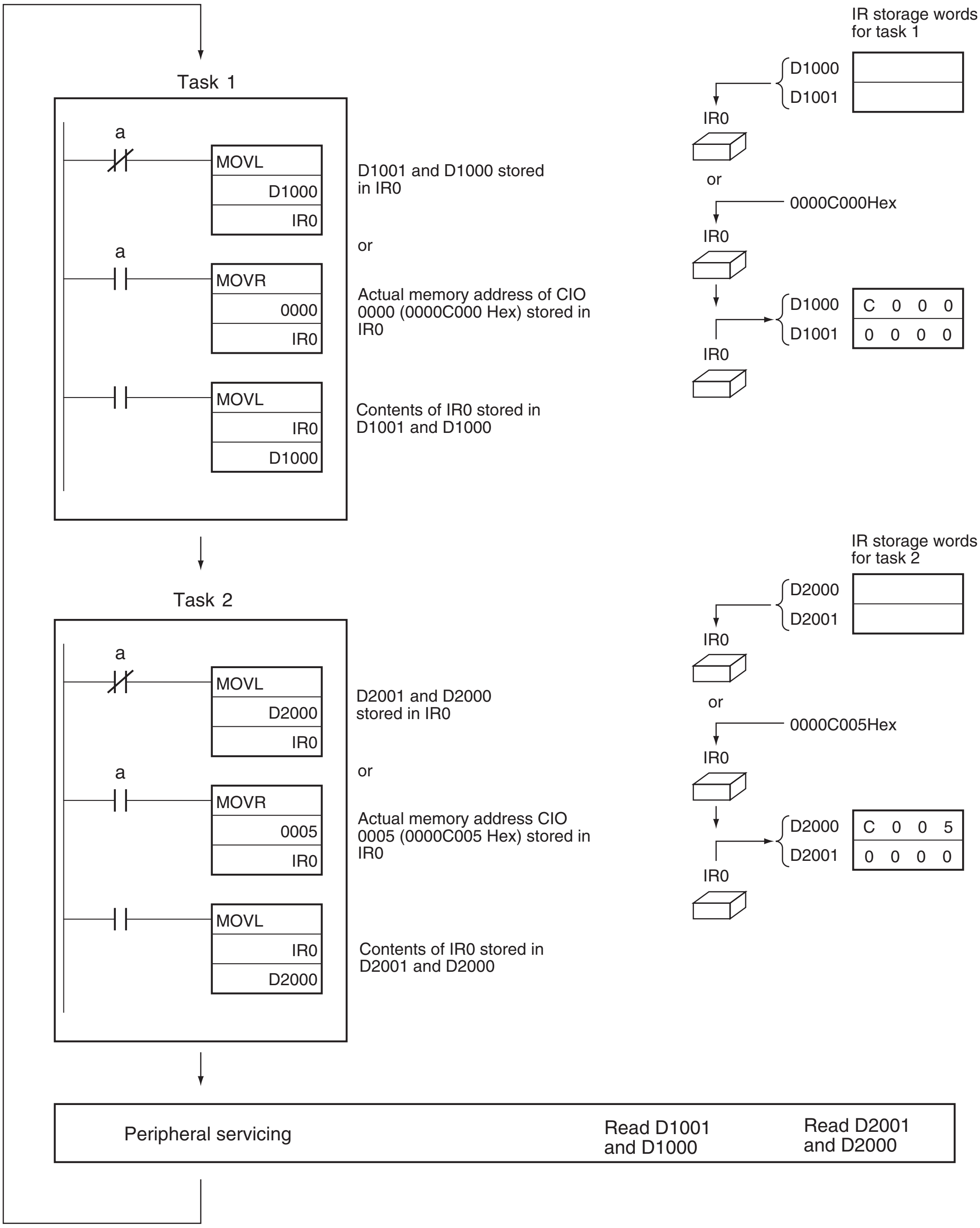
4-6-3 Monitoring Index Registers

It is possible to monitor Index Registers as follows:

- To use the CX-Programmer to monitor the final Index Register values for each task.
- 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.

● Example

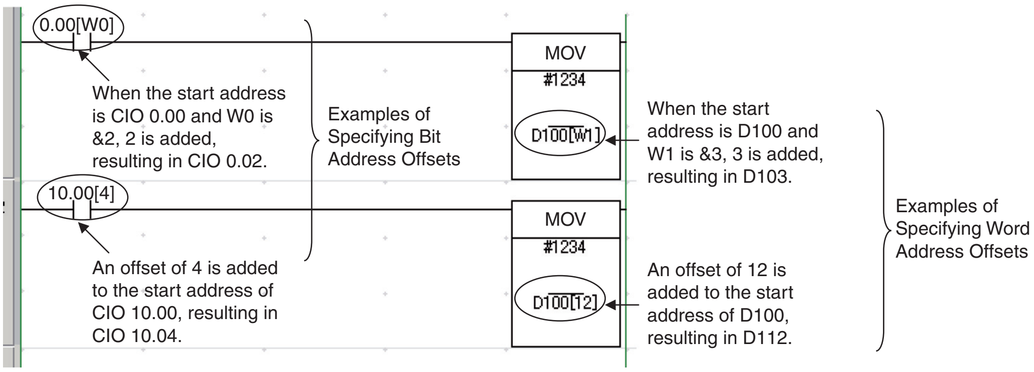
Note Be sure to use PLC memory addresses in Index Registers.



4-7 Specifying Offsets for Addresses

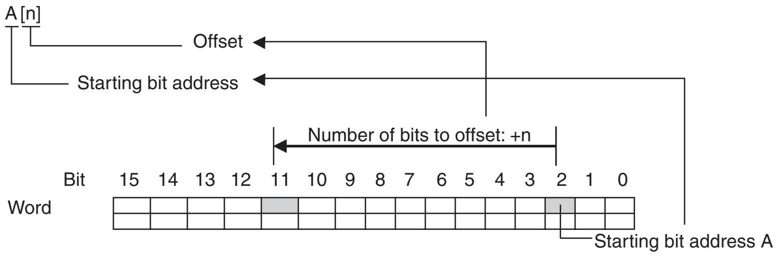
4-7-1 Overview

When an address is specified for an instruction operand, it is possible to change the specified address by specifying in brackets an offset for the specified address.



Bit Addresses

The bit address is offset by the amount specified by n (number of bits) from A (start bit address).



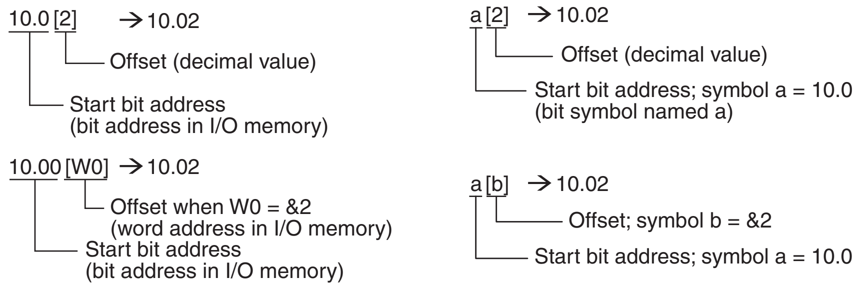
Start Bit Address

It is possible to specify the start bit address with a bit address or with a symbol (except the NUMBER data type cannot be used).
Offsetting is possible for all addresses except the DM Areas.
When specifying symbols, make the symbol table setting as the array variation. The number of arrays will be the maximum number of offset + 1 bit at least.
The I/O comment for the start bit address is displayed.

Offset

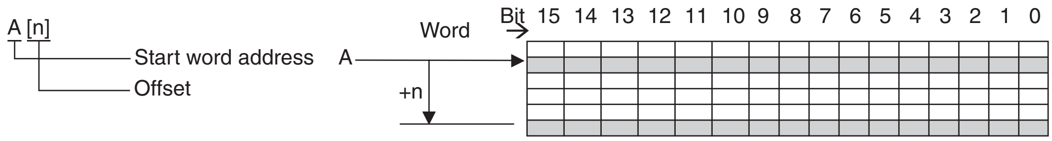
The offset can be specified as a decimal constant, word address (but CIO Area addresses cannot be specified), or a one-word symbol (i.e., symbols with the following data types: INT, UINT, WORD, CHANNEL).
Words in the Auxiliary Area (A) can only be specified as a decimal constant.
If a word address is specified, the contents of the specified word is used as the offset.
If the offset exceeds bit 15 in the specified word, offsetting will continue from bit 00 in the next word.
If the offset is specified indirectly, make sure that the final bit address does not exceed the upper limit of the memory area by using input comparison or other instruction.

Examples:



● Word Addresses

The word address is offset by the amount specified by n (number of offset words) from A (start word address).



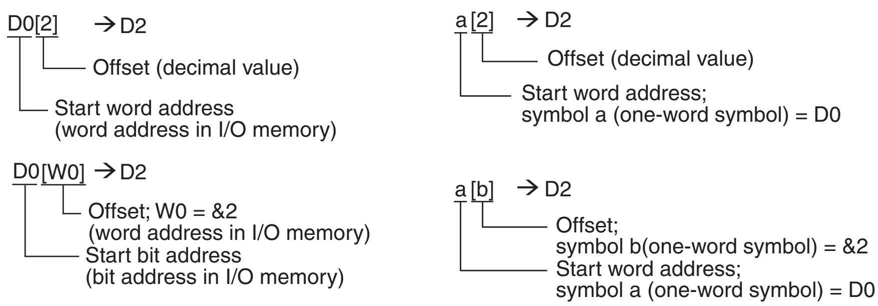
Start Word Address

It is possible to specify the start word address with a word address or with a symbol (except the NUMBER data type cannot be used).
Offsetting is possible only for addresses in the Holding, Word, and DM Areas.
The I/O comment for the start bit address is displayed.
When specifying symbols, make the symbol table setting as the array variation. The number of arrays will be the maximum number of offset + 1 word at least.

Offset

The offset can be specified as a decimal constant, word address (but CIO Area addresses cannot be specified), or one-word symbol (i.e., symbols with the following data types: INT, UINT, WORD, CHANNEL).
If a word address or symbol is specified, the contents of the specified word is used as the offset.
If the offset exceeds bit 15 in the specified word, offsetting will continue from bit 00 in the next word.
If the offset is specified indirectly, make sure that the final bit address does not exceed the upper limit of the memory area by using input comparison or other instruction.

Examples:



⚠ Caution

Program so that the memory area of the start address is not exceeded when using a word address or symbol for the offset.

For example, write the program so that processing is executed only when the indirect specification does not cause the final address to exceed the memory area by using an input comparison instruction or other instruction.

If an indirect specification causes the address to exceed the area of the start address, the system will access data in other area, and unexpected operation may occur.

4-7-2 Application Examples for Address Offsets

It is possible to dynamically specify the offset by specifying a word address in I/O memory for the offset in the brackets. The contents of the specified word address will be used as the offset.

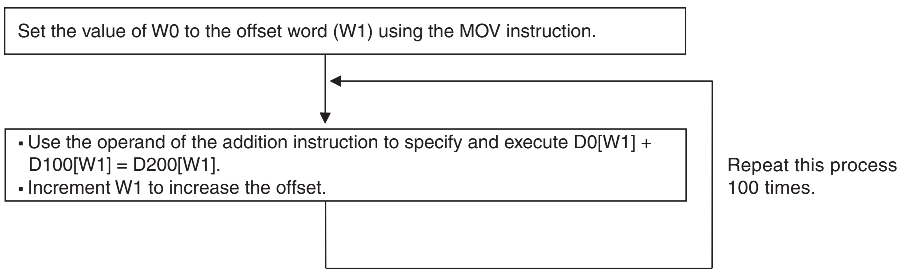
For example, execution can be performed by increasing the address by incrementing the value in the brackets and using only one instruction.

● Ladder Program Example

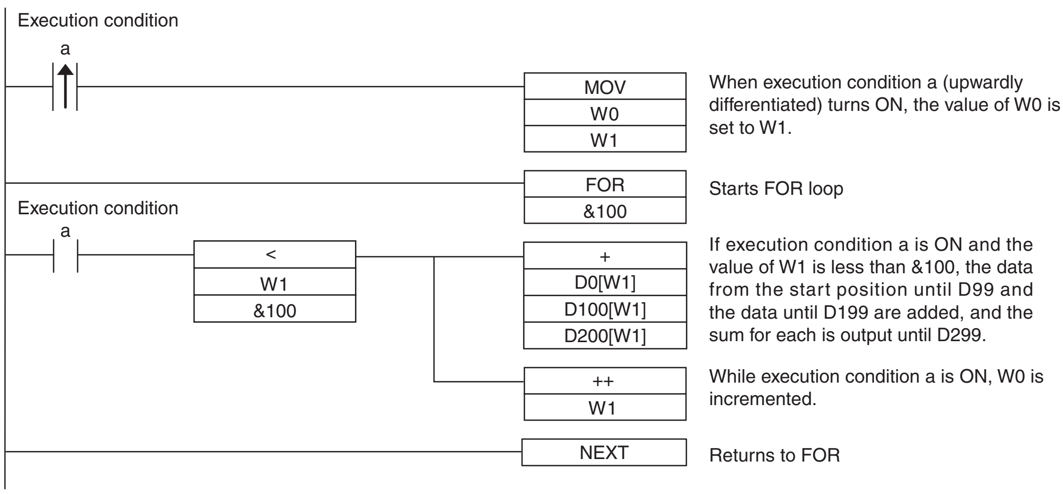
In this example, two areas of consecutive data are used: D0 to D99 and D100 to D199.

The contents of corresponding words are added starting from the specified starting point, W0, to the end of the areas and the sums are output to D200 to D299 starting from the specified offset from D200.

For example, if W0 is 30, the corresponding words from D30 to D99 and D130 to D199 are added, and the sums are output to D230 to D299.



Each process is performed with an input comparison instruction (<) as the execution condition so that W1 does not exceed &100 to make sure that the upper limit of the indirect addressing range is not exceeded.



4-8 Ladder Programming Precautions

4-8-1 Special Program Sections

For CP2E CPU Units, programs have special program sections that will control instruction conditions.

The following special program sections are available.

Program sections	Instructions	Instruction conditions	Status
Subroutine sections	SBS, SBN, and RET instructions	Subroutine program is executed.	The subroutine program section between SBN and RET instructions is executed.
IL-ILC sections	IL and ILC instructions	During IL	The output bits are turned OFF and timers are reset. Other instructions will not be executed and previous status will be maintained.
Step ladder sections	STEP instructions		
FOR-NEXT sections	FOR and NEXT instructions	Break in progress.	Looping

Instruction Combinations

The following table shows which of the special instructions can be used inside other program sections.

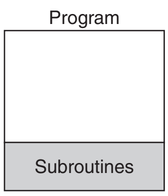
	Subroutine sections	IL-ILC sections	MILH and MILR-MILC sections	Step ladder sections	FOR-NEXT sections
Subroutine sections	No	No	No	No	No
IL-ILC sections	Yes	No	No	No	Yes
MILH and MILR-MILC sections	Yes	No	Yes	No	Yes
Step ladder sections	No	Yes	Yes	No	No
FOR-NEXT sections	Yes	Yes	Yes	No	Yes

Subroutines

Place all the subroutines together just after all of the main program and before the END instruction.

A subroutine cannot be placed in a step ladder, block program, or FOR-NEXT section.

If instructions other than those in a subroutine are placed after a subroutine (SBN to RET), those instructions will not be executed.



Instructions not Supported in Subroutines

The following instructions cannot be used in a subroutine.

Classification by function	Mnemonic	Instruction
Step Ladder Instructions	STEP	STEP DEFINE
	SNXT	STEP NEXT

Instructions not Supported in Step Ladder Program Sections

The following instructions cannot be used in step ladder program sections.

Classification by function	Mnemonic	Instruction
Sequence Control Instructions	FOR, NEXT, and BREAK	FOR, NEXT, and BREAK LOOP
	END	END
	IL and ILC	INTERLOCK and INTERLOCK CLEAR
	JMP and JME	JUMP and JUMP END
	CJP	CONDITIONAL JUMP and CONDITIONAL JUMP NOT
Subroutines	SBN and RET	SUBROUTINE ENTRY and SUBROUTINE RETURN

Note A step ladder program section can be used in an interlock section (between IL and ILC). The step ladder section will be completely reset when the interlock condition is ON.

5

I/O Memory

This section describes the types of I/O memory areas in a CP2E CPU Unit and the details.
Be sure you understand the information in the section before attempting to write ladder diagrams.
Refer to the *CP1E/CP2E CPU Unit Instructions Reference Manual* (Cat. No. W483) for detailed information on programming instructions.

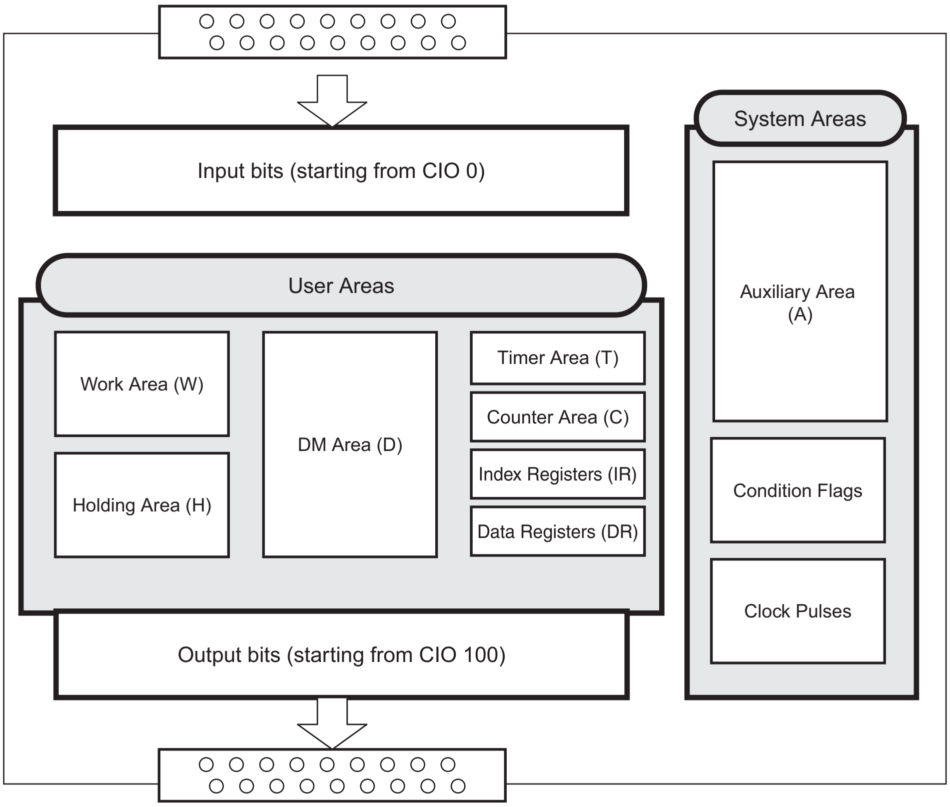
5

5-1 Overview of I/O Memory Areas

This section describes the I/O memory areas in a CP2E CPU Unit.

5-1-1 I/O Memory Areas

Data can be read and written to I/O memory from the ladder programs. I/O memory consists of an area for I/O with external devices, user areas, and system areas.



CIO Area (CIO 0 to CIO 289)

In the CIO Area, input bit addresses range from CIO 0 to CIO 99, output bit addresses range from CIO 100 to CIO 199 and addresses for serial PLC links range from CIO 200 to CIO 289.

The bits and words in the CIO Area are allocated to built-in I/O terminals on the CP2E CPU Unit and to the Expansion Units and Expansion I/O Units.

Input words and output bits that are not allocated may be used as work bits in programming.

Refer to *5-2 I/O Bits*

User Areas

These areas can be used freely by the user.

● Work Area (W)

The Word Area is part of the internal memory of the CPU Unit. It is used in programming. Unlike the input bits and output bits in the CIO Area, I/O to and from external devices is not refreshed for this area.

Use this area for work words and bits before using any words in the CIO Area. These words should be used first in programming because they will not be assigned to new functions in future versions of CP2E CPU Units.

Refer to *5-3 Work Area (W)*

● Holding Area (H)

The Holding Area is part of the internal memory of the CPU Unit. It is used in programming. Unlike the input bits and output bits in the CIO Area, I/O to and from external devices is not refreshed for this area.

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.

Automatically retained in the built-in non-volatile RAM even if the power supply is interrupted.

Refer to *5-4 Holding Area (H)*

● Data Memory Area (D)

This data area is used for general data storage and manipulation and is accessible only by word (16 bits).

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.

Specified words can be retained in the built-in Flash Memory using Auxiliary Area bits.

Automatically retained in the built-in non-volatile RAM even if the power supply is interrupted.

Refer to *5-5 Data Memory Area (D)*

● Timer Area (T)

There are two parts to the Timer Area: the Timer Completion Flags and the timer Present Values (PVs).

Up to 256 timers with timer numbers T0 to T255 can be used.

- Timer Completion Flags

Each Timer Completion Flag is accessed as one bit using the timer number.
A Completion Flag is turned ON when the set time of the timer elapses.

- Timer PVs

Each timer PV is accessed as one word (16 bits) using the timer number.
The PV increases or decreases as the timer operates.

Refer to *5-6 Timer Area (T)*

● Counter Area (C)

There are two parts to the Counter Area: the Counter Completion Flags and the Counter Present Values (PVs).

Up to 256 counters with counter numbers C0 to C255 can be used.

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.

Automatically retained in the built-in non-volatile RAM even if the power supply is interrupted.

- Counter Completion Flags

Each Counter Completion Flag is accessed as one bit using the counter number.
A Completion Flag is turned ON when the set value of the counter is reached.

- Counter PVs

Each counter PV is accessed as one word (16 bits) using the timer number.
The PVs count up or down as the counter operates.

Refer to *5-7 Counter Area (C)*

● Index Registers (IR)

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.

Refer to *5-8 Index Registers (IR)*

● Data Registers (DR)

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.

Refer to *5-9 Data Registers (DR)*

System Areas

System Areas contain bits and words with preassigned functions.

● Auxiliary Area (A)

The words and bits in this area have preassigned functions.

Refer to *A-2 Auxiliary Area Allocations by Address*

● Condition Flags

The Condition Flags include the flags that indicate the results of instruction execution, as well as the Always ON and Always OFF Flags.

The Condition Flags are specified with global symbols rather than with addresses. For example: P_on

● Clock Pulses

The Clock Pulses are turned ON and OFF by the CPU Unit's internal timer.

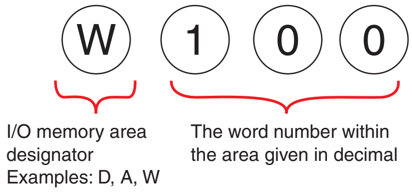
The Clock Pulses are specified with global symbols rather than with addresses. For example: P_0_02

5-1-2 I/O Memory Area Address Notation

An I/O memory can be addressed using word addresses or bit addresses. The word addresses and bit addresses are given in decimal format.

● Word Addresses

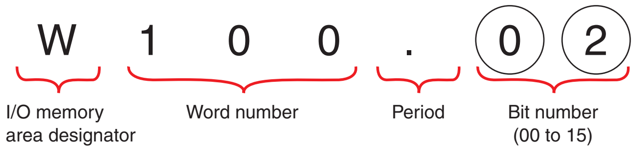
Specifies a16-bit word.



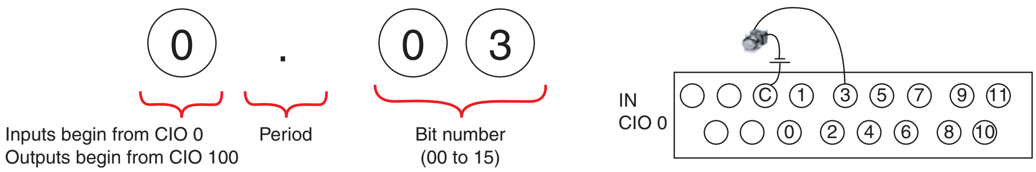
● Bit Addresses

A bit addresses specifies one of the 16 bits in a word.

The word number and bit number are separated with a period.



On the CX-Programmer, addresses in the CIO Area (including addresses for Serial PLC Links) are given with no I/O memory area designator. “CIO” is used as the I/O memory area designator in this manual for clarity.



5-1-3 I/O Memory Areas

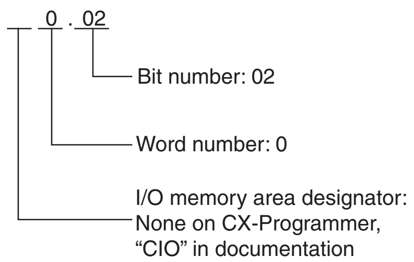
Name		No. of bits	Word addresses	Remarks	Reference
CIO Area	Input Bits	1,600 bits (100 words)	CIO 0 to CIO 99	–	Refer to *5-2 I/O Bits*.
	Output Bits	1,600 bits (100 words)	CIO 100 to CIO 199	–	
	Serial PLC Link Words	1,440 bits (90 words)	CIO 200 to CIO 289	–	Refer to *Section 14 Serial Communications.*
Work Area (W)		2,048 bits (128 words)	W0 to W127	–	Refer to *5-3 Work Area (W).*
Holding Area (H)		2,048 bits (128 words)	H0 to H127	Automatically retained even if the power supply is interrupted.	Refer to *5-4 Holding Area (H).*
Data Memory Area (D)	E☐☐-type CPU Unit	64K bits (4K words)	D0 to D4095	Automatically retained even if the power supply is interrupted. Data in specified words of the DM Area can be retained in the built-in Flash Memory by using a bit in the Auxiliary Area. Applicable words: D0 to D1499	Refer to *5-5 Data Memory Area (D).*
	S☐☐-type CPU Unit	128K bits (8K words)	D0 to D8191	Automatically retained even if the power supply is interrupted. Data in specified words of the DM Area can be retained in the built-in Flash Memory by using a bit in the Auxiliary Area. Applicable words: D0 to D6999	
	N☐☐-type CPU Unit	256K bits (16K words)	D0 to D16383	Automatically retained even if the power supply is interrupted. Data in specified words of the DM Area can be retained in the built-in Flash Memory by using a bit in the Auxiliary Area. Applicable words: D0 to D14999	
Timer Area (T)	Present values	256	T0 to T255	–	Refer to *5-6 Timer Area (T).*
	Timer Completion Flags	256			
Counter Area (C)	Present values	256	C0 to C255	Automatically retained even if the power supply is interrupted.	Refer to *5-7 Counter Area (C).*
	Counter Completion Flags	256			
Index Register (IR)	Index registers	16	IR0 to IR15	–	Refer to *5-8 Index Register (IR)*
Data Register (DR)	Data registers	16	DR0 to DR15	–	Refer to *5-9 Data Register (DR)*
Auxiliary Area (A)	Read only	7,168 bits (448 words)	A0 to A447	Specific addresses at startup are automatically retained even if the power supply is interrupted.	Refer to *A-2 Auxiliary Area Allocations by Address.*
	Read-write	8,192 bits (512 words)	A448 to A959		

5-2 I/O Bits

Overview

These words are allocated to built-in I/O terminals of CP2E CPU Units, CP-series Expansion Units and Expansion I/O Units.

Notation



Range

Input bits: CIO 0.00 to CIO 99.15 (100 words)

Output bits: CIO 100.00 to CIO 199.15 (100 words)

Applications

Built-in inputs can be used as basic inputs, interrupt inputs, quick-response inputs or high-speed counters.

Built-in outputs can only be used as basic outputs.

Refer to *Section 8 Overview of Built-in Functions and Allocations* for details.

Details

- Bits in the CIO Area can be force-set and force-reset.
- The contents of the CIO Area will be cleared in the following cases:
  - When the operating mode is changed between PROGRAM or MONITOR mode and RUN mode
  - When the PLC power is reset
  - When the CIO Area is cleared from the CX-Programmer
  - When PLC operation is stopped due to a fatal error other than an FALS error occurs. (The contents of the CIO Area will be retained when FALS is executed.)

Additional Information

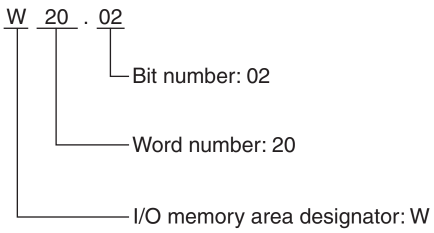
Words that are not allocated to the built-in I/O terminals of the CPU Units, the Expansion Units and Expansion I/O Units can only be used in programming. It is the same as the Work Area.

5-3 Work Area (W)

Overview

The Work Area is part of the internal memory of the CPU Unit. It is used in programming. Unlike the input bits and output bits in the CIO Area, I/O to and from external devices is not refreshed for this area.

Notation

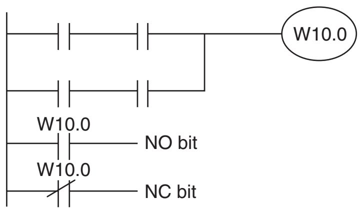


Range

The Work Area contains 128 words with addresses ranging from W0 to W127.

Applications

It is sometimes necessary to use the same set of input conditions many times in the same program. In this case a work bit can be used to store the final condition to simplify programming work and program design.



Storing a Condition in a Work Bit

Details

- 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:
  - When the operating mode is changed between PROGRAM or MONITOR mode and RUN mode
  - When the PLC power is reset
  - When the Work Area is cleared from the CX-Programmer
  - When PLC operation is stopped due to a fatal error other than an FALS error occurs. (The contents of the Work Area will be retained when FALS is executed.)

5-4 Holding Area (H)

Overview

The Holding Area is part of the internal memory of the CPU Unit. It is used in programming. Unlike the input bits and output bits in the CIO Area, I/O to and from external devices is not refreshed for this area.

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.

The Holding Area is automatically retained in the built-in non-volatile RAM even if the power supply is interrupted.

Notation

H 20 . 02

Bit number: 02

Word number: 20

I/O memory area designator: H

Range

The Holding area contains 128 words with addresses ranging from H0 to H127.

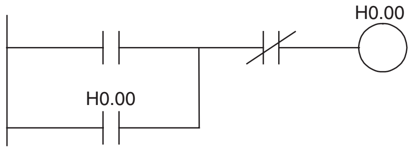
The Holding Area words from H512 to H1535 can be set for use with function blocks. These words cannot be specified as instruction operands in the user program.

Applications

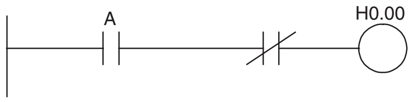
The Holding Area is used when you want to resume operation after a power interruption using the same status as before the power interruption.

Details

- Bits in the Holding Area can be force-set and force-reset.
- When a self-maintaining bit is programmed with a Holding Area bit, the self-maintaining bit will not be cleared even when the power is reset.
- 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.

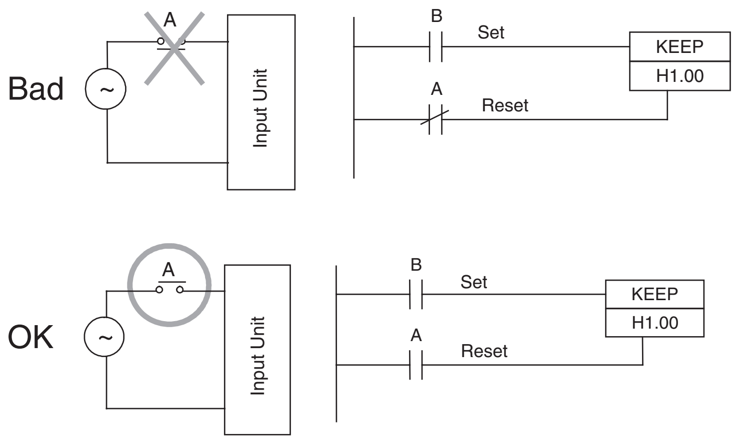


- If a Holding Area bit is used but not programmed as a self-maintaining bit, the bit will be turned OFF by execution condition A when the power is reset.



Precautions for Correct Use

- When a Holding Area bit is used in a KEEP instruction, never use a normally closed condition for the reset input.
  When the power supply goes OFF or is temporarily interrupted, the input will go OFF before the PLCs internal power supply and the Holding Area bit will be reset.



5-5 Data Memory Area (D)

Overview

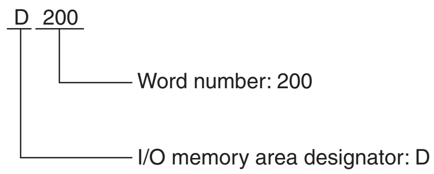
The DM area is used for general data storage and manipulation and is accessible only by word (16 bits).

These words retain their contents when the PLC is turned ON or the operating mode is switched between PROGRAM mode and RUN or MONITOR mode.

Some words in the DM Area can be saved to the built-in Flash Memory using Auxiliary Area bits. These words are specifically referred to as the backed up words in the DM Area.

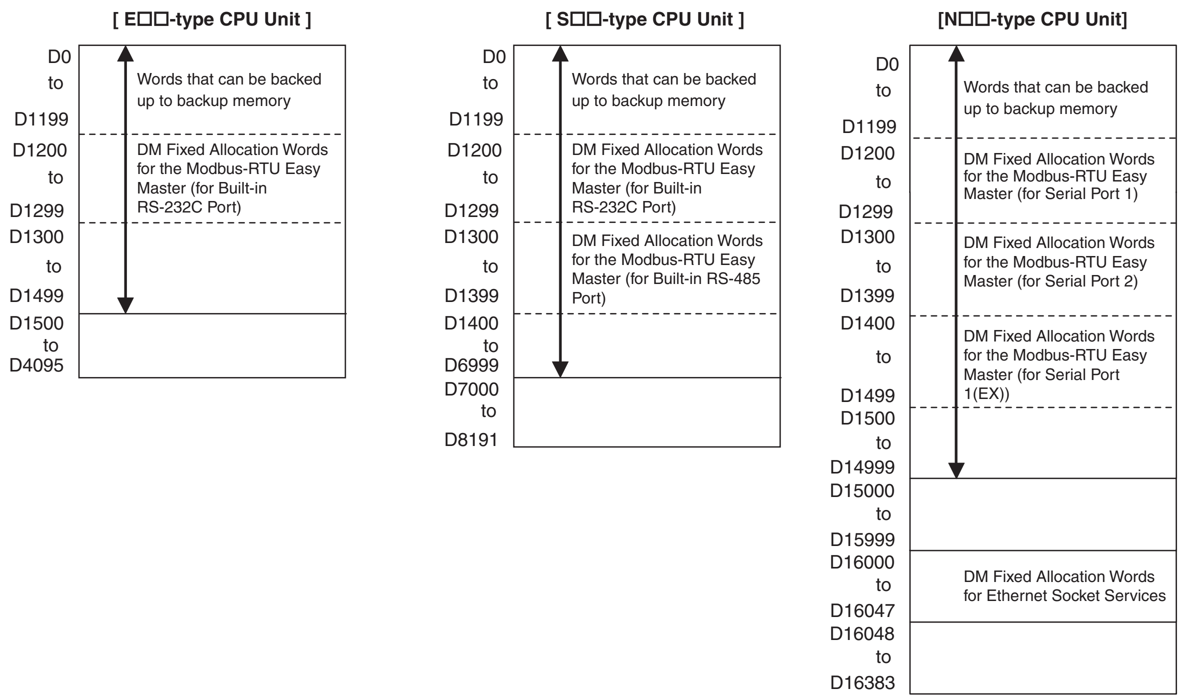
The DM Area is automatically retained in the built-in non-volatile RAM even if the power supply is interrupted.

Notation



Range

- E□□-type CPU Units have DM Area addresses ranging from D0 to D4095.
  Of these, D0 to D1499 can be backed up in backup memory (built-in Flash Memory).
- S□□-type CPU Units have DM Area addresses ranging from D0 to D8191.
  Of these, D0 to D6999 can be backed up in backup memory (built-in Flash Memory).
- N□□-type CPU Units have DM Area addresses ranging from D0 to D16383.
  Of these, D0 to D14999 can be backed up in backup memory (built-in Flash Memory).



Applications

The DM Area is for storing numeric data. It can be used for data exchange with Programmable Terminals, serial communications devices, such as Inverters, and Analog I/O Units or Temperature I/O Units.

Details

Bits in the DM Area cannot be addressed individually.

● Backing Up to the Built-in Flash Memory

- The number of words set in the PLC Setup can be saved to the built-in Flash Memory during operation by turning ON the DM Backup Start bit (A751.15).
- Specify in the PLC Setup whether to read the data in the DM Area words to the RAM as the initial values when the power supply is turned ON.

Refer to *16-3 DM Backup Function* for how to use DM Area words and bits.

● DM Fixed Allocation Words for the Modbus-RTU Easy Master

D1200 to D1499 are used as command and response storage areas with the Modbus-RTU Easy Master function. These words are used for other applications if the Modbus-RTU Easy Master function is not served.

Refer to *14-4 Modbus-RTU Easy Master Function* for how to use the DM Area words and bits.

● DM Fixed Allocation Words for Ethernet Socket Services

D16000 to D16047 are used as setting, status and parameter areas with the Socket Service function. These words are used for other applications if the Ethernet Socket Service function is not served.

Refer to *15-6-6 Data Memory Area Allocations* for how to use the DM Area words and bits.

● Indirect Addressing of the DM Area

Indirect addressing can be used in the DM Area.

There are two modes that can be used.

Binary-mode Addressing (@D)

If a "@" symbol is input before a DM Area address, the contents of that DM Area word is treated as a hexadecimal (binary) address and the instruction will operate on the DM Area word at that address.

The entire DM Area can be indirectly addressed with hexadecimal values 0000 to 3FFF.

Example: @D0 [0100] → D256 [ ]

Address actually used.

BCD-mode Addressing (*D)

If a * symbol is input before a DM Area address, the content of that DM Area word is treated as a BCD address and the instruction will operate on the DM Area word at that address.

Only part of the DM Area (D0 to D9999) can be indirectly addressed with BCD values 0 to 9999.

Example: *D0 [0100] → D100 [ ]

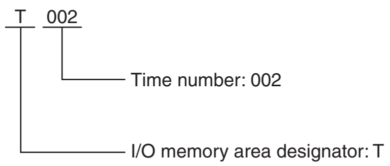
Address actually used.

5-6 Timer Area (T)

Overview

The Timer Area contains Timer Completion Flags (1 bit each) and timer PVs (16 bits each). The Completion Flag is turned ON when a decrementing timer PV reaches 0 (counting out) or an incrementing/decrementing timer PV reaches the set value or 0.

Notation



Range

Timer numbers range from T0 to T255.

Details

● Types of Timers

The following table shows which instructions are used to refresh timer PVs in BCD and binary mode.

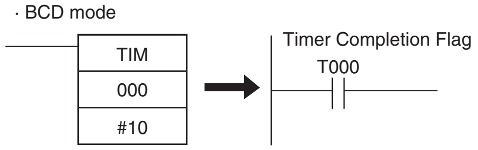
Timer instruction	BCD mode	Binary mode
HUNDRED-MS TIMER	TIM	TIMX
TEN-MS TIMER	TIMH	TIMHX
ONE-MS TIMER	TMHH	TMHHX
ACCUMULATIVE TIMER	TTIM	TTIMX

Timer numbers 0 to 255 are used by all timers listed above.

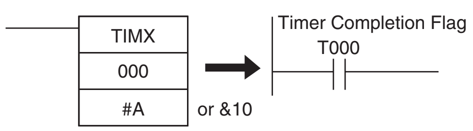
Note Only timer numbers 0 to 15 can be used in 1ms TIMER (TMHH/TMHHX).

● Timer Example: Timer Number 0 and a Timer Set Value of 1 s

· BCD mode



· Binary mode



● Timer PV Refresh Method

Timer numbers	Timer PV refresh method
T0 to T255	The timer PV is refreshed when the instruction is executed. This can cause a delay depending on the cycle time. • When the cycle time is longer than 100 ms, delay is generated by the TIM/TIMX instruction. • When the cycle time is longer than 10 ms, delay is generated by the TIMH/TIMHX instruction. • When the cycle time is longer than 1 ms, delay is generated by the TMHH/TMHHX instruction.

Precautions for Correct Use

It is not recommended to use the same timers number in two timer instructions because the timers will not operate correctly if they are operating simultaneously.

Do not use the same timer number for more than one instruction.

If two or more timer instructions use the same timer number, an error will be generated during the program check.

● Resetting or Maintaining Timers

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

Instruction	TIM/TIMX	TIMH/TIMHX	TMHH/TMHHX	TTIM/TTIMX
	HUNDRED-MS TIMER	TEN-MS TIMER	ONE-MS TIMER	ACCUMULATIVE TIMER
When the operating mode is changed between PROGRAM or MONITOR mode and RUN mode*1	PV=0 Flag=OFF			
When the PLC power is reset	PV=0 Flag=OFF			
CNR/CNRX instructions (timer/counter reset)*2	PV= 9999/FFFF Flag=OFF			
Jumps (JMP-JME)	Retained			
Interlocks (IL-ILC) with OFF interlock conditions	Reset (PV = SV, Timer Completion Flag = OFF)			Retained

*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 Since the TIML/TIMLX instructions do not use timer numbers, they are reset under different conditions.
The PV for a TIML/TIMLX instruction is reset to the SV.
Refer to the descriptions of these instructions for details.

5-7 Counter Area (C)

Overview

The Counter Area contains Completion Flags (1 bit each) and counter PVs (16 bits each). A Completion Flag is turned ON when the counter PV reaches the set value (counting out).

Completion Flags and counter PVs are automatically retained in the built-in non-volatile RAM even if the power supply is interrupted.

Notation

C 002

Counter number: 002

I/O memory area designator: C

Range

Counter numbers range from C0 to C255.

Details

● Types of Counters

The following table shows which instructions are used to refresh counter PVs in BCD and binary mode.

Counter instruction	BCD mode	Binary mode
COUNTER	CNT	CNTX
REVERSIBLE COUNTER	CNTR	CNTRX

Counter numbers 0 to 255 are used by all counters given above.

Built-in high-speed counters 0 to 5 do not use counter numbers.

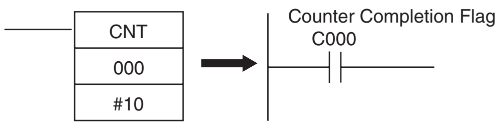
Precautions for Correct Use

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.

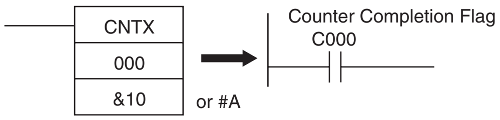
If two or more counter instructions use the same counter number, an error will be generated during the program check.

● Counter Example: Counter Number 0 with a Counter Set Value of 10

· BCD mode



· Binary mode



● Resetting or Maintaining Counter PVs

- 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 Counter 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.
- The following table shows when counters PVs are reset or maintained.

Instruction	CNT/CNTX	CNTR/CNTRX
	COUNTER	REVERSIBLE COUNTER
PV and Counter Completion Flag when counter is reset	PV=0 Counter Completion Flag = OFF	
When the operating mode is changed between PROGRAM or MONITOR mode and RUN mode	Retained	
When the PLC power is reset	Retained	
Reset Input	Reset	
CNR/CNRX instructions	Reset	
Interlocks (IL-ILC) with OFF interlock conditions	Retained	

5-8 Index Registers (IR)

Overview

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. These are different from the I/O memory area addresses in the CIO Area, DM Area, etc. They are the continuous RAM addresses.

The user cannot directly input PLC memory addresses in the Index Registers. Use MOVR to convert a regular data area address to its equivalent PLC memory address and write that value to the specified Index Register. (Use MOVRW to set the PLC memory address of a timer/counter PV in an Index Register.)

Notation

IR 13

Index register number: 13

Index register designator: IR

Range

Index register numbers range from IR0 to IR15.

Details

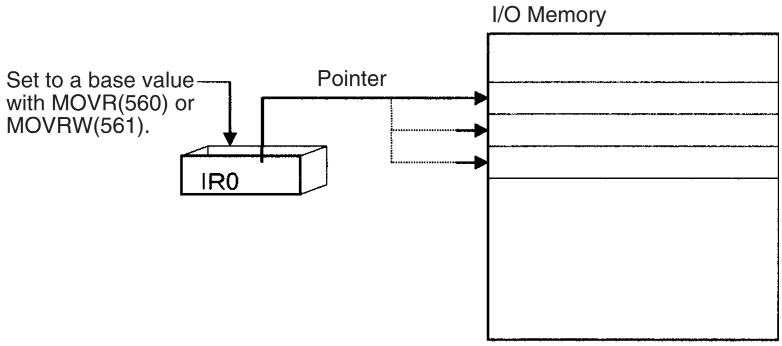
● Indirect Addressing

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 or MOVRW and then modified as pointers in each instruction.

For pointer operation, either special instructions that can directly specify Index Registers (MOVR, MOVRW, increment instructions, decrement instructions, or math instructions) are used, or indirect offsets, auto-incrementing, or auto-decremented is used.



Index Register Initialization

The Index Registers will be cleared in the following cases:

- The operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa and the IOM Hold Bit is OFF.
- The PLC's power supply is cycled and the IOM Hold Bit is OFF or not set to be held in the PLC Setup.
- A fatal error occurs (except for one created with FALS).

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

Precautions for Correct Use

Precaution for Indirect Register Addressing with Index Registers

Do not use indirect register addressing to access areas outside of the I/O memory or areas reserved by the system.

An Illegal Access Error will occur if access is attempted. Refer to *A-5 Memory Map* for details on the limits of PLC memory addresses.

Indirect Addressing Variations

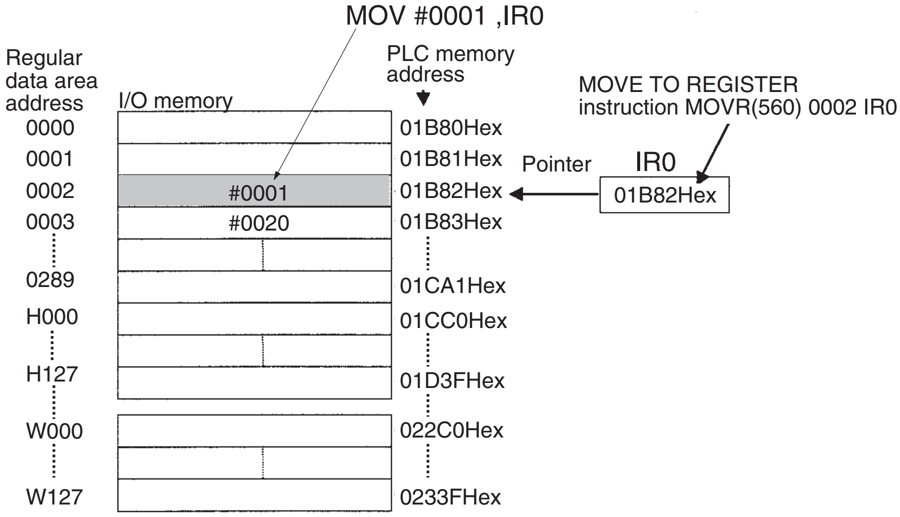
The following table shows the variations available when indirectly addressing I/O memory with Index Registers. (IR□ represents an Index Register from IR0 to IR15.)

Variation	Function	Syntax	Example	
Indirect addressing	The content of IR□ is treated as the PLC memory address of a bit or word.	,IR□	LD ,IR0	Loads the bit at the PLC memory address contained in IR0.
Indirect addressing with constant offset	The constant prefix is added to the content of IR□ and the result is treated as the PLC memory address of a bit or word. The constant may be any integer from –2,048 to 2,047.	Constant ,IR□ (Include a + or – in the constant.)	LD +5,IR0	Adds 5 to the contents of IR0 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□ and the result is treated as the PLC memory address of a bit or word.	DR□,IR□	LD DR0,IR0	Adds the contents of DR0 to the contents of IR0 and loads the bit at that PLC memory address.
Indirect addressing with auto-increment	After referencing the content of IR□ as the PLC memory address of a bit or word, the content is incremented by 1 or 2.	Increment by 1: ,IR□+ Increment by 2: ,IR□++	LD , IR0++	Loads the bit at the PLC memory address contained in IR0 and then increments the content of IR0 by 2.
Indirect addressing with auto-decrement	The content of IR□ 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□ Decrement by 2: ,– –IR□	LD , – –IR0	Decrements the content of IR0 by 2 and then loads the bit at that PLC memory address.

Example

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	2	IR0	Stores the PLC memory address of CIO 2 in IR0.
MOV	#0001	,IR0	Writes #0001 to the PLC memory address contained in IR0.
MOV	#0020	+1,IR0	Reads the content of IR0, adds 1, and writes #0020 to that PLC memory 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.

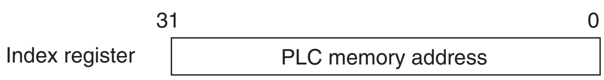
- Word Operand:
  MOVR 0000 IR2
  MOV D0 ,IR2

  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 MOVR sets the PLC memory address of CIO 2 in IR2 and the MOV instruction copies the contents of D0 to CIO 2.
- Bit Operand:
  MOVR 000013 ,IR2
  SET +5 , IR2

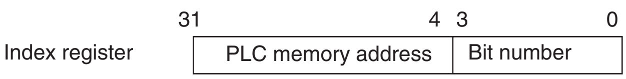
  When the operand is treated as a bit, the leftmost 7 digits of the Index Register specify the word address and the rightmost digit specifies the bit number. In this example, MOVR sets the PLC memory address of CIO 13 (0C00D hex) in IR2. The SET instruction adds +5 from bit 13 (D hex) to this PLC memory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

Additional Information

When MOVR is used to set a word address in an index register, the address is stored as follows:



When MOVR is used to set a bit address in an index register, the address is stored as follows:



Precautions for Correct Use

- Always set the value of an index register (IR) before using the index register. Operation will not be dependable if an index register is used without first setting its value.
- The value of an index register will not be dependable after an interrupt task is started. When using an index register in an interrupt task, set the value using MOVR or MOVRW before using the index register. (MOVRW is used for timer/counter PVs and MOVR is used for other values.)

Additional Information

- 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.

● Direct Addressing

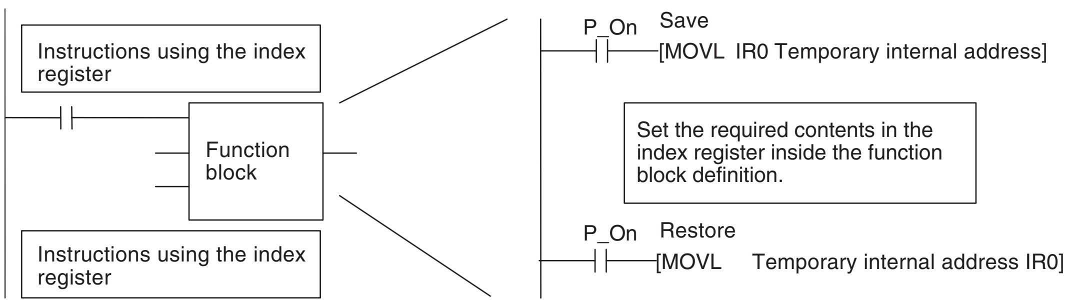
Instruction group	Instruction name	Mnemonic
Data Movement Instructions	MOVE TO REGISTER	MOVR(560)
	MOVE TIMER/COUNTER PV TO REGISTER	MOVRW(561)
	DOUBLE MOVE	MOVL(498)
Increment/Decrement Instructions	DOUBLE INCREMENT BINARY	++L(591)
	DOUBLE DECREMENT BINARY	– –L(593)
Comparison Instructions	DOUBLE EQUAL	=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)

● Precautions

It is possible to read the Index Register for only 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 CX-Programmer 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.

Additional Information

The contents of an index register used inside a function block may be corrupted when the function block is called. Always save the contents of the index register before calling the function block and then restore the contents after leaving the function block. Set the required contents in the index register inside the function block.



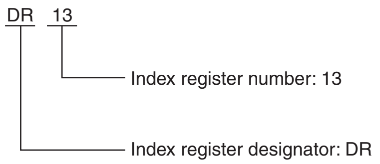
5-9 Data Registers (DR)

Overview

The sixteen Data Registers (DR0 to DR15) are used to offset the PLC memory addresses in Index Registers when addressing words indirectly. The Data Registers can be used to specify an offset to add to an Index Register when addressing words indirectly.

The content of Data Registers cannot be accessed (read or written) from the CX-Programmer.

Notation



Range

Index register numbers range from DR0 to DR15.

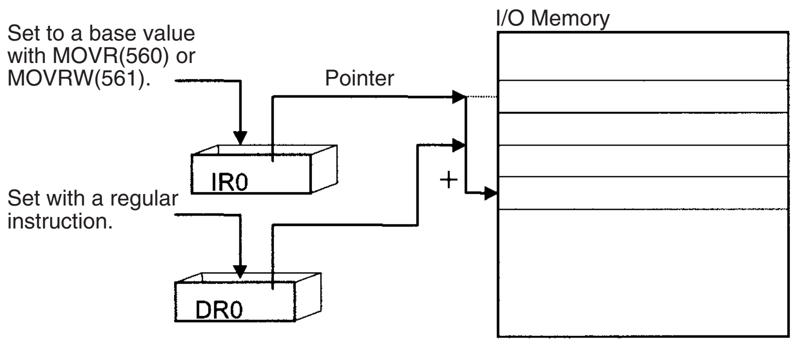
Details

● Forcing Bit Status

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.

Normal instructions can be use to store data in Data Registers.

Data Registers cannot be force-set or force-reset.



Example

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

LD	DR0 ,IR0		Adds the contents of DR0 to the contents of IR0 and loads the bit at that PLC memory address.
MOV	#0001	DR0 ,IR1	Adds the contents of DR0 to the contents of IR1 and writes #0001 to that PLC memory address.

Range of Values

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 FFFF	–32,768 to –1
0000 to 7FFF	0 to 32,767

The content of Data Registers cannot be accessed (read or written) from the CX-Programmer.

Data Register Initialization

The Data Registers will be cleared in the following cases:

- The operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa and the IOM Hold Bit is OFF.
- The PLC's power supply is cycled and the IOM Hold Bit is OFF or not set to be held in the PLC Setup.
- A fatal error occurs (except for one created with FALS).

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

Precautions for Correct Use

- 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.

5-10 Auxiliary Area (A)

Overview

The words and bits in this area have preassigned functions.

Specific addresses (error log, clock area) at startup are automatically retained in the built-in non-volatile RAM even if the power supply is interrupted.

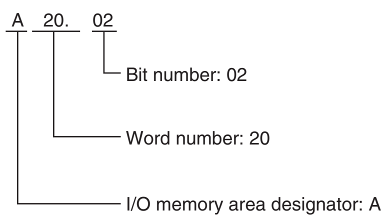
Refer to *A-2 Auxiliary Area Allocations by Address* for details.

Precautions for Correct Use

For an S/N☐☐-type CPU Unit without a Battery, the clock may be stop when the power supply is turned ON. When the clock stops, the data in the Auxiliary Area (A) related to clock will be reset to 01-01-01 01:01:01 Sunday. For an E☐☐-type CPU Unit, the data is always 01-01-01 01:01:01 Sunday.

Word/Bit	Name	Data retained when the clock of an S/N☐☐-type stops or in an E☐☐-type
A100~A199 CH	Error Log Area	01-01-01 01:01:01 Sunday
A351~A354 CH	Calendar/Clock Area	
A510~A511 CH	Startup Time	
A512~A513 CH	Power Interruption Time	
A515~A517 CH	Operation Start Time	
A518~A520 CH	Operation End Time	
A720~A749 CH	Power ON Clock Data 1 to 10	

Notation



Range

The Auxiliary Area contains 960 words with addresses ranging from A0 to A959.

Applications

Applications of the bits and words in the Auxiliary Area are predefined. Ladder programs can be simplified and controllability can be improved by effectively using the bits and words in this area.

Details

- Some words or bits are set automatically by the system and others are set and manipulated by the user.
  The Auxiliary Area includes error flags set by self-diagnosis, initial settings, control bits, and status data.
- Words and bits in this area can be read and written from the program or the CX-Programmer.
- The Auxiliary Area contains words that are read-only (A0 to A447) and words that can be read and written (A448 to A959).
- Even the read/write bits in the Auxiliary Area cannot be force-set and force-reset continuously.

● Auxiliary Area Words and Bits in the CX-Programmer's System-defined Symbols

The following table gives the Auxiliary Area bits and words pre-registered in the CX-Programmer's global symbol table as system-defined symbols.

Refer to *A-2 Auxiliary Area Allocations by Address* for details.

Word/Bit	Name	Name in CX-Programmer
A200.11	First Cycle Flag	P_First_Cycle
A200.12	Step Flag	P_Step
A200.15	First Cycle Task Flag	P_First_Cycle_Task
A262	Maximum Cycle Time	P_Max_Cycle_Time
A264	Present Cycle Time	P_Cycle_Time_Value
A401.08	Cycle Time Too Long Flag	P_Cycle_Time_Error
A402.04	Battery Error Flag	P_Low_Battery
A500.15	Output OFF Bit	P_Output_Off_Bit

5-11 Condition Flags

Overview

These flags include the flags that indicate the results of instruction execution, as well as the Always ON and Always OFF Flags. These bits are specified with symbols rather than addresses.

The CX-Programmer treats condition flags as system-defined symbols (global symbols) beginning with P_.

Notation

P_ ER

Condition flag name: ER

I/O memory area designator:
P_ (indicates a system symbol name)

Details

The Condition Flags are read-only; they cannot be written from instructions or from the CX-Programmer.

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

● Types of Condition Flags

Refer to *4-6 Ladder Programming Precautions* for details.

Name	Name in CX-Programmer	Function
Always ON Flag	P_On	Always ON.
Always OFF Flag	P_Off	Always OFF.
Error Flag	P_ER	Turned ON when the operand data in an instruction is incorrect (an instruction processing error) to indicate that an instruction ended because of an error. When the PLC Setup is set to stop operation for an instruction error (Instruction Error Operation), program execution will be stopped and the Instruction Processing Error Flag (A295.08) will be turned ON when the Error Flag is turned ON.
Access Error Flag	P_AER	Turned ON when an Illegal Access Error occurs. The Illegal Access Error indicates that an instruction attempted to access an area of memory that should not be accessed. When the PLC Setup is set to stop operation for an instruction error (Instruction Error Operation), program execution will be stopped and the Instruction Processing Error Flag (A4295.10) will be turned ON when the Access Error Flag is turned ON.
Carry Flag	P_CY	Turned ON when there is a carry in the result of an arithmetic operation or a 1 is shifted to the Carry Flag by a Data Shift instruction. The Carry Flag is part of the result of some Data Shift and Symbol Math instructions.
Greater Than Flag	P_GT	Turned ON when the first operand of a Comparison Instruction is 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 equal or the result of a calculation is 0.

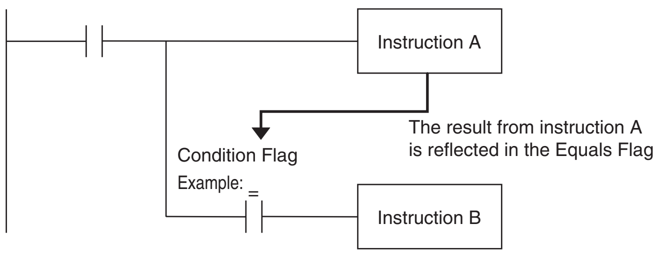
Name	Name in CX-Programmer	Function
Less Than Flag	P_LT	Turned ON when the first operand of a Comparison Instruction is less than the second or a value is below a specified range.
Negative Flag	P_N	Turned ON when the most significant bit of a result is ON.
Overflow Flag	P_OF	Turned ON when the result of calculation overflows the capacity of the result word(s).
Underflow Flag	P_UF	Turned ON when the result of calculation underflows the capacity of the result word(s).
Greater Than or Equals Flag	P_GE	Turned ON when the first operand of a Comparison Instruction is greater than or equal to the second.
Not Equal Flag	P_NE	Turned ON when the two operands of a Comparison Instruction are not equal.
Less than or Equals Flag	P_LE	Turned ON when the first operand of a Comparison Instruction is less than or equal to the second.

● Using the Condition Flags

The Condition Flags are shared by all of the instructions. Their status may change after each instruction execution in a single cycle.

Therefore, be sure to use Condition Flags on a branched output with the same execution condition immediately after an instruction to reflect the results of instruction execution.

Example: Using Instruction A Execution Results



Instruction	Operand
LD	
Instruction A	
AND	=
Instruction B	

Precautions for Correct Use

The Condition Flags are shared by all of the instructions. This means that 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 ladder programs to prevent unexpected operation.

5-12 Clock Pulses

Overview

The Clock Pulses are turned ON and OFF by the CPU Unit's internal timer. These bits are specified with symbols rather than addresses.

The CX-Programmer treats condition flags as system-defined symbols (global symbols) beginning with P_.

Notation

P_ 0_02s

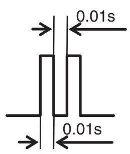
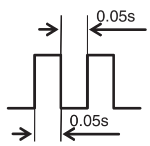
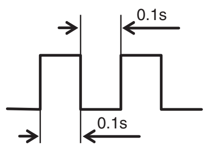
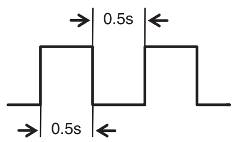
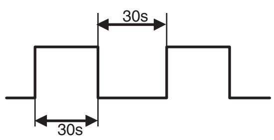
Clock pulse name: 0_02s

I/O memory area designator:
P_ (indicates a system symbol name)

Details

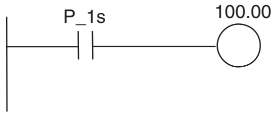
The Clock Pulses are read-only; they cannot be written from instructions or from the CX-Programmer.

● Clock Pulses

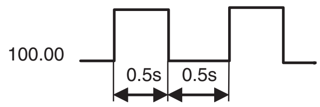
Name	Name in CX-Programmer	Description	
0.02-s Clock Pulse	P_0_02s		ON for 0.01 s OFF for 0.01 s
0.1-s clock pulse	P_0_1s		ON for 0.05 s OFF for 0.05 s
0.2-s clock pulse	P_0_2s		ON for 0.1 s OFF for 0.1 s
1-s clock pulse	P_1s		ON for 0.5 s OFF for 0.5 s
1-min clock pulse	P_1min		ON for 30 s OFF for 30 s

● Using the Clock Pulses

The following example turns a bit ON and OFF at 0.5-s intervals.



Instruction	Operand
LD	P_1s
OUT	100.00



6

I/O Allocation

This section describes I/O allocation used to exchange data between the CP2E CPU Unit and other units.
Be sure you understand the information in the section before attempting to write ladder diagrams.

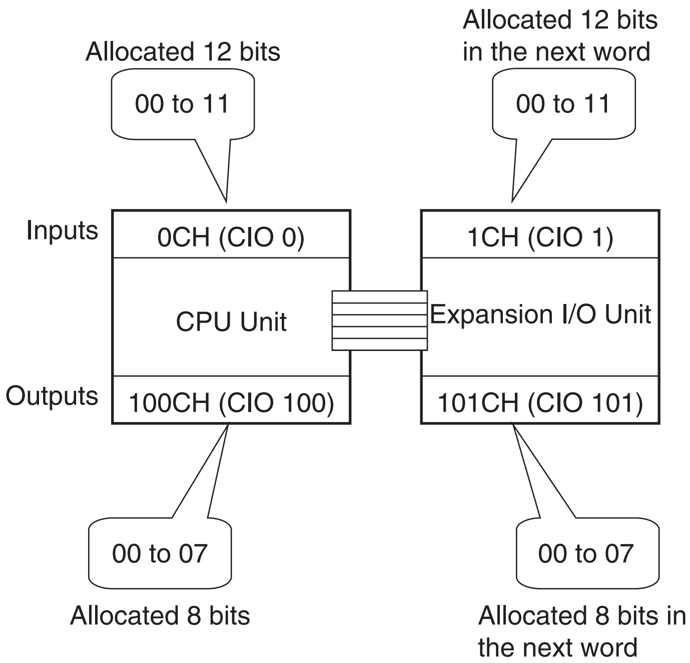
6-1 Allocation of Input Bits and Output Bits

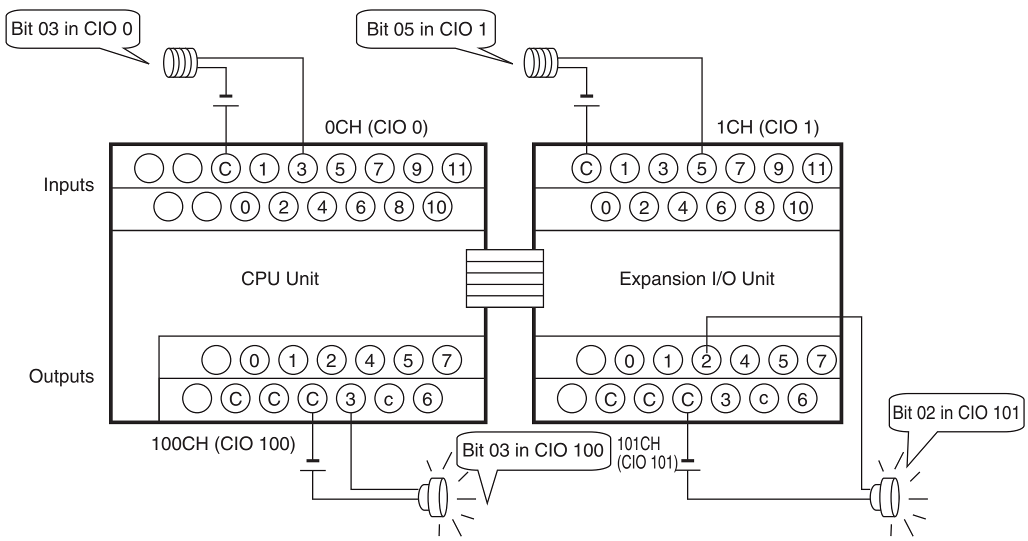
This section describes the allocation of input bits and output bits.

6-1-1 I/O Allocation

OMRON calls allocating I/O bits in memory "I/O allocation."

The I/O on Expansion I/O Units are allocated I/O bits in the words following the allocated words to the built-in I/O on the CPU Units.





6-1-2 I/O Allocation Concepts

The CPU Unit automatically allocates I/O bits to the Expansion I/O Units and Expansion Units, if connected when the power supply is turned ON.

It is not necessary to specify I/O bits allocation.

6-1-3 Allocations on the CPU Unit

● Input bits are allocated from CIO 0 and output bits are allocated from CIO 100

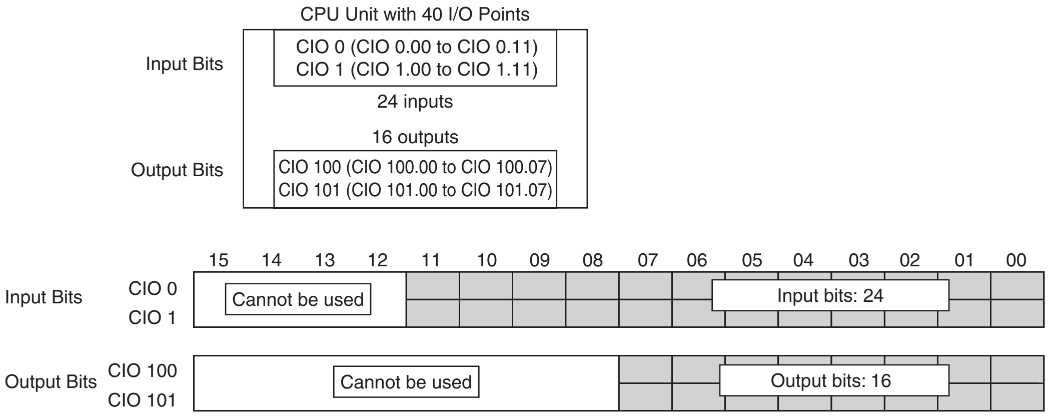
The first word from which input bits are allocated is CIO 0. The first word from which output bits are allocated is CIO 100. These cannot be changed.

● Words Allocated by the System and the Number of Connected Units

The starting words for inputs and outputs are predetermined for a CP2E CPU Unit. Input bits in CIO 0, or CIO 0 and CIO 1, and output bits in CIO 100, or CIO 100 and CIO 101, are automatically allocated to the built-in I/O on the CPU Unit.
The words from which bits are allocated by the system and the number of Expansion I/O Units and Expansion Units that can be connected are given in the following table.

CPU Unit	Allocated words		Number of Expansion Units and Expansion I/O Units connected
	Input Bits	Output Bits	
14 or 20 I/O Point CPU Unit	CIO 0	CIO 100	0 Unit
30 or 40 I/O Point CPU Unit	CIO 0 and CIO 1	CIO 100 and CIO 101	3 Units
60 I/O Point CPU Unit	CIO 0, CIO 1 and CIO 2	CIO 100, CIO 101 and CIO 102	3 Units

● Application Example: CPU Unit with 40 I/O Points



For a CPU Unit with 40 I/O points, 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 CIO 1).

6-1-4 Allocations to Expansion Units and Expansion I/O Units

Expansion Units and Expansion I/O Units connected to the CPU Unit are automatically allocated input bits and output bits in words following those allocated to the CPU Unit.

For example, if a CPU Unit with 40 I/O points is used, CIO 0 and CIO 1 are allocated for inputs and CIO 100 and CIO 101 are allocated for outputs. Thus, words from CIO 2 onward for inputs and words from CIO 102 onward for outputs are automatically allocated to the Expansion I/O Units and Expansion Units in the order that the Units are connected.

Allocations to Expansion I/O Units

There are Expansion I/O Units for expanding inputs, for expanding outputs, and for expanding both input and outputs.

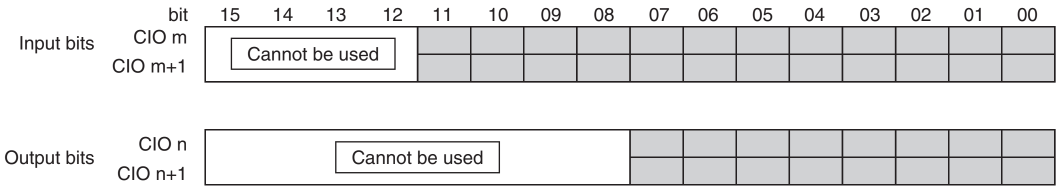
I/O bits starting from bit 00 in the next word after the word allocated to the previous Expansion Unit, Expansion I/O Unit, or CPU Unit are automatically allocated. This word is indicated as “CIO m” for input words and as “CIO n” for output words.

Model			Input bits			Output bits		
			No. of bits	No. of words	Addresses	No. of bits	No. of words	Addresses
8-point Input Unit		CP1W-8ED	8	1	CIO m, bits 00 to 07	–	None	None
8-point Output Unit	Relay outputs	CP1W-8ER	–	None	None	8	1	CIO n, bits 00 to 07
	Sinking transistor outputs	CP1W-8ET						
	Sourcing transis-tor outputs	CP1W-8ET1						
16-point Output Unit	Relay outputs	CP1W-16ER	–	None	None	16	2	CIO n, bits 00 to 07 CIO n+1, bits 00 to 07
	Sinking transistor outputs	CP1W-16ET						
	Sourcing transis-tor outputs	CP1W-16ET1						
20-point I/O Units	Relay outputs	CP1W-20EDR1	12	1	CIO m, bits 00 to 11	8	1	CIO n, bits 00 to 07
	Sinking transistor outputs	CP1W-20EDT						
	Sourcing transis-tor outputs	CP1W-20EDT1						
32-point Output Unit	Relay outputs	CP1W-32ER	–	None	None	32	4	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
	Sinking transistor outputs	CP1W-32ET						
	Sourcing transis-tor outputs	CP1W-32ET1						
40-point I/O Unit	Relay outputs	CP1W-40EDR	24	2	CIO m, bits 00 to 11 CIO m+1, bits 00 to 11	16	2	CIO n, bits 00 to 07 CIO n+1, bits 00 to 07
	Sinking transistor outputs	CP1W-40EDT						
	Sourcing transis-tor outputs	CP1W-40EDT1						

● I/O Bits Allocation with Expansion I/O Units Connected

Allocation Example: Expansion I/O Unit with 40 I/O Points (CP1W-40ED☐)

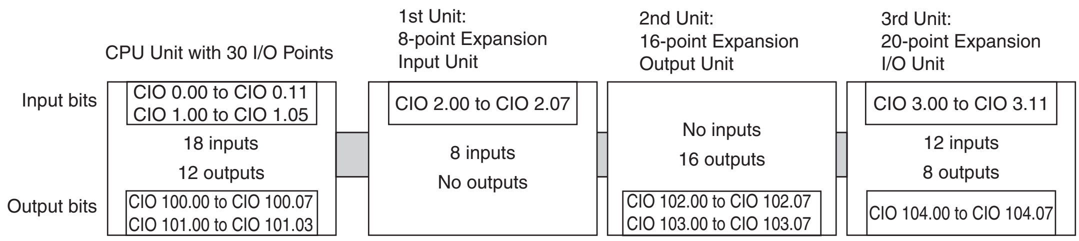
Twenty-four input bits in two words are allocated (bits 00 to 11 in CIO m and bits 00 to 11 CIO m+1). Sixteen output bits in two words are allocated in two words (bits 00 to 07 in CIO n and bits 00 to 07 in CIO n+1).

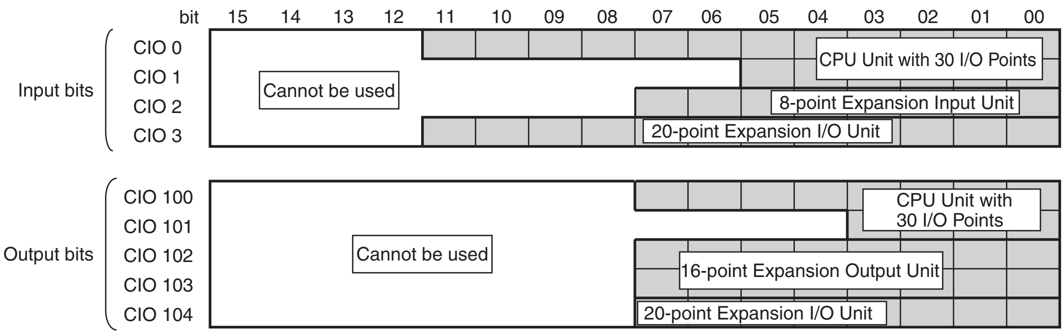


Two input words (24 bits) and two output words (16 bits) are allocated to a 40-point I/O Unit.

Allocation Example: Expansion Input Units and Expansion Output Units

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





Allocations for Expansion Units

● I/O Word Allocations to Expansion Units

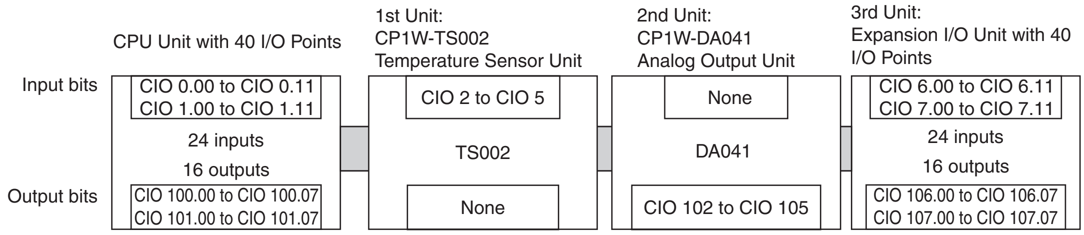
m: Indicates the next input word after the input word allocated to the Expansion Unit, Expansion I/O Unit, or CPU Unit connected to the left of the current Unit.

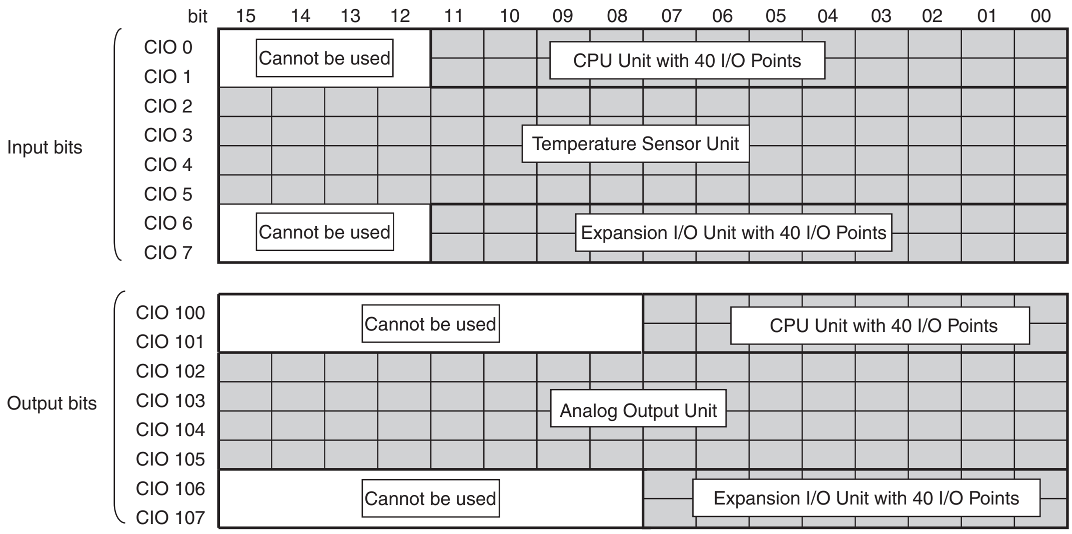
n: Indicates the next output word after the output word allocated to the Expansion Unit, Expansion I/O Unit, or CPU Unit connected to the left of the current Unit.

Name	Model number	Input words		Output words	
		No. of words	Addresses	No. of words	Addresses
Analog I/O Unit	CP1W-MAD11	2 words	CIO m to m+1	1 word	CIO n
	CP1W-MAD42	4 words	CIO m to m+3	2 word	CIO n to CIO n+1
	CP1W-MAD44	4 words	CIO m to m+3	4 word	CIO n to CIO n+3
Analog Input Unit	CP1W-AD041	4 words	CIO m to m+3	2 words	CIO n to CIO n+1
	CP1W-AD042	4 words	CIO m to m+3	2 words	CIO n to CIO n+1
Analog Output Unit	CP1W-DA021	None	–	2 words	CIO n to CIO n+1
	CP1W-DA041	None	–	4 words	CIO n to CIO n+3
	CP1W-DA042	None	–	4 words	CIO n to CIO n+3
Temperature Sensor Units	CP1W-TS001	2 words	CIO m to m+1	None	–
	CP1W-TS002	4 words	CIO m to m+3	None	–
	CP1W-TS003	4 words	CIO m to m+3	None	–
	CP1W-TS004	2 words	CIO m to m+1	1 word	CIO n
	CP1W-TS101	2 words	CIO m to m+1	None	–
	CP1W-TS102	4 words	CIO m to m+3	None	–

● I/O Word Allocations to Expansion Units

Allocation Example: CPU Unit with 40 I/O Points + Temperature Senser Unit (TS002) + Analog Output Unit (DA041) + Expansion I/O Unit with 40 I/O points





7

PLC Setup

This section describes the parameters in the PLC Setup, which are used to make basic settings for the CP2E CPU Unit.

7-1 Overview of the PLC Setup

The PLC Setup contains basic CPU Unit software parameter settings that the user can change to customize PLC operation.
These settings can be changed from a CX-Programmer. Change the PLC Setup in the following case. There is no need to reset, if the default (initial) settings are correct.
The setting from the CX-Programmer are saved to the built-in Flash Memory.

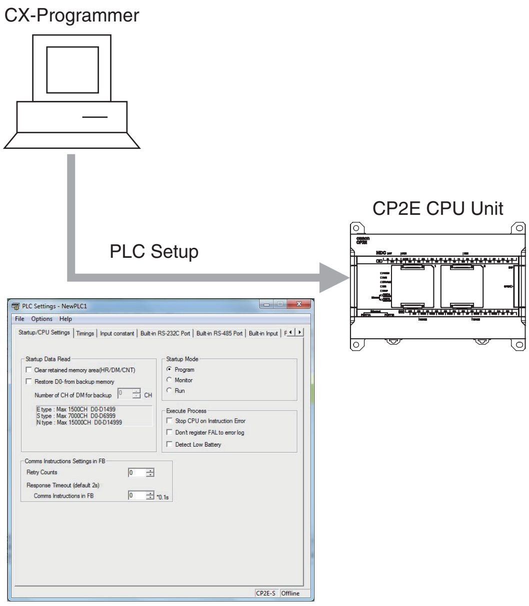
Application	Parameter
Reading the DM area words saved to the backup memory when power is turned ON.	Startup Data Read
Changing the Startup Mode to PROGRAM or MONITOR mode when debugging.	Startup Mode
Detection of low-battery errors is not required when using battery-free operation.	Detect Low Battery
Finding instruction errors when debugging.	Stop CPU on Instruction Error
A minimum cycle time setting to create a consistent I/O refresh cycle.	Minimum Cycle Time
Setting a watch cycle time.	Watch Cycle Time
Recording user-defined errors for FAL in the error log is not required.	FAL Error Log Registration

● Related Auxiliary Area Flags

Name	Word	Description	Read/write
PLC Setup Error Flag (Non-fatal error)	A402.10	ON when there is a setting error in the PLC Setup.	Read only

● Setting Methods for the PLC Setup

Set using the CX-Programmer



7-2 PLC Setup Settings

7-2-1 Startup and CPU Unit Settings

Startup Data Read Setting

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Clear retained memory area (HR/DM/CNT)	Do not clear.	Do not clear.	When power is turned ON
			Clear.	
2	Read D0- from backup memory	Do not read.	Do not read.	When power is turned ON
			Read.	
3	Number of CH of DM for backup	0	E□□-type CPU Unit: 0 to 1,499	When power is turned ON
			S□□-type CPU Unit: 0 to 6,999	
			N□□-type CPU Unit: 0 to 14,999	

Startup Mode Setting

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Startup Mode Setting	Run: RUN mode	Program: PROGRAM mode	When power is turned ON
			Monitor: MONITOR mode	
			Run: RUN mode	

Execute Process Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Stop CPU on Instruction Error	Do not stop.	Do not stop.	At start of operation
			Stop.	
2	Do not register FAL to error log	Register.	Register.	When power is turned ON
			Do not register.	
3	Do not detect Low Battery (N/S□□-type CPU Unit only)	Do not detect.	Do not detect.	Every cycle
			Detect.	

Comms Instructions Settings in FB: Settings for Communications Instructions in Function Blocks

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Retry Counts	0	0	At start of operation
			:	
			15	
2	Response Timeout	2 s	0: 2 s	At start of operation
			1: 1 × 0.1 s	
			:	
			65535: 65,535 × 0.1 s	

7-2-2 Timing and Interrupt Settings

Timing and Interrupt Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Watch Cycle Time	No setting (1s)	Setting	At start of operation
			1: 1 × 10 ms	
			:	
			100: 100 × 10 ms	
2	Constant Cycle Time	No setting (variable)	Setting	At start of operation
			1ms	
			:	
			1,000 ms	

7-2-3 Input Constants Settings

Input Constants

	Name	Default	Possible settings	When setting is read by CPU Unit
1	0CH: CIO 0	8 ms	No filter (0 ms)	When power is turned ON
			1 ms	
			2 ms	
			4 ms	
			8 ms	
			16 ms	
			32 ms	
2	1CH: CIO 1	Same as above.	Same as above.	Same as above.
3	2CH: CIO 2			
4	3CH: CIO 3			
5	4CH: CIO 4			
6	5CH: CIO 5			
7	6CH: CIO 6			
8	7CH: CIO 7			
9	8CH: CIO 8			
10	9CH: CIO 9			
11	10CH: CIO 10			
12	11CH: CIO 11			
13	12CH: CIO 12			
14	13CH: CIO 13			
15	14CH: CIO 14			
16	15CH: CIO 15			
17	16CH: CIO 16			
18	17CH: CIO 17			

Note The input constants of CP1W-40EDR/EDT/EDT1 are always 16ms regardless of the settings.

7-2-4 Serial Option Port 1/Built-in RS-232C Port

The settings are applicable to the N□□-type CPU Units with Serial Option port 1, and the E/S□□-type CPU units with built-in RS-232C port.

Since this setting is reflected by power OFF and ON, the PLC Setup and the actual operation settings may be different. The actual operation settings can be confirmed in words A617.

Communications Settings

			Name	Default	Possible settings	When setting is read by CPU Unit
1			Communications Settings	Standard (9,600; 1, 7, 2, E) (Default settings)	Standard Baud rate: 9,600 bps Start bits: 1 bit Data length: 7 bits Parity: Even Stop bits: 2 bits	When power is turned ON
					Custom	
2			Mode (When custom settings have been selected.)	Host Link	Host Link	When power is turned ON
					NT Link (1:N)	
					RS-232C (No-protocol)	
					PC Link (Slave)	
					PC Link (Master)	
					Modbus-RTU Easy Master	
					Modbus-RTU Slave	
	2-1		Host Link Settings			
		2-1-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-1-2	Format (data length, stop bits, parity)	7 bits, 2 bits, even	7 bits, 2 bits, even	When power is turned ON
					7 bits, 2 bits, odd	
					7 bits, 2 bits, no parity	
					7 bits, 1 bit, even	
					7 bits, 1 bit, odd	
					7 bits, 1 bit, no parity	
					8 bits, 2 bits, even	
					8 bits, 2 bits, odd	
					8 bits, 2 bits, no parity	
					8 bits, 1 bit, even	
					8 bits, 1 bit, odd	
					8 bits, 1 bit, no parity	
		2-1-3	Unit Number	0	0	When power is turned ON
					:	
					31	
	2-2		NT Link (1:N) Settings			
		2-2-1	Baud	115,200 bps	38,400 bps (standard)	When power is turned ON
					115,200 bps (high speed)	
		2-2-2	No.NT/PC Link Max. (Highest unit number of PT that can be connected to the PLC)	1	0	When power is turned ON
					:	
					7	

	Name				Default	Possible settings	When setting is read by CPU Unit
2	2-3	RS-232C (No-protocol) Settings					
		2-3-1	Baud		9,600 bps	1,200 bps	When power is turned ON
						2,400 bps	
						4,800 bps	
						9,600 bps	
						19,200 bps	
						38,400 bps	
						57,600 bps	
						115,200 bps	
		2-3-2	Format (data length, stop bits, parity)		7 bits, 2 bits, even	7 bits, 2 bits, even	When power is turned ON
						7 bits, 2 bits, odd	
						7 bits, 2 bits, no parity	
						7 bits, 1 bit, even	
						7 bits, 1 bit, odd	
						7 bits, 1 bit, no parity	
						8 bits, 2 bits, even	
						8 bits, 2 bits, odd	
						8 bits, 2 bits, no parity	
						8 bits, 1 bit, even	
						8 bits, 1 bit, odd	
						8 bits, 1 bit, no parity	
		2-3-3	Start Code		Disable.	Disable.	When power is turned ON
						Set.	
		2-3-4	Start Code (setting)		00 Hex	00 Hex	When power is turned ON
						:	
						FF hex	
		2-3-5	End Code		None (Received Bytes)	Received Bytes (no end code)	When power is turned ON
						CR, LF	
						Set End Code	
		2-3-6	Received Bytes (setting)		256 bytes	256 bytes	When power is turned ON
						1 byte	
						:	
						255 bytes	
		2-3-7	Set End Code (setting)		00 Hex	00 Hex	When power is turned ON
						:	
						FF Hex	
		2-3-8	Delay		0 ms	0 (×10 ms)	When power is turned ON
						:	
						9999 (×10 ms)	
	2-5	Modbus-RTU Easy Master Settings					
		2-5-1	Baud		9,600 bps	1,200 bps	When power is turned ON
						2,400 bps	
						4,800 bps	
						9,600 bps	
						19,200 bps	
						38,400 bps	
						57,600 bps	
						115,200 bps	

			Name	Default	Possible settings	When setting is read by CPU Unit
2	2-5	2-5-2	Format (data length, stop bits, parity)	7 bits, 2 bits, even	7 bits, 2 bits, even	When power is turned ON
					7 bits, 2 bits, odd	
					7 bits, 2 bits, no parity	
					7 bits, 1 bit, even	
					7 bits, 1 bit, odd	
					7 bits, 1 bit, no parity	
					8 bits, 2 bits, even	
					8 bits, 2 bits, odd	
					8 bits, 2 bits, no parity	
					8 bits, 1 bit, even	
					8 bits, 1 bit, odd	
					8 bits, 1 bit, no parity	
		2-5-3	Response Timeout	5 s	0 (5 s)	When power is turned ON
					1 (×100 ms)	
					:	
					255 (×100 ms)	
	2-6	PC Link (Slave) Settings				
		2-6-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-6-2	PLC Link Unit No.	0	0	When power is turned ON
					:	
					7	
	2-7	PC Link (Master) Settings				
		2-7-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-7-2	Link Words	10 Words	1 word	When power is turned ON
					:	
					10 words	
		2-7-3	PC Link Mode	ALL	ALL	When power is turned ON
					Masters	
		2-7-4	No.NT/PC Link Max. (Highest unit number of PT that can be connected to the PLC)	0	0	When power is turned ON
					:	
					7	

	Name			Default	Possible settings	When setting is read by CPU Unit
2	2-8	Modbus RTU Slave Settings				
		2-8-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-8-2	Format (data length, stop bits, parity)	8 bits, 1 bit, even	8 bits, 1 bit, even	When power is turned ON
					8 bits, 2 bits, no parity	
					8 bits, 1 bit, odd	
		2-8-3	Modbus Slave Address	1	0 (Slave Address 1)	When power is turned ON
					1	
					:	
					247	

7-2-5 Serial Option Port 2/Built-in RS-485 Port

The settings are applicable to the N30/40/60 CPU Units with Serial Option port 2, and the S30/40/60 CPU units with built-in RS-485 port.

Since this setting is reflected by power OFF and ON, the PLC Setup and the actual operation settings may be different. The actual operation settings can be confirmed in words A618.

Communications Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Communications Settings	Standard (9600; 1, 7, 2, E) (Default settings)	Standard Baud rate: 9,600 bps Start bits: 1 bit Data length: 7 bits Parity: Even Stop bits: 2 bits	When power is turned ON
			Custom	
2	Mode	Host Link	Host Link	When power is turned ON
			NT Link (1:N)	
			RS-232C (No-protocol)	
			PC Link (Slave)	
			PC Link (Master)	
			Modbus-RTU Easy Master	
			Modbus-RTU Slave	
2-1	Host Link Settings			
2-1-1	Baud	9,600 bps	1,200 bps	When power is turned ON
			2,400 bps	
			4,800 bps	
			9,600 bps	
			19,200 bps	
			38,400 bps	
			57,600 bps	
			115,200 bps	
2-1-2	Format (data length, stop bits, parity)	7 bits, 2 bits, even	7 bits, 2 bits, even	When power is turned ON
			7 bits, 2 bits, odd	
			7 bits, 2 bits, no parity	
			7 bits, 1 bit, even	
			7 bits, 1 bit, odd	
			7 bits, 1 bit, no parity	
			8 bits, 2 bits, even	
			8 bits, 2 bits, odd	
			8 bits, 2 bits, no parity	
			8 bits, 1 bit, even	
			8 bits, 1 bit, odd	
			8 bits, 1 bit, no parity	
2-1-3	Unit Number	0	0	When power is turned ON
			:	
			31	
2-2	NT Link (1:N) Settings			
2-2-1	Baud	115,200 bps	38,400 bps (standard)	When power is turned ON
			115,200 bps (high speed)	
2-2-2	No. NT/PC Link Max. (Highest unit number of PT that can be connected to the PLC)	1	0	When power is turned ON
			:	
			7	

			Name	Default	Possible settings	When setting is read by CPU Unit
2	2-3	RS-232C (No-protocol) Settings				
		2-3-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-3-2	Format (data length, stop bits, parity)	7 bits, 2 bits, even	7 bits, 2 bits, even	When power is turned ON
					7 bits, 2 bits, odd	
					7 bits, 2 bits, no parity	
					7 bits, 1 bit, even	
					7 bits, 1 bit, odd	
					7 bits, 1 bit, no parity	
					8 bits, 2 bits, even	
					8 bits, 2 bits, odd	
					8 bits, 2 bits, no parity	
					8 bits, 1 bit, even	
					8 bits, 1 bit, odd	
					8 bits, 1 bit, no parity	
		2-3-3	Start Code	Disable.	Disable.	When power is turned ON
					Set.	
		2-3-4	Start Code (setting)	00 hex	00 hex	When power is turned ON
					:	
					FF hex	
		2-3-5	End Code	None (Received Bytes)	Received Bytes (no end code)	When power is turned ON
					CR, LF	
					Set End Code	
		2-3-6	Received Bytes (setting)	256 bytes	256 bytes	When power is turned ON
					1 byte	
					:	
					255 bytes	
		2-3-7	Set End Code (setting)	00 hex	00 hex	When power is turned ON
					:	
					FF hex	
		2-3-8	Delay	0 ms	0 (×10 ms)	When power is turned ON
					:	
					9999 (×10 ms)	
	2-5	Modbus-RTU Easy Master Settings				
		2-5-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	

				Default	Possible settings	When setting is read by CPU Unit
	Name					
2	2-5	2-5-2	Format (data length, stop bits, parity)	7 bits, 2 bits, even	7 bits, 2 bits, even	When power is turned ON
					7 bits, 2 bits, odd	
					7 bits, 2 bits, no parity	
					7 bits, 1 bit, even	
					7 bits, 1 bit, odd	
					7 bits, 1 bit, no parity	
					8 bits, 2 bits, even	
					8 bits, 2 bits, odd	
					8 bits, 2 bits, no parity	
					8 bits, 1 bit, even	
					8 bits, 1 bit, odd	
					8 bits, 1 bit, no parity	
		2-5-3	Response Timeout	5 s	0 (5 s)	When power is turned ON
					1 (×100 ms)	
					:	
					255 (×100 ms)	
	2-6	PC Link (Slave) Settings				
		2-6-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-6-2	PLC Link Unit No.	0	0	When power is turned ON
					:	
					7	
	2-7	PC Link (Master) Settings				
		2-7-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-7-2	Link Words	10 words	1 word	When power is turned ON
					:	
					10 words	
		2-7-3	PC Link Mode	ALL	ALL	When power is turned ON
					Masters	
		2-7-4	No. NT/PC Link Max. (Highest unit number of PT that can be connected to the PLC)	0	0	When power is turned ON
					:	
					7	

	Name			Default	Possible settings	When setting is read by CPU Unit
2	2-8	Modbus RTU Slave Settings				
		2-8-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-8-2	Format (data length, stop bits, parity)	8 bits, 1 bit, even	8 bits, 1 bit, even	When power is turned ON
					8 bits, 2 bits, no parity	
					8 bits, 1 bit, odd	
		2-8-3	Modbus Slave Address	1	0 (Slave Address 1)	When power is turned ON
					1	
					:	
					247	

7-2-6 Serial Option Port 1 (EX)

The settings are applicable to the N☐☐-type CPU Units when Serial Option Board CP2W-CIFD☐ with 2 ports is mounted.

Since this setting is reflected by power OFF and ON, the PLC Setup and the actual operation settings-may be different. The actual operation settings can be confirmed in words A616.

Communications Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Communications Settings	Standard (9600; 1, 7, 2, E) (Default settings)	Standard Baud rate: 9,600 bps Start bits: 1 bit Data length: 7 bits Parity: Even Stop bits: 2 bits	When power is turned ON
			Custom	
2	Mode (When custom settings have been selected.)	RS-232C (No-protocol)	RS-232C (No-protocol)	When power is turned ON
			PC Link (Slave)	
			PC Link (Master)	
			Modbus-RTU Easy Master	
			Modbus RTU Slave	
2-1	RS-232C (No-protocol) Settings			
2-1-1	Baud	9,600 bps	1,200 bps	When power is turned ON
			2,400 bps	
			4,800 bps	
			9,600 bps	
			19,200 bps	
			38,400 bps	
			57,600 bps	
			115,200 bps	
2-1-2	Format (data length, stop bits, parity)	7 bits, 2 bits, even	7 bits, 2 bits, even	When power is turned ON
			7 bits, 2 bits, odd	
			7 bits, 2 bits, no parity	
			7 bits, 1 bit, even	
			7 bits, 1 bit, odd	
			7 bits, 1 bit, no parity	
			8 bits, 2 bits, even	
			8 bits, 2 bits, odd	
			8 bits, 2 bits, no parity	
			8 bits, 1 bit, even	
			8 bits, 1 bit, odd	
			8 bits, 1 bit, no parity	
2-1-3	Start Code	Disable.	Disable.	When power is turned ON
			Set.	
2-1-4	Start Code (setting)	00 Hex	00 Hex	When power is turned ON
			:	
			FF hex	
2-1-5	End Code	None (Received Bytes)	Received Bytes (no end code)	When power is turned ON
			CR, LF	
			Set End Code	

			Name	Default	Possible settings	When setting is read by CPU Unit
2	2-1	2-1-6	Received Bytes (setting)	256 bytes	256 bytes	When power is turned ON
					1 byte	
					:	
					255 bytes	
		2-1-7	Set End Code (setting)	00 Hex	00 Hex	When power is turned ON
					:	
					FF Hex	
		2-1-8	Delay	0 ms	0 (×10 ms)	When power is turned ON
					:	
					9999 (×10 ms)	
2	2-3	Modbus-RTU Easy Master Settings				
		2-3-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-3-2	Format (data length, stop bits, parity)	7 bits, 2 bits, even	7 bits, 2 bits, even	When power is turned ON
					7 bits, 2 bits, odd	
					7 bits, 2 bits, no parity	
					7 bits, 1 bit, even	
					7 bits, 1 bit, odd	
					7 bits, 1 bit, no parity	
					8 bits, 2 bits, even	
					8 bits, 2 bits, odd	
					8 bits, 2 bits, no parity	
					8 bits, 1 bit, even	
					8 bits, 1 bit, odd	
					8 bits, 1 bit, no parity	
		2-3-3	Response Timeout	5 s	0 (5 s)	When power is turned ON
					1 (×100 ms)	
					:	
					255 (×100 ms)	
	2-4	PC Link (Slave) Settings				
		2-4-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-4-2	PLC Link Unit No.	0	0	When power is turned ON
					:	
					7	

	Name			Default	Possible settings	When setting is read by CPU Unit
2	2-5	PC Link (Master) Settings				
		2-5-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-5-2	Link Words	10 Words	1 word	When power is turned ON
					:	
					10 words	
		2-5-3	PC Link Mode	ALL	ALL	When power is turned ON
					Masters	
	2-5	2-5-4	No.NT/PC Link Max. (Highest unit number of PT that can be con-nected to the PLC)	0	0	When power is turned ON
					:	
					7	
	2-6	Modbus RTU Slave Settings				
		2-6-1	Baud	9,600 bps	1,200 bps	When power is turned ON
					2,400 bps	
					4,800 bps	
					9,600 bps	
					19,200 bps	
					38,400 bps	
					57,600 bps	
					115,200 bps	
		2-6-2	Format (data length, stop bits, par-ity)	7 bits, 2 bits, even	7 bits, 2 bits, even	When power is turned ON
					7 bits, 2 bits, odd	
					7 bits, 2 bits, no parity	
					7 bits, 1 bit, even	
					7 bits, 1 bit, odd	
					7 bits, 1 bit, no parity	
					8 bits, 2 bits, even	
					8 bits, 2 bits, odd	
					8 bits, 2 bits, no parity	
					8 bits, 1 bit, even	
					8 bits, 1 bit, odd	
					8 bits, 1 bit, no parity	
		2-6-3	Modbus Slave Address	1	0 (Slave Address 1)	When power is turned ON
					1	
					:	
					247	

7-2-7 Built-in Inputs

High-speed Counter Settings

			Name	Default	Possible settings	When setting is read by CPU Unit
1			Use high-speed counter 0	Do not use.	Do not use.	When power is turned ON
					Use.	
	1-1		Counting mode	Linear mode	Linear mode	At start of operation
					Circular mode	
		1-1-1	Circular Max. Count	0	0	At start of operation
					:	
					4,294,967,295	
	1-2		Reset **Note** Only a software reset can be set if an increment pulse input is set for the input setting.	Z phase, software reset (stop comparing)	Z phase, software reset (stop comparing)	When power is turned ON
					Software reset (stop comparing)	
					Phase Z, software reset (comparing)	
					Software reset (comparing)	
	1-3		Input Setting **Note** Make the same input setting for high-speed counters 0, 1, and 2.	Differential phase input (×4)	Differential phase input (×4)	When power is turned ON
					Pulse + direction input	
					Up/Down pulse input	
					Increment pulse input	
2			Use high-speed counter 1	Do not use.	Do not use.	When power is turned ON
					Use.	
	2-1		Counting mode	Linear mode	Linear mode	At start of operation
					Circular mode	
		2-1-1	Circular Max. Count	0	0	At start of operation
					:	
					4,294,967,295	
	2-2		Reset **Note** Only a software reset can be set if an increment pulse input is set for the input setting.	Z phase, software reset (stop comparing)	Z phase, software reset (stop comparing)	When power is turned ON
					Software reset (stop comparing)	
					Phase Z, software reset (comparing)	
					Software reset (comparing)	
	2-3		Input Setting **Note** Make the same input setting for high-speed counters 0, 1, and 2.	Differential phase input (×4)	Differential phase input (×4)	When power is turned ON
					Pulse + direction input	
					Up/Down pulse input	
					Increment Pulse input	
3			Use high-speed counter 2	Do not use.	Do not use.	When power is turned ON
					Use.	
	3-1		Counting mode	Linear mode	Linear mode	At start of operation
					Circular mode	
		3-1-1	Circular Max. Count	0	0	At start of operation
					:	
					4,294,967,295	
	3-2		Reset	Software reset	Software reset	When power is turned ON
					Software reset (comparing)	
	3-3		Input Setting	Increment pulse input	Increment pulse input	When power is turned ON

			Name	Default	Possible settings	When setting is read by CPU Unit
4	Use high-speed counter 3			Do not use.	Do not use. Use.	When power is turned ON
	4-1	Counting mode		Linear mode	Linear mode Circular mode	At start of operation
		4-1-1	Circular Max. Count	0	0 : 4,294,967,295	At start of operation
	4-2	Reset		Software reset	Software reset Software reset (comparing)	When power is turned ON
	4-3	Input Setting		Increment pulse input	Increment pulse input	When power is turned ON
5	Use high-speed counter 4			Do not use.	Do not use. Use.	When power is turned ON
	5-1	Counting mode		Linear mode	Linear mode Circular mode	At start of operation
		5-1-1	Circular Max. Count	0	0 : 4,294,967,295	At start of operation
	5-2	Reset		Software reset	Software reset Software reset (comparing)	When power is turned ON
	5-3	Input Setting		Increment pulse input	Increment pulse input	When power is turned ON
6	Use high-speed counter 5			Do not use.	Do not use. Use.	When power is turned ON
	6-1	Counting mode		Linear mode	Linear mode Circular mode	At start of operation
		6-1-1	Circular Max. Count	0	0 : 4,294,967,295	At start of operation
	6-2	Reset		Software reset	Software reset Software reset (comparing)	When power is turned ON
	6-3	Input Setting		Increment pulse input	Increment pulse input	When power is turned ON

Interrupt Input Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	IN2: CIO 0.02	Normal	Normal Interrupt Quick	When power is turned ON
2	IN3: CIO 0.03	Normal	Normal Interrupt Quick	When power is turned ON
3	IN4: CIO 0.04	Normal	Normal Interrupt Quick	When power is turned ON
4	IN5: CIO 0.05	Normal	Normal Interrupt Quick	When power is turned ON
5	IN6: CIO 0.06	Normal	Normal Interrupt Quick	When power is turned ON
6	IN7: CIO 0.07	Normal	Normal Interrupt Quick	When power is turned ON

	Name	Default	Possible settings	When setting is read by CPU Unit
7	IN8: CIO 0.08 (N20/30/40/60 CPU Unit only)	Normal	Normal	When power is turned ON
			Interrupt	
			Quick	
8	IN9: CIO 0.09 (N20/30/40/60 CPU Unit only)	Normal	Normal	When power is turned ON
			Interrupt	
			Quick	

7-2-8 Pulse Output 0 Settings

The settings are applicable to the N/S☐☐-type CPU Units with transistor outputs.

Base Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Undefined Origin (operation for limit signal turning ON)	Hold	Hold	At start of operation
			Undefined	
2	Limit Input Signal Operation	Search Only	Search Only	When power is turned ON
			Always	
3	Limit Input Signal	NC	NC	At start of operation
			NO	
4	Search/Return Initial Speed	0 pps	0 pps	At start of operation
			:	
			100,000 pps	

Origin Search Settings

		Name	Default	Possible settings	When setting is read by CPU Unit
1	Use define origin operation		Do not use.	Do not use.	When power is turned ON
				Use.	
	1-1	Search Direction	CW	CW	At start of operation
				CCW	
	1-2	Detection Method	Method 0	Method 0	At start of operation
				Method 1	
				Method 2	
	1-3	Search Operation	Inverse 1	Inverse 1	At start of operation
				Inverse 2	
	1-4	Operation Mode	Mode 0	Mode 0	At start of operation
				Mode 1	
				Mode 2	
	1-5	Origin Input Signal	NC	NC	At start of operation
				NO	
	1-6	Proximity Input Signal	NC	NC	At start of operation
				NO	
	1-7	Search High Speed	0 pps (disabled)	1 pps	At start of operation
				:	
				100,000 pps	
	1-8	Search Proximity Speed	0 pps (disabled)	1 pps	At start of operation
				:	
				100,000 pps	

	Name		Default	Possible settings	When setting is read by CPU Unit
1	1-9	Origin Compensation Value	0 pps	-2,147,483,648	At start of operation
				:	
				0	
				:	
				+2,147,483,647	
	1-10	Origin Search Acceleration Ratio (Rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
				:	
				65,535 (pulse/4 ms)	
	1-11	Origin Search Deceleration Ratio (Rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
				:	
				65,535 (pulse/4 ms)	
	1-12	Positioning Monitor Time	0 (ms)	0 (ms)	At start of operation
				:	
				9,999 (ms)	

Origin Return Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Speed	0 pps (disabled)	1 pps	At start of operation
			:	
			100,000 pps	
2	Acceleration Ratio (rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
			:	
			65,535 (pulse/4 ms)	
3	Deceleration rate	0 (disabled)	1 (pulse/4 ms)	At start of operation
			:	
			65,535 (pulse/4 ms)	

7-2-9 Pulse Output 1 Settings

The settings are applicable to the N/S☐☐-type CPU Units with transistor outputs.

Base Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Undefined Origin (operation for limit signal turning ON)	Hold	Hold	At start of operation
			Undefined	
2	Limit Input Signal Operation	Search Only	Search Only	When power is turned ON
			Always	
3	Limit Input Signal	NC	NC	At start of operation
			NO	
4	Search/Return Initial Speed	0 pps	0 pps	At start of operation
			:	
			100,000 pps	

Origin Search Settings

	Name		Default	Possible settings	When setting is read by CPU Unit
1	Use define origin operation		Do not use.	Do not use.	When power is turned ON
				Use.	
	1-1	Search Direction	CW	CW	At start of operation
				CCW	
	1-2	Detection Method	Method 0	Method 0	At start of operation
				Method 1	
				Method 2	
	1-3	Search Operation	Inverse 1	Inverse 1	At start of operation
				Inverse 2	
	1-4	Operation Mode	Mode 0	Mode 0	At start of operation
				Mode 1	
				Mode 2	
	1-5	Origin Input Signal	NC	NC	When power is turned ON
				NO	
	1-6	Proximity Input Signal	NC	NC	At start of operation
				NO	
	1-7	Search High Speed	0 pps (disabled)	1 pps	At start of operation
				:	
				100,000 pps	
	1-8	Search Proximity Speed	0 pps (disabled)	1 pps	At start of operation
				:	
				100,000 pps	
	1-9	Origin Compensation Value	0 pps	-2,147,483,648	At start of operation
				:	
				0	
				:	
				+2,147,483,647	
	1-10	Origin Search Acceleration Ratio (Rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
				:	
				65,535 (pulse/4 ms)	
	1-11	Origin Search Deceleration Ratio (Rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
				:	
				65,535 (pulse/4 ms)	
	1-12	Positioning Monitor Time	0 (ms)	0 (ms)	At start of operation
				:	
				9,999 (ms)	

Origin Return Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Speed	0 pps (disabled)	1 pps	At start of operation
			:	
			100,000 pps	
2	Acceleration Ratio (rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
			:	
			65,535 (pulse/4 ms)	
3	Deceleration rate	0 (disabled)	1 (pulse/4 ms)	At start of operation
			:	
			65,535 (pulse/4 ms)	

7-2-10 Pulse Output 2 Settings

The settings are applicable to the N30/40/60 CPU Units with transistor outputs.

Base Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Undefined Origin (operation for limit signal turning ON)	Hold	Hold	At start of operation
			Undefined	
2	Limit Input Signal Operation	Search Only	Search Only	When power is turned ON
			Always	
3	Limit Input Signal	NC	NC	At start of operation
			NO	
4	Search/Return Initial Speed	0 pps	0 pps	At start of operation
			:	
			100,000 pps	

Origin Search Settings

	Name		Default	Possible settings	When setting is read by CPU Unit
1	Use define origin operation		Do not use.	Do not use.	When power is turned ON
				Use.	
	1-1	Search Direction	CW	CW	At start of operation
				CCW	
	1-2	Detection Method	Method 0	Method 0	At start of operation
				Method 1	
				Method 2	
	1-3	Search Operation	Inverse 1	Inverse 1	At start of operation
				Inverse 2	
	1-4	Operation Mode	Mode 0	Mode 0	At start of operation
				Mode 1	
				Mode 2	
	1-5	Origin Input Signal	NC	NC	At start of operation
				NO	
	1-6	Proximity Input Signal	NC	NC	At start of operation
				NO	

		Name	Default	Possible settings	When setting is read by CPU Unit
1	1-7	Search High Speed	0 pps (disabled)	1 pps	At start of operation
				:	
				100,000 pps	
	1-8	Search Proximity Speed	0 pps (disabled)	1 pps	At start of operation
				:	
				100,000 pps	
	1-9	Origin Compensation Value	0 pps	-2,147,483,648	At start of operation
				:	
				0	
				:	
				+2,147,483,647	
	1-10	Origin Search Acceleration Ratio (Rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
				:	
				65,535 (pulse/4 ms)	
	1-11	Origin Search Deceleration Ratio (Rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
				:	
				65,535 (pulse/4 ms)	
	1-12	Positioning Monitor Time	0 (ms)	0 (ms)	At start of operation
				:	
				9,999 (ms)	

Origin Return Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Speed	0 pps (disabled)	1 pps	At start of operation
			:	
			100,000 pps	
2	Acceleration Ratio (rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
			:	
			65,535 (pulse/4 ms)	
3	Deceleration rate	0 (disabled)	1 (pulse/4 ms)	At start of operation
			:	
			65,535 (pulse/4 ms)	

7-2-11 Pulse Output 3 Settings

The settings are applicable to the N30/40/60 CPU Units with transistor outputs.

Base Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Undefined Origin (operation for limit signal turning ON)	Hold	Hold	At start of operation
			Undefined	
2	Limit Input Signal Operation	Search Only	Search Only	When power is turned ON
			Always	
3	Limit Input Signal	NC	NC	At start of operation
			NO	
4	Search/Return Initial Speed	0 pps	0 pps	At start of operation
			:	
			100,000 pps	

Origin Search Settings

	Name		Default	Possible settings	When setting is read by CPU Unit
1	Use define origin operation		Do not use.	Do not use.	When power is turned ON
				Use.	
	1-1	Search Direction	CW	CW	At start of operation
				CCW	
	1-2	Detection Method	Method 0	Method 0	At start of operation
				Method 1	
				Method 2	
	1-3	Search Operation	Inverse 1	Inverse 1	At start of operation
				Inverse 2	
	1-4	Operation Mode	Mode 0	Mode 0	At start of operation
				Mode 1	
				Mode 2	
	1-5	Origin Input Signal	NC	NC	When power is turned ON
				NO	
	1-6	Proximity Input Signal	NC	NC	At start of operation
				NO	
	1-7	Search High Speed	0 pps (disabled)	1 pps	At start of operation
				:	
				100,000 pps	
	1-8	Search Proximity Speed	0 pps (disabled)	1 pps	At start of operation
				:	
				100,000 pps	
	1-9	Origin Compensation Value	0 pps	-2,147,483,648	At start of operation
				:	
				0	
				:	
				+2,147,483,647	
	1-10	Origin Search Acceleration Ratio (Rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
				:	
				65,535 (pulse/4 ms)	
	1-11	Origin Search Deceleration Ratio (Rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
				:	
				65,535 (pulse/4 ms)	
	1-12	Positioning Monitor Time	0 (ms)	0 (ms)	At start of operation
				:	
				9,999 (ms)	

Origin Return Settings

	Name	Default	Possible settings	When setting is read by CPU Unit
1	Speed	0 pps (disabled)	1 pps	At start of operation
			:	
			100,000 pps	
2	Acceleration Ratio (rate)	0 (disabled)	1 (pulse/4 ms)	At start of operation
			:	
			65,535 (pulse/4 ms)	
3	Deceleration rate	0 (disabled)	1 (pulse/4 ms)	At start of operation
			:	
			65,535 (pulse/4 ms)	

7-2-12 Built-in Ethernet Settings

The settings are applicable to the N☐☐-type CPU Units.

Ethernet Port Settings

	Name	Default	Possible Settings	When Setting is read by CPU Unit
1	Broadcast Address	4.3BSD	4.3BSD	When power is turned ON or when Ethernet is reset
			4.2BSD	
2	IP Address	0.0.0.0	0.0.0.0	When power is turned ON or when Ethernet is reset
			:	
			255.255.255.255	
3	Subnet Mask	0.0.0.0	0.0.0.0	When power is turned ON or when Ethernet is reset
			:	
			255.255.255.255	
4	TCP Keep-alive Time	120 min (Default value: 0)	0 (120 min)	When power is turned ON or when Ethernet is reset
			1 min	
			:	
			65,535 min	
5	FINS Node Address	Node address 1 (Default value: 0)	0 (1)	When power is turned ON or when Ethernet is reset
			:	
			254	

FINS/TCP Settings

	Name	Default	Possible Settings	When Setting is read by CPU Unit
6	FINS/TCP Port Number Mode	9600	Default (9600)	When power is turned ON or when Ethernet is reset
			User	
	FINS/TCP Port Number (User)	9600 (Default value: 0)	0 (9600)	When power is turned ON or when Ethernet is reset
			1	
			:	
			65,535	
7	IP Address Protect (Only for FINS/TCP Server)	Not Protect	Not Protect	When power is turned ON or when Ethernet is reset
			Protect	

		Name	Default	Possible Settings	When Setting is read by CPU Unit
8	FINS/TCP Connection Setting				
	FINS/TCP Connection No.1				
	8-1	FINS/TCP Server/Client Mode	Server	Server	When power is turned ON or when Ethernet is reset
				Client	
	8-2	Connection IP Address	0.0.0.0	0.0.0.0	When power is turned ON or when Ethernet is reset
				:	
				255.255.255.255	
	8-3	Auto FINS Node Allocation Address (Only valid for FINS/TCP Server mode)	251 (Default value: 0)	0 (251)	When power is turned ON or when Ethernet is reset
				:	
				253	
	8-4	Keep-alive	Disable	Disable	When power is turned ON or when Ethernet is reset
				Enable	
	FINS/TCP Connection No.2				
	8-5	FINS/TCP Server/Client Mode	Server	Server	When power is turned ON or when Ethernet is reset
				Client	
	8-6	Connection IP Address	0.0.0.0	0.0.0.0	When power is turned ON or when Ethernet is reset
				:	
				255.255.255.255	
	8-7	Auto FINS Node Allocation Address (Only valid for FINS/TCP Server mode)	252 (Default value: 0)	0 (252)	When power is turned ON or when Ethernet is reset
				:	
				253	
	8-8	Keep-alive	Disable	Disable	When power is turned ON or when Ethernet is reset
				Enable	
	FINS/TCP Connection No.3				
	8-9	FINS/TCP Server/Client Mode	Server	Server	When power is turned ON or when Ethernet is reset
				Client	
	8-10	Connection IP Address	0.0.0.0	0.0.0.0	When power is turned ON or when Ethernet is reset
				:	
				255.255.255.255	
	8-11	Auto FINS Node Allocation Address (Only valid for FINS/TCP Server mode)	253 (Default value: 0)	0 (253)	When power is turned ON or when Ethernet is reset
				:	
				253	
	8-12	Keep-alive	Disable	Disable	When power is turned ON or when Ethernet is reset
				Enable	

FINS/UDP Settings

	Name	Default	Possible Settings	When Setting is read by CPU Unit
9	FINS/UDP Port Number Mode	9600	Default (9600)	When power is turned ON or when Ethernet is reset
			User	
	FINS/UDP Port Number (User)	9600 (Default value: 0)	0 (9600)	When power is turned ON or when Ethernet is reset
			1	
			:	
			65,535	
10	IP Address Conversion Method	Auto Method	Auto Method	When power is turned ON or when Ethernet is reset
			Auto Method (Static)	
			Combined Method	
			IP Address Table Reference Method	
11	FINS/UDP Destination IP Mode	Dynamically	Dynamically	When power is turned ON or when Ethernet is reset
			Static	

DNS Server Settings

	Name	Default	Possible Settings	When Setting is read by CPU Unit
12	DNS Server IP Address	0.0.0.0	0.0.0.0	When power is turned ON or when Ethernet is reset
			:	
			255.255.255.255	
13	DNS Port Number	53 (Default value: 0)	0 (53)	When power is turned ON or when Ethernet is reset
			1	
			:	
			65,535	
14	Retry Time	10s (Default value: 0)	0 (10s)	When power is turned ON or when Ethernet is reset
			1s	
			:	
			65,535s	

SNTP Server

	Name		Default	Possible Settings	When setting is read by CPU Unit
15	Auto Adjust Time Mode		Disable	Disable	When power is turned ON or when Ethernet is reset
				Enable	
	15-1	Auto Adjust Time	0h0m0s	0h0m0s	When power is turned ON or when Ethernet is reset
				:	
				23h59m59s	
	15-2	SNTP Server Specify Method	IP Address	IP Address	When power is turned ON or when Ethernet is reset
				Host Name	
		SNTP Server IP Address	0.0.0.0	0.0.0.0	When power is turned ON or when Ethernet is reset
				:	
				255.255.255.255	
		SNTP Server Host Name (46 ASCII)	Blank	Any String	When power is turned ON or when Ethernet is reset

	Name		Default	Possible Settings	When setting is read by CPU Unit
15	15-3	SNTP Port Number	123 (Default value: 0)	0 (123)	When power is turned ON or when Ethernet is reset
				1	
				:	
				65,535	
	15-4	Retry Time	10s (Default value: 0)	0 (10s)	When power is turned ON or when Ethernet is reset
				1s	
				:	
				255s	
	15-5	Difference Time	+0h0m	-12h00m	When power is turned ON or when Ethernet is reset
				:	
				+13h00m	

8

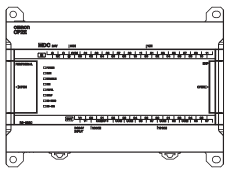
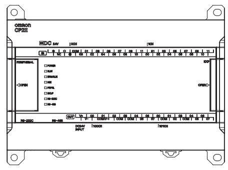
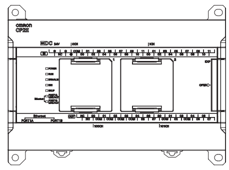
Overview of Built-in Functions and Allocations

This section describes the built-in functions, overall procedure, and allocations for functions of the CP2E.

8

8-1 Built-in Functions

The following built-in functions are provided by the CP2E CPU Units.

Type / Function	CP2E E□□-type CPU Units	CP2E S□□-type CPU Units	CP2E N□□-type CPU Units	Reference
Appearance				
Quick-response inputs	6 inputs	6 inputs	N14 CPU Units: 6 inputs N20/30/40/60 CPU Units: 8 inputs	*Section 9*
Input interrupts	6 inputs	6 inputs	N14 CPU Units: 6 inputs N20/30/40/60 CPU Units: 8 inputs	*Section 10*
Scheduled interrupts	1 interrupt	1 interrupt	1 interrupt	
High-speed counter	• Incremental: 100 kHz×2 counters, 10 kHz×4 counters • Up/down: 100 kHz×1 counter, 10 kHz×1 counter • Pulse plus direction: 100 kHz×2 counters • Differential phases (4×): 50 kHz×1 counter, 5 kHz×1 counter	• Incremental: 100 kHz×2 counters, 10 kHz×4 counters • Up/down: 100 kHz×1 counter, 10 kHz×1 counter • Pulse plus direction: 100 kHz×2 counters • Differential phases (4×): 50 kHz×1 counter, 5 kHz×1 counter	N14/20 CPU Units: • Incremental: 100 kHz×2 counters, 10 kHz×4 counters • Up/down: 100 kHz×1 counter, 10 kHz×1 counter • Pulse plus direction: 100 kHz×2 counters • Differential phases (4×): 50 kHz×1 counter, 5 kHz×1 counter N30/40/60 CPU Units: • Incremental: 100 kHz×3 counters, 10 kHz×3 counters • Up/down: 100 kHz×2 counters • Pulse plus direction: 100 kHz×2 counters • Differential phases (4×): 50 kHz×2 counters	*Section 11*
Pulse outputs	Not supported	2 outputs (pulse plus direction only) An external power supply is required for pulse outputs.	N14/20 CPU Units: 2 outputs (pulse plus direction only) N30/40/60 CPU Units: 4 outputs (pulse plus direction only) An external power supply is required for pulse outputs for N30/40/60 CPU Units.	*Section 12*
PWM outputs	Not supported	1 output An external power supply is required for PWM outputs.	1 output An external power supply is required for PWM outputs for N30/40/60 CPU Units.	*Section 13*
Built-in serial communications	1 port (RS-232C)	2 ports (RS-232C+RS-485)	Not supported	*Section 14*
Optional serial communications	Not supported	Not supported	N14/20 CPU Units: 2 ports max. N30/40/60 CPU Units: 3 ports max. (Using Serial Option Board with 2 ports)	*Section 14* *Section 17*
Built-in Ethernet	Not supported	Not supported	N14/20 CPU Unit: 1 port N30/40/60 CPU Units: 2 ports (Switch function)	*Section 15*
PID temperature control	Supported	Supported	Supported	*Section 16-1*
Clock functions	Not supported	Supported (While power is supplied)	Supported (While power is supplied)	*Section 16-2*
DM backup	Supported	Supported	Supported	*Section 16-3*
Security function	Supported	Supported	Supported	*Section 16-4*

8-2 Overall Procedure for Using CP2E Built-in Functions

The overall procedure for using built-in CP2E functions is described in this section.

	Step	Description	Reference
1	Select Functions	Select the functions to use. Example: Interrupts, high-speed counter inputs, and pulse outputs.	
2	Make the Settings in the PLC Setup	Set the functions with the applicable numbers using the CX-Programmer. Example: Using input interrupt IN3 and high-speed counter 0, parameters in the PLC Setup must be set for the following functions. • Input interrupts • Quick-response inputs • High-speed counters • Origin searches • Minimum cycle time • Serial communications • Ethernet • Analog Option Board	Refer to *Section 7 PLC Setup* and *8-3 Terminal Allocations for Built-in Functions*.
3	Create Ladder Program	Write ladder diagrams using the CX-Programmer.	
	Special Instructions	Example: Permitting interrupts with the MSKS instruction and programming high-speed counters with the CTBL instruction.	
	Writing Related Auxiliary Area Words	Example: Stopping high-speed counters.	Refer to *A-2 Auxiliary Area Allocations by Address*.
	Reading Related Auxiliary Area Words	Example: Reading the present value of a high-speed counter.	Refer to *A-2 Auxiliary Area Allocations by Address*.
4	Transfer PLC Setup and Ladder Program	Transfer the PLC Setup and ladder program from the CX-Programmer to the CP2E CPU Unit.	
5	Restart the CP2E	Once turn OFF the power supply to the CP2E CPU Unit, then turn ON again.	
6	Start Operation	Start the CP2E CPU Unit operation.	

8-3 Terminal Allocations for Built-in Functions

8-3-1 Specifying the Functions to Use

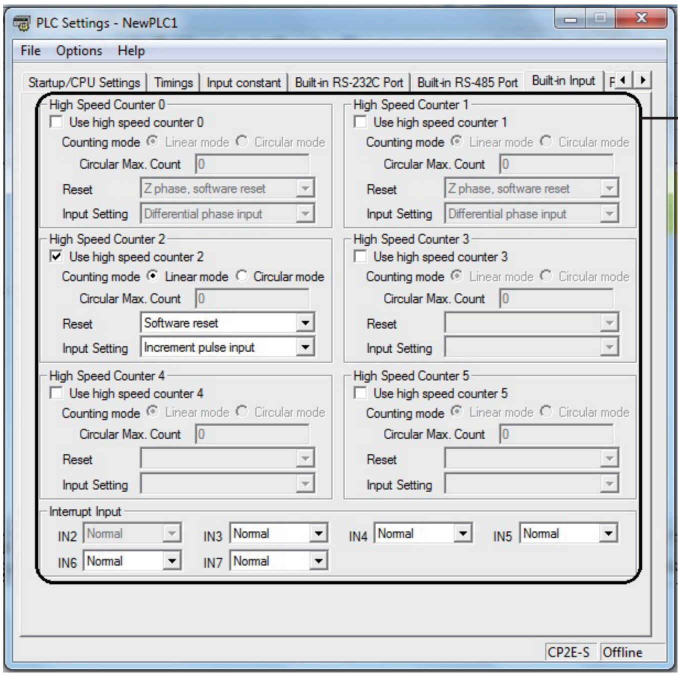
A CP2E CPU Unit uses the same built-in I/O terminals for different functions. Allocate the I/O terminals in advance, making sure that each terminal is used for only one function.

Specify the input functions in the PLC Setup from the CX-Programmer, and specify the output functions in PLC Setup and programming instructions.

8-3-2 Selecting Functions in the PLC Setup

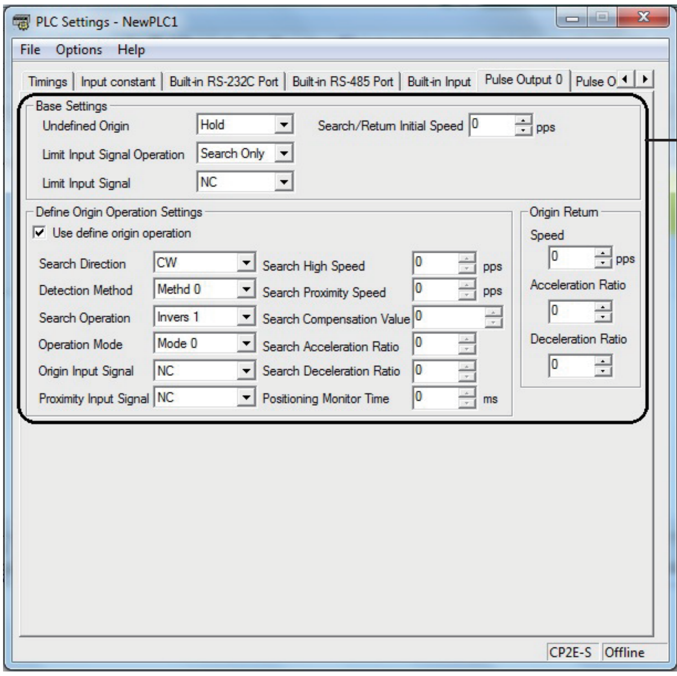
Functions are enabled by setting parameters in the PLC Setup. Set the functions so that no more than one function uses the same terminal. Select function numbers so that high-speed counter inputs and inputs for other functions, such as interrupt inputs, quick-response inputs, and origin inputs do no conflict with each other.

- Input functions can be selected by selecting the ***Use high speed counter*** Check Box in a ***High-speed Counter*** Area on the Built-in Input Tab Page or by setting an input to ***Interrupt*** or ***Quick*** in the ***Interrupt Input*** Area of the same page.



Select the *Use high speed counter* Check Box for a High-speed Counter

• The input and output terminals used by the origin search function can be enabled by selecting the ***Use define origin operation*** Check Box on a Pulse Output Tab Page.



Select the *Use define origin operation* Check Box.

8-3-3 Allocating Built-in Input Terminals

Allocating Functions to Built-in Input Terminals

Input terminals are allocated functions by setting parameters in the PLC Setup. Set the PLC Setup so that each terminal is used for only one function.

- **E20/30/40/60, S30/40/60 or N20/30/40/60 CPU Units**

Terminal block label	Terminal number	PLC Setup						
		Interrupt input settings on Built-in Input Tab Page			High-speed counter 0 to 5 settings on Built-in Input Tab Page			Origin search settings on Pulse Output 0 to 3 Tab Page
		Normal	Interrupt	Quick	Use			Use
		Normal input	Interrupt inputs	Quick-response inputs	Increment pulse input	Differential phase ×4 or up/down	Pulse/ direction	Origin search
CIO 0	00	Normal input 0	–	–	Counter 0, increment input	Counter 0, phase A or up input	Counter 0, pulse input	–
	01	Normal input 1	–	–	Counter 1, increment input	Counter 0, phase B or down input	Counter 1, pulse input	–
	02	Normal input 2	Interrupt input 2	Quick-response input 2	Counter 2, increment input	Counter 1, phase A or up input	Counter 0, direction	–
	03	Normal input 3	Interrupt input 3	Quick-response input 3	–	Counter 1, phase B or down input	Counter 1, direction	–
	04	Normal input 4	Interrupt input 4	Quick-response input 4	Counter 3, increment input	Counter 0, phase Z or reset input	Counter 0, reset input	–
	05	Normal input 5	Interrupt input 5	Quick-response input 5	Counter 4, increment input	Counter 1, phase Z or reset input	Counter 1, reset input	–
	06	Normal input 6	Interrupt input 6	Quick-response input 6	Counter 5, increment input	–	–	Pulse 0, Origin input signal
	07	Normal input 7	Interrupt input 7	Quick-response input 7	–	–	–	Pulse 1, Origin input signal
	08	Normal input 8	Interrupt input 8 (Note 1)	Quick-response input 8 (Note 1)	–	–	–	Pulse 2, Origin input signal (Note 2)
	09	Normal input 9	Interrupt input 9 (Note 1)	Quick-response input 9 (Note 1)	–	–	–	Pulse 3, Origin input signal (Note 2)
	10	Normal input 10	–	–	–	–	–	Pulse 0, Origin proximity input signal
	11	Normal input 11	–	–	–	–	–	Pulse 1, Origin proximity input signal
CIO 1	00	Normal input 12	–	–	–	–	–	Pulse 2, Origin proximity input signal (Note 2)
	01	Normal input 13	–	–	–	–	–	Pulse 3, Origin proximity input signal (Note 2)
	02 to 11	Normal input 14 to 23	–	–	–	–	–	–
CIO 2	00 to 11	Normal input 24 to 35	–	–	–	–	–	–

Note 1 Only supported by N20/30/40/60 CPU Units.

2 Only supported by N30/40/60 CPU Units.

3 The same pulse inputs must be used for high-speed counter 0 and high-speed counter 1.

4 High-speed counter 2 cannot be used if the input setting of high-speed counter 0 or high-speed counter 1 is set for differential phase inputs (4×), pulse + direction inputs, or up/down pulse inputs.

● E14 or N14 CPU Units

Terminal block label	Terminal number	PLC Setup						
		Interrupt input settings on Built-in Input Tab Page			High-speed counter 0 to 5 settings on Built-in Input Tab Page			Origin search settings on Pulse Output 0/1 Tab Page
		Normal	Interrupt	Quick	Use			Use
		Normal input	Interrupt inputs	Quick-response inputs	Increment pulse input	Differential phase ×4 or up/down	Pulse/ direction	Origin search
CIO 0	00	Normal input 0	–	–	Counter 0, increment input	Counter 0, phase A or up input	Counter 0, pulse input	–
	01	Normal input 1	–	–	Counter 1, increment input	Counter 0, phase B or down input	Counter 1, pulse input	–
	02	Normal input 2	Interrupt input 2	Quick-response input 2	Counter 2, increment input	Counter 1, phase A or up input	Counter 0, direction	–
	03	Normal input 3	Interrupt input 3	Quick-response input 3	–	Counter 1, phase B or down input	Counter 1, direction	Pulse 0, Origin proximity input signal
	04	Normal input 4	Interrupt input 4	Quick-response input 4	Counter 3, increment input	Counter 0, Phase Z or reset input	Counter 0, reset input	–
	05	Normal input 5	Interrupt input 5	Quick-response input 5	Counter 4, increment input	Counter 1, Phase Z or reset input	Counter 1, reset input	Pulse 1, Origin proximity input signal
	06	Normal input 6	Interrupt input 6	Quick-response input 6	Counter 5, increment input	–	–	Pulse 0, Origin input signal
	07	Normal input 7	Interrupt input 7	Quick-response input 7	–	–	–	Pulse 1, Origin input signal

Note 1 The same pulse inputs must be used for high-speed counter 0 and high-speed counter 1.

2 High-speed counter 2 cannot be used if the input setting of high-speed counter 0 or high-speed counter 1 is set for differential phase inputs (4×), pulse + direction inputs, or up/down pulse inputs.

Prohibiting Repeated Use of Input Terminal Number

The input terminals 00 to 11 of CIO 0 and 00 to 01 of CIO 1 are used for input interrupts, quick-response inputs, high-speed counters, origin searches and normal inputs. Therefore, do not use the input terminals repeatedly. For example, if quick-response input 2 is used, then input terminal 02 is occupied, so it cannot be used for normal input 2, input interrupt 2, quick-response input 2, counter 2 (increment), counter 1 (phase-A/increment) or counter 0 (direction).
A priority is as follows when used repeatedly.
Origin search settings > High-speed counter settings > Input settings

8-3-4 Allocating Built-in Output Temrinals

Allocating Functions to Built-in Output Terminals

Output terminals are allocated functions by setting parameters in the PLC Setup. Set the PLC Setup so that each terminal is used for only one function.

Output terminal block		Other than those shown at the right	When a pulse output instruction (ITPL, SPED, ACC, PLS2, ORG or IFEED) is executed	PLC Setup Origin search settings on Pulse Output 0 to 3 Tab Page	When the PWM instruction is executed
Terminal block label	Terminal number	Normal outputs	Fixed duty ratio pulse output		Variable-duty-factor output
			Pulse + Direction Mode	Use	PWM output
CIO 100	00	Normal output 0	Pulse output 0, pulse	–	–
	01	Normal output 1	Pulse output 1, pulse	–	PWM output 0
	02	Normal output 2	Pulse output 0, direction	–	–
	03	Normal output 3	Pulse output 1, direction	–	–
	04	Normal output 4	–	Pulse 0, Error counter reset output	–
	05	Normal output 5	–	Pulse 1, Error counter reset output	–
	06	Normal output 6	–	Pulse 2, Error counter reset output (Note)	–
	07	Normal output 7	–	Pulse 3, Error counter reset output (Note)	–
CIO 101	00	Normal output 8	Pulse output 2, pulse (Note)	–	–
	01	Normal output 9	Pulse output 3, pulse (Note)	–	–
	02	Normal output 10	Pulse output 2, direction (Note)	–	–
	03	Normal output 11	Pulse output 3, direction (Note)	–	–
	04 to 07	Normal output 12 to 15	–	–	–
CIO 102	00 to 07	Normal output 16 to 23	–	–	–

Note Only supported by N30/40/60 CPU Units.

Prohibiting Repeated Use of Output Terminal Number

The output terminals 00 to 07 of CIO 100 and 00 to 03 of CIO 101 are used for pulse outputs, PWM outputs and normal outputs. Therefore, do not use the output terminals repeatedly. For example, if pulse output 0 (direction) is used, then output terminal 02 is occupied, so it cannot be used for normal output 2.

9

Quick-response Inputs

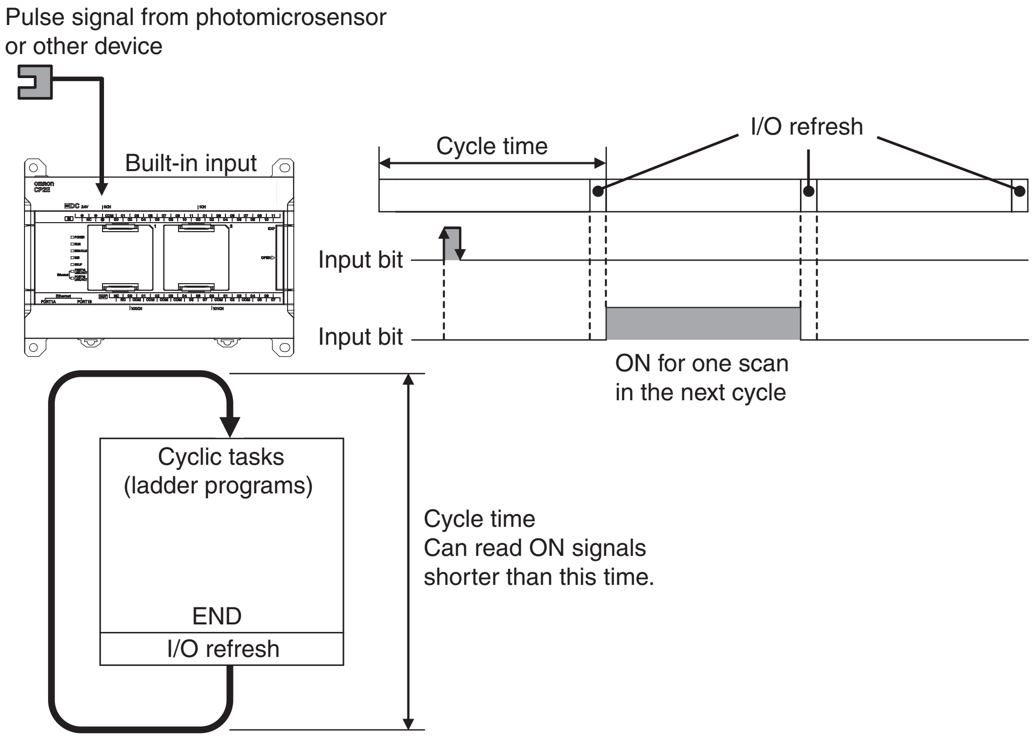
This section describes the quick-response inputs that can be used to read signals that are shorter than the cycle time.

9-1 Quick-response Inputs

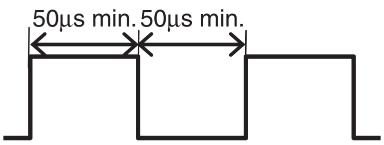
Quick-response inputs can be used with any model of CP2E CPU Unit.

9-1-1 Overview

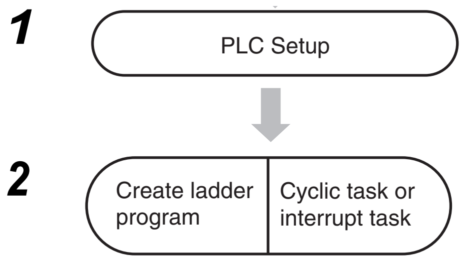
The quick-response inputs can read pulses with an ON time as short as 50 µs even if they are shorter than the cycle time. Use the quick-response inputs to read signals shorter than the cycle time, such as inputs from photomicrosensors.



The pulse widths of quick-response input signals must meet the following conditions.



9-1-2 Flow of Operation



- Set IN2 to IN9 for quick-response inputs on the Built-in Input Tab Page of the PLC Setup using the CX-Programmer.
- The terminals 02 to 09 of CIO 0 can be used for quick-response inputs. Bits CIO 0.02 to CIO 0.09 correspond to terminals 02 to 09.

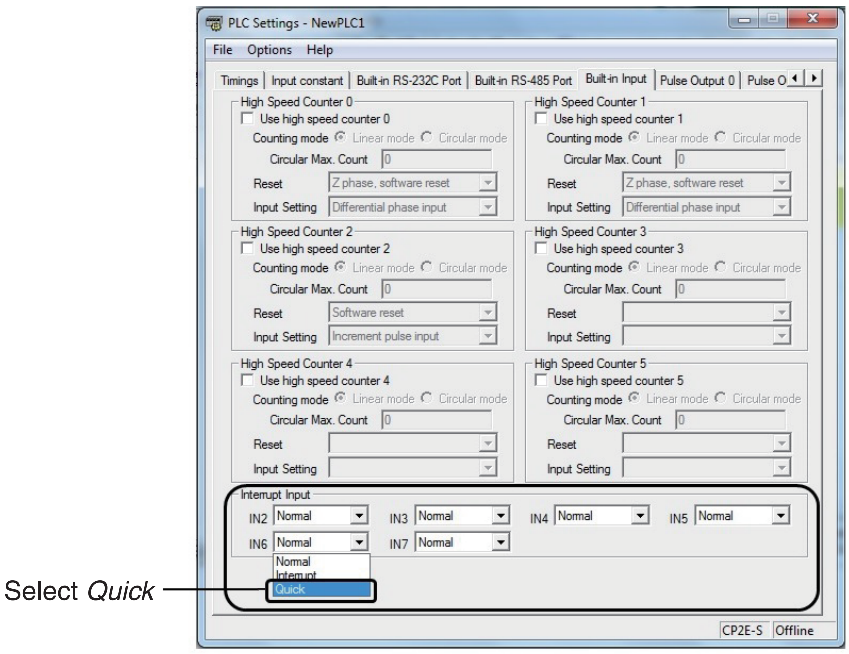
Read the status of CIO 0.02 to CIO 0.09 using the LD instruction or other instructions.

Precautions for Correct Use

A built-in input cannot be used as a quick-response input if it is being used as a normal input, interrupt input, or high-speed counter input. Refer to *8-3-3 Allocating Built-in Input Terminals* for details.

PLC Setup

Click the Built-in Input Tab and select ***Quick*** in the interrupt input settings.



Built-in Input Tab Page

Quick-response input setting		Corresponding bit address
IN2	Select *Quick* for IN2 to IN9.	CIO 0.02
IN3		CIO 0.03
IN4		CIO 0.04
IN5		CIO 0.05
IN6		CIO 0.06
IN7		CIO 0.07
IN8		CIO 0.08
IN9		CIO 0.09

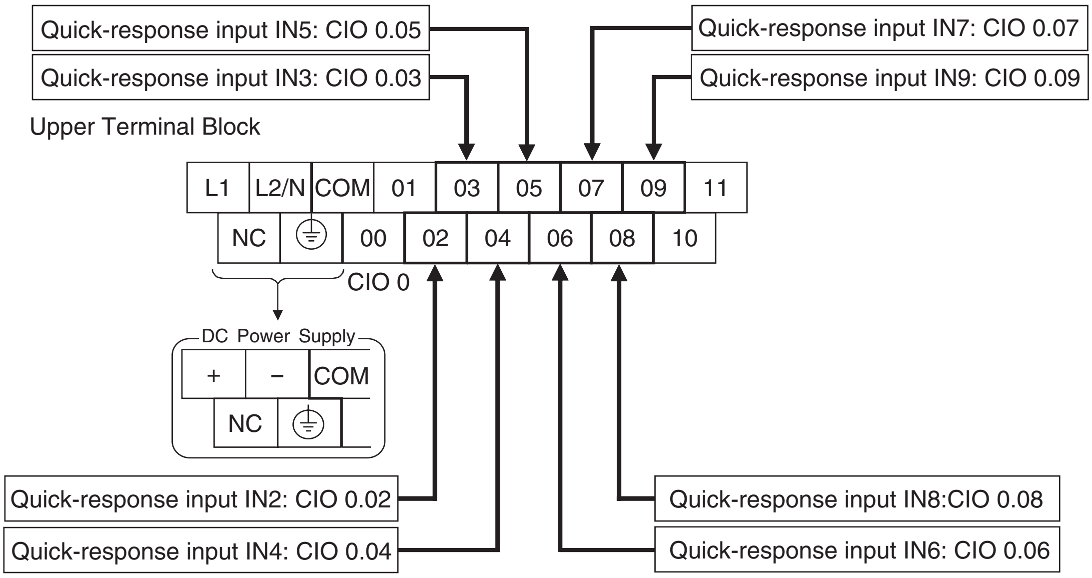
Note 1 The power supply must be restarted after the PLC Setup is transferred in order to validate the quick-response input settings.

2 IN8 and IN9 are only supported by N20/30/40/60 CPU Units.

Quick-response Input Terminal

The following terminals can be used for quick-response inputs.

● Input Terminal Block on CPU Unit with 20 I/O Points

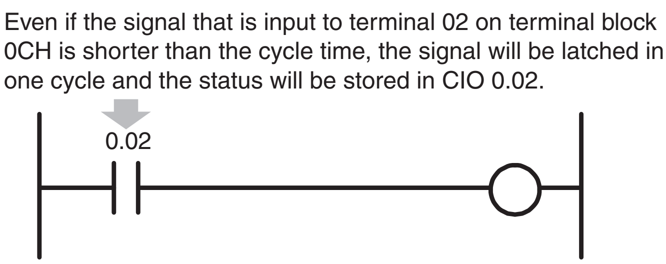


Creating Ladder Programs

Pulse inputs shorter than the cycle time can be read in the CPU Unit I/O memory using normal instructions. Simply set the interrupt setting for the required input to *Quick* in the PLC Setup.

The status of CIO 0.02 to CIO 0.09 can be read using instructions such as the LD instruction.

Example: Setting IN2 to *Quick* in the PLC Setup Interrupt Settings.



- The minimum pulse width (ON time) that can be read for a quick-response input is 50 μs.
- The status of the input that is stored in the I/O memory for a short input will be cleared during the next I/O refresh period.

10

Interrupts

This section describes the interrupts that can be used with CP2E PLCs, including input interrupts and scheduled interrupts.

10-1 Interrupts

10-1-1 Overview

CP2E CPU Units normally repeat processes in the following order: overseeing processes, program execution, I/O refreshing, peripheral servicing. During the program execution stage, cyclic tasks (ladder programs) are executed.

The interrupt function, on the other hand, allows a specified condition to interrupt a cycle and execute a specified program.

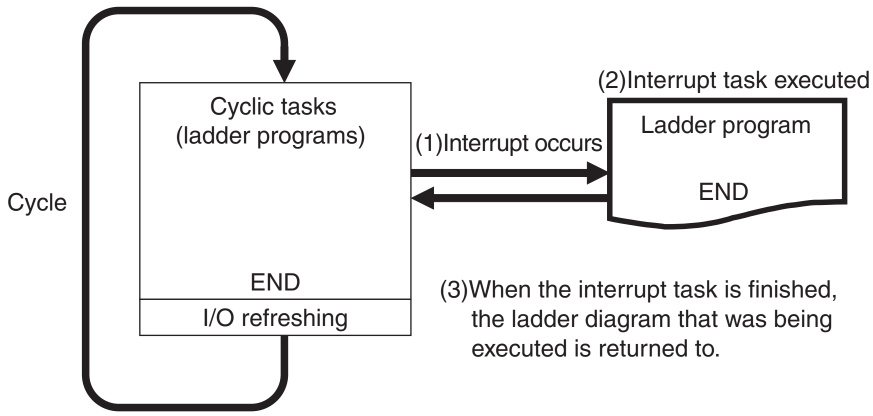
Interrupts can thus be used to perform high-speed processing that is not restricted by the cycle time.

The CP2E performs the following processing when an interrupt occurs.

(1)When an interrupt occurs, execution of the ladder programs in cyclic tasks is interrupted.

(2)The ladder program in the interrupt task is executed.

(3)When the interrupt task is finished, the ladder program that was being executed is returned to.



Interrupt Factors and Types of Interrupts

Interrupts are classified by the interrupt factor. There are the following three types of interrupts.

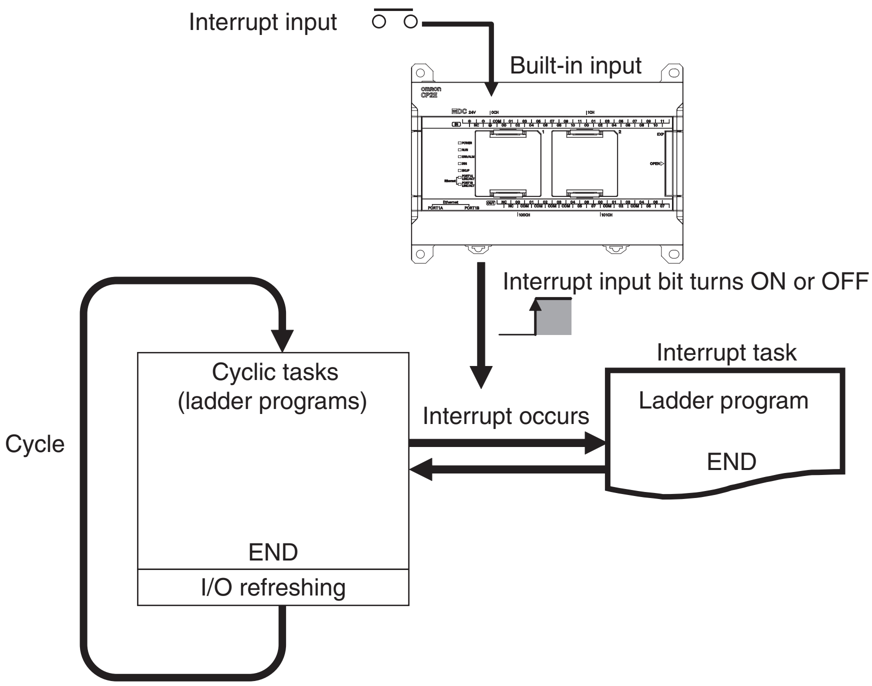
- Changes in status of built-in inputs on the CPU Unit → *Input Interrupts* in Page *10-3*
- Specified intervals measured by internal timers → *Scheduled Interrupts* in Page *10-10*
- PVs of high-speed counter → *High-speed Counter Interrupts* in Page *11-14*

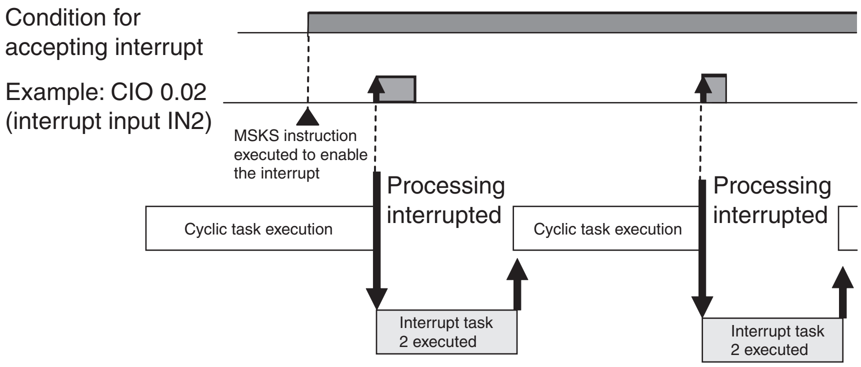
10-2 Input Interrupts

Input interrupts can be used with any model of CP2E CPU Unit.

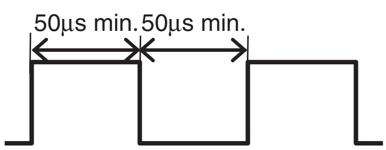
10-2-1 Overview

A corresponding interrupt task can be executed when a built-in input on the CPU Unit turns ON or turns OFF.

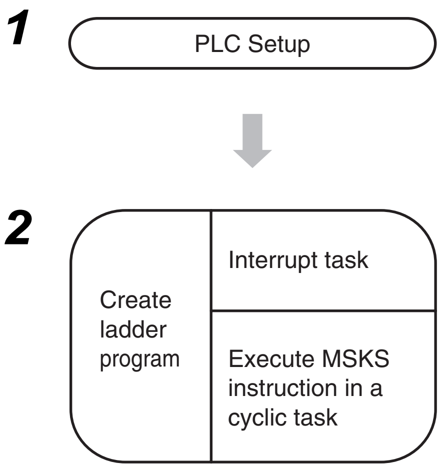




The pulse widths of interrupt input signals must meet the following conditions.



10-2-2 Flow of Operation



- Set IN2 to IN9 for interrupt inputs on the Built-in Input Tab Page of the PLC Setup using the CX-Programmer.
- Terminals 02 to 09 on the CIO 0 terminal block can be used for interrupt inputs. Bits CIO 0.02 to CIO 0.09 correspond to terminals 02 to 09.

Write the program in the interrupt task. Interrupt tasks 2 to 9 correspond to interrupt inputs 2 to 9.

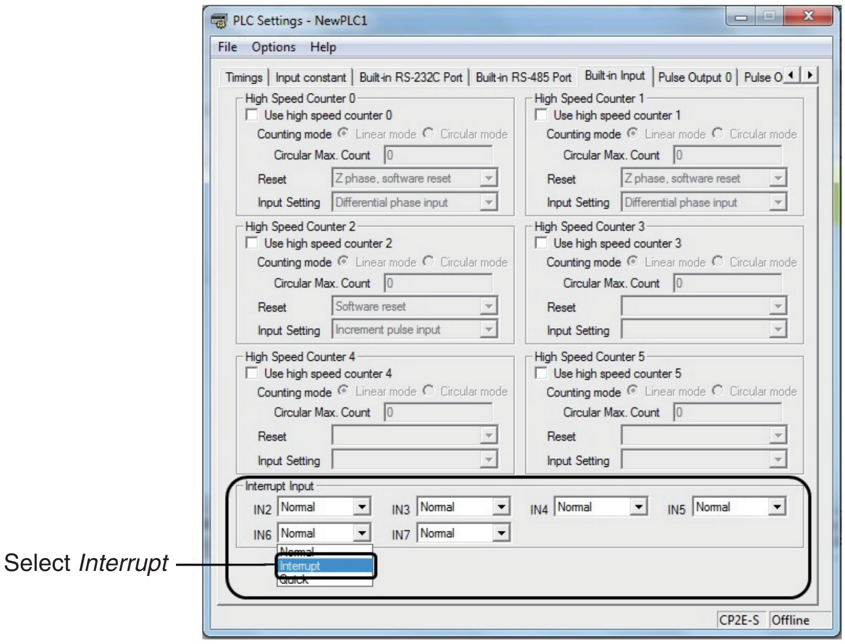
- Specify whether the interrupt is executed when the input turns ON or when it turns OFF in the MSKS instruction. Set N to 112 to 119 in the MSKS instruction.
- Enable input interrupts in the MSKS instruction. Set N to 102 to 109 in the MSKS instruction.

Precautions for Correct Use

A built-in input cannot be used as a normal input, high-speed counter input, or quick-response input if it is being used as an interrupt input. Refer to *8-3-3 Allocating Built-in Input Terminals* for details.

PLC Setup

Click the Built-in Input Tab and select ***Interrupt*** in the interrupt intput settings.



Built-in Input Tab Page

Interrupt input settings		Corresponding bit address	Input interrupt task
IN2	Select *Interrupt* for IN2 to IN9.	CIO 0.02	2
IN3		CIO 0.03	3
IN4		CIO 0.04	4
IN5		CIO 0.05	5
IN6		CIO 0.06	6
IN7		CIO 0.07	7
IN8		CIO 0.08	8
IN9		CIO 0.09	9

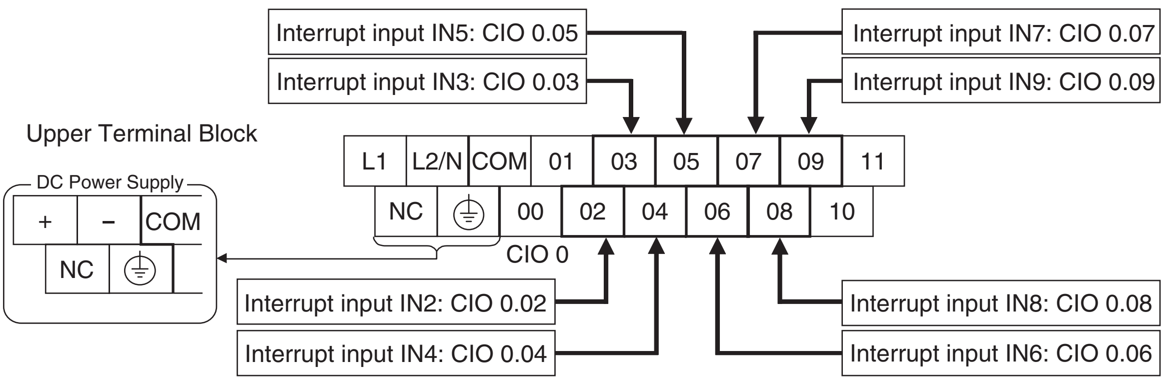
Note 1 The power supply must be restarted after the PLC Setup is transferred in order to enable the interrupt input settings.

2 IN8 and IN9 are only supported by N20/30/40/60 CPU Units.

Assigning Interrupt Input Terminals

The following input terminals can be used for interrupt inputs. These terminals correspond to CIO 0.02 to CIO 0.09 in I/O memory.

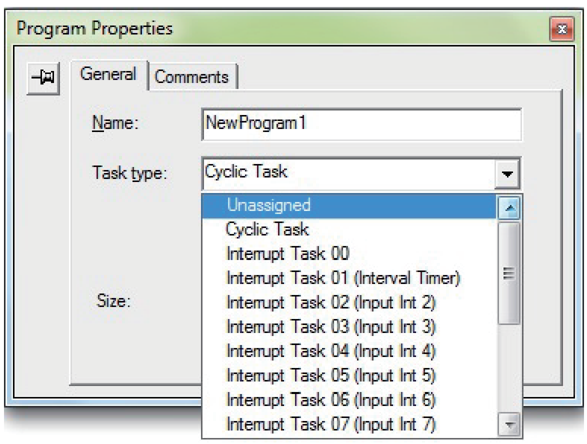
● Input Terminal Block on CPU Unit with 20 I/O Points



Writing the Ladder Program

● Writing the Interrupt Task's Ladder Program

Create ladder programs for interrupt tasks 2 to 9, which are executed for the corresponding interrupt inputs. Right-click a program in the CX-Programmer and select **Properties**. Select interrupt tasks 2 to 9 in the ***Task Type*** Field of the Program Properties Dialog Box.



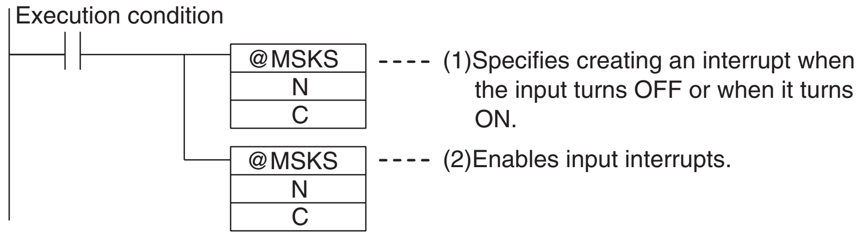
Execute MSKS Instruction in a Cyclic Task

Execute the MSKS instruction from the ladder program in a cyclic task to use input interrupts.

MSKS has the following two functions and two of this instruction are normally used in combination.

(1)Specifying whether to detect ON or OFF signals.

(2)Enabling input interrupts.



The MSKS instruction must be executed only once to make the settings, so in general execute MSKS in just one cycle using the upwardly differentiated variation of the instruction.
The first MSKS instruction can be omitted. If it is omitted, an interrupt will be created when the input turns ON by default.

Specifying MSKS Operands (N and C)

(1)Specifying to Detect ON or OFF Input Signals

Terminal	Corresponding bit address	PLC Setup on Built-in Input Tab Page	Interrupt task number	Operand N	Operand C
				Interrupt identifier	Specifying up/down differentiation of an interrupt input
02 on CIO 0 terminal block	CIO 0.02	Interrupt input IN2	2	112	#0000: Up-differentiation #0001: Down-differentiation
03 on CIO 0 terminal block	CIO 0.03	Interrupt input IN3	3	113	
04 on CIO 0 terminal block	CIO 0.04	Interrupt input IN4	4	114	
05 on CIO 0 terminal block	CIO 0.05	Interrupt input IN5	5	115	
06 on CIO 0 terminal block	CIO 0.06	Interrupt input IN6	6	116	
07 on CIO 0 terminal block	CIO 0.07	Interrupt input IN7	7	117	
08 on CIO 0 terminal block	CIO 0.08	Interrupt input IN8*	8	118	
09 on CIO 0 terminal block	CIO 0.09	Interrupt input IN9*	9	119	

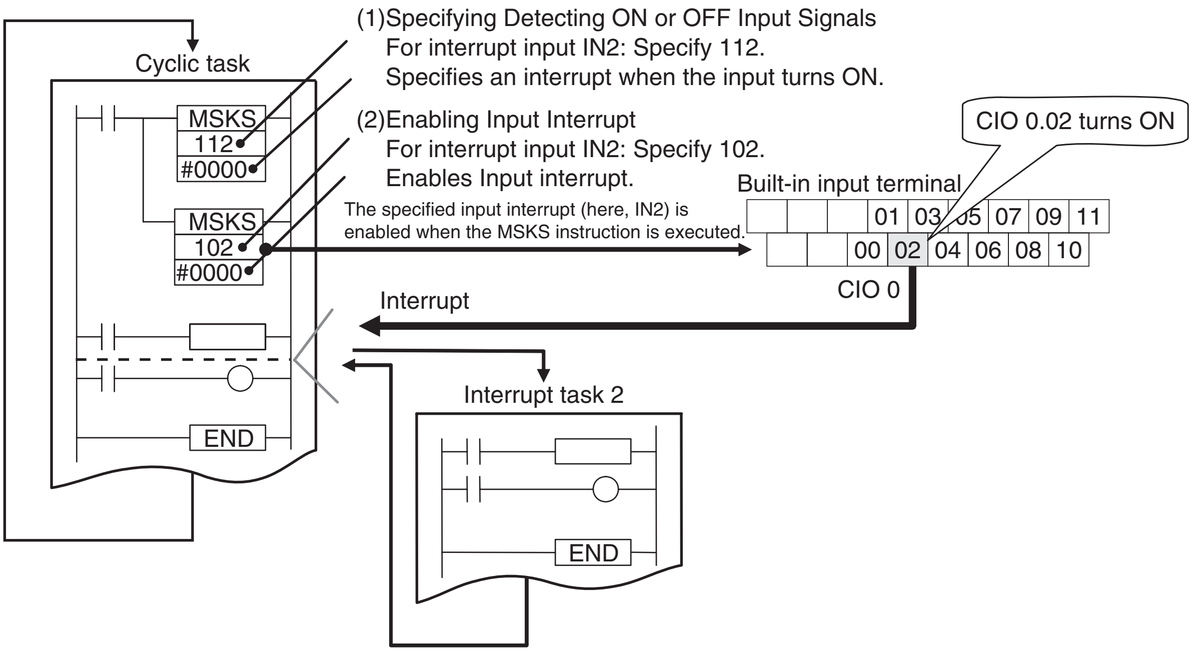
* Interrupt input 8 and 9 are only supported by N20/30/40/60 CPU Units.

(2)Enabling the Input Interrupt

Terminal	Corresponding bit address	PLC Setup on Built-in Input Tab Page	Interrupt task number	Operand N	Operand C
				Interrupt identifier	Enable/Disable
02 on CIO 0 terminal block	CIO 0.02	Interrupt input IN2	2	102	#0000: Enable interrupt #0001: Disable interrupt
03 on CIO 0 terminal block	CIO 0.03	Interrupt input IN3	3	103	
04 on CIO 0 terminal block	CIO 0.04	Interrupt input IN4	4	104	
05 on CIO 0 terminal block	CIO 0.05	Interrupt input IN5	5	105	
06 on CIO 0 terminal block	CIO 0.06	Interrupt input IN6	6	106	
07 on CIO 0 terminal block	CIO 0.07	Interrupt input IN7	7	107	
08 on CIO 0 terminal block	CIO 0.08	Interrupt input IN8*	8	108	
09 on CIO 0 terminal block	CIO 0.09	Interrupt input IN9*	9	109	

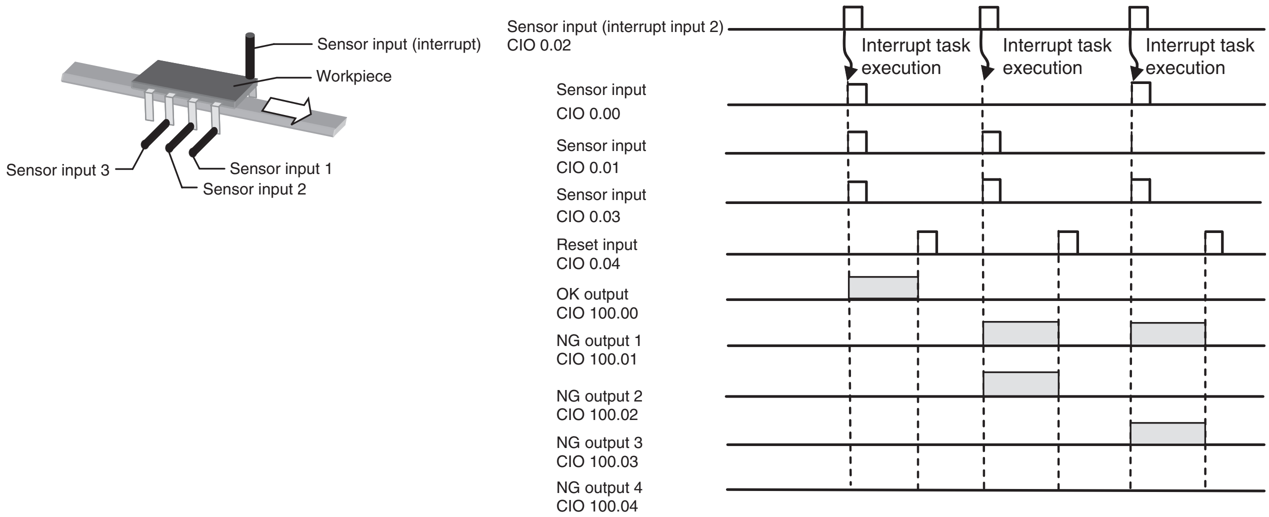
* Interrupt input 8 and 9 are only supported by N20/30/40/60 CPU Units.

- Example



10-2-3 Application Example

In this example, bent parts are detected in a moving workpiece, such as an IC component. When the sensor input (terminal 02 on terminal block 0CH = CIO 0.02) changes from OFF to ON, the interrupt task is executed.



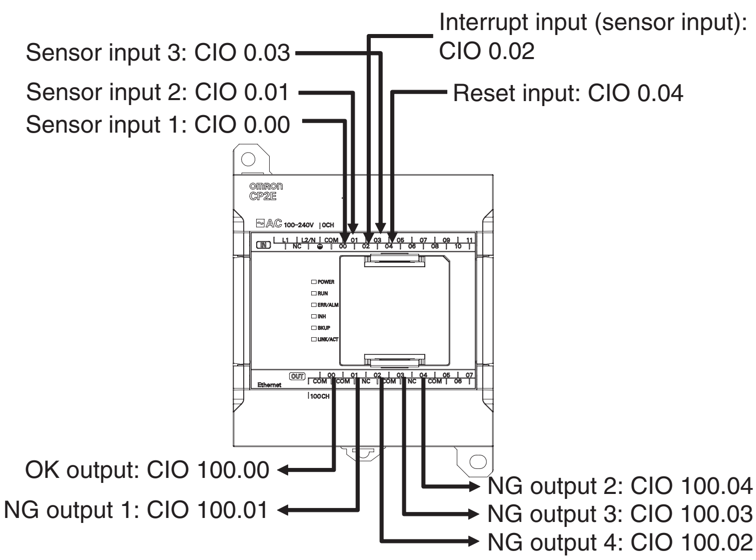
1 PLC Setup

Set IN2 to ***Interrupt*** in the interrupt input settings on the Built-in Input Tab Page.

2 Connecting Interrupt Input Terminals

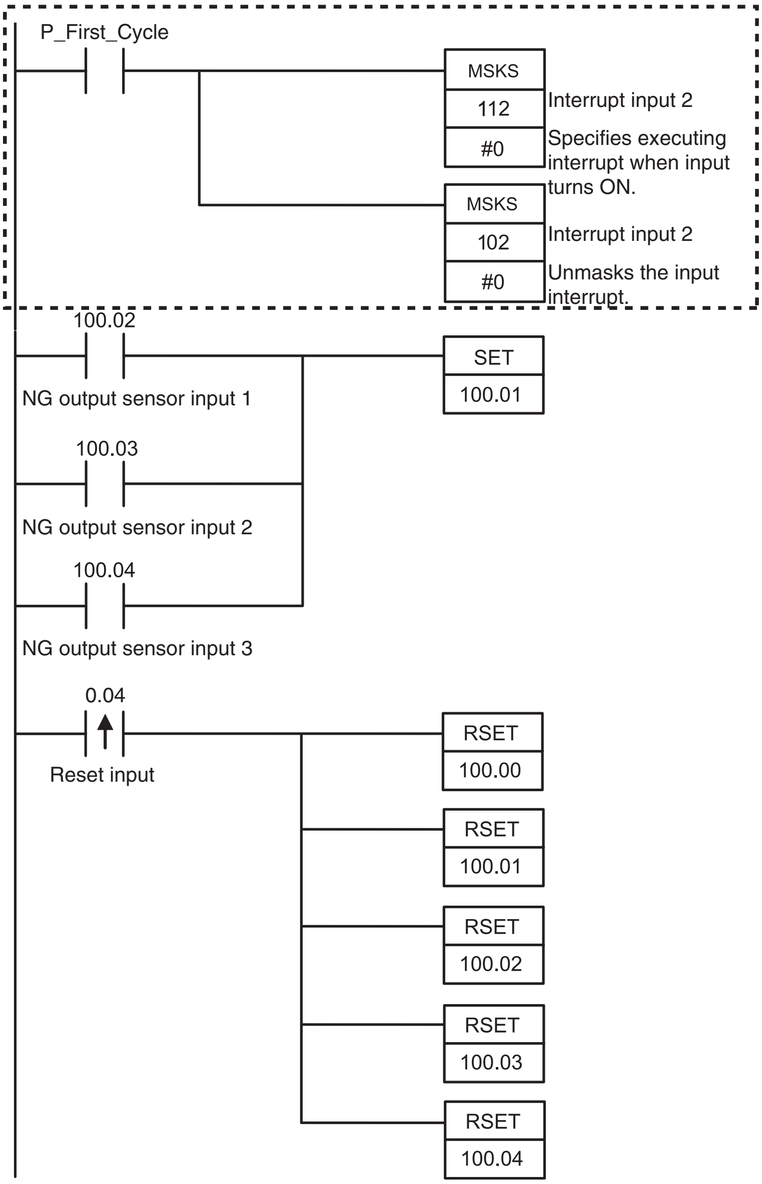
Terminal 2 on terminal block 0CH is interrupt input IN2.

Interrupt task 2 corresponds to interrupt input 2.



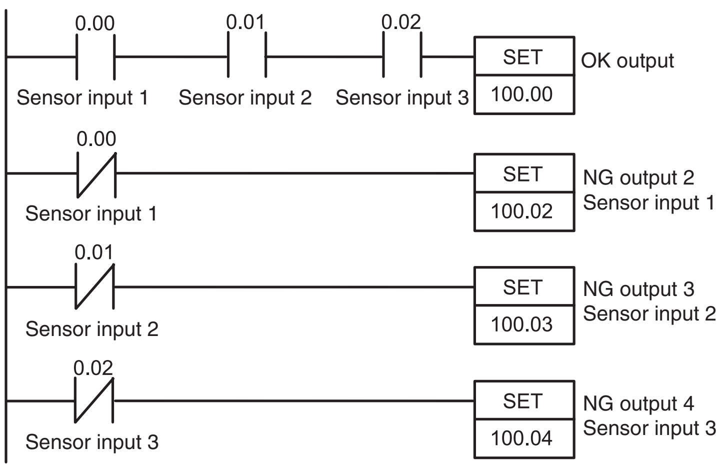
Programming Example

Cyclic Task



The MSKS instruction is used to specify an interrupt when the input turns ON and then it is used to unmask the input interrupt.

Interrupt Task 2

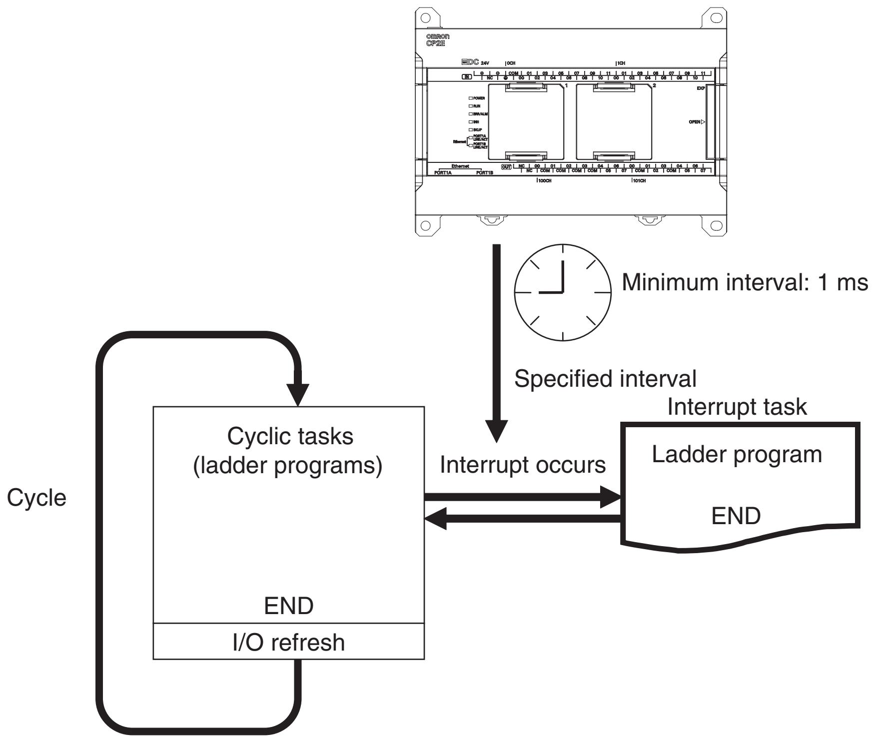


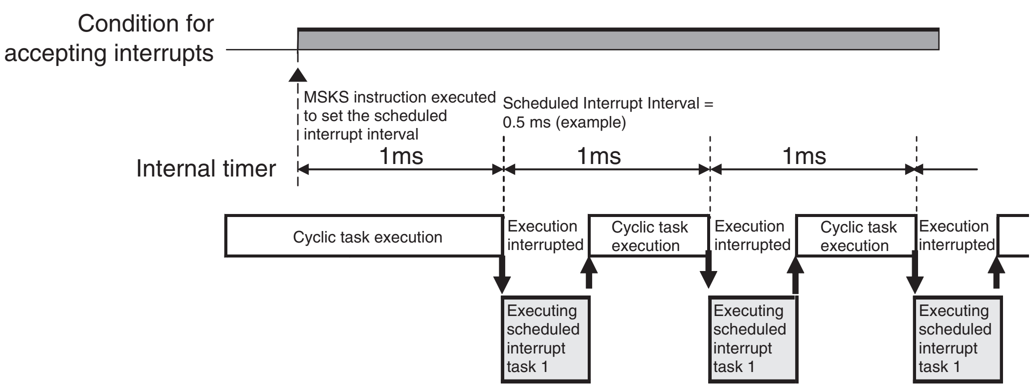
10-3 Scheduled Interrupts

Scheduled interrupts can be used with any model of CP2E CPU Unit.

10-3-1 Overview

Scheduled interrupts can be used to execute interrupt tasks at fixed time intervals measured by the CPU Unit's internal timer.





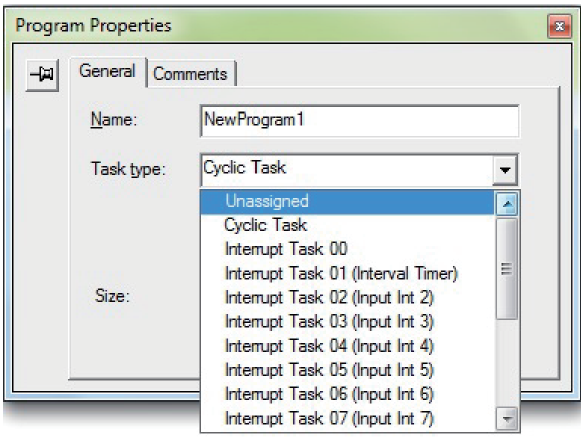
10-3-2 Flow of Operation

Create ladder program	Interrupt task	Write the program for the corresponding interrupt task 1 (fixed).
	Execute MSKS instruction in a cyclic task	Use MSKS to specify the scheduled interrupt interval. The setting can be 1 ms or longer. Set N to 4 or 14 in the MSKS instruction.

Writing the Ladder Program

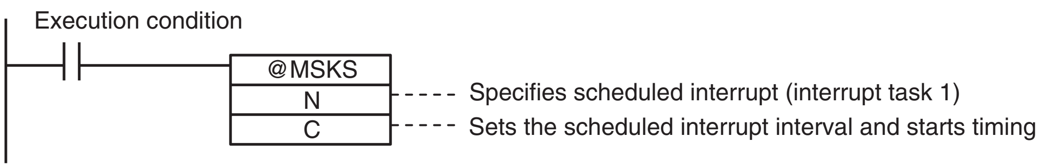
● Writing the Interrupt Task Program

Create the program for interrupt task 1, which is executed for the scheduled interrupt. Right-click a program in the CX-Programmer and select **Properties**. Select Interrupt Tasks 01 (scheduled interrupt) in ***Task Type*** Field of the Program Properties Dialog Box.



● Execute MSKS in a Cyclic Task

The MSKS instruction must be executed from the ladder program in a cyclic task in order to use scheduled interrupts.



The MSKS instruction must be executed only once to make the settings, so in general execute MSKS in just one cycle using the upwardly differentiated variation of the instruction.

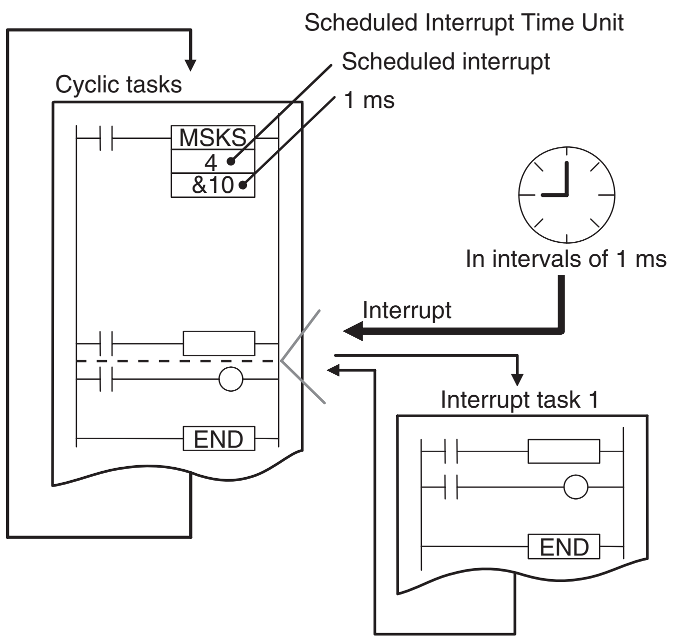
Specifying MSKS Operands (N and C)

MSKS Operands

MSKS Operands	
N	**C**
Interrupt number	**Scheduled interrupt interval**
Scheduled interrupt (interrupt task 1)* 14: Reset and restart 4: Reset and restart	0 decimal: Disable interrupt (stop internal timer) 10 to 9,999 decimal:Enable interrupt (Reset internal timer and then start timer with interrupt interval between 1.0 and 999.9 ms)

* Either is reset.

Example



Precautions for Correct Use

- Set a scheduled interrupt interval is longer than the time required to execute the corresponding interrupt task.
- If you shorten the scheduled interrupt interval and increase the execution frequency of the scheduled interrupt task, the cycle time will increase, and this will affect the execution timing of cyclic tasks.
- If an interrupt task is being executed for another interrupt (input interrupt or high-speed counter interrupt) when the scheduled interrupt occurs, the scheduled interrupt will not be executed until the other interrupt task had been completed.
  Even in this case, measurement of internal timer is continually executed in parallel, so the execution of scheduled interrupt tasks will not be delayed.
- Scheduled interrupt interval cannot be changed during the startup of scheduled interrupt. Change the interval after the scheduled interrupt has stopped.

10-4 Precautions for Using Interrupts

10-4-1 Interrupt Task Priority and Order of Execution

The priority of interrupt tasks is the same order for input interrupts, scheduled interrupts and high-speed counter interrupts. Therefore, if interrupt task A (an input interrupt, for example) is being executed when interrupt task B (a scheduled interrupt, for example) occurs, task A execution will not be interrupted. Task B execution will be started when task A had been completed.

For example, if an interrupt task is being executed for another interrupt (input interrupt or high-speed counter interrupt) when a scheduled interrupt occurs, the scheduled interrupt will not be executed until execution of the other interrupt task had been completed. Even in this case, internal timer is continually measured in parallel, so the execution of the scheduled interrupt task will not be delayed.

10-4-2 Related Auxiliary Area Words and Bits

When the processing time of an interrupt task exceeds 0.1ms, the processing time of the interrupt task and the task number of the interrupt with the maximum processing time can be found in the Auxiliary Area. The actual processing time can also be checked.

Name	Addresses	Description
Maximum Interrupt Task Processing Time	A440	Contains the maximum interrupt task processing time in units of 0.1 ms. This value is cleared at the start of operation.
Interrupt Task With Maximum Processing Time	A441	Contains the task number of the interrupt task with the maximum processing time. Here, #8000 to #800F correspond to tasks 0 to 15 (00 to 0F hex). A441.15 will turn ON when the first interrupt occurs after the start of operation. The maximum processing time for subsequent interrupt tasks will be stored in the rightmost digit in hexadecimal. This value is cleared at the start of operation.
Total of Interrupt Task Processing Time in One Cycle	A442	Contains the total of interrupt task processing time in one cycle in units of 0.1ms. Sets when the value is bigger than the last one once a cycle by common processing.This value is cleared at the start of operation.

10-4-3 Duplicate Processing in each Task

Observe the following precautions, if a word address in I/O memory is manipulated by instructions both in a cyclic task and an interrupt task.

- If the interrupt task overwrites an I/O memory address used by one of the interrupted instruction's operands, the data may be overwritten when the saved data is restored when processing returns to the cyclic task.
- To prevent certain instructions from being interrupted during processing, insert the DI or EI instruction just before and after the instructions, using the DI or EI instruction before the instructions to disable interrupts and the DI or EI instruction after the instructions to enable interrupts again.

Additional Information

Normally, if an interrupt occurs, execution of the cyclic task will be interrupted immediately, even during execution of an instruction in the cyclic task, and the partially processed data is saved. After the interrupt task had been completed, the cyclic task restarts with the data saved before the interrupt processing.

11

High-speed Counters

This section describes the high-speed counter inputs, high-speed counter interrupts, and the frequency measurement function.

11-1 Overview

High-speed counters can be used with any model of CP2E CPU Unit.

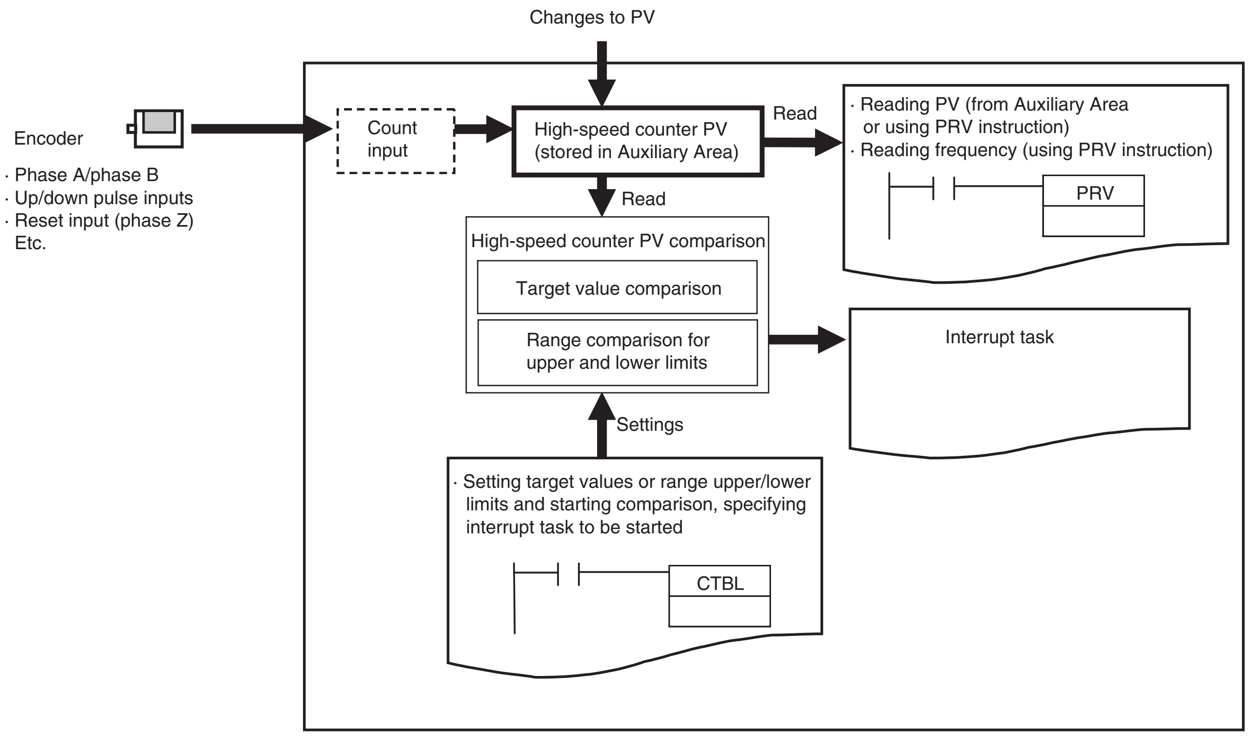
11-1-1 Overview

High-speed counters are used to measure high-speed pulse input signals that cannot be measured by counter (CNT) instructions.

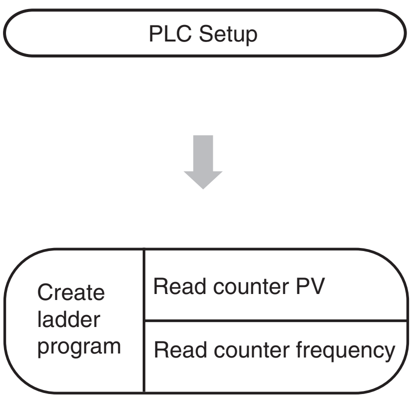
● Applications

- Detecting the position or length of a workpiece with an input from an incremental rotary encoder.
- Measuring the speed of a workpiece from its position data using frequency measurement and rotational speed conversion.
- High-speed processing according to the workpiece's position data.

The present value of the high-speed counter is stored in the Auxiliary Area and can be used as position data. When it reaches preset values, interrupts can be generated. The count can be started and stopped. Depending on the instruction, the frequency (speed) can be read from the present value of the high-speed counter.



11-1-2 Flow of Operation



1

- Enable the required high-speed counters.
- Select the *Use high speed counter* Check Box for high-speed counters 0 to 5. Set the input setting, counting mode and reset method on the Built-in Input Tab Page of the PLC Setup using the CX-Programmer.
- Terminals 00 to 06 on the CIO 0 terminal block can be used for high-speed counters. High-speed counters 0 to 5 correspond to terminals 00 to 06.

2

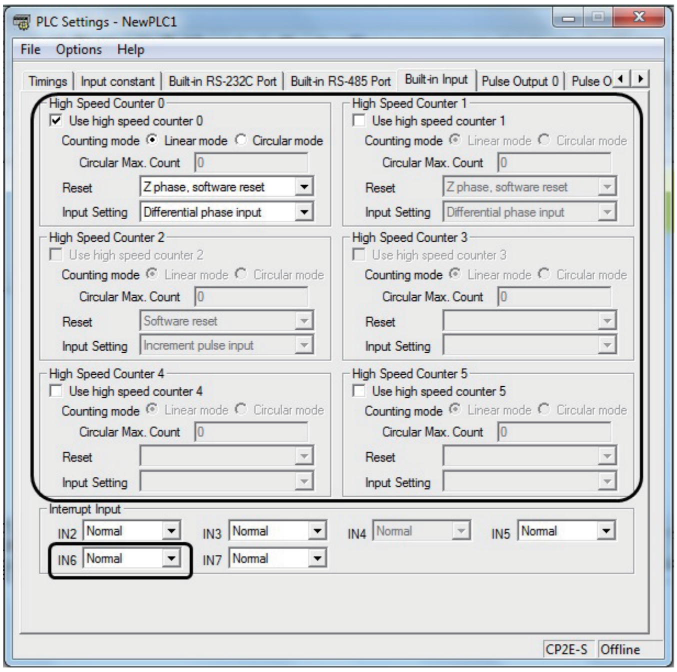
- Read the PV from Auxiliary Area or by executing a PRV instruction.
- Execute a PRV instruction.

Precautions for Correct Use

A built-in input cannot be used as a normal input, interrupt input, or quick-response input if it is being used as a high-speed counter input. Refer to *8-3-3 Allocating Built-in Input Terminals* for details.

PLC Setup

Click the Built-in Input Tab and select the ***Use high speed counter*** Check Box for high-speed counters 0 to 5. Set the counting mode, reset method, and input setting.



Built-in Input Tab Page

Item		Setting
Use high speed counter 0 to 5	Use high-speed counter	Select *Use high speed counter* for each counter to be used.
	Counting Mode	Select *Linear mode or Circular mode.*
	Circular Max. Count (maximum ring count)	If circular mode is selected, set the maximum ring count. 0 to 4,294,967,295 decimal
	Reset	• Phase Z and software reset • Software reset* • Phase Z and software reset (continue comparing) • Software reset (continue comparing)*
	Input Setting	• Differential phase inputs (4×) • Pulse + direction inputs • Up/down pulse inputs • Increment pulse input

* Only a software reset can be used if an increment pulse input is specified.

Note The power supply must be restarted after the PLC Setup is transferred in order to enable the high-speed counter settings.

Determining High-speed Counter

● Pulse Input Method and High-speed Counter Input Terminals

The following input terminals can be used for high-speed counters with the pulse input method.

E20/30/40/60, S30/40/60 or N20/30/40/60 CPU Units

Input terminal block		Pulse input method (Counting mode)			Other functions that cannot be used at the same time			
Terminal block label	Terminal	Increment pulse input	Differential phase ×4 or up/down input	Pulse/ direction input	Normal input	Interrupt input	Quick-response input	Origin searches for pulse outputs 0 to 3
CIO 0	00	High-speed Counter 0, increment input	High-speed Counter 0, phase A or up input	High-speed Counter 0, pulse input	Normal input 0	–	–	–
	01	High-speed Counter 1, increment input	High-speed Counter 0, phase B or down input	High-speed Counter 1, pulse input	Normal input 1	–	–	–
	02	High-speed Counter 2, increment input	High-speed Counter 1, phase A or up input	High-speed Counter 0, direction	Normal input 2	Interrupt input 2	Quick-response input 2	–
	03	–	High-speed Counter 1, phase B or down input	High-speed Counter 1, direction	Normal input 3	Interrupt input 3	Quick-response input 3	–
	04	High-speed Counter 3, increment input	High-speed Counter 0, phase Z or reset input	High-speed Counter 0, reset input	Normal input 4	Interrupt input 4	Quick-response input 4	–
	05	High-speed Counter 4, increment input	High-speed Counter 1, phase Z or reset input	High-speed Counter 1, reset input	Normal input 5	Interrupt input 5	Quick-response input 5	–
	06	High-speed Counter 5, increment input	–	–	Normal input 6	Interrupt input 6	Quick-response input 6	Pulse 0, Origin input signal
	07	–	–	–	Normal input 7	Interrupt input 7	Quick-response input 7	Pulse 1, Origin input signal
	08	–	–	–	Normal input 8	Interrupt input 8 (Note 1)	Quick-response input 8 (Note 1)	Pulse 2, Origin input signal (Note 2)

Input terminal block		Pulse input method (Counting mode)			Other functions that cannot be used at the same time			
Terminal block label	Terminal	Increment pulse input	Differential phase ×4 or up/down input	Pulse/ direction input	Normal input	Interrupt input	Quick-response input	Origin searches for pulse outputs 0 to 3
CIO 0	09	–	–	–	Normal input 9	Interrupt input 9 (Note 1)	Quick-response input 9 (Note 1)	Pulse 3, Origin input signal (Note 2)
	10	–	–	–	Normal input 10	–	–	Pulse 0, Origin proximity input signal
	11	–	–	–	Normal input 11	–	–	Pulse 1, Origin proximity input signal
CIO 1	00	–	–	–	Normal input 12	–	–	Pulse 2, Origin proximity input signal (Note 2)
	01	–	–	–	Normal input 13	–	–	Pulse 3, Origin proximity input signal (Note 2)

Note 1 Only supported by N20/30/40/60 CPU Units.

2 Only supported by N30/40/60 CPU Units.

3 The same pulse input must be used for high-speed counter 0 and high-speed counter 1.

4 High-speed counter 2 cannot be used if the input setting of high-speed counter 0 or high-speed counter 1 is set for differential phase inputs (4x), pulse + direction inputs, or up/down pulse inputs.

5 CIO 0.07 to CIO 0.11 and CIO 1.00 to CIO 1.01 do not overlap with the high-speed counter function, so they can also be used for high-speed counter.

E14 or N14 CPU Units

Input terminal block		Pulse input method (Counting mode)			Other functions that cannot be used at the same time			
Terminal block label	Terminal	Increment pulse input	Differential phase ×4 or up/down input	Pulse/ direction input	Normal input	Interrupt input	Quick-response input	Origin searches for pulse outputs 0 and 1
CIO 0	00	High-speed Counter 0, increment input	High-speed Counter 0, phase A or up input	High-speed Counter 0, pulse input	Normal input 0	–	–	–
	01	High-speed Counter 1, increment input	High-speed Counter 0, phase B or down input	High-speed Counter 1, pulse input	Normal input 1	–	–	–
	02	High-speed Counter 2, increment input	High-speed Counter 1, phase A or up input	High-speed Counter 0, direction	Normal input 2	Interrupt input 2	Quick-response input 2	–
	03	–	High-speed Counter 1, phase B or down input	High-speed Counter 1, direction	Normal input 3	Interrupt input 3	Quick-response input 3	Pulse 0, Origin proximity input signal
	04	High-speed Counter 3, increment input	High-speed Counter 0, phase Z or reset input	High-speed Counter 0, reset input	Normal input 4	Interrupt input 4	Quick-response input 4	–
	05	High-speed Counter 4, increment input	High-speed Counter 1, phase Z or reset input	High-speed Counter 1, reset input	Normal input 5	Interrupt input 5	Quick-response input 5	Pulse 1, Origin proximity input signal
	06	High-speed Counter 5, increment input	–	–	Normal input 6	Interrupt input 6	Quick-response input 6	Pulse 0, Origin input signal
	07	–	–	–	Normal input 7	Interrupt input 7	Quick-response input 7	Pulse 1, Origin input signal

Note 1 The same pulse input must be used for high-speed counter 0 and high-speed counter 1.

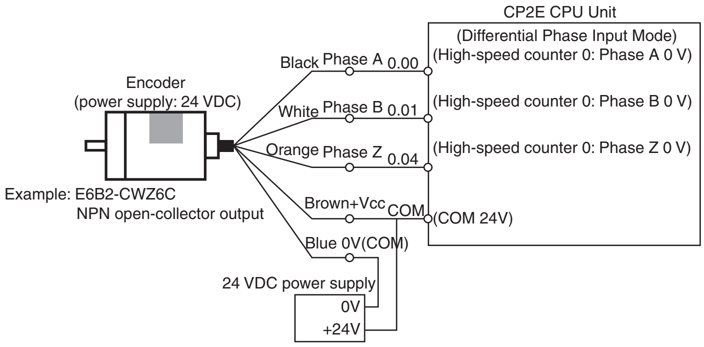
2 High-speed counter 2 cannot be used if the input setting of high-speed counter 0 or high-speed counter 1 is set for differential phase inputs (4x), pulse + direction inputs, or up/down pulse inputs.

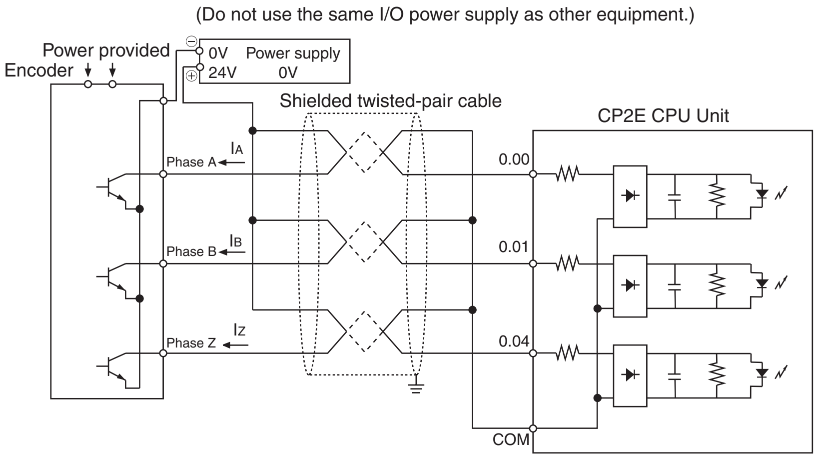
3 CIO 0.07 does not overlap with the high-speed counter function, so it can also be used for high-speed counter.

- **Wiring Example for High-speed Counter Input Terminals**

Using a 24-VDC Open-collector Encoder

The following example shows the connections of an encoder with phase-A, phase-B, and phase-Z inputs to high-speed counter 0.





Writing the Ladder Program

Execution	Program	Reference
Generating interrupts for the high-speed counter PV (number of pulses) and perform high-speed processing.	Specify interrupt tasks with CTBL instructions.	*11-3 High-speed Counter Interrupts*
Reading the high-speed counter PV (number of pulses).	Read the high-speed counter PV from the Auxiliary Area and convert it to position or length data using instructions or measure the length using conmparison instructions such as =, >, and <.	*11-2-4 Reading the Present Value*
Reading the high-speed counter frequency (speed).	Execute a PRV instruction.	*11-2-5 Frequency Measurement*

11-1-3 Specifications

Item		Description			
Pulse input method (Counting mode)		Increment pulse inputs	Differential phase inputs (×4)	Up/down pulse inputs	Pulse + direction inputs
Input signal		Increment	Phase-A	Up pulse	Pulse
		–	Phase-B	Down pulse	Direction
		–	Phase-Z	Reset	Reset
Frequency and number of high-speed counters	E/S☐☐-type CPU Unit	100 kHz: 2 counters 10 kHz: 4 counters	50 kHz: 1 counter 5 kHz: 1 counter	100 kHz: 1 counter 10 kHz: 1 counter	100 kHz: 2 counters
	N14/20 CPU Unit	100 kHz: 2 counters 10 kHz: 4 counters	50 kHz: 1 counter 5 kHz: 1 counter	100 kHz: 1 counter 10 kHz: 1 counter	100 kHz: 2 counters
	N30/40/60 CPU Unit	100 kHz: 3 counters 10 kHz: 3 counters	50 kHz: 2 counters	100 kHz: 2 counters	100 kHz: 2 counters
Counting mode		Linear mode or circular (ring) mode			
Count values		Linear mode: 8000 0000 to 7FFF FFFF hex Ring Mode: 0000 0000 to Ring SV			
High-speed counter PV storage locations		High-speed counter 0: A271 (upper 4 digits) and A270 (lower 4 digits) High-speed counter 1: A273 (upper 4 digits) and A272 (lower 4 digits) High-speed counter 2: A317 (upper 4 digits) and A316 (lower 4 digits) High-speed counter 3: A319 (upper 4 digits) and A318 (lower 4 digits) High-speed counter 4: A323 (upper 4 digits) and A322 (lower 4 digits) High-speed counter 5: A325 (upper 4 digits) and A324 (lower 4 digits) The PVs are refreshed in the overseeing processes at the start of each cycle. Use PRV to read the most recent PVs.			
		Data format: 8 digit hexadecimal • Range in linear mode: 8000 0000 to 7FFF FFFF hex • Range in Ring Mode: 0000 0000 to Ring SV (Circular Max. Count)			
Control method	Target value comparison	Up to 6 target values and corresponding interrupt task numbers can be registered.			
	Range comparison	Up to 6 ranges can be registered, with a separate upper limit, lower limit, and interrupt task number for each range.			
Counter reset method		• Phase-Z + Software reset The high-speed counter is reset when the phase-Z signal goes ON while the Reset Bit (A531.00 to A531.05) is ON. (Phase Z cannot be used for the increment pulse.) • Software reset The high-speed counter is reset when the Reset Bit (A531.00 to A531.05) is turned ON. Operation can be set to stop or continue the comparison operation when the high-speed counter is reset.			

11-2 High-speed Counter Inputs

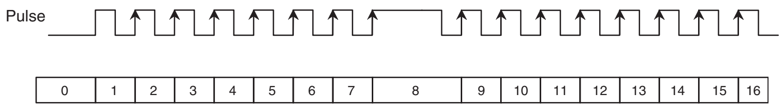
11-2-1 Pulse Input Methods Settings

There are four pulse input methods for high-speed counters.

- Increment pulse input
- Differential phase input (4×)
- Up/Down pulse input
- Pulse+direction input

Increment Pulse Input

The Increment Pulse Input counts signals on a single-phase pulse input. Only incrementing the count is possible in this mode.



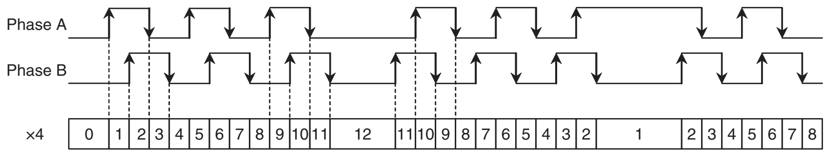
Conditions for Incrementing the Count

Pulse	Count value
OFF→ON	Increment
ON	No change
ON→OFF	No change
OFF	No change

· Only rising edges are counted.

Differential Phase Input (4×)

The Differential Phase Input uses two phase signals (phase A and phase B) and increments/decrements the count according to the status of Differential Phase (4×).

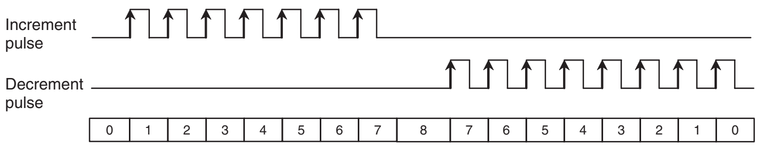


Conditions for Incrementing/Decrementing the Count

Phase A	Phase B	Count value
OFF→ON	OFF	Increment
ON	OFF→ON	Increment
ON→OFF	ON	Increment
OFF	ON→OFF	Increment
OFF	OFF→ON	Decrement
OFF→ON	ON	Decrement
ON	ON→OFF	Decrement
ON→OFF	OFF	Decrement

Up/Down Pulse Input

The Up/Down Pulse Input uses two signals, an increment pulse and a decrement pulse.



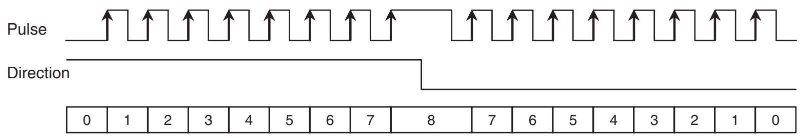
Conditions for Incrementing/Decrementing the Count

Decrement pulse	Increment pulse	Count value
OFF→ON	OFF	Decrement
ON	OFF→ON	No change
ON→OFF	ON	No change
OFF	ON→OFF	No change
OFF	OFF→ON	Increment
OFF→ON	ON	No change
ON	ON→OFF	No change
ON→OFF	OFF	No change

· The count is incremented for each increment pulse and decremented for each decrement pulse.
· Only rising edges are counted.

Pulse + Direction Input

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



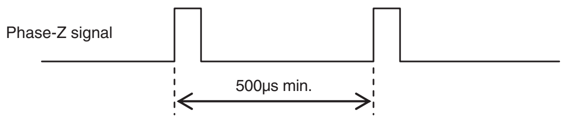
Conditions for Incrementing/
Decrementing the Count

Direction	Pulse	Count value
OFF→ON	OFF	No change
ON	OFF→ON	Increment
ON→OFF	ON	No change
OFF	ON→OFF	No change
OFF	OFF→ON	Decrement
OFF→ON	ON	No change
ON	ON→OFF	No change
ON→OFF	OFF	No change

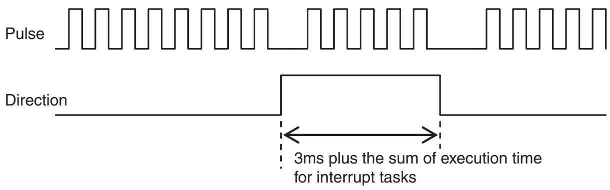
· The count is incremented when the direction signal is ON and decremented when it is OFF.
· Only rising edges are counted.

Precautions for Correct Use

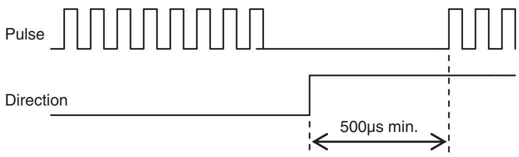
- Interval of Phase-Z input signal
  Do not input the phase-Z signal at a high frequency. The interval of phase-Z input signal must be longer than 500μs.
  If the phase-Z signal is input at a high frequency, cycle time exceeded error may occur.



- Interval of direction changing
  If the input setting is set for pulse+direction inputs, do not change the direction at a high frequency. The interval of direction signal ON or OFF must be longer than 3ms plus the sum of execution time for interrupt tasks that may possibly happen at the same time.
  The sum of execution time for interrupt tasks in one cycle is stored in A442.
  If the direction signal is ON or OFF at a high frequency, count values may not agree.
  Cycle time exceeded error possibly occurs when changing the direction at a high frequency. Therefore, please do not connect chattering equipment as direction signal input such as relay and switch.



- Interval of pulse input after direction changing
  If the input setting is set for pulse+direction inputs, the Interval of pulse input after the direction changing must be longer than 500μs.
  If the pulse is output immediately after the direction changing, count values may not agree.



Additional Information

The count of a 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.

High-speed counter	Address of High-speed Counter Count Direction Flag
High-speed counter 0	A274.10
High-speed counter 1	A275.10
High-speed counter 2	A320.10
High-speed counter 3	A321.10
High-speed counter 4	A326.10
High-speed counter 5	A327.10

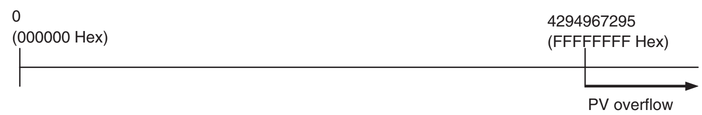
11-2-2 Counting Ranges Settings

The following counting modes can be selected for high-speed counters: Linear Mode that counts in a fixed range and Circular (Ring) Mode that counts in a set range of any maximum value.

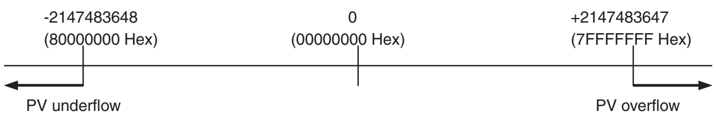
Linear Mode

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.

- Increment Mode



- Up/Down Mode

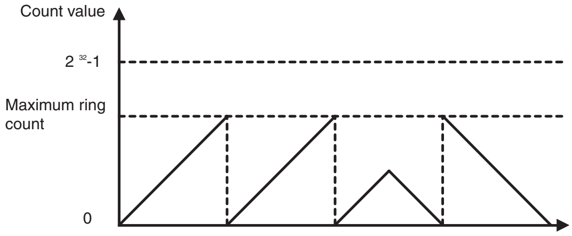


Circular (Ring) Mode

Input pulses are counted in a loop within the set range.

- If the count is incremented from the maximum 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 maximum ring count automatically and decrementing will continue.

Consequently, underflows and overflows cannot occur when Ring Mode is used.



● Maximum Ring Count

Use the PLC Setup to set the maximum ring count (Circular Max. Count), which is the maximum value of the input pulse counting range. The maximum ring count can be set to any value between 0000 0001 and FFFF FFFF hex (1 to 4,294,967,295 decimal).

Precautions for Correct Use

- There are no negative values in Ring Mode.
- If the maximum ring count is set to 0 in the PLC Setup, the counter will operate with a maximum ring count of FFFF FFFF hex.

11-2-3 Reset Methods

It is called reset that a high-speed counter's PV is set to 0.

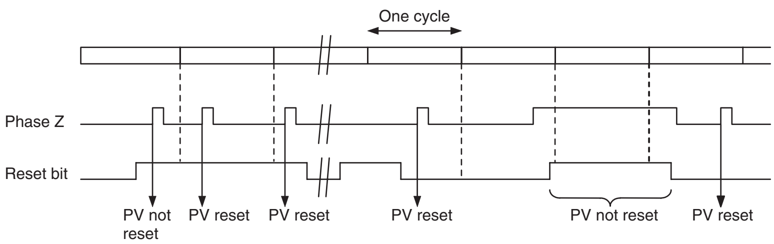
There are two reset methods

- Phase-Z signal + software reset
- software reset

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 (A531.00 to A531.05) 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.

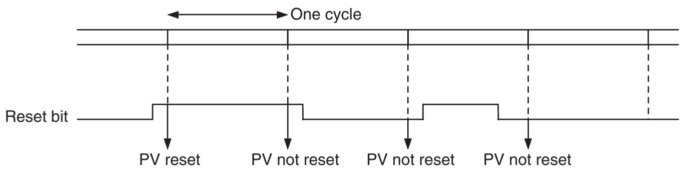


Note The phase-Z signal cannot be used if an incremental counter is specified. Only a software reset can be used.

Software Reset

The high-speed counter's PV is reset when the corresponding High-speed Counter Reset Bit (A531.00 to A531.05) 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-to-ON transition will not be recognized if the Reset Bit goes OFF again within the same cycle.



Additional Information

The comparison operation can be selected to stop or continue in the PLC Setup 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.

11-2-4 Reading the Present Value

The present value of a high-speed counter can be read in the following two ways.

- Value refreshed at the I/O refresh timing → Read PV from Auxiliary Area.
- Value updated when a ladder program is executed → Read PV by executing a PRV instruction.

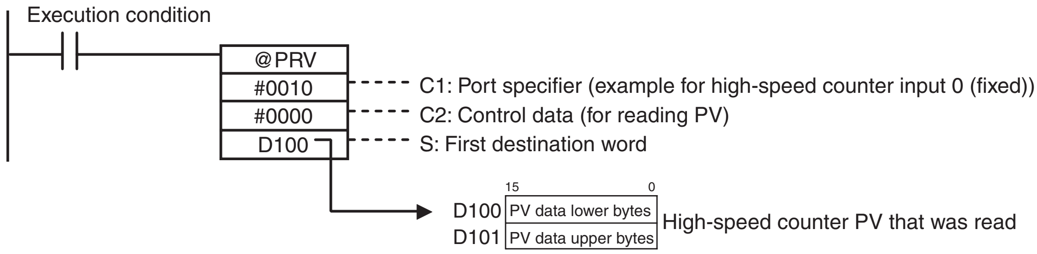
Reading the Value Refreshed at the I/O Refresh Timing

The PV that is stored in the following words can be read using the MOVL instruction or other instructions.

Read PV	Auxiliary Area word
High-speed counter 0	A271 (upper digits) and A270 (lower digits)
High-speed counter 1	A273 (upper digits) and A272 (lower digits)
High-speed counter 2	A317 (upper digits) and A316 (lower digits)
High-speed counter 3	A319 (upper digits) and A318 (lower digits)
High-speed counter 4	A323 (upper digits) and A322 (lower digits)
High-speed counter 5	A325 (upper digits) and A324 (lower digits)

Reading the Value When a Ladder Program is Executed

● Reading the High-speed Counter PV with a PRV Instruction



11-2-5 Frequency Measurement

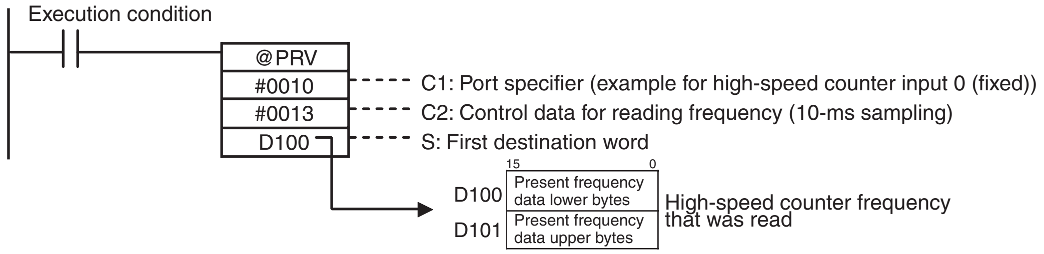
Overview

This function measures the frequency of the high-speed counter (input pulses.)

The input pulse frequency can be read by executing the PRV 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.

● Reading the High-speed Counter Frequency with a PRV Instruction



Precautions for Correct Use

The frequency measurement function can be used with high-speed counter 0 only.

● Specifications

Item		Specifications
Number of frequency measurement inputs		1 input (high-speed counter 0 only)
Frequency measurement range		High-speed counter 0: Differential phase inputs: 0 to 50 kHz All other input modes: 0 to 100 kHz
Measurement method		Execution of the PRV instruction
Stored data	Unit	Hz
	Output data range*	Differential phase input: 0000 0000 to 0003 0D40 hex (0 to 200 kHz) All other input modes: 0000 0000 to 0001 86A0 hex (0 to 100 kHz)

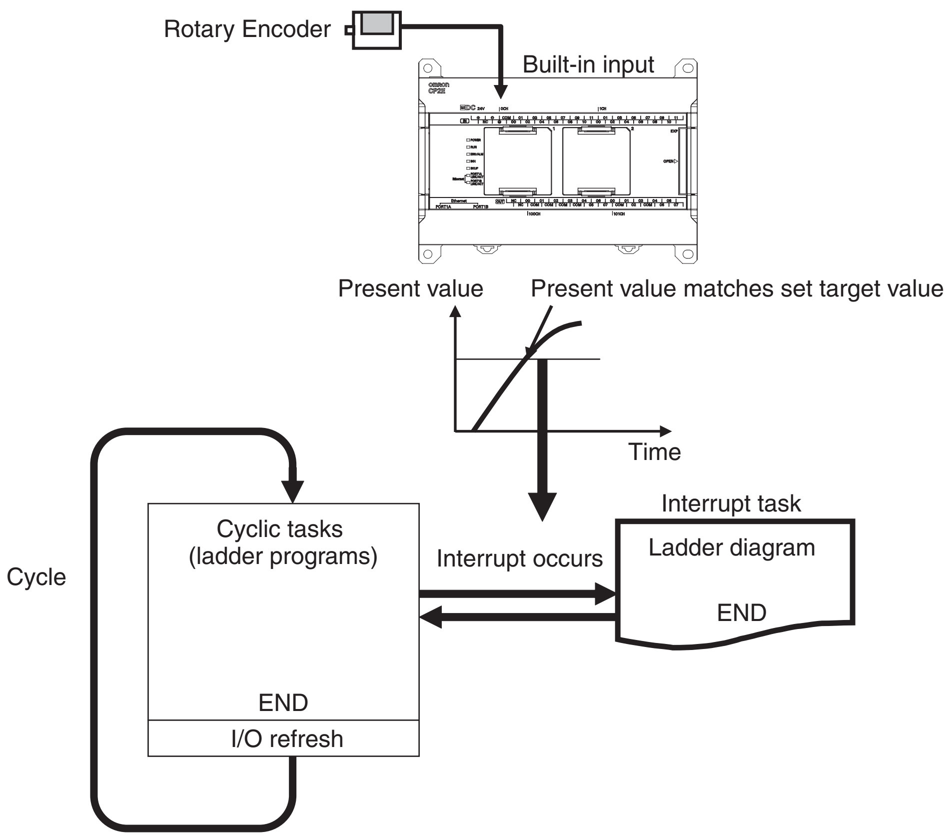
* If the frequency exceeds the maximum value, the maximum value will be stored.

11-3 High-speed Counter Interrupts

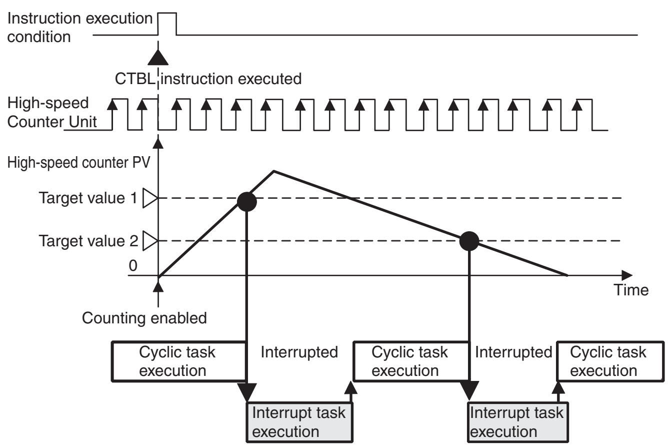
High-speed counter interrupts can be used with any model of CP2E CPU Unit.

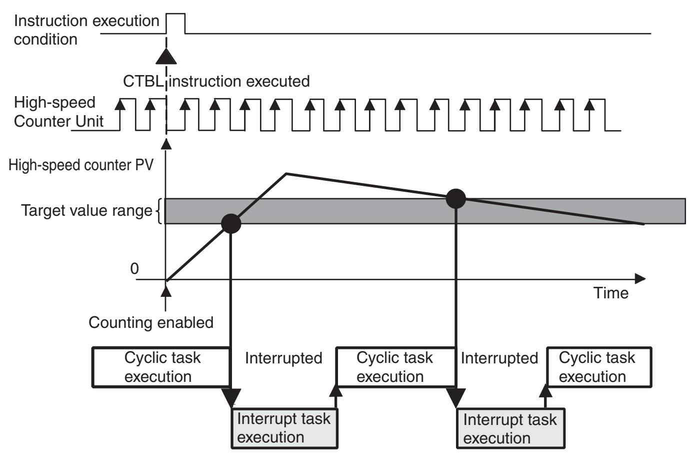
11-3-1 Overview

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 15 can be allocated with the CTBL instruction.



Target value comparison	Range comparison
The specified interrupt program can be started when the present value of the high-speed counter matches a target value.	The specified interrupt program can be started when the present value of the high-speed counter enters a set range.





Flow of Operation

1

PLC Setup

- Enable the required high-speed counters.
- Select the *Use high speed counter* Check Box for high-speed counters 0 to 5. Set the input setting, counting mode and reset method on the Built-in Tab Page of the PLC Setup using the CX-Programmer.
- Terminals 00 to 06 on the 0CH terminal block can be used for high-speed counters. High-speed counters 0 to 5 correspond to terminals 00 to 05.

2

Create ladder program	Interrupt task
	Execution of CTBL and INI instructions in a cyclic task

Write a program for interrupt tasks 0 to 15.

- Set the comparison values for the high-speed counter and the interrupt tasks (0 to 15) to be started using the CTBL instruction.
- Start the comparison using the INI instruction. The comparison can be started simultaneously when registering the comparison values using the CTBL instruction.

● High-speed Counter Interrupts Settings

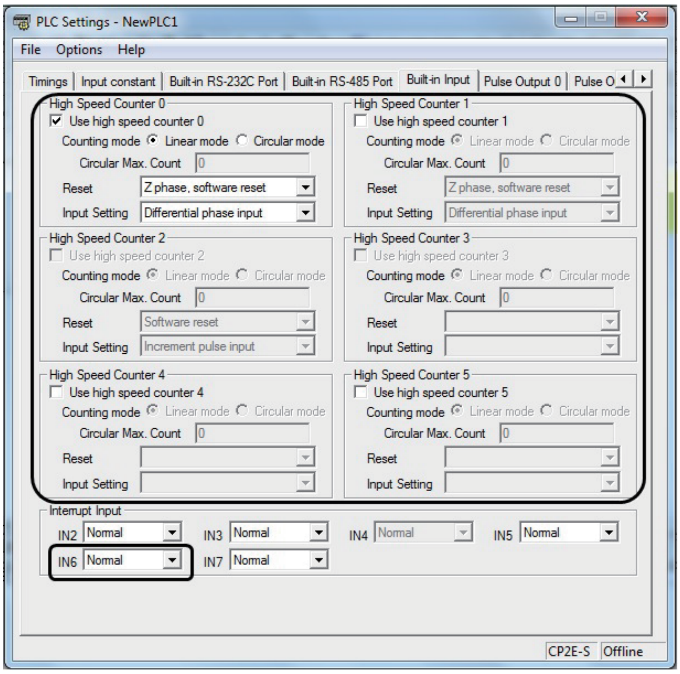
Setting in PLC Setup on Built-in Input Tab Page		Instruction	CTBL port specifier (C1)	Interrupt task number
High-speed counter 0	Select *Use* Check Box.	CTBL	#0000	0 to 15 (Specified by user.)
High-speed counter 1			#0001	
High-speed counter 2			#0002	
High-speed counter 3			#0003	
High-speed counter 4			#0004	
High-speed counter 5			#0005	

Precautions for Correct Use

A built-in input cannot be used as a normal input, interrupt input, or quick-response input if it is being used as a high-speed counter input. Refer to *8-3-3 Allocating Built-in Input Terminals* for details.

PLC Setup

Click the Built-in Input Tab and select the *Use high-speed counter* Check Box for high-speed counters 0 to 5, and then set the counting mode, reset method, and input setting.



Refer to *11-1-2 Flow of Operation* in Page *11-3* for details.

Determining High-speed Counter

High-speed counters 0 to 5 can be used for high-speed counter interrupts.

- Refer to *8-3-3 Allocating Built-in Input Terminals* for high-speed counter interrupt.
- Refer to *10-1 Interrupts* for the interrupts excluding high-speed counter interrupts.

Writing the Ladder Program

● Writing the Interrupt Task Program

Create programs for interrupt tasks 0 to 15, which are executed for the corresponding high-speed counter interrupts. Right-click a program in the CX-programmer and select **Properties**. Select any interrupt task in the **Task type** Field of the **Program Properties** Dialog Box.

- **Execution of CTBL and INI Instructions for Cyclic Task**

Execute the instructions in the following order.

Register the comparison table	Register the comparison table with the CTBL (COMPARISON TABLE LOAD) instruction. Specify the interrupt tasks to be started in this step.
⬇	
Start comparison	Start comparison with the CTBL (COMPARISON TABLE LOAD) or INI (MODE CONTROL) instruction. Here, high-speed counter interrupts will be valid.
⬇	
Stop comparison	Stop with the INI (MODE CONTROL) instruction.

Refer to *11-3-2 Present Value Comparison* for details.

11-3-2 Present Value Comparison

The comparison of the high-speed counter PV has the following two ways: Target Value Comparison and Range Comparison.

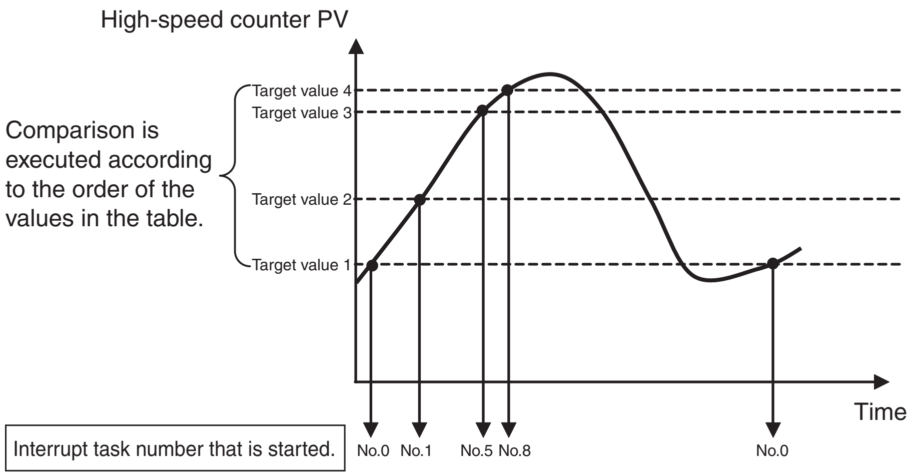
Target Value Comparison

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 high-speed counter PV matches the registered target value.
- Comparison is executed in the order set in the comparison table. Once comparison has cycled through the comparison table, it will return and wait for a match with the first target value again.

The following examples show the operation of an interrupt task for a comparison table.

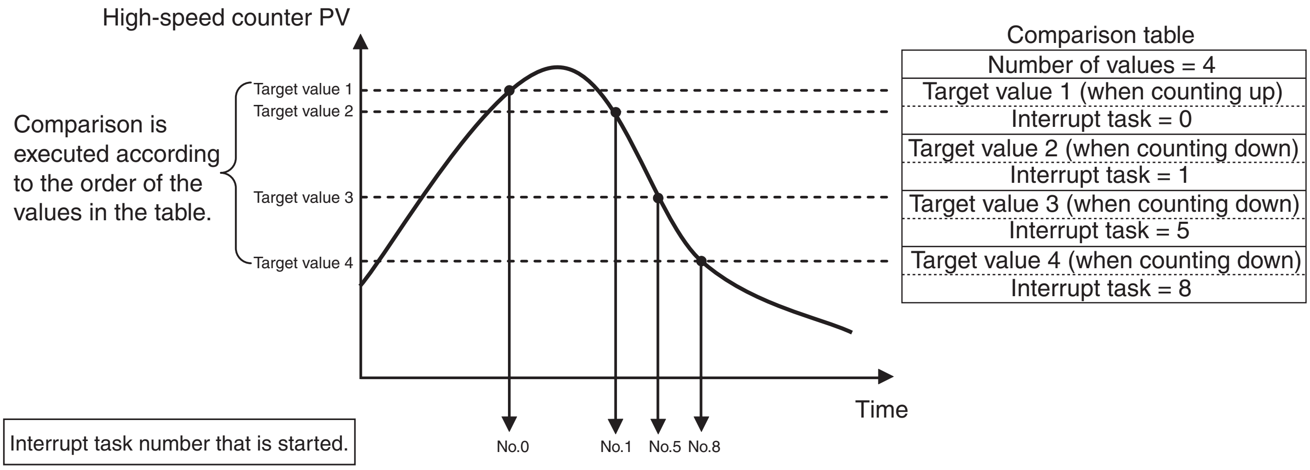
Example 1



Comparison table

Number of values = 4
Target value 1 (when counting up)
Interrupt task = 0
Target value 2 (when counting up)
Interrupt task = 1
Target value 3 (when counting up)
Interrupt task = 5
Target value 4 (when counting up)
Interrupt task = 8

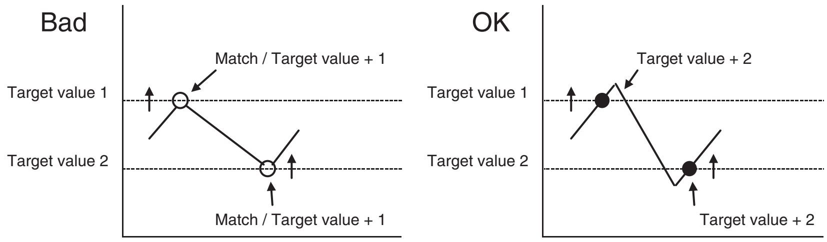
Example 2



- Up to 6 target values (between 1 and 6) can be registered in the comparison table.
- A different interrupt task can be registered for each target value.
- 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.

Precautions for Correct Use

- When the count direction (incrementing/decrementing) changes at a PV that matches a target value or a count after 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.



- The maximum response frequencies of the high-speed counters are given in the following table.

Item		E/S□□-type and N4/20 CPU Unit	N30/40/60 CPU Unit
High-speed counter 0	Incremental pulse	100kHz	100kHz
	Up and down pulses		
	Pulse plus direction		
	Differential phase (×4)	50kHz	50kHz
High-speed counter 1	Incremental pulse	100kHz	100kHz
	Up and down pulses	10kHz	
	Pulse plus direction	100kHz	
	Differential phase (×4)	5kHz	50kHz
High-speed counter 2	Incremental pulse	10kHz	100kHz
High-speed counter 3	Incremental pulse		10kHz
High-speed counter 4	Incremental pulse		
High-speed counter 5	Incremental pulse		

Precautions for Correct Use

- There are restrictions on the maximum response frequencies of the high-speed counters when using target matching.
  Use the counters for target matching under the frequencies in the following table.
  If the pulse frequencies input to the high-speed counters are higher than those in the table, target matching may be omitted.

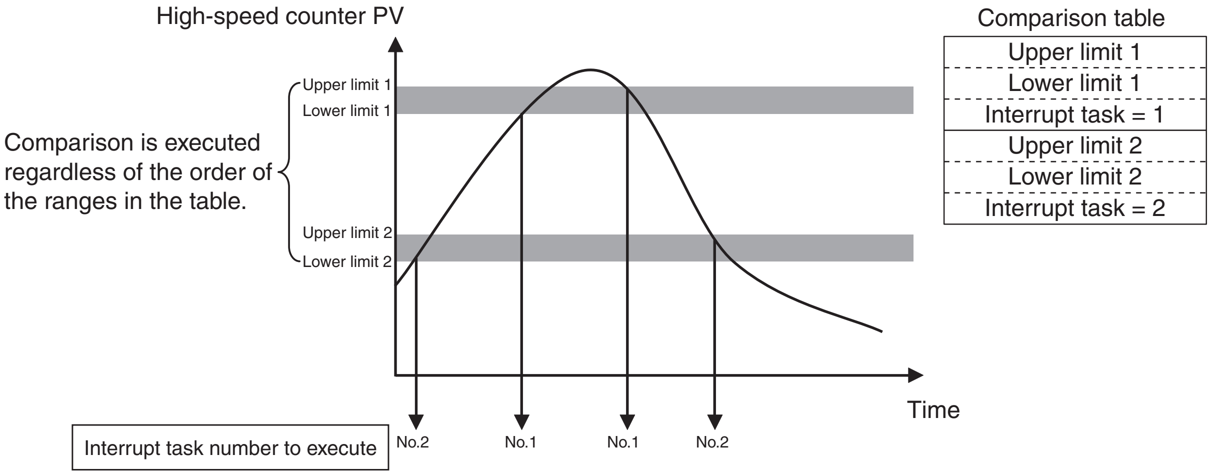
Enabling/Disabling pulse outputs	Counter numbers for target matching	Increment pulse Pulse plus direction Up and down pulses	Differential phase (×4)
Disabling pulse outputs or enabling 3-axes or less pulse outputs	–	100kHz max.	50kHz max.
Enabling 4-axes pulse outputs	Only one point	100kHz max.	50kHz max.
	More than two points	100kHz max.	40kHz max.

- When using target matching, the interval between interrupts for target matches must be longer than 3 ms plus the sum of execution time for interrupt tasks that may possibly happen at the same time.
  The sum of execution time for interrupt tasks in one cycle is stored in A442.

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 high-speed counter PV is in the range (Lower limit ≤ PV ≤ Upper limit).



- A total of 6 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 6 ranges once each cycle.
- The interrupt task is executed just once when the comparison condition goes from unmet to met.

Precautions for Correct Use

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.

Additional Information

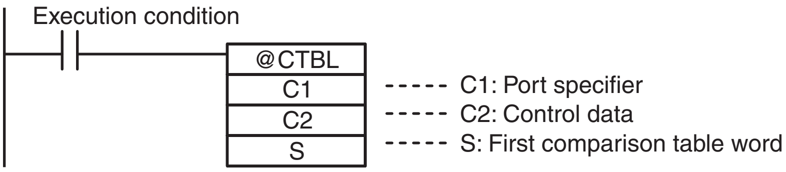
The range comparison table can be used without starting an interrupt task when the comparison condition is met. The range comparison function can be useful when you just want to know whether or not the high-speed counter PV is within a particular range.

Use the Range Comparison Condition Met Flags to determine whether the high-speed counter PV is within a registered range.

11-3-3 High-speed Counter Interrupt Instruction

COMPARISON TABLE LOAD Instruction: CTBL

The CTBL instruction compares the PV of a high-speed counter (0 to 5) to target values or ranges and executes the corresponding interrupt task (0 to 15) when the specified condition is met.

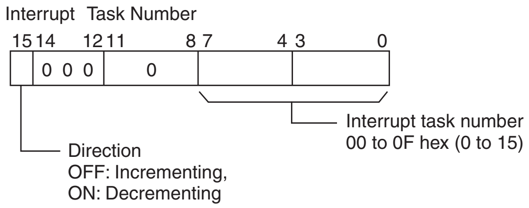


Operand		Settings	
C1	Port specifier	#0000	High-speed counter 0
		~	~
		#0005	High-speed counter 5
C2	Control data	#0000	Registers a target-value comparison table and starts the comparison operation.
		#0001	Registers a range comparison table and starts the comparison operation.
		#0002	Registers a target-value comparison table.
		#0003	Registers a range comparison table.
S	First comparison table word	Specifies the first word address of the comparison table, which is described below.	

● Contents of the Comparison Table

- 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 19 words.

Word	15 ... 0	
S	Number of target values	0001 to 0006 hex (1 to 6 target values)
S+1	Lower word of target value 1	00000000 to FFFFFFFF hex
S+2	Upper word of target value 1	
S+3	Interrupt task number for target value 1	
⋮	⋮	
S+16	Lower word of target value 6	00000000 to FFFFFFFF hex
S+17	Upper word of target value 6	
S+18	Interrupt task number for target value 6	



- Range Comparison Table
  The range comparison table requires a continuous block of 30 words for comparison conditions 1 to 6 require 5 words each (two words for the upper range value, two words for the lower range value, and one word for the interrupt task number).

Word	15 ... 0	
S	Lower word of range 1 lower limit	0000 0000 to FFFF FFFF hex (See note.)
S+1	Upper word of range 1 lower limit	
S+2	Lower word of range 1 upper limit	0000 0000 to FFFF FFFF hex (See note.)
S+3	Upper word of range 1 upper limit	
	Range 1 interrupt task number	
⋮	⋮	
S+25	Lower word of range 6 lower limit	0000 0000 to FFFF FFFF hex (See note.)
S+26	Upper word of range 6 lower limit	
S+27	Lower word of range 6 upper limit	0000 0000 to FFFF FFFF hex (See note.)
S+28	Upper word of range 6 upper limit	
S+29	Range 6 interrupt task number	

Interrupt task number
0000 to 000F hex: Interrupt task number 0 to 15
AAAA hex: Do not execute interrupt task.
FFFF hex: Ignore the settings for this range.

Note Always set the upper limit greater than or equal to the lower limit for any one range.

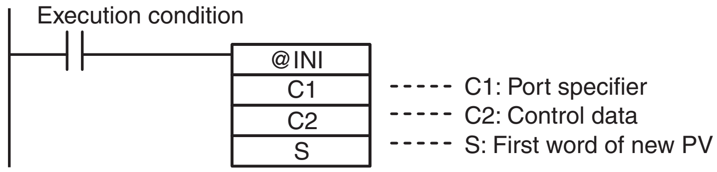
MODE CONTROL Instruction: INI

The INI instruction is used for the following items.

- Starting and stopping comparison with the high-speed counter comparison table
  Use the CTBL instruction to register the target value or range comparison table before using INI to start or stop comparison.

  If the comparison is started simultaneously when registering the comparison table and then the high-speed counter interrupts are always valid, the INI instruction is not required.

- Changing the PV of a High-speed Counter



Operand		Settings	
C1	Port specifier	#0010	High-speed counter 0
		~	~
		#0015	High-speed counter 5
C2	Control data	#0000	Start comparison.
		#0001	Stop comparison.
		#0002	Change the PV.
S	First word of new PV	S contains the first word of the new PV when C is set to #0002 (change the PV).	

Example 1: Target Value Comparison

In this example, high-speed counter 0 operates in linear mode and starts interrupt task 10 when the PV reaches 30,000 (0000 7530 hex) and starts interrupt task 11 when the PV reaches 20,000 (0000 4E20 hex).

1 Set high-speed counter 0 in the PLC Setup's Built-in Input Tab.

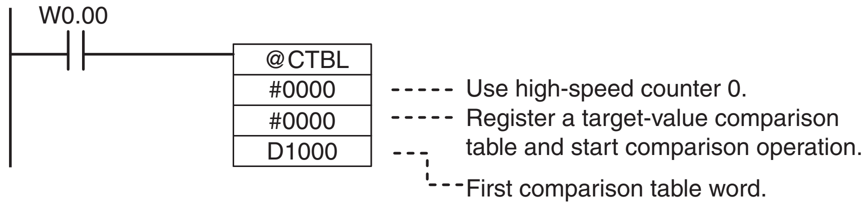
Item	Setting
High-speed counter 0	Use counter
Counting mode	Linear mode
Circular Max. Count	–
Reset method	Software reset
Input Setting	Up/Down inputs

2 Set the target-value comparison table in words D1000 to D1006.

Word	Setting	Function	
D1000	#0002	Number of target values = 2	
D1001	#7530	Rightmost 4 digits of the target value 1 data (30000)	Target value = 30,000(0000 7530 hex)
D1002	#0000	Leftmost 4 digits of the target value 1 data (30000)	
D1003	#000A	Target value 1 Bit 15: 0 (incrementing) Bits 00 to 07: A hex (interrupt task number 10)	
D1004	#4E20	Rightmost 4 digits of the target value 2 data (20000)	Target value = 20,000(0000 4E20 hex)
D1005	#0000	Leftmost 4 digits of the target value 2 data (20000)	
D1006	#800B	Target value 2 Bit 15: 1 (decrementing) Bits 00 to 07: B hex (interrupt task number 11)	

3 Create the programs for interrupt tasks 10 and 11.

4 Use the CTBL instruction to start the comparison operation with high-speed counter 0 and interrupt tasks 10 and 11.

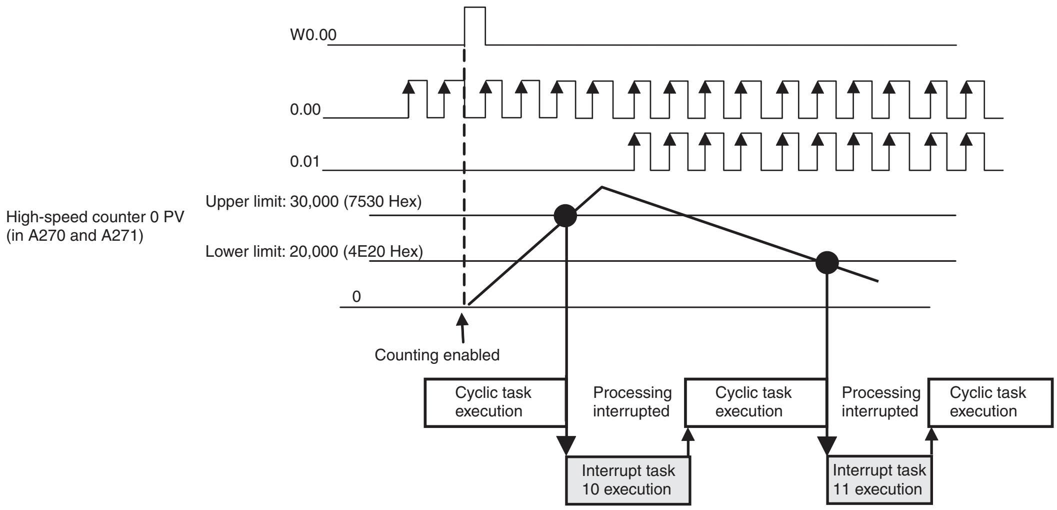


When execution condition W0.00 turns ON, the comparison starts with high-speed counter 0.

When the PV of high speed counter 0 reaches 30,000, cyclic task execution is interrupted, and interrupt task 10 is executed.

When the PV of high speed counter 0 reaches 20,000, cyclic task execution is interrupted, and interrupt task 11 is executed.

When interrupt task 10 or 11 execution has been completed, execution of the interrupted cyclic task resumes.



Example 2: Range Comparison

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 to 50,000 (0000 C350 hex).

1 Set high-speed counter 1 on the PLC Setup's Built-in Input Tab Page.

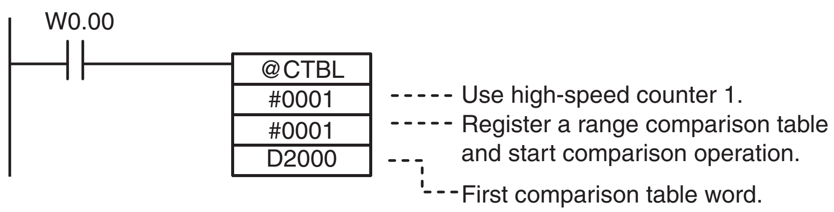
Item	Setting
High-speed counter 1	Use counter
Counting mode	Circular mode
Circular Max. Count	50,000
Reset method	Software reset (continue comparing)
Input Setting	Up/Down inputs

2 Set the range comparison table starting at word D2000. Even though range 1 is the only range being used, all 30 words must still be dedicated to the range comparison table.

Word	Setting	Function	
D2000	#61A8	Rightmost 4 digits of range 1 lower limit	Lower limit value: 25,000
D2001	#0000	Leftmost 4 digits of range 1 lower limit	
D2002	#639C	Rightmost 4 digits of range 1 upper limit	Upper limit value: 25,500
D2003	#0000	Leftmost 4 digits of range 1 upper limit	
D2004	#000C	Range 1 interrupt task number = 12 (C hex)	
D2005 to D2008	All #0000	Range 2 lower and upper limit values (Not used and do not need to be set.)	Range 2 settings
D2009	#FFFF	Disables range 2.	
≀			
D2014 D2019 D2024 D2029	#FFFF	Set the fifth word for ranges 3 to 6 (listed at left) to #FFFF (Range settings are invalid) to disable those ranges.	

3 Create the program for interrupt task 12.

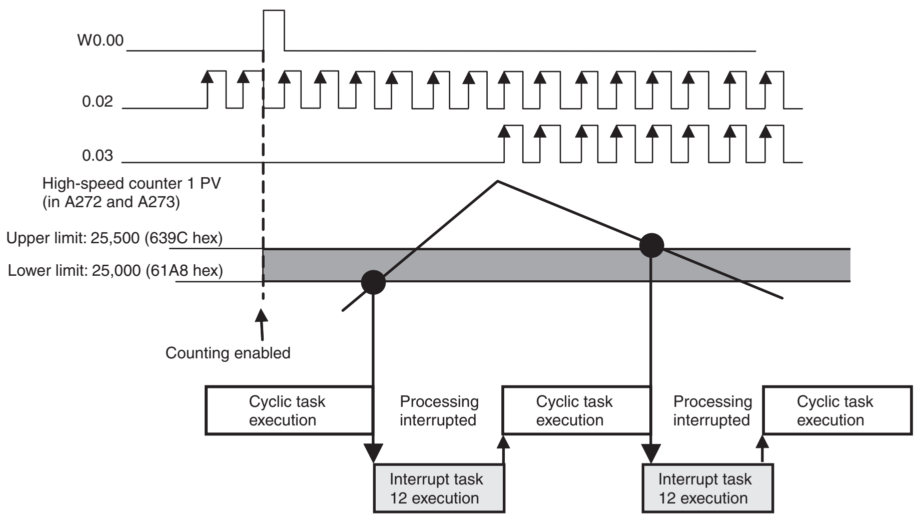
4 Use the CTBL instruction to start the comparison operation with high-speed counter 1 and interrupt task 12.



When execution condition W0.00 turns 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 execution is interrupted, and interrupt task 12 is executed.

When interrupt task 12 execution is completed, execution of the interrupted cyclic task resumes.



11-4 Related Auxiliary Area Bits and Words

Bits and Words Allocated in the Auxiliary Area

Contents		High-speed counter 0	High-speed counter 1	High-speed counter 2	High-speed counter 3	High-speed counter 4	High-speed counter 5
High-speed counter PV storage words	Leftmost 4 digits	A271	A273	A317	A319	A323	A325
	Rightmost 4 digits	A270	A272	A316	A318	A322	A324
Range Comparison Condition Met Flags	Range 1 Comparison Condition Met Flag (ON for match.)	A274.00	A275.00	A320.00	A321.00	A326.00	A327.00
	Range 2 Comparison Condition Met Flag (ON for match.)	A274.01	A275.01	A320.01	A321.01	A326.01	A327.01
	Range 3 Comparison Condition Met Flag (ON for match.)	A274.02	A275.02	A320.02	A321.02	A326.02	A327.02
	Range 4 Comparison Condition Met Flag (ON for match.)	A274.03	A275.03	A320.03	A321.03	A326.03	A327.03
	Range 5 Comparison Condition Met Flag (ON for match.)	A274.04	A275.04	A320.04	A321.04	A326.04	A327.04
	Range 6 Comparison Condition Met Flag (ON for match.)	A274.05	A275.05	A320.05	A321.05	A326.05	A327.05
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	A326.08	A327.08
Overflow/ Underflow Flags	ON when an overflow or underflow has occurred in the high-speed counter's PV.	A274.09	A275.09	A320.09	A321.09	A326.09	A327.09
Count Direction Flags	0: Decrementing 1: Incrementing	A274.10	A275.10	A320.10	A321.10	A326.10	A327.10
High-speed Counter Reset Flags	ON at a software reset	A531.00	A531.01	A531.02	A531.03	A531.04	A531.05

11-5 Application Example

Using a Rotary Encoder to Measure Positions

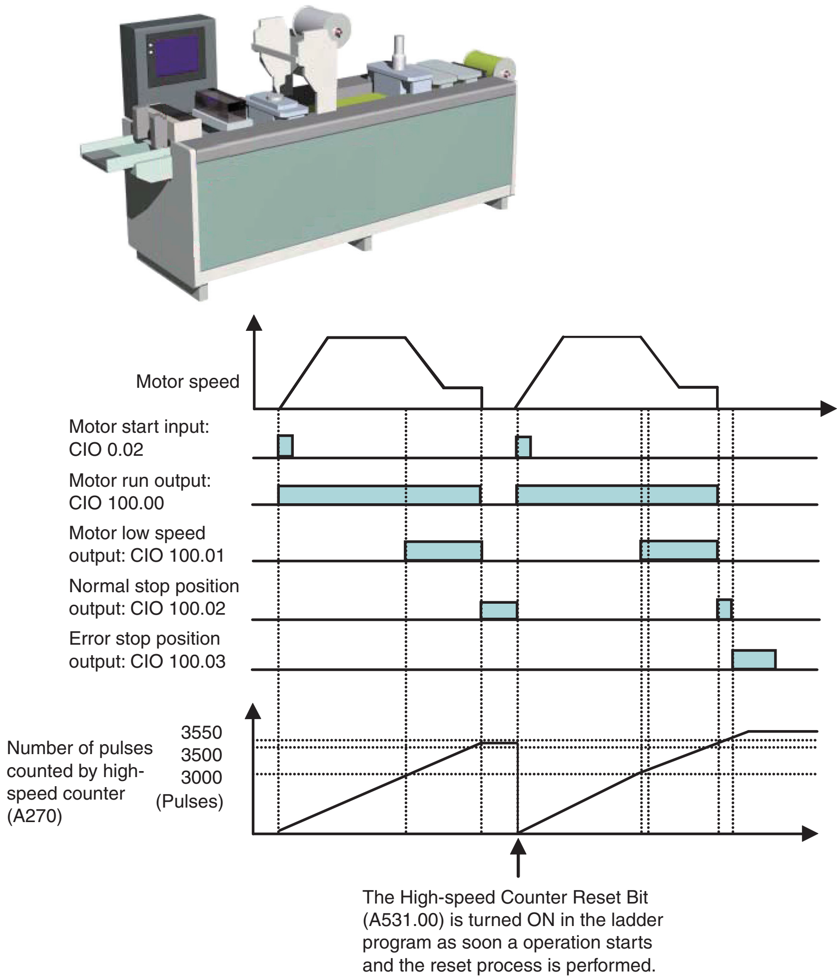
● Functions Used: High-speed Counting for a Built-in Input

A high-speed counter input can be used by connecting a rotary encoder to a built-in input. A CP2E CPU Unit is equipped with more than one high-speed counter input, making it possible to control devices for multiple axes with a single PLC.

High-speed counters can be used for high-speed processing, using either target value comparison or range comparison to create interrupts. Interrupt tasks are executed when the counter value reaches a specific target value or range.

● Operation Overview

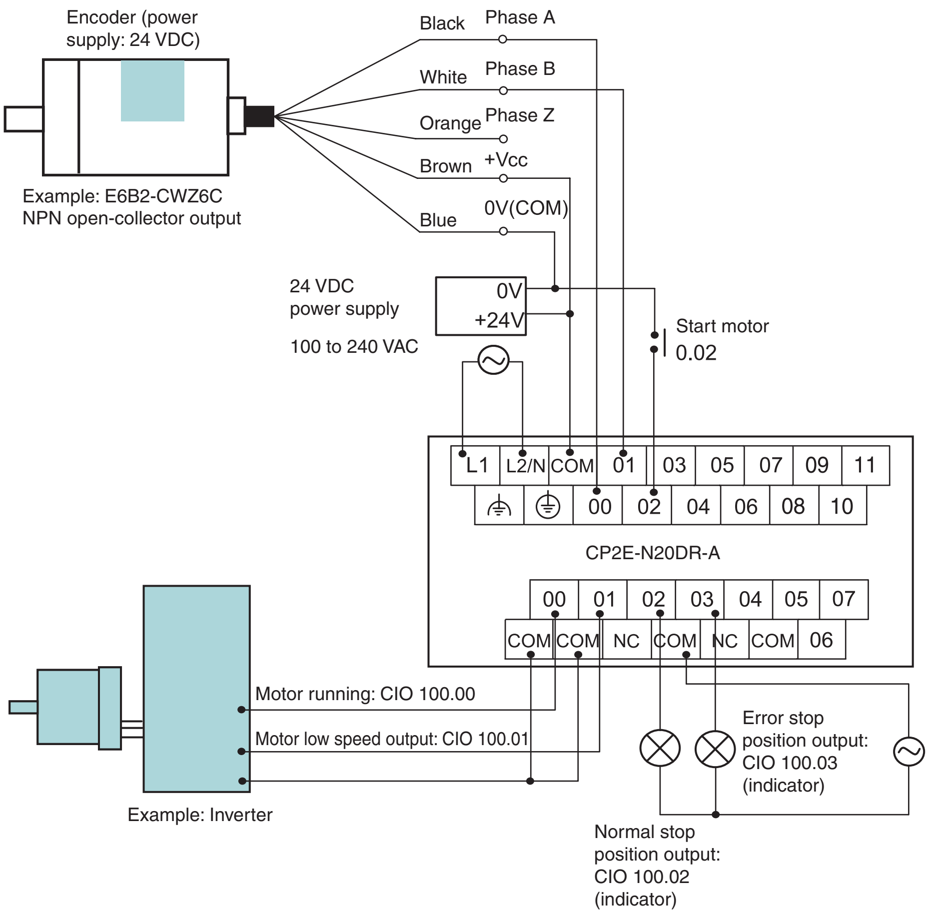
A sheet feeder is controlled to feed constant lengths in a given direction, e.g., for vacuum packing of food products.



While the pulse count is between 3,500 and 3,550, normal stop position output (CIO 100.02) will be ON. If the pulse count exceeds 3550, the error stop position output (CIO 100.03) will turn ON.

● System Configuration

Wiring Example

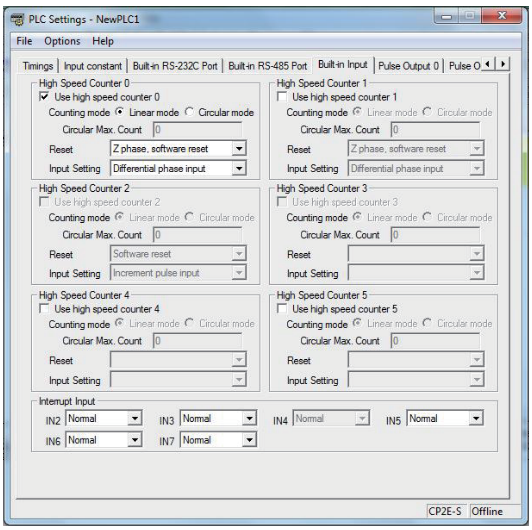


PLC Setup

Use the following procedure to enable high-speed counter 0.

1 Open the PLC Settings Dialog Box.

2 Click the Built-in Input Tab.



3 Select the ***Use high speed counter 0*** Check Box for high-speed counter 0.

4 Select ***Linear Mode*** for the counting mode.

5 Select ***Software reset (comparing)*** for the reset method.

6 Select ***Differential phase input*** for the input setting.

7 Close the PLC Settings Dialog Box.

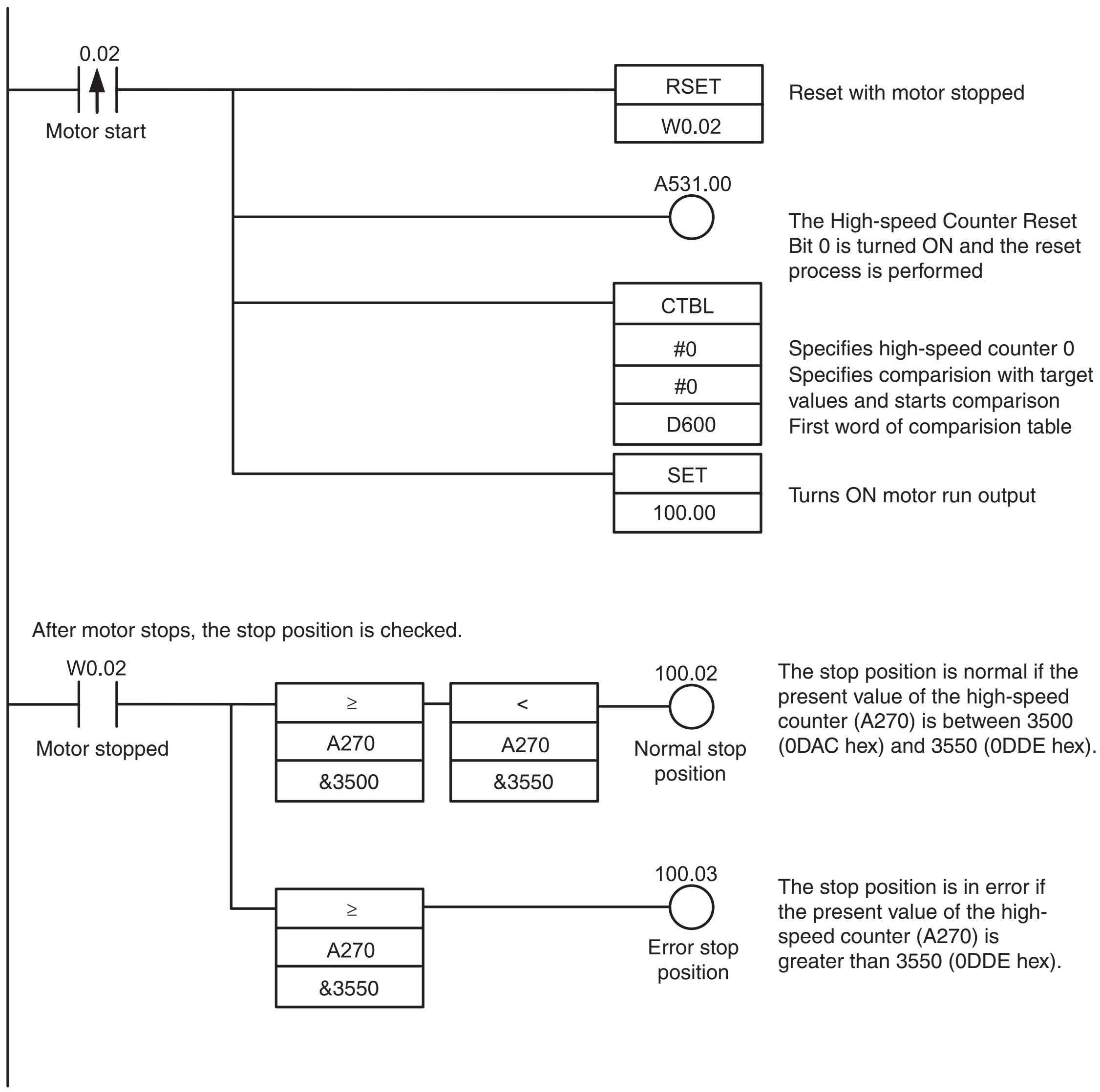
8 Restart the PLC.
The changes made to the PLC Setup is applied.

● Programming Example 1

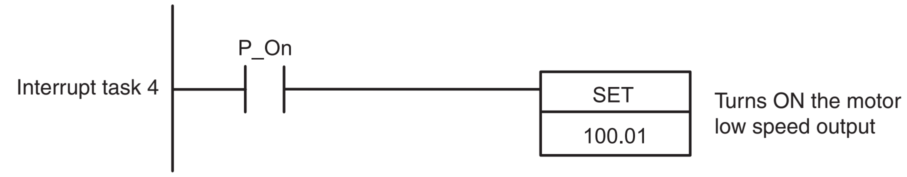
In this example, the CTBL (COMPARISON TABLE LOAD) instruction is used to create an interrupt when the target value is reached. Slowing and stopping are executed as interrupt tasks, allowing high-speed processes to be executed without affecting the cycle time.

Ladder Program

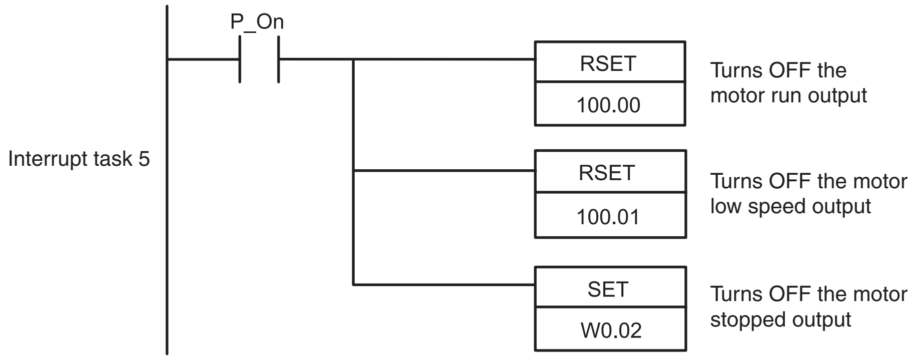
Use the CTBL instruction to execute interrupt tasks when the target positions are reached.



When the PV of the high-speed counter matches target value 1 (3000), interrupt task 4 is executed.



When the present vale of the high-speed counter matches target value 2 (3500), interrupt task 5 is executed.



DM Area Setup

The comparison table for the CTBL (COMPARISON TABLE LOAD) instruction is set in D600 through D606.

Word	Value	Contents
D600	0002	Number of target values: 2
D601	0BB8	Target value 1: 3000 (BB8 hex)
D602	0000	
D603	0004	Target value 1: Interrupt task No.4
D604	0DAC	Target value 2: 3500 (DAC hex)
D605	0000	
D606	0005	Target value 2: Interrupt task No.5

12

Pulse Outputs

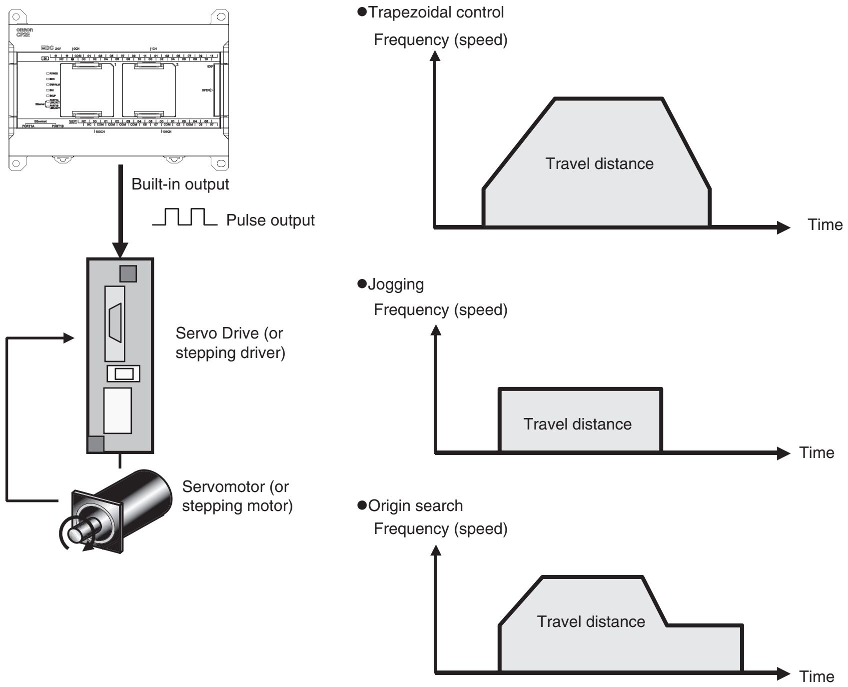
This section describes positioning functions such as trapezoidal control, jogging, and origin searches.

12-1 Overview

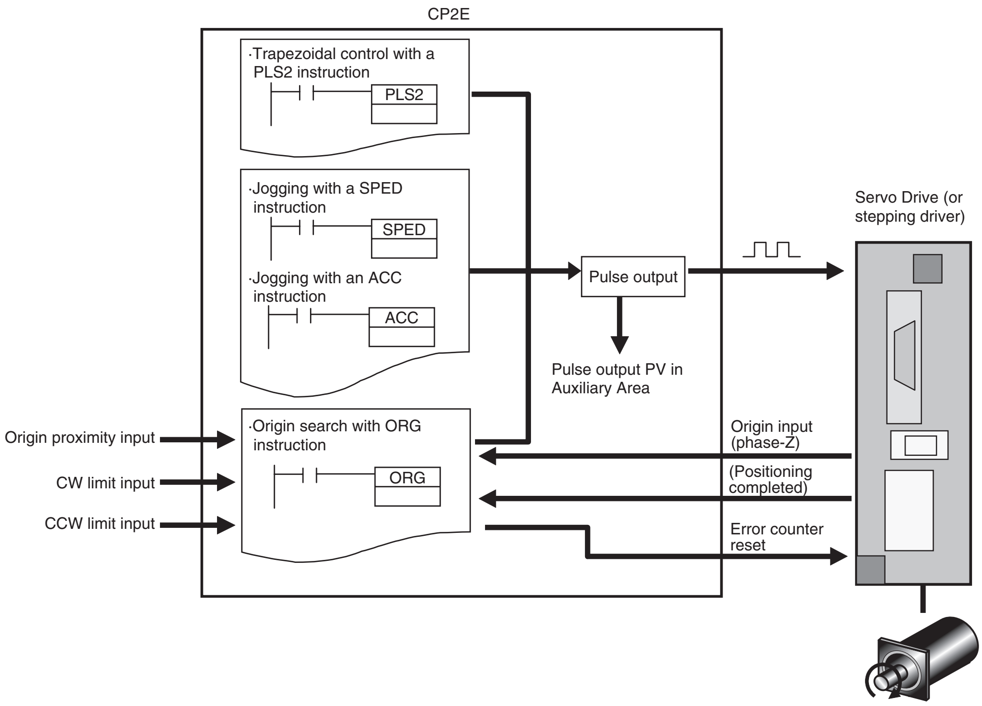
Pulse outputs can be used only with the CP2E N/S☐☐-type CPU Unit with transistor outputs.

12-1-1 Overview

Pulse outputs can be output from the CPU Unit's built-in outputs using instructions to perform positioning or speed control with a servomotor or a stepping motor that accepts pulse inputs. It is also possible to perform origin searches or origin returns.



Positioning is performed with a servomotor or stepping motor in the following configuration.



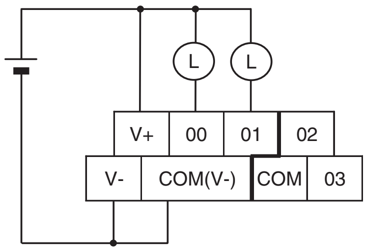
Wiring for S□□-type CPU Unit

An external power supply is required for S□□-type CPU Units when using the PWM output.

Provide a DC24V external power supply to V+ and V- terminals as follows.

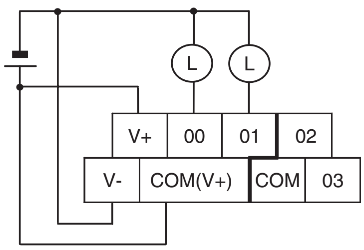
Wiring Example

Sinking outputs



Although V- and COM(V-) are connected internally, also wire them externally.

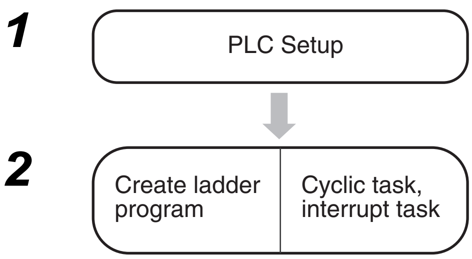
Sourcing outputs



Although V+ and COM(V+) are connected internally, also wire them externally.

Do not connect an external power supply to N□□-type CPU Units.

12-1-2 Flow of Operation

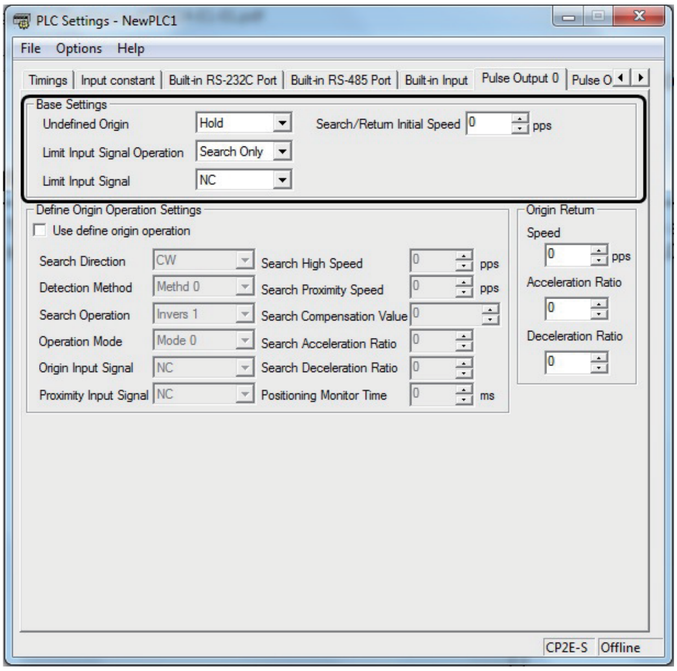


- Setting is required for the following situations:
  - Performing an origin search.
  - Using the Limit Input Signal as an input to functions other than origin searches.

Execute instructions related to pulse control.
Set pulse output 0 to 03, and whether to use terminals 00 and 02, or 01 and 03 on the CIO 100 and CIO 101 terminal blocks for pulse outputs.

PLC Setup

To perform an origin search or to use a Limit Input Signal as an input to a function other than origin search, set the parameters on the Pulse Output 0 to 3 Tab Pages in the PLC Setup.



Pulse Output 0 to 3 Tab Page

Item		Setting	Description
Base Settings	**Undefined Origin**	Hold	When a Limit Input Signal is input, the pulse output is stopped and the previous status is held.
		Undefined	When a Limit Input Signal is input, the pulse output is stopped and origin becomes undefined.
	Limit Input Signal Operation	Search Only	The CW/CCW Limit Input Signal is used for origin searches only.
		Always	The CW/CCW Limit Input Signal is used by functions other than origin search.
	Limit Input Signal	NC	Select when using NC contact (b contact) for the Limit Input Signal.
		NO	Select when using NO contact (a contact) for the Limit Input Signal.
	Search/Return Initial Speed	Set the motor’s starting speed when performing an origin search. Specified in pulses per second (pps).	

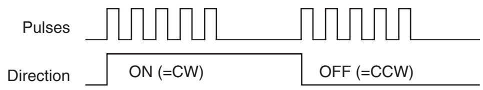
Note The power supply must be restarted after the PLC Setup is transferred in order to enable the pulse output settings.

Refer to *12-6 Defining Origin Position* for origin search settings in the PLC Setup.

Setting the Pulse Output Port Number, Assigning Pulse Output Terminals, and Wiring

● Pulse Output Method

Only the following pulse output plus a direction output can be used as the pulse output method.



● Pulse Output Port Number and Output Terminals

The following terminals are used for pulse outputs according to the pulse output port number.

Output terminal block		Pulse output method	Other functions that cannot be used at the same time	
Terminal block label	Terminal number	Pulse plus direction	Normal output	PWM output
CIO 100	00	Pulse output 0, pulse	Normal output 0	–
	01	Pulse output 1, pulse	Normal output 1	PWM output
	02	Pulse output 0, direction	Normal output 2	–
	03	Pulse output 1, direction	Normal output 3	–
CIO 101	00	Pulse output 2, pulse	Normal output 8	–
	01	Pulse output 3, pulse	Normal output 9	–
	02	Pulse output 2, direction	Normal output 10	–
	03	Pulse output 3, direction	Normal output 11	–

Origin Searches

Use the following input and output terminals for origin searches.

Input Terminals

- N20/30/40/60 or S30/40/60 CPU Units

Input terminal block		Setting in PLC Setup	Other functions that cannot be used at the same time			
Terminal block label	Terminal number	Enable origin searches for pulse outputs 0 to 3	Normal inputs	Interrupt inputs	Quick-response inputs	High-speed counter setting Increment pulse input
CIO 0	06	Pulse 0, Origin input signal	Normal input 6	Interrupt input 6	Quick-response input 6	High-speed counter 5
	07	Pulse 1, Origin input signal	Normal input 7	Interrupt input 7	Quick-response input 7	–
	08	Pulse 2, Origin input signal (Note 1)	Normal input 8	Interrupt input 8 (Note 2)	Quick-response input 8 (Note 2)	–
	09	Pulse 3, Origin input signal (Note 1)	Normal input 9	Interrupt input 9 (Note 2)	Quick-response input 9 (Note 2)	–
	10	Pulse 0, Origin proximity input signal	Normal input 10	–	–	–
	11	Pulse 1, Origin proximity input signal	Normal input 11	–	–	–
CIO 1	00	Pulse 2, Origin proximity input signal (Note 1)	Normal input 12	–	–	–
	01	Pulse 3, Origin proximity input signal (Note 1)	Normal input 13	–	–	–

Note 1 Only supported by N30/40/60 CPU Units.

2 Only supported by N20/30/40/60 CPU Units.

- N14 CPU Units

Input terminal block		Setting in PLC Setup	Other functions that cannot be used at the same time			
Terminal block label	Terminal number	Enable origin searches for pulse outputs 0 and 1	Normal inputs	Interrupt inputs	Quick-response inputs	High-speed counter setting Increment pulse input
CIO 0	03	Pulse 0, Origin proximity input signal	Normal input 3	Interrupt input 3	Quick-response input 3	–
	:	:				
	05	Pulse 1, Origin proximity input signal	Normal input 5	Interrupt input 5	Quick-response input 5	High-speed counter 4
	06	Pulse 2, Origin input signal	Normal input 6	Interrupt input 6	Quick-response input 6	High-speed counter 5
	07	Pulse 3, Origin input signal	Normal input 7	Interrupt input 7	Quick-response input 7	–

Output Terminals

Output terminal block		Setting in PLC Setup	Other functions that cannot be used at the same time
Terminal block label	Terminal number	Enable origin searches for pulse outputs 0 and 1	Normal outputs
CIO 100	04	Pulse 0, Error counter reset output	Normal output 4 (Note 2)
	05	Pulse 1, Error counter reset output	Normal output 5 (Note 2)
	06	Pulse 2, Error counter reset output (Note 1)	Normal output 6
	07	Pulse 3, Error counter reset output (Note 1)	Normal output 7

Note 1 Only supported by N30/40/60 CPU Units.

2 When the origin search is in operating mode 0, normal output 4 and 5 can be used at the same time.

Connecting the Servo Drive and External Sensors

● Connections for Pulse Output 0

Terminal block		Addresses		Signal	Origin search		
Terminal block label	Terminal number				Operating mode 0	Operating mode 1	Operating mode 2
CIO 100	00	CIO 100.00	Stored in A276 and A277.	Pulse	Connect to Servo Drive's pulse input (PULS).		
	02	CIO 100.02		Direction	Connect to Servo Drive's direction input (SIGN).		
Normal input		The external signal must be received as an input and the input status must be written to A540.08 in the ladder program.		CW limit sensor	Connect sensor to a normal input terminal.		
Normal input		The external signal must be received as an input and the input status must be written to A540.09 in the ladder program.		CCW limit sensor	Connect sensor to a normal input terminal.		
CIO 0	03	CIO 0.03		Origin proximity input	Connect to sensor for N14 CPU Unit.		
	06	CIO 0.06		Origin input	Connect to open-collector output from sensor or other device.	Connect to the phase-Z signal from the Servo Drive.	Connect to the phase-Z signal from the Servo Drive.
	10	CIO 0.10		Origin proximity input	Connect to sensor for N20/30/40/60 CPU Unit.		
CIO 100	04	CIO 100.04		Error counter reset output	Not used.	Connect to error counter reset (ECRST) of the Servo Drive.	
Normal input		The external signal must be received as an input and the input status must be written to A540.10 in the ladder program.		Positioning completed input		Not used.	Connect the Positioning Completed Signal (INP) from the Servo Drive to a normal input terminal.

● Connections for Pulse Output 1

Terminal block label	Terminal number	Addresses		Signal	Origin search: Operating mode 0	Origin search: Operating mode 1	Origin search: Operating mode 2
CIO 100	01	CIO 100.01	Stored in A278 and A279	Pulse	Connect to Servo Drive's pulse input (PULS).		
	03	CIO 100.03		Direction	Connect to Servo Drive's direction input (SIGN).		
Normal input		The external signal must be received as an input and the input status must be written to A541.08 in the ladder program.		CW limit sensor	Connect sensor to a normal input terminal.		
Normal input		The external signal must be received as an input and the input status must be written to A541.09 in the ladder program.		CCW limit sensor	Connect sensor to a normal input terminal.		
CIO 0	05	CIO 0.05		Origin proximity input	Connect to sensor for N14 CPU Unit.		
	07	CIO 0.07		Origin input	Connect to open-collector output from sensor or other device.	Connect to the phase-Z signal from the Servo Drive.	Connect to the phase-Z signal from the Servo Drive.
	11	CIO 0.11		Origin proximity input	Connect to sensor for N20/30/40/60 CPU Unit.		
CIO 100	05	CIO 100.05		Error counter reset output	Not used.	Connect to error counter reset (ECRST) of the Servo Drive.	
Normal input		The external signal must be received as an input and the input status must be written to A541.10 in the ladder program.		Positioning completed input		Not used.	Connect the Positioning Completed Signal (INP) from the Servo Drive to a normal input terminal.

● Connections for Pulse Output 2 (Only for N30/40/60 CPU Units)

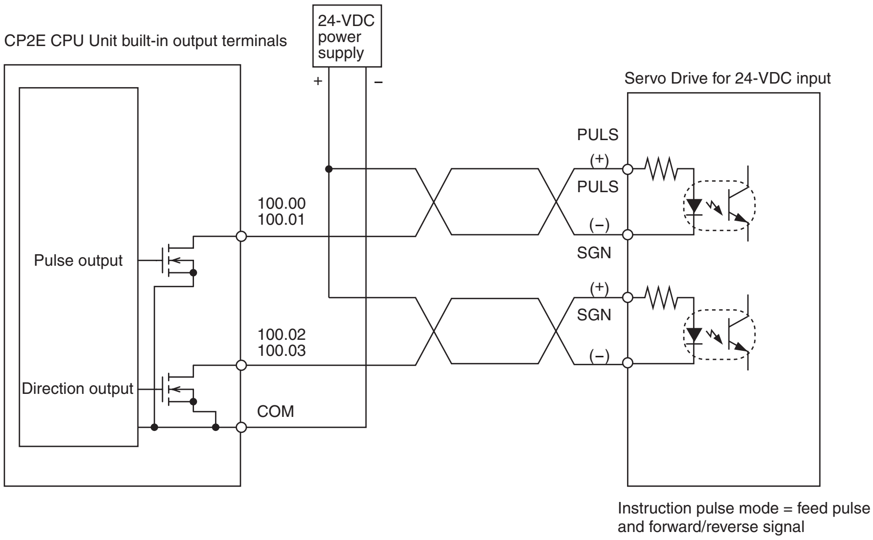
Terminal block label	Terminal number	Addresses		Signal	Origin search: Operating mode 0	Origin search: Operating mode 1	Origin search: Operating mode 2
CIO 101	00	CIO 101.01	Stored in A52 and A53.	Pulse	Connect to Servo Drive's pulse input (PULS).		
	02	CIO 101.02		Direction	Connect to Servo Drive's direction input (SIGN).		
Normal input		The external signal must be received as an input and the input status must be written to A542.08 in the ladder program.		CW limit sensor	Connect sensor to a normal input terminal.		
Normal input		The external signal must be received as an input and the input status must be written to A542.09 in the ladder program.		CCW limit sensor	Connect sensor to a normal input terminal.		
CIO 0	08	CIO 0.08		Origin input	Connect to open-collector output from sensor or other device.	Connect to the phase-Z signal from the Servo Drive.	Connect to the phase-Z signal from the Servo Drive.
CIO 1	00	CIO 1.00		Origin proximity input	Connect to sensor		
CIO 100	06	CIO 100.06		Error counter reset output	Not used.	Connect to error counter reset (ECRST) of the Servo Drive.	
Normal input		The external signal must be received as an input and the input status must be written to A542.10 in the ladder program.		Positioning completed input		Not used.	Connect the Positioning Completed Signal (INP) from the Servo Drive to a normal input terminal.

● Connections for Pulse Output 3 (Only for N30/40/60 CPU Units)

Terminal block		Addresses		Signal	Origin search		
Terminal block label	Terminal number				Operating mode 0	Operating mode 1	Operating mode 2
CIO 101	01	CIO 101.01	Stored in A54 and A55.	Pulse	Connect to Servo Drive's pulse input (PULS).		
	03	CIO 101.03		Direction	Connect to Servo Drive's direction input (SIGN).		
Normal input		The external signal must be received as an input and the input status must be written to A543.08 in the ladder program.		CW limit sensor	Connect sensor to a normal input terminal.		
Normal input		The external signal must be received as an input and the input status must be written to A543.09 in the ladder program.		CCW limit sensor	Connect sensor to a normal input terminal.		
CIO 0	09	CIO 0.09		Origin input	Connect to open-collector output from sensor or other device.	Connect to the phase-Z signal from the Servo Drive.	Connect to the phase-Z signal from the Servo Drive.
CIO 1	01	CIO 1.01		Origin proximity input	Connect to sensor		
CIO 100	07	CIO 100.07		Error counter reset output	Not used.	Connect to error counter reset (ECRST) of the Servo Drive.	
Normal input		The external signal must be received as an input and the input status must be written to A543.10 in the ladder program.		Positioning completed input		Not used.	Connect the Positioning Completed Signal (INP) from the Servo Drive to a normal input terminal.

● Pulse Output Wiring

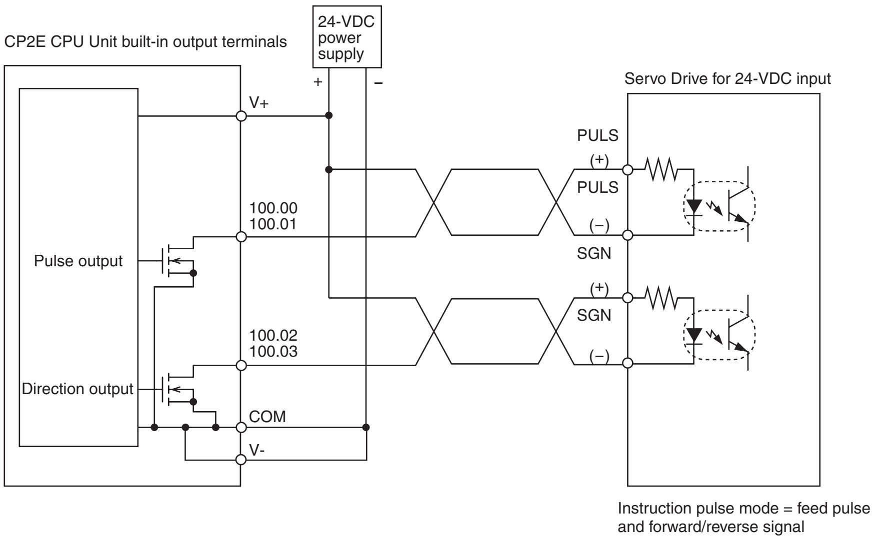
N☐☐-type (Example: Sinking outputs)



S☐☐-type (Example: Sinking outputs)

It is necessary to wire an external power supply to S☐☐-type CPU Units.
Connect a DC24V external power supply between V+ and V- in order to use terminals 00 and 01 on terminal block CIO 100.

Note COM corresponding to CIO 100.00 and CIO 100.01 has been internally connected with V- for sinking output models, with V+ for sourcing output models.



Connecting to OMRON Servo Drives

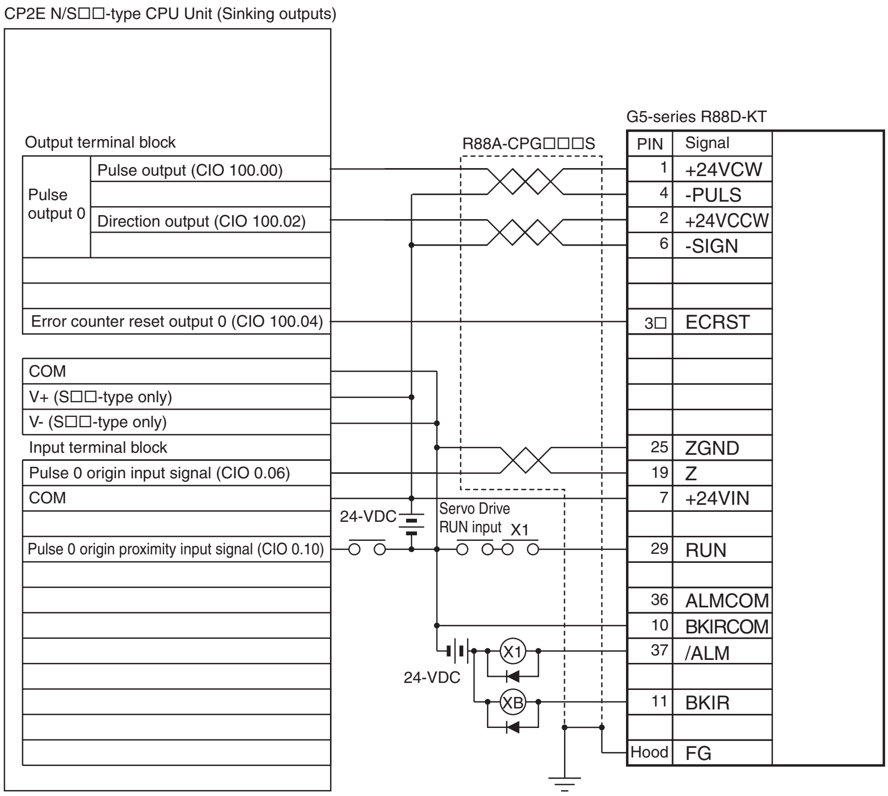
Use the following cables to connect to an OMRON Servo Drive.

OMRON Servo Drive	Cable mode: □ Indicates the cable length (1m or 2m)
G5 Series General-purpose Input Type (R88D-KT)	R88A-CPG□□□S

Set the Servo Drive's command pulse mode to feed pulse and forward/reverse signals because the method of pulse output from a CP2E CPU Unit is pulse + direction.

● Connecting to a G5-series Servo Drive

Origin Search: Operating Mode 1



Only S□□-type CPU Units can wire V+ and V-. Do not wire them in N□□-type CPU Units.

R88A-CPG☐☐☐S Cables for G5-series Servo Drives

No.	Wire Mark Colors	Symbol	No.	Wire Mark Colors	Symbol
1	Orange/Red (1)	+24VCW	26	White/Red (3)	SI3
2	Orange/Black (1)	+24VCCW	27	Pink/Black (3)	SI4
3	Gray/Red (1)	+CW/+PULS/+FA	28	White/Black (3)	SI5
4	Gray/Black (1)	-CW/-PULS/-FA	29	Yellow/Red (3)	SI6
5	White/Red (1)	+CCW/+SIGN/+FB	30	Pink/Red (3)	SI7
6	White/Black (1)	-CCW/-SIGN/-FB	31	Yellow/Black (3)	SI8
7	Yellow/Red (1)	+24VIN	32	Gray/Black (4)	SI9
8	Pink/Red (1)	SI1	33	Orange/Red (4)	SI10
9	Pink/Black (1)	SI2	34	White/Red (4)	SO2-
10	Orange/Red (2)	SO1-	35	White/Black (4)	SO2+
11	Orange/Black (2)	SO1+	36	Yellow/Red (4)	ALMCOM
12	Yellow/Black (1)	-	37	Yellow/Black (4)	/ALM
13	Gray/Black (2)	SENGND	38	Pink/Red (4)	SO4-
14	White/Red (2)	REF/TREF1/VLIM	39	Pink/Black (4)	SO4+
15	White/Black (2)	AGND1	40	Gray/Red (4)	-
16	Yellow/Red (2)	PCL/TREF2	41	Orange/Black (4)	-
17	Yellow/Black (2) Pink/Black (2)	AGND2	42	Gray/Red (5)	BAT
			43	Gray/Black (5)	BATGND
18	Pink/Red (2)	NCL	44	White/Red (5)	+CWLD
19	Orange/Red (2)	Z	45	White/Black (5)	-CWLD
20	Gray/Red (2)	SEN	46	Yellow/Red (5)	+CCWLD
21	Orange/Red (3)	+A	47	Yellow/Black (5)	-CCWLD
22	Orange/Black (3)	-A	48	Pink/Black (5)	-B
23	Gray/Red (3)	+Z	49	Pink/Red (5)	+B
24	Gray/Black (3)	-Z	50	-	-
25	Orange/Black (5)	ZGND	Hood	-	FG

10150-3000PE Connector Plug (3M)

10350-52A0-008 Connector Plug (3M)

AWG24 × 25P UL20276 Cable

Each twisted pair has wires of the same color and number of marks.

Example: Orange/Red (1) and Orange/Black (1) are twisted pair cables

Executing Pulse Control Instructions in a Ladder Program

The pulse outputs are used by executing pulse control instructions in the ladder program.

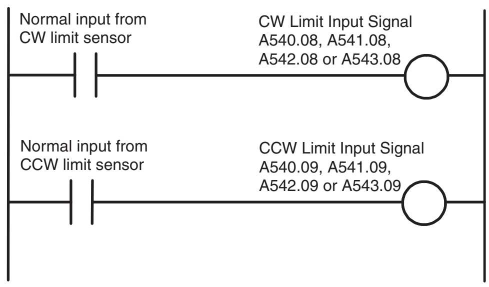
● Applicable Instructions

The following instructions are used.

Purpose		Overview	Instruction	Reference
Performing trapezoidal control		Performs trapezoidal pulse output control with independent acceleration and deceleration rates. (The number of pulses can be set.)	PLS2: PULSE OUTPUT	Refer to *12-2*
Jogging	Without acceleration and deceleration	Performs pulse output control without acceleration or deceleration.	SPED: SPEED OUTPUT	Refer to *12-3*
	With acceleration and deceleration	Performs trapezoidal pulse output control with the same acceleration and deceleration rates.	ACC: ACCELERATION CONTROL	
Performing interrupt feeding		Performs interrupt feeding with interrupt input and pulse output.	IFEED: INTERRUPT FEEDING	Refer to *12-4*
Positioning Linear Interpolation		Performs 2-axis to 4-axis linear interpolation.	ITPL: LINEAR INTERPOLATION	Refer to *12-5*
Performing origin searches		Actually moves the motor with pulse outputs and defines the machine origin based on the Origin Proximity Input and Origin Input signals.	ORG: ORIGIN SEARCH	Refer to *12-6-4*
Performing origin returns		Returns to the origin position from any position.	ORG: ORIGIN SEARCH	Refer to *12-6-6*
Changing or reading the pulse output PV		Changes the PV of the pulse output. (This operation defines the origin location.)	INI: MODE CONTROL	Refer to *12-6-7*
		Reads the PV of the pulse output.	PRV: HIGH-SPEED COUNTER PV READ	Refer to *12-7*

● Outputting to the Auxiliary Area Using the OUT Instruction

The OUT instruction in the ladder program is used to write signals received from the CW limit sensor and CCW limit sensor connected to normal inputs to the Auxiliary Area bits.



Bits Written in the Auxiliary Area

Auxiliary Area		Name	
Word	Bit		
A540	08	Pulse Output 0 CW Limit Input Signal	Signals must be received from external sensors connected to normal inputs and then written to the Auxiliary Area by the user program.
	09	Pulse Output 0 CCW Limit Input Signal	
A541	08	Pulse Output 1 CW Limit Input Signal	
	09	Pulse Output 1 CCW Limit Input Signal	
A542	08	Pulse Output 2 CW Limit Input Signal	
	09	Pulse Output 2 CCW Limit Input Signal	
A543	08	Pulse Output 3 CW Limit Input Signal	
	09	Pulse Output 3 CCW Limit Input Signal	

12-1-3 Specifications

Item	Specifications
Output mode	Continuous mode (for speed control) or independent mode (for position control)
Positioning (independent mode) instructions	PULS and SPED, PULS and ACC, or PLS2
Speed control (continuous mode) instructions	SPED or ACC
Origin (origin search and origin return) instructions	ORG
Interrupt feeding instructions	IFEED
Linear interpolation instructions	ITPL (N□□-type CPU Unit only)
Output frequency	1 Hz to 100 kHz (1 Hz units) S□□-type and N14/20 CPU Unit: 2 pulse outputs N30/40/60 CPU Unit: 4 pulse outputs
Frequency acceleration and deceleration rates	Set in increments of 1 Hz for acceleration/deceleration rates from 1 to 65,535 Hz (every 4 ms).
Changing SVs during instruction execution	The target frequency, acceleration/deceleration rate, and target position can be changed. ORG, IFEED and ITPL instructions cannot be changed.
Duty factor	Fixed at 50% The duty factor is less than 50% only when the output is less than 125 Hz by linear interpolation.
Pulse output method	Pulse + direction outputs (CW/CCW outputs cannot be used.)
Number of output pulses	Relative coordinates: 0000 0000 to 7FFF FFFF hex(Accelerating or decelerating in either direction: 2,147,483,647) Absolute coordinates: 8000 0000 to 7FFF FFFF hex(-2,147,483,648 to 2,147,483,647)
Pulse output PV's relative/absolute coordinate specifications	Absolute coordinates are specified automatically when the origin location has been defined by setting the pulse output PV with the INI instruction or performing an origin search with the ORG instruction. Relative coordinates are used when the origin location is undefined.
Relative pulse/absolute pulse specifications	The pulse type can be specified with an operand in the PULS or PLS2 instruction. Only absolute pulse can be specified with linear interpolation (ITPL) instruction. **Note** The absolute pulse specification can be used when absolute coordinates are specified for the pulse output PV, i.e. the origin location has been defined.The absolute pulse specification cannot be used when relative coordinates are specified, i.e. the origin location is undefined. An instruction error will occur.
Pulse output PV's storage location	The following Auxiliary Area words contain the pulse output PVs Pulse output 0: A277 (leftmost 4 digits) and A276 (rightmost 4 digits) Pulse output 1: A279 (leftmost 4 digits) and A278 (rightmost 4 digits) Pulse output 2: A53 (leftmost 4 digits) and A52 (rightmost 4 digits) Pulse output 3: A55 (leftmost 4 digits) and A54 (rightmost 4 digits) The PVs are refreshed during regular I/O refreshing.

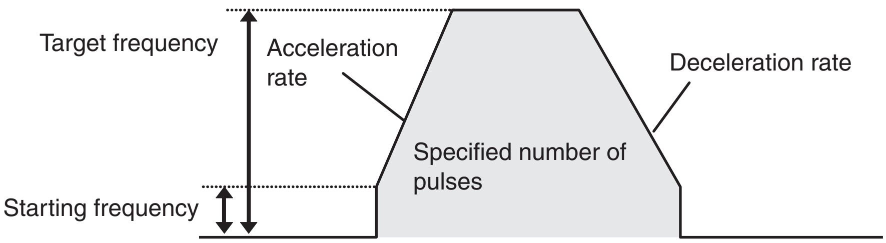
12-2 Positioning Control

This section describes how to use pulse outputs with trapezoidal acceleration and deceleration when using the PLS2 instruction.

12-2-1 Positioning Control Configuration

If the target frequency, starting frequency, acceleration and deceleration rate, direction are set beforehand, the following time chart will perform trapezoidal positioning control.

Specify the output waveform in the instruction operands.



Target frequency	1 Hz to 100 kHz (in increments of 1 Hz)
Starting frequency	0 Hz to 100 kHz (in increments of 1 Hz)
Acceleration rate	Set in increments of 1 Hz from 1 to 65,535 Hz (every 4 ms).
Deceleration rate	Set in increments of 1 Hz from 1 to 65,535 Hz (every 4 ms).
Direction specification	Set to CW or CCW.
Specified number of pulses	Relative coordinates: 0000 0000 to 7FFF FFFF hex (Incrementing and decrementing in each direction: 2,147,483,647) Absolute coordinates: 8000 0000 to 7FFF FFFF hex (-2,147,483,648 to 2,147,483,647)

12-2-2 Relative Positioning and Absolute Positioning

● Selecting Relative or Absolute Coordinates

The pulse output PV's coordinate system (absolute or relative) is selected automatically, as follows:

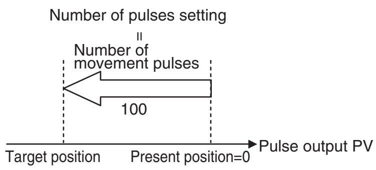
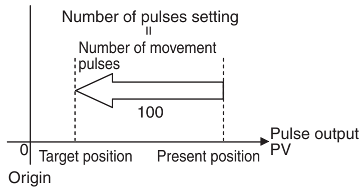
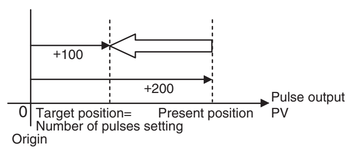
- When the origin is undefined, the system operates in relative coordinates.
- When the origin has been defined, the system operates in absolute coordinates.

Conditions	Origin has been defined by an origin search	Origin has been defined by executing the INI instruction to change the PV	Origin undefined (Origin search has not been performed and PV has not been changed with the INI instruction.)
Pulse output PV's coordinate system	Absolute coordinates		Relative coordinates

Refer to *12-6-1 Origin Searches* for details.

● 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 the PULS or PLS2 instruction is executed.

Pulse output specified in PULS or PLS2	Relative coordinate system	Absolute coordinate system
	Origin undefined: The No-origin Flag will be ON.	**Origin defined: The No-origin Flag will be OFF.**
Relative pulse specification	Positions the system to another position relative to the present position. Number of movement pulses = Number of pulses setting	
	The pulse output PV after instruction execution = Number of movement pulses = Number of pulses setting The pulse output PV is reset to 0 just before pulses are output. After that, the specified number of pulses is output. The following example shows the number of pulses setting = 100 counterclockwise.  Pulse output PV range: 8000 0000 to 7FFF FFFF hex Number of pulses setting range: 0000 0000 to 7FFF FFFF hex	The pulse output PV after instruction execution = PV + Number of movement pulses. The following example shows the number of pulses setting = 100 counterclockwise.  Pulse output PV range: 8000 0000 to 7FFF FFFF hex Number of pulses setting range: 0000 0000 to 7FFF FFFF hex
Absolute pulse specification	The absolute pulse specification cannot be used when the origin location is undefined, 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 present position (pulse output PV) and target position. The following example shows the number of pulses setting = +100.  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: 8000 0000 to 7FFF FFFF hex Number of pulses setting range: 8000 0000 to 7FFF FFFF hex

Precautions for Correct Use

The absolute pulse cannot be specified with the origin undefined. Please specify them when the origin is defined by performing the origin searches.

Additional Information

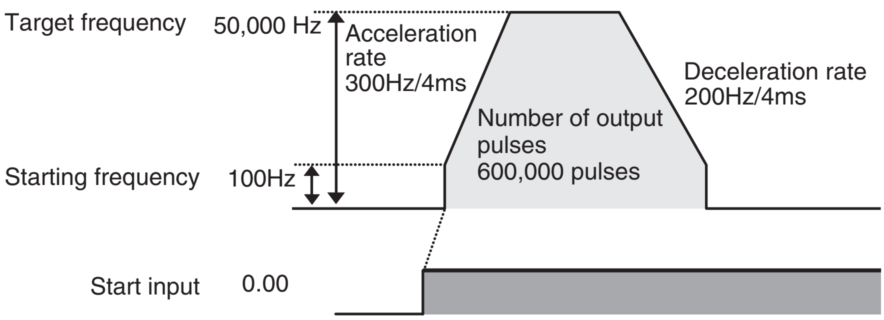
The origin position is undefined in the following case. Please define the origin position by performing the origin searches again.

- When the pulse output reset flag is turned ON
- When the RUN or MONITOR mode is changed to the PROGRAM mode

12-2-3 Application Example

Specifications and Operation

When the start input (CIO 0.00) goes ON, this example program outputs 600,000 pulses from pulse output 1 to turn the motor.



Applicable Instructions

PLS2

Preparations

● PLC Setup

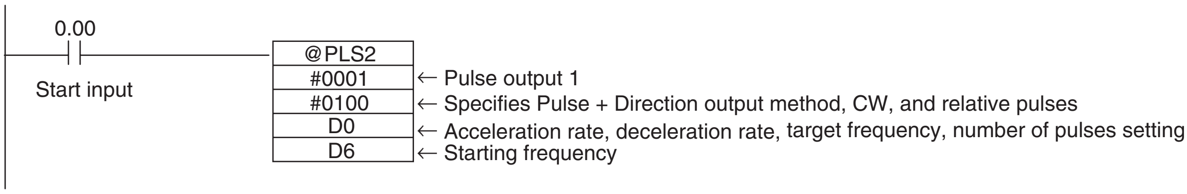
There are no settings that need to be made in the PLC Setup.

● DM Area Settings

- Settings for PLS2 Instruction (D0 to D7)

Setting	Address	Data
Acceleration rate: 300 Hz/4 ms	D0	#012C
Deceleration rate: 200 Hz/4 ms	D1	#00C8
Target frequency: 50,000 Hz	D2	#C350
	D3	#0000
Number of output pulses: 600,000 pulses	D4	#27C0
	D5	#0009
Starting frequency: 100 Hz	D6	#0064
	D7	#0000

Ladder Program



Additional Information

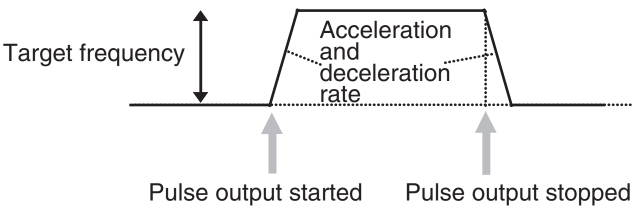
- Absolute pulses can be specified when the origin position has been defined.
- If a target frequency that cannot be reached has been set, the target frequency will be reduced automatically, i.e., triangular control will be performed. In some cases where the acceleration rate is substantially greater than the deceleration rate, the operation will not be true triangular control. The motor will be operated at a constant speed for a short time between the acceleration and deceleration.

12-3 Jogging

Jogging can be performed by using the SPED (SPEED OUTPUT) and ACC (ACCELERATION CONTROL) instructions. This section describes the steps for jogging.

12-3-1 High-speed Jogging

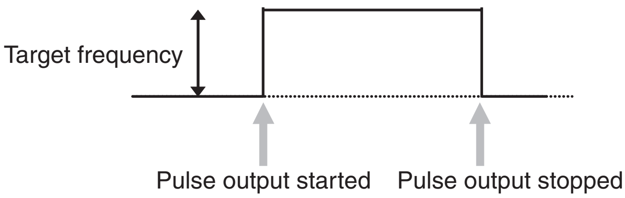
Start pulse output with acceleration or deceleration using the ACC instruction. In this case, acceleration and deceleration rate must be the same. Set the target frequency of the ACC instruction to 0 Hz to stop the pulse output.



Target frequency	Starting pulse output: 1 Hz to 100 kHz (in increments of 1 Hz) Stopping pulse output: 0 Hz
Acceleration and deceleration rate	Set in increments of 1 Hz from 1 to 65,535 Hz (every 4 ms).
Direction specification	Set to CW or CCW.
Mode specification	Set to continuous mode.

12-3-2 Low-speed Jogging

Start pulse output without acceleration or deceleration using the SPED instruction. Set the target frequency of the SPED instruction to 0 Hz to stop the pulse output.



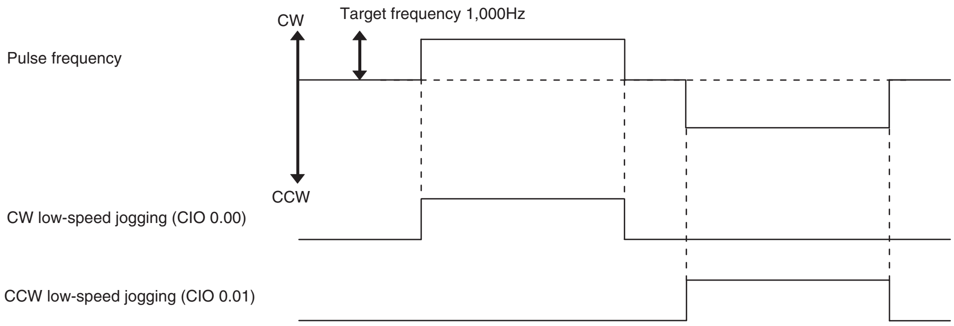
Target frequency	Starting pulse output: 1 Hz to 100 kHz (in increments of 1 Hz) Stopping pulse output: 0 Hz
Direction specification	Set to CW or CCW.
Mode specification	Set to continuous mode.

12-3-3 Application Example

Specifications and Operation

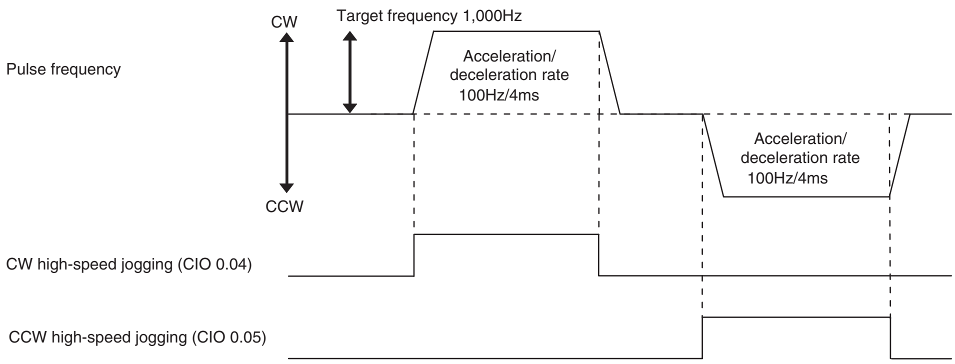
The following example shows jogging without acceleration or deceleration executed using a SPED instruction. It is used for low-speed jogging.

- Clockwise low-speed jogging will be executed from pulse output 1 while CIO 0.00 is ON.
- Counterclockwise low-speed jogging will be executed from pulse output 1 while CIO 0.01 is ON.



The example shows jogging with acceleration and deceleration executed using an ACC instruction. It is used for high-speed jogging.

- Clockwise high-speed jogging will be executed from pulse output 1 while CIO 0.04 is ON.
- Counterclockwise high-speed jogging will be executed from pulse output 1 while CIO 0.05 is ON.



Preparations

● PLC Setup

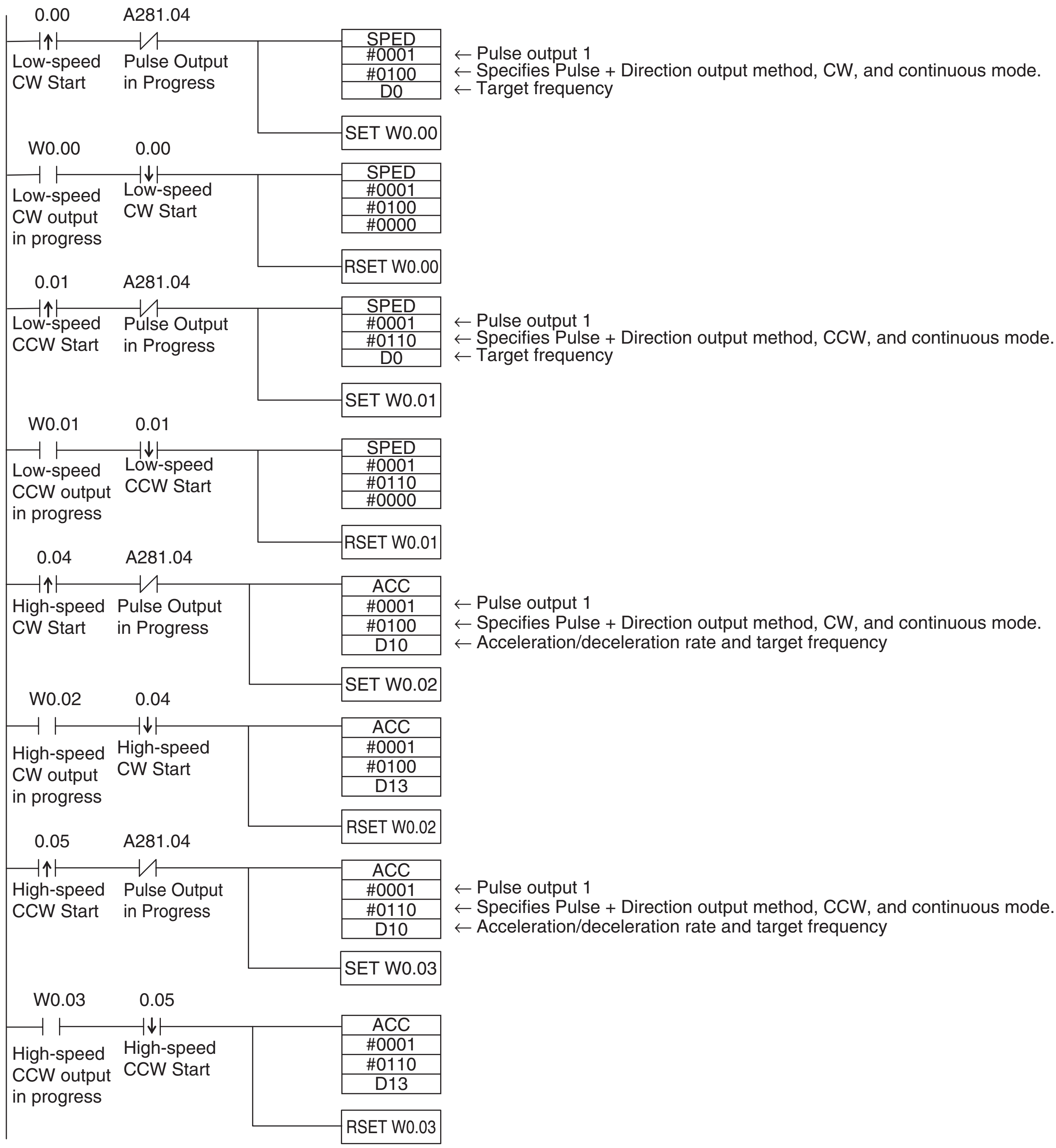
There are no settings that need to be made in the PLC Setup.

● DM Area Settings

- Settings to Control Speed while Jogging (D0 to D1 and D10 to D15)

Setting	Address	Data
Target frequency (low speed): 1,000 Hz	D0	#03E8
	D1	#0000
Acceleration rate: 100 Hz/4 ms	D10	#0064
Target frequency (high speed): 100,000 Hz	D11	#86A0
	D12	#0001
Acceleration/deceleration rate: 100 Hz/4 ms (Not used.)	D13	#0064
Target frequency (stop): 0 Hz	D14	#0000
	D15	#0000

Ladder Program



Additional Information

The PLS2 instruction can be used to set a starting frequency or separate acceleration and deceleration rates, but there are limitations on the operating range because the end point must be specified in the PLS2 instruction.

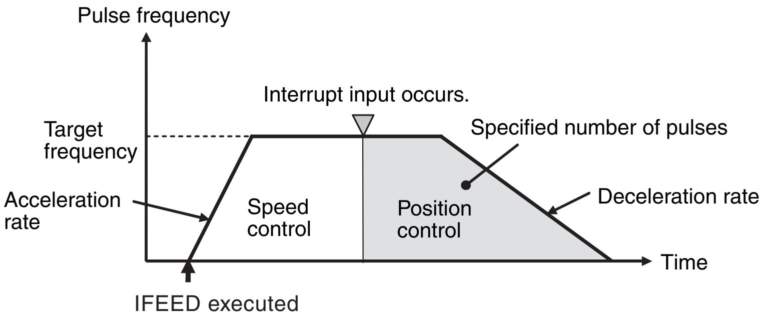
12-4 Implementing Interrupt Feeding

This section describes how to use interrupt feeding when using the IFEED instruction.

12-4-1 Interrupt Feeding

Interrupt feeding is performed with the IFEED instruction. IFEED controls interrupt feeding by combining the specified pulse output and interrupt input. An interrupt input is used as a trigger during speed control to switch to position control and then move a specified amount before decelerating to a stop.

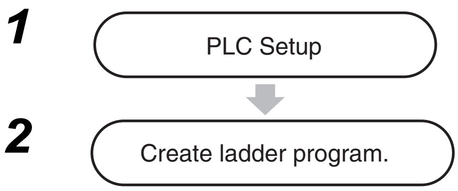
An interrupt task is not necessary, so no delays are caused by the interrupt startup time or the occurrence of other interrupts. The accuracy of feeding after an interrupt input occurs can therefore be improved.



Additional Information

- Only specific pulse outputs and interrupt inputs can be used together.
  If you want to pair any other pulse outputs and interrupt inputs, or if you want to change settings during pulse output, use the ACC and PLS2 instructions together.
  If the ACC and PLS2 instructions are used, delays will occur for the interrupt startup time and possibly for other interrupts.
- The IFEED instruction cannot be executed when the pulse output port specified by the axis specifier of the IFEED instruction is already outputting pulses by the SPED, ACC, PLS2, ORG, IFEED and ITPL instructions. P_ER flag turns ON.

12-4-2 Flow of Operation

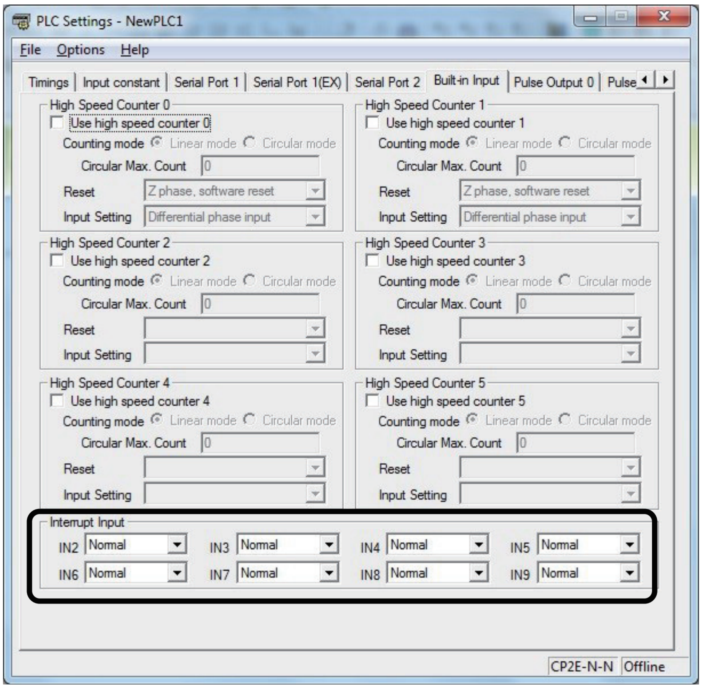


- Select *Interrupt Input* in the Interrupt Input Dialog Box that is accessed from the Built-in Input Tab Page of the PLC Setup using the CX-Programmer.
- Set whether to start the interrupt on OFF transitions or ON transitions in the input using the MSKS instruction.
- Set pulse output ports 0 to 3, output mode, output direction, acceleration/deceleration rate, target frequency, and number of output pulses.
- Execute the IFEED instruction.

12-4-3 Application Example

Pulse output 0 and interrupt input 6 are used.

PLC Setup

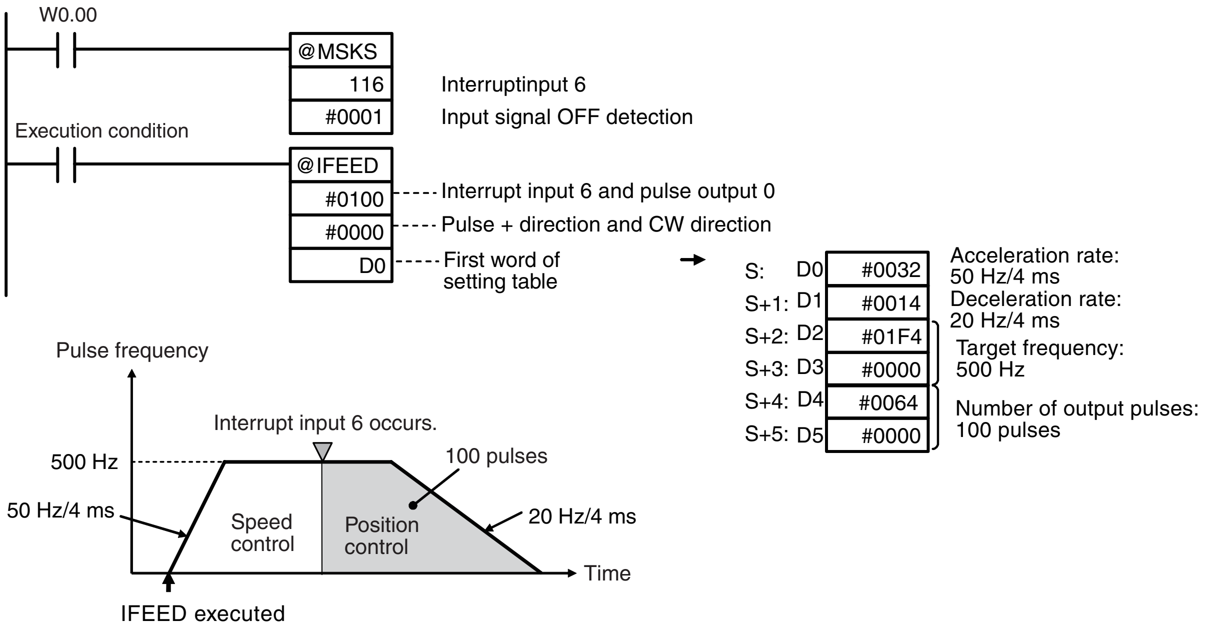


Pulse Ouptput	Interrupt Input
Pulse Ouptput 0	IN6
Pulse Ouptput 1	IN7
Pulse Ouptput 2*	IN8
Pulse Ouptput 3*	IN9

* Pulse output 2 and pulse output 3 can be used with N30/40/60 CPU unit.

Ladder Program

The IFEED instruction is executed after turning interrupt input 6 to OFF with the MSKS instruction.



Precautions for Correct Use

- Before executing the IFEED instruction, use the MSKS instruction to disable the specified interrupt if it is currently not masked. An instruction error will occur if the IFEED instruction is executed when the interrupt is not masked.
- Interrupt inputs 6, 7, 8, and 9 are used with the IFEED instruction. The terminals used for interrupt inputs 6, 7, 8 and 9 are also used for the origin and origin proximity inputs for pulse outputs. If the IFEED instruction is used for pulse output, do not use the origin search function.

● Checking Status during Interrupt Feeding

The interrupt feeding status can be read from the following bits.

Name	Pulse output 0	Pulse output 1	Pulse output 2	Pulse output 3	Refresh timing
Interrupt Feeding In-progress Flag	A280.08	A281.08	A56.08	A57.08	• Cleared when power is turned ON. • Cleared when starting/stopping operation. • Cleared after completing interrupt feeding. • Turned ON when interrupt input is received after starting pulse output with IFEED instruction
Interrupt Feeding Error Flag	A280.09	A281.09	A56.09	A57.09	• Cleared when power is turned ON. • Cleared when operation starts. • Cleared when IFEED instruction processing is started. • Turned ON if an overflow or underflow occurs when an interrupt input is received, or if an overflow or underflow occurs while the specified number of pulses is being moved, after operation is started with the IFEED instruction with the origin defined.

12-5 Positioning Linear Interpolation

This section describes how to position linear interpolation when using the ITPL instruction.

Linear interpolation can be used only with the CP2E N☐☐-type CPU Unit with transistor outputs.

12-5-1 Positioning Linear Interpolation

Linear interpolation positioning is performed with the ITPL instruction. The ITPL instruction executes positioning by using linear interpolation operation to the specified postion.

Positioning by linear interpolation enables higher-speed positioning than single-axis positioning.

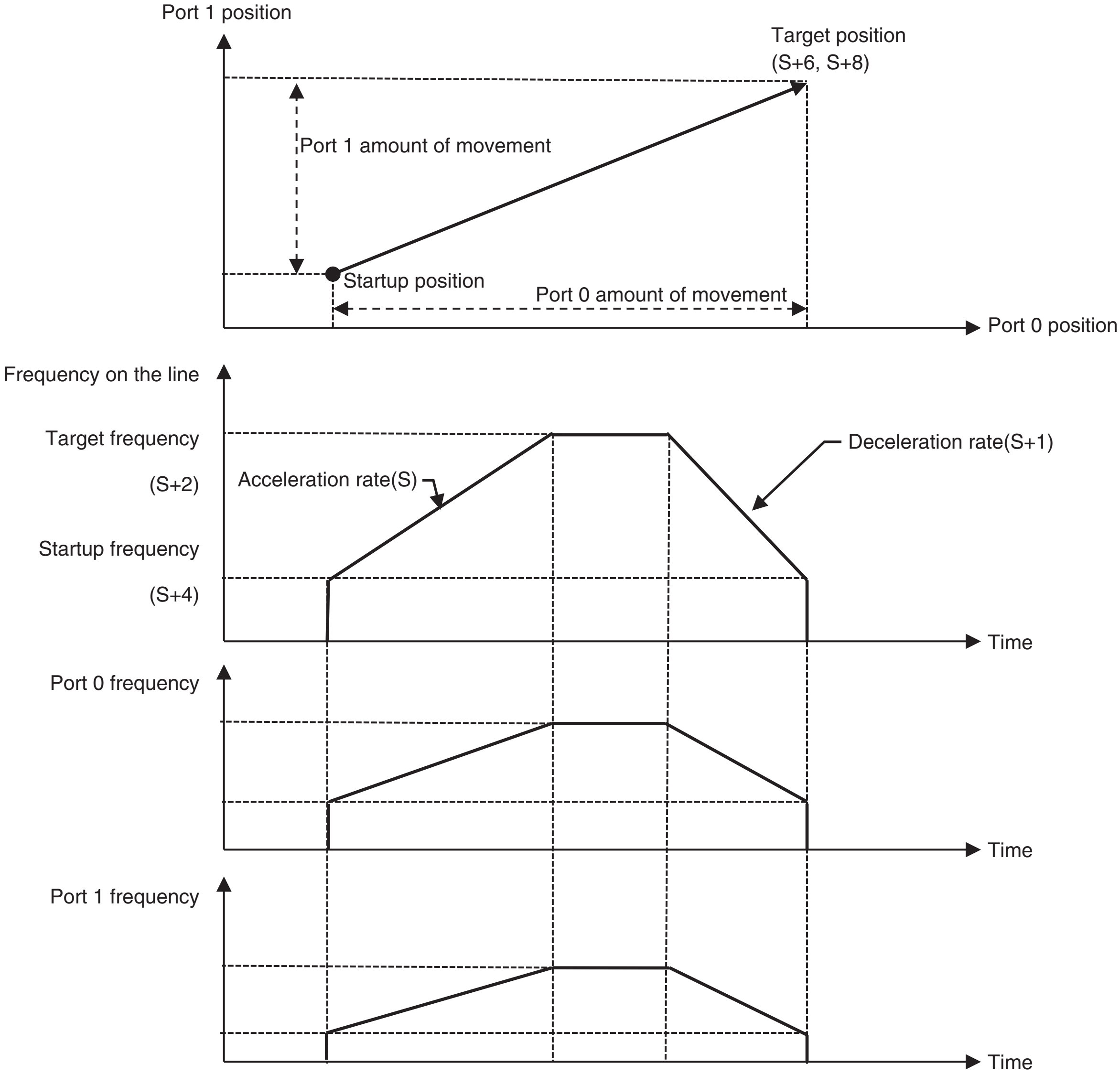
- Two linear interpolations can be used for an N14/20 CPU Unit, and up to four linear interpolations can be used for an N30/40/60 CPU Unit.
- Linear interpolation can be used with absolute coordinates, it cannot be used with relative coordinates.
  Execute the ITPL instruction after the origin is defined.

	Two-axis interpolation	Three-axis interpolation	Four-axis interpolation
Linear interpolation 0	Pulse output 0, 1	Pulse output 0, 1, 2	Pulse output 0, 1, 2,3
Linear interpolation 1	Pulse output 2, 3	-	-

12-5-2 Positioning Linear Interpolation Configuration

The target frequency, starting frequency, acceleration and deceleration rate, the number of output pulses are set beforehand, and linear interpolation positioning control is performed by executing the instruction.

The following example shows the two-axis linear interpolation.



- The target frequency and starting frequency set the frequency of linear interpolation positioning. The frequency and travel distance of each axis are automatically set internally.
- The stop during linear interpolation can be immediate stop or deceleration stop.
  In the case of a deceleration stop, it stops on a linear interpolation line. Please execute the linear interpolation instruction again to restart.

Precautions for Correct Use

The linear interpolation positioning can not be performed with the origin undefined. Please specify them when the origin is defined by performing the origin searches.

If the output speed is less than 125 Hz, pulse output cannot be performed because the number of pulse per cycle is less than one pulse. As a result, the pulse duty factor may not be 50%.

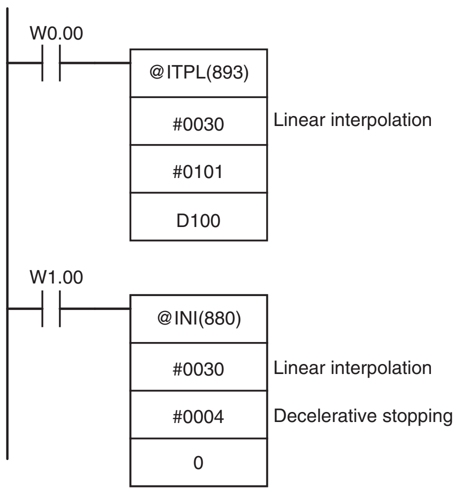
Additional Information

- The ITPL instruction can not be executed when the pulse output port specified by the axis specifier of the ITPL instruction is already outputting pulses by the SPED, ACC, PLS2, ORG, IFEED and ITPL instructions. P_ER flag turns ON.
- The distance and the operation time that can be moved by linear interpolation have an upper limit. Please specify the linear distance to 100,000,000 pulses or less, and the operation time to 1,000 seconds or less.
- Set the starting frequency ≤ target frequency. If the starting frequency > target frequency, it operates at the target frequency.

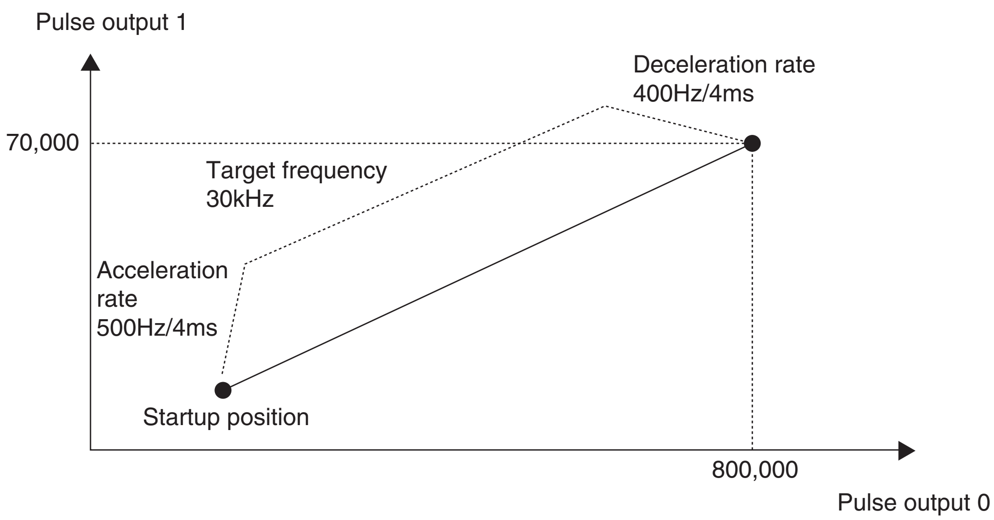
12-5-3 Application Example

The ITPL instruction uses pulse output 0 and pulse output 1 to perform linear interpolation positioning. When W1.00 is turned ON, the linear interpolation will decelerate to stop.

Note The origin must be defined before executing the ITPL instruction.



D100	01F4	Acceleration rate: 500Hz/4ms
D101	0190	Deceleration rate: 400Hz/4ms
D102	7530	Target frequency: 30kHz
D103	0000	
D104	03E8	Startup frequency: 1000Hz
D105	0000	
D106	3500	Target position 0: 800,000
D107	000C	
D108	1170	Target position 1: 70,000
D109	0001	



Related Auxiliary Area Flags

There is no special related auxiliary area flags for linear interpolation positioning. The actions of the pulse output related auxiliary area flags during the linear interpolation positioning are as follows.

Name	Action
Pulse Output PV Storage Words	Current value is stored.
Pulse Output Reser Bit	During linear interpolation, the pulse output PV cannot be cleared even if the reset flag is turned ON.
CW Limit Input Signal Flag	Valid only when *Limit Input Signal Operation* in PLC Setting is *Always*.
CCW Limit Input Signal Flag	Valid only when *Limit Input Signal Operation* in PLC Setting is *Always*.
Accel/Decel Flag	ON when the linear interpolation is accelerating or decelerating. All the pulse outputs specified by the linear interpolation instruction operate in the same way. It still operates when the linear interpolation movement length is 0.
Output Completed Flag	ON when the linear interpolation is completed. All the pulse outputs specified by the linear interpolation instruction operate in the same way. It still operates when the linear interpolation movement length is 0.
Output In-progress Flag	ON during the linear interpolation. All the pulse outputs specified by the linear interpolation instruction operate in the same way. It still operates when the linear interpolation movement length is 0.

12-6 Defining Origin Position

The CP2E CPU Units have two methods that can be used to define the origin position.

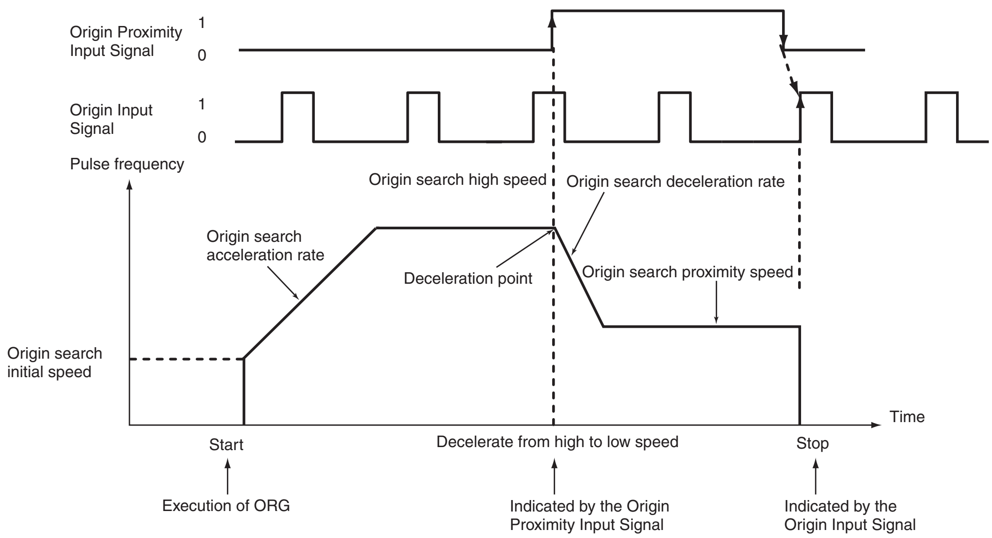
- Origin Search
  The ORG instruction outputs pulses to turn the motor according to the pattern specified in the origin search parameters. As the motor turns, the origin search function defines the origin from the following three kinds of position input signals.
  - Origin input signal
  - Origin proximity input signal
  - CW limit input signal and CCW limit input signal
- Changing the Pulse Output PV
  When setting the current position as the origin, execute INI to reset the pulse output PV to 0.

12-6-1 Origin Searches

When the ORG instruction executes an origin search, it outputs pulses to actually move the motor and defines 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.

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.



(Example for reversal mode 1 and method 0 (described later))

Additional Information

The motor can be moved even if the origin position has not been defined, 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 present position to 0.

12-6-2 Flow of Operation

1 PLC Setup

- Set the origin search parameters in the Pulse Output 0 to 3 Tab Pages of the PLC Setup using the CX-Programmer.

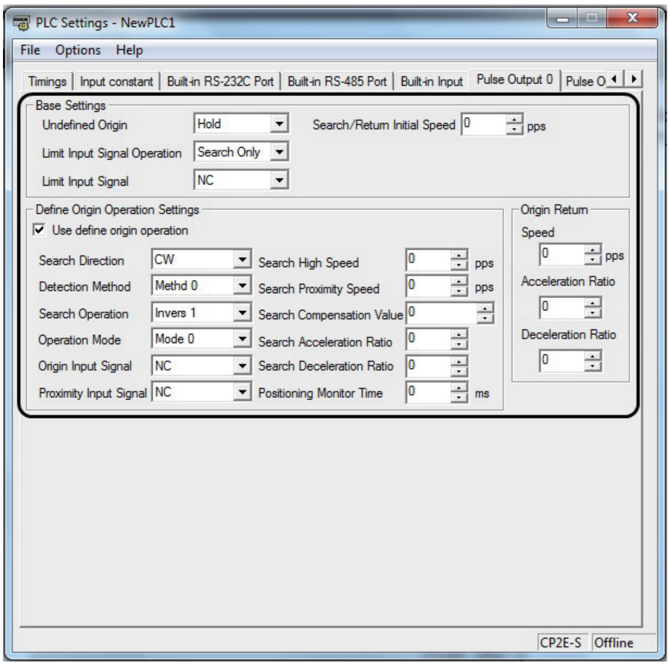
↓

2 Ladder program | Cyclic task, interrupt task

- Set pulse output 0 to 3.
- Output the status of the Limit Signal Inputs and Positioning Completed Signal to Auxiliary Area bits.
- Execute ORG. Specify an origin search.

12-6-3 Settings in PLC Setup

To perform an origin search or to use a Limit Input Signal as an input to a function other than origin search, set the parameters on the Pulse Output 0 to 3 Tab Pages in the PLC Setup.



Pulse Output 0 to 3 Tab Page

Item		Selection	Description
Base Settings	**Undefined Origin**	Hold	When a Limit Input Signal is input, the pulse output is stopped and the previous status is held.
		Undefined	When a Limit Input Signal is input, the pulse output is stopped and origin becomes undefined.
	Limit Input Signal Operation	Search Only	The CW/CCW Limit Input Signal is used for origin searches only.
		Always	The CW/CCW Limit Input Signal is used by functions other than origin search.
	Limit Input Signal	NC	Select when using NC contact (b contact) for the Limit Input Signal.
		NO	Select when using NO contact (a contact) for the Limit Input Signal.
	Search/ Return Initial Speed	Set the motor's starting speed when performing an origin search or origin return. Specified in units of pulses per second (pps). Setting range: 0 to 100k pps The origin search will not be performed in these cases: Origin search high speed ≤ Origin search proximity speed. Origin search proximity speed ≤ Origin search initial speed.	
Define Origin Operation Settings	**Use define origin operation**	Select this check box to use origin searches.	
	Search Direction	Set the direction for detecting the Origin Input Signal. An origin search is performed so that the Origin Input Signal's rising edge is detected when moving in the origin search direction.	
		CW	Performs origin search in the clockwise direction.
		CCW	Performs origin search in the counterclockwise direction.
	Detection Method	Set one of the following three methods to determine the parameters related to the Origin Proximity Input Signal.	
		Method 0	The direction is reversed at the Origin Proximity Input Signal. The Origin Input Signal is accepted after the Origin Proximity Input Signal turns ON and then OFF.
		Method 1	The direction is not reversed at the Origin Proximity Input Signal. The Origin Input Signal is accepted after the Origin Proximity Input Signal turns ON.
		Method 2	The Origin Proximity Input Signal is not used. The Origin Input Signal is accepted without using the Origin Proximity Input Signal. Only origin search proximity speed can be the origin search speed.
	Search Operation	Select one of the following two modes for the origin search operation pattern.	
		Inverse 1	The direction is reversed when the Limit Input Signal is received while moving in the origin search direction.
		Inverse 2	An error is generated and operation is stopped if the Limit Input Signal is received while moving in the origin search direction.
	Operation Mode	This parameter determines the I/O signals that are used for origin search.	
		Mode 0	Use when connecting to a stepping motor that does not have a Positioning Completed Signal.
		Mode 1	In this mode, the Positioning Completed Signal from the Servo Drive is not used. Use this mode when you want to reduce the processing time.
		Mode 2	In this mode, the Positioning Completed Signal from the Servo Drive is used. Use this mode when you want high positioning accuracy.
	Origin Input Signal	Specifies the type of Origin Input Signal (NC or NO).	
		NC	Sets a normally closed (b contact) Origin Input Signal.
		NO	Sets a normally open (a contact) Origin Input Signal.
	Proximity Input Signal	Specifies the type of Origin Proximity Input Signal (NC or NO).	
		NC	Sets a normally closed (b contact) Origin Proximity Input Signal.
		NO	Sets a normally open (a contact) Origin Proximity Input Signal.

Item		Selection	Description
Define Origin Opera-tion Settings	**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). Setting range: 1 to 100k pps The origin search will not be performed in these cases: Origin search high speed ≤ Origin search proximity speed. Origin search proximity speed ≤ Origin search initial speed.	
	Search Prox-imity 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). Setting range: 1 to 100k pps The origin search will not be performed in these cases: Origin search high speed ≤ Origin search proximity speed. Origin search proximity speed ≤ Origin search initial speed.	
	Search Com-pensation Value	After the origin has been defined, the origin compensation can be set to compensate for a shift in the Proximity Sensor's ON position, motor replacement, or other change. Setting range: -2,147,483,648 to 2,147,483,647 pulses Once the origin has been detected in an origin search, the number of pulses specified in the origin compensation is output, the present position is reset to 0, and the pulse output's No-origin Flag is turned OFF.	
	Search Acceleration Ratio	Sets the motor's acceleration rate when the origin search is executed. Specify the amount to increase the speed (Hz) per 4-ms interval.	Setting range: 1 to 65,535 Hz/ 4 ms
	Search Deceleration Ratio	Sets the motor's deceleration rate when the origin search function is decelerating. Specify the amount to decrease the speed (Hz) per 4-ms interval.	Setting range: 1 to 65,535 Hz/ 4 ms
	Positioning Monitor Time	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 com-pleted. A Positioning Timeout Error (error code 0300) will be generated if the motor driver's Positioning Com-pleted Signal does not come ON within the specified time.	Setting range: 0 to 9,999 ms*
Origin Return	**Speed**	Sets the motor's target speed when the origin return is executed. Specify the speed in the number of pulses per second (pps).	Setting range: 1 to 100k pps
	Acceleration Ratio	Sets the motor's acceleration rate when the origin return operation starts. Specify the amount to increase the speed (Hz) per 4-ms interval.	Setting range: 1 to 65,535 Hz/ 4 ms
	Deceleration Ratio	Sets the motor's deceleration rate when the origin return function is decelerating. Specify the amount to decrease the speed (Hz) per 4-ms interval.	Setting range: 1 to 65,535 Hz/ 4 ms

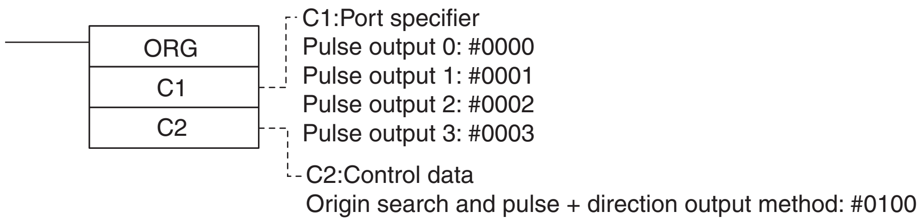
* The actual monitoring time will be the Positioning Monitor Time rounded up to the nearest 10-ms unit + 10 ms max. If the Positioning Monitor Time is set to 0, the function will be disabled and the Unit will continue waiting for the Positioning Completed Signal to come ON. (A Positioning Timeout Error will not be generated.)

Note The power supply must be restarted after the PLC Setup is transferred in order to enable the settings for using the origin search.

12-6-4 Origin Search Instructions

Origin Search Instruction: ORG

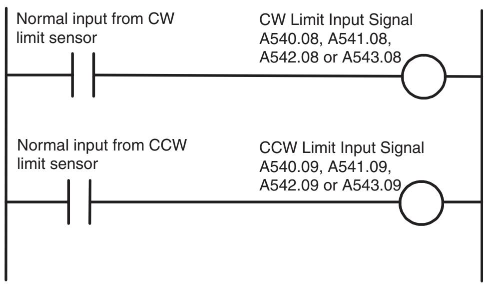
Execute the ORG instruction in the ladder program to perform an origin search with the specified parameters.



Precautions for Correct Use

Limit Sensor Application

Create a program that can identify the limit sensor when using the origin search.
The OUT instruction is used in the ladder program to write signals received from the CW limit sensor and CCW limit sensor connected to normal inputs to the Auxiliary Area bits.



Bits Written in the Auxiliary Area

Auxiliary Area		Name	
Word	Bit		
A540	08	Pulse Output 0 CW Limit Input Signal	Signals received from external sensors connected to normal inputs must be written to the Auxiliary Area bits in the user program.
	09	Pulse Output 0 CCW Limit Input Signal	
A541	08	Pulse Output 1 CW Limit Input Signal	
	09	Pulse Output 1 CCW Limit Input Signal	
A542	08	Pulse Output 2 CW Limit Input Signal	
	09	Pulse Output 2 CCW Limit Input Signal	
A543	08	Pulse Output 3 CW Limit Input Signal	
	09	Pulse Output 3 CCW Limit Input Signal	

12-6-5 Origin Search Operations

Operating Mode

The operating mode parameter specifies the kind of I/O signals that are used in the origin search.

<table>
<tr><th colspan="2">I/O signal</th><th>Mode 0</th><th>Mode 1</th><th>Mode 2</th></tr>
<tr><th colspan="2">Driver</th><th>Stepping motor*</th><th colspan="2">Servomotor</th></tr>
<tr><td rowspan="2">Operation</td><td>Origin Input Signal</td><td>Inputs signals are arranged so deceleration starts when the Origin Proximity Input Signal is received and then the Origin Input Signal is received while the motor is decelerating to the origin search proximity speed. If an Origin Input Signal is detected during this deceleration, an Origin Input Signal error will occur and the motor will decelerate to a stop.</td><td colspan="2">Even if an Origin Input Signal is received during deceleration, it is ignored. After the motor has reached the origin search proximity speed and the Origin Input Signal is received, the motor stops, completing the origin search process.</td></tr>
<tr><td>Positioning Completed Signal</td><td>The Positioning Completed Signal from the driver is not connected. *</td><td>The Positioning Completed Signal from the driver is not connected.
Use this mode when you want to reduce the processing time, even at the expense of positioning accuracy.</td><td>After detecting the origin, the origin search process is not completed until the Positioning Completed Signal is received.
Use this mode when you want high positioning accuracy.</td></tr>
</table>

* There are stepping motor drivers that are equipped with a Positioning Completed Signal like a servomotor. Operating modes 1 and 2 can be used with these stepping motor drivers.

The use of an error counter reset output and positioning completed input depends on the mode as described in the following table.

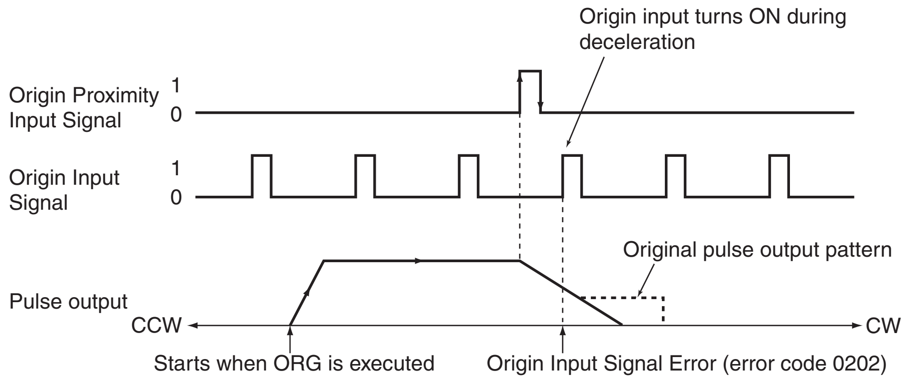
I/O signal	Mode 0	Mode 1	Mode 2
Origin Input Signal	Connected to the open-collector output from a sensor or other device.	Connected to the phase-Z signal from the Servo Drive.	Connected to the phase-Z signal from the Servo Drive.
Error counter reset output	Not used. (The origin search operation is completed when the origin is detected.)	Connected to the error counter reset of the Servo Drive.	Connected to the error counter reset of the Servo Drive.
Positioning completed input	Not used.	Not used.	Connected to the Positioning Completed Signal from the Servo Drive.

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 NO contacts.

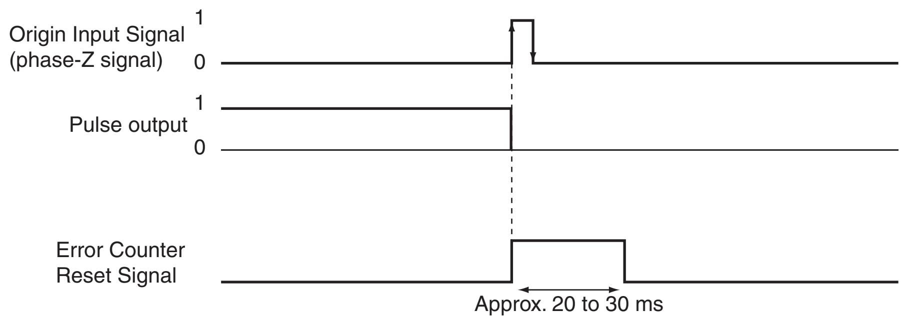
When the Origin Proximity Input Signal is received, the motor will begin decelerating from the origin search high speed to the origin search proximity speed. In this operating mode, the Origin Input Signal will be detected if it is received during this deceleration and an Origin Input Signal Error (error 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 Drive to the Origin Input Signal.

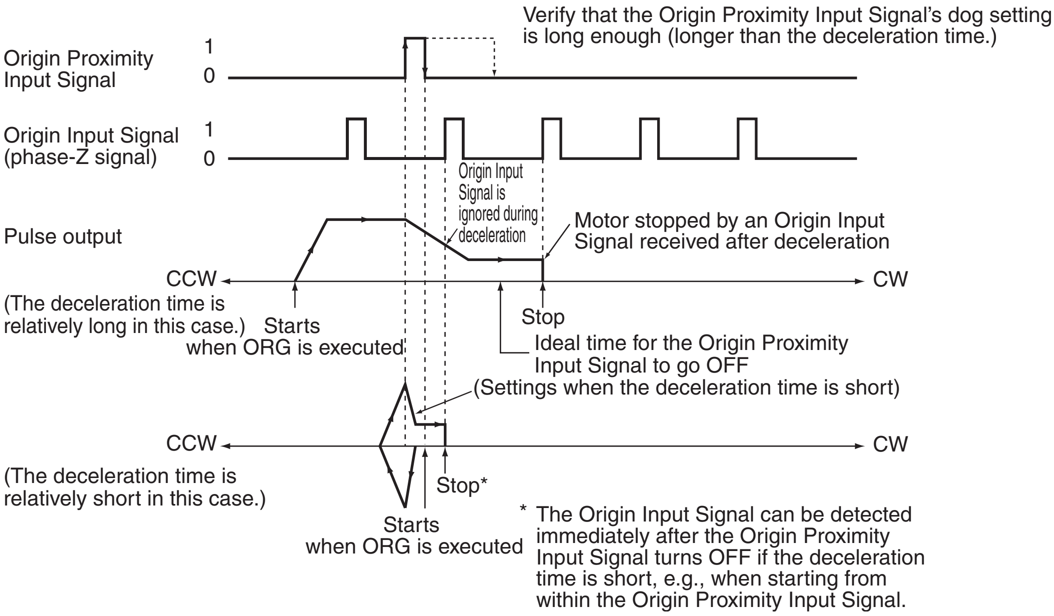
When the Origin Input Signal is received, the pulse output will be stopped and the Error Counter Reset Signal will be output for about 20 to 30 ms.



Though the Origin Proximity Input Signal is received, the signal will be ignored and 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.

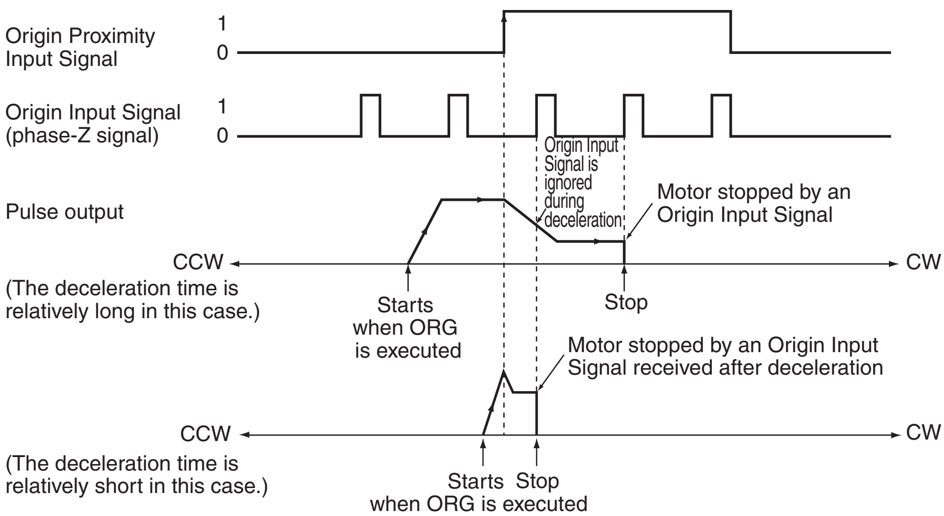
Operating Mode 1 with Origin Proximity Input Signal Reverse (Origin Detection Method Setting = 0)

The Origin Input Signal can be detected immediately after the Origin Proximity Input Signal turns OFF if the deceleration time is short, e.g., when starting from within the Origin Proximity Input Signal. Set an Origin Proximity Input Signal dog setting that is long enough (longer than the deceleration time.)



Operating Mode 1 without Origin Proximity Input Signal Reverse (Origin Detection Method Setting = 1)

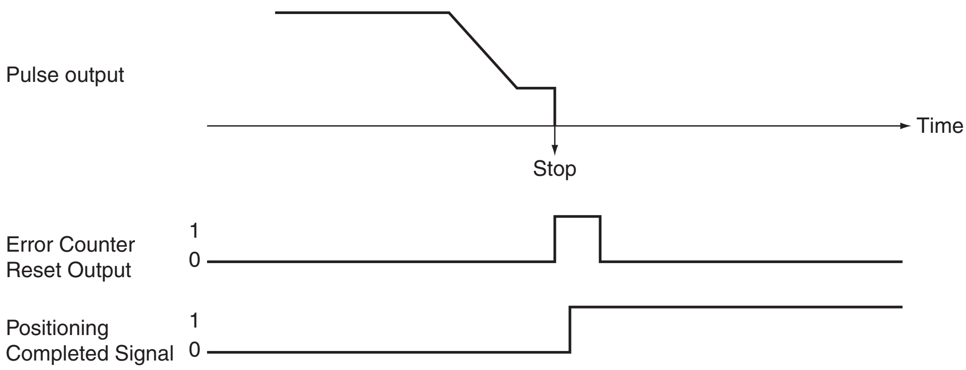
Depending on the length of the deceleration time, the stopping position may change when the Origin Input Signal is detected during deceleration.



● Operating Mode 2 (with Error Counter Reset Output, with Positioning Completed Input)

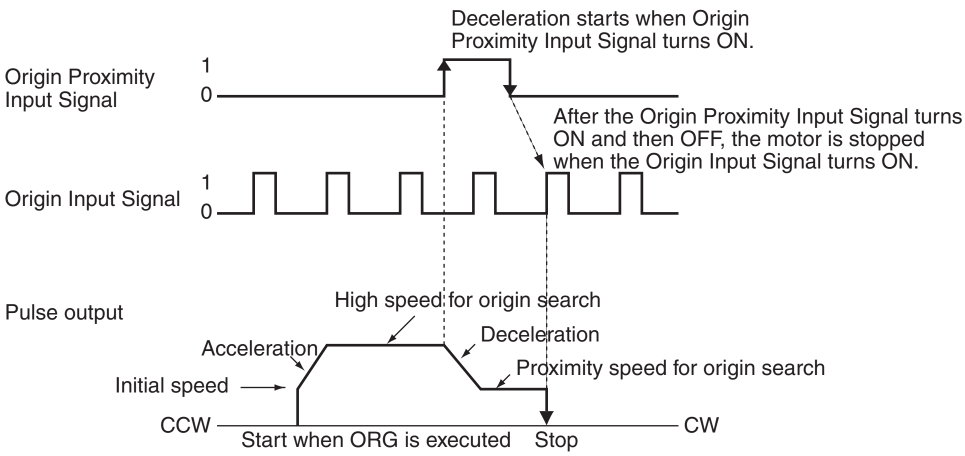
This operating mode is the same as mode 1, except the Positioning Completed Signal (INP) from the Servo Drive is used. Connect the Positioning Completed Signal from the Servo Drive to a normal input.

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.

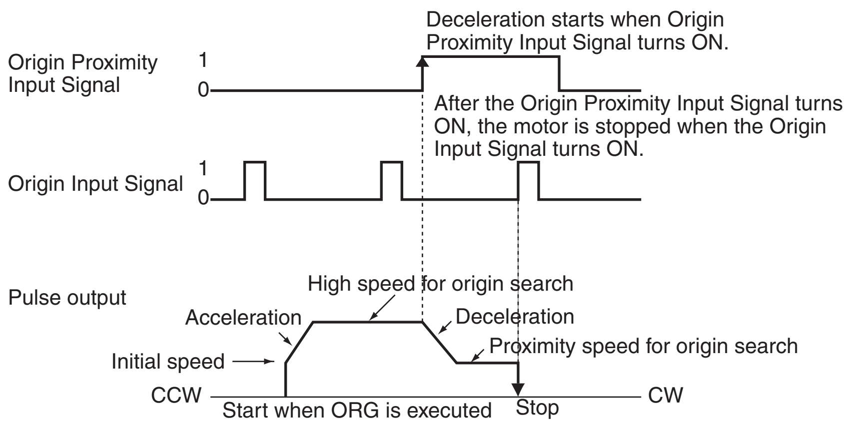


Origin Detection Method Setting

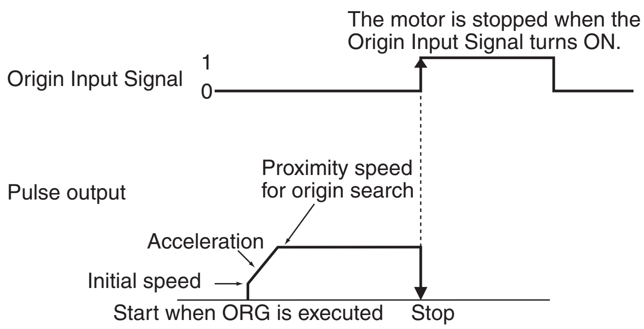
● Origin Detection Method 0: Origin Proximity Input Signal Reversal Required (Recommended Method)



- **Origin Detection Method 1: Origin Proximity Input Signal Reversal Not Required**



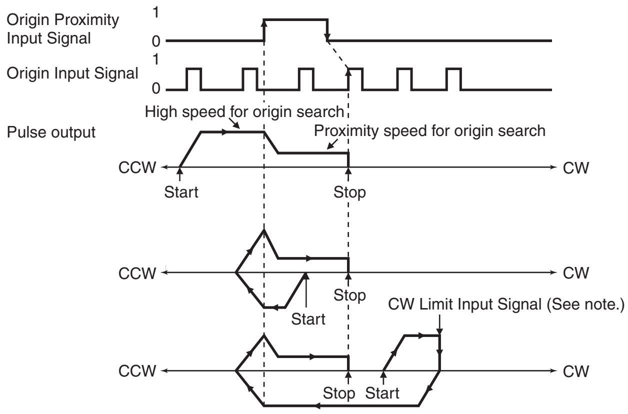
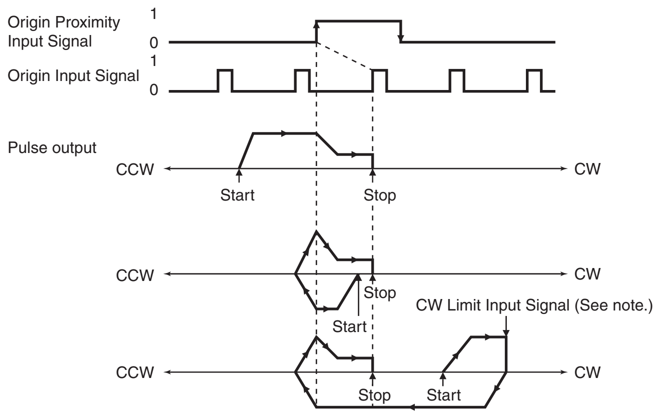
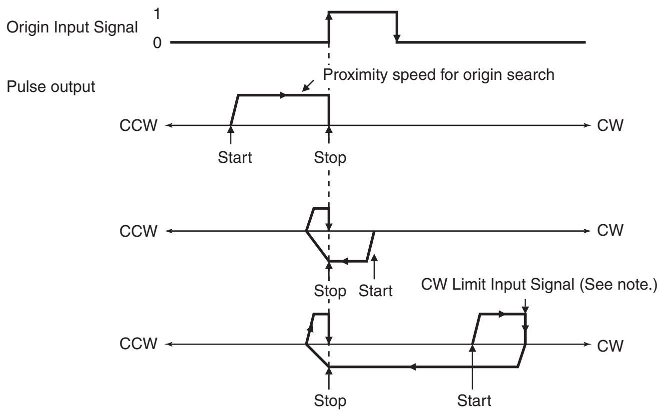
- **Origin Detection Method 2: Origin Proximity Input Signal Not Used**



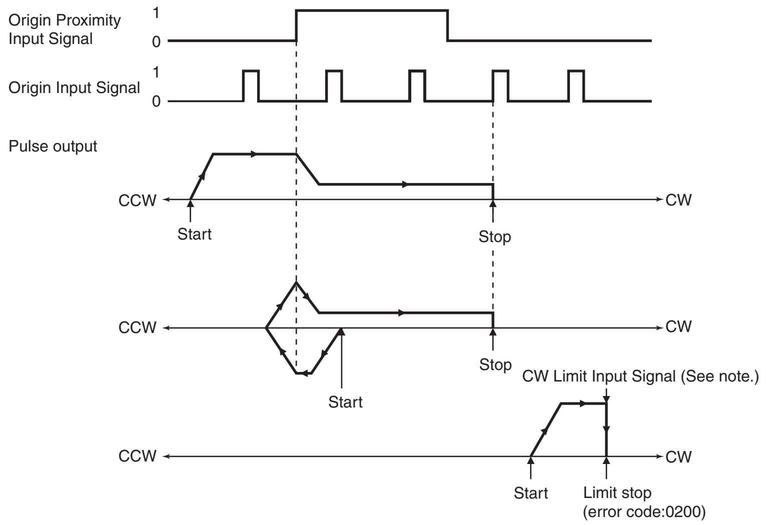
Operation Patterns for Origin Search Operating Mode and Origin Detection Method Settings

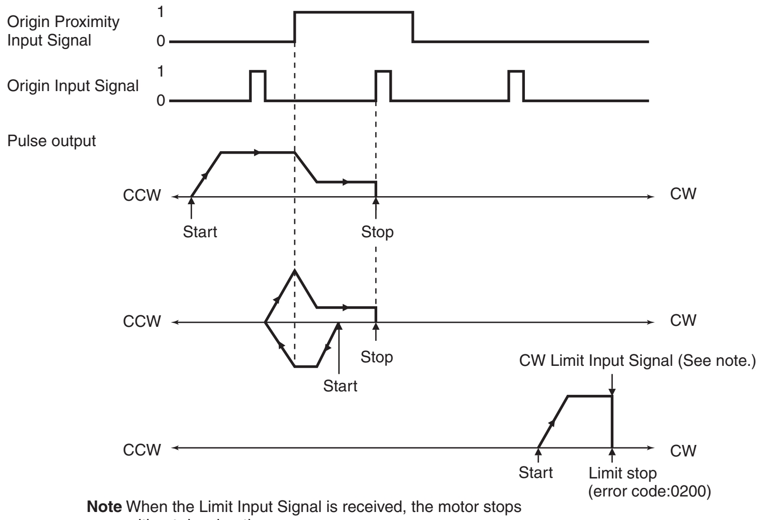
The following examples show how the operation patterns are affected by the origin detection method and origin search operating mode.
These examples have a CW origin search direction. (The search direction and Limit Input Signal direction would be different for an origin search in the CCW direction.)
Method 0 is the recommended method for reversal mode 1 (Inverse 1).

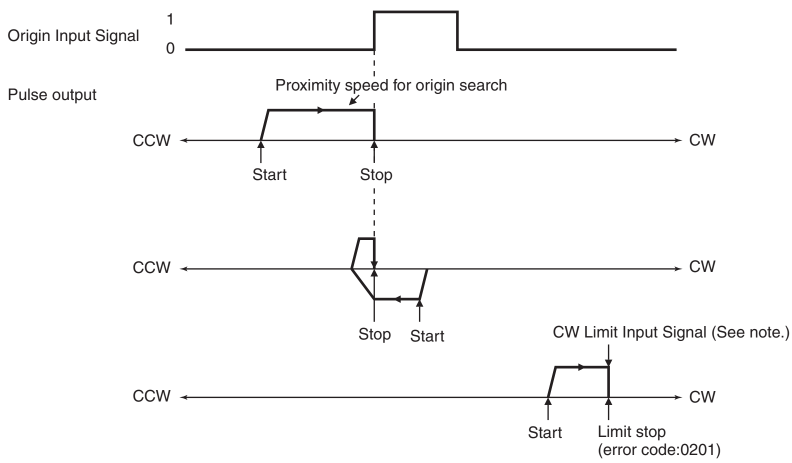
● Using Reversal Mode 1 (Inverse 1)

Origin search operation / Origin detection method	Reversal mode 1 (Inverse 1)
0: Origin Proximity Input Signal reversal required. (Recommended method)	 **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 Limit Input Signal is received, the motor stops without deceleration, reverses direction, and accelerates.

● Using Reversal Mode 2 (Inverse 2)

<table>
<tr><th>Origin search operation
Origin detection method</th><th>Reversal mode 2 (Inverse 2)</th></tr>
<tr><td>0: Origin Proximity Input Signal reversal required.</td><td>

Note When the Limit Input Signal is received, the motor stops without deceleration.</td></tr>
<tr><td>1: Origin Proximity Input Signal reversal not required.</td><td>

Note When the Limit Input Signal is received, the motor stops without deceleration.</td></tr>
<tr><td>2: Origin Proximity Input Signal not used.</td><td>

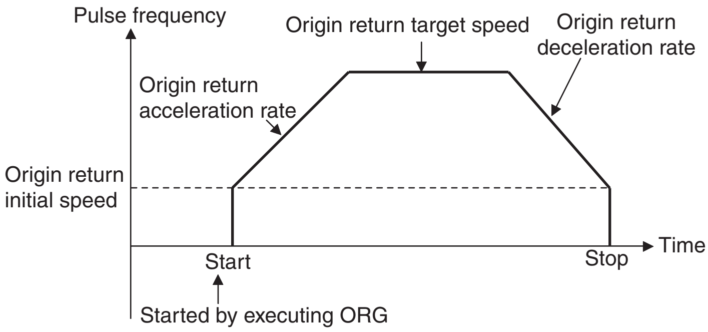
Note When the Limit Input Signal is received, the motor stops without deceleration.</td></tr>
</table>

12-6-6 Origin Return

It is the function to move the origin to the defined position by origin searches or changing PVs.

An origin return operation moves the motor to the origin position from any other position. The origin return operation is controlled by ORG.

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.



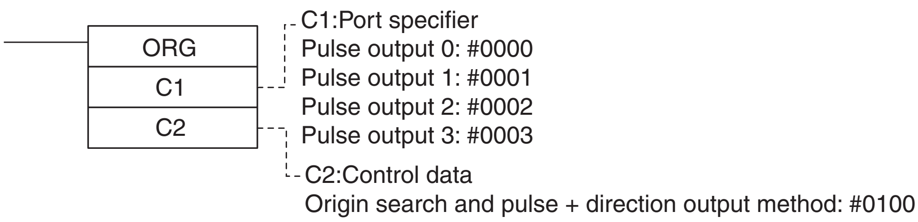
PLC Setup

The various origin return parameters are set on the Pulse Output 0 Tab Page in the PLC Setup.

● Origin Return Parameters

Name		Setting	Setting range
Base Settings	Search/Return Initial Speed	Sets the motor's starting speed when the origin return is executed. Specify the speed in the number of pulses per second (pps).	0 to 100k pps
Origin Return	Speed	Sets the motor's target speed when the origin return is executed. Specify the speed in the number of pulses per second (pps).	1 to 100k pps
	Acceleration Ratio (Rate)	Sets the motor's acceleration rate when the origin return function is accelerating. Specify the amount to increase the speed (Hz) per 4-ms interval.	1 to 65,535 (Hz/4ms)
	Deceleration Ratio (Rate)	Sets the motor's deceleration rate when the origin return function is decelerating. Specify the amount to decrease the speed (Hz) per 4-ms interval.	1 to 65,535 (Hz/4ms)

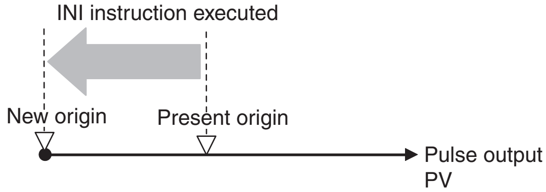
Origin Return Instruction



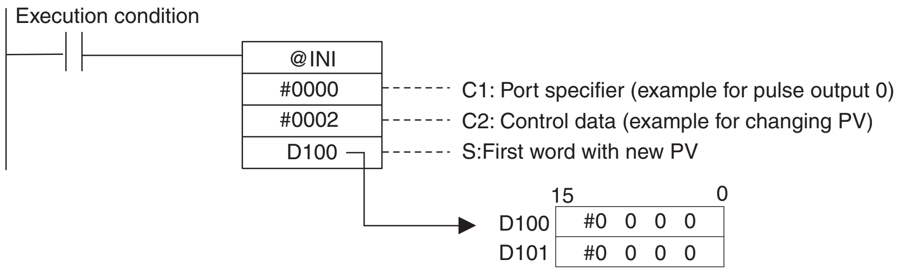
Note An instruction execution error will occur if the origin is not defined (relative coordinate system) when the ORG instruction is executed to perform an origin return operation.

12-6-7 Changing the Present Value of the Pulse Output

The present value of the pulse output can be changed by using the INI instruction. To define the present value as the origin, set the pulse output PV to 0 using the INI instruction.



● Example: Setting the Present Position as the Origin



Operands		Settings	
C1	Port specifier	#0000	Pulse output 0
		#0001	Pulse output 1
		#0002	Pulse output 2
		#0003	Pulse output 3
C2	Control data	#0002	Changes PV
S	First word with new PV	Store the new PV in S and S+1 (32 bits).	

12-7 Reading the Pulse Output Present Value

The present value of a pulse output can be read in the following two ways.

- Value refreshed at the I/O refresh timing → Read PV from Auxiliary Area.
- Value updated when a program is executed → Read PV by executing a PRV instruction.

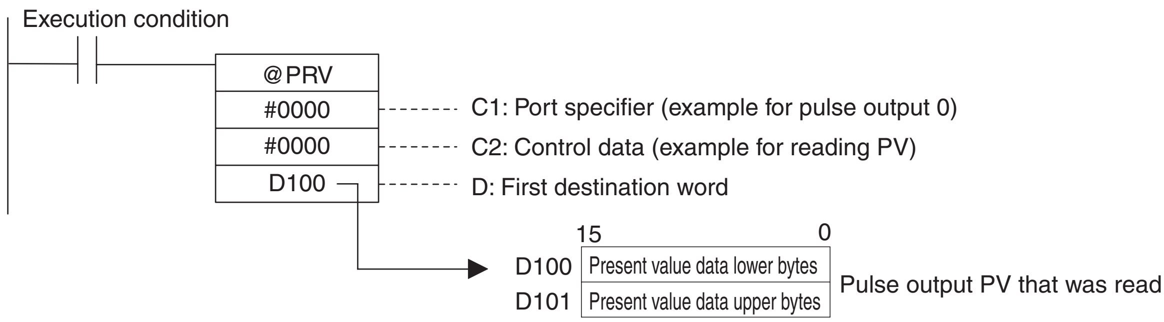
Reading the PV Refreshed at the I/O Refresh Timing

The PV that is stored in the following words can be read using the MOVL instruction or other instructions.

Read PV	Auxiliary Area words
Pulse output 0	A277 (upper digits) and A276 (lower digits)
Pulse output 1	A279 (upper digits) and A278 (lower digits)
Pulse output 2	A53 (upper digits) and A52 (lower digits)
Pulse output 3	A55 (upper digits) and A54 (lower digits)

Reading the Value When a Program is Executed

● Reading the Pulse Output PV with a PRV Instruction



12-8 Related Auxiliary Area Flags

Auxiliary Area Allocations

Name	Description	Values	Pulse output 0	Pulse output 1	Pulse output 2	Pulse output 3
Pulse Output PV Storage Words	PV range: 8000 0000 to 7FFF FFFF hex (-2,147,483,648 to 2,147,483,647)	Leftmost 4 digits	A277	A279	A53	A55
		Rightmost 4 digits	A276	A278	A52	A54
Pulse Output Reset Bit	The pulse output PV will be cleared when this bit is turned ON.	0: Not cleared. 1: Clear PV.	A540.00	A541.00	A542.00	A543.00
CW Limit Input Signal Flag	This flag shows the status of the CW Limit Input Signal, which is used in the origin search. The status of the signal from the CW limit input sensor connected to a normal input must be written to A540.08, A541.08, A542.08 or A543.08.	ON when turned ON from an external input.	A540.08	A541.08	A542.08	A543.08
CCW Limit Input Signal Flag	This flag shows the status of the CCW Limit Input Signal, which is used in the origin search. The status of the signal from the CCW limit input sensor connected to a normal input must be written to A540.09, A541.09, A542.09 or A543.09.	ON when turned ON from an external input.	A540.09	A541.09	A542.09	A543.09
Positioning completed input signal	This flag shows the status of the positioning completed input signal, which is used in the origin search. The status of the Positioning Completed Signal from the Servo Drive connected to a normal input must be written to A540.10, A541.10, A542.10 or A543.10.	ON when turned ON from an external input.	A540.10	A541.10	A542.10	A543.10
Accel/Decel Flag	ON when pulses are being output according to an ORG, ACC, PLS2, IFEED or ITPL instruction and the output frequency is being changed in steps (accelerating or decelerating).	0: Constant speed 1: Accelerating or decelerating	A280.00	A281.00	A56.00	A57.00
Overflow/Underflow Flag	ON when an overflow or underflow has occurred in the pulse output PV.	0: Normal 1: Overflow or underflow	A280.01	A281.01	A56.01	A57.01
Output Amount Set Flag	ON when the number of output pulses has been set with the PULS instruction.	0: No setting 1: Setting made	A280.02	A281.02	A56.02	A57.02
Output Completed Flag	ON when the number of output pulses set with the PULS, PLS2, IFEED or ITPL instruction has been output.	0: Output not completed. 1: Output completed.	A280.03	A281.03	A56.03	A57.03
Output In-progress Flag	ON when pulses are being output from the pulse output.	0: Stopped 1: Outputting pulses.	A280.04	A281.04	A56.04	A57.04
No-origin Flag	ON when the origin has not been defined for the pulse output.	0: Origin defined. 1: Origin undefined.	A280.05	A281.05	A56.05	A57.05
At-origin Flag	ON when the pulse output PV matches the origin (0).	0: Not stopped at origin. 1: Stopped at origin.	A280.06	A281.06	A56.06	A57.06
Output Stopped Error Flag	ON when an error occurred while outputting pulses in the origin search function.	0: No error 1: Stop error occurred.	A280.07	A281.07	A56.07	A57.07
Stop Error Code	When a Pulse Output Stop Error occurs, the error code is stored in that pulse outputs corresponding Stop Error Code word.	–	A444	A445	A438	A439

12-9 Application Examples

12-9-1 Vertically Conveying PCBs (Multiple Progressive Positioning)

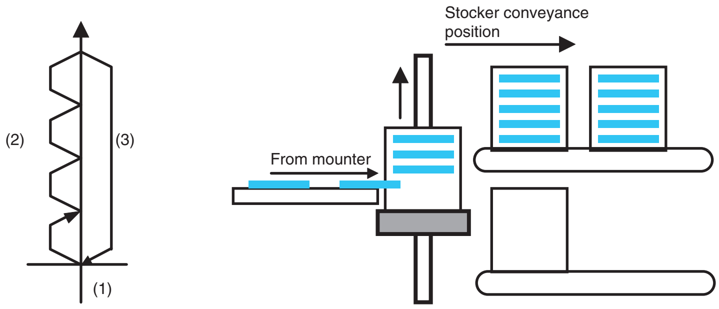
Specifications and Operation

● Outline

(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

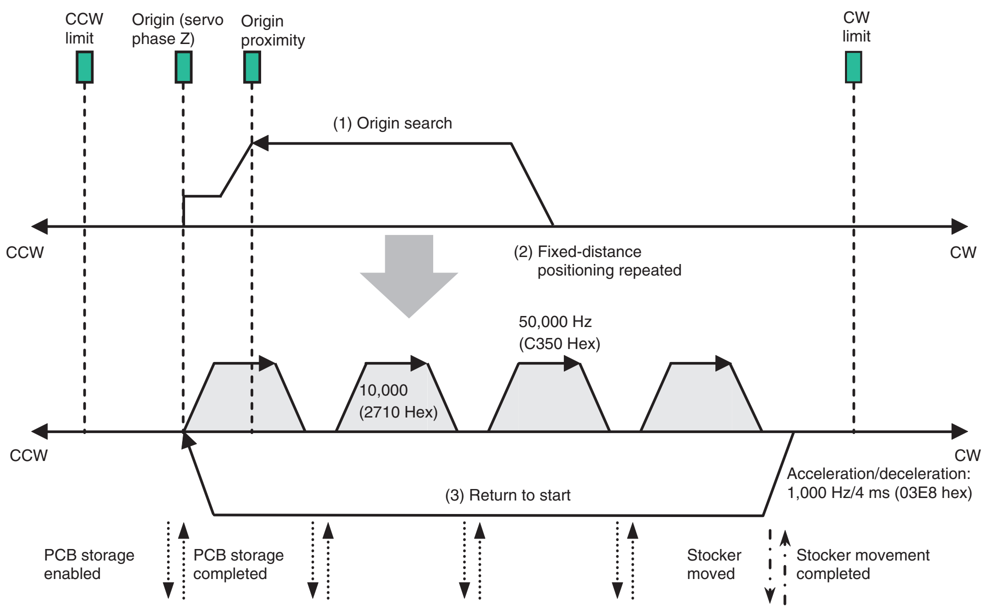


● Operation Pattern

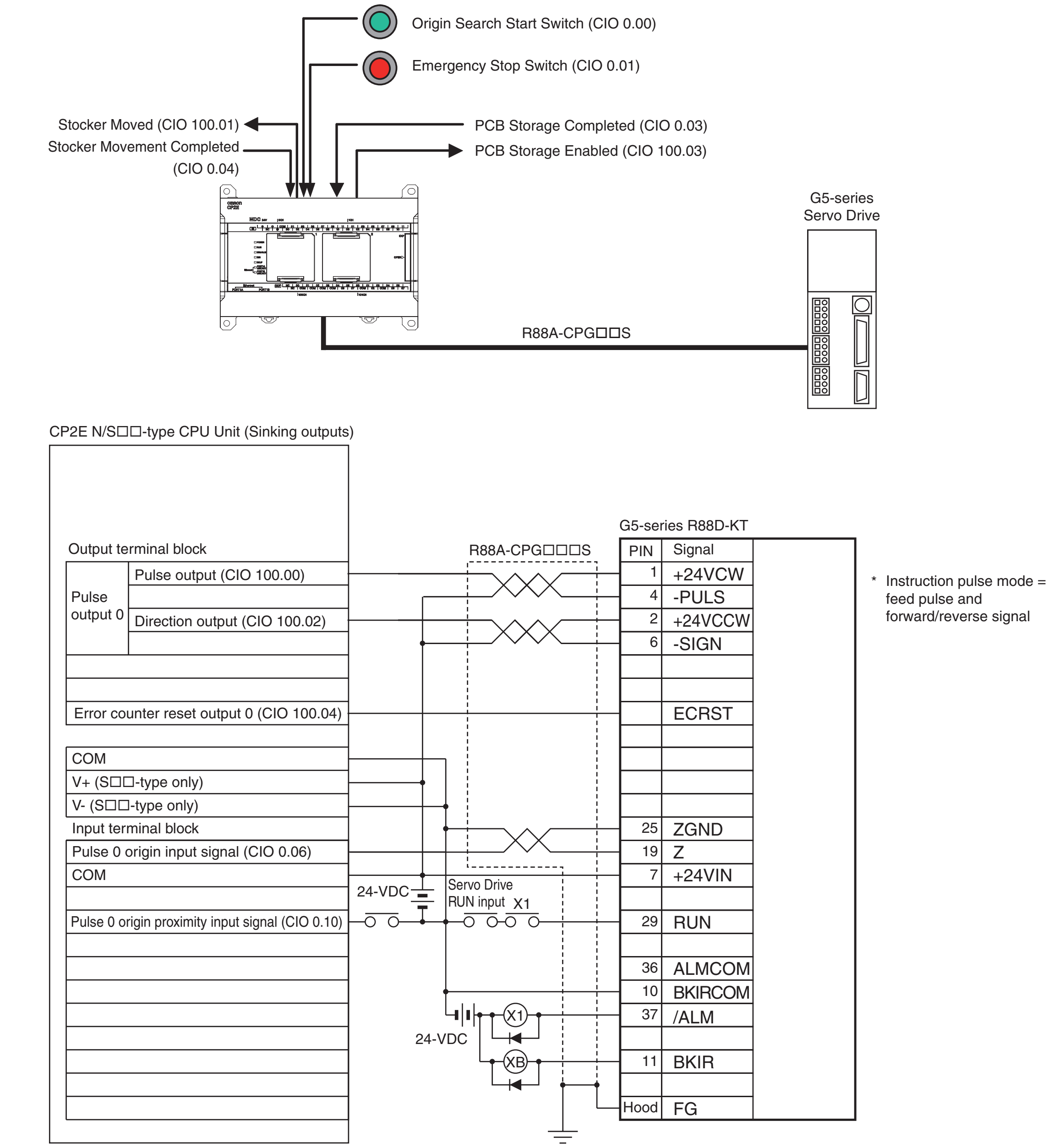
(1) An origin search is performed.

(2) Fixed-distance positioning is repeated.

(3) The system is returned to the original position.



● Wiring Example Using G5-series Servo Drive



Only S□□-type CPU Units can wire V+ and V-. Do not wire them in N□□-type CPU Units.

● Operation

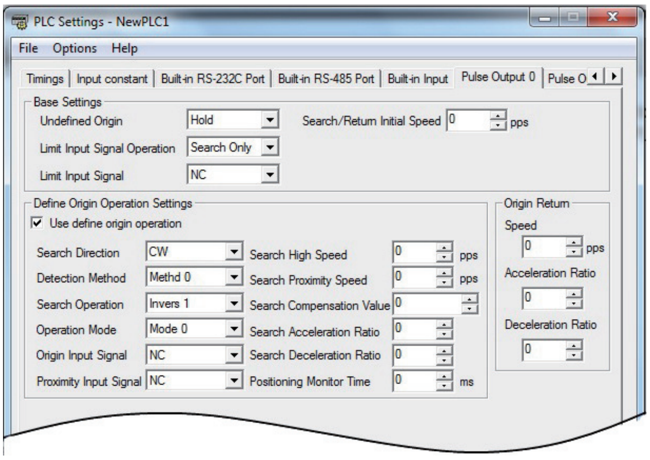
1. An origin search is performed using the Origin Search Start Switch (CIO 0.00).
2. When the origin search is finished, the PCB Storage Enabled Output (CIO 100.03) is turned ON.
3. When a PCB has been stored, the stocker is raised (relative positioning) using the PCB Storage Completed Input (CIO 0.03).
4. Storing PCBs is repeated until the stocker is full.
5. The number of PCBs in the stocker is counted with counter C0 by counting the number of times the stocker is raised.
6. When the stocker is full, it is moved (CIO 100.01) and only the conveyor is lowered (absolute positioning) when stoker movement is completed (CIO 0.04).
7. An emergency stop is executed to stop pulse output with the Emergency Stop Switch Input (CIO 0.01).

Preparations

● PLC Setup

Setting
Use define origin operation for pulse output 0.

Note The Use define origin operation setting is read from the PLC Setup when the power supply is turned ON.



● DM Area Settings

- Settings for PLS2 for Fixed-distance Positioning (D0 to D7)

Setting details	Address	Data
Acceleration rate: 1,000 Hz/4 ms	D0	#03E8
Deceleration rate: 1,000 Hz/4 ms	D1	#03E8
Target frequency: 50,000 Hz	D2	#C350
	D3	#0000
Number of output pulses: 10,000 pulses	D4	#2710
	D5	#0000
Starting frequency: 0 Hz	D6	#0000
	D7	#0000

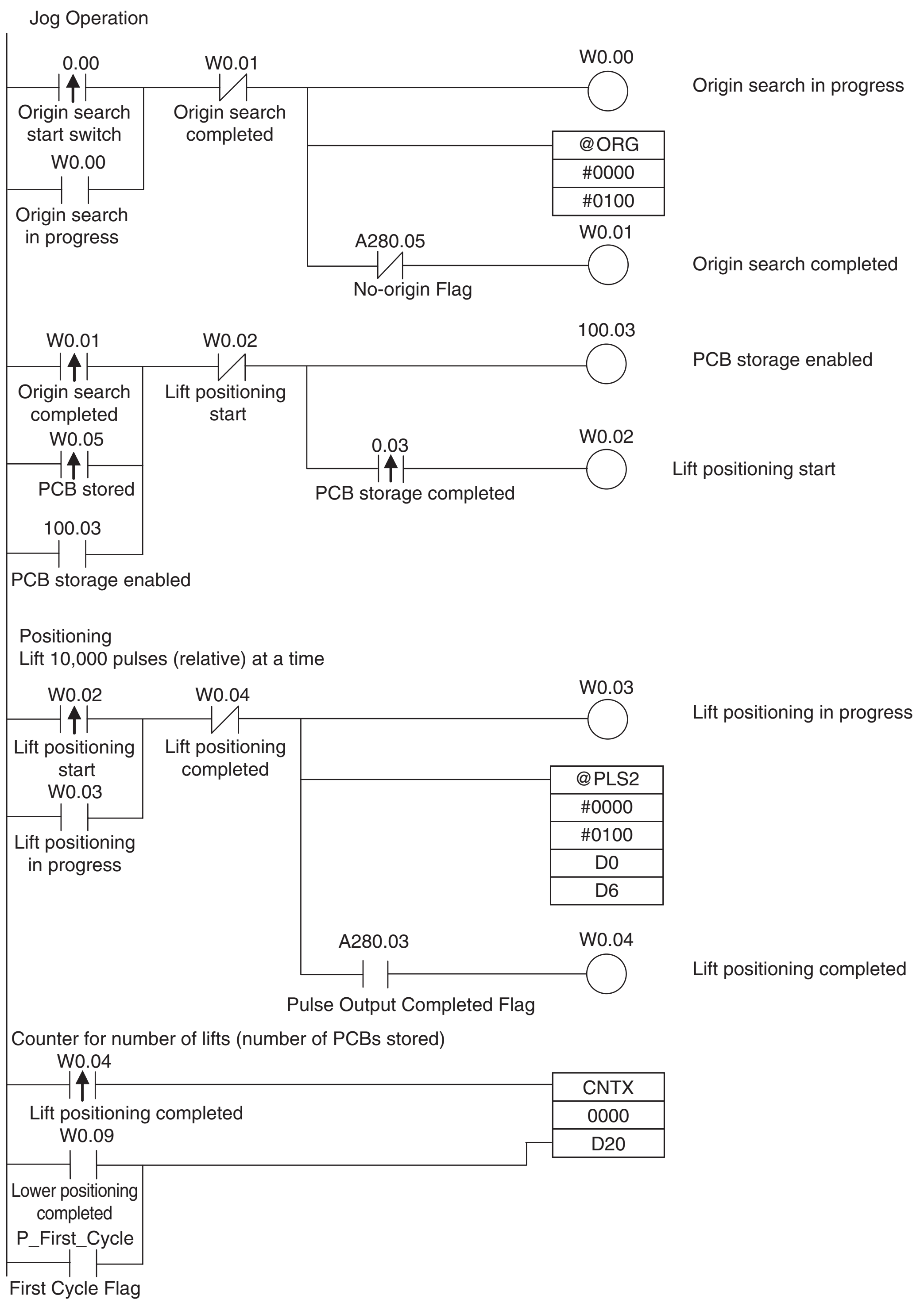
- Settings for PLS2 to Return to Start (D10 to D17)

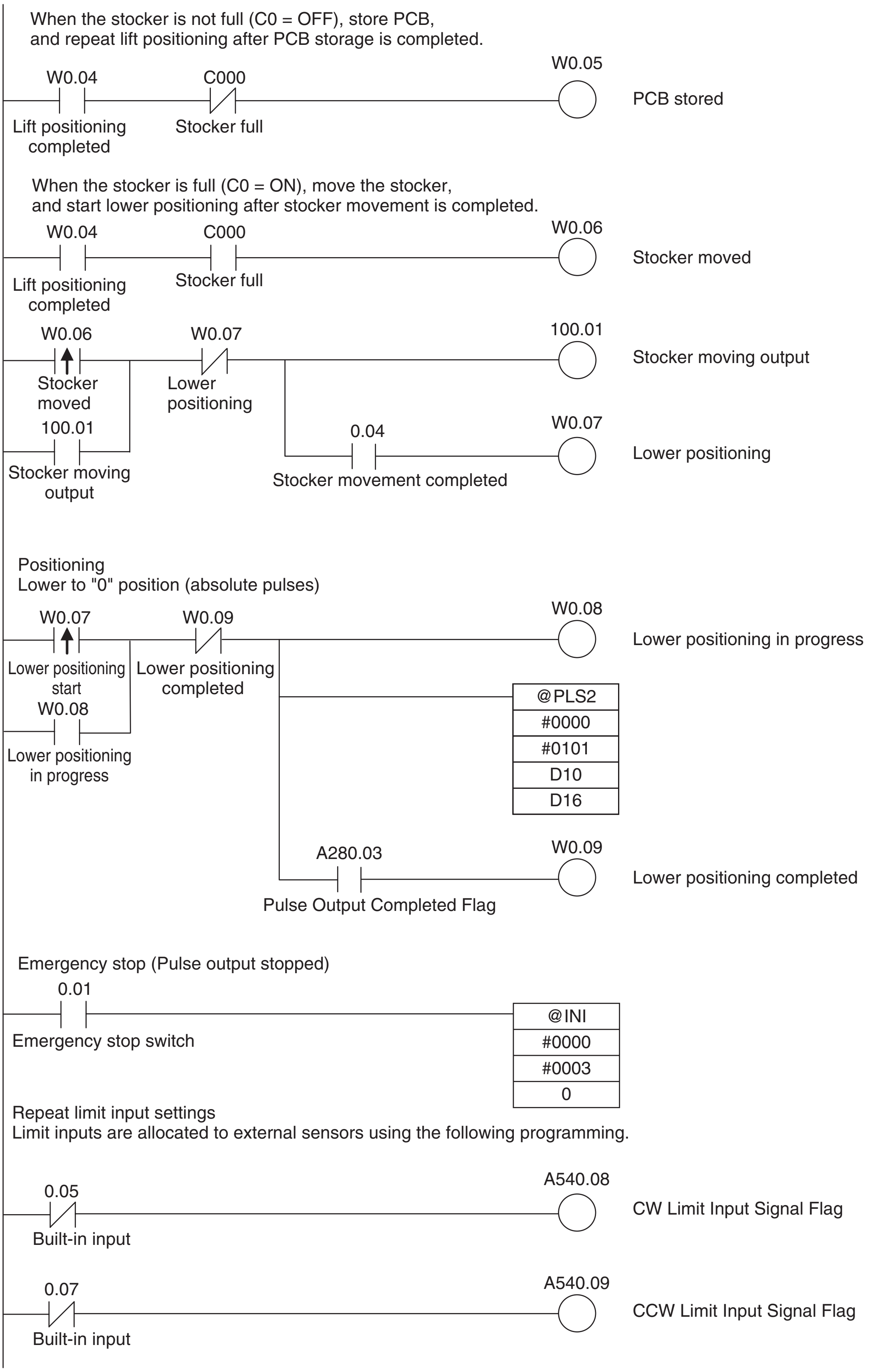
Setting details	Address	Data
Acceleration rate: 300 Hz/4 ms	D10	#012C
Deceleration rate: 200 Hz/4 ms	D11	#00C8
Target frequency: 50,000 Hz	D12	#C350
	D13	#0000
Number of output pulses: 0 pulse	D14	#0000
	D15	#0000
Starting frequency: 100 Hz	D16	#0064
	D17	#0000

- Number of Repeats of Fixed-distance Positioning Operation (D20)

Setting details	Address	Data
Number of repeats of fixed-distance positioning operation (number of PCBs in stocker)	D20	#000F

Ladder Program

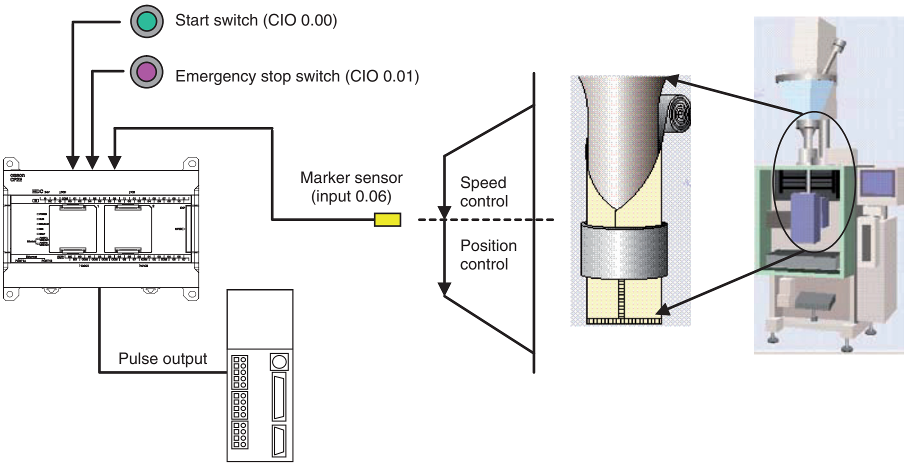



When the stocker is not full (C0 = OFF), store PCB,
and repeat lift positioning after PCB storage is completed.
W0.04
C000
W0.05
PCB stored
Lift positioning completed
Stocker full
When the stocker is full (C0 = ON), move the stocker,
and start lower positioning after stocker movement is completed.
W0.04
C000
W0.06
Stocker moved
Lift positioning completed
Stocker full
W0.06
W0.07
100.01
Stocker moving output
Stocker moved
Lower positioning
100.01
0.04
W0.07
Lower positioning
Stocker moving output
Stocker movement completed
Positioning
Lower to "0" position (absolute pulses)
W0.07
W0.09
W0.08
Lower positioning in progress
Lower positioning start
Lower positioning completed
@PLS2
#0000
#0101
D10
D16
W0.08
Lower positioning in progress
A280.03
W0.09
Lower positioning completed
Pulse Output Completed Flag
Emergency stop (Pulse output stopped)
0.01
@INI
#0000
#0003
0
Emergency stop switch
Repeat limit input settings
Limit inputs are allocated to external sensors using the following programming.
0.05
A540.08
CW Limit Input Signal Flag
Built-in input
0.07
A540.09
CCW Limit Input Signal Flag
Built-in input

12-9-2 Feeding Wrapping Material: Interrupt Feeding

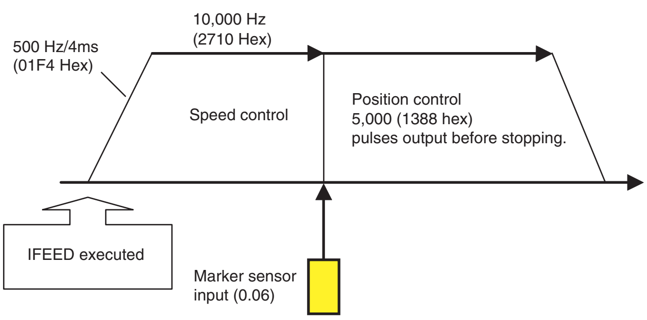
Specifications and Operation

● 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 (rising) is received, fixed-distance positioning is performed before stopping.



● Operation

1 Speed control is used to feed wrapping material to the initial position by executing the IFEED instruction when the Start Switch (CIO 0.00) is activated.

2 When the Marker Sensor Input (CIO 0.06) is received, the operation is switched to position control.

3 The axis is moved the specified travel amount and then stopped.

4 An emergency stop is executed to stop pulse output with the Emergency Stop Switch input (CIO 0.01).

Preparations

● PLC Setup

Setting
Enable using built-in input IN6 as an interrupt input.

Note The interrupt input setting is read from the PLC Setup when the power supply is turned ON.

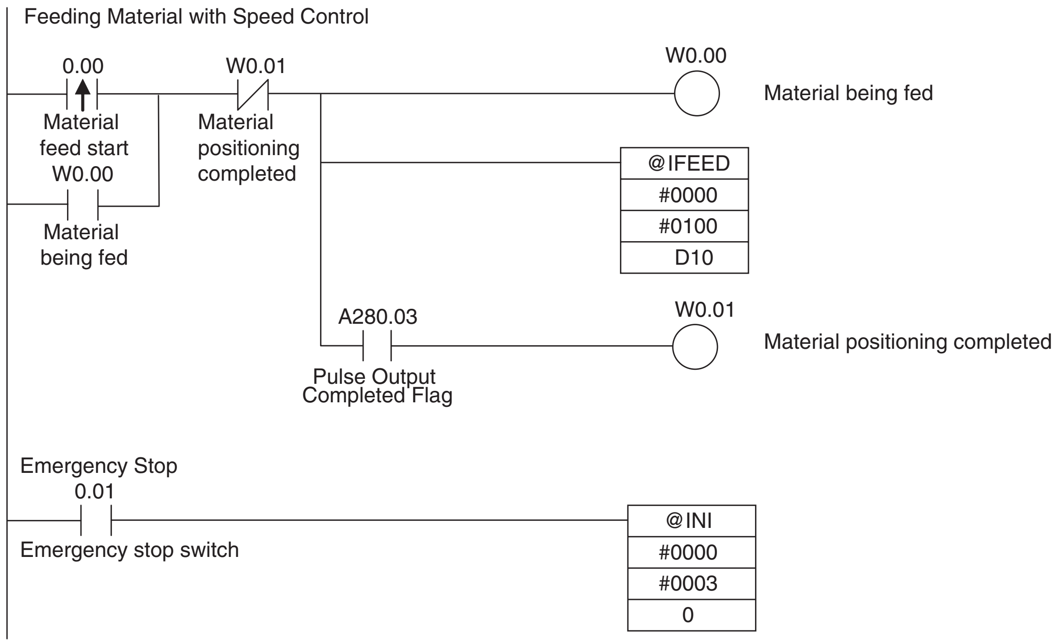
● DM Area Settings

- Speed Control Settings to Feed Wrapping Material to Initial Position and Positioning Control Settings for Wrapping Material

Setting	Address	Data
Acceleration rate: 500 Hz/4 ms	D10	#01F4
Deceleration rate: 500 Hz/4 ms	D11	#01F4
Target frequency: 10,000 Hz	D12	#2710
	D13	#0000
Number of output pulses: 5,000 pulses	D14	#1388
	D15	#0000

Ladder Program

● Cyclic Task Program (Executed at Startup)

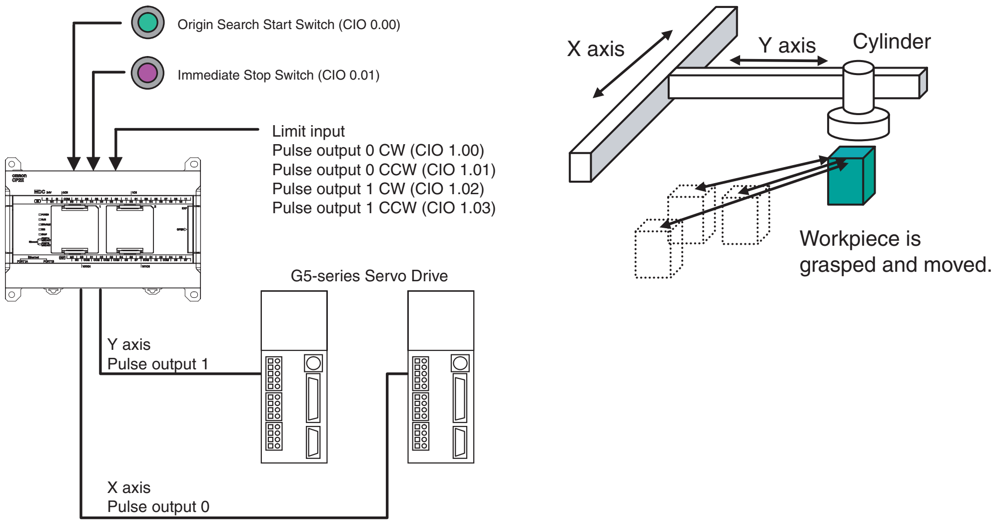


Note When the interrupt input is falling, it is necessary to specify edge detection falling with MSKS instruction. In this example, the MSKS instruction can be omitted because it is rising.

12-9-3 Palletize: Two-axis Multipoint Positioning

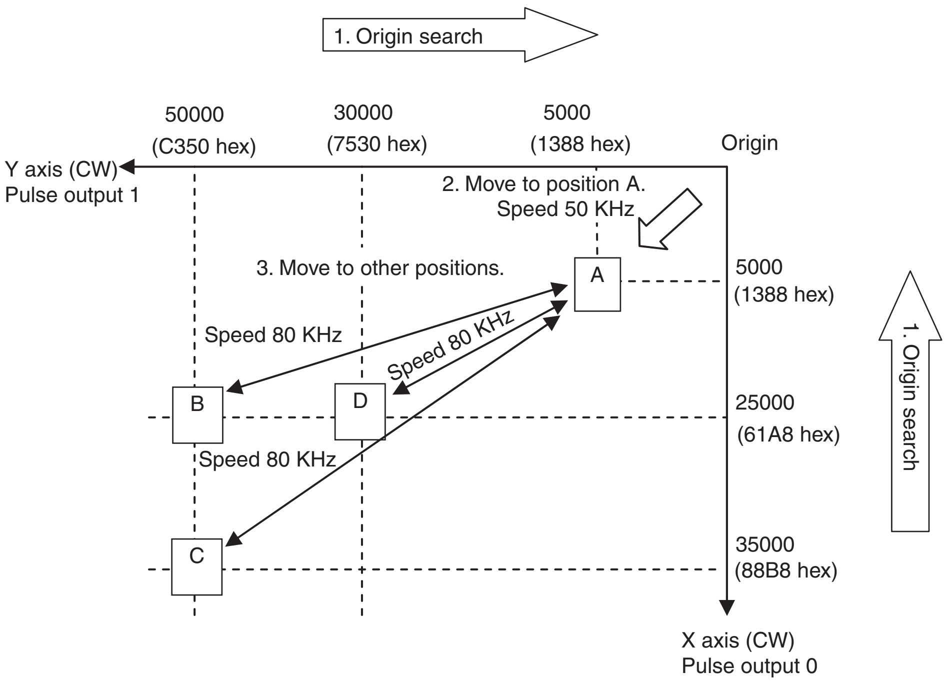
Specifications and Operation

● Overview



● Operation Pattern

1. Perform origin search.
2. A workpiece is grasped and moved to position A.
3. The workpiece is repeatedly moved between the grasp position and the assembly positions using linear interpolation operation.



- **Operation**

1 An origin search of X axis (pulse output 0) and Y axis (pulse output 1) is performed using the Origin Search Start Switch (CIO 0.00).

2 When the origin search is finished, the following operations are performed continuously using linear interpolation 0.

Move in to A.

Move to B and return to A.

Move to C and return to A.

Move to D and return to A.

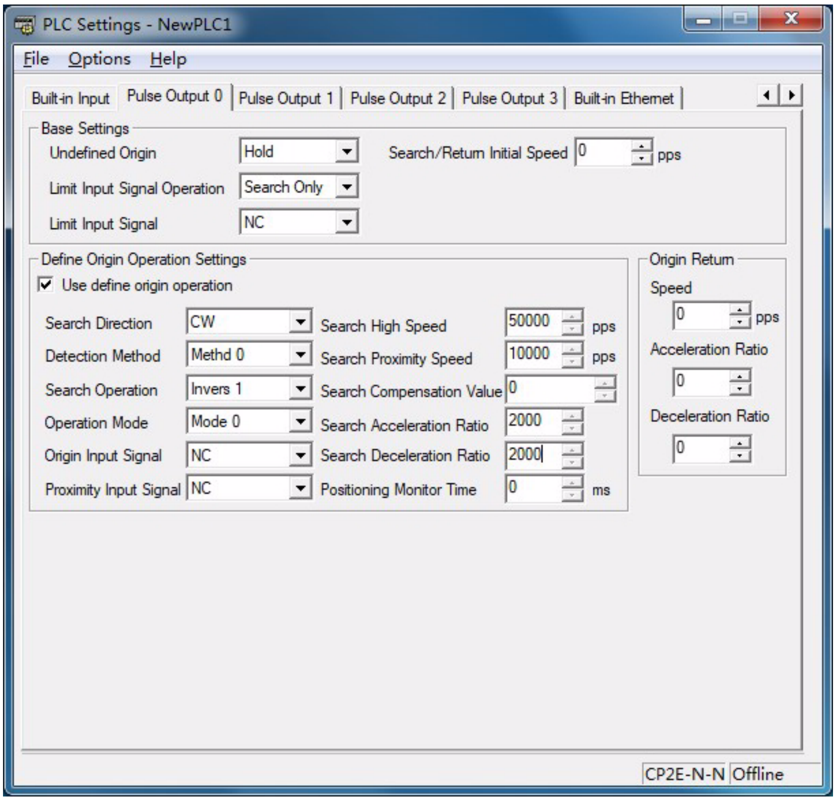
3 An immediate stop is executed to stop pulse output with the Immediate Stop input (CIO 0.01).

Preparations

- **PLC Setup**

Setting
Origin Search Detailed Settings for pulse output 0 and 1

Note The setting of the option to use the origin search is read from the PLC Setup when the power supply is turned ON.



DM Area Settings

- ITPL(893) Settings to Move from Origin to Position A

Setting	Address	Data
Acceleration rate: 2,000 pps/4 ms	D10	& 2,000
Deceleration rate: 2,000 pps/4 ms	D11	& 2,000
Target frequency: 50,000 pps	D12, D13	& 50,000
Starting frequency: 0 pps	D14, D15	& 0
Number of output pulses (X axis): 5,000 pulses	D16, D17	+5,000
Number of output pulses (Y axis): 5,000 pulses	D18, D19	+5,000

- Settings to Move from Position A to Position B using Linear Interpolation

Setting	Address	Data
Acceleration rate: 2,000 pps/4 ms	D30	& 2,000
Deceleration rate: 2,000 pps/4 ms	D31	& 2,000
Target frequency: 80,000 pps	D32, D33	& 80,000
Starting frequency: 0 pps	D34, D35	& 0
Number of output pulses (X axis): 25,000 pulses	D36, D37	+25,000
Number of output pulses (Y axis): 50,000 pulses	D38, D39	+50,000

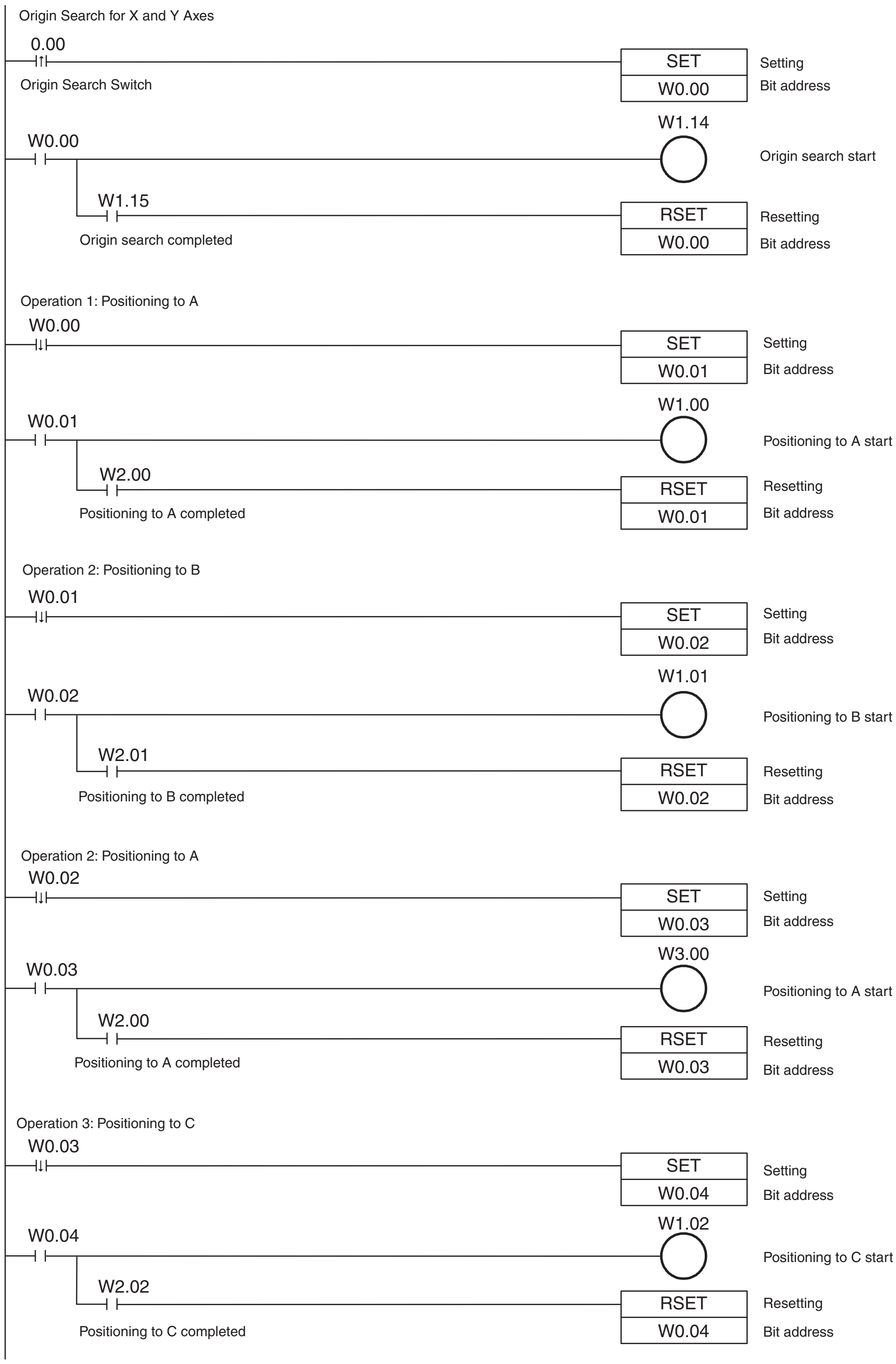
- Settings to Move from Position A to Position C using Linear Interpolation

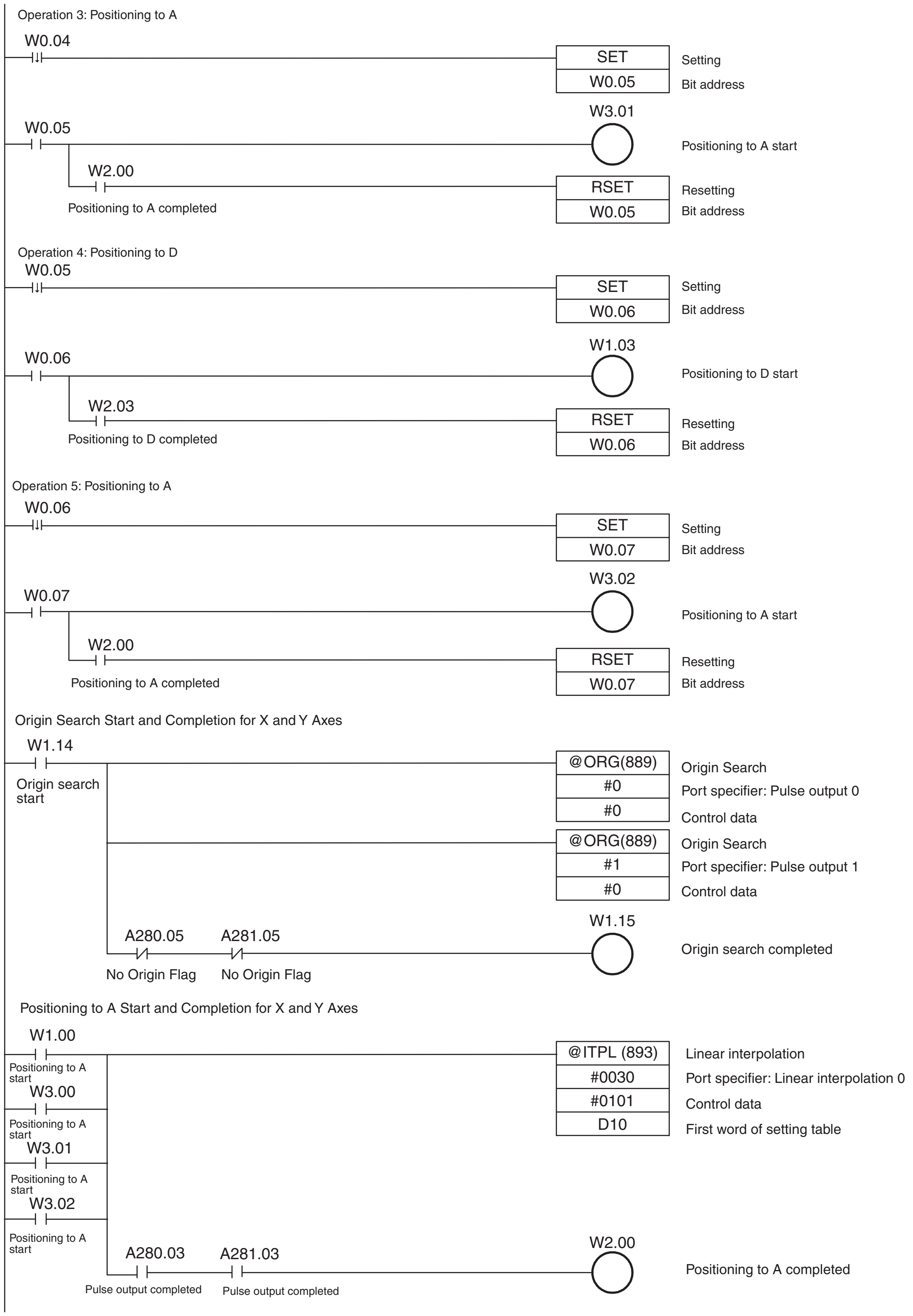
Setting	Address	Data
Acceleration rate: 2,000 pps/4 ms	D50	& 2,000
Deceleration rate: 2,000 pps/4 ms	D51	& 2,000
Target frequency: 80,000 pps	D52, D53	& 80,000
Starting frequency: 0 pps	D54, D55	& 0
Number of output pulses (X axis): 35,000 pulses	D56, D57	+35,000
Number of output pulses (Y axis): 50,000 pulses	D58, D59	+50,000

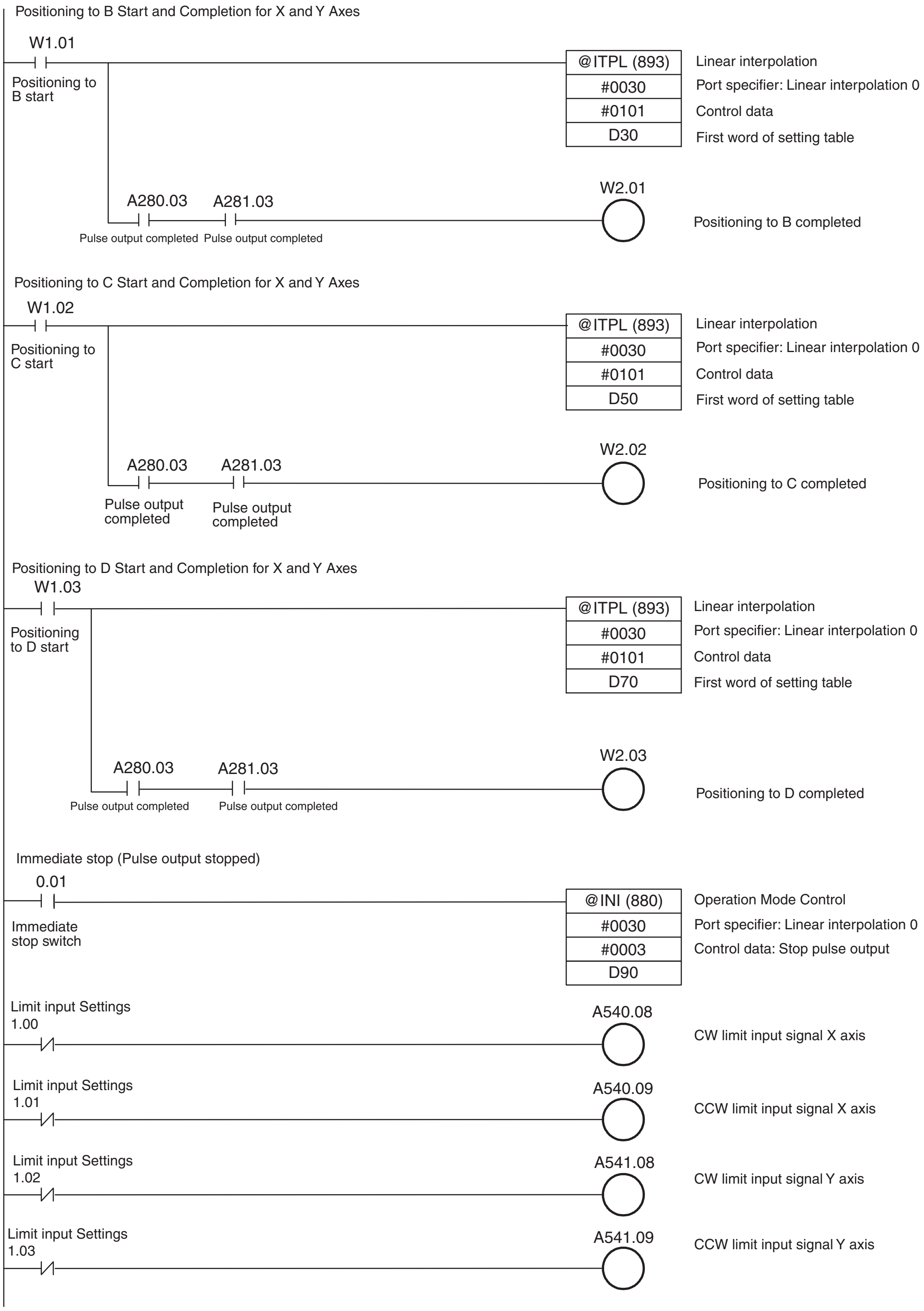
- Settings to Move from Position A to Position D using Linear Interpolation

Setting	Address	Data
Acceleration rate: 2,000 pps/4 ms	D70	& 2,000
Deceleration rate: 2,000 pps/4 ms	D71	& 2,000
Target frequency: 80,000 pps	D72, D73	& 80,000
Starting frequency: 0 pps	D74, D75	& 0
Number of output pulses (X axis): 25,000 pulses	D76, D77	+25,000
Number of output pulses (Y axis): 30,000 pulses	D78, D79	+30,000

Ladder Program




Operation 3: Positioning to A
W0.04
SET
W0.05
Setting
Bit address
W0.05
W3.01
Positioning to A start
W2.00
RSET
W0.05
Resetting
Bit address
Positioning to A completed
Operation 4: Positioning to D
W0.05
SET
W0.06
Setting
Bit address
W0.06
W1.03
Positioning to D start
W2.03
RSET
W0.06
Resetting
Bit address
Positioning to D completed
Operation 5: Positioning to A
W0.06
SET
W0.07
Setting
Bit address
W0.07
W3.02
Positioning to A start
W2.00
RSET
W0.07
Resetting
Bit address
Positioning to A completed
Origin Search Start and Completion for X and Y Axes
W1.14
Origin search start
@ORG(889)
#0
#0
Origin Search
Port specifier: Pulse output 0
Control data
@ORG(889)
#1
#0
Origin Search
Port specifier: Pulse output 1
Control data
A280.05
A281.05
No Origin Flag
No Origin Flag
W1.15
Origin search completed
Positioning to A Start and Completion for X and Y Axes
W1.00
Positioning to A start
W3.00
Positioning to A start
W3.01
Positioning to A start
W3.02
Positioning to A start
@ITPL (893)
#0030
#0101
D10
Linear interpolation
Port specifier: Linear interpolation 0
Control data
First word of setting table
A280.03
A281.03
Pulse output completed
Pulse output completed
W2.00
Positioning to A completed


Positioning to B Start and Completion for X and Y Axes
W1.01
Positioning to B start
@ITPL (893)
#0030
#0101
D30
Linear interpolation
Port specifier: Linear interpolation 0
Control data
First word of setting table
A280.03
A281.03
Pulse output completed
Pulse output completed
W2.01
Positioning to B completed
Positioning to C Start and Completion for X and Y Axes
W1.02
Positioning to C start
@ITPL (893)
#0030
#0101
D50
Linear interpolation
Port specifier: Linear interpolation 0
Control data
First word of setting table
A280.03
A281.03
Pulse output completed
Pulse output completed
W2.02
Positioning to C completed
Positioning to D Start and Completion for X and Y Axes
W1.03
Positioning to D start
@ITPL (893)
#0030
#0101
D70
Linear interpolation
Port specifier: Linear interpolation 0
Control data
First word of setting table
A280.03
A281.03
Pulse output completed
Pulse output completed
W2.03
Positioning to D completed
Immediate stop (Pulse output stopped)
0.01
Immediate stop switch
@INI (880)
#0030
#0003
D90
Operation Mode Control
Port specifier: Linear interpolation 0
Control data: Stop pulse output
Limit input Settings
1.00
A540.08
CW limit input signal X axis
Limit input Settings
1.01
A540.09
CCW limit input signal X axis
Limit input Settings
1.02
A541.08
CW limit input signal Y axis
Limit input Settings
1.03
A541.09
CCW limit input signal Y axis

12-10 Precautions when Using Pulse Outputs

Movement Direction when Specifying Absolute Pulses

When operating with the absolute pulse specification, the movement direction (CW/CCW) 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, SPED or PLS2 instruction is not effective.

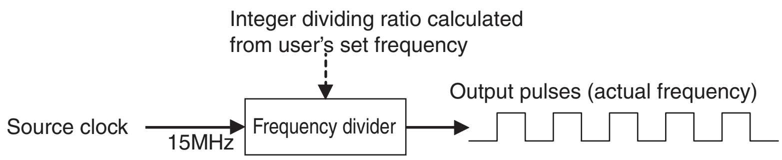
Using CW/CCW Limit Inputs for Pulse Output Functions other than Origin Searches

Pulse outputs will stop according to the PLC Setup when either the CW or CCW Limit Input Signals turns ON. It is also possible to select whether or not the defined origin will be cleared when a CW or CCW Limit Input Signal turns ON for a pulse output function.

Difference between Set Frequencies and Actual Frequencies

The CP2E CPU Unit's pulse output frequency is determined by dividing the source clock frequency (15 MHz) by an integer ratio. 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

$$\text{Actual frequency (Hz)} = \frac{\text{Source clock frequency}}{\text{Dividing ratio}}$$

$$\text{Dividing ratio} = \text{INT}\left(\frac{\text{Source clock frequency} \times 2 + \text{Set frequency}}{\text{Set frequency (Hz)} \times 2}\right)$$

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: 15 MHz

Set frequency (kHz)	Actual frequency (kHz)
99.668 to 100.000	100.000
99.010 to 99.667	99.338
:	:
49.917 to 50.083	50.000
49.752 to 49.916	49.834
49.587 to 49.751	49.669
:	:
9.997 to 10.003	10.000
9.991 to 9.996	9.993
9.984 to 9.990	9.987

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.

A second independent-mode positioning instruction can be started if an independent-mode positioning instruction is being executed, and a second continuous-mode speed control instruction can be started if a continuous-mode speed control instruction is being executed. Operation cannot be switched between the independent and continuous modes, although a PLS2 instruction can be executed while a ACC instruction (continuous mode) is being executed.

It is possible to start another operation during acceleration/deceleration and start another positioning instruction during positioning.

Yes:Can be executed. No:Error occurs.

Instruction being executed		Instruction being started								
		INI	SPED (Independent)	SPED (Continuous)	ACC (Independent)	ACC (Continuous)	PLS2	ORG	IFEED	ITPL
SPED (Independent)		Yes	Yes (*1)	No	Yes (*3)	No	No	No	No	No
SPED (Continuous)		Yes	No	Yes (*2)	No	Yes (*5)	No	No	No	No
ACC (Independent)	Steady speed	Yes	No	No	Yes (*4)	No	Yes (*6)	No	No	No
	Accelerating or decelerating	Yes	No	No	Yes (*4)	No	Yes (*6)	No	No	No
ACC (Continuous)	Steady speed	Yes	No	No	No	Yes (*5)	Yes (*7)	No	No	No
	Accelerating or decelerating	Yes	No	No	No	Yes (*5)	Yes (*7)	No	No	No
PLS2	Steady speed	Yes	No	No	Yes (*4)	No	Yes (*8)	No	No	No
	Accelerating or decelerating	Yes	No	No	Yes (*4)	No	Yes (*8)	No	No	No
ORG	Steady speed	Yes	No	No	No	No	No	No	No	No
	Accelerating or decelerating	Yes	No	No	No	No	No	No	No	No
IFEED	Steady speed	Yes	No	No	No	No	No	No	Yes (*9)	No
	Accelerating or decelerating	Yes	No	No	No	No	No	No	Yes (*9)	No
ITPL	Steady speed	Yes	No	No	No	No	No	No	No	No
	Accelerating or decelerating	Yes	No	No	No	No	No	No	No	No

*1 SPED (Independent) to SPED (Independent)
- The number of output pulses cannot be changed.
- The frequency can be changed.

*2 SPED (Continuous) to SPED (Continuous)
- The frequency can be changed.

*3 SPED (Independent) to ACC (Independent)
- The number of output pulses cannot be changed.
- The frequency can be changed.
- The acceleration/deceleration rate can be changed.

*4 ACC (Independent) to ACC (Independent) or PLS2 to ACC (Independent)
- The number of output 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.)

*5 SPED (Continuous) to ACC (Continuous) or ACC (Continuous) to ACC (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.)

*6 ACC (Independent) to PLS2
- The number of output 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.)

*7 ACC (Continuous) to PLS2
- 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.)

*8 PLS2 to PLS2
- The number of output 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.)

*9 IFEED to IFEED
- Possible only when target frequency is 0 Hz (deceleration stop).

Origin Search Error Processing

The CP2E 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	Settings	Pulse output 0	Pulse output 1	Pulse output 2	Pulse output 3
Output Stopped Error Flags ON when an error occurred while outputting pulses in the origin search function.	0: No error 1: Stop error occurred.	A280.07	A281.07	A56.07	A57.07
Stop Error Codes When a Pulse Output Stop Error occurs, the error code is stored in that pulse outputs corresponding Stop Error Code word.		A444	A445	A438	A439

Pulse Output Stop Error Codes

Error name	Error code	Likely cause	Corrective action	Operation after error
CW Limit Stop Input Signal	0100	Stopped due to a CW limit signal input.	Move in the CCW direction.	Immediate stop No effect on other port
CCW Limit Stop Input Signal	0101	Stopped due to a CCW limit signal input.	Move in the CW direction.	
No Origin Proximity Input Signal	0200	The parameters indicate that the Origin Proximity Input Signal is being used, but a Origin Proximity Input Signal was not received during the origin search.	Check the wiring of the Origin Proximity Input Signal as well as the PLC Setup's Origin Proximity Input Signal Type setting (NC or NO) and execute the origin search again.	No effect on other port
No Origin Input Signal	0201	The Origin Input Signal was not received during the origin search.	Check the wiring of the Origin Input Signal as well as the PLC Setup's Origin Input Signal Type setting (NC or NO) and execute the origin search again.	
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. • Increase the distance between the Origin Proximity Input Signal sensor and Origin Input Signal sensor. • Decrease the origin search high speed.	Decelerates to a stop. 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.	Operation will not start. 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.	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. • 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.	Immediate stop No effect on other port

Error name	Error code	Likely cause	Corrective action	Operation after error
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. • 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.	Immediate stop No effect on other port
Positioning Timeout Error	0300	The Servo Drive'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.	No effect on other port

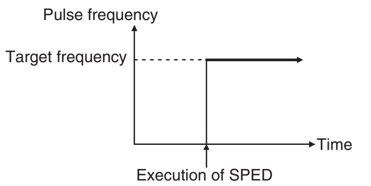
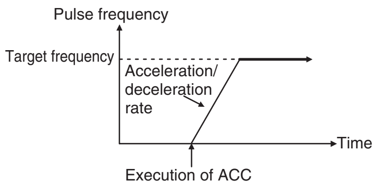
12-11 Pulse Output Pattern

The CP2E CPU Unit's pulse output function enables operation in Continuous Mode, for which the number of output pluses is not specified, or in Independent Mode, for which the number of output pulses is specified. Continuous Mode is used for speed control and Independent Mode is used for positioning.

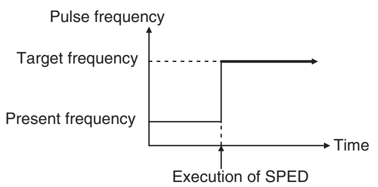
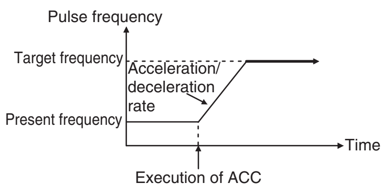
12-11-1 Speed Control (Continuous Mode)

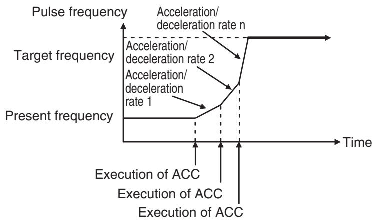
The following operations can be performed in Continuous Mode by combining instructions.

Starting a Pulse Output

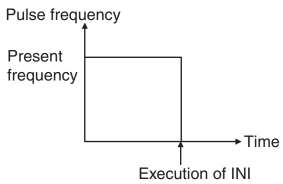
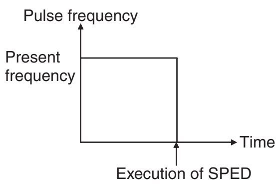
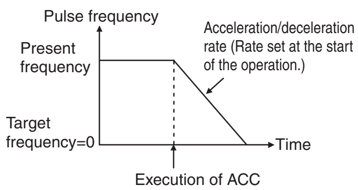
Operation	Example application	Frequency changes	Description	Procedure	
				Instruction	Settings
Output with specified speed	Changing the speed (frequency) in one step		Outputs pulses at a specified frequency.	SPED (Continuous)	• Port • Pulse + direction • Continuous • Target frequency
Output with specified acceleration and speed	Accelerating the speed (frequency) at a fixed rate		Outputs pulses and changes the frequency at a fixed rate.	ACC (Continuous)	• Port • Pulse + direction • Continuous • Acceleration/ deceleration rate • Target frequency

Changing Settings

Operation	Example application	Frequency changes	Description	Procedure	
				Instruction	Settings
Change speed in one step	Changing the speed during operation		Changes the frequency (higher or lower) of the pulse output in one step.	SPED (Continuous) ↓ SPED (Continuous)	• Port • Continuous • 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 or SPED (Continuous) ↓ ACC (Continuous)	• Port • Continuous • Target frequency • Acceleration/ deceleration rate

Operation	Example application	Frequency changes	Description	Procedure	
				Instruction	Settings
	Changing the speed in a polyline curve during operation		Changes the acceleration or deceleration rate during acceleration or deceleration.	ACC (Continuous) ↓ ACC (Continuous)	• Port • Continuous • Target frequency • Acceleration/ deceleration rate
Change direction	Not supported.				

Stopping a Pulse Output

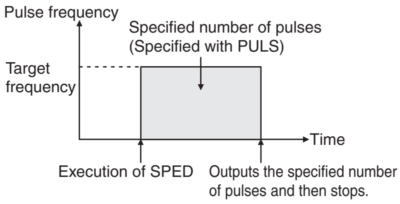
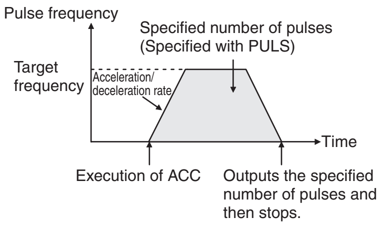
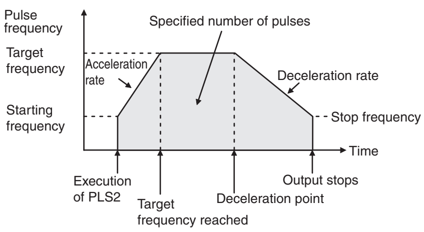
Operation	Example application	Frequency changes	Description	Procedure	
				Instruction	Settings
Stop pulse output	Immediate stop		Stops the pulse output immediately.	SPED or ACC (Continuous) ↓ INI	• Port • Stop pulse output
Stop pulse output	Immediate stop		Stops the pulse output immediately.	SPED (Continuous) ↓ SPED (Continuous)	• Port • Continuous • Target frequency=0
Stop pulse output smoothly	Decelerate to a stop		Decelerates the pulse output to a stop.*	SPED or ACC (Continuous) ↓ ACC (Continuous)	• Port • Continuous • Target frequency=0

* If an ACC instruction started the operation, the original acceleration/deceleration rate will remain in effect.
If a SPED instruction started the operation, the acceleration/deceleration rate will be invalid and the pulse output will stop immediately.

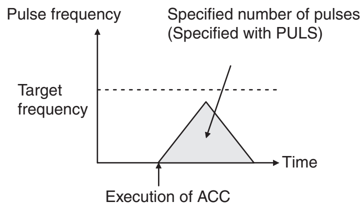
12-11-2 Positioning Control (Independent Mode)

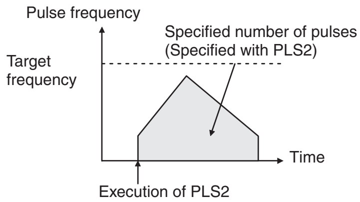
The following operations can be performed in Independent Mode by combining instructions.

Starting a Pulse Output

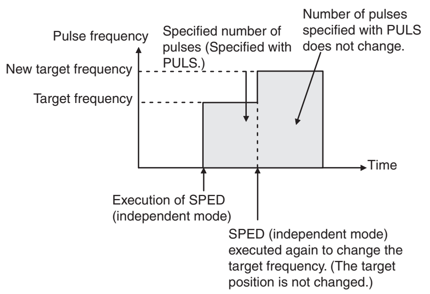
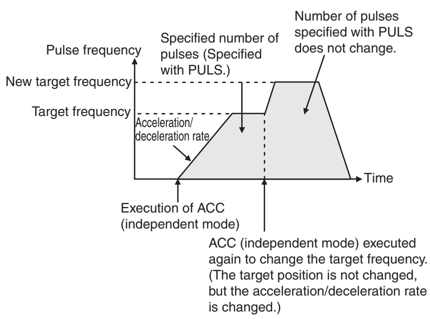
Operation	Example application	Frequency changes	Description	Procedure	
				Instruction	Settings
Output with specified speed	Positioning without acceleration or deceleration		Starts outputting pulses at the specified frequency and stops immediately when the specified number of pulses has been output. The target position (specified number of pulses) cannot be changed during positioning.	PULS ↓ SPED (Independent)	• Number of pulses • Relative or absolute pulse specification • Port • Pulse + Direction • Independent • Target frequency
Simple trapezoidal control	Positioning with trapezoidal acceleration and deceleration (Same rate used for acceleration and deceleration; no starting speed). The number of pulses cannot be changed during positioning.		Accelerates and decelerates at the same fixed rate and stops immediately when the specified number of pulses has been output.*	PULS ↓ ACC (Independent)	• Number of pulses • Relative or absolute pulse specification • Port • Pulse + Direction • Independent • Acceleration and deceleration rate • Target frequency
Complex trapezoidal control	Positioning with trapezoidal acceleration and deceleration (Separate rates used for acceleration and deceleration; starting speed) The number of pulses can be changed during positioning.		Accelerates and decelerates at a fixed rates. The pulse output is stopped when the specified number of pulses has been output.* The target position (specified number of pulses) can be changed during positioning.	PLS2	• Number of pulses • Relative or absolute pulse specification • Port • Pulse + Direction • Acceleration rate • Deceleration rate • Target frequency • Starting frequency

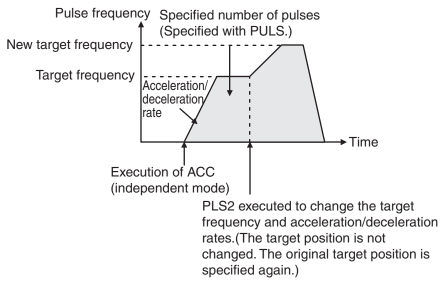
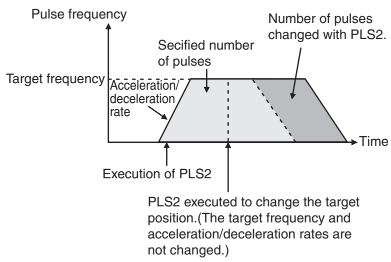
* Triangular Control
If the specified number of pulses is less than the number required just to reach the target frequency and return to zero, the function will automatically reduce the acceleration/deceleration time and perform triangular control (acceleration and deceleration only.) An error will not occur.

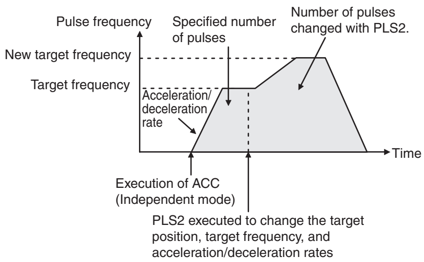
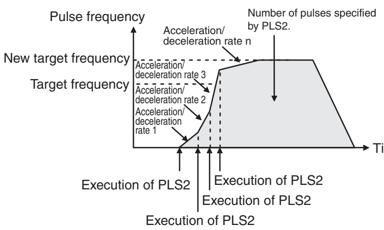
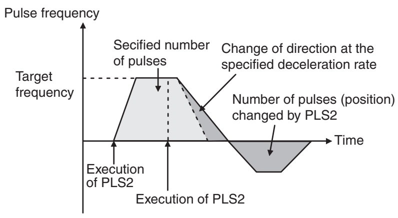




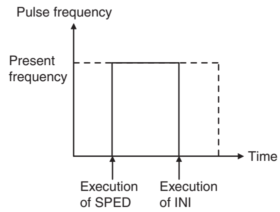
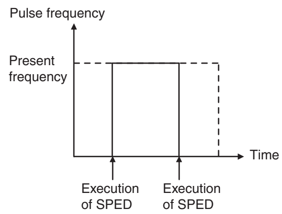
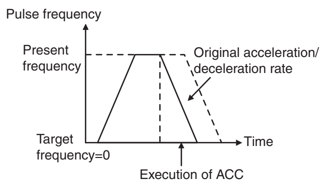
Changing Settings

Operation	Example application	Frequency changes	Description	Procedure	
				Instruction	Settings
Change speed in one step	Changing the speed in one step during operation		SPED can be executed during positioning to change (raise or lower) the pulse output frequency in one step. The target position (specified number of pulses) is not changed.	PULS ↓ SPED (Independent) ↓ SPED (Independent)	• Number of pulses • Relative or absolute pulse specification • Port • Pulse + Direction • Independent • Target frequency
Change speed smoothly (with acceleration rate = deceleration rate)	Changing the target speed (frequency) during positioning (acceleration rate = deceleration rate)		ACC can be executed during positioning to change the acceleration/ deceleration rate and target frequency. The target position (specified number of pulses) is not changed.	PULS ↓ ACC (Independent) ↓ ACC (Independent) PLS2 ↓ ACC (Independent)	• Number of pulses • Relative or absolute pulse specification • Port • Pulse + Direction • Independent • Acceleration/deceleration rate • Target frequency

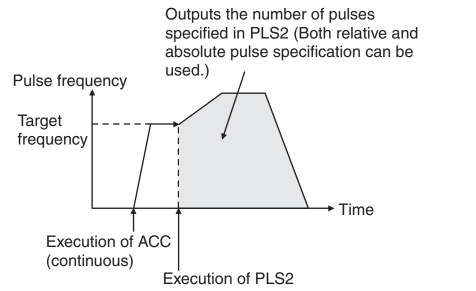
Operation	Example application	Frequency changes	Description	Procedure: Instruction	Procedure: Settings
Change speed smoothly (with unequal acceleration and deceleration rates)	Changing the target speed (frequency) during positioning (different acceleration and deceleration rates)		PLS2 can be executed during positioning to change the acceleration rate, deceleration rate, and target frequency. To prevent the target position from being changed intentionally, the original target position must be specified in absolute coordinates.	PULS ↓ ACC (Independent) ↓ PLS2 ― PLS2 ↓ PLS2	• Number of pulses • Relative or absolute pulse specification • Port • Pulse + Direction • Acceleration rate • Deceleration rate • Target frequency • Starting frequency
Change target position	Change the target position during positioning (multiple start function)		PLS2 can be executed during positioning to change the target position (number of pulses). 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 ↓ ACC (Independent) ↓ PLS2 ― PLS2 ↓ PLS2	• Number of pulses • Relative or absolute pulse specification • Port • Pulse + Direction • Acceleration rate • Deceleration rate • Target frequency • Starting frequency

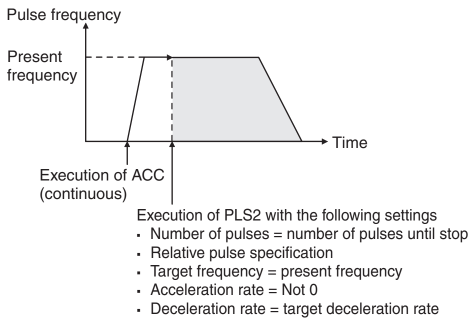
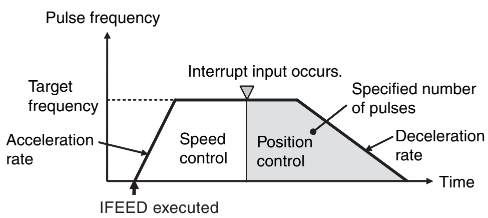
Operation	Example application	Frequency changes	Description	Procedure	
				Instruction	Settings
Change target position and speed smoothly	Change the target position and target speed (frequency) during positioning (multiple start function)		PLS2 can be executed during positioning to change the target position (number of pulses), acceleration rate, deceleration rate, and target frequency. 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.	PULS ↓ ACC (Independent) ↓ PLS2	• Number of pulses • Relative or absolute pulse specification • Port • Pulse + Direction • Acceleration rate • Deceleration rate • Target frequency • Starting frequency
	Change the acceleration and deceleration rates during positioning (multiple start function)		PLS2 can be executed during positioning (acceleration or deceleration) to change the acceleration rate or deceleration rate.	PLS2 ↓ PLS2	• Number of pulses • Acceleration rate • Deceleration rate
Change direction	Change the direction during positioning		PLS2 can be executed during positioning with absolute pulse specification to change to absolute pulses and reverse direction.	PULS ↓ ACC (Independent) ↓ PLS2 PLS2 ↓ PLS2	• Number of pulses • Absolute pulse specification • Port • Pulse + Direction • Acceleration rate • Deceleration rate • Target frequency • Starting frequency

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 ↓ ACC or SPED (Independent) ↓ INI PLS2 ↓ INI	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.	PULS ↓ SPED (Independent) ↓ SPED (Independent)	• Port • Independent • Target frequency = 0
Stop sloped pulse output smoothly. (Number of pulses setting is not preserved.)	Decelerate to a stop		Decelerates the pulse output to a stop. If ACC started the operation, the original acceleration/deceleration rate will remain in effect. If SPED started the operation, the acceleration/deceleration rate will be invalid and the pulse output will stop immediately.	PULS ↓ ACC or SPED (Independent) ↓ ACC (Independent) PLS2 ↓ ACC (Independent)	• Port • Independent • Target frequency = 0

Switching from Speed Control (Continuous Mode) to Positioning (Independent Mode)

<table>
<tr><th rowspan="2">Example application</th><th rowspan="2">Frequency changes</th><th rowspan="2">Description</th><th colspan="2">Procedure</th></tr>
<tr><th>Instruction</th><th>Settings</th></tr>
<tr><td>Change from speed control to fixed distance positioning during operation</td><td>
</td><td rowspan="2">PLS2 can be executed during a speed control operation started with ACC to change to positioning operation.

An error will occur if a constant speed cannot be achieved after switching the mode. If this happens, the instruction execution will be ignored and the previous operation will be continued.</td><td rowspan="2">ACC (Continuous)
↓
PLS2</td><td rowspan="2">• Port
• Acceleration rate
• Deceleration rate
• Target frequency*
• Number of pulses</td></tr>
<tr><td>Fixed distance feed interrupt</td><td>
</td></tr>
<tr><td>High-speed interrupt feeding</td><td>
</td><td>When an interrupt input occurs during speed control for the IFEED instruction, operation changes to positioning. An interrupt task is not used.

There is no delay for the starting time of the interrupt task, improving the feeding accuracy.</td><td>IFEED</td><td>• Port
• Acceleration rate
• Target frequency
• Pulse output set value
• Deceleration rate</td></tr>
</table>

* The starting frequency is ignored.

13

PWM Outputs

This section describes the PWM Outputs (variable-duty-factor pulse outputs).

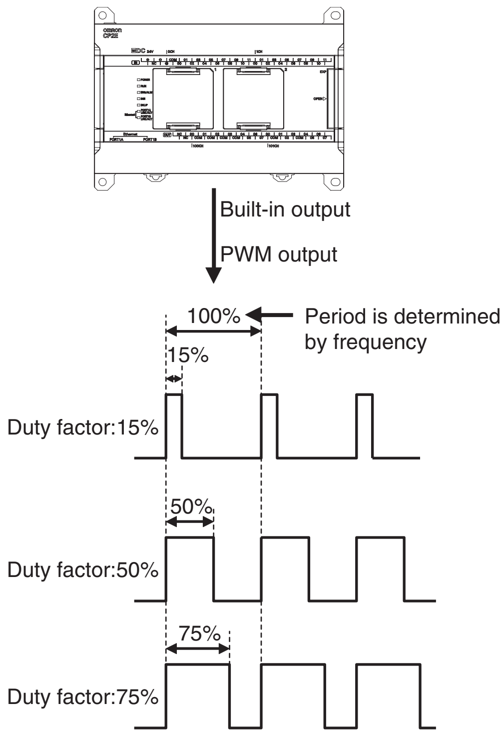
13-1 PWM Outputs (Variable-duty-factor Pulse Outputs)

PWM outputs can be used only with the CP2E N/S☐☐-type CPU Unit with transistor outputs.

A PWM (Pulse Width Modulation) pulse 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 instruction to generate PWM pulses from a built-in output. The duty factor can be changed during pulse output.

Application example

- Controlling temperature on a time-proportional basis using the PWM output.
- Controlling the brightness of lighting.



Specifications

Item	Specification
Duty factor	0.0% to 100.0% in 0.1% increments (Duty factor accuracy is +1%/-0% at 10 kHz, +5%/-0% at 10 to 32 kHz .)
Frequency	2.0 Hz to 6,553.5 Hz (Set in 0.1-Hz increments.)* 2 Hz to 32,000 Hz (Set in 1-Hz increments.)*
Output mode	Continuous mode
Instruction	PWM

* The duty factor accuracy declines significantly at high frequencies because of limitations in the output circuit at high frequencies.

Additional Information

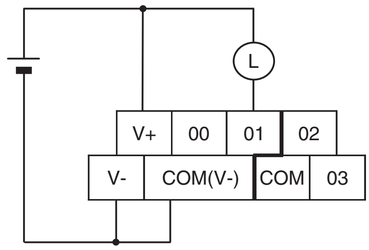
For N30/40/60 CPU Units, the output indicator of terminal 01 on terminal block CIO100 is always lit during PWM output.

Wiring for S☐☐-type CPU Unit

An external power supply is required for S☐☐-type CPU Units when using the PWM output.
Provide a DC24V external power supply to V+ and V- terminals as follows.

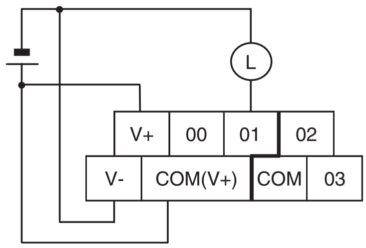
● Wiring Example

Sinking outputs



Although V- and COM(V-) are connected internally, also wire them externally.

Sourcing outputs



Although V+ and COM(V+) are connected internally, also wire them externally.
Do not connect an external power supply to N☐☐-type CPU Units.

13-1-1 Flow of Operation

1 Setting pulse output port number, assigning pulse output terminals, and wiring.

Terminal 01 on terminal block CIO100 is used for PWM output 0.

2 Greate ladder program | Cyclic task, interrupt task.

- The PWM instruction is used to control PWM outputs.
- PWM outputs are stopped with the INI instruction.

● Pulse Output Port Number and Pulse Output Terminals

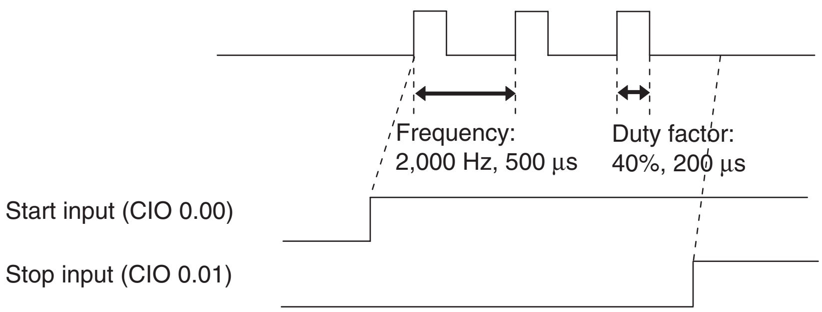
The following terminals can be used for pulse outputs according to the pulse output method.

Output terminal block		Specifications made with PWM instruction	Other functions that cannot be used at the same time	
Terminal block label	Terminal number		Pulse output method	Normal output
			Pulse + direction	
CIO 100	00	–	Pulse output 0, pulse	Normal output 0
	01	PWM output 0	Pulse output 1, pulse	Normal output 1
	02	–	Pulse output 0, direction	Normal output 2
	03	–	Pulse output 1, direction	Normal output 3
CIO 101	00	–	Pulse output 2, pulse	Normal output 8
	01	–	Pulse output 3, pulse	Normal output 9
	02	–	Pulse output 2, direction	Normal output 10
	03	–	Pulse output 3, direction	Normal output 11

13-1-2 Ladder Program Example

Specifications and Operation

When the start input (CIO 0.00) turns ON in this example, pulses with a duty factor of 40% at a frequency of 2,000 Hz are output from PWM output 0. When the stop input (CIO 0.01) turns ON, PWM output 0 is stopped.



Applicable Instructions

PWM

INI

Preparations

● PLC Setup

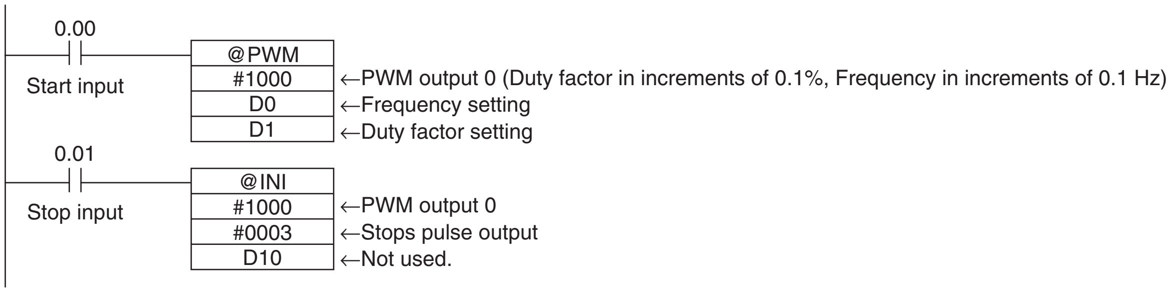
There are no settings that need to be made in the PLC Setup.

● DM Area Settings

- PWM Operand Settings (D0 and D1)

Setting	Operand	Data
Frequency: 2,000.0 Hz	D0	#4E20
Duty factor: 40.0%	D1	#0190

● Ladder Diagram



14

Serial Communications

This section describes communications with Programmable Terminals (PTs) without using communications programming, no-protocol communications with general components, and connections with a Modbus-RTU Easy Master, Modbus-RTU slave, Serial PLC Link, and host computer.

14-1 Serial Communications

Serial communications can be used with any model of CP2E CPU Unit.

14-1-1 Types of CPU Units and Serial Ports

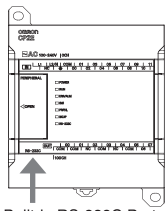
Serial Ports

- **E/S□□-type CPU Unit**

E14/20/30/40/60 CPU Units have only one built-in RS-232C port. There are no option slots.

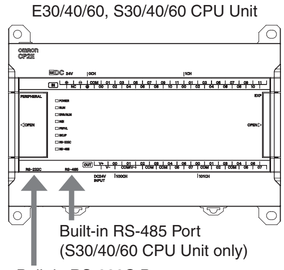
S30/40/60 CPU Units have one built-in RS-232C port and one built-in RS-485 port. There are no option slots.

E14/20 CPU Unit



Built-in RS-232C Port

E30/40/60, S30/40/60 CPU Unit



Built-in RS-485 Port
(S30/40/60 CPU Unit only)
Built-in RS-232C Port

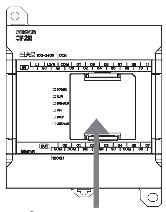
- **N□□-type CPU Unit**

N14/20 CPU Units have only one option slot.

N30/40/60 CPU Units have two option slots.

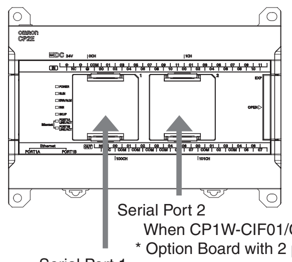
RS-232C or RS-422A/485 Option Boards can be mounted for serial communications.

N14/20 CPU Unit



Serial Port 1
When CP1W-CIF01/CIF11/CIF12-V1, CP2W-CIFD□ Option Board is mounted.
Serial Port 1 (EX)
Only when CP2W-CIFD□ Option Board with 2 ports is mounted.

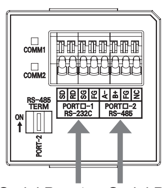
N30/40/60 CPU Unit



Serial Port 2
When CP1W-CIF01/CIF11/CIF12-V1 Option Board is mounted.
* Option Board with 2 ports cannot be used.
Serial Port 1
When CP1W-CIF01/CIF11/CIF12-V1, CP2W-CIFD□ Option Board is mounted.
Serial Port 1 (EX)
Only when CP2W-CIFD□ Option Board with 2 ports is mounted.

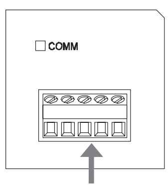
Installation Example

Option Board with 2 ports
CP2W-CIFD□



Serial Port 1 Serial Port 1 (EX)
Option Slot 1

RS-422A/485 Option Board
CP1W-CIF12-V1



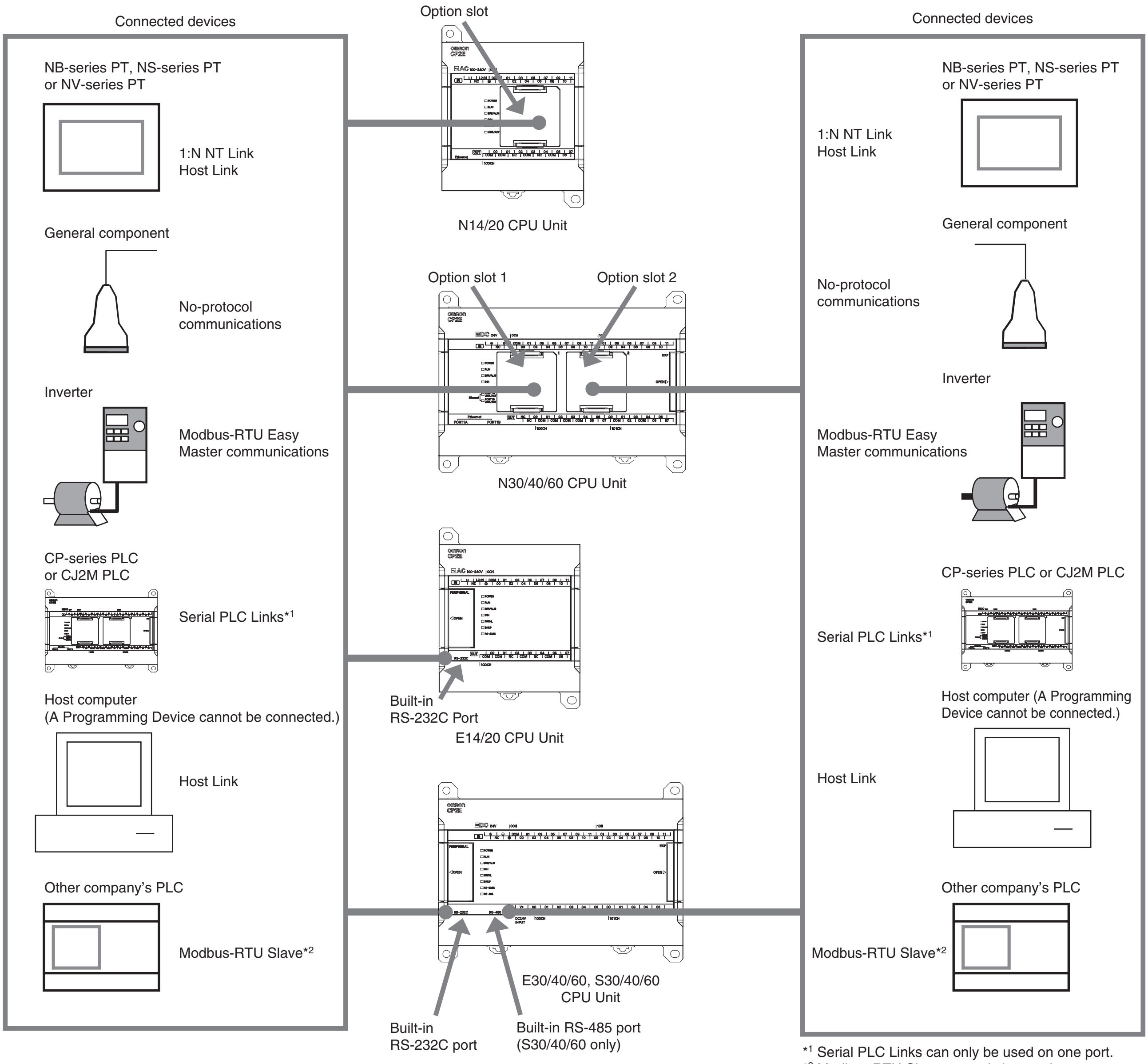
Serial Port 2
Option Slot 2

CP2W-CIFD□ Serial Option Board with two ports can be mounted to option slot 1, so two ports can be controlled by one slot, and a total of three-port serial communications is possible together with option slot 2.

Serial Ports and Compatible Protocols

Protocol \ Serial Port	Serial Port 1 Built-in RS-232C	Serial Port 2 Built-in RS-485	Serial Port 1 (EX)
Host Link	Supported	Supported	Not supported
1:N NT Link	Supported	Supported	Not supported
Serial PLC Links (Master)	Supported	Supported	Supported
Serial PLC Links (Slave)	Supported	Supported	Supported
RS-232C (No-protocol mode)	Supported	Supported	Supported
Modbus-RTU Easy Master	Supported	Supported	Supported
Modbus-RTU Slave	Supported	Supported	Supported

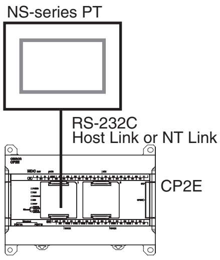
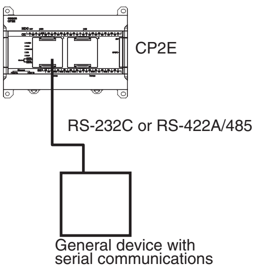
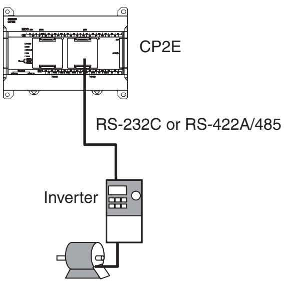
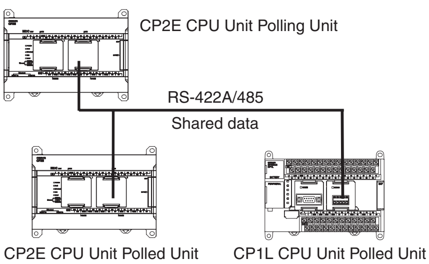
Overview of Connection with Serial Ports

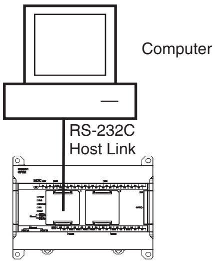
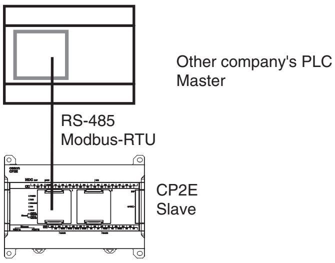


*1 Serial PLC Links can only be used on one port.
*2 Modbus-RTU Slave can only be used on one port.

14-1-2 Overview of Serial Communications

The CP2E CPU Units support the following types of serial communications.

Connected devices	Description	Communications protocol	Built-in RS-232C/ Serial Port 1	Built-in RS-485/ Serial Port 2	Serial Port 1 (EX)
Programmable Terminal 	Data can be exchanged with PTs without using a communications program in the CPU Unit. **Note** Only one PT can be connected when using a 1:N NT Link. It is not possible to connect two PTs.	Host Link or 1:N NT Link	OK	OK	NO
General component 	Communicates with general devices, such as bar-code readers and measuring instruments, with an RS-232C or RS-422A/485 port without a command-response format. The TXD and RXD instructions are executed in the ladder program in the CPU Unit to transmit data from the transmission port or read data in the reception port.	No-protocol communications	OK	OK	OK
Modbus-RTU slave devices, such as inverters 	Data can be easily exchanged with general devices that support Modbus-RTU slave functionality (such as inverters) and are equipped with an RS-232C port or RS-422A/485 port.	Modbus-RTU Easy Master Function	OK	OK	OK
Data links between CPU Units 	Data links can be created for up to nine CP-series or CJ2M CPU Units, including one Polling Unit and up to eight Polled Units. Up to 10 words can be shared per Unit.*1	Serial PLC Links*2	OK	OK	OK

Connected devices	Description	Communications protocol	Built-in RS-232C/ Serial Port 1	Built-in RS-485/ Serial Port 2	Serial Port 1 (EX)
Host computers 	PLC data can be read by the host computer or written to the PLC from the computer. The host computer sends a Host Link command (C Mode) or a FINS command to the CPU Unit to read/write I/O memory, change the operating mode, or to force-set/reset bits in the CPU Unit.	Host Link	OK	OK	NO
Modbus-RTU Master devices 	Data can be read or written to the CP2E by Modbus-RTU master of other company's PLCs.	Modbus-RTU Slave Fuction*2	OK	OK	OK

*1 A PT cannot be included in the Serial PLC Links.

*2 Serial PLC Links and Modbus-RTU Slave cannot be used on two ports at the same time.

Note The Built-in RS-485 port of S□□-type CPU Units and the RS-485 port of CP2W-CIFD2/CIFD3 can only communicate in half duplex.

Additional Information

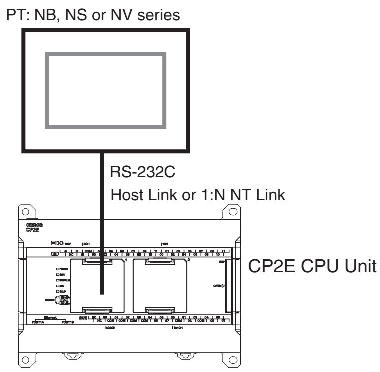
Refer to *A-3 Wiring for Serial Communications* in the *CP2E CPU Unit Hardware User's Manual* (Cat.No.W613) for Serial communication wiring.

14-2 Program-free Communications with Programmable Terminals

14-2-1 Overview

Communications without special communications programming is possible between a CP2E CPU Unit and a Programmable Terminal (PT) by using the Host Link or 1:N NT Link protocol.

Connect the serial port of the CP2E CPU Unit and PT with Host Link or NT Link (1:N) communication mode, and connect the CP2E CPU Unit and PT 1:1 as shown below.



● Connectable Programmable Terminals (PTs)

High-speed Links (115,200 bps) can be used with NB-series, NS-series, or NV-series PTs.

Precautions for Correct Use

- Communications are not possible for CP2E CPU Units using the 1:1 NT Link protocol. Do not connect more than one PT to a CP2E CPU Unit even if the 1:N NT Link protocol is used.
- SAP (Smart Active Parts) on NS-series PTs cannot be used for CP2E CPU Units.

14-2-2 Flow of Connection

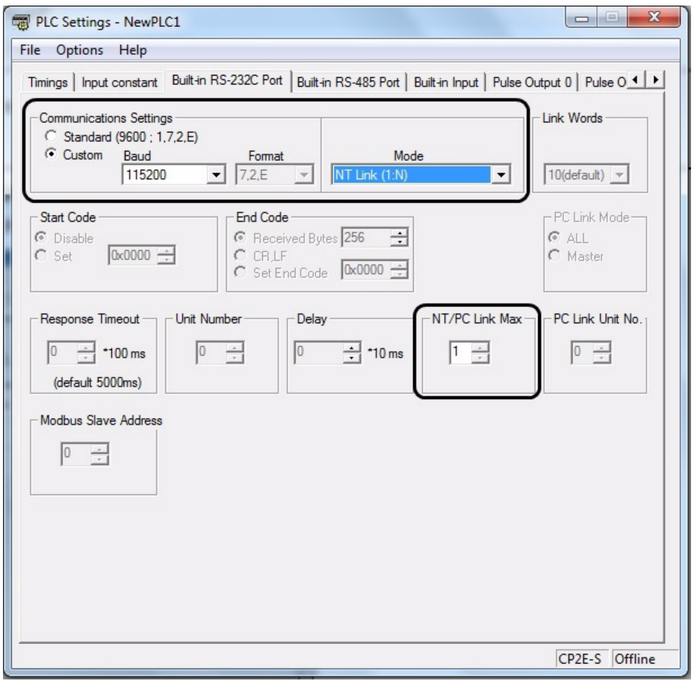
	CP2E CPU Unit		PT (e.g. NB-series)	
1	PLC Setup	Select *Built-in RS-232C*, *Built-in RS-485* or *Serial Port* in the PLC Setup of the CP2E CPU Unit using the CX-Programmer. (Set the serial communication mode to Host Link and set the baud rate.)	CX-Designer	Create a project using the NB-Designer and select *COM1* or *COM2* in the communications settings.
2	Transfer the PLC Setup.		Transfer screen	Transfer screen data created using the NB-Designer to the NB-series PT.
3			NB-series PT System Menu	Check the communications settings in the NB-series PT on the Comm Settings Tab Page in the system menu.
4	Set the same communications settings in the CP2E CPU Unit's PLC Setup and in the NB-series PT.			
5	Connect the CP2E CPU Unit and PT using the RS-232C or RS-422A/485 ports.			

14-2-3 PLC Setup and PT System Settings

Set the parameters in the PLC Setup and the PT's System Menu.

PLC Setup

Click the **Built-in RS-232C**, **Built-in RS-485** or **Serial Port** Tab in the PLC Settings Dialog Box.



Built-in RS-232C, Built-in RS-485 or Serial Port Tab Page

Parameter	Setting
Communications Settings	Select the *Custom Option* and set the baud rate to 115,200 and the format to 7,2,E. (It is not necessary to change the format setting when connecting to NS-series with 1:N NT Link.)
Mode	Select *Host Link.* Select *NT Link (1: N)* for the NS-series.
Unit Number (When connecting with Host Link)	Set to 0.
NT/PC Link Max. (When connecting to NS-series with NT Link)	If only one NS-series PT (unit number 0) is connected, set this parameter to 1. In any other case of NS-series PTs, select the unit number (1 to 7) of the connected NS-series PT.

PT System Menu

Example: NB-series PT

- Set the baud rate to 115,200, the data bit to 7, the parity check to even, and the stop bit to 2 on the COM1 Settings or the COM2 Settings in the PT property.

Example: NS-series PT

1. Select *NT Links (1:N)* from Serial Port A or Serial Port B on the Memory Switch Menu under the System Menu on the PT.
2. Press the **SET** Touch Switch to set the baud rate to high speed.
   A baud rate of 115,200 bps in the PLC Setup is the same as setting high speed for the PT.

Connection with Other Company's Display Devices

Select ***Host Link*** in the serial communications mode settings of the CP2E CPU Unit and set all other communications parameters to the same values as the other company's display device.

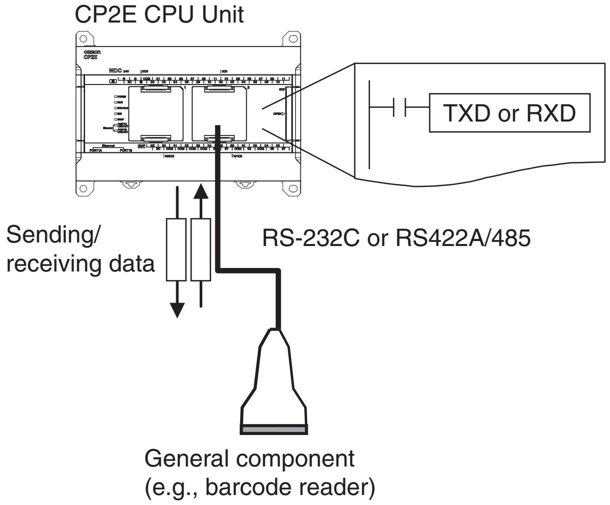
14-3 No-protocol Communications with General Components

14-3-1 Overview

CP2E CPU Units and general devices with serial communications ports can be used for no-protocol communications.

No-protocol communications enable sending and receiving data using the TRANSMIT (TXD) and RECEIVE (RXD) 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 serial communications mode is set to RS-232C.



No-protocol communications are used to send data in one direction to or from general external devices that have an RS-232C or RS-422A/485 port using TXD or RXD.

For example, simple (no-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 CP2E PLCs.

Communica-tions	Transfer direction	Method	Max. amount of data	Frame format		Other functions
				Start code	End code	
Data transmission	PLC → External device	Execution of TXD in the ladder program	256 bytes	Yes: 00 to FF hex No: None	Yes: 00 to FF hex or CR+LF No: None (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 execution and sending data from specified port): 0 to 99,990 ms at the minimum(unit: 10 ms) • Controlling RS and ER signals
Data reception	External device → PLC	Execution of RXD in the ladder program	256 bytes			Monitoring CS and DR signals

Note 1 Because the built-in RS-485 port of the S□□-type CPU Unit and the RS-485 port of CP2W-CIFD2/CIFD3 use 2-wire connections, so they can only communicate in half duplex. Communications are not possible in full duplex.

2 The following serial communication ports cannot control RS and ER signals or monitor CS and DR signals.

- ER and DR signals are not supported by the built-in RS-232C port on the E/S☐☐-type CPU Unit.
- RS, ER, CS and DR signals are not supported by the built-in RS-485 port of the S☐☐-type CPU Unit and CP1W-CIF11/CIF12-V1 Option Board.
- RS, ER, CS and DR signals are not supported by the RS-232C and RS-485 port of CP2W-CIFD1/CIFD2/CIFD3 Option Board with two ports.

14-3-2 Flow of Operation

1 **Wiring communications**

Connect the CP2E CPU Unit and external device using RS-232C or RS-422A/485 ports.

2 **PLC Setup**

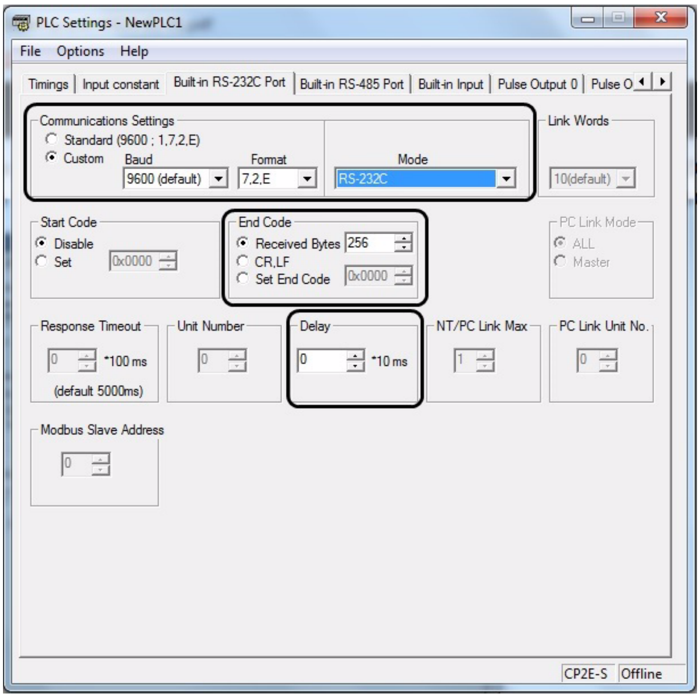
Select *Built-in RS-232C*, *Built-in RS-485* or *Serial Port* in the PLC Setup and transfer the PLC Setup from the CX-Programmer to the CP2E CPU Unit.
(Set the serial communications mode to RS-232C, and set the communications conditions.)

3 **Create ladder Program** (Cyclic tasks / Interrupt tasks)

- PLC to External device: Execute the TXD instruction.
- External device to PLC: Execute the RXD instruction.

14-3-3 PLC Setup

Click the **Built-in RS-232C**, **Built-in RS-485** or **Serial Port** Tab in the PLC Settings Dialog Box.



Built-in RS-232C, Built-in RS-485 or Serial Port Tab Page

Parameter	Setting
Communications Settings	Set the communications settings to the same values as the connected device. If the connected device is set to 9,600 bps, two stop bits, and even parity, select the *Custom* Option, set the baud rate to 9,600 and format to *7,2,E*.
Mode	Select *RS-232C*.
End Code	• To specify the number of bytes of received data, select Received *bytes* and set the number of bytes from 1 to 256. • To use CR+LF as the end code, set CR+LF. • To set the end code to any value between 00 to FF hex, set a value between 0x0000 and 0x00FF.

14-3-4 Related Auxiliary Area Bits and Words

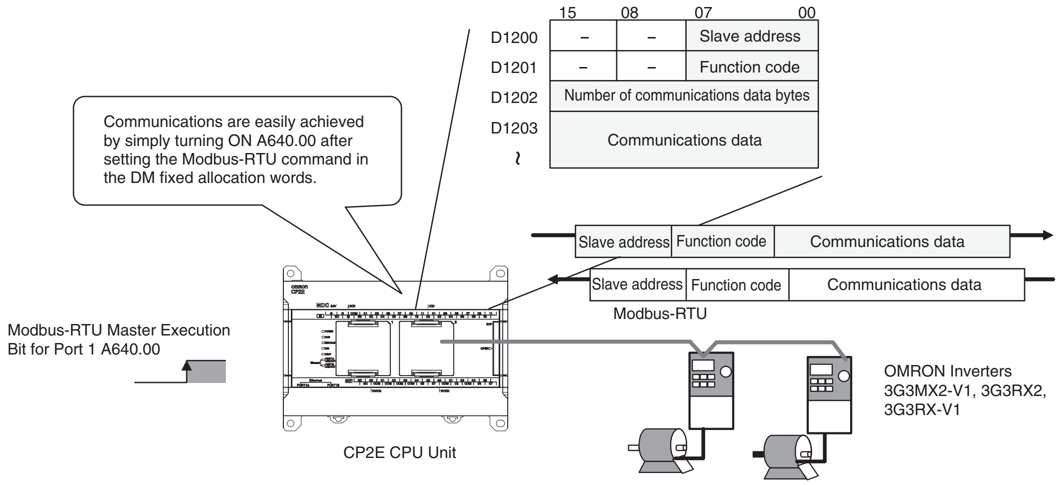
Address	Name	Details
A392.04	Serial Port 1/Built-in RS-232C Port Communications Error Flag	ON when a communications error occurs at the serial port 1 or built-in RS-232C port. The port must be restarted when this flag turns ON.
A392.05	Serial Port 1/Built-in RS-232C Port Send Ready Flag (No-protocol mode)	ON when the serial port 1 or built-in RS-232C port is able to send data in no-protocol mode.
A392.06	Serial Port 1/Built-in RS-232C Port Reception Completed Flag (No-protocol mode)	ON when the serial port 1 or built-in RS-232C port has completed the reception in no-protocol mode. • When the number of bytes was specified: ON when the specified number of bytes is received. • When the end code was specified: ON when the end code is received or 256 bytes are received.
A392.07	Serial Port 1/Built-in RS-232C Port Reception Overflow Flag (No-protocol mode)	ON when a data overflow occurred during reception through the serial port 1 or built-in RS-232C port in no-protocol mode. • When the number of bytes was specified: ON when more data is received after the reception was completed but before RXD was executed. • When the end code was specified: ON when more data is received after the end code was received but before RXD is executed. ON when 257 bytes are received before the end code. If a start code is specified, ON when the end code is received after the start code is received.
A392.12	Serial Port 2/Built-in RS-485 Port Communications Error Flag	ON when a communications error has occurred at the serial port 2 or built-in RS-485 port. The port must be restarted when this flag turns ON.
A392.13	Serial Port 2/Built-in RS-485 Port Send Ready Flag (No-protocol Mode)	ON when the serial port 2 or built-in RS-485 port is able to send data in no-protocol mode.
A392.14	Serial Port 2/Built-in RS-485 Port Reception Completed Flag (No-protocol Mode)	ON when the serial port 2 or built-in RS-485 port has completed the reception in no-protocol mode. • When the number of bytes was specified: ON when the specified number of bytes is received. • When the end code was specified: ON when the end code is received or 256 bytes are received.

Address	Name	Details
A392.15	Serial Port 2/Built-in RS-485 Port Reception Overflow Flag (No-protocol Mode)	ON when a data overflow occurred during reception through the serial port 2 or built-in RS-485 port in no-protocol mode. • When the number of bytes was specified: ON when more data is received after the reception was completed but before RXD was executed. • When the end code was specified: ON when more data is received after the end code was received but before RXD is executed. ON when 257 bytes are received before the end code. If a start code is specified, ON when the end code is received after the start code is received.
A50.04	Serial Port 1 (EX) Communication Error Flag	ON when a communications error occurs at the serial port 1 (EX). The port must be restarted when this Flag turns ON.
A50.05	Serial Port 1 (EX) Send Ready Flag (No-protocol mode)	ON when the serial port 1 (EX) is able to send data in no-protocol mode.
A50.06	Serial Port 1 (EX) Reception Completed Flag (No-protocol mode)	ON when the serial port 1 (EX) has completed the reception in no-protocol mode. • When the number of bytes was specified: ON when the specified number of bytes is received. • When the end code was specified: ON when the end code is received or 256 bytes are received.
A50.07	Serial Port 1 (EX) Reception Overflow Flag (No-protocol mode)	ON when a data overflow occurred during reception through the serial port 1 (EX) in no-protocol mode. • When the number of bytes was specified: ON when more data is received after the reception was completed but before RXD was executed. • When the end code was specified: ON when more data is received after the end code was received but before RXD is executed.
A393.00 to A393.15	Serial Port 1/Built-in RS-232C Port Reception Counter (No-protocol Mode)	• Indicates (in binary) the number of bytes of data received when the serial port 1 or built-in RS-232C port is in no-protocol mode. • The start code and end code are not included.
A394.00 to A394.15	Serial Port 2/Built-in RS-485 Port Reception Counter (No-protocol Mode)	• Indicates (in binary) the number of bytes of data received when the serial port 2 or built-in RS-485 port is in no-protocol mode. • The start code and end code are not included.
A51.00 to A51.15	Serial Port 1 (EX) Reception Counter (No-protocl Mode)	• Indicates (in binary) the number of bytes of data received when the serial port 1 (EX) in no-protocol mode. • The start code and end code are not included.

14-4 Modbus-RTU Easy Master Function

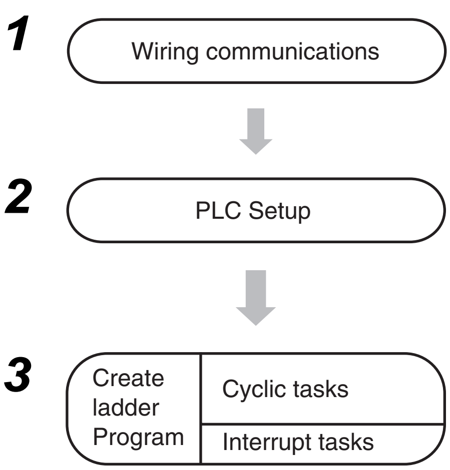
14-4-1 Overview

Using the Modbus-RTU Easy Master enables easy control of Modbus-compatible slaves, such as inverters, using serial communications. The serial communications mode is set to Modbus-RTU Easy Master.



Modbus-RTU commands can be sent 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 automatically stored in the DM fixed allocation words for the Modbus-RTU Easy Master.

14-4-2 Flow of Operation



1 Connect the CP2E CPU Unit and Modbus-RTU Slave using RS-232C or RS-422A/485 ports.

2 Select *Built-in RS-232C*, *Built-in RS-485* or *Serial Port* in the PLC Setup and transfer the PLC Setup from the CX-Programmer to the CP2E CPU Unit. (Set the serial communications mode to Modbus Easy Master, and set the communications conditions.)

3
- Set the Modbus-RTU frame in the DM Fixed Allocation Words.
- Turn ON the Modbus-RTU Master Execution Bit (A640.00, A641.00 or A638.00).

14-4-3 Setting and Word Allocation

DM fixed allocation words and Auxiliary Area words are allocated for the Modbus-RTU Easy Master according to the CPU Unit type and connected port as shown below.

CP2E CPU Unit serial port		DM fixed allocation words	Auxiliary Area bits
CP2E E□□-type CPU Unit	Built-in RS-232C port	D1200 to D1299	A640.00 to A640.02
CP2E S□□-type CPU Unit	Built-in RS-232C port	D1200 to D1299	A640.00 to A640.02
	Built-in RS-485 port	D1300 to D1399	A641.00 to A641.02
CP2E N14/20 CPU Unit	Serial Port 1	D1200 to D1299	A640.00 to A640.02
	Serial Port 1 (EX)	D1400 to D1499	A638.00 to A638.02
CP2E N30/40/60 CPU Unit	Serial Port 1	D1200 to D1299	A640.00 to A640.02
	Serial Port 2	D1300 to D1399	A641.00 to A641.02
	Serial Port 1 (EX)	D1400 to D1499	A638.00 to A638.02

● DM Fixed Allocation Words

Word			Bits	Contents	
Built-in RS232C port of CP2E E/S□□-type CPU Unit or Serial Port 1 of CP2E N□□-type CPU Unit	Built-in RS-485 port of CP2E S□□-type CPU Unit or Serial Port 2 of CP2E N30/40/60 CPU Unit	Serial Port 1 (EX) of CP2E N□□-type CPU Unit			
D1200	D1300	D1400	00 to 07	Command	Slave address (00 to F7 hex)
			08 to 15		Reserved (Always 00 hex.)
D1201	D1301	D1401	00 to 07		Function code
			08 to 15		Reserved (Always 00 hex.)
D1202	D1302	D1402	00 to 15		Number of communications data bytes (0000 to 005E hex)
D1203 to D1249	D1303 to D1349	D1403 to D1449	00 to 15		Communications data (94 bytes maximum)
D1250	D1350	D1450	00 to 07	Response	Slave address (01 to F7 hex)
			08 to 15		Reserved (Always 00 hex.)
D1251	D1351	D1451	00 to 07		Function code
			08 to 15		Reserved
D1252	D1352	D1452	00 to 07		Error code (See error codes in the follow-ing table.)
			08 to 15		Reserved (Always 00 hex.)
D1253	D1353	D1453	00 to 15		Number of response bytes (0000 to 03EA hex)
D1254 to D1299	D1354 to D1399	D1454 to D1499	00 to 15		Response data (92 bytes maximum)

● Error Codes

Code	Description	Description
00 hex	Normal end	–
01 hex	Illegal address	The slave address specified in the parameter is illegal (248 or higher).
02 hex	Illegal function code	The function code specified in the parameter is illegal.
03 hex	Data length overflow	There are more than 94 data bytes.
04 hex	Serial communications mode error	The Modbus-RTU Easy Master function was executed when the serial communications mode was not the Modbus-RTU Easy Master Mode or when the option board is not equipped.
80 hex	Response timeout	A response was not received from the slave.
81 hex	Parity error	A parity error occurred.
82 hex	Framing error	A framing error occurred.
83 hex	Overrun error	An overrun error occurred.
84 hex	CRC error	A CRC error occurred.
85 hex	Incorrect confirmation address	The slave address in the response is different from the one in the request.
86 hex	Incorrect confirmation function code	The function code in the response is different from the one in the request.
87 hex	Response size overflow	The response frame is larger than the storage area (92 bytes).
88 hex	Exception response	An exception response was received from the slave.
89 hex	Service being executed	A service is already being executed (reception traffic congestion).
8A hex	Execution canceled	Executing the service has been canceled.
8F hex	Other error	Other FINS response code was received.

● Related Auxiliary Area Words and Bits

The Modbus-RTU command set in the DM fixed allocation words for the Modbus-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	02	Built-in RS232C port of CP2E E/S□□-type CPU Unit or Serial Port 1 of CP2E N□□-type CPU Unit	Modbus-RTU Master Execution Error Flag ON: Execution error. OFF: Execution normal or still in progress.
	01		Modbus-RTU Master Execution Normal Flag ON: Execution normal. OFF: Execution error or still in progress.
	00		Modbus-RTU Master Execution Bit Turned ON: Execution started ON: Execution in progress. OFF: Not executed or execution completed.
A641	02	Built-in RS-485 port of CP2E S□□-type CPU Unit or Serial Port 2 of CP2E N30/40/60 CPU Unit	Modbus-RTU Master Execution Error Flag ON: Execution error. OFF: Execution normal or still in progress.
	01		Modbus-RTU Master Execution Normal Flag ON: Execution normal. OFF: Execution error or still in progress.
	00		Modbus-RTU Master Execution Bit Turned ON: Execution started ON: Execution in progress. OFF: Not executed or execution completed.

Word	Bit	Port	Contents
A638	02	Serial Port 1 (EX) of CP2E N☐☐-type CPU Unit	Modbus-RTU Master Execution Error Flag ON: Execution error. OFF: Execution normal or still in progress.
	01		Modbus-RTU Master Execution Normal Flag ON: Execution normal. OFF: Execution error or still in progress.
	00		Modbus-RTU Master Execution Bit Turned ON: Execution started ON: Execution in progress. OFF: Not executed or execution completed.

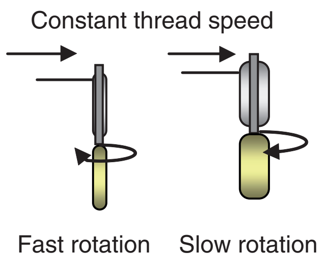
When the Modbus-RTU communication abnormality occurs, the following bits are turned ON.

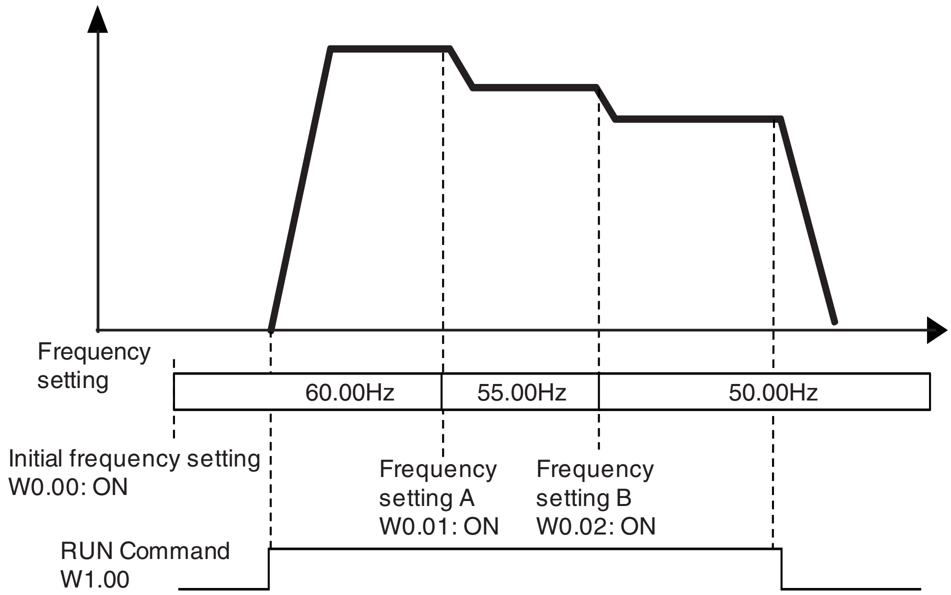
Address	Name	Contents
A392.04	Serial Port 1/Built-in RS-232C Port Communication Error Flag	Turns ON when a timeout error, an overrun error, a framing error or a parity error occurs on Modbus-RTU Easy Master with serial port 1 or built-in RS-232C port.
A392.12	Serial Port 2/Built-in RS-485 Port Communication Error Flag	Turns ON when a timeout error, an overrun error, a framing error or a parity error occurs on Modbus-RTU Easy Master with serial port 2 or built-in RS-485 port.
A50.04	Serial Port 1(EX) Communication Error Flag	Turns ON when a timeout error, an overrun error, a framing error or a parity error occurs on Modbus-RTU Easy Master on serial port 1(EX).

14-4-4 Programming Examples

A bobbin winder on a spinning machine will be used in the following example.

The speed of the bobbin winder must be controlled as the thread is wound because the speed of the thread is constant.



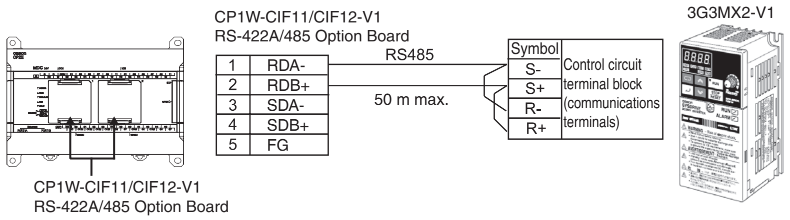


The start/stop and target speed is changed according to inputs from multiple contacts. Acceleration and deceleration are controlled using the acceleration and deceleration of an inverter.

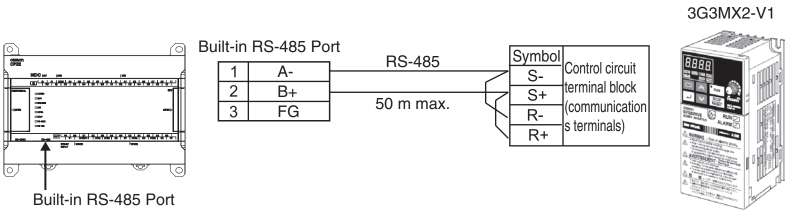
Wiring Examples

The OMRON 3G3MX2-V1 Inverter is connected to the option slot 2 of the CP2E N☐☐-type CPU Unit or the built-in RS-485 port of the CP2E S☐☐-type CPU Unit using RS-485 for frequency and start/stop control.

● When a RS-422A/485 Option Board is Mounted on a CP2E N☐☐-type CPU Unit

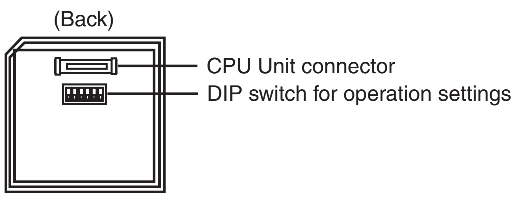


● When Using a Built-in RS-485 Port of CP2E S☐☐-type CPU Unit



● CP1W-CIF11/CIF12-V1 Settings

Set the DIP switch as shown in the following table



No.	Setting	ON / OFF	Description
1	Terminating resistance selection	ON	Connects terminating resistance
2	2/4-wire selection	ON	2-wire connections
3	2/4-wire selection	ON	2-wire connections
4	–	OFF	Always OFF
5	RS control for RD	ON	Enabled
6	RS control for SD	ON	Enabled

● 3G3MX2-V1 Settings

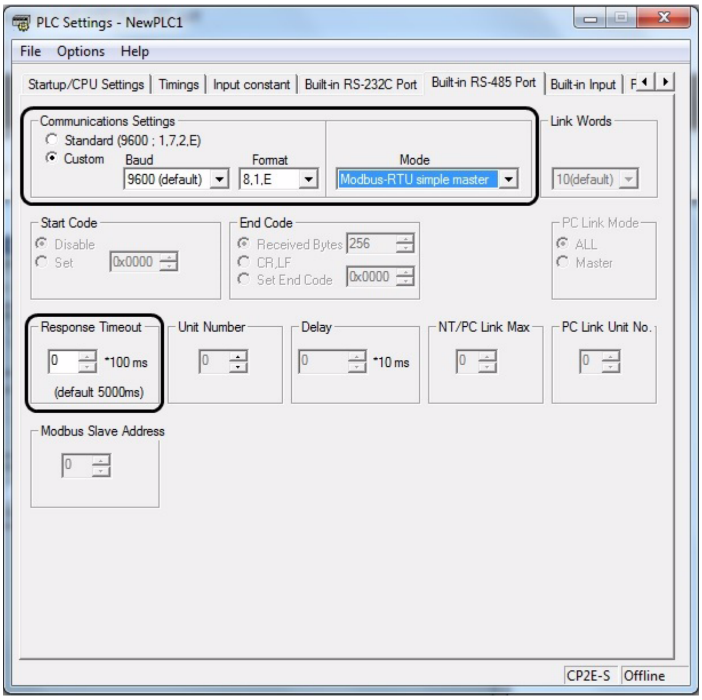
Set the following parameters according to the communication specifications.

As for C071, C074 and C075, modified data are not reflected until the power is reconnected or a reset is performed. To perform a reset, turn the reset terminal (18: RS) OFF, ON and then OFF again.

Parameter No.	Function name	Data	Default setting	Unit	Setting example
A001	Frequency Reference Selection 1	03: Modbus communication (Modbus-RTU)	02	-	03
A002	RUN Command Selection 1	03: Modbus communication (Modbus-RTU)	02	-	03
C071	Communication Speed Selection	03: 2,400 bps	05	-	05 (9600bps)
		04: 4,800 bps			
		05: 9,600 bps			
		06: 19.2 kbps			
		07: 38.4 kbps			
		08: 57.6 kbps			
		09: 76.8 kbps			
		10: 115.2 kbps			
C072	Communication Station No. Selection	1. to 247. 1	1	-	1
C074	Communication Parity Selection	00: No parity	00	-	01 (Even)
		01: Even			
		02: Odd			
C075	Communication Stop Bit Selection	1: 1 bit	1	-	1 (1 bit)
		2: 2 bits			
C076	Operation Selection on Communication Error	00: Trip	02	-	02 (Ignore)
		01: Trip after deceleration stop			
		02: Ignore			
		03: Free run			
		04: Deceleration stop			
C077	Communications Error Timeout Time	0.00: Timeout disabled	0.00	s	0.00s
		0.01 to 99.99			
C078	Communication Wait Time	0. to 1000.	0	ms	0ms

● PLC Setup

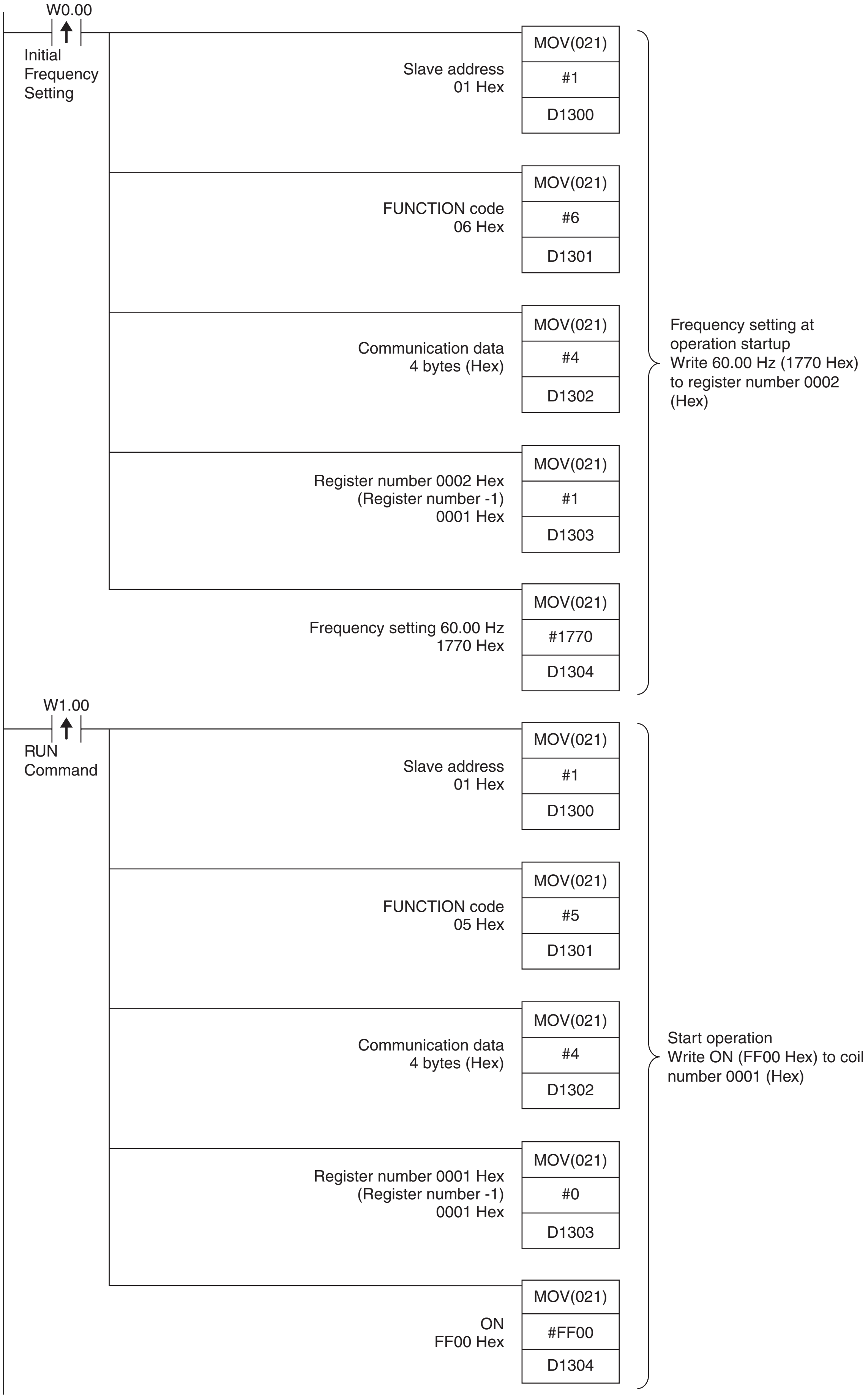
Click the **Built-in RS-232C, Built-in RS-485 or Serial Port** Tab in the PLC Settings Dialog Box.

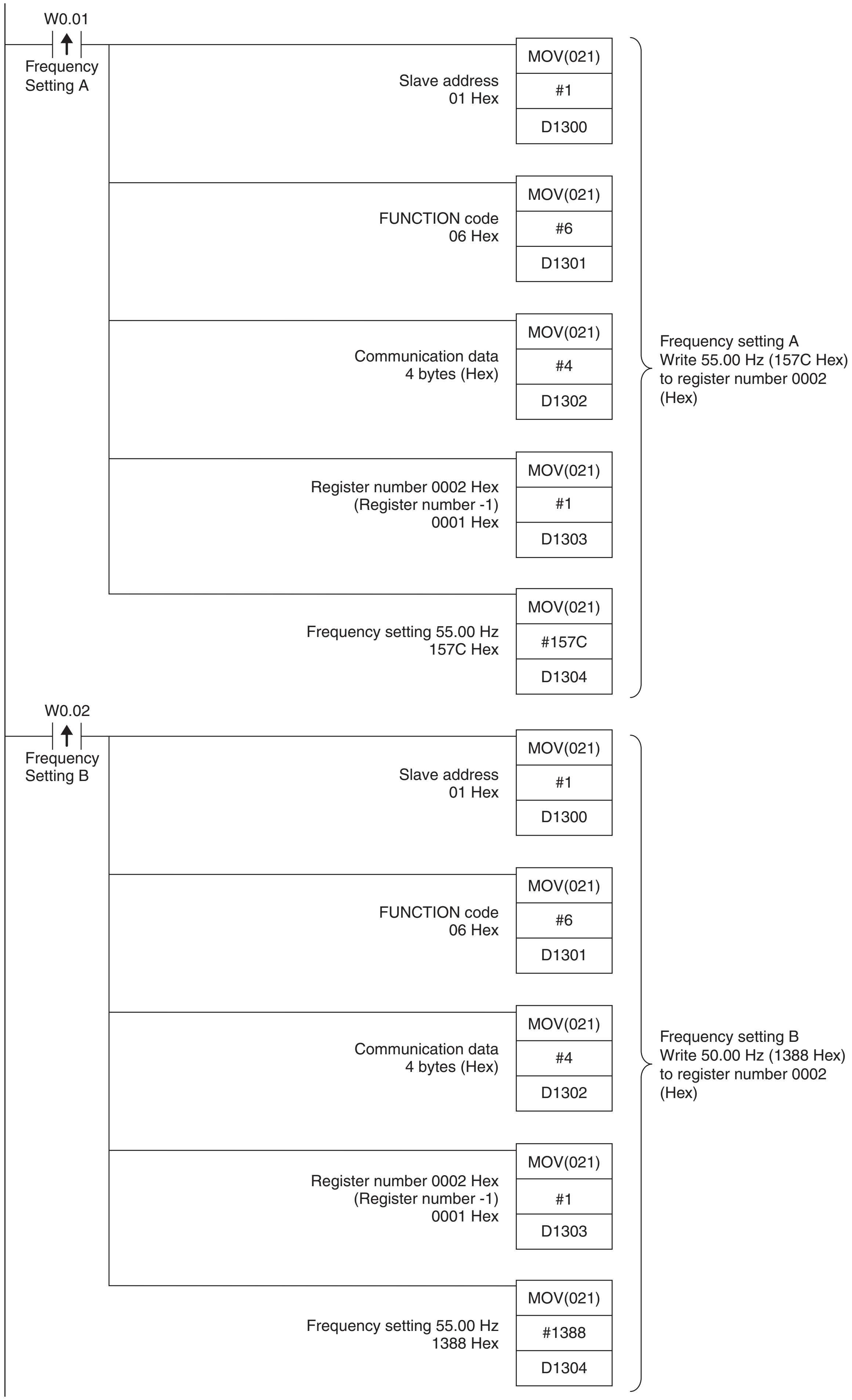


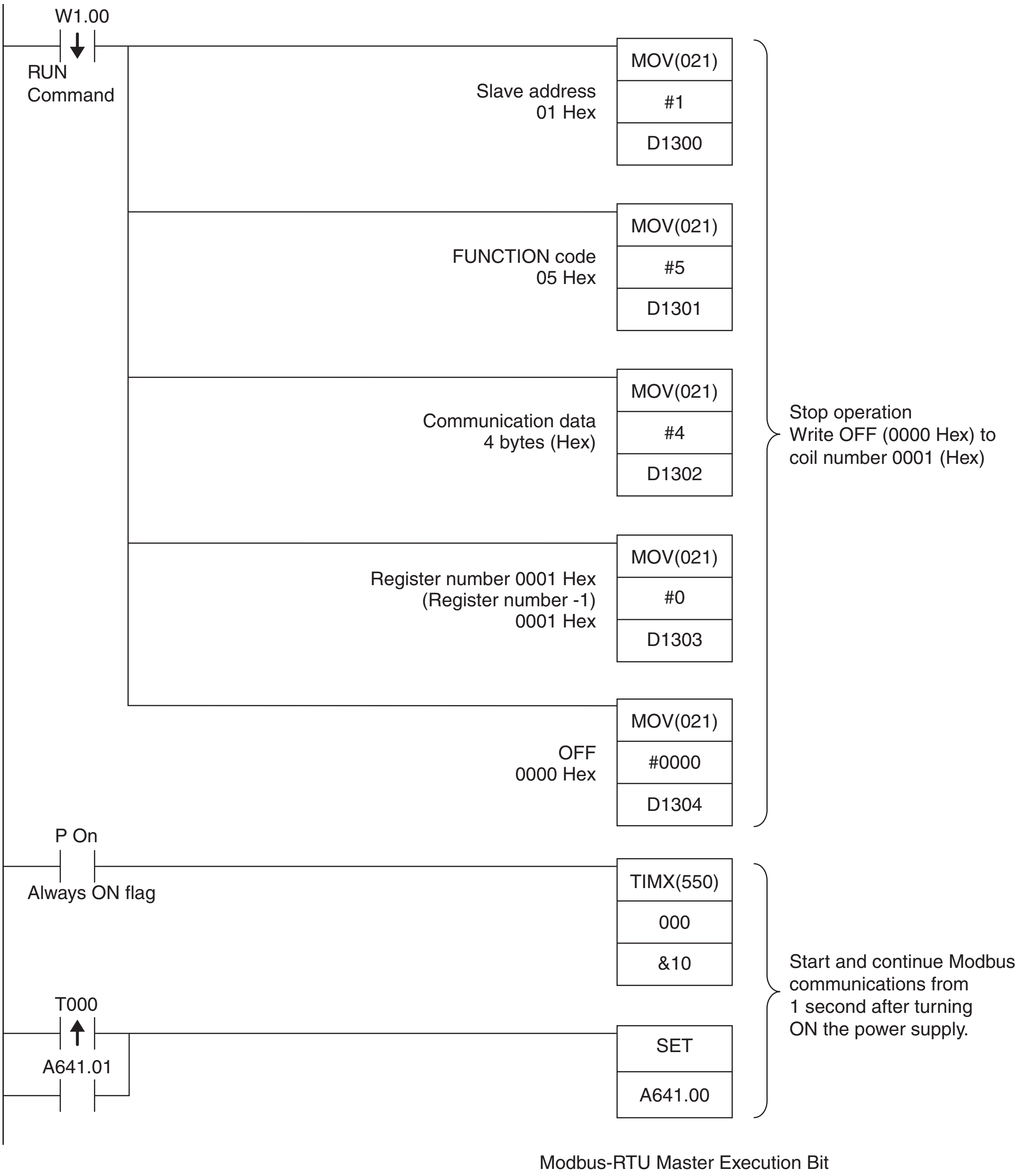
Built-in RS-232C, Built-in RS-485 or Serial Port Tab Page

Parameter	Settings
Communications Settings	Set the Modbus communications settings to match those of the Inverter. If the Inverter is set to 9,600 bps, one stop bit, and even parity, select the *Custom* Option and set the baud rate to 9,600. Set the format to *8,1,E.*
Mode	Select *Modbus-RTU Easy Master.*
Response Timeout	Set the default value of 0×100 ms.

- **Programming Example**

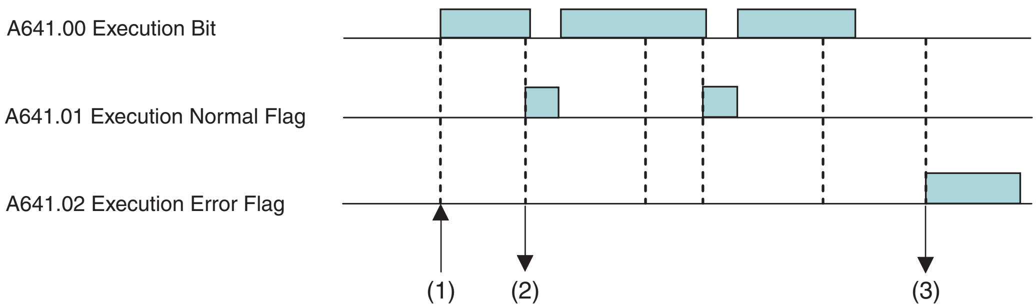



W0.01
Frequency Setting A
Slave address 01 Hex
MOV(021)
#1
D1300
FUNCTION code 06 Hex
MOV(021)
#6
D1301
Communication data 4 bytes (Hex)
MOV(021)
#4
D1302
Register number 0002 Hex (Register number -1) 0001 Hex
MOV(021)
#1
D1303
Frequency setting 55.00 Hz 157C Hex
MOV(021)
#157C
D1304
Frequency setting A Write 55.00 Hz (157C Hex) to register number 0002 (Hex)
W0.02
Frequency Setting B
Slave address 01 Hex
MOV(021)
#1
D1300
FUNCTION code 06 Hex
MOV(021)
#6
D1301
Communication data 4 bytes (Hex)
MOV(021)
#4
D1302
Register number 0002 Hex (Register number -1) 0001 Hex
MOV(021)
#1
D1303
Frequency setting 55.00 Hz 1388 Hex
MOV(021)
#1388
D1304
Frequency setting B Write 50.00 Hz (1388 Hex) to register number 0002 (Hex)



W1.00
RUN
Command
Slave address
01 Hex
MOV(021)
#1
D1300
FUNCTION code
05 Hex
MOV(021)
#5
D1301
Communication data
4 bytes (Hex)
MOV(021)
#4
D1302
Register number 0001 Hex
(Register number -1)
0001 Hex
MOV(021)
#0
D1303
OFF
0000 Hex
MOV(021)
#0000
D1304
Stop operation
Write OFF (0000 Hex) to
coil number 0001 (Hex)
P On
Always ON flag
TIMX(550)
000
&10
T000
A641.01
SET
A641.00
Start and continue Modbus
communications from
1 second after turning
ON the power supply.
Modbus-RTU Master Execution Bit

● Flags for Modbus-RTU Easy Master for Serial Port 2 or Built-in RS-485 Port



(1)Turn ON A641.00 (Execution Bit) to send command data stored starting at D1300. For details, refer to *Data Memory (DM) Area Settings* on page *14-26*.

Words Serial Option Port	Bits	Setting	
D1300	00 to 07	Command	Slave address (00 to F7 hex)
	08 to 15		Reserved (Always 00 hex.)
D1301	00 to 07		Function code
	08 to 15		Reserved (Always 00 hex.)
D1302	00 to 15		Number of communications data bytes (0000 to 005E hex)
D1303 to D1349	00 to 15		Communications data (94 bytes max.)

Precautions for Correct Use

The Execution Bit will automatically turn OFF. Do not turn OFF the bit through the ladder.

(2)When a command has been sent successfully, A641.01 (Execution Normal Flag) will turn ON, and the response data will be stored starting from D1350.

Words Serial Option Port	Bits	Setting	
D1350	00 to 07	Response	Slave address (01 to F7 hex)
	08 to 15		Reserved (Always 00 hex.)
D1351	00 to 07		Function code
	08 to 15		Reserved
D1352	00 to 07		Error code
	08 to 15		Reserved (Always 00 hex.)
D1353	00 to 15		Number of response bytes (0000 to 03EA hex)
D1354 to D1399	00 to 15		Response data (92 bytes max.)

(3)If a communications error occurs, A641.02 (Execution Error Flag) will turn ON, and the error code will be stored in D1352.

- **Data Memory (DM) Area Settings**

DM Fixed Allocation Words for Modbus-RTU Easy Master

The settings are changed by the MOV instruction, and are used to change, start and stop frequency reference.

RUN Command (Example of coil writing)

<table>
<tr><th>Setting</th><th colspan="2">Slave address</th><th colspan="2">FUNCTION code</th><th colspan="2">Communication data bytes</th><th colspan="4">Communication data
D1303 to D1349 (maximum)
94 bytes (47 words) max.</th></tr>
<tr><td>Address</td><td colspan="2">D1300</td><td colspan="2">D1301</td><td colspan="2">D1302</td><td colspan="2">D1303</td><td colspan="2">D1304</td></tr>
<tr><td>Value</td><td>00</td><td>01</td><td>00</td><td>05</td><td>00</td><td>04</td><td>00</td><td>00</td><td></td><td></td></tr>
<tr><td>Description</td><td colspan="2">3G3MX2-V1
Slave address: 01 (Hex)</td><td colspan="2">Coil writing command
05 (Hex)</td><td colspan="2">Communication data
4 bytes (Hex)</td><td colspan="2">Coil number for RUN command: 0001 Hex
* (Coil number) -1</td><td colspan="2">Start: FF00 (Hex)
Stop: 0000 (Hex)</td></tr>
</table>

Output Frequency (Example of holding register writing)

<table>
<tr><th>Setting</th><th colspan="2">Slave address</th><th colspan="2">FUNCTION code</th><th colspan="2">Communication data bytes</th><th colspan="4">Communication data
D1303 to D1349 (maximum)
94 bytes (47 words) max.</th></tr>
<tr><td>Address</td><td colspan="2">D1300</td><td colspan="2">D1301</td><td colspan="2">D1302</td><td colspan="2">D1303</td><td colspan="2">D1304</td></tr>
<tr><td>Value</td><td>00</td><td>01</td><td>00</td><td>06</td><td>00</td><td>04</td><td>00</td><td>01</td><td></td><td></td></tr>
<tr><td>Description</td><td colspan="2">3G3MX2-V1
Slave address: 01 (Hex)</td><td colspan="2">holding register writing command
06 (Hex)</td><td colspan="2">Communication data
4 bytes (Hex)</td><td colspan="2">Output frequency (lower) register number: 0002 Hex
* (Register number) -1</td><td colspan="2">Frequency setting value
Set in 0.01 Hz
Set to 60.00 Hz, it is 1770 (Hex)</td></tr>
</table>

For details on 3G3MX2-V1 parameters and Modbus-RTU communication data, refer to the *3G3MX2-V1 User's Manual* (I585).

With the Modbus-RTU Easy Master, a CRC-16 checksum does not need to be set in the DM Area, because it is calculated automatically.

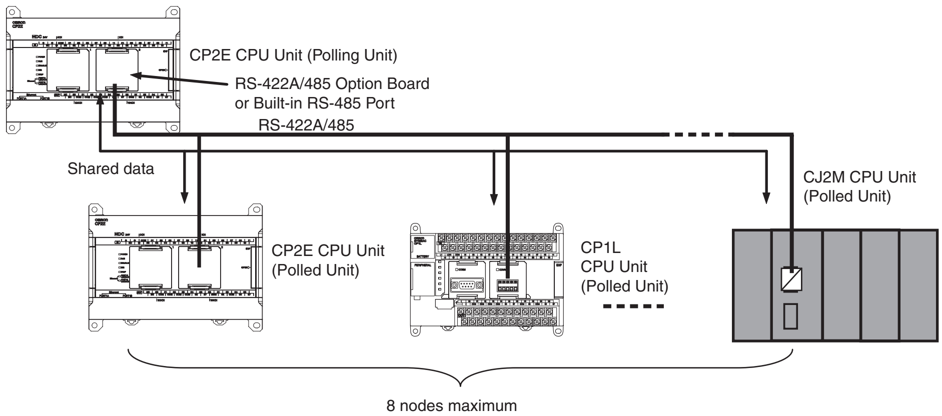
14-5 Serial PLC Links

14-5-1 Overview

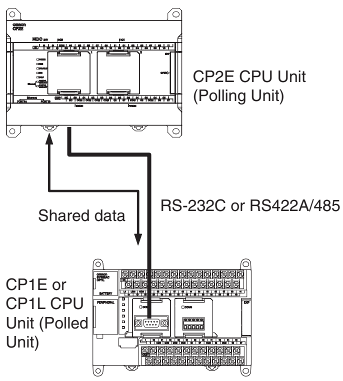
Serial PLC Links enable exchanging data between CP2E CPU Units, CP1H/CP1L/CP1E CPU Units, or CJ2M CPU Units without using special programming. The serial communications mode is set to Serial PLC Links. Up to 9 PLCs can be linked.

Configuration

- **Connecting CP2E, CP1H, CP1L, CP1E or CJ2M CPU Units 1:N (8 Nodes Maximum)**



- **Connecting CP2E, CP1H, CP1L, CP1E or CJ2M CPU Units 1:1**



Precautions for Correct Use

With the CP2E CPU Units, a Programmable Terminal (PT) cannot be included in a Serial PLC Link.

14-5-2 Flow of Operation

1 Wiring communications — Connect the CP2E CPU Unit and other CPU Units using RS-232C or RS-422A/485 ports.

↓

2 PLC Setup — Set *Built-in RS-232C*, *Built-in RS-485* or *Serial Port* in the PLC Setup and transfer the PLC Setup from the CX-Programmer to the CP2E CPU Unit. (Set the serial communications mode to *Serial PC Link (Master)* or *Serial PC Link (Slave)* and set the communications conditions, link words, and PLC Link method.)

↓

3 Start communications

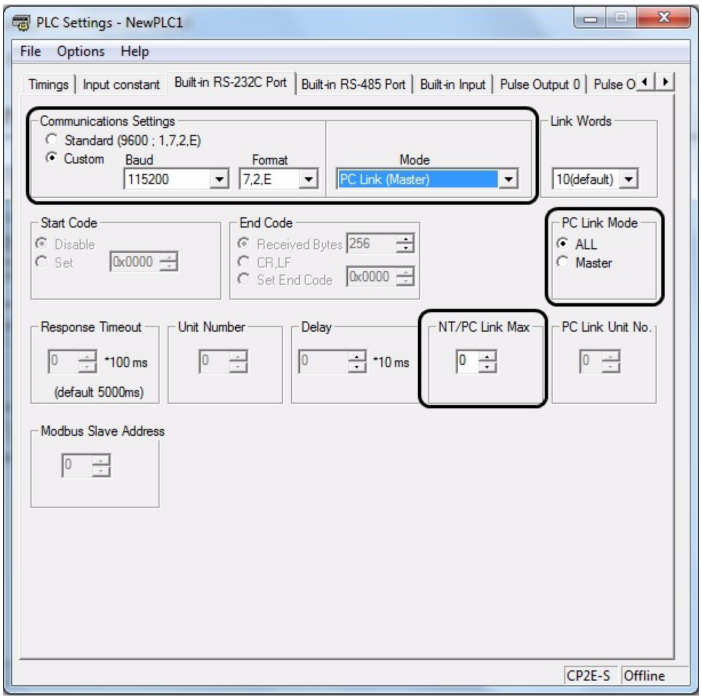
Precautions for Correct Use

More than two serial ports cannot be used for PLC Links at the same time.

If more than two serial ports are set for PLC Links (either as polling or polled nodes), a PLC Setup setting error (nonfatal error) will occur and the PLC Setup Setting Error Flag (A402.10) will turn ON.

14-5-3 PLC Setup

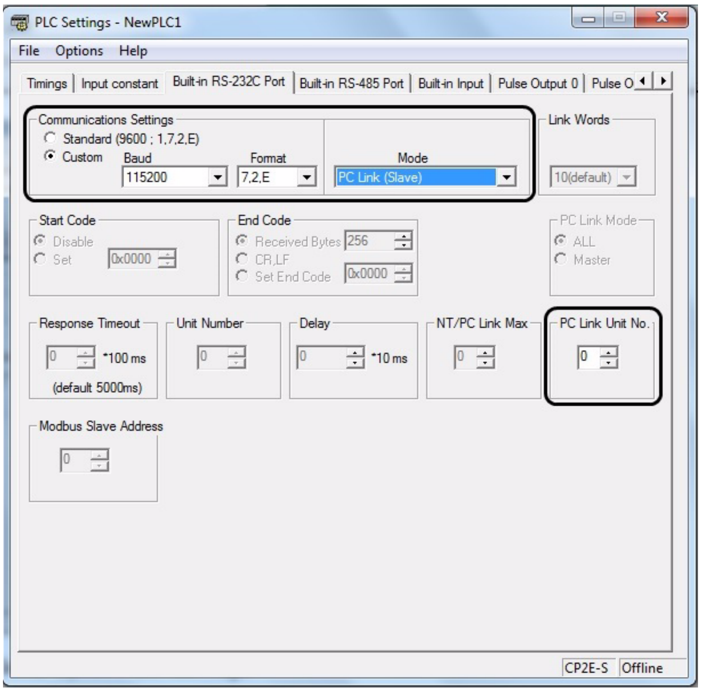
Settings at the Polling Unit



Built-in RS-232C, Built-in RS-485 or Serial Port Tab Page

Parameter	Setting
Communications Settings	Set the communications settings to the same values as the connected PLCs. If the connected PLCs are set to 115,200 bps, select the *Custom* Option, set the baud rate to 115200. The format can be set to any value.
Mode	Select *PC Link (Master).*
Link Words	Set to 10 (default) for the Master only. 10 words (default)
PC Link Mode	Select *All* or *Master.*
NT/PC Link Max.	Set the highest unit number of the connected slaves.

Settings at the Polled Unit



Built-in RS-232C, Built-in RS-485 or Serial Port Tab Page

Parameter	Setting
Communications Settings	Set the communications settings to the same values as the connected PLCs. If the connected PLCs are set to 115,200 bps, select the *Custom* Option, set the baud rate to 115200. The format can be set to any value.
Mode	Select *PC Link (Slave)*.
PC Link Unit No.	Set the unit number (0 to 7).

14-5-4 Operating Specifications

Serial PLC Links can be used for all the serial communication ports for CP2E CPU Units. However, three serial ports cannot be used simultaneously for Serial PLC Links.

Item	Specifications
Applicable PLCs	CP2E, CP1H, CP1L, CP1E, CJ2M
Baud rate	38,400 bps, 115,200 bps
Applicable serial ports	Built-in RS-232C ports, built-in RS-485 ports or serial option ports If more than two ports are set for Serial PLC Links (either as polling node or polled node), a PLC Setup setting error (nonfatal error) will occur and the PLC Setup Setting Error Flag (A402.10) will turn ON.
Connection method	RS-422A/485 or RS-232C connection via RS-422A/485 Option Board, built-in RS-485 port or RS-232C port.
Words allocated in CIO Area	Serial PLC Link Words: CIO 200 to CIO 289 (Up to 10 words can be allocated for each CPU Unit.)
Maximum number of Units	9 Units max., comprising 1 Polling Unit and 8 Polled Units.
Link methods (data refresh methods)	Complete link method or Polling Unit link method

Data Refresh Methods

The following two methods can be used to refresh data.

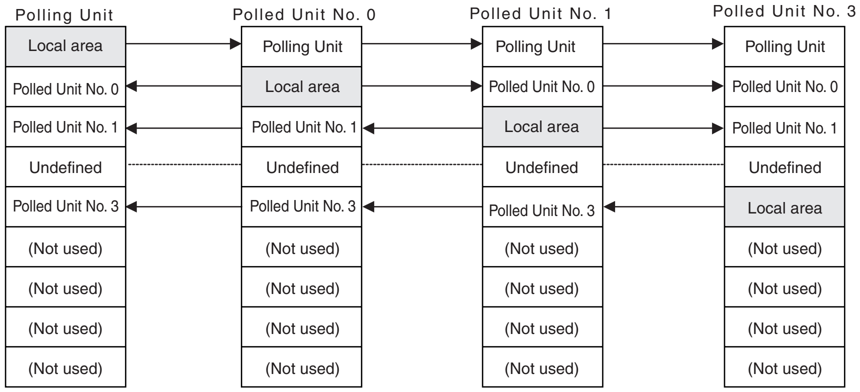
- Complete link method
- Polling Unit link method

● Complete Link

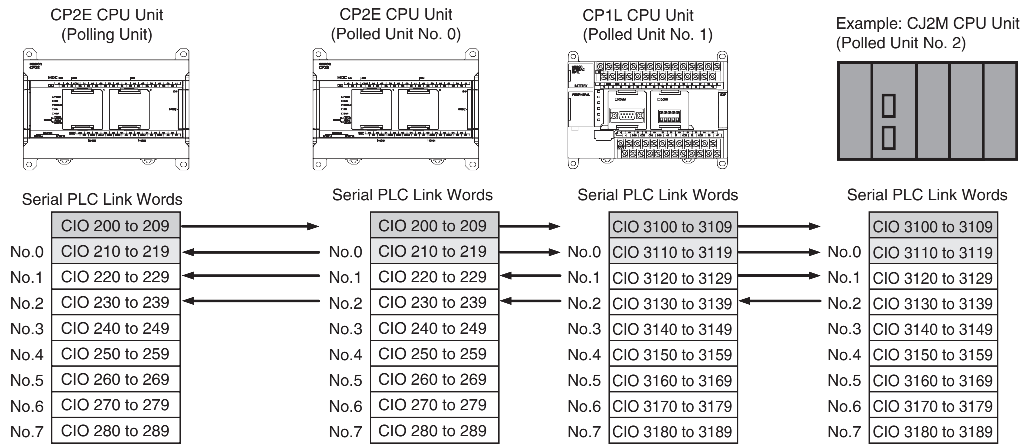
The data from all nodes in the Serial PLC Links are reflected in both the Polling Unit and the Polled Units.

The only exceptions are 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 a Unit not present in the network, so the area allocated for Polled Unit No. 2 is undefined in all nodes.



Example for Ten Link Words (Maximum Number of Words)
Each CPU Unit (either CP2E, CP1E, CP1L, CP1H, or CJ2M) sends data to the same words in all other CPU Units for the Polling Unit and all Polled Units. Data is sent between the words that are allocated to the Polling Unit and Polled Units according to unit numbers.



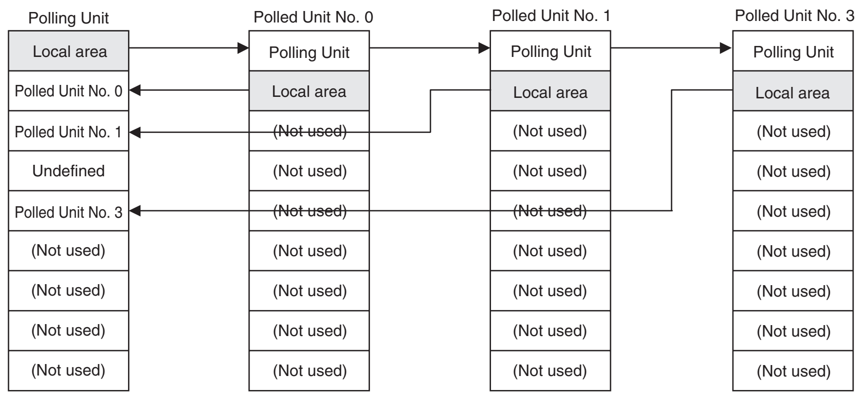
● Polling Unit Link Method

The data for all the Polled Units in the Serial PLC Links are reflected in the Polling Unit only, and each Polled Unit reflects the data of the Polling Unit only.

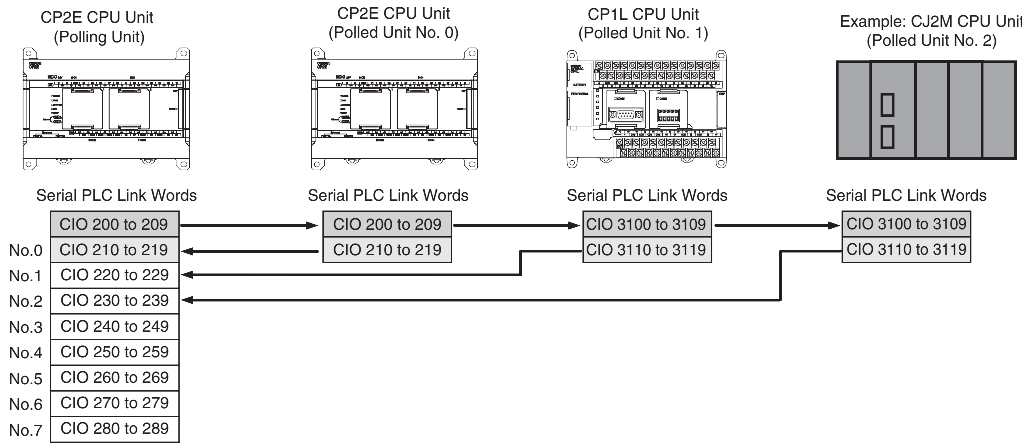
The advantage of the Polling Unit link method is that the addresses allocated for the local Polled Unit data are the same in each Polled Unit, allowing data to be accessed using common ladder programming.

The areas allocated for 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 Unit not participating in the network, so the corresponding area in the Polling Unit is undefined.



Example for Ten Link Words (Maximum Number of Words)
The CPU Unit that is the Polling Unit (either CP2E, CP1E, CP1H, CP1L, or CJ2M) sends its data (CIO 200 to CIO 209) to the same words (CIO 200 to CIO 209) in all other CPU Units.
The Polled Units (either CP2E, CP1E, CP1H, CP1L, or CJ2M) send their data (CIO 210 to CIO 219) to consecutive sets of 10 words (CIO 210 to CIO 289) in the Polling Unit.



● Allocated Words

Complete Link Method

Address

CIO 200

Serial PLC Link Area

CIO 289

Link words	1 word	2 words	3 words	to	10 words
Polling Unit	CIO 200	CIO 200 to 201	CIO 200 to 202		CIO 200 to 209
Polled Unit No. 0	CIO 201	CIO 202 to 203	CIO 203 to 205		CIO 210 to 219
Polled Unit No. 1	CIO 202	CIO 204 to 205	CIO 206 to 208		CIO 220 to 229
Polled Unit No. 2	CIO 203	CIO 206 to 207	CIO 209 to 211		CIO 230 to 239
Polled Unit No. 3	CIO 204	CIO 208 to 209	CIO 212 to 214		CIO 240 to 249
Polled Unit No. 4	CIO 205	CIO 210 to 211	CIO 215 to 217		CIO 250 to 259
Polled Unit No. 5	CIO 206	CIO 212 to 213	CIO 218 to 220		CIO 260 to 269
Polled Unit No. 6	CIO 207	CIO 214 to 215	CIO 221 to 223		CIO 270 to 279
Polled Unit No. 7	CIO 208	CIO 216 to 217	CIO 224 to 226		CIO 280 to 289
Not used.	CIO 209 to 289	CIO 218 to 289	CIO 227 to 289		---

Polling Unit Link Method

Address

CIO 200

Serial PLC Link Words

CIO 289

Link words	1 word	2 words	3 words	to	10 words
Polling Unit	CIO 200	CIO 200 to 201	CIO 200 to 202		CIO 200 to 209
Polled Unit No. 0	CIO 201	CIO 202 to 203	CIO 203 to 205		CIO 210 to 219
Polled Unit No. 1	CIO 201	CIO 202 to 203	CIO 203 to 205		CIO 210 to 219
Polled Unit No. 2	CIO 201	CIO 202 to 203	CIO 203 to 205		CIO 210 to 219
Polled Unit No. 3	CIO 201	CIO 202 to 203	CIO 203 to 205		CIO 210 to 219
Polled Unit No. 4	CIO 201	CIO 202 to 203	CIO 203 to 205		CIO 210 to 219
Polled Unit No. 5	CIO 201	CIO 202 to 203	CIO 203 to 205		CIO 210 to 219
Polled Unit No. 6	CIO 201	CIO 202 to 203	CIO 203 to 205		CIO 210 to 219
Polled Unit No. 7	CIO 201	CIO 202 to 203	CIO 203 to 205		CIO 210 to 219
Not used.	CIO 202 to 289	CIO 204 to 289	CIO 206 to 289		---

Related Auxiliary Area Bits and Words

Serial Port1/Built-in RS-232C Port

Name	Address	Details	Read/write	Refresh timing
Serial Port 1/Built-in RS-232C Port Communicating with Polled Unit Flags*	A393.00 to A393.07	When the serial port 1 or built-in RS-232C port 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. ON: Communicating OFF: Not communicating	Read	• Cleared when power is turned ON. • Turns ON the bit corresponding to the unit number of the Polled Unit that is communicating via the serial port 1 or built-in RS-232C port in NT link mode or Serial PLC Link mode. • Bits 00 to 07 correspond to unit numbers 0 to 7, respectively.
Serial Port 1/Built-in RS-232C Port Restart Bit	A526.00	Turn ON this bit to restart the serial port 1 or built-in RS-232C port.	Read/write	• Cleared when power is turned ON. • Turn ON to restart the serial port 1 or built-in RS-232C port. **Note** The bit is automatically turned OFF by the system when restart processing has been completed.
Serial Port 1/Built-in RS-232C Port Error Flags	A528.00 to A528.07	When an error occurs at the serial port 1 or built-in RS-232C port, the corresponding error bit is turned ON. Bit 0: Not used. Bit 1: Not used. Bit 2: Parity error Bit 3: Framing error Bit 4: Overrun error Bit 5: Timeout error Bit 6: Not used. Bit 7: Not used.	Read/write	• Cleared when power is turned ON. • When an error occurs at the serial port 1 or built-in RS-232C port, the corresponding error bit is turned ON. • The flag is automatically turned OFF by the system when the serial port 1 or built-in RS-232C port is restarted. • In NT link mode, only bit 05 (timeout error) is enabled. • In Serial PLC Link mode, only the following bits are enabled. Errors at the Polling Unit: Bit 05: Timeout error Errors at Polled Units: Bit 04: Overrun error Bit 03: Framing error Bit 02: Parity error **Note** If the error occurred in Serial PLC link mode, the console will retry before communication establish. Rehabilitation of the communications is no need for port restart. If user eliminates error, the communication will automatically establish between console and servo. However, error flag will be saved as the record. If you want to clear the error flag, please restart port.

Serial Port 2/Built-in RS-485 Port

Name	Address	Details	Read/write	Refresh timing
Serial Port 2/ Built-in RS-485 Port Communicating with Polled Unit Flags*	A394.00 to A394.07	When the serial port 2 or built-in RS-485 port 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. ON: Communicating OFF: Not communicating	Read	• Cleared when power is turned ON. • Turns ON the bit corresponding to the unit number of the Polled Unit that is communicating via the serial port 2 or built-in RS-485 port in NT link mode or Serial PLC Link mode. • Bits 00 to 07 correspond to unit numbers 0 to 7, respectively.
Serial Port 2/ Built-in RS-485 Port Restart Flags	A526.01	Turn ON this bit to restart the serial port 2 or built-in RS-485 port.	Read/write	• Cleared when power is turned ON. • Turn ON to restart the serial port 2 or built-in RS-485 port. **Note** The bit is automatically turned OFF by the system when restart processing has been completed.
Serial Port 2/ Built-in RS-485 Port Error Flags	A528.08 to A528.15	When an error occurs at the serial port 2 or built-in RS-485 port, the corresponding error bit is turned ON. Bit 8: Not used. Bit 9: Not used. Bit 10: Parity error Bit 11: Framing error Bit 12: Overrun error Bit 13: Timeout error Bit 14: Not used. Bit 15: Not used.	Read/Write	• Cleared when power is turned ON. • When an error occurs at the serial port 2 or built-in RS-485 port, the corresponding error bit is turned ON. • The flag is automatically turned OFF by the system when the serial port 2 or built-in RS-485 port is restarted. • In NT link mode, only bit 13 (timeout error) is enabled. • In Serial PLC Link mode, only the following bits are enabled. Errors at the Polling Unit: Bit 13: Timeout error Errors at Polled Units: Bit 12: Overrun error Bit 11: Framing error Bit 10: Parity error **Note** If the error occurred in Serial PLC link mode, the console will retry before communication establish. Rehabilitation of the communications is no need for port restart. If user eliminates error, the communication will automatically establish between console and servo. However, error flag will be saved as the record. If you want to clear the error flag, please restart port.

Serial Port 1 (EX)

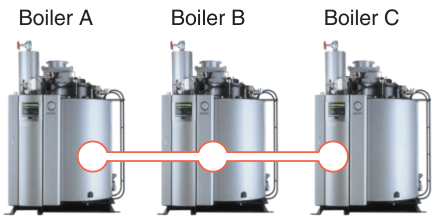
Name	Address	Details	Read/write	Refresh timing
Serial Port 1 (EX) Communication with Polled Unit Flags*	A51.00 to A51.07	When the serial port 1 (EX) is being used in PLC link mode, the bit corresponding to the Unit performing communications will be ON. Bits 00 to 07 correspond to unit numbers 0 to 7, respectively. ON: Communicating OFF: Not communicating	Read	• Cleared when power is turned ON. • Turns ON the bit corresponding to the unit number of the Polled Unit that is communicating via the serial port 1 (EX) in Serial PLC Link mode. • Bits 00 to 07 correspond to unit numbers 0 to 7, respectively.
Serial Port 1 (EX) Restart Bit	A526.02	Turn ON this bit to restart the serial port 1 (EX).	Read/write	• Cleared when power is turned ON. • Turn ON to restart the serial port 1 (EX). **Note** The bit is automatically turned OFF by the system when restart processing has been completed.
Serial Port 1 (EX) Error Flags	A521.00 to A521.08	When an error occurs at the serial port 1 (EX), the corresponding error bit is turned ON. Bit 0: Not used. Bit 1: Not used. Bit 2: Parity error Bit 3: Framing error Bit 4: Overrun error Bit 5: Timeout error Bit 6: Not used. Bit 7: Not used.	Read/Write	• Cleared when power is turned ON. • When an error occurs at the serial port 1 (EX), the corresponding error bit is turned ON. • The flag is automatically turned OFF by the system when the serial port 1 (EX) is restarted. • In Serial PLC Link mode, only the following bits are enabled. Errors at the Polling Unit: Bit 05: Timeout error Errors at Polled Units: Bit 04: Overrun error Bit 03: Framing error Bit 02: Parity error **Note** If the error occurred in Serial PLC link mode, the console will retry before communication establish. Rehabilitation of the communications is no need for port restart. If user eliminates error, the communication will automatically establish between console and servo. However, error flag will be saved as the record. If you want to clear the error flag, please restart port.

* In the same way as for the 1:N NT Link, the status (communicating/not communicating) of the Polled Unit in Serial PLC Links can be checked from the Polling Unit (CPU Unit) by reading the Serial Port 1 Communicating with Polled Unit Flag (A393.00 to A393.07 for unit numbers 0 to 7) , the Serial Port 1 (EX) Communicating with Polled Unit Flag (A51.00 to A51.17 for unit number of 0 to 7) or the Serial Port 2 Communicating with Polled Unit Flag (A394.00 to A394.07 for unit numbers 0 to 7).
When errors occur in the Serial PLC Links communication, PC Link Master station would retry before the communication reestablished. So it is not necessary for users to restart the port for restoring the communication. If the errors have been removed, the communication between PC Link (Master) and PC Link (Slave) will be reestablished automatically. But the error flags remained. Please restart the port if you want to clear the error flags.

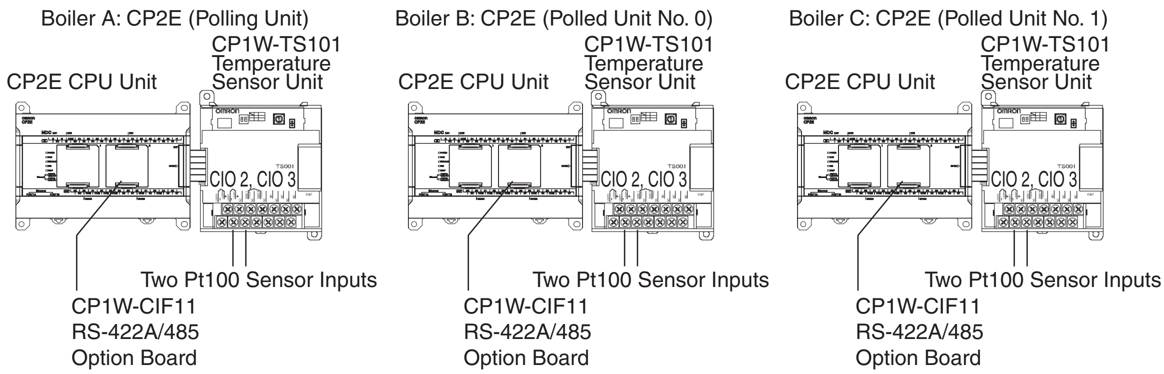
14-5-5 Example Application

Operation

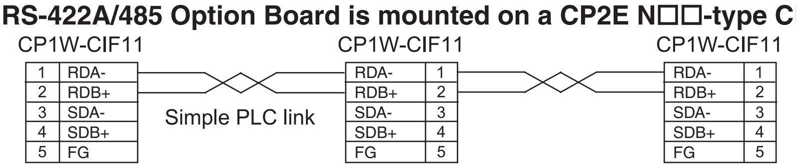
The present temperature information is exchanged between the boilers. This information is used to adjust the temperature control of one boiler depending on the status of the other boilers and for monitoring individual boilers.



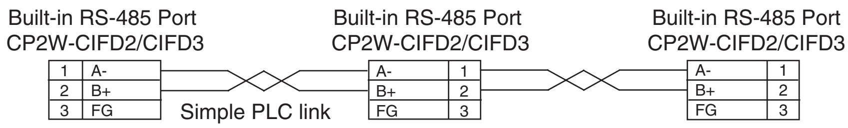
- **Wiring Example**



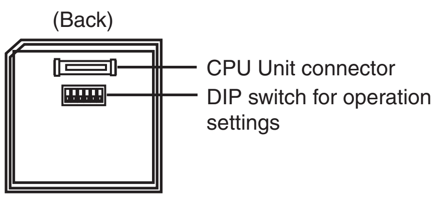
When a RS-422A/485 Option Board is mounted on a CP2E N☐☐-type CPU Unit



Built-in RS-485 Port of CP2E S☐☐-type CPU Unit or When an Option Board with 2 ports is mounted on a CP2E N☐☐-type CPU Unit



- **CP1W-CIF11 RS422/485 Option Board DIP Switch Settings**



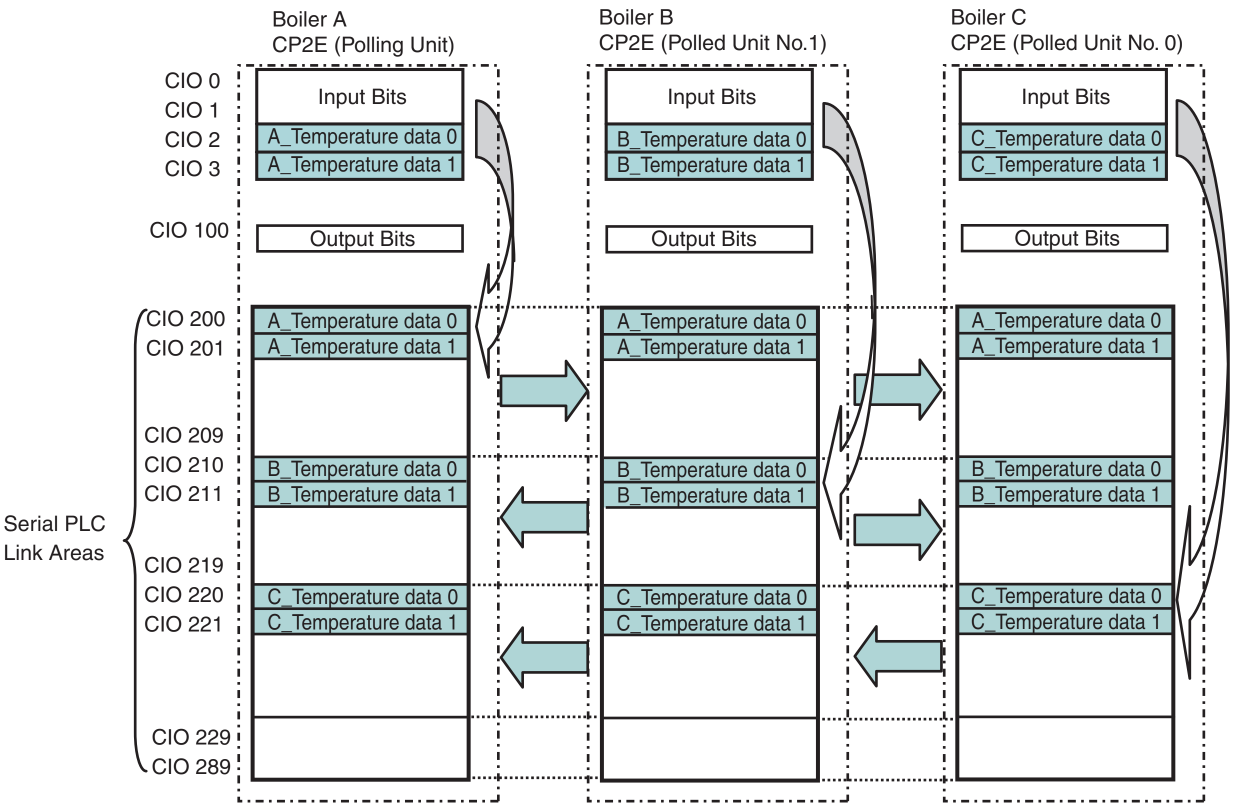
No.	Settings	Polling Unit	Polled Unit No. 0	Polled Unit No. 1	Description
1	Terminating resistance selection	ON	OFF	ON	PLCs at both ends must have terminating resistance connected.
2	2-wire or 4-wire selection	ON	ON	ON	2-wire
3	2-wire or 4-wire selection	ON	ON	ON	2-wire
4	–	OFF	OFF	OFF	Always OFF
5	RS control selection for RD	OFF	OFF	OFF	Control disabled
6	RS control selection for SD	ON	ON	ON	Control enabled

● PLC Setup

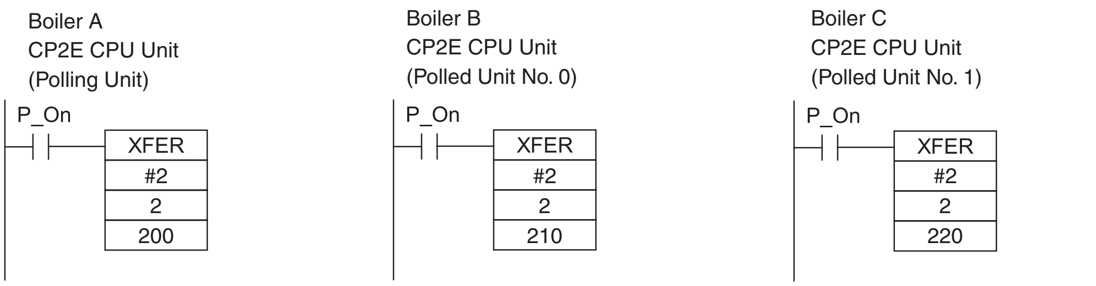
Item	Boiler A (Polling Unit)	Boiler B (Polled Unit No. 0)	Boiler C (Polled Unit No. 1)
Communications Settings	Custom		
Baud Rate	115200bps		
Parameters	7.2.E (default)		
Mode	PC Link (Master)	PC link (Slave)	
Link words	10 (default)	–	–
PC Link Mode	ALL	–	–
NT/PC Link Max.	1	–	–
PC Link Unit No.	–	0	1

● Programming Example

Data in the Serial PLC Link Areas are transferred using data links by the Serial PLC Link and without using any special programming. The ladder program is used to transfer the data that needs to be linked to the data link area.



● Ladder Diagram



Transfer CIO 2 and CIO 3 to CIO 200 and CIO 201 using a BLOCK TRANSFER instruction.

Transfer CIO 2 and CIO 3 to CIO 210 and CIO 211 using a BLOCK TRANSFER instruction.

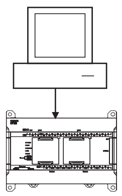
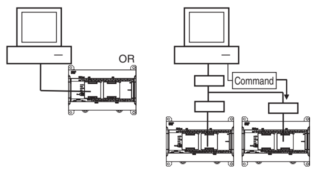
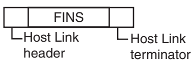
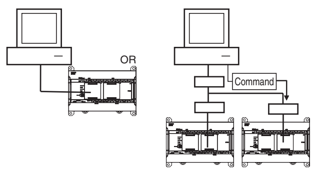
Transfer CIO 2 and CIO 3 to CIO 220 and CIO 221 using a BLOCK TRANSFER instruction

14-6 Connecting the Host Computer

14-6-1 Overview

Commands are sent from a host computer to the CP2E CPU Unit to read and write data. The serial communications mode is set to Host Link.

Note Because the built-in RS-485 port of the S☐☐-type CPU Unit and the RS-485 port of CP2W-CIFD2/CIFD3 use 2-wire connections, so they can only communicate in half duplex. Communications are not possible in full duplex.

Command flow	Command type	Communica-tions method	Configuration	Application	Remarks
Host computer → PLC 	Host link command (C Mode) Host link command	Create frame in the host computer and send the command to the PLC. Receive the response.	Directly connect the host computer in a 1:1 or 1:N system. 	Use this method when communicating primarily from the host computer to the PLC.	–
	FINS command (with Host Link header and terminator) sent. 		Directly connect the host computer in a 1:1 system or 1:N system. 	Use these methods when communicating primarily from the host computer to PLCs in the network.	The FINS command must be placed between a Host Link header and terminator and then sent by the host computer.

14-6-2 Flow of Operation

1 Communications wiring

Connect the computer and CP2E CPU Unit using RS-232C ports or RS-485 ports.

2 PLC Setup

Set the PLC Setup (select Host Link for the serial communications mode and set the communications conditions) and transfer the PLC Setup from the CX-Programmer to the CP2E CPU Unit.

3 Program from host

Send the following commands from the host computer.

- C-mode commands
- FINS commands

14-6-3 Command/response Format and List of Commands

The outline of command/response format and each command are listed below.

For the details of the host link commands and FINS commands, refer to Communication *Instructions Reference Manual* (Cat.No.W342).

● List of C Mode Commands

C mode commands (host link commands) are shown below.

Type	Header code	Name	Function
I/O memory read	RR	CIO area read	Read the specified words from the specified words of CIO area
	RH	Holding area read	Read the specified words from the specified words of Holding area (H)
	RC	Timer and Counter PV area read	Read the specified present values of timer and counter from the specified words
	RG	Timer and Counter Completion Flag read	Read the specified Completion Flag of timer and counter from the specified words
	RD	DM area read	Read the specified words from the specified words of DM area (D)
	RJ	Auxiliary area read	Read the specified words from the specified words of Auxiliary area (A)
I/O memory write	WR	CIO area write	Write the specified source words from the specified words of CIO area in the unit of word
	WH	Holding area write	Write the specified source words from the specified words of Holding area (H) in the unit of word
I/O memory write	WC	Timer and Counter PV area write	Write the specified source words from the specified words of Timer and Counter present value area in the unit of word
	WD	DM area write	Write the specified source words from the specified words of DM area (D) in the unit of word
	WJ	Auxiliary area write	Write the specified source words from the specified words of Auxiliary area (A) in the unit of word
CPU Unit status related	MS	CPU Unit status read	Read the CPU Unit operating conditions (operating mode, forced set/reset, fatal error)
	SC	Status change	Change the operating mode of CPU Unit
	MF	Error information read	Read the occurring error information of CPU Unit (fatal error, non-fatal error)
Test	TS	Test	Directly return 1 block sent from the host computer
I/O memory area mixed read	QQMR	I/O memory area mixed read registration	Register the I/O memory words or bits that need to read into the table
	QQIR	I/O memory area mixed registration	Read all of the I/O memory area words and bits that were registered
Host link communication processing	XZ	Abort (command only)	Interrupt the operation that being processed using the host link command, and return to the initial status after abortion
	**	Initial (command only)	Initialize the transmission control sequence for all the host link unit numbers
	IC	Command undefined error (response only)	Response when the command's header code cannot be broken

List of FINS commands

FINS commands are shown below.

Type	Command code		Name	Function
I/O memory area access	01	01	I/O memory area read	Read the contents of continuous I/O memory area
	01	02	I/O memory area write	Write the contents of continuous I/O memory area
	01	03	I/O memory area write all at once	Replenish the specified ranges of I/O memory area with the same data
	01	04	I/O memory area mixed read	Read the contents of discontinuous I/O memory area
Parameter area access	02	01	Parameter area read	Read the contents of continuous parameter area
	02	02	Parameter area write	Write the contents of continuous parameter area (unable to execute in MONITOR or RUN mode)
	02	03	Parameter area write (clear) all at once	Replenish the specified ranges of parameter area with the same data
Operating mode change	04	01	Operating mode change (Operation start)	Change the operating mode of CPU Unit to RUN or MONITOR mode
	04	02	Operating mode change (Operation stop)	Change the operating mode of CPU Unit to PROGRAM mode
System configuration read	05	01	CPU Unit information read	Read CPU Unit information
Status read	06	01	CPU Unit status read	Read the status information of CPU Unit
	06	20	Cycle time read	Read cycle time (MAX, MIN, AVERAGE)
Time information access	07	01	Time information read	Read present year, month, day of the month, hour, minute, second, day of the week
	07	02	Time information write	Change present year, month, day of the month, hour, minute, second, day of the week
Message display related	09	20	Message read/cancel	Read FAL and FALS
Debugging related	21	03	Error log pointer clearance	Clear all the pointer of error log to zero
	23	01	Force-set/reset	Force-set, force-reset and release (unable to specify multi-bit)
	23	02	All bits release	Release the forced status of all bits

14-6-4 Restrictions on the Usage of Host Link

The CP2E serial port does not support the following functions:

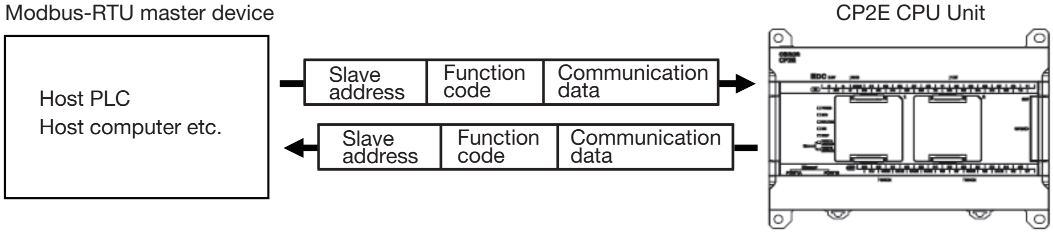
- The issue of FINS command by the SEND, RECV or CMND instruction from the serial port is not supported.
- Serial gateway function is not supported.

14-7 Modbus-RTU Slave Function

14-7-1 Overview

Modbus-RTU Slave enables to read and write CP2E data from Modbus-compatible masters, such as host PLCs or host computers, using serial communications.

The serial communication mode is set to Modbus-RTU Slave.



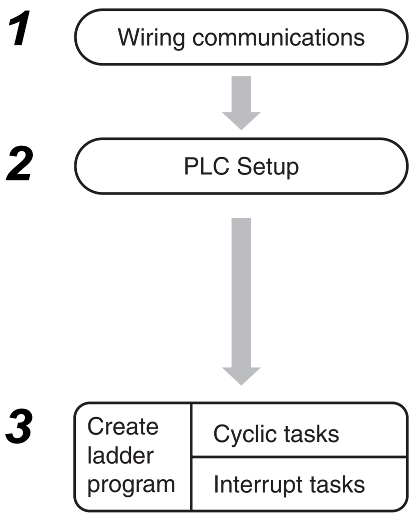
By setting the Modbus-RTU slave address on the CP2E side, the target slave address, function code and data from the master devices can be transmited.

Specifications

Item	Contents
Mode	Modbus-RTU Slave
Baud rate	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Data length	8 bits *1
Parity	Odd, Even, None
Stop bits	1 bit (2 bits only when setting to no parity) *2
Slave addresse	1 to 247 *3
Frame format	Slave address: 1 byte Function code: 1 byte Data: 0 to 252 bytes CRC code: 2 bytes

*1 The data length is always 8 bits.
*2 The number of stop bits is determined by the parity setting.
When setting with parity (even/odd), 1 bit
When setting without parity, 2 bits
*3 The address cannot be set to 0. In commands, 0 is used to indicate broadcast commands.

14-7-2 Flow of Operation

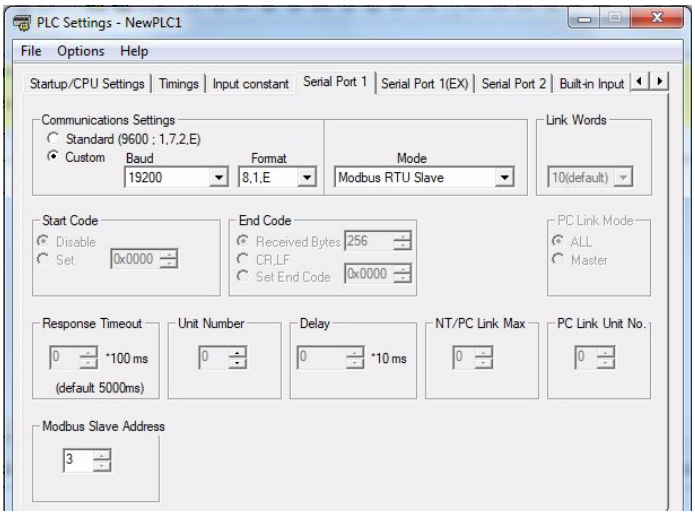


1. Connect CP2E CPU Unit and Modbus-RTU master device using RS-232C or RS-422A/485 ports.

2. Select *Built-in RS232C, Built-in RS485* or *Serial Port* in the PLC Setup and transfer the PLC Setup from the CX-Programmer to the CP2E CPU Unit.
   - Set the seial communication mode to Modbus-RTU Slave
   - Set the baud rate and format
   - Set Modbus Slave address

3. When receiving a Modbus-RTU command from a Modbus-RTU Master, it automatically responds without a ladder program.

14-7-3 PLC Setup



Built-in RS-232C, Built-in RS-485 or Serial Port Tab Page

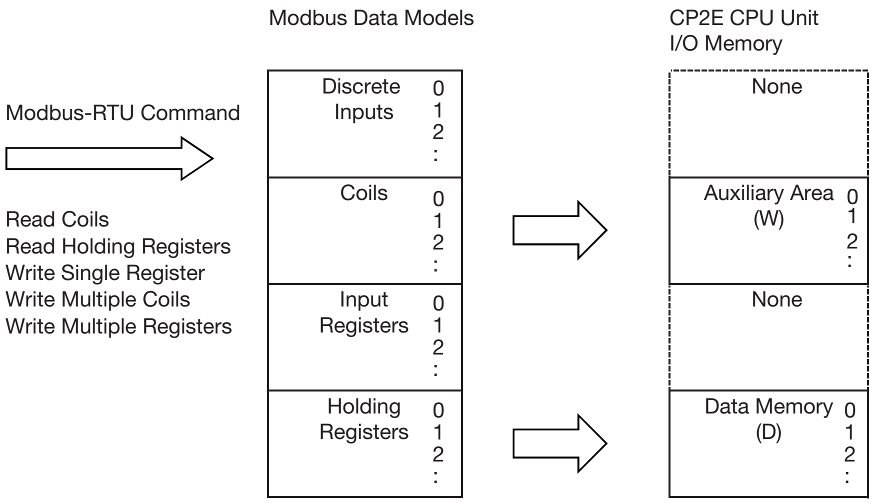
Parameter	Settings
Communications Settings	Set the communications settings to the same values as the connected Modbus-RTU master device. If the connected device is set to 19,200 bps, one stop bit, and even parity, select the *Custom Option*. Set the baud rate to 19,200 and the format to *8,1,E*.
Mode	Select *Modbus RTU Slave*.
Modbus Slave Address	Set the CP2E slave address.

14-7-4 Operation Specifications

Modbus has the following four common data models.

CP2E allocates each area of these data models to an I/O Memory area.

Modbus data model	Data type	Read/Write	CP2E CPU unit I/O Memory allcation
Discrete Inputs	Bit	read	None
Coils	Bit	read	Auxiliary Area (W)
Input Registers	Word (16 bit)	read/write	None
Holding Registers	Word (16 bit)	read/write	Data Memory (D)



● CP2E Fixed Allocations

The following table gives the relationship between Modbus data model and CP2E I/O memory of CP2E CPU unit.

Modbus data model	Modbus address	Address specified in Modbus-RTU commands	Corresponding CP2E I/O Memory address
Discrete Inputs	---	---	---
Coils	1 to 2048	0 to 2047	W0.00 to W127.15
Input Registers	---	---	---
Holding Registers	1 to 4096	0 to 4095	CP2E E□□-type D0 to D4095
	1 to 8192	0 to 8191	CP2E S□□-type D0 to D8191
	1 to 16384	0 to 16383	CP2E N□□-type D0 to D16383

Additional Information

Addresses in Modbus data models start from 1, but addresses specified in Modbus-RTU commands and addresses in the CP2E CPU Unit start from 0. Refer to the above table when specifying addresses in applications.

14-7-5 Command and Response Details

Supported Command List

The CP2E CPU unit supports the following Modbus-RTU commands.

Function Code	Modbus Name	Function
01 Hex	Read Coils	Reads multiple bits from the Auxiliary Area (W) of I/O Memory
03 Hex	Read Holding Registers	Reads multiple words from the Data Memory (D)
06 Hex	Write Single Register	Write a channel to Data Memory (D)
0F Hex	Write Multiple Coils	Writes multiple bits to the Auxiliary Area (W)
10 Hex	Write Multiple Registers	Write multiple channel to Data Memory (D)

The frame format used in Modbus-RTU slave mode is as follows:

Slave address	Function code	Data	CRC
1 byte	1 byte	0 to 252 bytes	2 bytes (See note.)

Note The CRC code is given in the order low byte, high byte.

Additional Information

Setting the slave address to 0 indicates a broadcast command. The CP2E CPU unit does not return a response for broadcast commands.

Precautions for Correct Use

Program retry processing in the application of Modbus-RTU master (PLC, Host computer, etc.) when a transmission error or other communications error occurs.

Command and Response Details

● Read Multiple Bits from the Auxiliary Area (W) (Read Coils)

Command (Modbus-RTU Master)

Field name	Data length	Data
Function code	1 byte	01 Hex
Coil starting address	2 bytes	0 to 7FFHex (0 to 2047) W0.00 to W127.15
Quantity of coils	2 bytes	1 to 7FFHex (1 to 2047)

Note The maximum number of coils depends on the assigned starting address.

Response (CP2E)

Field name	Data length	Data
Function code	1 byte	01 Hex
Byte count	1 byte	N
Coil status	n bytes	n=N or N+1

Example: Reading 19 bits from W1.04 to W2.06

Command (Modbus-RTU Master)

Field name	Data
Function code	01 Hex
Coil starting address (H)	00 Hex
Coil starting address (L)	14 Hex (20 bits W1.04~)
Quantity of coils (H)	00 Hex
Quantity of coils (L)	13 Hex (19 bits) (W1.04 to W2.06)

Response (CP2E)

Field name	Data
Function code	01 Hex
Byte count	03 Hex
Coil status 27 to 20	CD Hex (W1.11 to W1.04)
Coil status 35 to 28	B6 Hex (W2.03 to W1.12)
Coil status 38 to 36	05 Hex* (W3.06 to W3.04)

* The remaining bits less than one byte will be read as 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0CH	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1CH	31_1	30_0	29_1	28_1	27_1	26_1	25_0	24_0	23_1	22_0	21_1	20_0	19	18	17	16
2CH	47	46	45	44	43	42	41	40	39	38_1	37_0	36_1	35_0	34_1	33_1	32_0
3CH	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

Note The subscript numbers in the shaded boxes indicate the ON/OFF (1/0) status of the bits that are read.

● Read Multiple Words from the Data Memory (D) (Read Holding Registers)

Command (Modbus-RTU Master)

Field name	Data length	Data
Function code	1 byte	03 Hex
Register starting address	2 bytes	E□□-type: 0 to 0FFF Hex (D0 to D4095) S□□-type: 0 to 1FFF Hex (D0 to D8191) N□□-type: 0 to 3FFF Hex (D0 to D16383)
Quantity of Registers	2 bytes	1 to 7D Hex (1 to 125)

Note The maximum number of coils depends on the assigned starting address.

Response (CP2E)

Field name	Data length	Data
Function code	1 byte	03 Hex
Byte count	1 byte	2×N (N: Quantity of registers)
Register value	2×N bytes	

Example: Reading 3 words from D1000 to D1002

Command (Modbus-RTU Master)

Field name	Data
Function code	03 Hex
Register starting address (H)	03 Hex
Register starting address (L)	E8 Hex (D1000~)
Quantity of Registers (H)	00 Hex
Quantity of Registers (L)	03Hex (3CH) (D1000 to D1002)

Response (CP2E)

Field name	Data
Function code	03 Hex
Byte count	06 Hex
Register value (H)	AB Hex (D1000 H)
Register value (L)	12 Hex (D1000 L)
Register value (H)	56 Hex (D1001 H)
Register value (L)	78 Hex (D1001 L)
Register value (H)	97 Hex (D1002 H)
Register value (L)	13 Hex (D1002 L)

DM	15 to 12	11 to 8	7 to 4	3 to 0
1000	A	B	1	2
1001	5	6	7	8
1002	9	7	1	3

- **Write a word in the Data Memory (D) (Write Single Register)**

Command (Modbus-RTU Master)

Field name	Data length	Data
Function code	1 byte	06 Hex
Register address	2 bytes	E□□-type: 0 to 0FFF Hex (D0 to D4095) S□□-type: 0 to 1FFF Hex (D0 to D8191) N□□-type: 0 to 3FFF Hex (D0 to D16383)
Register value	2 bytes	0000 to FFFF Hex

Response (CP2E)

Field name	Data length	Data
Function code	1 byte	06 Hex
Register address	2 bytes	E□□-type: 0 to 0FFF Hex (D0 to D4095) S□□-type: 0 to 1FFF Hex (D0 to D8191) N□□-type: 0 to 3FFF Hex (D0 to D16383)
Register value	2 bytes	0000 to FFFF Hex

Example: Writing 3AC5 Hex to D2000

Command (Modbus-RTU Master)

Field name	Data
Function code	06 Hex
Register address (H)	07 Hex
Register address (L)	D0 Hex (D2000)
Register value (H)	3A Hex
Register value (L)	C5 Hex

Response (CP2E)

Field name	Data
Function code	06 Hex
Register address (H)	07 Hex
Register address (L)	D0 Hex
Register value (H)	3A Hex (D1000 H)
Register value (L)	C5 Hex (D1000 L)

DM	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
2000	3	A	C	5
2001				
2002				

● Write Multiple Bits in the Auxiliary Area (W) (Write Multiple Coils)

Command (Modbus-RTU Master)

Field name	Data length	Data
Function code	1 byte	0F Hex
Starting address	2 bytes	0 to 07FF Hex (0 to 2047) W0.00 to W127.15
Quantity of output	2 bytes	0001 to 07B0 Hex (1 to 1968)
Byte count	1 byte	N = Quantity of output / 8, if the remainder is not 0, N+1
Output value	N bytes	

Response (CP2E)

Field name	Data length	Data
Function code	1 byte	0F Hex
Starting address	2 bytes	0 to 07FF Hex (0 to 2047) W0.00 to W127.15
Quantity of output	2 bytes	0001 to 07B0 Hex (1 to 1968)

Example: Writing 10 bits (xxxx xx11 1100 1101) from W1.04 to W1.13

Command (Modbus-RTU Master)

Field name	Data
Function code	06 Hex
Starting address (H)	00 Hex
Starting address (L)	14 Hex (W1.04~)
Quantity of output (H)	00 Hex
Quantity of output (L)	0A Hex (10 bits)
Byte count	02 Hex
Output value(H)	3A Hex
Output value(L)	01 Hex

Response (CP2E)

Field name	Data
Function code	0F Hex
Starting address (H)	00 Hex
Starting address (L)	14 Hex
Quantity of output (H)	00 Hex
Quantity of output (L)	0A Hex

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0CH	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1CH	31_0	30_0	29_0	28_1	27_0	26_0	25_1	24_1	23_1	22_0	21_1	20_0	19_0	18_0	17_0	16_0

Note The subscript numbers in the shaded boxes indicate the ON/OFF (1/0) status of the bits that are written. Other bits in the same word are set to 0.

● Write Multiple Words in the Data Memory (D) (Write Multiple Registers)

Command (Modbus-RTU Master)

Field name	Data length	Data
Function code	1 byte	10 Hex
Starting address	2 bytes	E□□-type: 0 to 0FFF Hex (D0 to D4095) S□□-type: 0 to 1FFF Hex (D0 to D8191) N□□-type: 0 to 3FFF Hex (D0 to D16383)
Quantity of registers	2 bytes	0001 to 07B Hex (1 to 123)
Byte count	1 byte	2×N (N: Quantity of registers)
Registers value	2×N bytes	

Response (CP2E)

Field name	Data length	Data
Function code	1 byte	10 Hex
Starting address	2 bytes	0 to 7FF Hex (0 to 2047) W0.00 to W127.15
Quantity of registers	2 bytes	0001~07BHex (1~123)

Example: Writing 3AC5, 9713 Hex to 2 Words, D1000 and D1001

Command (Modbus-RTU Master)

Field name	Data
Function code	10 Hex
Starting address (H)	03 Hex
Starting address (L)	E8 Hex (D1000~)
Quantity of registers (H)	00 HEX
Quantity of registers (L)	02 Hex (2CH)
Byte count	04 Hex
Registers value (H)	3A Hex
Registers value (L)	C5 Hex
Registers value (H)	97 Hex
Registers value (L)	13 Hex

Response (CP2E)

Field name	Data
Function code	10 Hex
Starting address (H)	03 Hex
Starting address (L)	E8 Hex
Quantity of registers (H)	00 Hex
Quantity of registers (L)	02 Hex

DM	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
1000	3	A	C	5
1001	9	7	1	3

● Error Response

Field name	Data length	Data
Function code	1 byte	Function code + 80 Hex For example: 86 Hex for Write Single Register (06Hex)
Exception code	1 byte	Reference below

Error code	Field name	Description
01 Hex	Illegal function code	• An unsupported function code is specified
02 Hex	Illegal data address	• There is an error in the specified start address • The specified start address and Data length exceed the valid range
03 Hex	Illegal data value	• Data number does not match data length

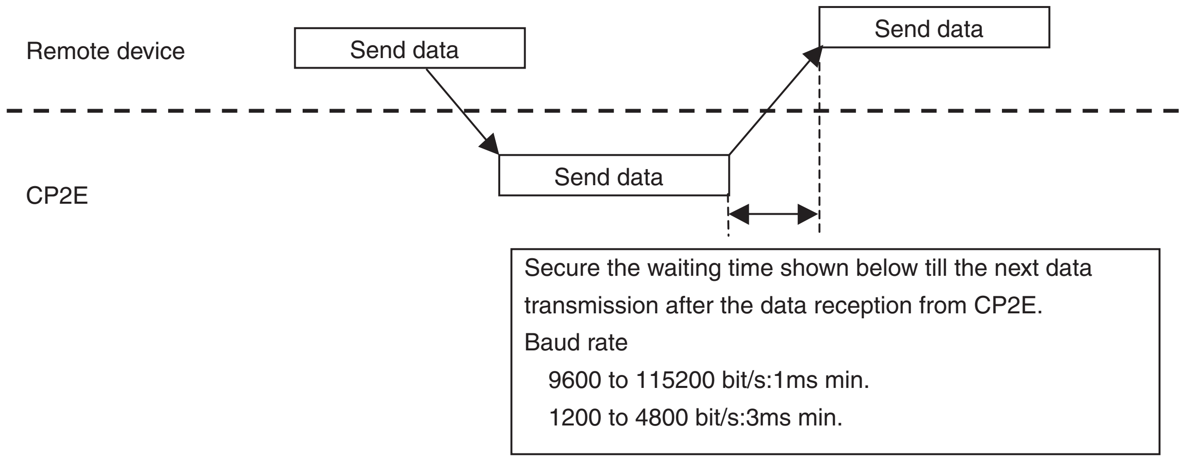
14-7-6 Related special auxiliary relay

Address	Name	Content
A392.04	Built-in RS-232C Port/Serial Port 1 Communication Error Flag	Turns ON when a communication error (framing error, parity error, overrun error, CRC error) occurs in the built-in RS-232C port or serial port 1. If this flag occurs, it is necessary to restart the port. However, in the case of a CRC error, port restart is not necessary because it turns off automatically when the next normal Modbus-RTU command is received.
A526.00	Built-in RS-232C Port/Serial Port 1 Port Restart Flag	When performing port restart on the built-in RS-232C port or serial port 1, perform 0 → 1. After restart processing, it will be 0 (OFF) automatically.
A528.00 to A528.07	Built-in RS-232C Port/Serial Port 1 Error Flag	When an error occurs in built-in RS-232C port or serial port 1, the error code is stored. Bit 2: Parity error Bit 3: Framing error Bit 4: Overrun error
A392.12	Built-in RS-485 Port/Serial Port 2 Communication Error Flag	Turns ON when a communication error (framing error, parity error, overrun error, CRC error) occurs in the built-in RS-485 port or serial port 2. If this flag occurs, it is necessary to restart the port. However, in the case of a CRC error, port restart is not necessary because it turns off automatically when the next normal Modbus-RTU command is received.
A526.01	Built-in RS-485 Port/Serial Port 2 Restart Flag	To restart the port for built-in RS-485 port or serial port 2, perform 0 → 1. After restart processing, it will be 0 (OFF) automatically.
A528.08 to A528.15	Built-in RS-485 Port/Serial Port 2 Error Flag	When an error occurs in built-in RS-485 port or serial port 2, the error code is stored. Bit 2: Parity error Bit 3: Framing error Bit 4: Overrun error
A50.04	Serial Port 1 (EX) Communication Error Flag	• Turns ON when a communication error (framing error, parity error, overrun error, CRC error) occurs in serial port 1 (EX). If this flag occurs, it is necessary to restart the port. However, in the case of a CRC error, port restart is not necessary because it turns off automatically when the next normal Modbus-RTU command is received.
A526.02	Serial Port 1 (EX) Port Restart Flag	• When performing port restart on serial port 1 (EX), perform 0 → 1. After restart processing, it will be 0 (OFF) automatically.
A521.00 to A521.07	Serial Port 1 (EX) Error Flag	• When an error occurs in serial port 1 (EX), the error code is stored. Bit 2: Parity error Bit 3: Framing error Bit 4: Overrun error

14-8 Precautions on the usage of RS-485

When using 2-wire RS-485 with the built-in RS-485 of the S□□-type CPU Unit, or the RS422A/485 Option Board CP1W-CIF11/CIF12-V1 and CP2W-CIFD2/CIFD3 Option Board with two ports, pay attention to the following precautions and construct application.

When using the RS-485 (2-wire), it can only communicate in half duplex.
Please secure the waiting time shown below till the next data transmission after the remote device receives data from CP2E. If the receive data are transmitted from the remote device within the waiting time shown below after CP2E data transmission, the data may not be received by CP2E.



15

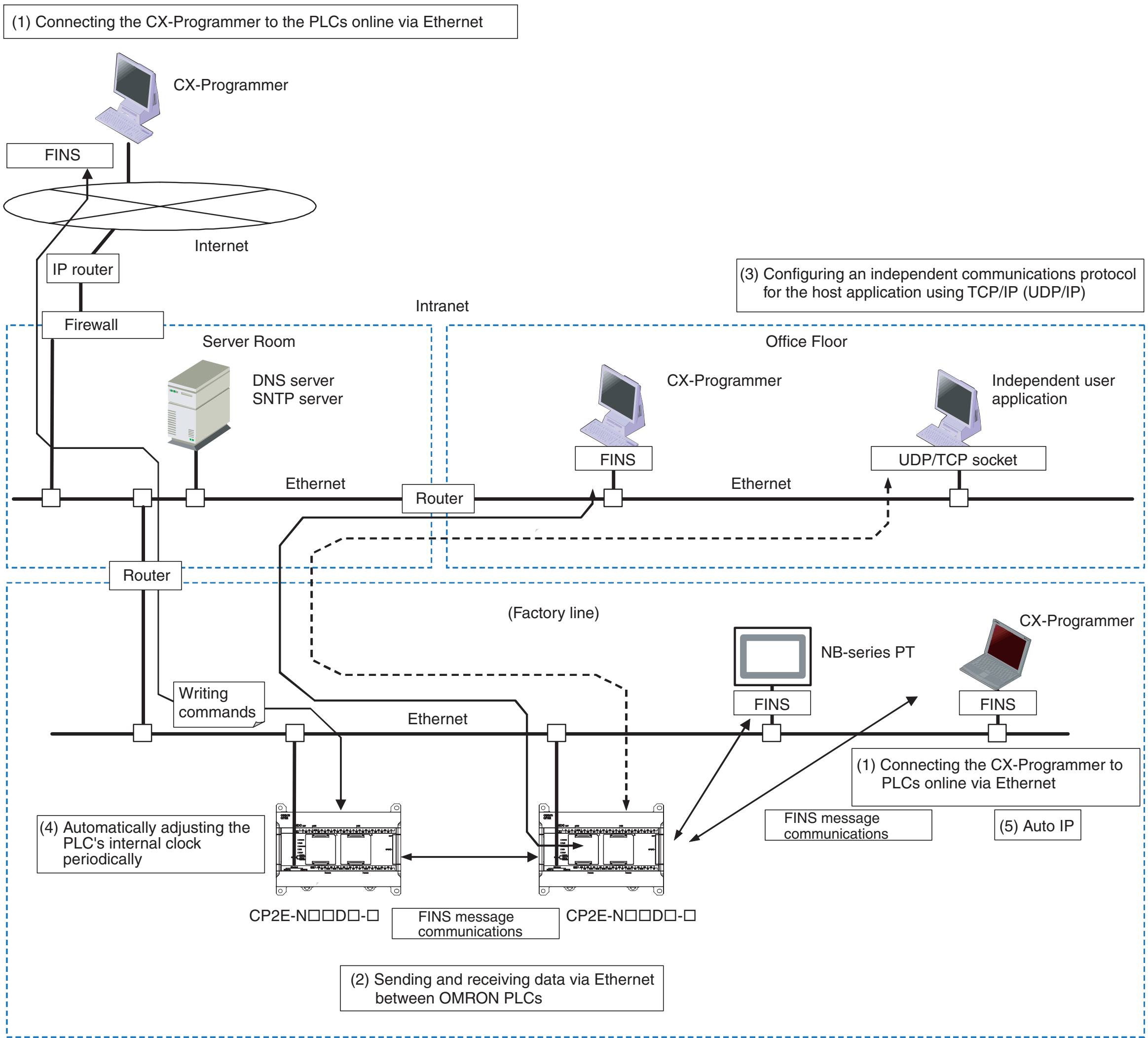
Ethernet

This section gives an outline of the built-in Ethernet function, explains its specification and how to make the settings required for operation.

15-1 System Configuration and Features

N14/20 CPU Units have one built-in Ethernet port. N30/40/60 CPU Units have two Ethernet ports, and the second port is only for switch. A variety of protocols make available a wide range of applications for use on an Ethernet network. The protocols can be selected include sending and receiving data by TCP/IP or UDP/IP (socket services), sending and receiving commands by OMRON's standard protocol FINS, and automatically adjusting the PLC's internal clock by SNTP.

The following diagram shows an example of an overall system configuration using CP2E N□□-type CPU Units.



15-1-1 Connecting the CX-Programmer to PLCs Online via Ethernet

Auto IP Operation

Auto IP protocol is supported by CP2E N☐☐-type CPU Units.

With Auto-IP, CX-Programmer's online connection command can be executed without any IP setting in computer side when PLC is physically connected with the computer directly.

CX-Programmer can list all the CP2E N☐☐-type CPU Units connected in a same segment with the computer. And the information of PLC (such as IP address, MAC address) will be displayed.

Connecting within the Same Segment

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 OMRON Ethernet Units. The CX-Programmer can be connected and used with FINS/UDP.

Connecting through Multiple Segments

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 multi-level routing.
For CX-Programmer, FINS/TCP can be used to directly connect to the PLC online.

Connecting from a Personal Computer with a Dynamic Private IP Address

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 FINS/UDP service or use the FINS/TCP 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.

15-1-2 Exchanging Data between OMRON PLCs using Ethernet

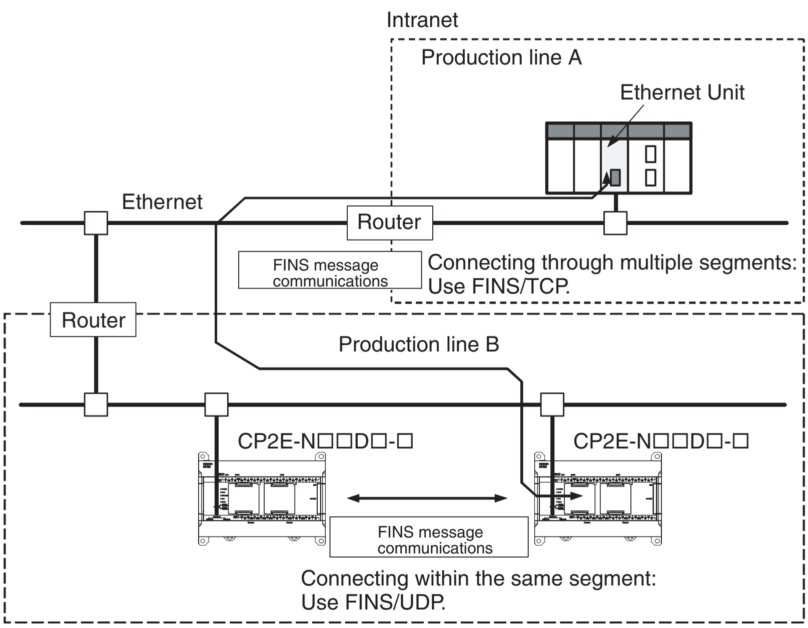
Connecting within the Same Segment

Use the FINS/UDP, and construct applications using the SEND(090), RECV(098), and CMND(490) instructions in the ladder program. 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.

Connecting through Multiple Segments

Use the FINS/TCP, and construct applications using the SEND(090), RECV(098), and CMND(490) instructions in the ladder program. FINS/TCP is supported by many OMRON products and is compatible with earlier OMRON Ethernet Units. It provides automatic recovery at the TCP/IP layer from communications errors (such as packet loss) that occur during multilevel routing.



15-1-3 Creating an Original Communications Procedure Using TCP/IP (UDP/IP) for the Host Application or Communicating with PLCs from Another Manufacturer

Communications by UDP/IP and TCP/IP (Socket Services Function)

The standard Ethernet protocols, UDP/IP and TCP/IP, are supported, making it possible to communicate with a wide range of devices, workstations, computers, and Ethernet Units from other manufacturers.

Up to three ports can be used for various protocols, enabling the use of various applications.

Simplified Socket Services

The socket services function for TCP or UDP can be simplified by presetting parameters and using dedicated bits. In addition, the size of received data accumulated in the reception buffer is now stored, and a Data Received Flag has been added. These features eliminate the need for ladder programs to monitor the timing for completion of socket service processing, and thus reduce the amount of labor required for program development.

15-1-4 Automatically Adjusting the PLC's Internal Clock at Regular Intervals

Operation

Use the automatic clock adjustment function.

With the automatic clock adjustment function, the SNTP server's clock is taken as the standard for automatically adjusting the built-in clock of the PLC. The adjustment can be regularly executed at a specified time (once a day) and it can be executed each time by the ladder program.

To use the automatic clock adjustment function, there must be a separate SNTP server on the network. Aside from the IP address, the host name used for DNS service can be specified for the SNTP server.

Specification of Servers by Host Name

In addition to directly specifying the IP address for a SNTP server, it is also possible (by means of the PLC's DNS client function) to specify the server by host name. This enables automatic searches for IP addresses for purposes such as system checking, even when the IP addresses for servers have been changed.

Note A separate DNS server is required to specify servers by host name using DNS.

15-2 Specifications

15-2-1 General Specifications (Ethernet)

Item		Specifications	
Type		100/10Base-TX (Auto-MDIX)	
Transfer	Media access method	CSMA/CD	
	Modulation method	Baseband	
	Transmission paths	Star form	
	Baud rate	100 Mbit/s (100Base-TX)	10 Mbit/s (10Base-T)
		• Half/full auto-negotiation for each port • Link speed auto-sensing for each port	
	Transmission media	• Unshielded twisted-pair (UDP) cable Categories: 5, 5e • Shielded twisted-pair (STP) cable Categories: 100Ω at 5, 5e	• Unshielded twisted-pair (UDP) cable Categories: 3, 4, 5, 5e • Shielded twisted-pair (STP) cable Categories: 100Ω at 3, 4, 5, 5e
	Transmission distance	100 m (distance between hub and node)	
	Cascade connection	There is no limitation on the usage of switching hub	
Protocols		TCP, UDP, ARP, ICMP (ping only), SNTP, DNS	
Applications		FINS, Socket, SNTP, DNS (Client)	

● Switching Hub for CP2E N30/40/60 CPU Units

Ethernet	100Base-TX, 10Base-T
Auto MID/MID-X	Yes
Auto negotiation	Yes
Store-and-forward system	Yes
Buffer	32K bytes
MAC address	1000
Broadcast storm detection	Yes
QoS	No
SNMP	No
VLAN	No
IGMP snooping	No
STP (Spanning Tree Protocol)	No
Port mirroring	No

Additional Information

The switching hub function works normally even if a fatal/non-fatal error or CPU WDT error occurs. The switching hub function does not work when the power of the CPU Unit is turned OFF.

15-2-2 Comparison with Previous Models (Ethernet Related)

Model		CP2E N□□-type CPU Units	CP1W-CIF41	CS1W-ETN21 CJ1W-ETN21
Local IP address		192.168.250.FINS node address	192.168.250.1	192.168.250.FINS node address
FINS node address		Set in PLC setup	Set in system settings	Set by rotary switch
Physical layer		100/10Base-TX (Auto-MDIX)	100/10Base-TX (Auto-MDIX)	100/10Base-TX
Number of nodes		254	254	254
Data length of FINS message		1004 bytes (Max)	1004 bytes (Max)	2012 bytes (Max)
FINS buffer size		16K bytes	8K bytes	392K bytes
Driver buffer number		Input: 8×608 bytes Output: 4×1600 bytes	Input: 16×256 bytes Output: 8×256 bytes	Input: 50×1.5K bytes Output: 50×1.5K bytes
Process of driver buffer overflow		The last packet will be dropped.	The last packet will be dropped.	The last packet will be dropped.
Connection number (FINS/TCP)		3 for user 1 for CX-Programmer auto connection	2 (only server)	16
PLC maintenance via the Internet		Not supported	Not supported	Not supported
Server specification		Specification by IP address or by host name (DNS Client Function)	Not supported	Specification by IP address or by host name (DNS Client Function)
FINS comm. service	Automatic IP address acquisition	A computer automatically acquiring IP addresses can send commands to the PLC and receive responses.	A computer automatically acquiring IP addresses can send commands to the PLC and receive responses.	A computer automatically acquiring IP addresses can send commands to the PLC and receive responses.
	FINS communication with computer without fixed node address	Possible (with automatic allocation) (Client FINS automatic node address allocation function, TCP/IP only)	Possible (with automatic allocation) (Client FINS automatic node address allocation function, TCP/IP only)	Possible (with automatic allocation) (Client FINS automatic node address allocation function, TCP/IP only)
	Handling TCP/IP	With FINS communications, both UDP/IP and TCP/IP (3 max.) possible.	With FINS communications, both UDP/IP and TCP/IP (2 max.) possible. (Only can be set to server)	With FINS communications, both UDP/IP and TCP/IP (16 max.) possible.
	Simultaneous connection of multiple applications in a computer	Possible (with both UDP/IP and TCP/IP)	Possible (with both UDP/IP and TCP/IP)	Possible (with both UDP/IP and TCP/IP)
Mail function		Not supported	Not supported	E-mail attachments with I/O memory data are possible for the mail send function. (SMTP, file attachment) With the mail receive function, commands can be received from the PLC. (POP3, mail receive)
FTP server function		Not supported	Not supported	Supported
Socket services function		Supported	Not supported	Supported
Automatic clock information adjustment		Supported	Not supported	Supported
IP conflict (GARP)		Supported	Not supported	Supported
TCP keep-alive function		Supported	Not supported	Supported
Multicast function		Not supported	Not supported	Not supported
Web function		Not supported	Supported	Supported
Switch function		Supported	Not supported	Not Supported

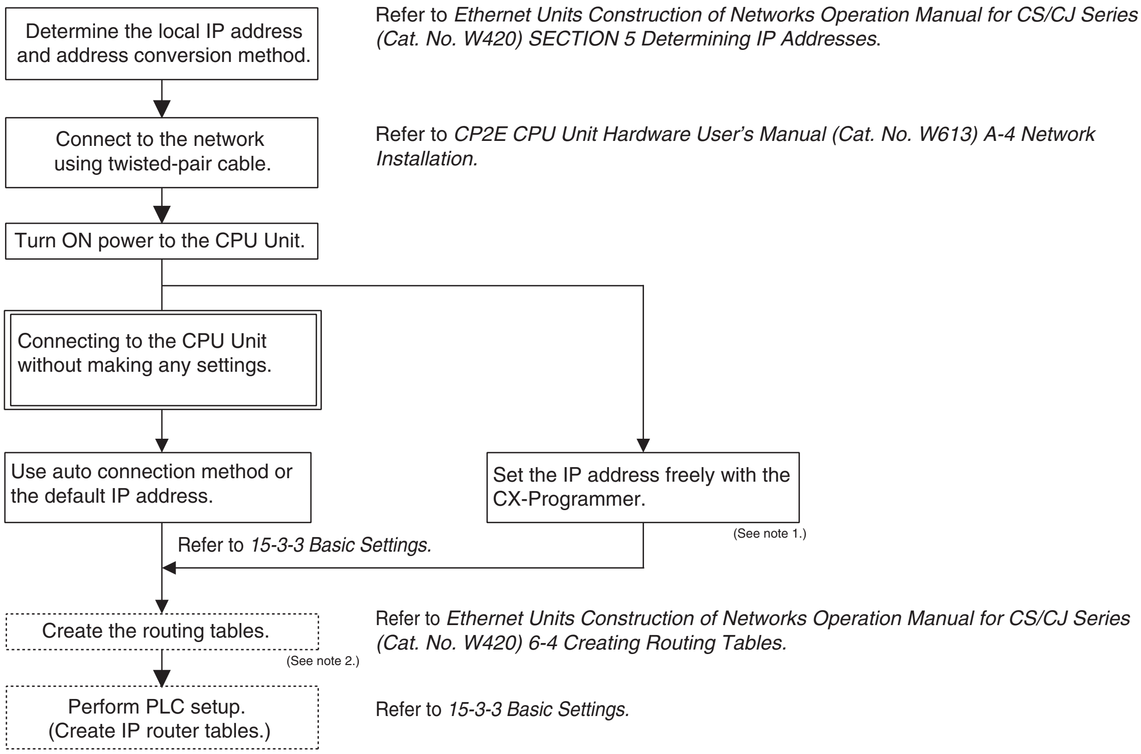
Improved FINS Message Communications from CP1W-CIF41

The following functions have been maintained according to the existing Ethernet Unit models for CP1W-CIF41.

- 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, and it is enabled in TCP/IP with up to 3 simultaneous connections.
  →Previously CP1W-CIF41 is enabled in TCP/IP with up to 2 simultaneous connections and all can only be set to server.
- Multiple FINS applications, such as the CX-Programmer, on the same computer can be connected online to the PLC via Ethernet.

15-3 Basic Setting for Ethernet

15-3-1 Overview of Startup Procedure



Note 1 The local IP address and other parameters can be set from the CX-Programmer.

2 It is not necessary step, and the CX-Integrator version 2.67 or higher (CX-One version 4.51 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 routing tables are used for one or more other nodes on the same network

15-3-2 PLC Setup Procedure

Use the CX-Programmer (Ver. 9.72 or higher) for the CP2E N□□-type CPU Unit Setup, and follow the procedure described below.

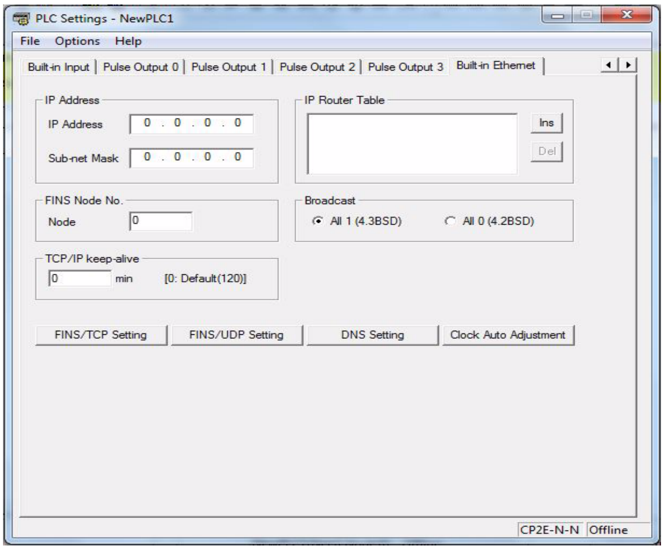
1 Connect the CX-Programmer online.

The CX-Programmer can be connected to the PLC in the following ways:

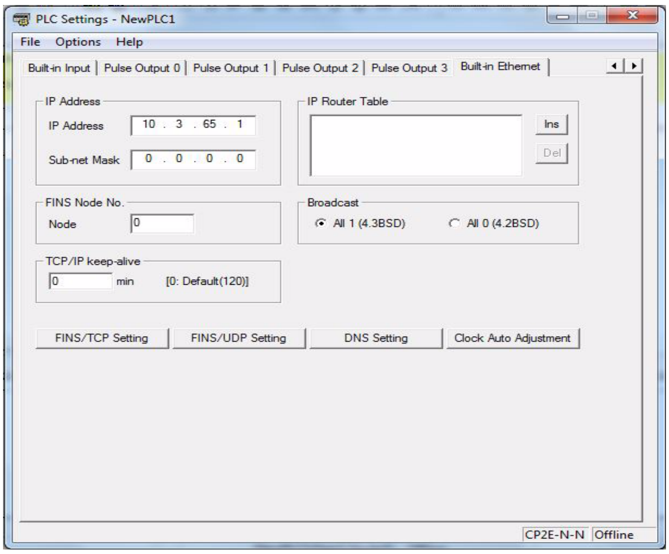
Connect the personal computer to the PLC by Ethernet.
With Auto-IP function, CX-Programmer's online connection command can be executed without any IP setting in computer side when the PLC is physically connected with the computer directly.

For details on connecting the CX-Programmer to the PLC, refer to *4-3 Connection Method with an Ethernet Port* in the *CP2E CPU Unit Hardware User's Manual* (Cat. No. W613).

2 Move the cursor to the **Settings** and double click. Select the Built-in Ethernet Tab to display the window for making the Ethernet Port Setup.
The default settings are shown below.

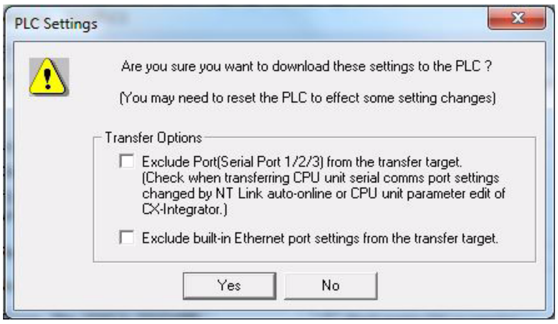


3 Make the required settings (i.e., the IP address in this case).



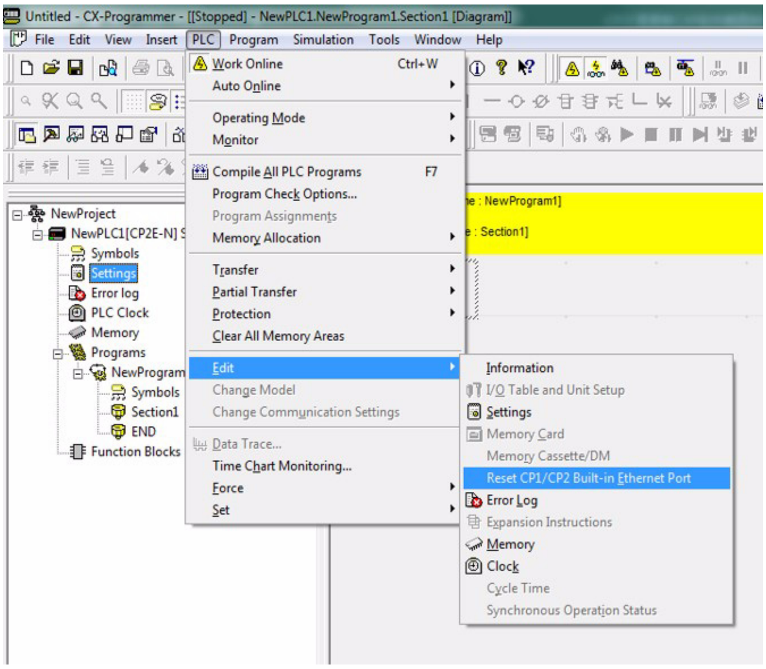
4 Transfer the settings to the PLC.

Click on **Yes** in the following dialog box.



5 In order for the Ethernet Setup to go into effect, the Ethernet Port must be restarted.

Please use the following way to reset the Ethernet Port.



After the LNK/ACT indicator has turned OFF and then turned ON again (Ethernet cable should be connected), the Ethernet port will recognize the new settings.

15-3-3 Basic Settings

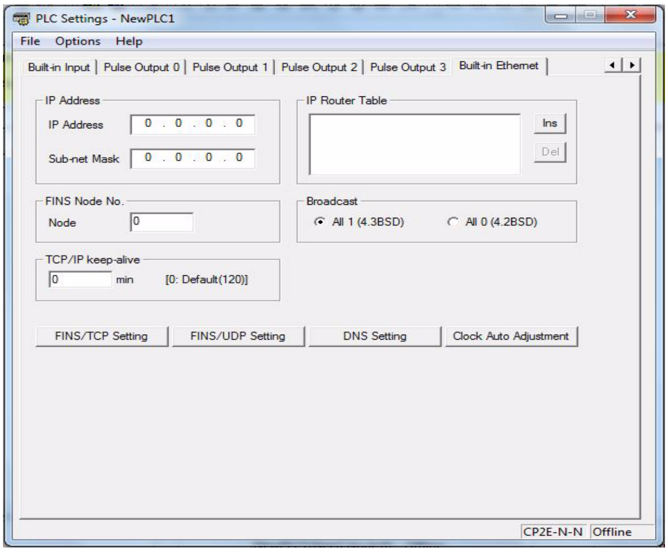
The following items comprise the basic settings in the PLC's Ethernet port setup.

Basic Setting

CX-Programmer tab	Settings
Built-in Ethernet	IP address
	Subnet mask
	Broadcast
	TCP/IP keep-alive
	IP router table

CX-Programmer Setup

Move the cursor to the **Settings** and double click. Select the Built-in Ethernet Tab in PLC setup dialog.



Item	Contents	Default
IP Address	Set the local IP address.	0.0.0.0 (192.168.250. FINS node address)
Sub-net Mask	Set the subnet mask, which supports CIDR. This is required if a method other than the IP address table method is used for address conversion. The subnet mask can be set between 192.0.0.0 and 255.255.255.252.	0.0.0.0 (default net mask for IP address setting)
Broadcast	Set the method for specifying IP addresses for broadcasting in FINS/UDP. • 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.	All 1 (4.3BSD)
TCP/IP keep-alive	Set the liveness-checking interval. When socket services using either FINS/TCP or TCP/IP are used, the connection will be terminated if there is no response from the remote node (either a server or client) within the time set here. (Enabled for socket services using FINS/TCP or TCP/IP only.) Setting range: 0 to 65,535 minutes This setting applies to the keep-alive setting for each connection set with the FINS/TCP Setting button.	0 (120 minutes)
IP Router Table	Set when the PLC is to communicate through the IP router with nodes on another IP network segment.	None

Note 1 Make settings using the PLC settings function in the CX-Programmer (to be included in version 9.72 and higher).

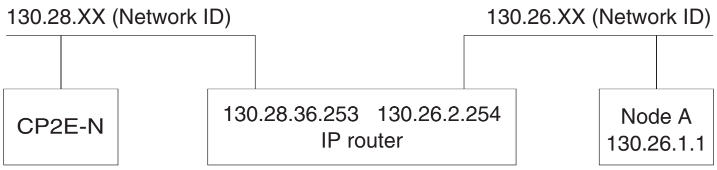
2 For details, refer to *2-9 Basic Settings* in the *Ethernet Units Construction of Networks Operation Manual* (Cat. No. W420).

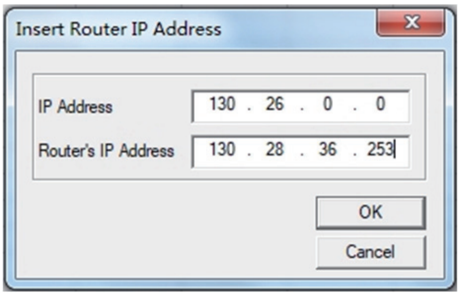
IP Router Table

An IP router table is a table of correspondences for finding IP addresses for the IP routers that relay target segments when the Unit communicates via IP routers with nodes on other IP network segments.

● Setting Example

In this example setting for Node A, a network with a Network ID of 130.26.1.1 is connected to an IP router with an IP address of 130.28.36.253.





Set the IP Address to the network number (i.e., the net ID) of the other IP network segment with which communications are to be executed. The length of the network number (i.e., the number of bytes) will vary depending on the IP address class. Four bytes are reserved for setting the IP Address, so set the network number from the beginning and then set 00 in the remaining space.

A maximum of eight settings can be registered. The default is no setting.

Only one default IP router can be set.

The default IP router is selected when no Network ID exists in the IP router table for the destination network number. To set the default IP router, set 0.0.0.0 for the IP address and set the default IP router's IP address for the router address.

15-3-4 Communications Test

If the basic settings (in particular the IP address and subnet mask) have been made correctly, then it should be possible to communicate with nodes on the Ethernet.

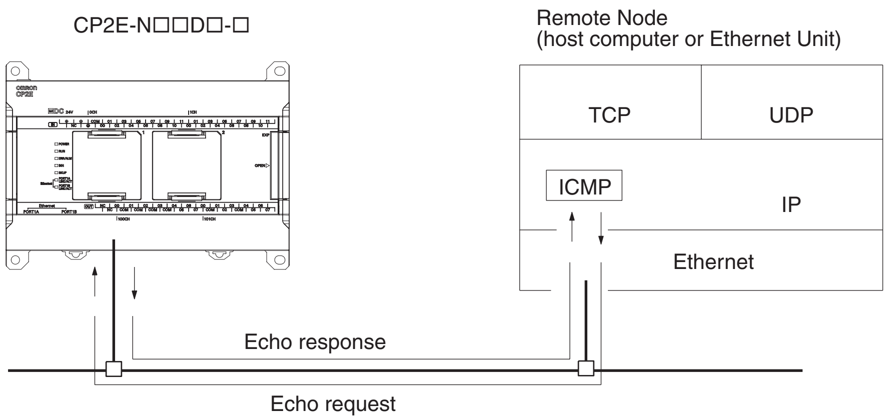
The following describes how to use the PING command to perform communications testing between CP2E N□□-type CPU Units.

PING Command

The PING command sends an echo request packet to a remote node and receives an echo response packet to confirm that the remote node is communicating correctly. The PING command uses the ICMP echo request and responses. The echo response packet is automatically returned by the ICMP.

The PING command is normally used to check the connections of remote nodes when configuring a network. The PLC supports the ICMP echo reply functions.

If the PLC returns a normal response to the PING command, then the remote nodes are physically connected correctly.



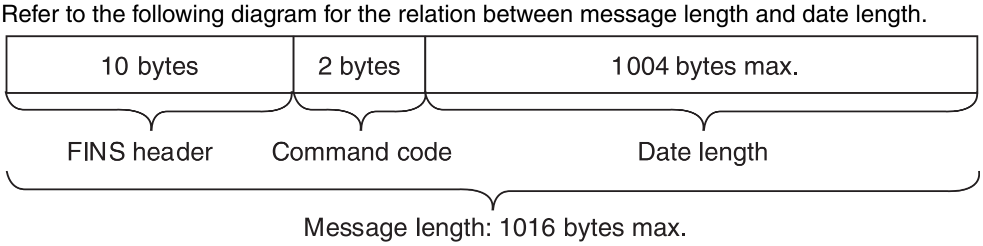
The PLC automatically returns the echo response packet in response to an echo request packet sent by another node (host computer or other Ethernet Unit).

15-4 FINS Communications

15-4-1 FINS Communications Service Specifications

Item	Specification	
Number of nodes	254	
Message Length	1016 bytes max.	
Date Length (See note 1.)	1004 bytes max.	
Number of buffer	16	
Protocol name	FINS/UDP method	FINS/TCP method
Protocol used	UDP/IP	TCP/IP
	The selection of UDP/IP or TCP/IP is made by means of the FINS/UDP or FINS/TCP button in Built-in Ethernet Tab in the CX-Programmer's PLC Setup.	
Number of connections	---	3 for user, 1 for CX-Programmer auto connection
Port number (See note 2.)	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 • Broadcast • Address conversion method	Items set for each connection • Server/client specification • Remote IP address specification Server: Specify IP addresses of clients permitted to connect. Client: Specify remote Ethernet Unit (server) IP address. • Automatic FINS node address allocation: Specify automatic allocation of client FINS node addresses. • Keep-alive: Specify whether remote node keep-alive is to be used.
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. The following functions are enabled by using this table. • IP address conversion using the FINS/UDP method • Automatic FINS node address conversion after a connection is established using the FINS/TCP method • Automatic client FINS node address allocation using the FINS/TCP method • Simultaneous connection of multiple FINS applications	

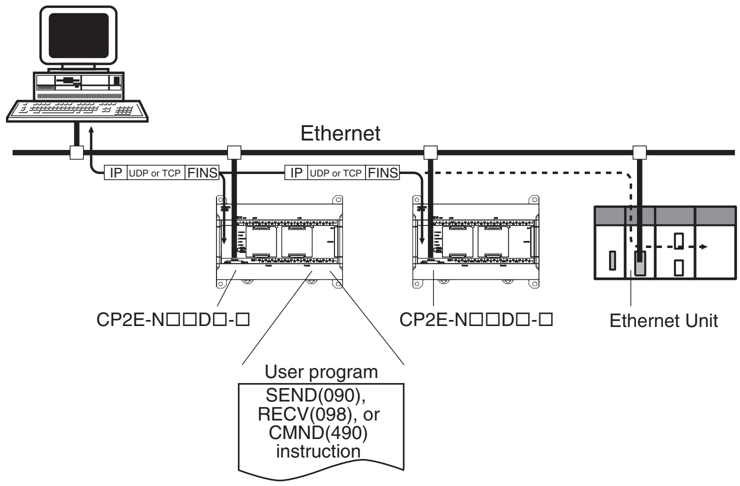
Note 1 Refer to the following diagram for the relation between message length and date length.



2 The FINS/UDP and FINS/TCP port numbers for CX-Programmer Auto Connection are always set to 9600, although the FINS/UDP and FINS/TCP port numbers for user are set to other values.

15-4-2 FINS Communications Service

FINS commands can be sent to or 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.



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.

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.

15-4-3 Procedure for Using FINS/UDP, FINS/TCP

Procedure for Using FINS/UDP

1. Make the basic settings.
   Refer to *15-3-3 Basic Settings*.

⇩

2. Make the settings for FINS/UDP in the PLC Setup with CX-Programmer.

⇩

3. Make the routing table settings and transfer them to each PLC. (See note.)
   Set the routing tables with CX-Integrator, and transfer it to each PLC.

⇩

4. Create a ladder program that includes the SEND, RECV, and CMND instructions.

Note Routing tables are required in the following situations:

- When communicating with a PLC or computer on another network (e.g., remote programming or monitoring using FINS messages or a CX-Programmer).
- When routing tables are used for one or more other nodes on the same network.
  It is not necessary to set routing tables if the nodes are connected as one network.
- Routing table can be set by CX-Integrator version 2.67 or higher (CX-One version 4.51 or higher). For the details of the routing table, refer to *CX-Integrator Ver.2.□ Operation Manual* (Cat. No. W464).

Procedure for Using FINS/TCP

1. Make the basic settings. Refer to *15-3-3 Basic Settings*.
⇩
2. Make the settings for FINS/TCP in the PLC Setup with CX-Programmer.
⇩
3. Make the routing table settings and transfer them to each PLC. (See note.) Set the routing tables with CX-Integrator, and transfer it to each PLC.
⇩
4. Create a ladder program that includes the SEND, RECV, and CMND instructions.

Note Routing tables are required in the following situations:

- When communicating with a PLC or computer on another network (e.g., remote programming or monitoring using FINS messages or a CX-Programmer).
- When routing tables are used for one or more other nodes on the same network.
  It is not necessary to set routing tables if the nodes are connected as one network.
- Routing table can be set by CX-Integrator version 2.67 or higher (CX-One version 4.51 or higher). For the details of the routing table, refer to *CX-Integrator Ver.2.□ Operation Manual* (Cat. No. W464).

15-4-4 PLC Setup for FINS/UDP and FINS/TCP Applications

Aside from the basic settings, the required settings vary depending on the particular communications applications that are used. All these settings are in the Built-in Ethernet Tab. Click on the relative button can open the setup dialog.

FINS/UDP and FINS/TCP Setting

● FINS/UDP

Button name	Settings
FINS/UDP Setting	Conversion
	FINS/UDP Port
	IP Address Table
	Destination IP Address Change Dynamically

● FINS/TCP

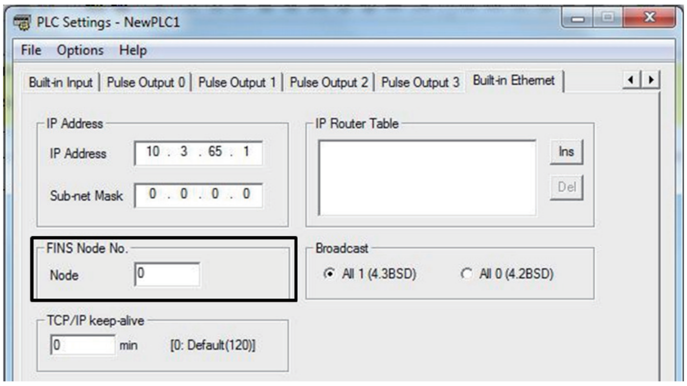
Button name	Settings
FINS/TCP Setting	FINS/TCP Port
	FINS/TCP Connection Setting

CX-Programmer Setup

● FINS/UDP

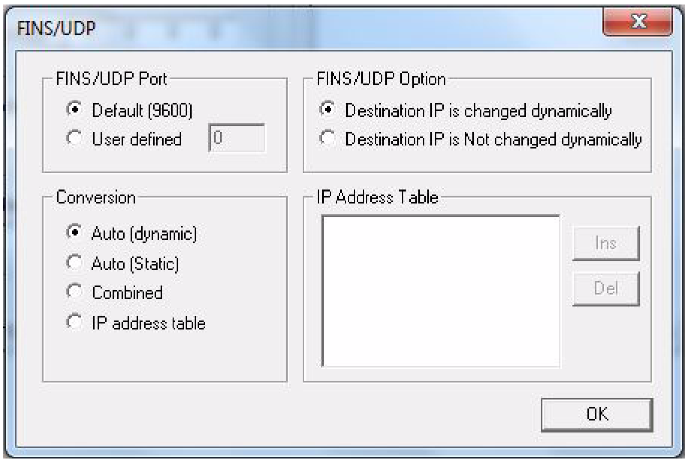
Move the cursor to the **Settings** and double click. Select the Built-in Ethernet Tab. Click the **FINS/UDP Setting** button to display the FINS/UDP setup dialog.

Built-in Ethernet Tab



Item	Contents	Default
Fins Node No.	Set the node address of the CP2E N☐☐-type CPU Unit. • When conversion method is set to Auto(Dynamic/Static), Fins Node Address = Host ID of IP Address. IP Address: xx.xx.xx.FINS Node Address • When conversion method is set to IP address table or Combined, the node address can be set regardless of the host ID of IP address.	Node address 1 (Setting value 0)

FINS/UDP



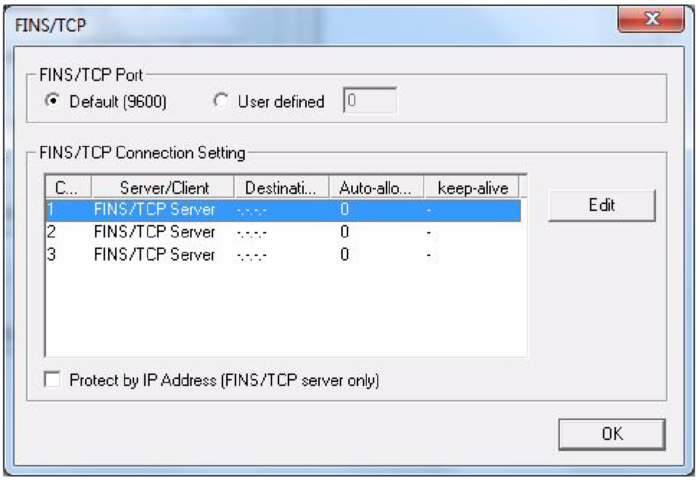
Item	Contents	Default
FINS/UDP Port	Specify the local UDP port number to be used for the FINS communications service. The UDP port number is the number used for UDP identification of the application layer (i.e., the FINS communications service in this case). • Default (9,600) • User defined (Setting range: 1 to 65,535) **Note** Make the settings so that UDP port number does not overlap with port number 123 for SNTP and port number 53 for DNS.	0 (9,600)
Conversion	Select any of the following as the method for finding and converting IP addresses from FINS node addresses. (Enabled for FINS/UDP only.) • Automatic generation (dynamic): Auto (dynamic) • Automatic generation (static): Auto (Static) • IP address table method: Table used • Combined method: Mixed	Auto (dynamic)
Destination IP Address (Change Dynamically)	Select to dynamically change the remote (destination) IP address for FINS/UDP. To prohibit dynamic changes, deselect this box.	Checked (Change Dynamically)
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 or combined method is set as the IP address conversion method.	None

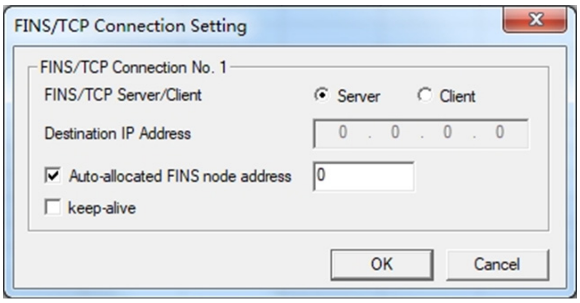
Note 1 Make settings using the PLC settings function in the CX-Programmer (to be included in version 9.72 and higher).

2 For the details of IP address conversion, IP address dynamically changing and IP address table, refer to *Section 5 Determining IP Addresses* in the *CS/CJ Series Ethernet Units Construction of Networks Operation Manual* (Cat. No. W420).

● FINS/TCP

Move the cursor to the **Settings** and double click. Select the Built-in Ethernet Tab. Click the **FINS/TCP Setting** button to display the FINS/TCP setup dialog.





Item	Contents	Default
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). • Default (9,600) • User defined (Setting range: 1 to 65,535)	0 (9,600)
FINS/TCP connection Setting	Shows the connection number. This is a network API used when TCP is used for the FINS communications service. It corresponds to a socket in the socket services. Up to 3 can be used at a time, and they are identified by connection numbers 1 to 3. The PLC can thus simultaneously execute the FINS communications service by TCP with up to 3 remote nodes.	
Protect by IP Address	When this option is selected, if the PLC is set for use as a server, and if a connection number other than 0.0.0.0 is set for a destination IP address, any connection request from other than the number set for that IP address will be denied. Select this option to prevent faulty operation (by FINS commands) from specific nodes from affecting the PLC.	Not protected

The following settings can be made for each connection number.

Item	Contents	Default
FINS/TCP Server/Client	For each connection number, this setting specifies the PLC for use as either a server or a client. • When the PLC is used as a server: The PLC opens a connection with that connection number and waits for service requests from clients. Connection numbers are used in ascending order and allocated to clients in the order connections are made. • When the PLC is used as a client: The PLC establishes a connection with the server set as the destination IP address. Once the connection has been established, FINS/TCP is used for FINS communications.	Server
Destination IP Address	• When the PLC 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 of 0.0.0.0 can be used. • When the PLC is used as a client: Set the IP address for the remote PLC (i.e., the server) that is to be connected by FINS/TCP. It is required that an IP address be set for the remote PLC.	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 253) will automatically be allocated as the client's FINS node address.	Connection No.1 to 3 No.1: 0 (251) No.2: 0 (252) No.3: 0 (253)
keep-alive	For each connection number, set whether or not the remote node connection check function is to be used for the FINS/TCP server and client. If the keep-alive box is checked here, then, when the remote node goes without responding for longer than the monitor time set in the Setup, the connection will be terminated. If a remote node turns OFF without warning, the connection will remain open indefinitely, so this option should be used whenever possible.	Not use

For details, refer to *SECTION 6 FINS Communications Service* in the *Ethernet Units Operation Manual Construction of Networks* (Cat. No. W420).

15-4-5 Auxiliary Area Allocations

The following table and descriptions cover the words and bits in the Auxiliary Area of PLC memory that are related to the FINS/UDP and FINS/TCP.

Address	Bit(s)	Name	Status	Unit operation	Access
A47	0	FINS/TCP Connection Flag 1	ON	Turned ON by the Unit when a connection is established.	Read only
			OFF	Turned OFF by the Unit when the connection is terminated.	
	1	FINS/TCP Connection Flag 2	ON	Turned ON by the Unit when a connection is established.	
			OFF	Turned OFF by the Unit when the connection is terminated.	
	2	FINS/TCP Connection Flag 3	ON	Turned ON by the Unit when a connection is established.	
			OFF	Turned OFF by the Unit when the connection is terminated.	

15-4-6 New FINS Commands

New FINS Commands Code List

The command codes listed in the following table are new added commands to CP2E N☐☐-type CPU Units.

For the details of other FINS commands, refer to the *SYSMAC CS/CJ/CP/NSJ-series Communications Commands Reference Manual* (Cat. No. W342).

Command code		Name	Unit address
MRC	SRC		
04	03	RESET	0xFA
05	01	ETHERNET PORT DATA READ	
27	30	FINS/TCP CONNECTION REMOTE NODE CHANGE REQUEST	0x00
	31	FINS/TCP CONNECTION STATUS READ	
	50	IP ADDRESS TABLE WRITE	
	51	IP ROUTER TABLE WRITE	
	60	IP ADDRESS TABLE READ	
	61	IP ROUTER TABLE READ	

Note 1 There are two unit address (DA2 in FINS header) used in CP2E N☐☐-type CPU Units.

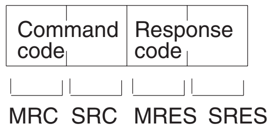
0xFA	It is used for the two new added FINS commands relative with built-in Ethernet Port. (Ethernet controller reset command and Ethernet Port information read command)
0x00	It is used for the FINS commands other than the two commands above.

2 Ethernet controller reset command and Ethernet Port information read command can only be executed by other PLCs or computers on the Ethernet network. It cannot be executed in the local node by CMND instruction.

Response Code List

Response codes are 2-byte codes which indicate the results of command execution. They are returned in the response following the command code.

The first byte of a response code is the MRES (main response code), which categorizes the results of command execution. The second byte is the SRES (sub-response code) which specifies the results.



MRC: Main request code
SRC: Sub-request code
MRES: Main response code
SRES: Sub-response code

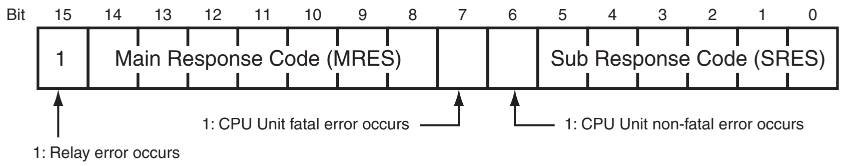
Additional Information

Certain bits (6, 7 or 15) of the response code will be ON.

When bit 6 or 7 turns ON, it indicates that an error has occurred in the destination CPU Unit. Refer the the CPU Unit manual to remove the error.

When bit 15 turns ON, it indicates that an error has occurred during the network relay.

The following diagram describes what is a response code.



The MRES codes are shown in the following table along with the results they indicate.

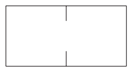
MRES	Execution results
00	Normal completion
01	Local node error
02	Remote node error
03	Unit error (controller error)
04	Service not supported
05	Routing error
10	Command format error
11	Parameter error
22	Status error
23	Operating environment error
25	Unit error

Refer to the *SYSMAC CS/CJ/CP/NSJ-series Communications Commands Reference Manual* (Cat. No. W342) or the operation manuals for the relevant unit for further information on response codes.

Command/Response Reference

This section describes the FINS commands that can be sent to PLC's Ethernet module and the responses to each command.

The command, response, and (where applicable) the results storage blocks are given with the commands in graphic form as shown in the following diagram. If the data is fixed, it is included in the blocks. If the data is variable, it is described following the blocks. Each box represents 1 byte; every two boxes represents 1 word. The following diagram shows 2 bytes, or 1 word.



Two bytes

The results storage format is the format used to store transfer results.

Response codes applicable to the command are described at the end of the command description. If any UNIX error codes are generated, these are also described. Refer to your UNIX error symbol definition file /usr/include/sys/errno.h for details. UNIX errors are returned in the results storage area.

Note Except for special cases, all send/receive data is in hexadecimal format.

New FINS Commands Addressed to Built-in Ethernet Port (0xFA)

● Command Code List

The command codes listed in the following table can be sent to the built-in Ethernet port.

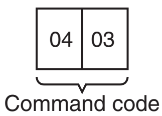
The destination unit address (DA2) in FINS frame should be set as 0xFA.

Command code		Name
MRC	SRC	
04	03	RESET
05	01	ETHERNET PORT DATA READ

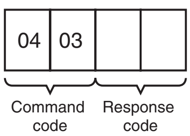
● RESET: 0403

Reset the Ethernet Unit.

Command Block



Response Block



Precautions for Correct Use

No response will be returned if the command ends normally. A response will be returned only if an error occurs.

In some cases, send requests (SEND/RECV instructions) made from the PLC to the built-in Ethernet port just before execution of the RESET command may not be executed.

Except for the FINS communications service sockets, all open sockets (for sockets services) are closed immediately before resetting.

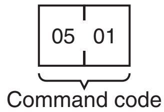
Response Codes

Response code	Description
1004	Command format error

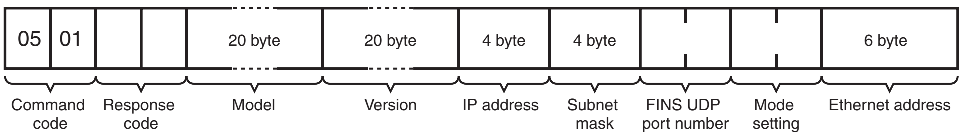
● ETHERNET PORT CONTROLLER DATA READ: 0501

Reads the following data from the Ethernet port, PLC model, PLC version, IP address, subnet mask, FINS UDP port number, mode settings, Ethernet address.

Command Block



Response Block



Parameters

Model, Version (Response)

The PLC model and version are returned as ASCII characters occupying 20 bytes each (i.e., 20 characters each). If all bytes are not used, the remaining bytes will be all spaces (ASCII 20 Hex).

Example Model:	CP2E-ETN21
Version:	V1.00

IP Address, Subnet Mask (Response)

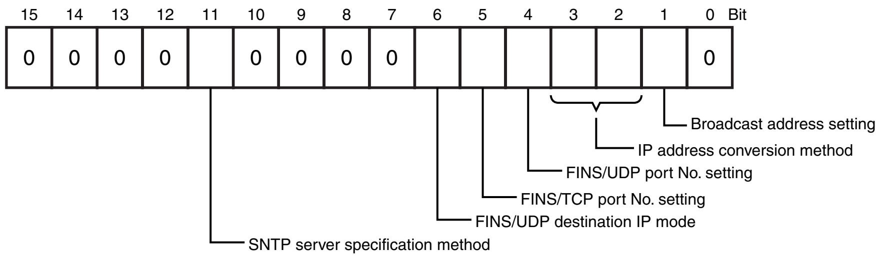
The built-in Ethernet port's IP address and subnet mask are returned as 4 bytes each.

FINS UDP Port Number (Response)

The built-in Ethernet port's UDP port number for FINS is returned as 2 bytes.

Mode Setting (Response)

The mode setting in the system setup is returned.



- **Broadcast Address Setting**

  0: Broadcast with host number set to all ones (4.3BSD specifications)

  1: Broadcast with host number set to all zeroes (4.2BSD specifications)

- **IP Address Conversion Method Setting**

  00, 01: Automatic generation method

  10: IP address table reference method

  11: Combined method (IP address table reference + automatic generation)

- **FINS/UDP Port Number Setting**

  0: Default (9600)

  1: Unit Setup value

- **FFINS/TCP Port Number Setting**

  0: Default (9600)

  1: Unit Setup value

- **FINS/UDP Destination IP Mode**

  0: Dynamical mode

  1: Static mode

- **SNTP Server Specification Method**

  0: IP address

  1: Host name

Ethernet Address (Response)

The Ethernet address of the Ethernet port is returned. The Ethernet address is the address marked on the label on the top of the PLC.

Response Codes

Response code	Description
0000	Normal
1004	Command format error

New FINS Commands Addressed to CPU Port (0x00)

● Command Code List

This section describes the new FINS commands that can be sent to the CPU port and the responses that are returned.

The command codes listed in the following table can be sent to the CPU port.

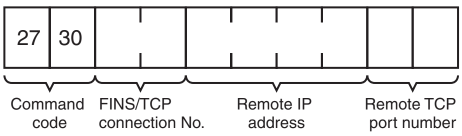
The destination unit address (DA2) in FINS frame should be set as 0x00.

Command code		Name
MRC	SRC	
27	30	FINS/TCP CONNECTION REMOTE NODE CHANGE REQUEST
	31	FINS/TCP CONNECTION STATUS READ
	50	IP ADDRESS TABLE WRITE
	51	IP ROUTER TABLE WRITE
	60	IP ADDRESS TABLE READ
	61	IP ROUTER TABLE READ

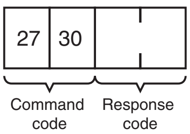
● FINS/TCP CONNECTION REMOTE NODE CHANGE REQUEST: 2730

Requests a remote node change for the FINS/TCP connection.

Command Block



Response Block



Parameters

FINS/TCP Connection No. (Command)

Specifies, in two bytes, the FINS/TCP connection number (1 to 3) for which the change is to be made.

Remote IP Address (Command)

Specifies the remote node’s IP address (must be non-zero) in hexadecimal.

Remote Port Number (Command)

Specifies the remote TCP port number (must be non-zero) with this command.

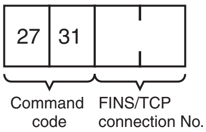
Response Codes

Response code	Description
0000	Normal
0105	Node address setting error Local IP address setting error
1004	Command format error
1100	Connection number not set from 1 to 3 Remote IP address set to 0 Remote TCP port number set to 0
2230	Connection already established with specified remote node
2231	Specified connection number not set as FINS/TCP client in Unit Setup
2232	Remote node change processing for specified connection number aborted because change request received during processing

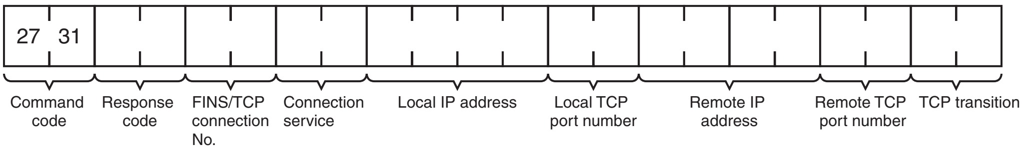
● FINS/TCP CONNECTION STATUS READ: 2731

Reads the FINS/TCP connection status.

Command Block



Response Block



Parameters

FIFINS/TCP Connection No. (Command, Response)

Command: Specifies, in two bytes, the FINS/TCP connection number (1 to 3) for which the status is to be read.

Response: Specifies the FINS/TCP connection number (1 to 3) for which the status was read.

Connection Service (Response)

Specifies the service that is being used for the FINS/TCP connection as a number.

0003: FINS/TCP server

0004: FINS/TCP client

Local IP Address (Response)

Specifies the IP address for the local node in hexadecimal.

Local TCP Port Number (Response)

Specifies the TCP port number for the local node.

Remote IP Address (Response)

Specifies the IP address for the remote node in hexadecimal.

Remote TCP Port Number (Response)

Specifies the TCP port number for the remote node.

TCP Transitions (Response)

Specifies the TCP connection status using the following numbers.

For details on TCP status changes, refer to *A-6-1 TCP Status Transitions*.

Number	Status	Meaning
00000000	CLOSED	Connection closed.
00000001	LISTEN	Waiting for connection.
00000002	SYN SENT	SYN sent in active status.
00000003	SYN RECEIVED	SYN received and sent.
00000004	ESTABLISHED	Already established.
00000005	CLOSE WAIT	FIN received and waiting for completion.
00000006	FIN WAIT 1	Completed and FIN sent.
00000007	CLOSING	Completed and exchanged FIN. Awaiting ACK.
00000008	LAST ACK	FIN sent and completed. Awaiting ACK.
00000009	FIN WAIT 2	Completed and ACK received. Awaiting FIN.
0000000A	TIME WAIT	After closing, pauses twice the maximum segment life (2MSL).

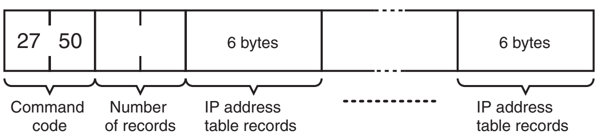
Response Codes

Response code	Description
0000	Normal
0105	Node address setting error Local IP address setting error
1004	Command format error
1100	Connection number not set from 1 to 3

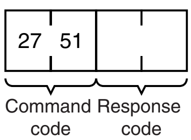
● IP ADDRESS TABLE WRITE: 2750

Writes the IP address table.

Command Block



Response Block



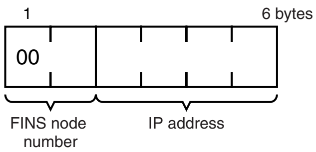
Parameters

Number of Records (Command)

The number of records to write is specified in hexadecimal between 0000 and 0020 (0 to 32 decimal) in the command. If this value is set to 0, the IP address table will be cleared so that no records are registered.

IP Address Table Records (Command)

Specify the IP address table records. The number of records specified must be provided. The total number of bytes in the IP address table records is calculated as the number of records × 6 bytes/record. The configuration of the 6 bytes of data in each record is as shown in the following diagram.



- **FINS Node Address**

  Node address for communications via the FINS command (hexadecimal).

- **IP Address**

  IP address used by TCP/IP protocol (hexadecimal).

Precautions for Correct Use

The new I/O address table records will not be effective until the PC is restarted or the Ethernet Unit is reset.

An error response will be returned if the IP address conversion method in the system mode settings is set for automatic generation.

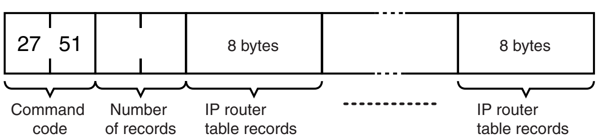
Response Codes

Response code	Description
0000	Normal (echo reply received from the remote node)
1004	Command format error
1003	The number of records specified does not match the sent data length.
110C	The number of records is not between 0 and 32. The FINS node address is not between 1 and 126 The IP address is 0.
2307	IP address conversion method is set for automatic generation.

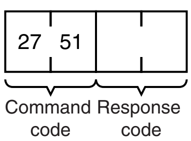
● IP ROUTER TABLE WRITE: 2751

Writes the IP router table.

Command Block



Response Block



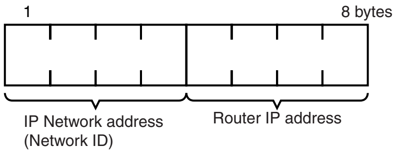
Parameters

Number of Records (Command)

The number of records to write is specified in hexadecimal between 0000 and 0008 in the command. If this value is set to 0, the IP router table will be cleared so that no records are registered.

IP Router Table Records (Command)

Specify the IP router table records. The number of records specified must be provided. The total number of bytes in the IP router table records is calculated as the number of records × 8 bytes/record. The configuration of the 8 bytes of data in each record is as shown in the following diagram.



- **IP Network Address**

  The network ID from the IP address in hexadecimal. The network ID part corresponding to the address class (determined by the leftmost 3 bits) set here, is enabled.

- **Router IP Address**

  The IP address (in hexadecimal) of a router connected to a network specified with IP addresses.

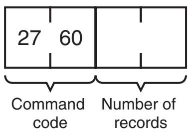
Response Codes

Response code	Description
0000	Normal
1004	Command format error
1003	The number of records specified does not match the sent data length.
110C	The number of records is not between 0 and 8. The router IP address is 0.

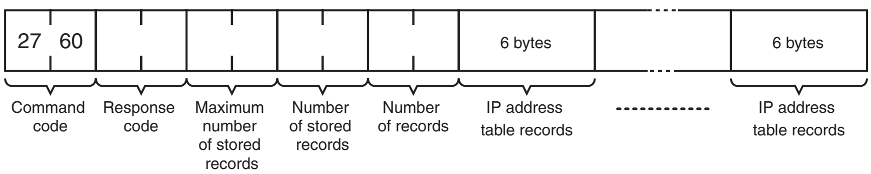
● IP ADDRESS TABLE READ: 2760

Reads the IP address table.

Command Block



Response Block



Parameters

Number of Records (Command, Response)

The number of records to read is specified between 0000 and 0020 (0 to 32 decimal) in the command. If this value is set to 0, the number of stored records is returned but the IP address table records are not returned. The response returns the actual number of records read.

Maximum Number of Stored Records (Response)

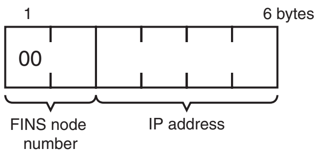
The maximum number of records that can be stored in the IP address table is returned. The maximum number of stored records is fixed at 0020 (32 records).

Number of Stored Records (Response)

The number of IP address table records stored at the time the command is executed is returned as a hexadecimal number.

IP Address Table Records (Response)

The number of IP address table records specified in the number of records parameter is returned. The total number of bytes in the IP address table records is calculated as the number of records × 6 bytes/record. The configuration of the 6 bytes of data in each record is as shown in the following diagram.



- **FINS Node Address**

  Node address for communications via the FINS command (in hexadecimal).

- **IP Address**

  IP number used by TCP/IP protocol (in hexadecimal).

Precautions for Correct Use

If the IP address table contains fewer records than the number specified in the number of records parameter, all the records contained in the IP address table when the command is executed will be returned and the command execution will end normally.

An error response will be returned if the IP address conversion method in the system mode settings is set to the automatic generation method.

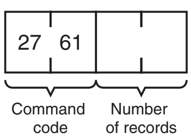
Response Codes

Response code	Description
0000	Normal
1004	Command format error
2307	IP address conversion method is set to the automatic generation method.

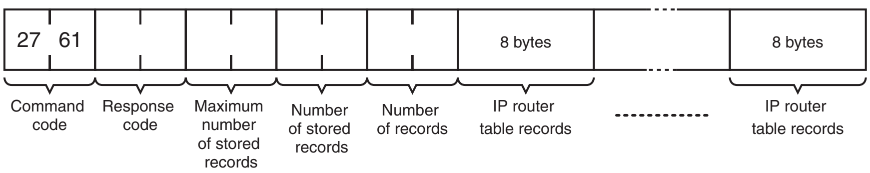
● IP ROUTER TABLE READ: 2761

Reads the IP router table.

Command Block



Response Block



Parameters

Number of Records (Command, Response)

The number of records to read is specified between 0000 and 0008 (0 to 8 decimal) in the command. If this value is set to 0, the number of stored records will be returned but the IP router table records will not be returned. The response returns the actual number of records read.

Maximum Number of Stored Records (Response)

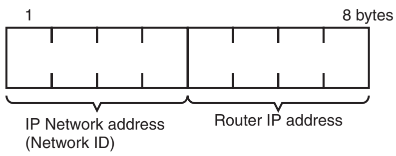
The maximum number of records that can be stored in the IP router table is returned. The maximum number of stored records is fixed at 0008 (8 records).

Number of Stored Records (Response)

The number of IP router table records stored at the time the command is executed is returned in hexadecimal.

IP Router table Records (Response)

The number of IP router table records specified in the *number of records* parameter is returned. The total number of bytes in the IP router table records is calculated as the number of records × 8 bytes/record. The configuration of the 8 bytes of data in each record is shown below.



- **IP Network Address**

  The network ID from the IP address in hexadecimal. The network ID part corresponding to the address class (determined by the leftmost 3 bits) set here, is enabled.

- **Router IP Address**

  The IP address (in hexadecimal) of a router connected to a network specified with IP addresses.

Precautions for Correct Use

If the IP router table contains fewer records than the number specified in the number of records parameter, all the records contained in the IP router table when the command is executed will be returned and the command execution will end normally.

Response Codes

Response code	Description
0000	Normal
1004	Command format error

15-4-7 CMND/SEND/RECV Instructions

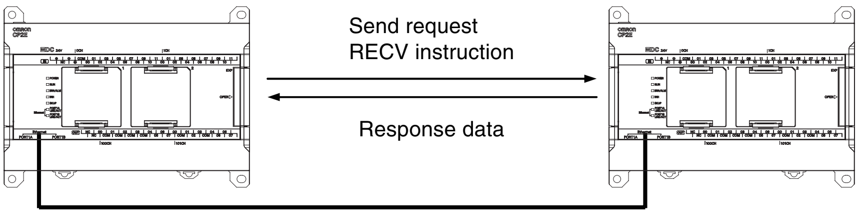
The data and FINS commands can be transmitted between the CP2E N□□-type CPU Unit and other devices using the CMND, SEND or RECV instuction.

Setting the network address and the node address of the instrction in the CP2E N□□-type CPU Unit ladder program, it is possible to send the data and FINS commands to another device, or receive data from another device.

Sample Program

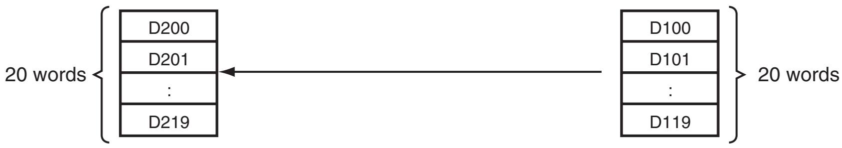
①CP2E N□□-type CPU Unit requests the data to be transmitted from ②CP2E N□□-type CPU Unit (node address 3) in the local network and receives the data.

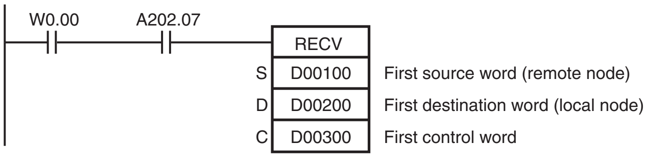
When W0.00 and A202.7 (the Communications Port Enabled Flag for port 07) are ON, 20 words are read from D100 to D119 of ②CP2E N□□-type (node address 3), transmitted to ①CP2E N□□-type and stored in D200 to D219 using the RECV instruction.



① CP2E N□□-type CPU Unit
Port number 7
Response requested
Retries: 3
Response monitoring time 10s

② CP2E N□□-type CPU Unit
Node address: 3
Unit address: 00 (CPU Unit)



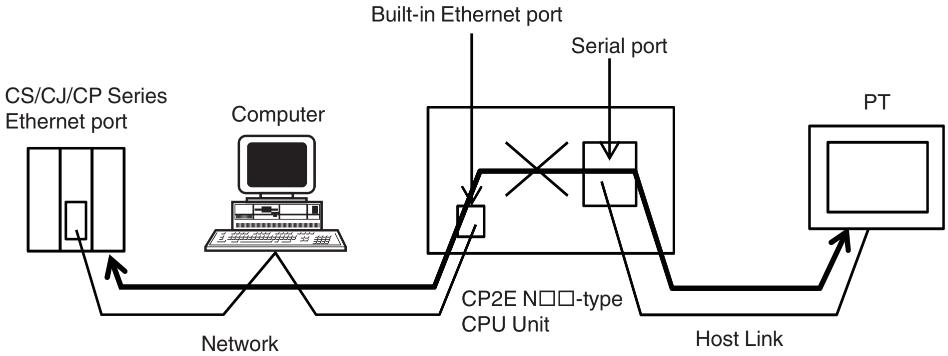


	15	87	0	
D: D00300	0 0	1 4		Number of words to receive (send request): 0014 hex (20 words)
C+1: D00301	0 0	0 0		Source network address: 00 hex (local network)
C+2: D00302	0 3	0 0		Source node address: 3 hex, Source unit address: 00 hex (CPU Unit)
C+3: D00303	0 7	0 3		Response requested (fixed), Logical port number: 7 hex, Retries: 3 hex
C+4: D00304	0 0	6 4		Response monitoring time: 0064 hex (10 sec.)

15-4-8 Restrictions When Using FINS Communication Services

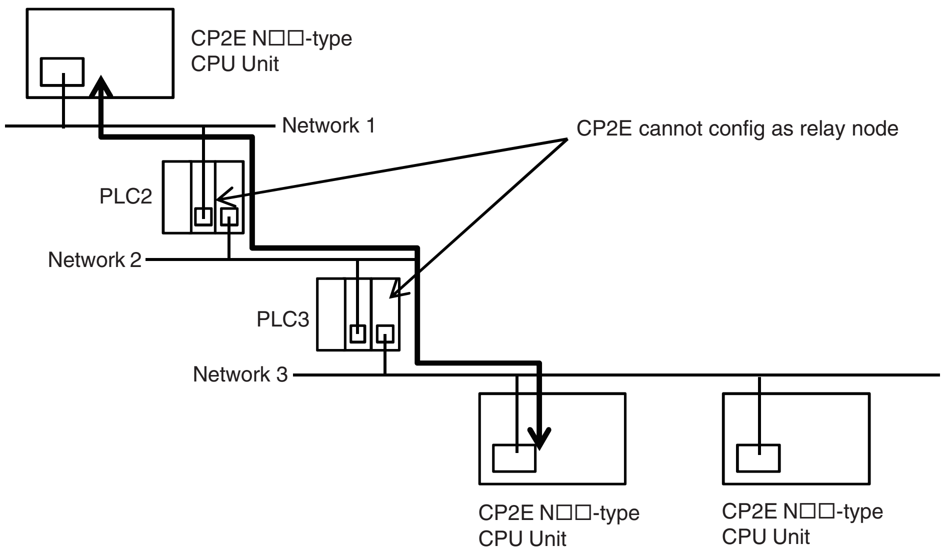
CP2E CPU Unit does not support network relay function.

The PTs and host computers connected to the CP2E N☐☐-type CPU Unit cannot communicate (such as FINS message communication, remote programming or monitoring by the CX-Programmer) with the PLCs or computers on the network by Host Link.



CP2E N☐☐-type CPU Unit can be only configed as the end point of the network.

It cannot be used as a rely node for the network. It can send or receive FINS command through up to 3 layers of the network.



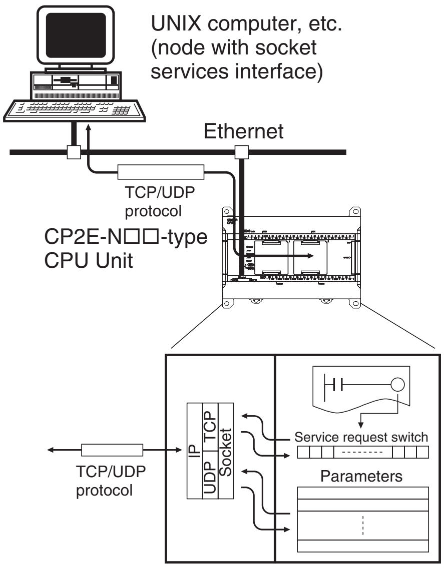
15-5 Socket Services

The socket services allow devices on the Ethernet to send and receive various data using either the UDP or TCP protocol.

15-5-1 Overview of Socket Service

The way to use socket services is to set the required parameters in the parameter area allocated in the DM Area, and then to request particular UDP or TCP socket services by turning ON dedicated control bits in the AR Area. When the PLC has completed the requested process, the same bit is turned OFF to provide notification. Data that is sent or received is automatically handled according to the I/O memory locations specified in the parameter area.

A total of three ports (UDP and TCP combined) can be used for socket services.



15-5-2 Procedure for Using Socket Service Functions

1. Make the basic settings.
   Refer to *15-3-3 Basic Settings*.

⇩

2. Use the CX-Programmer or Programming Console to make the socket service settings in the socket service parameter areas 1 to 3 (m+8 to m+37) allocated in the DM Area.
   Note The first word m in the allocated DM Area = D16000

⇩

3. Select **Transfer to PLC** from the Options Menu, and then click the **Yes** button. The Setup data in the allocated DM Area will be transferred to the CPU Unit.

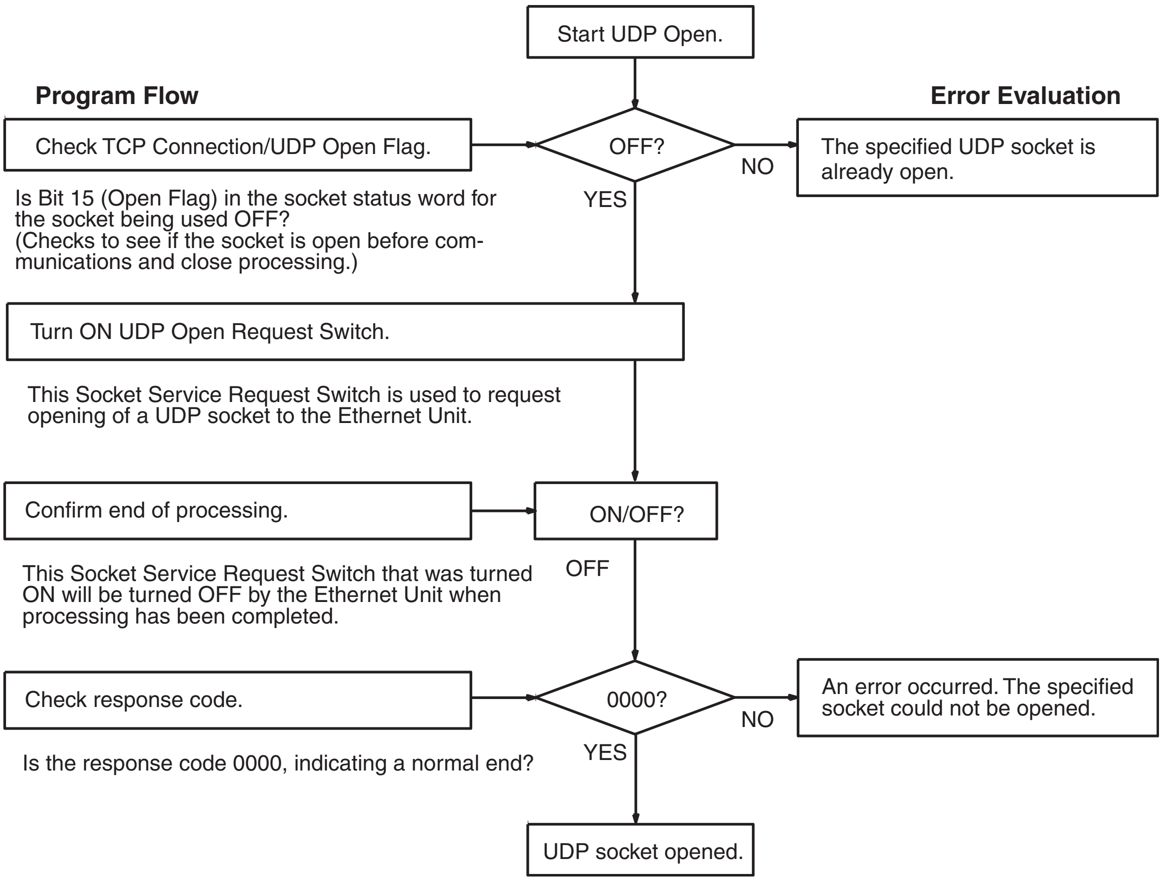
Precautions for Correct Use

A Socket Service Parameter Area cannot be used for other sockets once open processing has been successfully completed for it. Check the socket status before attempting to open a socket. TCP socket status is provided in words m+4 to m+6 in the DM Area for sockets 1 to 3.

When a send or receive request is made, the data will be automatically sent or received according to the send/receive data address in the Socket Service Parameter Area. When processing has been completed, a response code will be automatically stored in the Socket Service Parameters.

15-5-3 Socket Services and Socket Status

When using socket services, it is important to consider the timing of the status changes in the Socket Status Area. The diagram below shows a flowchart for opening UDP. The flow is similar for other socket services. Replace the names of the appropriate flags in the flowchart to adapt it to other socket services.

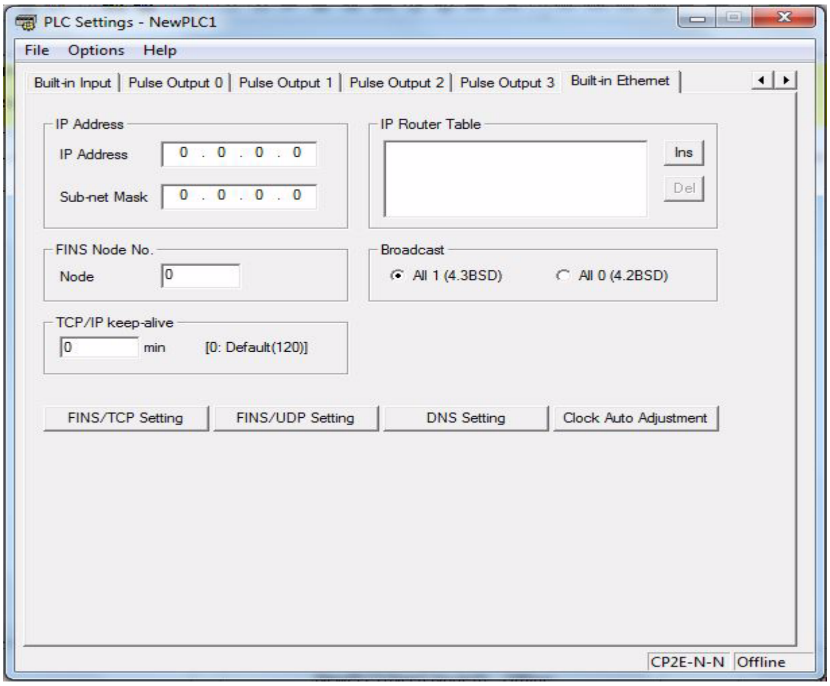


Note For details about timing charts, refer to *SECTION 6 Socket Services* in the *Ethernet Units Construction of Applications Operation Manual* (Cat. No. W421).
Socket services cannot support CMND command.

15-5-4 PLC Setup for Socket Services

Socket Services Setting

CX-Programmer tab	Setting
Built-in Ethernet	TCP/IP keep-alive



Item	Contents	Default
TCP/IP keep-alive	Set the liveness-checking interval. When socket services using either FINS/TCP or TCP/IP are used, the connection will be terminated if there is no response from the remote node (either a server or client) within the time set here. (Enabled for socket services using FINS/TCP or TCP/IP only.) Setting range: 0 to 65,535 minutes This setting applies to the keep-alive setting for each connection set with the FINS/TCP Setting button.	0 (120 minutes)

Note Make settings using the PLC settings function in the CX-Programmer (to be included in version 9.72 and higher).

15-5-5 Auxiliary Area Allocations

The following table and descriptions cover the words and bits in the Auxiliary Area of PLC memory that are related to the socket services.

Ethernet Service Request

Address	Bit(s)	Name	Status	Unit operation	Access
A566	2	Socket Force-close Switch	ON	All sockets are forcibly closed when this bit turns ON.	Read/Write
			OFF	Turned OFF by Unit after sockets are closed.	

● Socket Force-close Switch (Bit 2)

All UDP and TCP sockets used for socket services can be force-closed by turning ON this switch. This can be used for operations such as error processing.

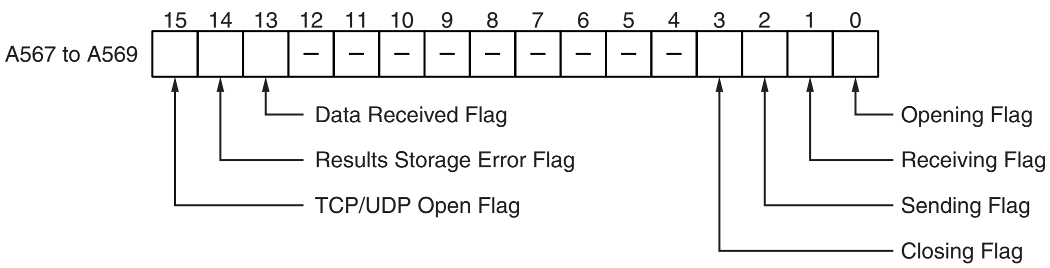
Be careful not to force-close sockets during communications, or an error will occur. After all sockets have been force-closed, the PLC will turn the switch OFF again. Do not attempt to forcibly manipulate this switch before it is automatically turned OFF by the PLC.

Ports used exclusively by the Ethernet Unit will not be closed.

Socket Service

Bit	15 to 8	7 to 0
A567	TCP/UDP Socket No. 1 Status	
A568	TCP/UDP Socket No. 2 Status	
A569	TCP/UDP Socket No. 3 Status	
A570	Reserved	
A571	Socket Service Request Switches 2	Socket Service Request Switches 1
A572	Reserved	Socket Service Request Switches 3

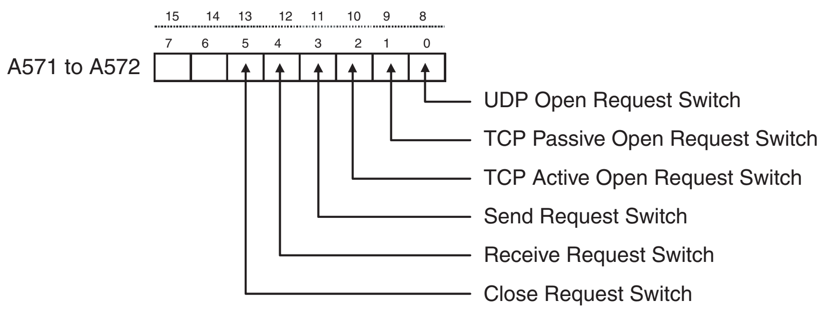
● Status of TCP/UDP Sockets 1 to 3



Bit	Flag	Status	Manipulated by	Unit operation	Access
0	Opening Flag	ON	Unit	ON during open processing. (Turns ON when open request is received.)	Read only
		OFF	Unit	OFF when open processing has been completed.	
1	Receiving Flag	ON	Unit	ON during receive processing. (Turns ON when receive request is received.)	
		OFF	Unit	OFF when receive processing has been completed.	
2	Sending Flag	ON	Unit	ON during send processing. (Turns ON when send request is received.)	
		OFF	Unit	OFF when send processing has been completed.	
3	Closing Flag	ON	Unit	ON during close processing. (Turns ON when close request is received.)	
		OFF	Unit	OFF when close processing has been completed.	
4 to 12	(Not used.)	---	---	---	---
13	Data Received Flag	ON	Unit	ON when data from a remote node has been received at an open TCP socket.	Read only
		OFF	Unit	OFF when receive processing has been requested for an open TCP socket.	
14	Results Storage Error Flag	ON	Unit	CP2E CPU Units are supported by CX-One version 4.51 or higher and CX-Programmer version 9.72 or higher.	
		OFF	Unit	Turns OFF when the next request is received.	
15	TCP/UDP Open Flag	ON	Unit	ON when open processing has been completed.	
		OFF	Unit	OFF when close processing has been completed. (Stays OFF for abnormal open processing completion.)	

Note Do not forcibly manipulate the above status flags during socket service is used.

● Socket Service Request Switches 1 to 3



Bit		Switch	Status	Manipulated by	Unit operation	Access
8	0	UDP Open Request Switch	ON	User	UDP socket opened when switch is turned ON.	Read/Write
			OFF	Unit	Unit turns OFF switch when open processing has been completed (i.e., when a connection has been made).	
9	1	TCP Passive Open Request Switch	ON	User	Passive TCP socket opened when switch is turned ON.	
			OFF	Unit	Unit turns OFF switch when open processing has been completed (i.e., when a connection has been made).	
10	2	TCP Active Open Request Switch	ON	User	Active TCP socket opened when switch is turned ON.	
			OFF	Unit	Unit turns OFF switch when open processing has been completed (i.e., when a connection has been made).	
11	3	Send Request Switch	ON	User	Send processing executed when switch is turned ON. (The protocol (TCP/UDP) is determined when the socket is opened.)	
			OFF	Unit	Unit turns OFF switch when send processing has been completed.	
12	4	Receive Request Switch	ON	User	Receive processing executed when switch is turned ON. (The protocol (TCP/UDP) is determined when the socket is opened.)	
			OFF	Unit	Unit turns OFF switch when receive processing has been completed.	
13	5	Close Request Switch	ON	User	Close processing executed when switch is turned ON. (The protocol (TCP/UDP) is determined when the socket is opened.)	
			OFF	Unit	Unit turns OFF switch when close processing has been completed.	
14	6	Reserved	---	---	---	---
15	7	Reserved	---	---	---	---

15-5-6 Data Memory Area Allocations

The memory allocation about socket service is shown in the following diagram. These data will be allocated to the DM area of the PLC.

Beginning word m = 16000

Offset	Word	Bit 15 to 08, 07 to 00
m	D16000	TCP Socket No. 1 Number of Bytes Received
m+1	D16001	TCP Socket No. 2 Number of Bytes Received
m+2	D16002	TCP Socket No. 3 Number of Bytes Received
m+3	D16003	Reserved
m+4	D16004	TCP Socket No. 1 Connection Status
m+5	D16005	TCP Socket No. 2 Connection Status
m+6	D16006	TCP Socket No. 3 Connection Status
m+7	D16007	Reserved
m+8 m+17	D16008 to D16017	Socket Services Parameter Area 1
m+18 m+27	D16018 to D16027	Socket Services Parameter Area 2
m+28 m+37	D16028 to D16037	Socket Services Parameter Area 3
m+38 m+47	D16038 to D16047	Reserved

● TCP Socket No. (1 to 3): Number of Bytes Received

	15 to 0
m to m+2	Number of Bytes Received (0000 to 0400 hex)

For each TCP socket, the number of bytes of data in the reception buffer is stored in one word. A maximum of 4,096 bytes of data can be held in the reception buffer, but a value of only up to the maximum value (1,024 bytes) that can be set for receive requests by manipulating control bits is stored.

0000 hex: 0 bytes

0400 hex: 1,024 bytes

The Data Received Flag in the CIO Area turns ON and OFF linked to this word. This area is given a value of 0000 hex when a receive request is executed by manipulating control bits. If any data remains in the reception buffer after the receive request processing is completed, the remaining number of bytes is stored and the Data Received Flag turns ON again.

Before a receive request is executed, a check is performed to confirm that the required data is available.

● TCP Socket No. (1 to 3): Connection Status

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
m+4 to m+6	—	—	—	—	—	—	—	—	—	—	—	—	TCP connection status			

The connection status for each TCP socket is stored by code in this word. For details, refer to *A-6-1 TCP Status Transitions*.

● Socket Services Parameter Area 1 to 3

Offset	Socket No. 1	...	Socket No. 3	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
+0	m+8	...	m+28	Socket option	UDP/TCP socket number (1 to 3)
+1	m+9		m+29	Local UDP/TCP port number (0000 to FFFF Hex)	
+2	m+10		m+30	Remote IP address (00000000 to FFFFFFFF Hex)	
	m+11		m+31		
+4	m+12	...	m+32	Remote UDP/TCP port number (0000 to FFFF Hex)	
+5	m+13		m+33	Number of send/receive bytes (0000 to 0400 Hex (1024))	
+6	m+14		m+34	Send/receive data address (Same as FINS variable area designation method.)	
	m+15		m+35		
+8	m+16		m+36	Timeout value (0000 to FFFF Hex)	
+9	m+17	...	m+37	Response code	

When socket services are requested by control bit manipulation, the settings must be made in advance in a Socket Service Parameter Area. The parameters used will vary depending on the service requested.

Parameter Settings

The following table shows the parameters that are required for each service and the use of the parameters by the socket service.

● UDP Socket Services

Parameter	No. of words	Range (decimal values in parentheses)	Socket service			
			UDP open	UDP receive	UDP send	UDP close
Socket option	1	Specified bit	---	---	---	---
UDP/TCP socket No.		0001 to 0003 hexadecimal (1 to 3)	W	W	W	W
Local UDP/TCP port No.	1	0000 to FFFF hexadecimal (0 to 65,535)	W	---	---	---
Remote IP address	2	00000000 to FFFFFFFF hexadecimal (0.0.0.0 to 255.255.255.255)	---	R	W	---
Remote UDP/TCP port No.	1	0000 to FFFF hexadecimal (0 to 65,535)	---	R	W	---
Number of bytes to send/receive	1	0000 to 0400 hexadecimal (0 to 1,024 bytes)	---	RW	RW	---
Send/Receive data address	2	Memory area address	---	W	W	---
Time out time (Unit: 100 ms)	1	0000 to FFFF hexadecimal (0 to 65,535) (0: No limit, 0.1 to 6,553.5 s)	---	W	---	---
Response code	1	---	R	R	R	R

Note W: Written by user
RW: Written by user at execution and then read for results at completion
R: Read by user for results at completion
---: Not used.

● TCP Socket Services

Parameter	No. of words	Range (decimal values in parentheses)	Socket service				
			TCP passive open	TCP active open	TCP receive	TCP send	TCP close
Socket option	1	Specified bit	W	W	---	---	---
UDP/TCP socket No.		0001 to 0003 hexadecimal (1 to 3)	W	W	W	W	W
Local UDP/TCP port No.	1	0000 to FFFF hexadecimal (0 to 65,535)	W	RW	---	---	---
Remote IP address	2	00000000 to FFFFFFFF hexadecimal (0.0.0.0 to 255.255.255.255)	RW	W	---	---	---
Remote UDP/TCP port No.	1	0000 to FFFF hexadecimal (0 to 65,535)	RW	W	---	---	---
Number of bytes to send/ receive	1	0000 to 04D0 hexadecimal (0 to 1,024 bytes)	---	---	RW	RW	---
Send/Receive data address	2	Memory area address	---	---	W	W	---
Time out time (Unit: 100 ms)	1	0000 to FFFF hexadecimal (0 to 65,535) (0: No limit, 0.1 to 6,553.5 s)	W	---	W	---	---
Response code	1	---	R	R	R	R	R

Note W: Written by user
RW: Written by user at execution and then read for results at completion
R: Read by user for results at completion
n---: Not used.

Parameters

● Socket Option

For the TCP OPEN REQUEST (ACTIVE or PASSIVE) command, specifies whether or not the keep-alive function is to be used. When the keep-alive function is used, bit 8 is ON.

● UDP/TCP Socket No.

Specify the number of the UDP or TCP socket to open.

● Local UDP/TCP Port No.

Specify the number of the UDP or TCP port for the socket to use for communications.

- Do not specify the port being used as the FINS UDP port (default: 9600) in an open request for a UDP socket.
- Do not specify the port being used as the FINS TCP port (default: 9600) in an open request for a TCP (active or passive) socket.
- Do not specify auto connection UDP port number 9600 in an open request for a UDP socket.
- Do not specify auto connection TCP port number 9600 in an open request for a TCP (active or passive) socket.
- As a rule, use port numbers 1,024 and higher.

If port number 0 is specified when for an active TCP open, the TCP port number will be automatically allocated and the number of the port that was opened will be stored in the local UDP/TCP port number in the Socket Service Parameter Area (i.e., the actual port number will be overwritten on the value of 0 set by the user).

Remote IP Address

Specify the IP address of the remote device.

- Offset +2 in the Socket Service Parameter Area contains the upper bytes of the Remote IP Address, and offset +3 contains the lower bytes.
  Example: The contents of offsets +2 and +3 would be as shown below when the Remote IP Address is 196.36.32.55 (C4.24.20.37 hexadecimal).
  +2: C424
  +3: 2037
- This parameter is not used when making a receive request for a UDP socket. The remote IP address will be stored with the response data and will be written as the Remote IP Address in the Socket Service Parameter Area.
- When opening a passive TCP socket, the combination of the remote IP address and the remote TCP port number can be used to affect processing as shown in the following table.

Remote IP Address	Remote TCP Port No.	Processing
0	0	All connection requests accepted.
0	Not 0	Connection requests accepted only for the same port number.
Not 0	0	Connection requests accepted only for the same IP address.
Not 0	Not 0	Connection requests accepted only for the same port number and IP address.

If the Remote IP Address is set to 0, a connection can be made to any remote node and the remote IP address of the node that is connected will be stored as the Remote IP Address in the Socket Service Parameter Area. If a specific remote IP address is set, then a connection can be made only to the node with the specified address.

If the Remote TCP Port No. is set to 0, a connection can be made to any remote node regardless of the TCP port number it is using. If a specific remote TCP port number is set, then a connection can be made only to a node using the specified TCP port number.

Remote UDP/TCP Port No.

Specify the UDP or TCP port number used by the remote device.

- This parameter is not used when making a receive request for a UDP socket. The remote UDP/TCP port number will be stored with the response data and will be written as the Remote UDP/TCP Port No. in the Socket Service Parameter Area.
- When opening a passive TCP socket, the combination of the remote IP address and the remote TCP port number can be used to affect processing as shown in the table for the Remote IP Address, above. If the Remote UDP/TCP Port No. is set to 0, the UDP/TCP port number of the remote device will be written as the Remote UDP/TCP Port No. in the Socket Service Parameter Area.

Time Out Time

Set the time limit in units of 0.1 s for completion of communications from the time that the Receive Request Switch (TCP or UDP) or the TCP Passive Open Request Switch is turned ON. A response code of 0080 hexadecimal (timeout) will be stored if communications time out. If 0 is set, the requested service will not be timed.

Number of Bytes to Send/Receive

Send the number of bytes to be sent or the number of bytes to receive. When the transfer has been completed, the actual number of bytes that have been sent or received will be written here.

● Send/Receive Data Address

Specify the address of the first word to send or the address of the first word where data is to be received. Always set the bit number to 00 hexadecimal.

Offset	15 8	7 0
+6	Area designation	Leftmost 2 digits of word address
+7	Rightmost 2 digits of word address	Bit number (always 00 Hex)

The following specifications can be used.

Area		Word address	Area designation (hexadecimal)	Word address (hexadecimal)
CIO, HR, and AR Areas	CIO	0000 to 0289	B0	0000 to 0121
	HR	H000 to H127	B2	0000 to 007F
	AR	A448 to A959	B3	01C0 to 03BF
DM Area	DM	D00000 to D16383	82	0000 to 3FFF

● Response Codes

When processing of a request has been completed for socket services executed using Socket Service Request Switches, a response code will be stored in the Response Code word in the Socket Service Parameter Area. The following response codes will be stored depending on the service that was requested.

UDP Socket Open Request

Response code	Meaning
0000	Normal end
0105	Local IP address setting error.
1100	UDP socket number is not 1 to 8 or local UDP port number is 0.
110C	Request Switch turned ON during other processing.
220F	Specified socket is already open.
2211	Unit is busy; cannot execute.
2606	Specified socket is already open as TCP socket; cannot open UDPsocket.
2607	Specified Socket Service Parameter Area is already being used foranother socket.
0049	The same UDP port number has been specified more than once (EADDRINUSE).
0081	The specified socket was closed during open processing.

UDP Socket Receive Request

Response code	Meaning
0000	Normal end
0302	CPU Unit error; cannot execute.
1100	Number of bytes to receive is not in allowable range.
1101	The area designation of the Send/Receive Data Address is not inallowable range.
1103	The bit number in the Send/Receive Data Address is not 00.
110C	Request Switch turned ON during other processing.
220F	Specified socket is already processing a receive request.
2210	The specified socket is not open.
2211	Unit is busy; cannot execute service.
2607	Specified Socket Service Parameter Area is already being used foranother socket.
0080	Receive request timed out.
0081	The specified socket was closed during reception processing.

UDP Socket Send Request

Response code	Meaning
0000	Normal end
0302	CPU Unit error; cannot execute.
1100	Number of bytes to send is not in allowable range or the remote IPaddress is 0.
1101	The area designation of the Send/Receive Data Address is not inallowable range.
1103	The bit number in the Send/Receive Data Address is not 00.
110C	Request Switch turned ON during other processing.
220F	Specified socket is already processing a send request.
2210	The specified socket is not open.
2211	Unit is busy; cannot execute.
2607	Specified Socket Service Parameter Area is already being used foranother socket.
003E	Internal buffer cannot be obtained due to high reception traffic (ENOBUFS).
004C	The network ID is incorrect or the remote IP address is incorrect(EADDRNOTAVAIL)
004E	The network ID is not in the IP router table, router settings are incorrect, or the remote IP address is incorrect (ENETUNREACH).
0081	The specified socket was closed during send processing.

UDP Socket Close Request

Response code	Meaning
0000	Normal end
0302	CPU Unit error; cannot execute.
2210	The specified socket is not open.
2607	Specified Socket Service Parameter Area is already being used foranother socket.

TCP Socket Passive Open Request

Response code	Meaning
0000	Normal end
0105	Local IP address setting error.
1100	TCP socket number is not 1 to 8 or local TCP port number is 0.
110C	Request Switch turned ON during other processing.
220F	Specified socket is already open or already processing an openrequest.
2211	Unit is busy; cannot execute.
2606	Specified socket is already open as UDP socket; cannot open TCP socket.
2607	Specified Socket Service Parameter Area is already being used foranother socket.
0045	Error in communications with remote node (ECONNABORTED).
0049	The same TCP port number has been specified more than once(EADDRINUSE).
004B (See note.)	Error in communications with remote node (ECONNRESET).
0053	Error in communications with remote node (ETIMEDOUT) or remotenode does not exist.
0080	Open request timed out.
0081	The specified socket was closed during open processing.

Note These response codes will be returned only on large, multilevel networks.

TCP Socket Active Open Request

Response code	Meaning
0000	Normal end
0105	Local IP address setting error.
1100	TCP socket number is not 1 to 8 or local TCP port number is 0.
110C	Request Switch turned ON during other processing.
220F	Specified socket is already open or already processing an openrequest.
2211	Unit is busy; cannot execute.
2606	Specified socket is already open as UDP socket; cannot open TCP socket.
2607	Specified Socket Service Parameter Area is already being used foranother socket.
000D	Remote IP address parameter error (EACCES).
0045	Error in communications with remote node (ECONNABORTED).
0049	The same port number has been specified more than once (EAD-DRINUSE).
004B (See note.)	Error in communications with remote node (ECONNRESET).
004C	Remote IP address parameter error (EADDRNOTAVAIL).Wrong parameter designation.An attempt was made to set the local TCP port of the local node toActive Open.
0053	Communications error with remote node (ETIMEDOUT).No remote node.
0081	The specified socket was closed during open processing.

Note These response codes will be returned only on large, multilevel networks.

TCP Socket Receive Request

Response code	Meaning
0000	Normal end
0302	CPU Unit error; cannot execute.
1100	Number of receive bytes not in allowable range.
1101	The area designation of the Send/Receive Data Address is not inallowable range.
1103	The bit number in the Send/Receive Data Address is not 00.
110C	Request Switch turned ON during other processing.
220F	Specified socket is already processing a receive request.
2210	Specified socket has not been connected.
2211	Unit is busy; cannot execute.
2607	Specified Socket Service Parameter Area is already being used foranother socket.
0045 (See note.)	Error in communications with remote node (ECONNABORTED).
0053	Error in communications with remote host (ETIMEDOUT).
0080	Receive request timed out.
0081	The specified socket was closed during receive processing.

Note These response codes will be returned only on large, multilevel networks.

TCP Socket Send Request

Response code	Meaning
0000	Normal end
0302	CPU Unit error; cannot execute.
1100	Number of bytes to send not in allowable range.
1101	The area designation of the Send/Receive Data Address is not inallowable range.
1103	The bit number in the Send/Receive Data Address is not 00.
110C	Request Switch turned ON during other processing.
220F	Specified socket is already processing a send request.
2210	The specified socket is not been connected.
2211	Unit is busy; cannot execute.
2607	Specified Socket Service Parameter Area is already being used foranother socket.
003E	Internal buffer cannot be obtained due to high reception traffic(ENOBUFS).
0045 (See note.)	Error in communications with remote node (ECONNABORTED).
004E (See note.)	Remote IP address parameter error (ENETUNREACH).
0081	The specified socket was closed during send processing.

TCP Socket Close Request

Responsecode	Meaning
0000	Normal end
0302	CPU Unit error; cannot execute.
2210	The specified socket is not been connected.
2607	Specified Socket Service Parameter Area is already being used foranother socket.

Note These response codes will be returned only on large, multilevel networks.

For details, refer to *SECTION 6* Socket Services in the *Ethernet Units Construction of Applications Operation Manual (Cat. No. W421)*.

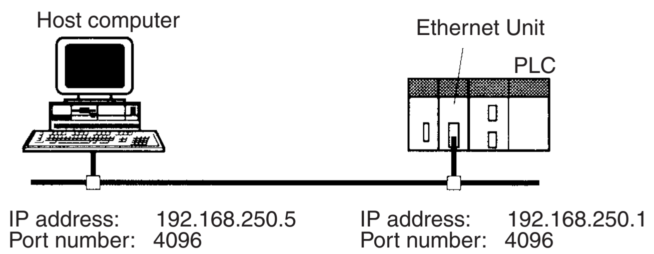
15-5-7 Socket/TCP Programming Example

TCP/IP Communications Programming Example

The following programming example illustrates transferring 100 bytes of databetween an Ethernet Unit and a host computer using TCP/IP communications.

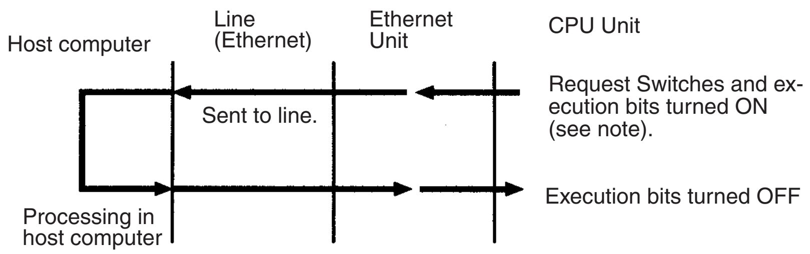
● System Configuration

The programming example uses the following system configuration. For the TCP connection, the Ethernet Unit uses a passive open and the host computer uses an active open.



● Data Flow

The data will flow between the CPU Unit, Ethernet Unit, and host computer as shown in the following diagram.



Note Here, “execution bits” refer to W0.00 to W0.03, which are used in the ladder diagram to control execution of communications.

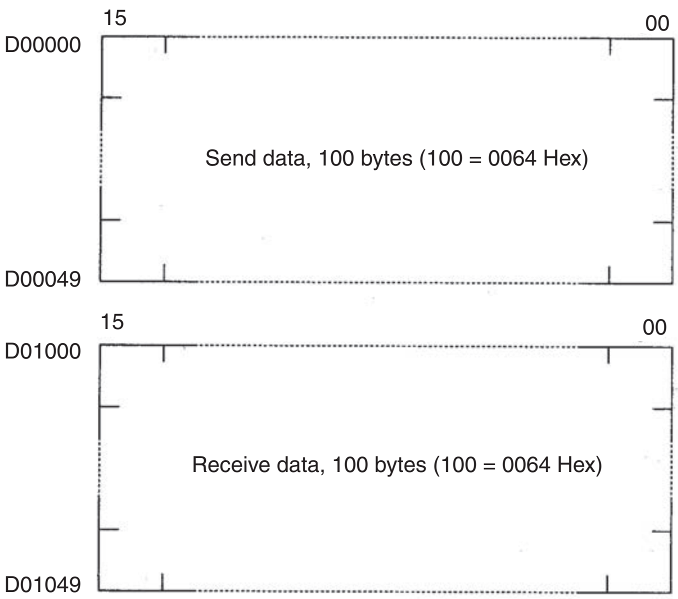
Basic Operations

- W0.00 is turned ON to request opening a TCP socket from the Ethernet Unit.
- W0.01 is turned ON to request closing the TCP socket from the Ethernet Unit.
- W0.02 is turned ON to request sending data from the Ethernet Unit. Data (100 bytes) is sent beginning at D00000.
- W0.03 is turned ON to request receiving data from the Ethernet Unit. The data that is received (100 bytes) is stored beginning at D01000.
- One of the bits between W1.00 and W1.03 will turn ON if an error occurs. Refer to 15-7-5 Socket Service Request Switches for information on errors.

Program Memory Map

The send and receive data and bits (flags) used by the program are shown in the following diagram.

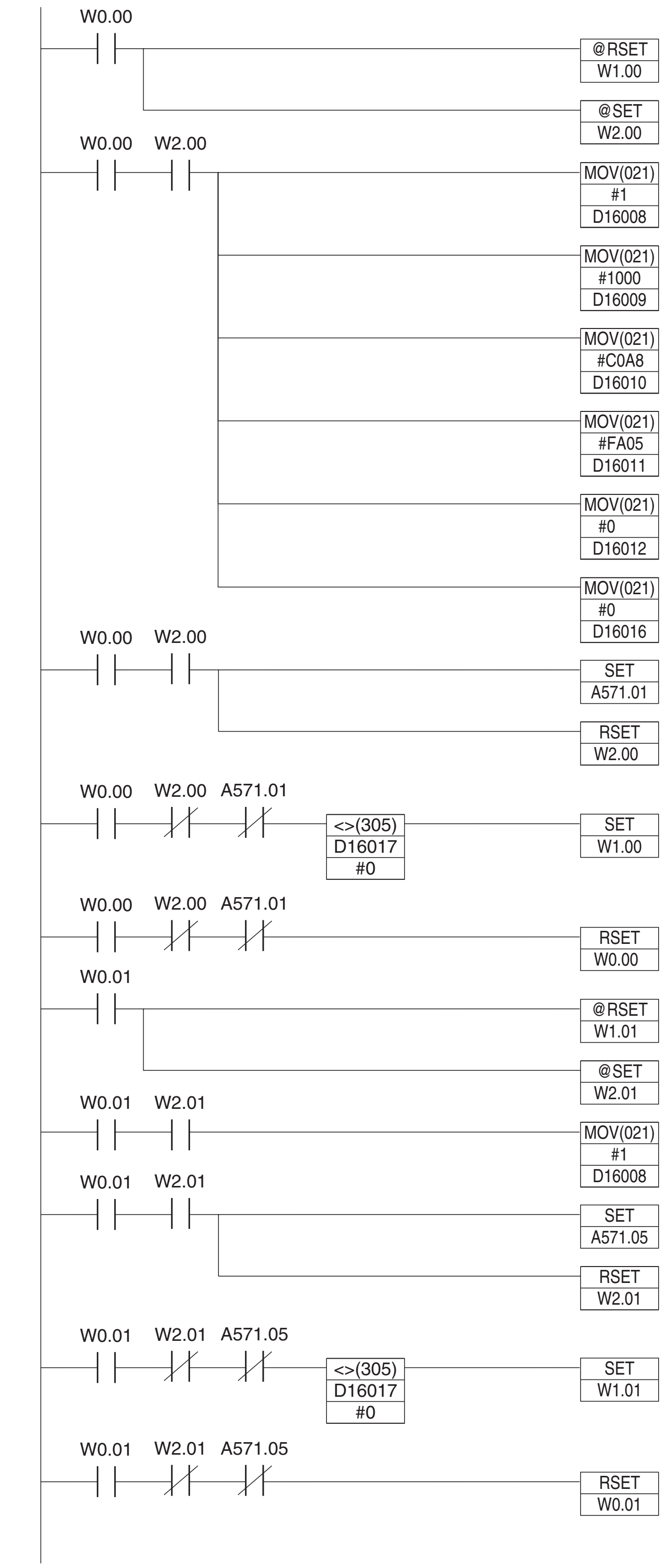
DM Area



WR Area

	15		03	02	01	00
W0			TCP Receive Bit	TCP Send Bit	TCP Close Bit	TCP Open Bit
W1			TCP Receive Error Flag	TCP Send Error Flag	TCP Close Error Flag	TCP Open Error Flag
W2			TCP Receiving Flag	TCP Sending Flag	TCP Closing Flag	TCP Opening Flag

Programming Example



TCP Passive Open

When the TCP Open Bit (W0.00) turns ON, the TCP Open Error Flag (W1.00) is turned OFF and the TCP Opening Flag (W2.00) is turned ON to initialize processing.

When the TCP Opening Flag (W2.00) turns ON, the following parameters are written to the parameter area for socket number 1.
D16008: 0001 Hex = UDP/TCP socket No. 1
D16009: 1000 Hex = Local UDP/TCP port No. 4096
D16010 and D16011:
C0A8 FA05 Hex =
Remote IP address 192.168.250.5
D16012: 0000 Hex = Any remote UDP/TCP port No.
D16016: 0000 Hex = No timeout time

After the parameters have been set, the TCP Passive Open Request Switch (A571.01) is turned ON and the TCP Opening Flag (W2.00) is turned OFF.

If the TCP Passive Open Request Switch (A571.01) turns OFF while the TCP Opening Flag (W2.00) is OFF, the contents of the response code (D16017) in the Socket Service Parameter Area is checked, and if it is not 0000 Hex (normal end), the TCP Open Error Flag (W1.00) is turned ON.

After the execution results have been checked, the TCP Open Bit (W0.00) is turned OFF.

TCP Close

When the TCP Close Bit (W0.01) turns ON, the TCP Close Error Flag (W1.01) is turned OFF and the TCP Closing Flag (W2.01) is turned ON to initialize processing.

When the TCP Closing Flag (W2.01) turns ON, the following parameter is written to the parameter area for socket number 1.
D16008: 0001 Hex = UDP/TCP socket No. 1

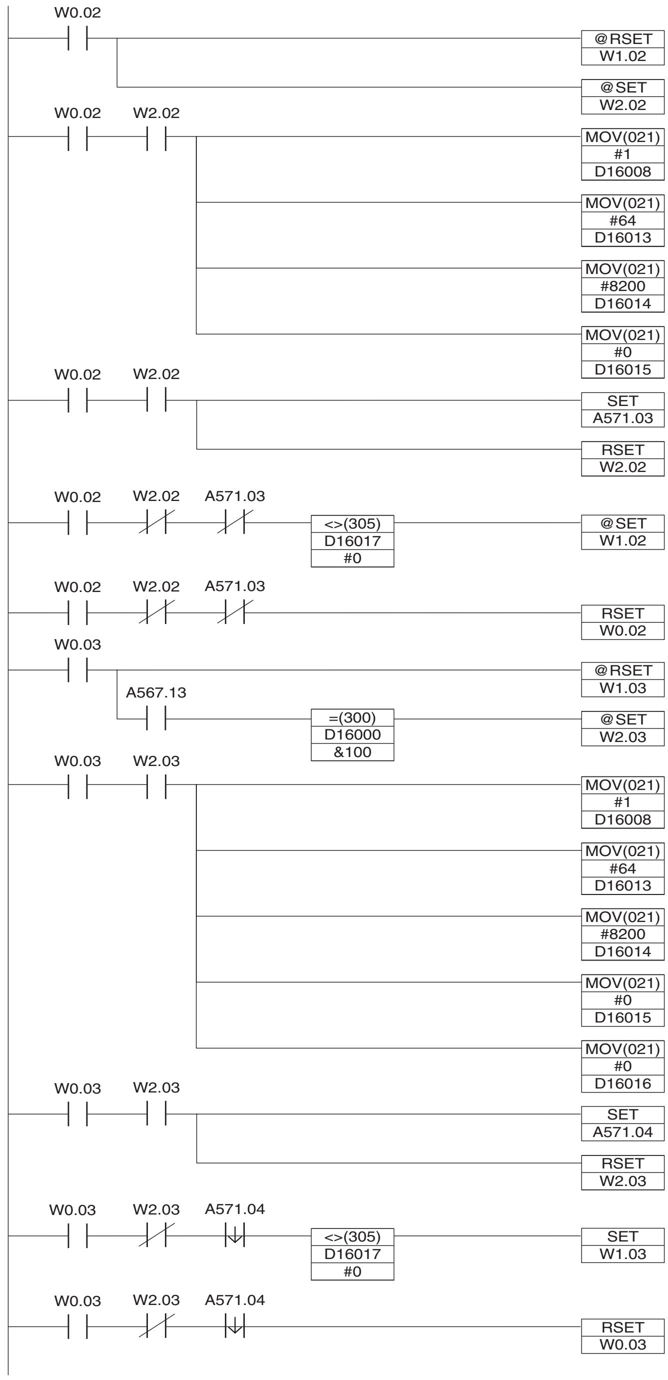
After the parameter has been set, the Close Request Switch (A571.05) is turned ON and the TCP Closing Flag (W2.01) is turned OFF.

If the Close Request Switch (A571.05) turns OFF while the TCP Opening Flag (W2.01) is OFF, the contents of the response code (D16017) in the Socket Service Parameter Area is checked, and if it is not 0000 Hex (normal end), the TCP Close Error Flag (W1.01) is turned ON.

After the execution results have been checked, the TCP Close Bit (W0.01) is turned OFF.

Continued on next page.

Continued from previous page.



TCP Send

When the TCP Send Bit (W0.02) turns ON, the TCP Send Error Flag (W1.02) is turned OFF and the TCP Sending Flag (W2.02) is turned ON to initialize processing.

When the TCP Sending Flag (W2.02) turns ON, the following parameters are written to the parameter area for socket number 1
D16008: 0001 Hex = UDP/TCP socket No. 1
D16013: 0064 Hex = No. of send/receive bytes is 100
D16014 and D16015:
8200 0000 Hex =
Send/receive data address D00000

After the parameters have been set, the Send Request Switch (A571.03) is turned ON and the TCP Sending Flag (W2.02) is turned OFF.

If the Send Request Switch (A571.03) turns OFF while the TCP Sending Flag (W2.02) is OFF, the contents of the response code (D16017) in the Socket Service Pa-rameter Area is checked, and if it is not 0000 Hex (normal end), the TCP Send Error Flag (W1.02) is turned ON.

After the execution results have been checked, the TCP Send Bit (W0.02) is turned OFF.

TCP Receive

When the TCP Receive Bit (W0.03) turns ON, the TCP Receive Error Flag (W1.03) is turned OFF and the TCP Data Received/Requested Flag (A567.13), and the Number of Bytes Received at TCP Socket (D16000) are checked. If the data is stored in the buffer, the TCP Receiving Flag (W2.03) turns ON.When the TCP Receiving Flag (W2.03) turns ON, the following parameters are written to the parameter area for socket number 1.
D16008: 0001 Hex = UDP/TCP socket No. 1
D16013: 0064 Hex = No. of send/receive bytes is 100
D16014 and D16015:
8203 E800 Hex =
Send/receive data address D01000
D16016: 0000 Hex = No timeout time.

After the parameter has been set, the Receive Request Switch (A571.04) is turned ON and the TCP Receiving Flag (W2.03) is turned OFF.

If the Receive Request Switch (A571.04) turns OFF while the TCP Receiving Flag (W2.03) is OFF, the contents of the response code (D16017) in the Socket Service Parameter Area is checked, and if it is not 0000 Hex (normal end), the TCP Receive Error Flag (W1.03) is turned ON.

After the execution results have been checked, the TCP Receive Bit (W0.03) is turned OFF.

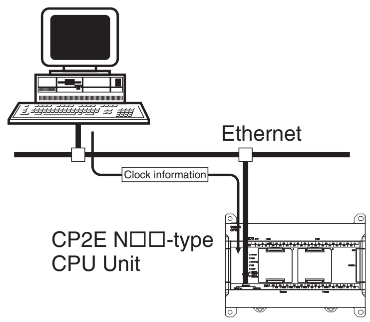
Note When using the above programming example, change the bit and word addresses as necessary to avoid using the same areas used by other parts of the user program or the CPU Bus Unit.

15-6 Automatic Clock Adjustment and Specifying Servers by Host Name

15-6-1 Automatic Clock Adjustment Function

The built-in clock of the PLC connected to the Ethernet can be automatically adjusted, with the SNTP server clock taken as the standard. Automatic adjustments through the entire system enable the various records generated by production equipment to be managed according to clock information and analyzed.

The PLC can acquire clock information from the SNTP server at a particular time or when a dedicated bit turns ON, and it can refresh the internal clock information automatically.



Note 1 An SNTP server is required to use this function.

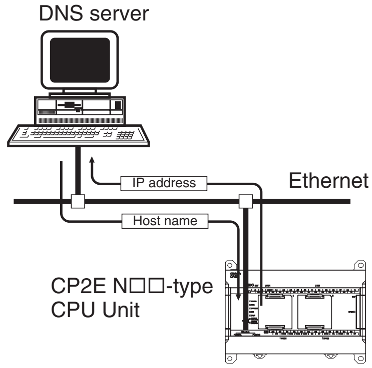
2 SNTP server settings require specialized knowledge, so they should always be handled by the network administrator.

3 When using the Internet, depending on the condition of the network it may not be possible to acquire the clock information.

15-6-2 Specifying Servers by Host Name

SNTP servers can be specified by host name rather than by IP address by using of the DNS client function.

This enables automatic searches for IP addresses for purposes such as system checking, even when the IP addresses for servers have been changed.



Note 1 A DNS server is required to specify servers by IP address.

2 The IP address is specified directly for the DNS server.

15-6-3 Procedure for Using the Automatic Clock Adjustment Function

1. Make the basic settings.
   Refer to *15-3-3 Basic Settings*.

⇩

2. With the CX-Programmer online, set the following items in the PLC Setup.
   - SNTP server specification (required)
   - Access to the SNTP server is enabled when writing clock information from the SNTP server to the CPU Unit when the Automatic Clock Adjustment Switch is turned from OFF to ON and at a set automatic adjustment time.
   - Automatic clock adjustment setting

⇩

3. To perform automatic clock adjustment manually, turn the Automatic Clock Adjustment Switch (A566.4) from OFF to ON.

⇩

4. Select **Transfer to PLC** from the Options Menu and click the **Yes** button.
   The PLC Setup will be transferred to the CPU Unit.

15-6-4 PLC Setup for DNS and Automatic clock Adjustment

DNS and Automatic Clock Adjustment

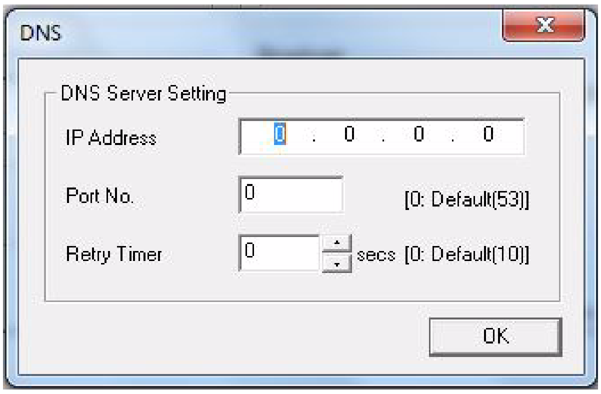
All these settings are in the Built-in Ethernet Tab. Click on the relative button can open the setup dialog.

Button name	Settings
DNS Setting	IP Address
	Port No.
	Retry Timer
Clock Auto Adjustment	SNTP Server Setting
	Auto Adjustment
	Designation Method
	IP Address
	Host Name
	Port No.
	Retry Timer
	Time Lag Adjustment

CX-Programmer Setup

● DNS Setting

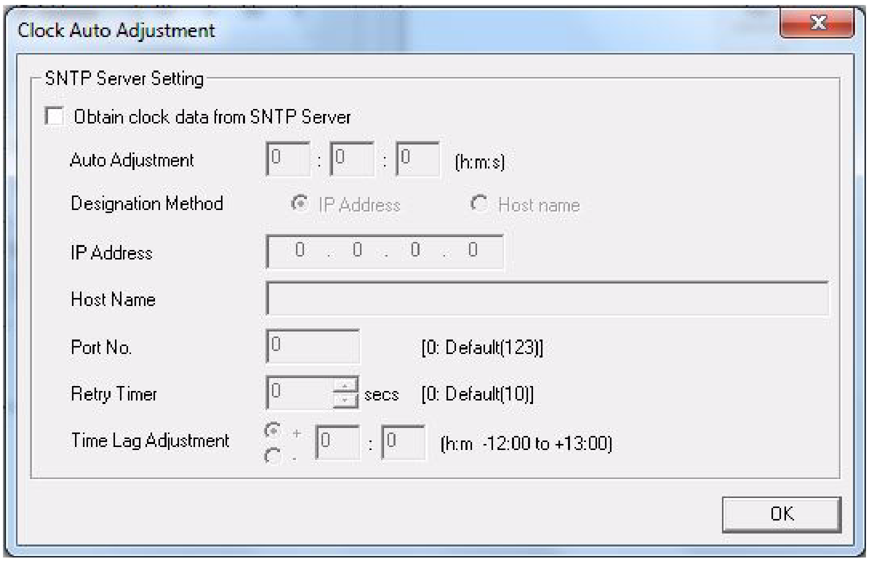
Move the cursor to the **Settings** and double click. Select the Built-in Ethernet Tab. Click the **DNS Setting** button to display the DNS setup dialog.



Item	Contents	Default
IP Address	Set the IP address for the DNS server. The DNS server is required when specifying the SNTP servers by host name.	None
Port No.	Set the port to be used for connecting to the DNS server. This setting does not normally need to be changed.	0 (Number 53 is used.)
Retry timer	Set the time to elapse before retrying when a connection to the DNS server fails. This setting does not normally need to be changed.	0 (10 s)

● Clock Auto Adjustment

Move the cursor to the **Settings** and double click. Select the Built-in Ethernet Tab. Click the **Clock Auto Adjustment** button to display the Clock Auto Adjustment setup dialog.



Item	Contents	Default
Obtain clock data from SNTP server	If this option is selected, the CPU Unit's clock is set to the time at the SNTP server's clock.	Not checked
Auto Adjust-ment	Set the time at which the SNTP server is to be accessed to syn-chronize the clocks. When the time that is set here arrives, the SNTP server is accessed and the CPU Unit clock is adjusted to match the SNTP server clock. The clock data from SNTP server is UTC (Universal Time Coordi-nated), so it needs to be adjusted to the local time.	0:0:0
Designation Method	Select whether the SNTP server used for automatic clock adjust-ment is to be specified by IP address or by host domain name (i.e., by host name).	IP Address
IP Address	Set the IP address for the SNTP server that is to be used for auto-matic clock adjustment. This setting is enabled only when server specification by IP address has been selected.	0.0.0.0
Host Name	Set the host domain name (i.e., the host name) for the SNTP server that is to be used for automatic clock adjustment. This setting is enabled only when server specification by host name has been selected.	None
Port No.	Set the port number for connecting to the SNTP server that is to be used for automatic clock adjustment. This setting does not normally need to be changed.	0 (Number 123 is used.)
Retry Timer	Retry within the setting time when a connection to the SNTP server fails. An Ethernet server connection error will occur at the end of the time.	0 (10 s)
Time Lag Adjustment	This sets in the CPU Unit's clock data the time difference made up from the SNTP server's clock data. To use the clock data from the SNTP server just as it is, input 0.	+0:0

15-6-5 Auxiliary Area Allocations

The following table and descriptions cover the words and bits in the Auxiliary Area of PLC memory that are related to the Automatic Clock Adjustment and Specifying Servers by Host Name function.

Service Status

Address	Bit(s)	Name	Status	Unit operation	Access
A46	5	DNS Server Error	ON	ON when the following errors occur during DNS server operation: • An illegal server IP address is set. • A timeout occurs during communications with the server.	Read only
			OFF	OFF when DNS server operation is normal.	
	11	SNTP Server Error	ON	ON when the following errors occur during SNTP server operation: • An illegal server IP address is set. • A timeout occurs during communications with the server.	
			OFF	OFF when SNTP server operation is normal.	

Service Request

Address	Bit(s)	Name	Status	Unit operation	Access
A566	4	Automatic Clock Adjustment Switch	ON	The automatic clock adjustment is executed when this bit turns ON.	Read/Write
			OFF	Turned OFF by Unit after automatic clock adjustment has been completed.	

- **Automatic Clock Adjustment Switch (Bit 4)**

The automatic clock adjustment can be executed by turning this switch ON.

The SNTP server required for the automatic clock adjustment is set in the PLC Setup.

After the automatic clock adjustment has been completed, the Unit will automatically turn this switch OFF. Until then, do not forcibly manipulate the switch.

15-7 Status Allocations of Bulit-in Ethernet Port

The following table and descriptions cover the words and bits in the Auxiliary Area of PLC memory that are related to the status of built-in Ethernet port.

Address	Bit(s)	Name	Status	Unit operation	Access
A40 to A44	---	Ethernet Communication Error Information	---	When an error occurs in Ethernet communication, the error information can be stored. Refer to *6-1-11 Other Errors* in the *CP2E CPU Unit Hardware User's Manual* (Cat. No. W613).	Read only
A45	14	Link Status (N20: Ethernet port, N30/40/60: PORT1A)	ON	Ethernet link is established.	Read only
			OFF	Ethernet link is terminated.	
	15	Link Status (N30/40/60: PORT1B)	ON	Ethernet link is established.	
			OFF	Ethernet link is terminated.	
A46	2	IP Address Setting Error Flag	ON	ON if any of the following conditions apply to the IP address. • All bits in the host ID are 0 or 1. • All bits in the network ID are 0 or 1. • All bits in the subnet ID are 1. • The IP address begins with 127 (0x7F)	Read only
			OFF	OFF when the IP address is normal.	
	3	IP Address Table Error Flag	ON	ON if the IP address table information is incorrect.	
			OFF	OFF when the IP address table is normal.	
	4	IP Router Table Error Flag	ON	ON if the IP router table information is incorrect.	
			OFF	OFF when the IP address table is normal.	
	5	DNS Server Error Flag	ON	ON when the following errors occur during DNS server operation: • An illegal server IP address is set. • A timeout occurs during communications with the server.	
			OFF	OFF when DNS server operation is normal.	
	6	Routing Table Error Flag	ON	ON if the routing table information is incorrect.	
			OFF	OFF when the routing table is normal.	
	11	SNTP Server Error	ON	ON when the following errors occur during SNTP server operation: • An illegal server IP address or host name is set. • A timeout occurs during communications with the server.	
			OFF	OFF when SNTP server operation is normal.	
	14	Address Disagreement Flag	ON	ON if the remote IP address is set to automatic generation but the local IP address host number and FINS node address do not agree	
			OFF	OFF under all other circumstances	
	15	Ethernet Communication Error Flag	ON	ON when Ethernet communication error occurs.	
			OFF	OFF when Ethernet communication error has been cleared by Ethernet communication error clear flag.	

Address	Bit(s)	Name	Status	Unit operation	Access
A47	0	FINS/TCP Connection Flag 1	ON	Turned ON by the Unit when a connection is established.	Read only
			OFF	Turned OFF by the Unit when the connection is terminated.	
	1	FINS/TCP Connection Flag 2	ON	Turned ON by the Unit when a connection is established.	
			OFF	Turned OFF by the Unit when the connection is terminated.	
	2	FINS/TCP Connection Flag 3	ON	Turned ON by the Unit when a connection is established.	
			OFF	Turned OFF by the Unit when the connection is terminated.	
A500	11	Ethernet Communication Error Clear Flag	ON	If this flag is set to ON, Ethernet communication error flag (A46.15) will be cleared (OFF).	Read/write
			OFF	OFF after Ethernet communication error has been cleared.	

16

Other Functions

This section describes PID temperature control, clock functions, DM backup functions, security functions.

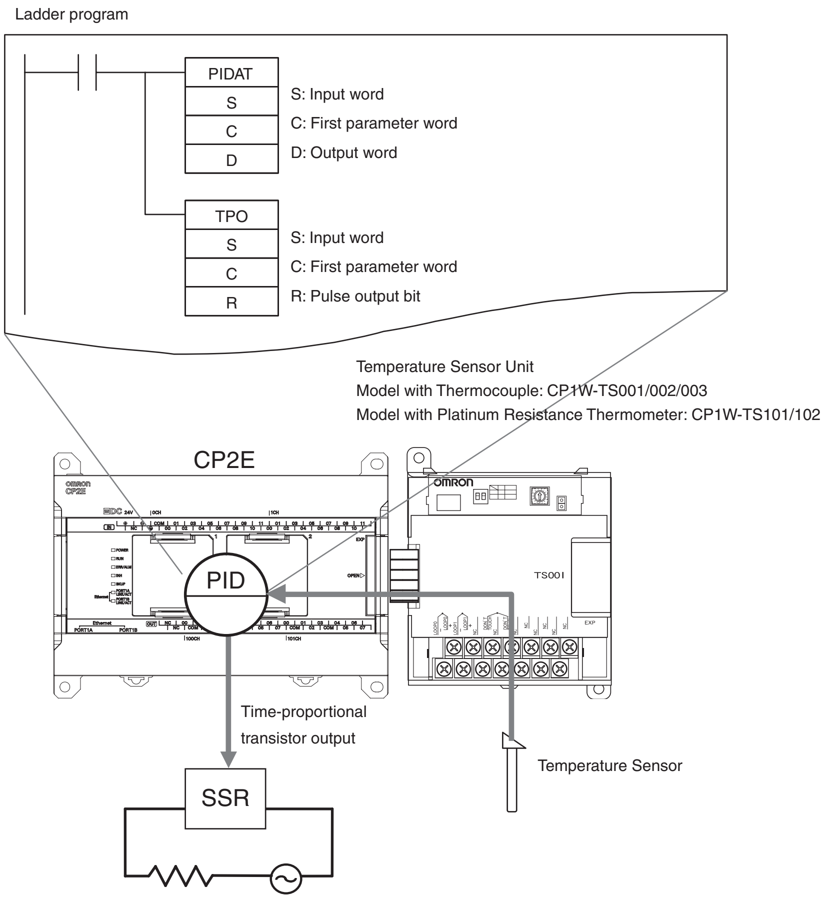
16-1 PID Temperature Control

PID temperature control can be used with any model of CP2E CPU Unit.

16-1-1 Overview

The CP2E CPU Unit supports PID instructions with the autotuning function. Ladder programs can be written to perform PID temperature control.

- Temperature input: Input from Temperature Sensor Unit to words in the Input Area.
- PID control: Execute using the PIDAT instruction in ladder program.
  The PIDAT instruction is used in combination with the TPO instruction (TIME-PROPORTIONAL OUTPUT) to perform time-proportional control.
- Control output: To connect an SSR, connect a 24-V power supply to the transistor output and output voltage pulses.



Additional Information

The sampling cycle set for a PIDAT instruction is between 10 ms to 99.99 s in increments of 10 ms. The actual calculation cycle is determined by the relationship with cycle time. Refer to the *CP1E/CP2E CPU Unit Instructions Reference Manual* (Cat. No. W483) for the PIDAT instruction.

16-1-2 Flow of Operation

Step	Operation	Description
1	Setting the Temperature Sensor Unit	Set the temperature range with the rotary switch on the front panel.
2	Wiring I/O	• Connect the temperature sensor to the Temperature Sensor Unit. • Connect the SSR to the transistor output.
3	Setting PIDAT and TPO instructions parameters	Set parameters with the MOV instruction or other instructions.
4	Executing the PIDAT instruction	Execute the PIDAT instruction.
5	Autotuning	Execute autotuning for the PID constants.
6	Starting PID control	Start PID control.

Inputting the Temperature Sensor's PV to PIDAT Instructions

● Temperature Sensor Unit

- Setting the Temperature Range
  Set the temperature range with the rotary switch on the front panel of the Temperature Sensor Unit. If the rotary switch is set to 1 for a CP1W-TS001 Temperature Sensor Unit, the temperature range is 0.0 to 500.0°C.
- Temperature Data Storage Format
  Temperature data is automatically stored in words in the Input Area allocated to the Temperature Sensor Unit as an Expansion Unit using four-digit hexadecimal.
  Example: 100°C is stored as 0064 hex.
  - When the range code is a decimal number to one decimal point, the value is multiplied by a factor of 10 and converted to a hexadecimal number without a sign, then stored as binary data.

    Example: 500.0°C multiplied by 10 is 5000 decimal. This is converted to 1388 in hexadecimal and stored.
  - If the temperature is negative, it is stored as signed hexadecimal.

    Example: -200°C is stored as FF38 hex.

● PIDAT Instruction

The PIDAT instruction treats the PV as unsigned hexadecimal data (0000 to FFFF hex). Signed data cannot be used, so if the temperature range includes negative values, apply scaling with the APR instruction.

Autotuning Procedure

● Automatically Executing Autotuning When PIDAT Is Executed

To automatically autotune the PID constants, turn ON the AT Command Bit when the PIDAT instruction is executed.

1 Set the PID parameter in words C to C+10. Word C is specified by the second operand.

Example: Place the set value (SV) in C and place the input range in bits 08 to 11 of C+6. Turn ON bit 15 of C+9 (AT Command Bit).

2 Turn ON the PIDAT instruction's input condition.

The PIDAT instruction will execute autotuning. When it has finished, the AT Command Bit (bit 15 in C+9) will turn OFF. At the same time the proportional band (C+1), integral constant (C+2), and derivative constant (C+3) calculated by autotuning will be stored and PID control will be started.

● Executing Autotuning for Other Conditions When PIDAT Is Executed

Here, the AT Command Bit is left OFF when the PIDAT instruction is being executed. Later it is turned ON by some other condition to start autotuning.

1 Set the PID parameter in words C to C+10. Word C is specified by the second operand.

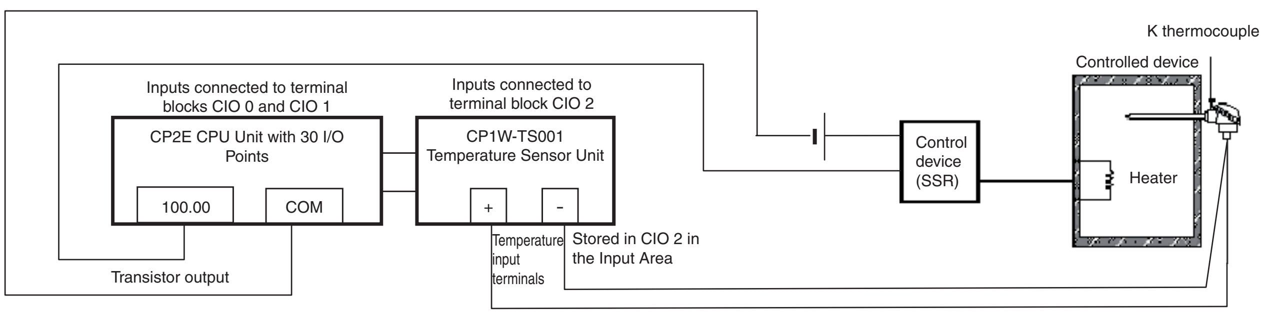
Example: Place the set value (SV) in C, the proportional band in C+1, the integral constant in C+2, the derivative constant in C+3, and the input range in bits 08 to 11 of C+6. Turn OFF bit 15 of C+9 (AT Command Bit).

2 Turn ON the PIDAT instruction's input condition. PID control will be started with the specified PID constants.

3 Turn ON bit 15 in C+9 (the AT Command Bit) while the input condition for the PID instruction is ON. Autotuning will be performed. When it has finished, the AT Command Bit (bit 15 in C+9) will turn OFF. The proportional band (C+1), integral constant (C+2), and derivative constant (C+3) calculated by autotuning will be stored and PID control will be started with those PID constants.

16-1-3 Application Example

System Configuration



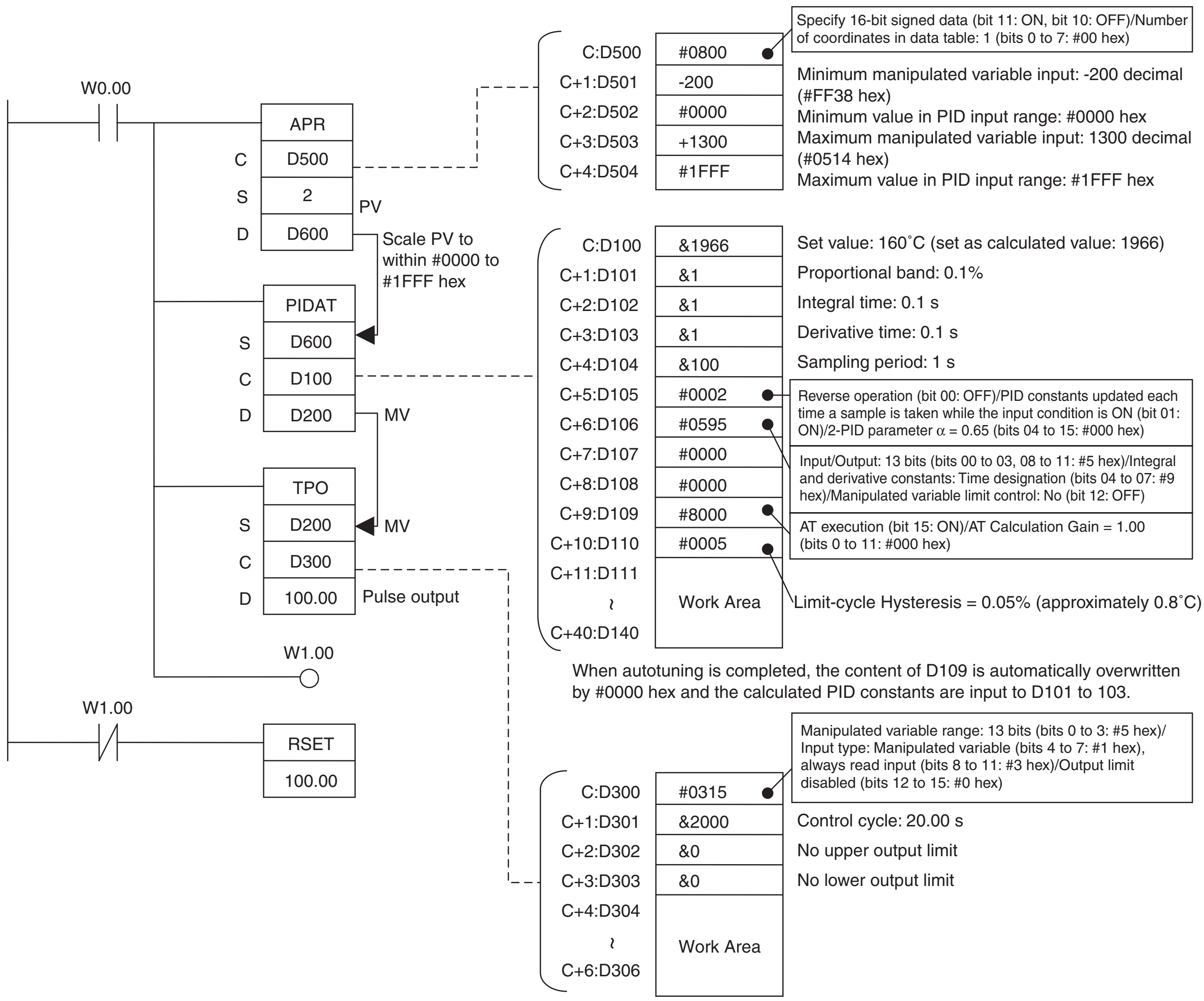
- A K thermocouple is used for the temperature input. Use a CP1W-TS001 Temperature Sensor Unit (thermocouple input).

- The Temperature Sensor Unit's temperature input PV is stored in CIO 2.
- The control output is the transistor output used to control the heater through the SSR using time-proportional control.
- The PIDAT sampling cycle is 1 second.
- Control cycle: 20 s
- When W0.00 turns ON, autotuning is immediately executed and PID control is started with the PID constants calculated by autotuning.

Ladder Programming Example for an Input Range of -200 to 1300°C for a K Thermocouple

The CP1W-TS001 Temperature Sensor Unit is used with an input type of K -200 to 1300°C (set the rotary switch to 0). The decimal values -200 to 1300°C are converted to signed hexadecimal data (FF38 to 0514 hex) and stored in CIO 2 in the Input Area.

However, the PIDAT instruction can only handle unsigned hexadecimal data as the PV. The value is thus converted from the range FF38 to 0514 to the PIDAT instruction input range of 0000 to 1FFF hex (0 to 8191) using the APR instruction.



Description

- When W0.00 turns ON, the work area in D111 to D140 is initialized (cleared) according to the parameters set in D100 to D110. After the work area has been initialized, autotuning is started and the PID constants are calculated from the results from changing the manipulated variable. After autotuning has been completed, PID control is executed according to the calculated PID constants set in D101 to D103. The manipulated variable is output to D200. The manipulated variable in D200 is divided by the manipulated variable range using the TPO instruction. This value is treated as the duty factor which is converted to a time-proportional output and output to CIO100.00 as a pulse output.
- When W0.00 turns OFF, PID is stopped and CIO100.00 turns OFF.
- When W0.00 is ON, the Thermocouple's PV (-200 to 1300) is scaled to the PIDAT instruction input range (#0 to #1FFF hex). The set values must be input according to the scaled PV. For example, if the PV is 160°C, it is set as [8191/(1300+200)] × (160+200) = 1966].

16-2 Clock

The clock can be used only with the CP2E N/S□□-type CPU Unit.

The current data is stored in the following words in the Auxiliary Area.

Name	Address	Function
Clock data	A351 to A354	The seconds, minutes, hour, day of month, month, year, and day of week are stored each cycle.
	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

Additional Information

The clock cannot be used if a battery is not installed or the battery voltage is low.

When the clock stops, the clock data will be reset to 2001-01-01 01:01:01 Sunday.

● Related Auxiliary Area Bits and Words

Name	Address	Contents
Start-up Time	A510 and A511	The time at which the power was turned ON (day of month, hour, minutes, and seconds).
Power Interruption Time	A512 and A513	The time at which the power was last interrupted (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).

● Time-related Instructions

Name	Mnemonic	Function
CALENDAR ADD	CADD	Adds time to the calendar data in the specified words.
CALENDAR SUBTRACT	CSUB	Subtracts time from the calendar data in the specified words.
CLOCK ADJUSTMENT	DATE	Changes the internal clock setting to the setting in the specified source words.

Precautions for Correct Use

The clock may stop if a battery is not installed or the battery voltage is low. If the clock stops, it cannot be set during 3 seconds after the power is ON. Please set the clock 3 seconds after the power is ON.

If set the clock during 3 seconds after the power is ON, the following error response will occur.

- DATE instruction: The clock cannot be set. Error flag (P_ER) is ON.
- When the clock is set with a FINS command (command code 0702), the response end code will be 2108 (Cannot be set during clock initialization).

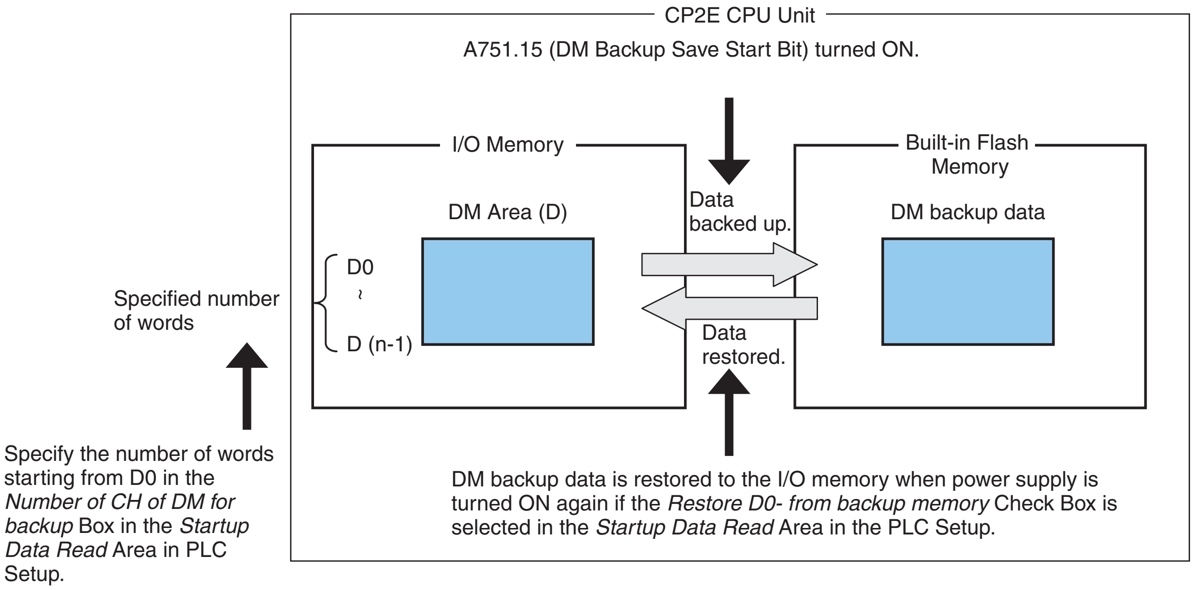
16-3 DM Backup Function

This section describes the function that saves specified words from the DM Area in the built-in Flash Memory.

16-3-1 Backing Up and Restoring DM Area Data

Overview

The contents of the DM Area (D) can be saved when the user needs it. The contents of the specified words in the DM Area data can be backed up from I/O memory to the built-in Flash Memory during operation by turning ON a bit in the Auxiliary Area. The number of DM Area words to back up is specified in the *Number of CH of DM for backup* Box in the PLC Setup. If the *Restore D0- from backup memory* Check Box is selected in the PLC Setup, the backup data will automatically be restored to I/O memory when the power is turned back ON so that data is not lost even if power is interrupted.



● Conditions for Executing Backup

Specified words starting from D0 in the I/O memory can be saved to the built-in Flash Memory by turning ON A751.15. (These words are called the DM backup words and the data is called the DM backup data.) A751.15 (DM Backup Save Start Bit) can be used in any operating mode (RUN, MONITOR, or PROGRAM mode).

● Words that can be Backed Up

- E□□-type CPU Units: D0 to D1499
- S□□-type CPU Units: D0 to D6999
- N□□-type CPU Units: D0 to D14999

Number of Words to Back Up

The number of words to back up starting from D0 is set in the *Number of CH of DM for backup Box in the Startup Data Read* Area in the *PLC Setup*.

Restoring DM Backup Data to the I/O Memory When Power is Turned ON

The DM backup data can be restored to the I/O memory when power is turned ON by selecting the *Restore D0- from backup memory* Check Box in the *Startup Data Read* Area in the PLC Setup.

The DM backup data will be read from the backup memory even if the *Clear retained memory area (HR/DM/CNT)* Check Box is selected in the PLC Setup.

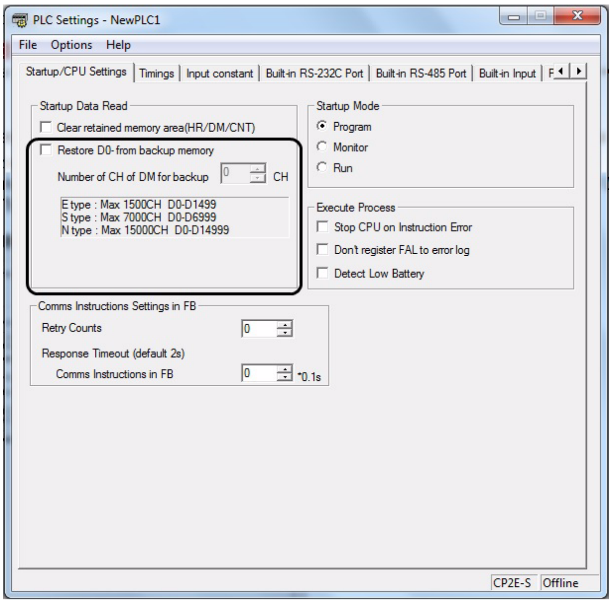
Related Auxiliary Area Bits

Name	Address	Description
DM Backup Save Start Bit	A751.15	The number of words in the DM Area specified in the *Number of CH of DM for backup* Box in the *Startup Data Read* Area in the PLC Setup are saved from the I/O memory to the built-in Flash Memory when this bit is turned ON. This bit will not automatically turn OFF again if the bit turns ON. Design the ladder program so that this bit is turned ON and OFF again using upwardly differentiated bits. If this bit is turned ON and OFF while the DM Backup Save Flag (A751.14) is ON, it will be ignored and the data will not be backed up again. To backup the data again, make sure that A751.14 is OFF and then turn ON A751.15. A751.15 is turned OFF when the power supply is turned ON.
DM Backup Save Flag	A751.14	This flag turns ON when A751.15 is turned ON to start the saving operation. This flag stays ON while data is being saved and turns OFF when finished. Use this flag to confirm when the DM backup operation has been completed. The flag is turned OFF when the power supply is turned ON.
DM Backup Restore Failed Flag	A751.11	This flag turns ON if the DM backup data could not be restored correctly. If this flag turns ON, data will not be read from the built-in Flash Memory to the I/O memory. For example, if power was interrupted while data was being backed up, the DM Area data would not be backed up properly and the next time power is turned ON, the DM backup data will not be restored. If this happens, this flag will be turned ON. If the number of the backed up DM area words is different from the *Number of CH of DM for backup* in the PLC Setup, this flag will be turned ON. This flag turns OFF in the following cases: • Data is successfully restored from the built-in Flash Memory to the I/O memory when the power supply is turned ON. • All memory is cleared.

16-3-2 Procedure

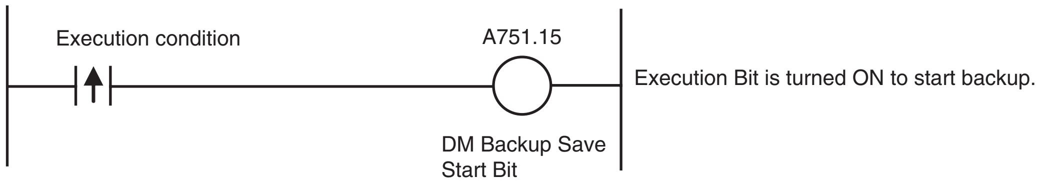
Perform the following procedure to save the DM data to the built-in Flash Memory during operation or while stopped.

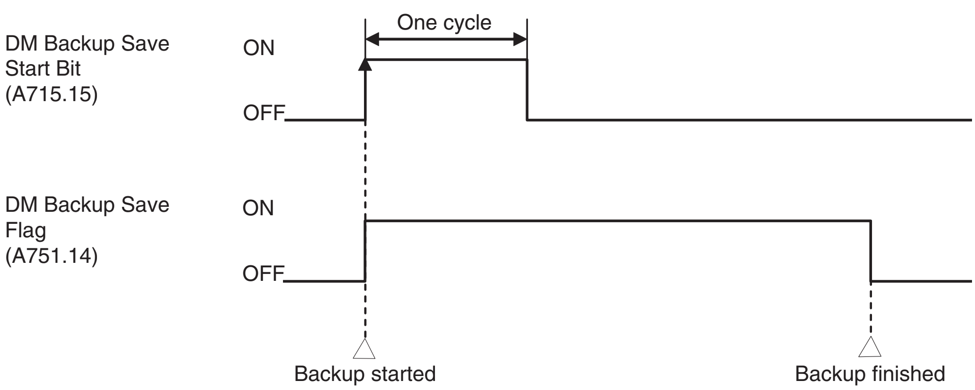
1 Check the *Restore D0- from backup memory* Check Box in the *Startup Data Read* Area of the PLC Setup from the CX-Programmer.
Also, set the number of words to be backed up starting from D0 in the *Number of CH of DM for backup* Box. Transfer the PLC Setup to the CPU Unit and turn ON the power supply.



2 Turn ON A751.15 (DM Backup Save Start Bit) from the CX-Programmer, a Programmable Terminal (PT), or a ladder program.
The specified number of words in the DM Area starting from D0 will be backed up to the built-in Flash Memory.

- Using a Ladder Program





When the saving operation has been completed, A751.14 (DM Backup Save Flag) will turn OFF.

Precautions for Safe Use

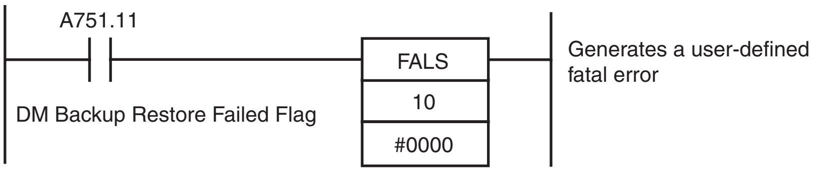
Power Interruptions during Backup

The BKUP indicator on the front of the CPU Unit will be lit when DM Area data is being saved to the built-in Flash Memory.

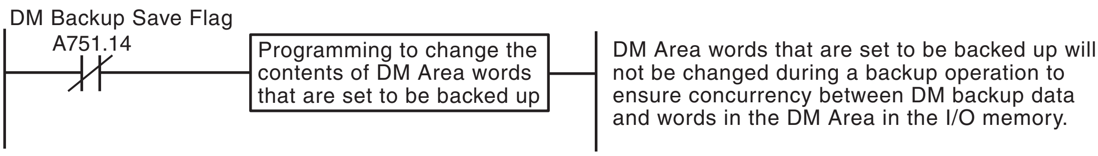
Do not turn OFF the power supply to the PLC while the indicator is lit. If the power supply to the PLC is turned OFF while the BKUP indicator is lit, data will not be backed up. In this case, the DM Backup Restore Failed Flag (A751.11) will turn ON when the power supply is turned ON again. Therefore, the backup data will not be restored to the DM Area. Transfer the data from the CX-Programmer to the DM Area again.

Precautions for Correct Use

- To prevent operation from starting if the DM backup data is not restored correctly when the power supply is turned ON, insert the following instructions into the ladder program to generate a fatal error.



- To ensure concurrency between DM backup data and the contents of the DM Area in the I/O memory, use exclusive processing in the ladder program so that contents of the DM Area words in the I/O memory that are set to be backed up are not changed during a backup operation.



- Data can be written up to 100,000 times to the built-in Flash Memory. Data cannot be written once this limit is exceeded. If writing fails, A315.15 (Backup Memory Error Flag) will turn ON.

Additional Information

Confirming Completion of DM Area Backup

If user programs or the parameter area is being saved to the backup memory using operations from the CX-Programmer, the backup operation will not be executed immediately even if A751.15 (DM Backup Save Start Bit) is turned ON. A751.14 (DM Backup Save Flag) will remain ON during this time and turn OFF when the DM backup operation has been completed. You can confirm the completion of DM backup by checking to see if the DM Backup Save Flag (A751.14) has been turned OFF.

16-4 Security Functions

The Security function can be used with any model of CP2E CPU Unit.

16-4-1 Ladder Program Read Protection

Read Protection

With the CX-Programmer, it is possible to set read protection using a password for the whole ladder program.

When the program is read-protected using a password, it is not possible to display or edit any of the ladder programs using the CX-Programmer unless the password is entered in the Disable Password Dialog Box from the CX-Programmer.

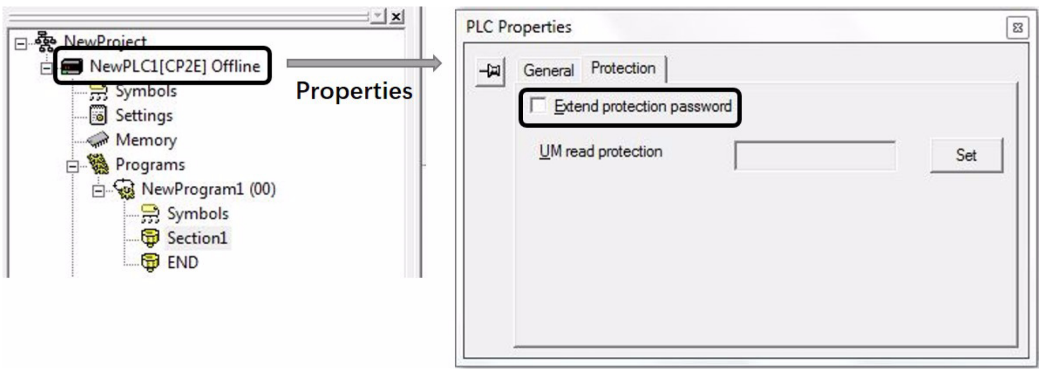
This enables improved security for PLC data in equipment.

● Read Protection Using Extended Passwords

Extend protection with UM read protection to ensure better protection for your design assets. Also, if the password is incorrect five times in a row, a password cannot be entered for two hours.

● Setting Protection

1 Right-click the PLC in the project tree to open the **Protection** Tab Page of the **PLC Properties**.



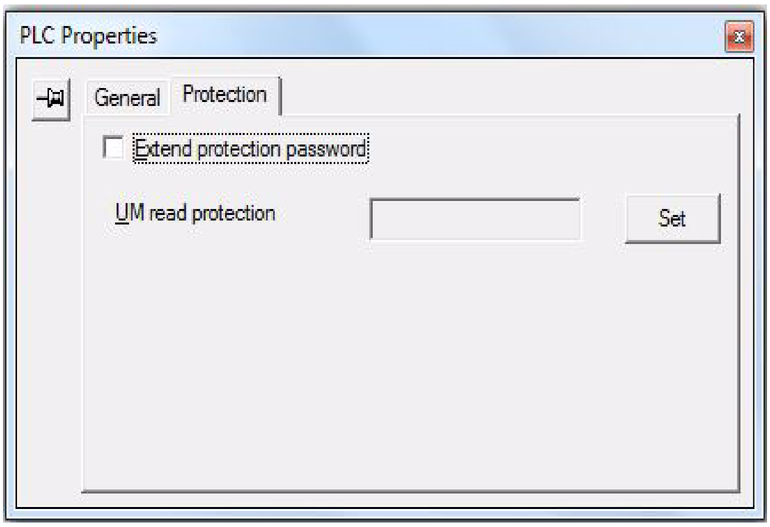
2 Set any password.

3 Select ***Protection-Setting*** from the PLC Menu.

4 Confirm that the setting item is checked, then click the **OK** button.

- Extend Password Protection

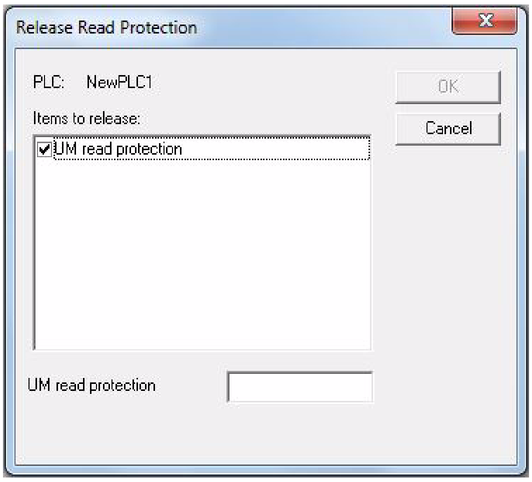
  It is possible to use longer passwords for UM read protection. Click the *Protection* Tab in the PLC Properties Dialog Box, select the *Extend protection password* Check Box, and enter the passwords.



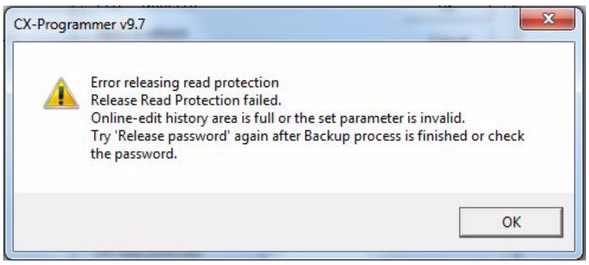
  The limits to the password text string lengths are 16 characters.

● Protection Release Procedure

1 Go online and select ***Protection - Release Password*** from the PLC menu.
The Release Read Protection Dialog Box will be displayed.



2 Enter the registered password.
If the password is incorrect, the message shown on the right will be displayed, and protection will not be released.
If incorrect passwords are entered for five consecutive attempts, the CPU Unit does not accept any more passwords for two hours.



● Auxiliary Area Bits Related to Password Protection

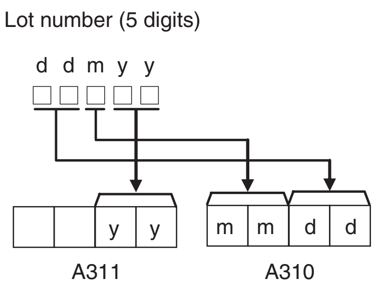
Name	Bit address	Description	Status after mode change	Startup hold settings
UM Read Protection Status	A99.00	Indicates whether or not the whole ladder programs are read-protected. OFF: UM read protection is not set. ON: UM read protection is set.	Hold	Hold

16-4-2 Protecting Program Execution Using the Lot Number

● Overview

The lot number is stored in words A310 and A311 of the CP2E CPU Unit's Auxiliary Area. These words generate a fatal error to prevent a user program from running on a PLC with a different production lot number. In addition, by setting passwords, you can prohibit loading of programs, and thereby prohibit copying of user programs. The lot number cannot be changed by the user.

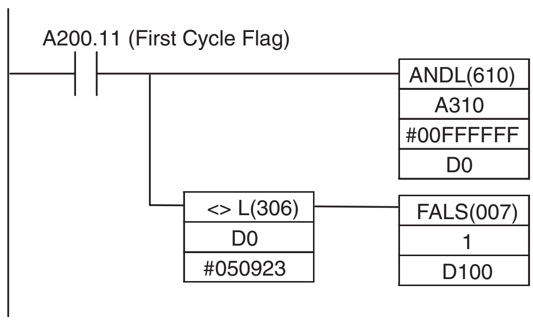
- The upper digits of the lot number are stored in A311 and the lower digits are stored in A310, as shown below.



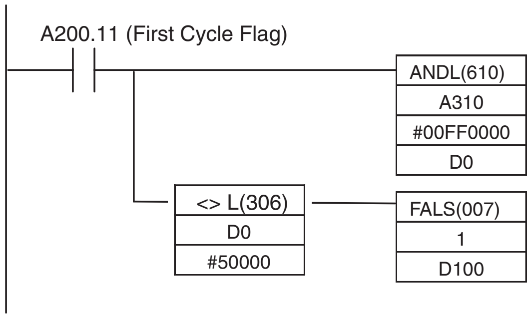
- X, Y, and Z in the lot number are converted to 10, 11, and 12, respectively, in A310 and A311.

● Programming Example

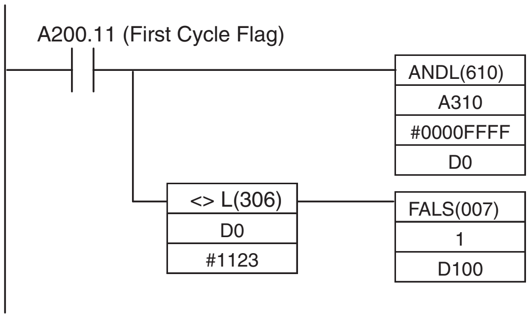
(1) The following instructions will create a fatal error to prevent the program from being executed when the lot number is not 23905.



(2) The following instructions will create a fatal error to prevent the program from being executed when the lot number does not end in 05.



(3) The following instructions will create a fatal error to prevent the program from being executed when the lot number does not begin with 23Y.



17

Analog Input/Output Option Board

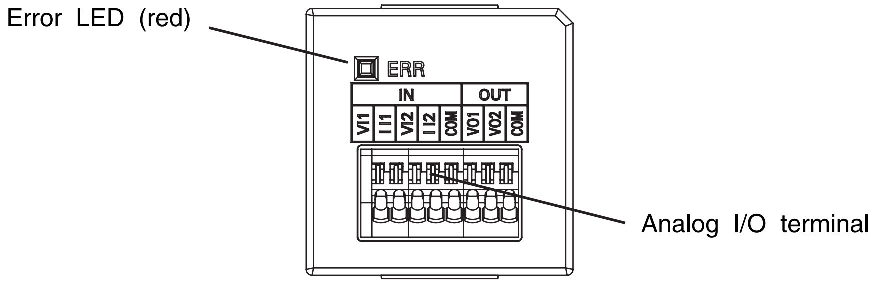
This section describes an overview of the Analog Option Board, describes its installation and setting methods, memory allocations, startup operation, refresh time, troubleshooting and how to use the Analog Option Board.

17-1 General Specifications

CP1 series Analog Option Boards are non-isolated analog units which allow you to easily realize analog input/output function for CP2E N☐☐-type CPU Unit.

Analog Option Board		Voltage Input 0V~10V (Resolution: 1/4000)	Current Input 0mA~20mA (Resolution: 1/2000)	Voltage Output 0V~10V (Resolution: 1/4000)
Analog I/O Option Board	CP1W-MAB221	2CH		2CH
Analog Input Option Board	CP1W-ADB21	2CH		---
Analog Output Option Board	CP1W-DAB21V	---		2CH

17-2 Part Names



Terminal Arrangement

● CP1W-ADB21

VI1	II1	VI2	II2	COM

● CP1W-DAB21V

VO1	VO2	COM

● CP1W-MAB221

VI1	II1	VI2	II2	COM	VO1	VO2	COM

Note Two COM are connected in inner circuit.

LED pattern

LED	Color	Description	Status	Remark
ERR	Red	Fault condition indicator	Flash	A communication error with CPU Unit has occurred at the unit.
			Lit	Other errors except the communication error.
			Not lit	Operation is normal.

17-3 Installation and Setting

17-3-1 Installation

The following processing explains how to install and remove an Analog Option Board.

Precautions for Correct Use

Always turn OFF the power supply to the CPU Unit and wait until all the operation indicators go out before installing or removing the Analog Option Board.

Not doing so may result in an unexpected operation.

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 Analog Option Board fit in the Option Board slot, and firmly press the Analog Option Board in until it snaps into place.

Only one Analog Option Board can be installed.

If two Analog Option Boards are mounted on a CP2E N30/40/60 CPU Unit, an option board error (non-fatal error) will occur. Both Analog Option Boards do not work (There is no DA output data, and AD conversion data cannot be read from CIO area), and ERR LED will be lit.

CPU Unit	Option board slot 1 (left)	Option board slot 2 (right)	Status after installation
CP2E N30/N40/N60 CPU Unit	○	×	OK
	×	○	OK
	○	○	NG

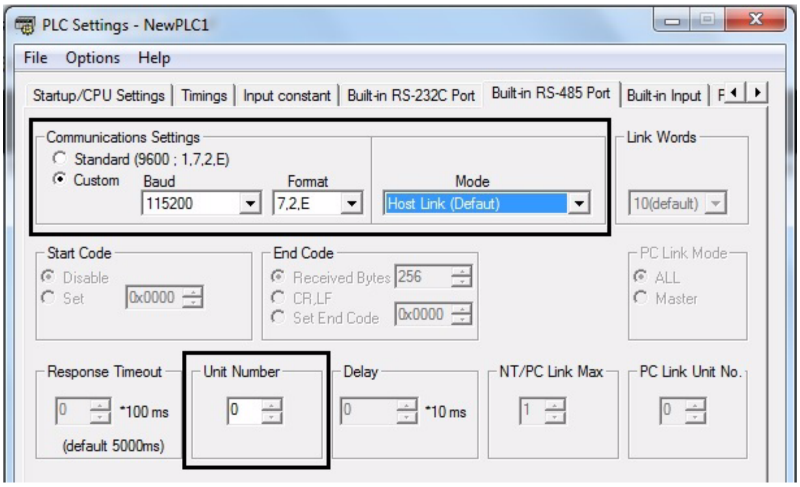
○ : Analog Option Board installed

× : Analog Option Board not installed

17-3-2 Setting

To use the analog option board on CP2E N☐☐-type CPU Unit, it is necessary to set the serial communication settings in the PLC Settings.

Connect the CX-Programmer to the CPU Unit, and then change the PLC Settings as follows.



Serial Port Tab Page

Parameter	Setting
Communications Settings	Select the *Custom* option, set the baud rate to *115200* and the format to *7,2,E*.
Mode	Select *Host Link (default)* or *Host Link*.
Unit Number	Select *0*.

Precautions for Correct Use

When the Analog Option Board is applied on the CP2E CPU Unit, it is necessary to set the baud rate to 115,200 and the mode to Host Link.

However, except the settings mentioned above, ERR LED will be lit if the communications is impossible between the CPU Unit and the Analog Option Board. Check and change the settings.

17-3-3 Removing

Always turn OFF the power supply to the CPU unit and wait until all the operation indicators go out.

Press the up/down lock-levers on both sides of the Analog Option Board at the same time to unlock the Option Board, and then pull it out.

17-4 Memory Allocation

17-4-1 CIO Area Allocation

The memory allocation about analog conversion in the CIO area of PLC is shown as the following diagram.

The range of the CIO area is CIO80 to CIO89.

The details of allocated CIO channels are described in the following table.

Channel	Contents		
	CP1W-ADB21	CP1W-DAB21V	CP1W-MAB221
CIO80	Analog Input 1	---	Analog Input 1
CIO81	Analog Input 2	---	Analog Input 2
CIO82 to CIO84	---	---	---
CIO85	---	Analog Output 1	Analog Output 1
CIO86	---	Analog Output 2	Analog Output 2
CIO87 to CIO89	---	---	---

17-4-2 Auxiliary Area Allocation

Analog Option Unit Status Area

Option board status area: A435 (initial value “0000H”)

CPU Unit	Option board slot	AR bits	Content	Error Process
CP2E N30/N40/N60 CPU Unit	Option board slot 1 (left)	A435.14	I/O option board run state	0: Initial state or unit abnormal state 1: Work normally
	Option board slot 2 (right)	A435.15		
CP2E N14/N20 CPU Unit	Option board slot	A435.14		

Note The flag sets ON if Analog Option Board has already worked normally. Then A/D input data can be read and D/A output data can be written.

Output off bit: A500.15

AR Bits	Content	Error Process
A500.15	Output Off Bit	0: Output effective 1: Analog option board DA output clear

Note This bit will also affect other PLC output channels.

Option board error bit: A315.13

AR Bits	Content	Error Process
A315.13	Option board error bit	• Turn ON when two Analog Option Boards are installed on CP2E N30/40/60 CPU Unit. • Cleared when the corresponding anomaly is cleared.

Option board error detail information: A424

CPU Unit	Option board slot	AR bits	Content	Error Process
CP2E N30/N40/N60 CPU Unit	Option board slot 1 (left)	A424.00	Option board error detail information	0: Cleared at the timing when the cause is released. 1: An option board error occurs.
	Option board slot 2 (right)	A424.01		
CP2E N14/N20 CPU Unit	Option board slot	A424.00		

Note When Analog Option Boards are installed in both option board slots, A424.00 and A424.01 are all set ON.

17-5 Analog Input Option Board

Each CP1W-ADB21 Analog Input Option Board provides two analog inputs.

- The analog input signal ranges are 0 to 10 V (with a resolution 1/4,000) and 0 to 20 mA (with a resolution 1/2,000).

17-5-1 Main Specifications

Item		Specifications	
		Voltage Input	Current Input
Input signal range		0 V to 10 V	0 mA to 20 mA
Max. rated input		0 V to 15 V	0 mA to 30 mA
External input impedance		200 kΩ min.	Approx. 250 Ω
Resolution		1/4000 (full scale)	1/2000 (full scale)
Overall accuracy	25°C	±0.5% (full scale)	±0.6% (full scale)
	0 to 60°C	±1.0% (full scale)	±1.2% (full scale)
	-20 to 0°C	±1.3% (full scale)	±1.5% (full scale)
A/D conversion data		0000 to 0FA0 hex	0000 to 07D0 hex
Averaging function		Not supported	
Conversion time		Inner sample time 2ms/point Refresh time refers to *17-9 Analog Option Board Refresh Time*	
Isolation method		None	
Current consumption		5 VDC: 20 mA max.	

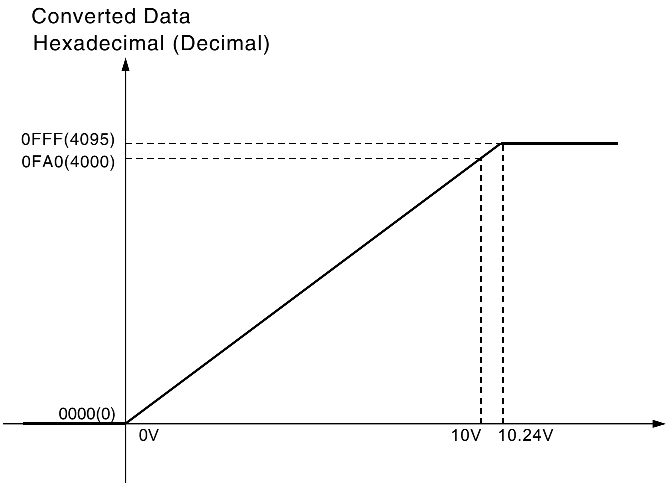
17-5-2 Analog Input Signal Ranges

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.

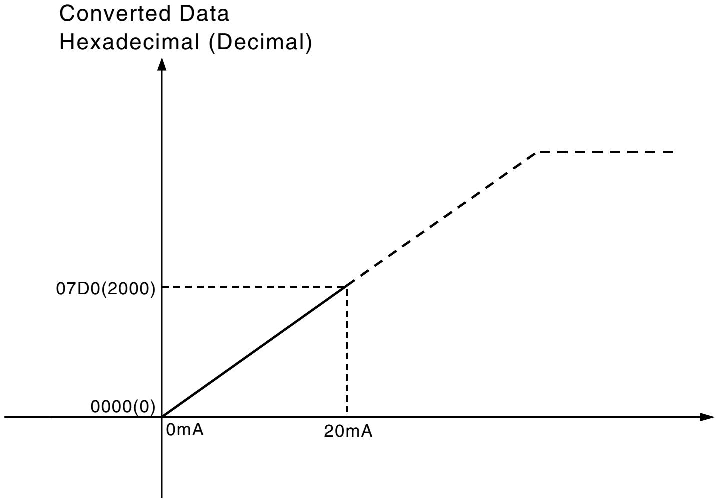
● 0 to 10 V

The 0 to 10 V range corresponds to the hexadecimal values 0000 to 0FA0 (0 to 4000). The entire data range is 0000 to 0FFF (0 to 4095).

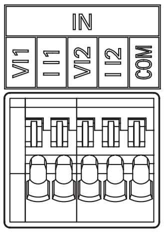


- **0 to 20 mA**

The 0 to 20 mA range corresponds to the hexadecimal values 0000 to 07D0 (0 to 2000). The possible data range is 0000 to 0FFF (0 to 4095). But it is strongly suggested that the input current must not exceed 30 mA.



Analog Input Terminal Arrangement

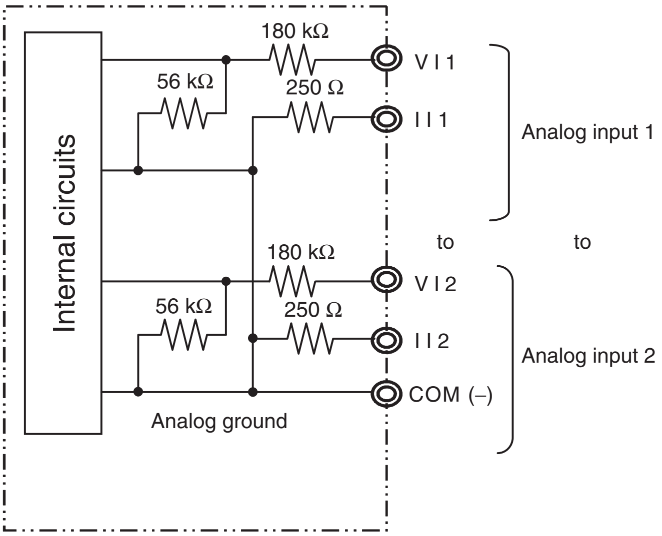


VI1	Voltage Input 1
II1	Current Input 1
VI2	Voltage Input 2
II2	Current Input 2
COM	Input Common

Note When using current inputs, voltage input terminals must be short-circuited with current input terminals.

17-5-3 Wiring

Internal Circuits



Applicable Cables and Terminal Wiring

Applicable Cables

Solid wire or ferrules can be used.

- Recommended solid wire

Wire type	Wire size
Solid Wire	0.2mm^2 to 0.5mm^2 (AWG24 to AWG20)

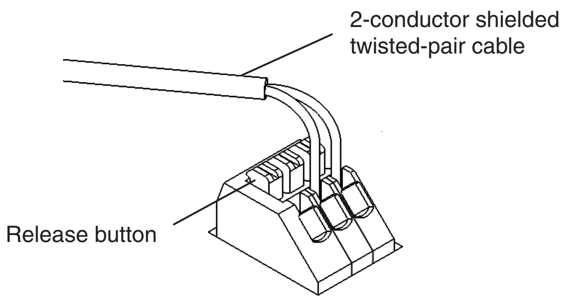
- Recommended ferrules

Manufacturer	Model	Applicable wire
Phoenix Contact	AI-0.25-12	0.2mm^2 (AWG24)

Note Do not connect bare stranded wires directly to terminals.

Terminal Wiring

When wiring the analog I/O terminal block, treat either solid or stranded wires directly.

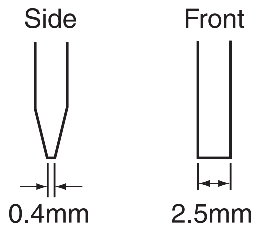


- To make the connection, press the release button in with a small flat blade screwdriver and push the line in while the lock is released. Remove the screwdriver and lock it inside.
- To disconnect the wiring, press the release button in with a small flat blade screwdriver and pull the line out while the lock is released.

Note 1 Ferrules without plastic sleeve cannot be used.
2 When using stranded wire, twist the core so that the barbed wires cannot protrude.
3 Do not solder-plate the end of cable.

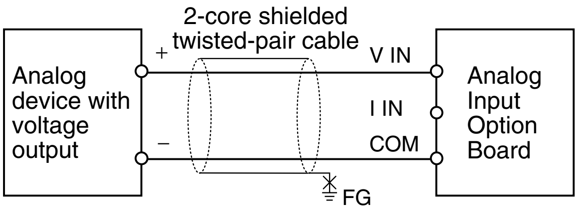
The screwdriver shown below is recommended for wiring.

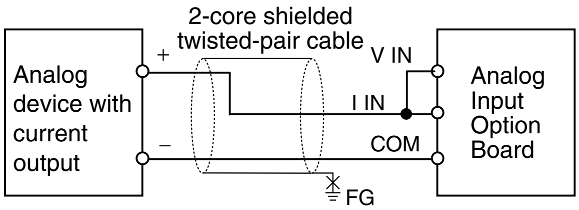
Model	Manufacturer
SZS 0.4×2.5	Phoenix Contact



Wiring for Analog Inputs

To prevent noise, 2-core shielded twisted-pair cable should be used.





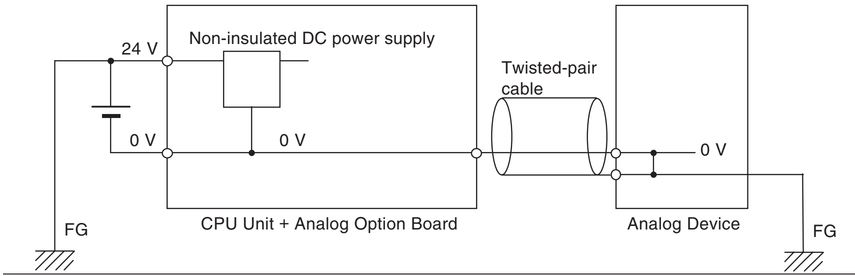
Note 1 When an input is not being used, short the V IN, I IN and COM terminals.

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.

Precautions for Correct Use

When connecting the analog option board to an outside analog device, either ground the 0 V side of the PLC's external power supply or do not ground the PLC's external power supply at all. Otherwise the PLC's external power supply may be shorted depending on the connection methods of the outside analog device. DO NOT ground the 24 V side of the PLC's external power supply, as shown in the following diagram.



17-6 Analog Output Option Board

Each CP1W-DAB21V Analog Output Option Board provides two analog outputs.

- The analog output signal range is 0 to 10 V (with a resolution 1/4,000).

17-6-1 Main Specifications

Item		Specifications	
		Voltage Output	Current Output
Output signal range		0 V to 10 V	---
External output allowable load resistance		2 kΩ min.	---
External output impedance		0.5 Ω max.	---
Resolution		1/4,000 (full scale)	---
Overall accuracy	25°C	±0.5% (full scale)	---
	0 to 60°C	±1.0% (full scale)	---
	-20 to 0°C	±1.3% (full scale)	---
D/A conversion data		0000 to 0FA0 hex	---
Conversion time		Inner conversion time 2ms/point Refresh time refers to *17-9 Analog Option Board Refresh Time*	
Isolation method		None	
Current consumption		5 VDC: 60 mA max.	

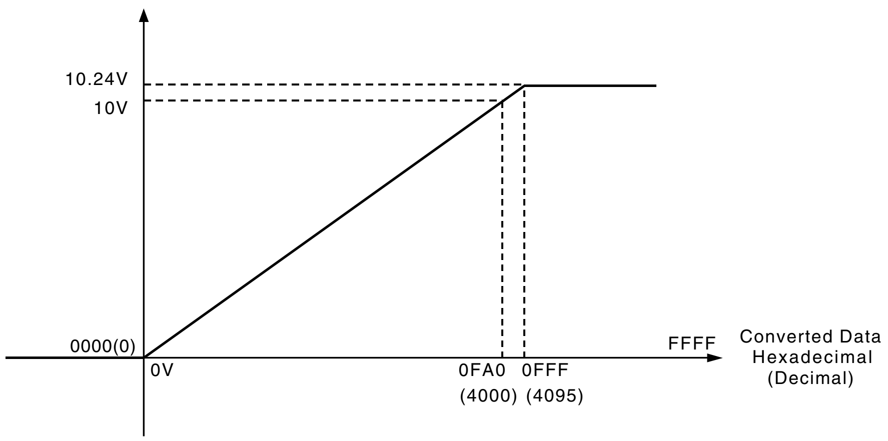
17-6-2 Analog Output Signal Ranges

The analog values depend on the output signal range, as shown in the following diagram.

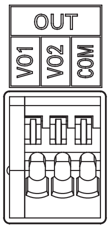
Note When the output exceeds the specified range, the output signal will be fixed at either the lower limit or upper limit.

● 0 to 10 V

The hexadecimal values 0000 to 0FA0 (0 to 4000) correspond to an analog voltage range of 0 to 10 V. The entire output range is 10 to 10.24 V.



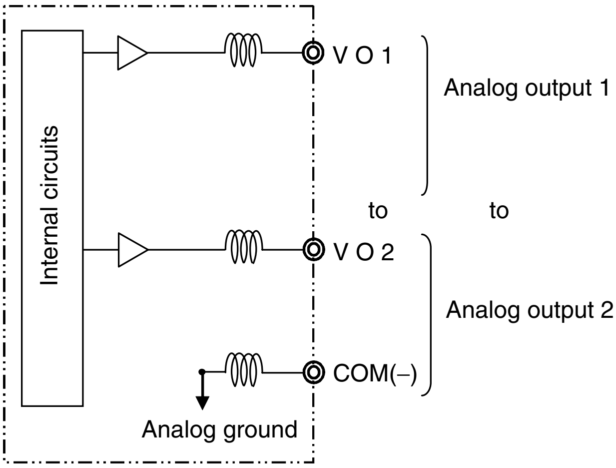
Analog Output Terminal Arrangement



VO1	Voltage Output 1
VO2	Voltage Output 2
COM	Output Common

17-6-3 Wiring

Internal Circuits



Applicable Cables and Terminal Wiring

● Applicable Cables

Solid wire or ferrules can be used.

- Recommended solid wire

Wire type	Wire size
Solid Wire	0.2mm^2 to 0.5mm^2 (AWG24 to AWG20)

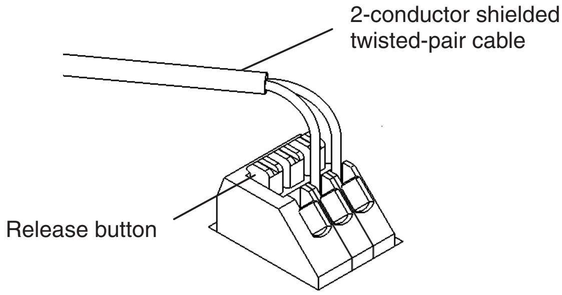
- Recommended ferrules

Manufacturer	Model	Applicable wire
Phoenix Contact	AI-0.25-12	0.2mm^2 (AWG24)

Note Do not connect bare stranded wires directly to terminals.

● Terminal Wiring

When wiring the analog I/O terminal block, treat either solid or stranded wires directly.

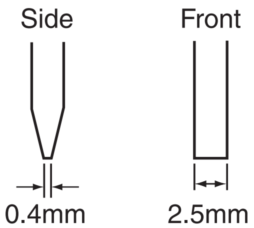


- To make the connection, press the release button in with a small flat blade screwdriver and push the line in while the lock is released. Remove the screwdriver and lock it inside.
- To disconnect the wiring, press the release button in with a small flat blade screwdriver and pull the line out while the lock is released.

Note 1 Ferrules without plastic sleeve cannot be used.
2 When using stranded wire, twist the core so that the barbed wires cannot protrude.
3 Do not solder-plate the end of cable.

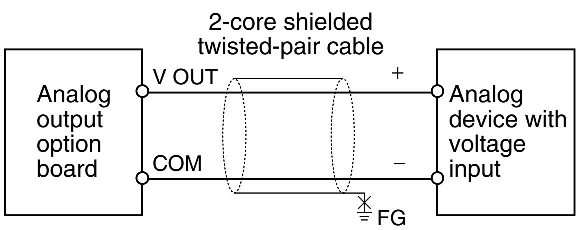
The screwdriver shown below is recommended for wiring.

Model	Manufacturer
SZS 0.4×2.5	Phoenix Contact



Wiring for Analog Outputs

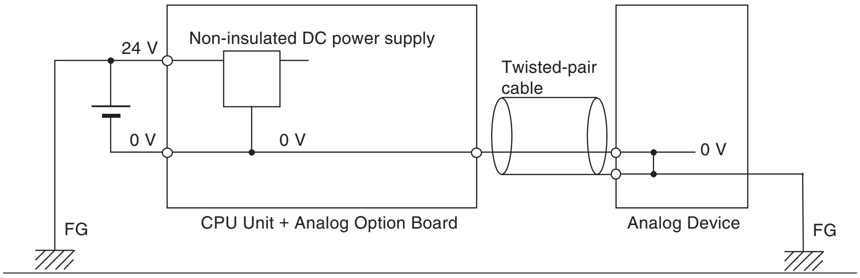
To prevent noise, 2-core shielded twisted-pair cable should be used.



Note 1 Separate wiring from power lines (AC power supply lines, high-voltage lines, etc.)
2 When there is noise in the power supply line, install a noise filter on the input section and the power supply.
3 When external power is supplied, 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 CP2E N☐☐-type 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 CP2E N☐☐-type CPU Unit.

Precautions for Correct Use

When connecting the analog option board to an outside analog device, either ground the 0 V side of the PLC's external power supply or do not ground the PLC's external power supply at all. Otherwise the PLC's external power supply may be shorted depending on the connection methods of the outside analog device. DO NOT ground the 24 V side of the PLC's external power supply, as shown in the following diagram.



17-7 Analog I/O Option Board

Each CP1W-MAB221 Analog I/O Option Board provides two analog inputs and two analog outputs.

- The analog input signal ranges are 0 to 10 V (with a resolution 1/4,000) and 0 to 20 mA (with a resolution 1/2,000).
- The analog output signal ranges are 0 to 10 V (with a resolution 1/4,000).

17-7-1 Main Specifications

Item			Specifications	
			Voltage I/O	Current I/O
Analog Input Section	Input signal range		0 V to 10 V	0 mA to 20 mA
	Max. rated input		0 V to 15 V	0 mA to 30 mA
	External input impedance		200 kΩ min.	Approx. 250 Ω
	Resolution		1/4,000 (full scale)	1/2,000 (full scale)
	Overall accuracy	25°C	±0.5% (full scale)	±0.6% (full scale)
		0 to 60°C	±1.0% (full scale)	±1.2% (full scale)
		-20 to 0°C	±1.3% (full scale)	±1.5% (full scale)
	A/D conversion data		0000 to 0FA0 hex	0000 to 07D0 hex
	Averaging function		Not supported	
Analog Output Section	Output signal range		0 V to 10 V	---
	External output allowable load resistance		2 kΩ min.	---
	External output impedance		0.5 Ω max.	---
	Resolution		1/4,000 (full scale)	---
	Overall accuracy	25°C	±0.5% (full scale)	---
		0 to 60°C	±1.0% (full scale)	---
		-20 to 0°C	±1.3% (full scale)	---
	D/A conversion data		0000 to 0FA0 hex	---
Conversion time			Inner conversion time 6ms (4CH total) Refresh time refers to *17-9 Analog Option Board Refresh Time*	
Isolation method			None	
Current consumption			5 VDC: 80 mA max.	

17-7-2 Analog I/O Signal Ranges

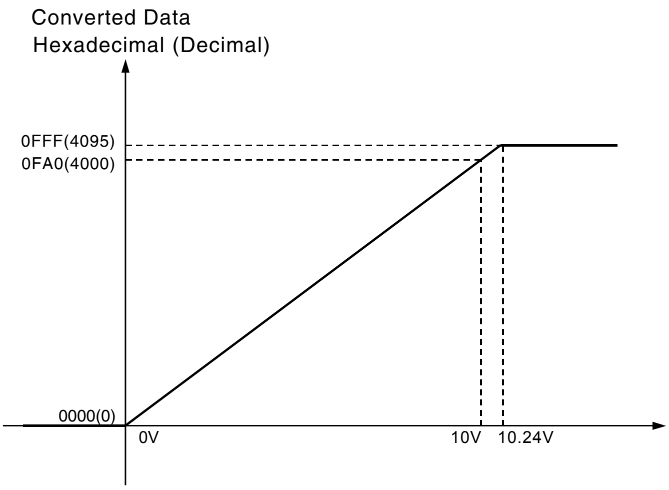
The analog values depend on the I/O signal ranges, as shown in the following diagrams.

Note When the input exceeds the specified range, the A/D conversion data will be fixed at either the lower limit or upper limit.
When the output exceeds the specified range, the output signal will be fixed at either the lower limit or upper limit.

Analog Input Signal Ranges

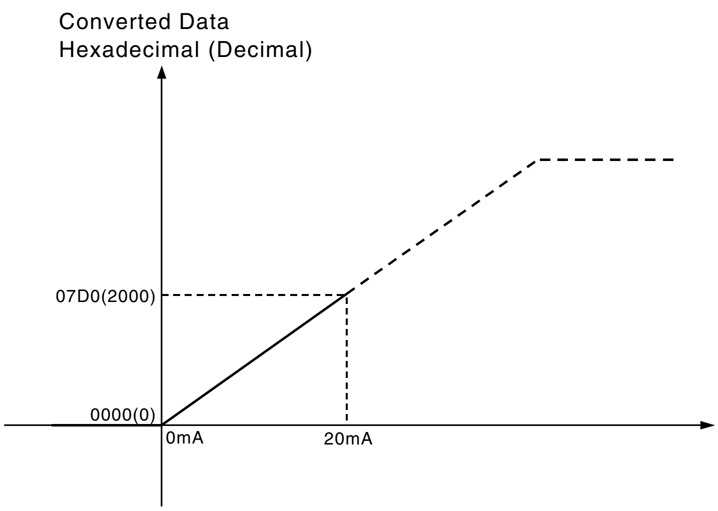
● 0 to 10 V

The 0 to 10 V range corresponds to the hexadecimal values 0000 to 0FA0 (0 to 4000). The entire data range is 0000 to 0FFF (0 to 4095).



● 0 to 20 mA

The 0 to 20 mA range corresponds to the hexadecimal values 0000 to 07D0 (0 to 2000). The possible data range is 0000 to 0FFF (0 to 4095). But it is strongly suggested that the input current mustn't exceed 30 mA.



17-7 Analog I/O Option Board

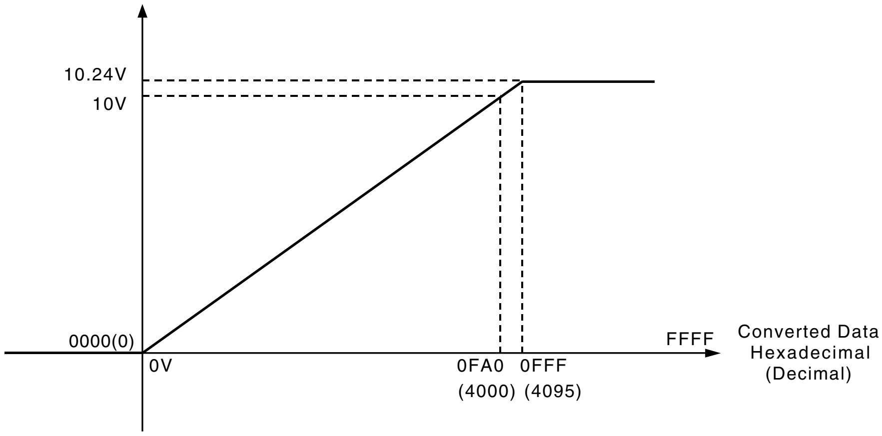
17

17-7-2 Analog I/O Signal Ranges

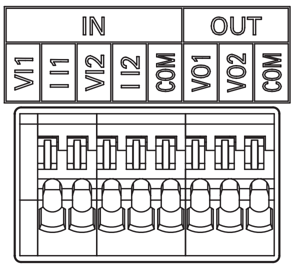
Analog Ouput Signal Ranges

- **0 to 10 V**

The hexadecimal values 0000 to 0FA0 (0 to 4000) correspond to an analog voltage range of 0 to 10 V. The entire output range is 10 to 10.24 V.



Analog I/O Terminal Arrangement

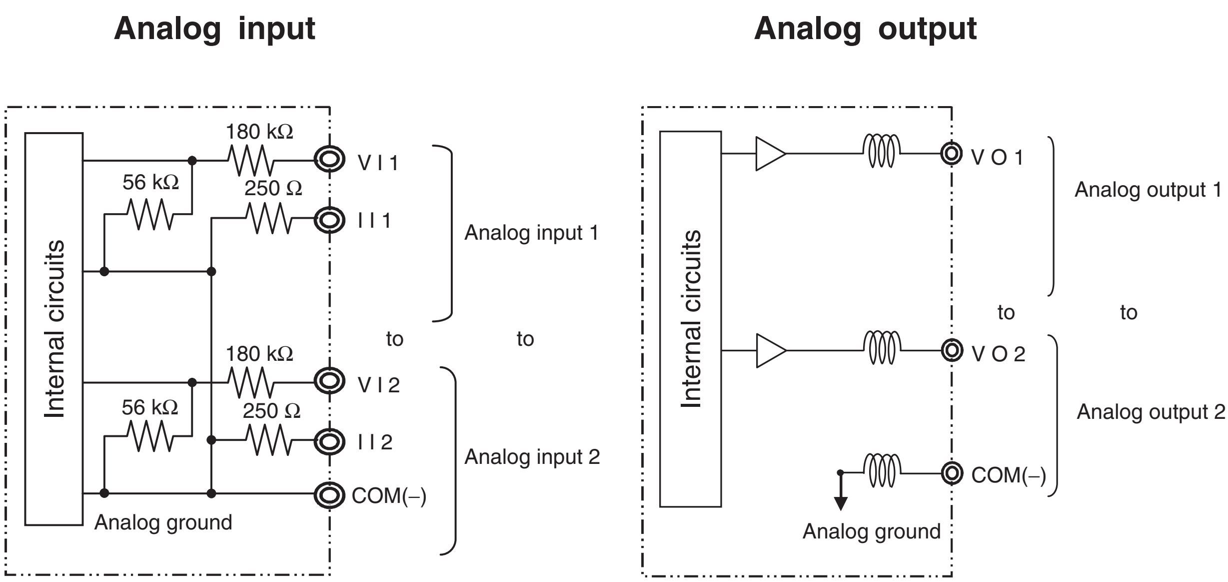


VI1	Voltage Input 1
II1	Current Input 1
VI2	Voltage Input 2
II2	Current Input 2
COM	Analog I/O Common
VO1	Voltage Output 1
VO2	Voltage Output 2
COM	Analog I/O Common

Note When using current inputs, voltage input terminals must be short-circuited with current input terminals.

17-7-3 Wiring

Internal Circuits



Applicable Cables and Terminal Wiring

● Applicable Cables

Solid wire or ferrules can be used.

- Recommended solid wire

Wire type	Wire size
Solid Wire	0.2mm^2 to 0.5mm^2 (AWG24 to AWG20)

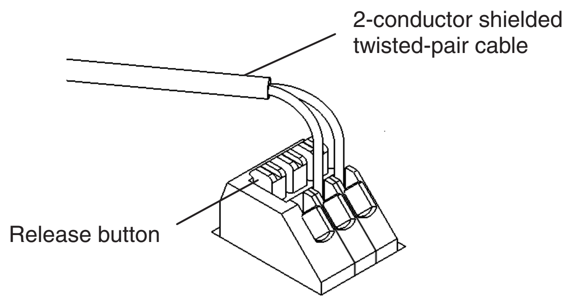
- Recommended ferrules

Manufacturer	Model	Applicable wire
Phoenix Contact	AI-0.25-12	0.2mm^2 (AWG24)

Note Do not connect bare stranded wires directly to terminals.

● Terminal Wiring

When wiring the analog I/O terminal block, treat either solid or stranded wires directly.

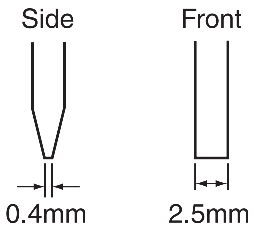


- To make the connection, press the release button in with a small flat blade screwdriver and push the line in while the lock is released. Remove the screwdriver and lock it inside.
- To disconnect the wiring, press the release button in with a small flat blade screwdriver and pull the line out while the lock is released.

Note 1 Ferrules without plastic sleeve cannot be used.
2 When using stranded wire, twist the core so that the barbed wires cannot protrude.
3 Do not solder-plate the end of cable.

The screwdriver shown below is recommended for wiring.

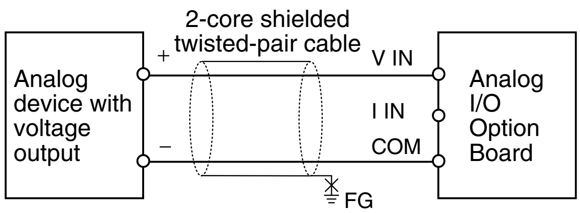
Model	Manufacturer
SZS 0.4×2.5	Phoenix Contact

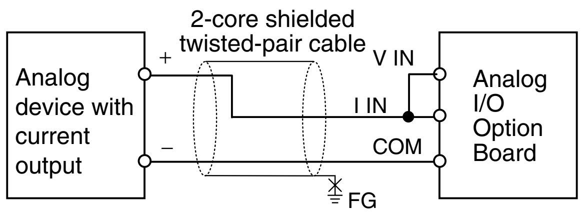


Wiring for Analog I/O

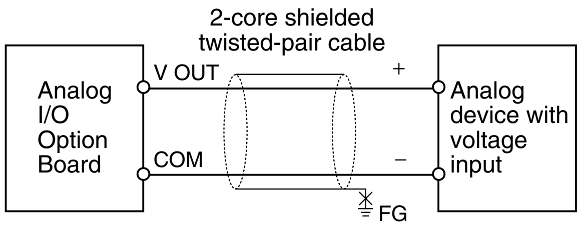
To prevent noise, 2-core shielded twisted-pair cable should be used.

● Wiring for Analog Inputs





● Wiring for Analog Outputs

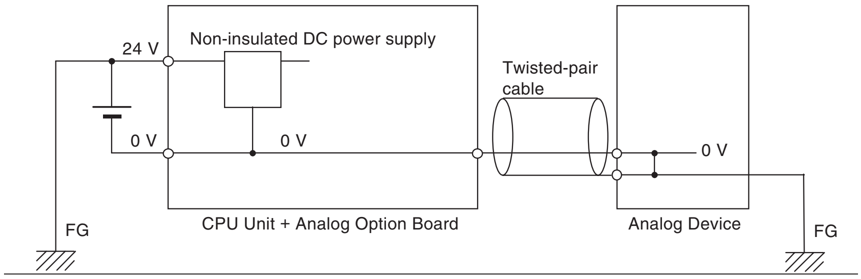


Note 1 When an input is not being used, short the V IN, I IN and COM terminals.
2 Separate wiring from power lines (AC power supply lines, high-voltage lines, etc.)
3 When there is noise in the power supply line, install a noise filter on the input section and the power supply.
4 When external power is supplied, 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 CP2E N□□-type 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 CP2E N□□-type CPU Unit.

Precautions for Correct Use

When connecting the analog option board to an outside analog device, either ground the 0 V side of the PLC's external power supply or do not ground the PLC's external power supply at all. Otherwise the PLC's external power supply may be shorted depending on the connection methods of the outside analog device. DO NOT ground the 24 V side of the PLC's external power supply, as shown in the following diagram.



17-8 Startup Operation

After the power is turned ON, analog option board starts the initialization process. If the initialization finishes normally, the initialization completed flag in related status area (Refer to *17-4-2 Auxiliary Area Allocation: A435*) will be set. Therefore, status monitor content must be added in ladder. Only when the initialization process has finished, user can use the A/D conversion data or write the output data.

The analog input data will be 0000 until the initial processing is completed.

It takes about 3 seconds from the CP2E CPU Unit running to the Analog Option Board initialization completed.

17-9 Analog Option Board Refresh Time

The inner conversion time of the Analog Option Board is 2ms/point. The refresh time of data conversion in the CPU Unit is shown as follows.

The refresh time differs in accordance with the cycle time.

Below are typical values for reference only.

Analog Option Board	Cycle time (ms)		
	1ms	10ms	20ms
CP1W-ADB21	16~40	20~60	20~100
CP1W-DAB21V	9~37	26~58	46~86
CP1W-MAB221(AD)	14~62	18~109	20~160
CP1W-MAB221(DA)	9~53	26~102	46~150

17-10 Trouble Shooting

Trouble-shooting with Indicators

ERR Indicator	Error	Probably Cause	Correction	Auxiliary Area Allocations	AD/DA function
Lit	CPU Unit service monitoring error	Service from the CPU Unit was not completed within the fixed interval.	Check and correct the CPU Unit's operating environment. Check serial communication setting.	A435.14, A435.15 will be OFF	AD/DA conversion will stop. The analog input conversion data stops refreshing and the analog output conversion output becomes 0V.
	Option board error	An error occurred in the Analog Option Board.	Restart the CPU Unit. Replace the Analog Option Board if the error recurs.		
		Two Analog Option Boards are installed.	Install one Analog Option Board only.	A353.13 will be ON	
Flashing	Communication error	The communication between PLC is out of service	Check if PLC is running normally.	A435.15 will be OFF	AD/DA conversion will stop. The analog input conversion data stops refreshing and the analog output conversion output becomes 0V. If the communication recovers from error, the AD/DA conversion will start again.

17-11 The Use of Analog Option Board

17-11-1 Procedure

Step	Description
Mount the Analog Option Board	Mount the Analog Option Board into the option port.
↓	
Connection with the analog input/output device	Connect to the analog input/output device.
↓	
Connection with the analog input/output device	Set communication protocol in PLC Settings to Host Link, 115200bps, 7,2,E.
↓	
Restart the CPU Unit	Turn the power of the CPU Unit OFF and ON again.
↓	
Make the ladder program	Wait until the Analog Option Board finishes initialization and works normally.
↓	
AD/DA convert	Read the analog input data/write the analog output data.

Note 1 If PLC communication protocol setting is error, the option board will always try to link the PLC, and the error LED will be lit.

2 Only when the initialization process has finished (A435.15 sets on), user can use the A/D conversion data or write the D/A output data.

17-11-2 Program Example

Use the analog option board to carry out 2CH AD inputs and 1CH DA output at the same time.

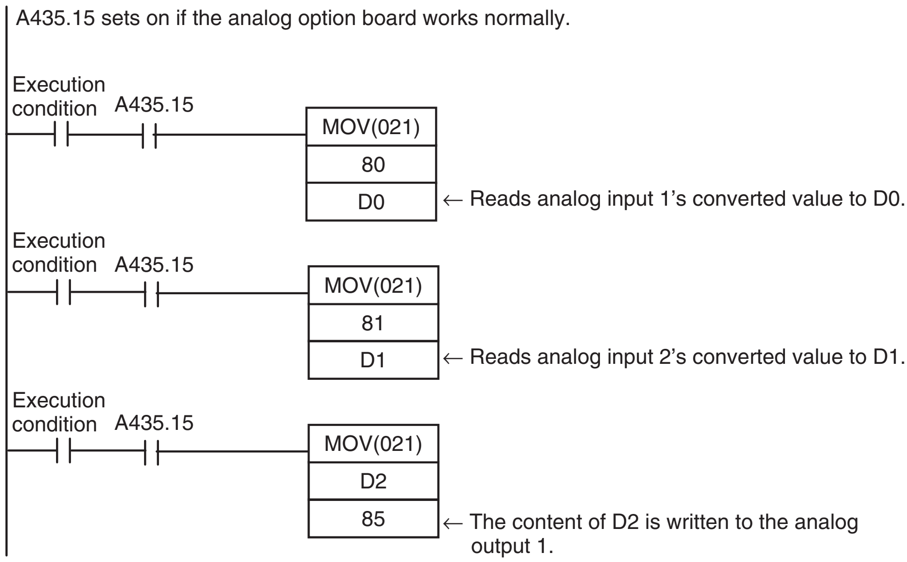
The ranges of AD/DA are as follows:

Analog input1: 0~10V

Analog input2: 0~20mA

Analog output1: 0~10V

System composing: CP2E-N☐☐D☐-☐ + CP1W-MAB221



18

Programming Device Operations

This section describes the use of the CX-Programmer to create a ladder programs to operate the CP2E, transfer the program to the CP2E, and debug the program. It also describes other basic functions of the CX-Programmer.

18-1 Programming Devices Usable with the CP2E

Refer to *Section 4 Programming Device* in the *CP2E CPU Unit Hardware User's Manual* (Cat. No. W613) for the connection method with a CP2E CPU Unit.

Precautions for Correct Use

- When using the full version of CX-Programmer provided in CX-One, refer to the *CX-Programmer Operation Manual* (Cat. No. W446).
- A Programing Console cannot be used for the CP2E. Use the CX-Programmer.

18-2 Overview of CX-Programmer

18-2-1 CX-Programmer

The CX-Programmer is a programming application for creating the ladder programs that are executed in a CP2E CPU Unit.
In addition to ladder program creation, the CX-Programmer also has functions that are needed to set up and operate the CP2E, including functions for debugging ladder programs, displaying addresses and present values, monitoring, setting the connected PLC, programming, and monitoring.

The CX-Programmer can be installed foom CX-One installer. Refer to *CX-One FA Integrated Tool Package Setup Manual* (Cat. No. W463) for details.

For details on the operation of the CX-Programmer, refer to the CX-Programmer Online Help.

18-2-2 CX-Programmer Flow from Startup to Operation

The flow of using the CX-Programmer from startup through starting PLC operation is shown below.

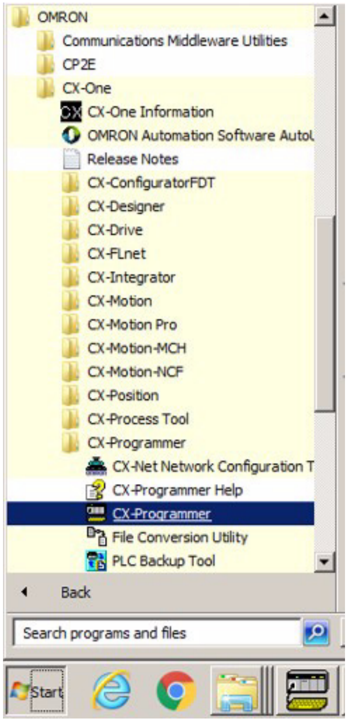
1	Start CX-Programmer	Select ***Start - Programs - OMRON - CX-One - CX-Programmer - CX-Programmer.***
2	Create new project	Select *CP2E* or *CP2E-N* in the Change PLC Dialog Box.
3	Create program	Example: Press the **C** Key to input a NC input condition.
4	Save program	Select ***File - Save As.***
5	Connect online to the CP2E	Select ***PLC - Work Online.***
6	Change operating mode	Select ***PLC - Operating mode - Program.***
7	Transfer program/PLC system settings	Select ***PLC - Transfer - Transfer [PC → PLC].***
8	Restart the CP2E	Once turn OFF the power supply to the CP2E CPU Unit, then turn ON again.
9	Start operation	CP2E CPU Unit operation will start.

Start CX-Programmer

Select ***Start - Programs - OMRON - CX-One - CX-Programmer.***

The CX-Programmer will start.

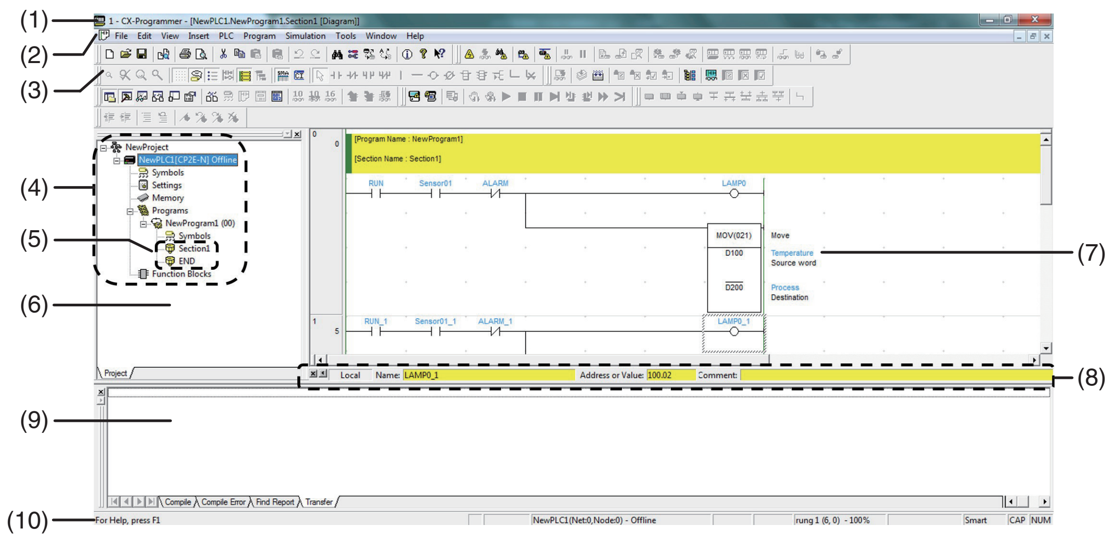
The title display will appear, followed by the Main Window.



Names and Functions of Parts of the Main Window

This section describes the names and functions of each part of the Main Window of the CX-Programmer. For details on the functions and operation of CX-Programmer, refer to the CX-Programmer Online Help.

● Main Window



(1)Title Bar
Displays the name of the project.

(2)Main Menu
Displays the menus from which commands are selected.

(3)Toolbar
Displays the icons for executing commands.

(4)Project Tree and (6) Project Workspace
Used to manage programs and settings.

(5)Sections
Allow ladder programming to be split up into a number of parts.

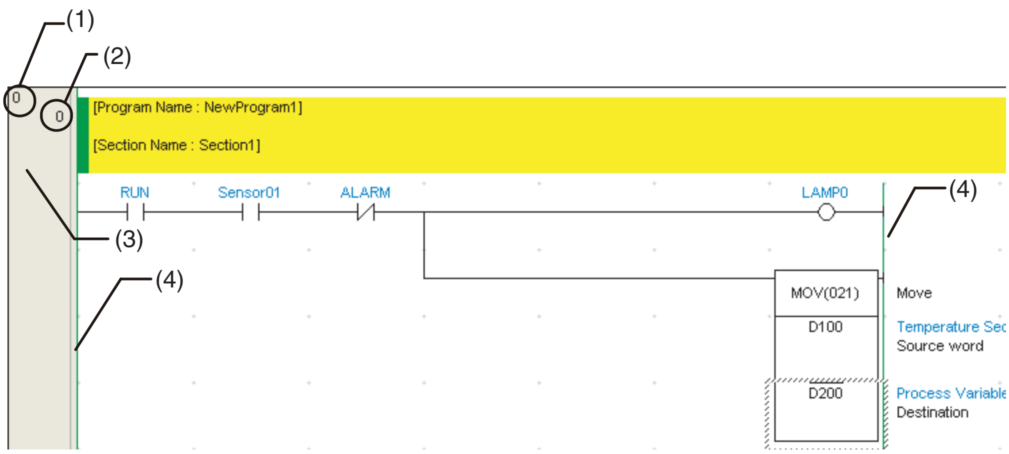
(7)Ladder Section Window
A window that is used to create and edit ladder programs.

(8)I/O Comment Bar
Displays the name, address, value, and I/O comment of the symbol selected with the cursor.

(9)Output Window
Displays messages, such as search results and errors.

(10)Status Bar
Displays information such as the PLC name, online/offline status, and position of the active cell.

● Ladder Section Window



(1)Rung Number

(2)Program Address

(3)Rung Header
If a rung is incomplete, a red line will be displayed on the right side of the rung header.

(4)Bus Bar

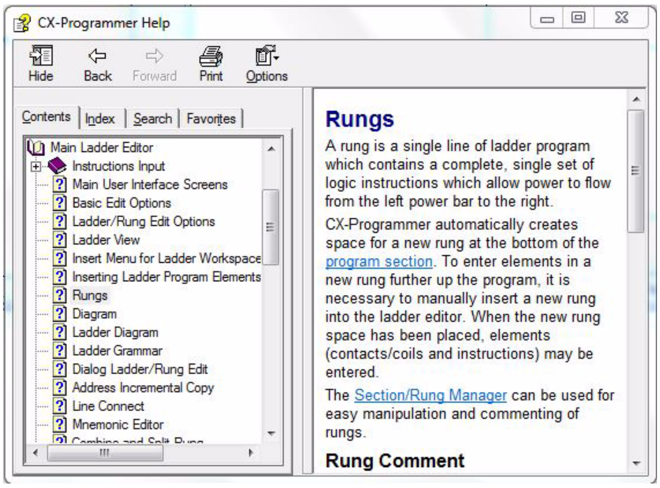
18-2-3 Help

The CX-Programmer Help describes all the operations of CX-Programmer. It provides an introduction to the various windows and panes and describes basic operations, ladder program creation, and monitoring. It also describes each of the instructions, including operand notation and contents.

Accessing CX-Programmer Help

Press the **F1** Key from the CX-Programmer.

The Help Window will be displayed.



Accessing the CX-Programmer Instruction Reference

For an explanation of an instruction used in ladder programming, refer to the CX-Programmer Instruction Reference.

- **Displaying the Instruction Reference from the Main Menu of the CX-Programmer**

  Select ***Instruction Reference - CP2E*** from the Help Menu.

  The CX-Programmer Instruction Reference Window will be displayed.

- **Displaying the Instruction Reference while Creating a Ladder Program**

  While creating an instruction in a ladder program in Smart Input Mode, press the F1 Key to display the Instruction Reference page for the instruction being edited.

Accessing the CP2E I/O Memory Reference

To check the CP2E I/O memory address map from the CX-Programmer, select ***I/O Memory Reference*** from the Help Menu.

18-3 Creating a Ladder Program

This section describes the use of CX-Programmer to create a ladder program.

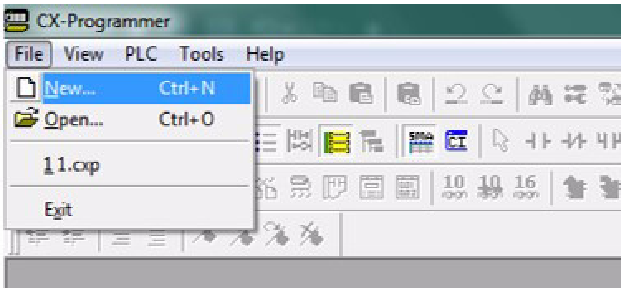
18-3-1 Inputting a Ladder Program

This section shows how to input a ladder program for an example application using the CX-Programmer commands.

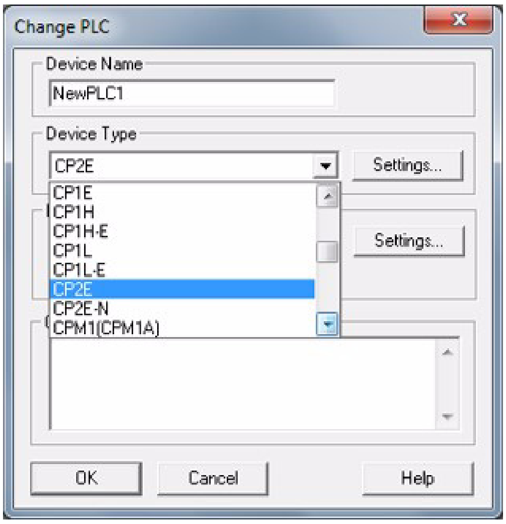
Creating a New Project

To use the CX-Programmer, the first step is to create a new project. To create a new project, we must specify the PLC type and CPU Unit model for which the ladder program and data to be created will be used.

1. Select **New** from the File Menu.The Change PLC Dialog Box will be displayed.

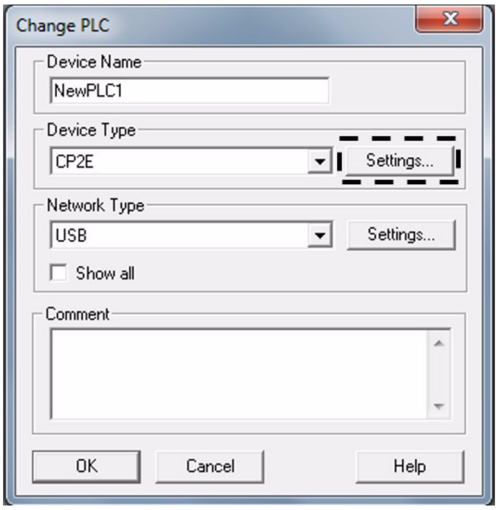


2. The CP2E will already be selected as the Device Type.

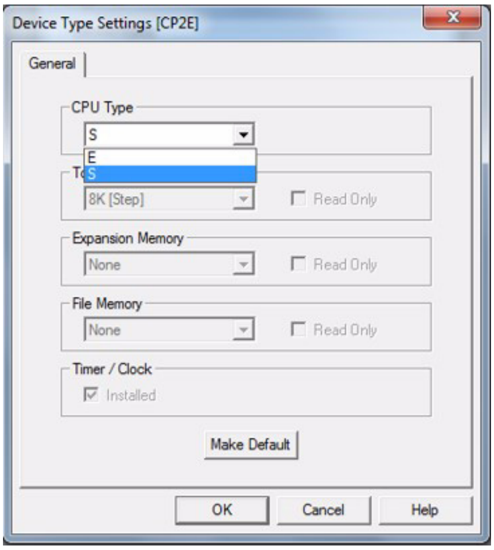


3 Click the **Settings** Button.

The PLC Type Settings Dialog Box will be displayed.

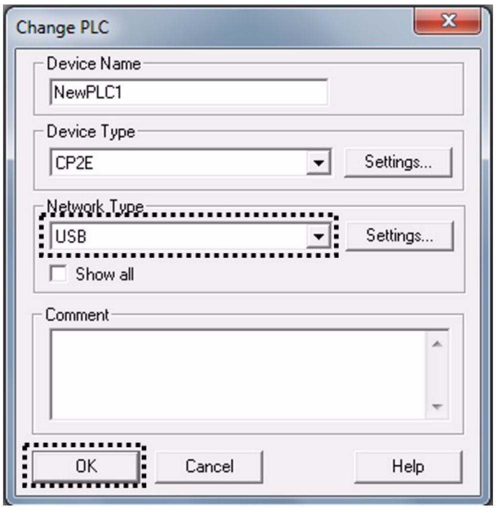


4 Select a CPU Unit model in the CPU Type box, and then click the **OK** Button. The PLC Type Settings Dialog Box will close.



5 Confirm that "USB" is displayed as the network type, and then click the **OK** Button.

The Change PLC Dialog Box will close, and the Main Window will be displayed for a new project.



Additional Information

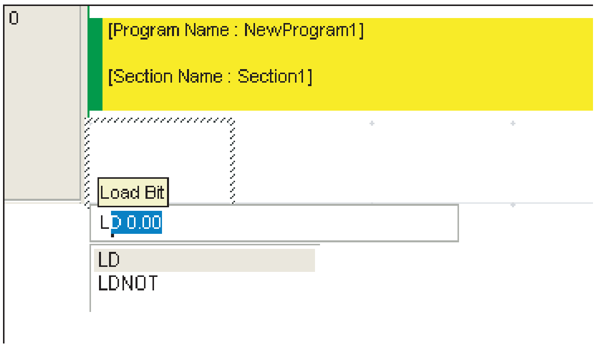
If "USB" is not displayed for the network type, refer to *CX-Programmer Ver.9. □ Operation Manual* (Cat. No. W446), and check that the USB driver has been installed correctly.

Entering NO and NC Input Conditions

- For a NO input condition using the LD instruction, press the **L** or **C** Key and select *LD*. For an OR input condition, press the **O** or **W** Key and select *OR*.
- For a NC input condition, press the **L** or / Key, and then select *LD NOT*. For an OR NOT input condition, press **O** or **X** and select *OR NOT*.
- Press the **Enter** Key, and then enter the address.

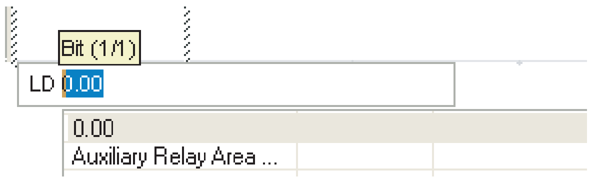
● Inputting a NO Input Condition

1 Press either the **L** or **C** Key. “LD 0.00” will be displayed.

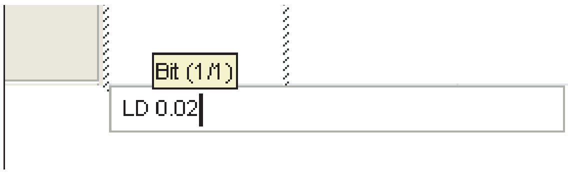


2 Press the **Enter** Key.

“Bit (1/1)” will be displayed and “0.00” will be displayed in reverse video.

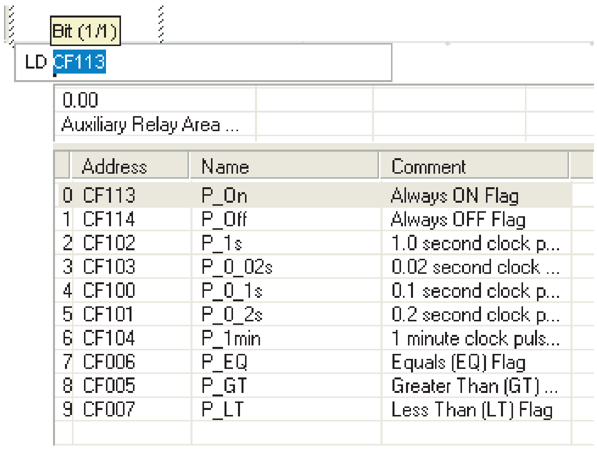


3 If the address is not CIO 0.00, input the correct address from the keyboard. For example, input “0.02.”



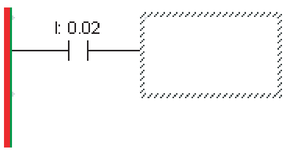
To select an Auxiliary Area bit*, press the Down Cursor Key to move the cursor to the Auxiliary Area List, press the Enter Key, and then select a bit from the list.

* Condition Flag or previously registered Auxiliary Area bit.



4 Press the **Enter** Key.

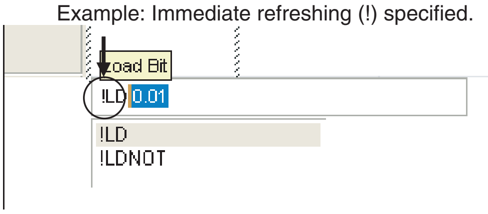
This completes inputting the LD instruction.



Additional Information

- The following instruction variations can be input.
  - Upward differentiation (@)
  - Downward differentiation (%)
  - Immediate refreshing (!)

Example: Immediate refreshing (!) specified.



The symbols indicating these instruction variations will be added to the beginning of the instruction whenever they are input regardless of whether the cursor is before (example: |LD), in the middle (example:L|D), or at the end (example: LD|) of the instruction.

- After an instruction has been entered, the variation can be changed as follows.
  - @: Upward differentiation
  - %: Downward differentiation
  - !: Immediate refreshing
  - Shift + 0: No differentiation

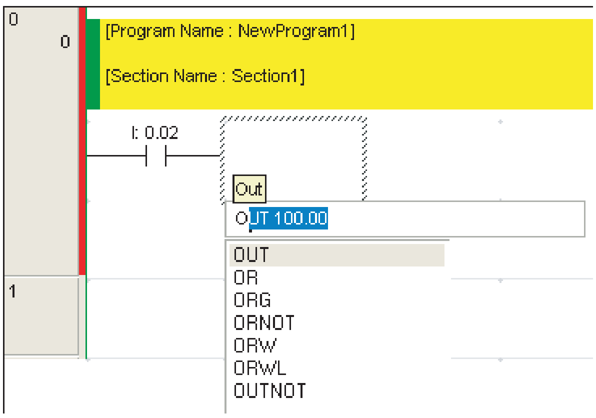
Inputting an OUTPUT Instruction

- To input an OUTPUT instruction, press the **O** Key and select *OUT.*
- To input an OUTPUT NOT instruction, press the **O** or **Q** Key, and then select *OUT NOT.*
- Press the **Enter** Key, and then enter the address.

● Input Example

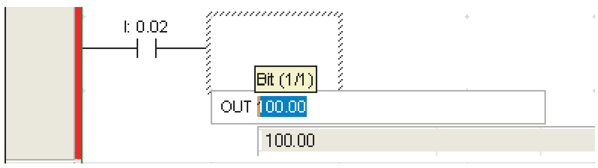
1 Press the **O** Key.

"OUT 100.00" will be displayed.

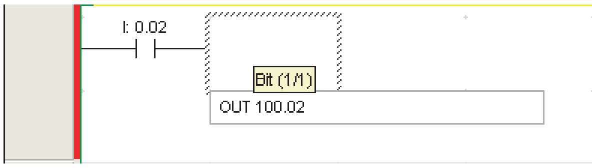


2 Press the **Enter** Key.

An OUTPUT instruction will be displayed with "100.00" in reverse video.

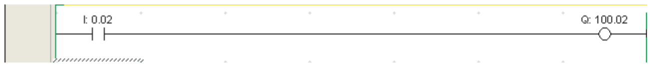


3 For an address other than CIO 100.00, input the address from the keyboard. Here, “100.02” has been input.



4 Press the **Enter** Key.

This completes inputting the OUTPUT instruction.



Inputting Instructions

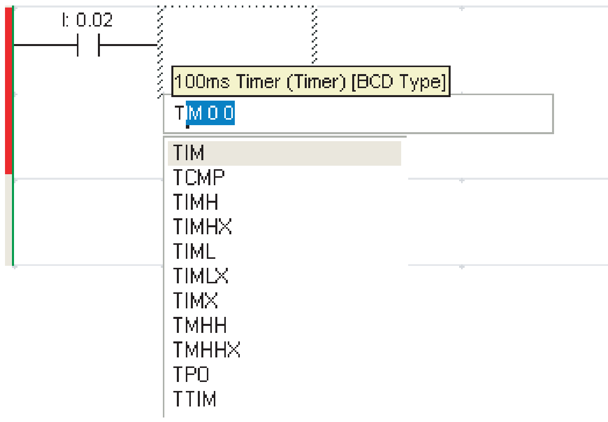
A mnemonic can be entered directly as a character string.

When you enter the first letter, a list of candidate mnemonics will be displayed. Use the **Up Cursor** and **Down Cursor** Keys to move up and down through this list, and then press the **Enter** Key to make a selection. Then, input the operands.

● Example: TIM Instruction

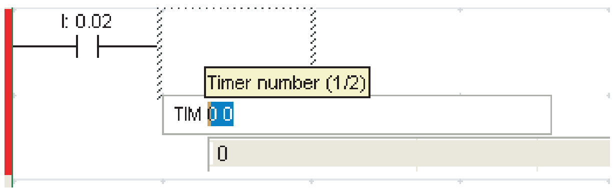
1 Press the **T** Key.

A list of instructions beginning with T will be displayed.



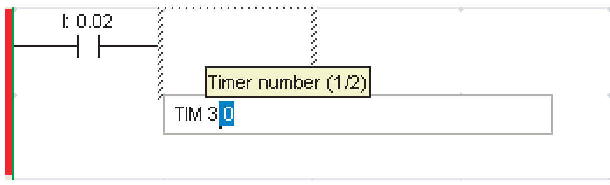
2 Press the **Enter** Key.

“Timer number (1/2)” will be displayed, and “0” will be displayed in reverse video.



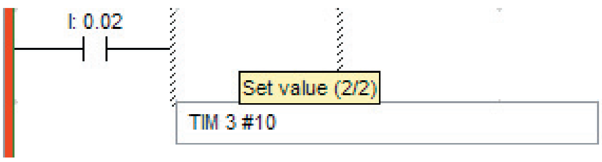
3 Input the timer number.

For example, input “3” and then press the **Enter** Key.



4 Input the timer set value.

For example, input “#10.”



5 Press the **Enter** Key.

This completes inputting the TIM instruction.

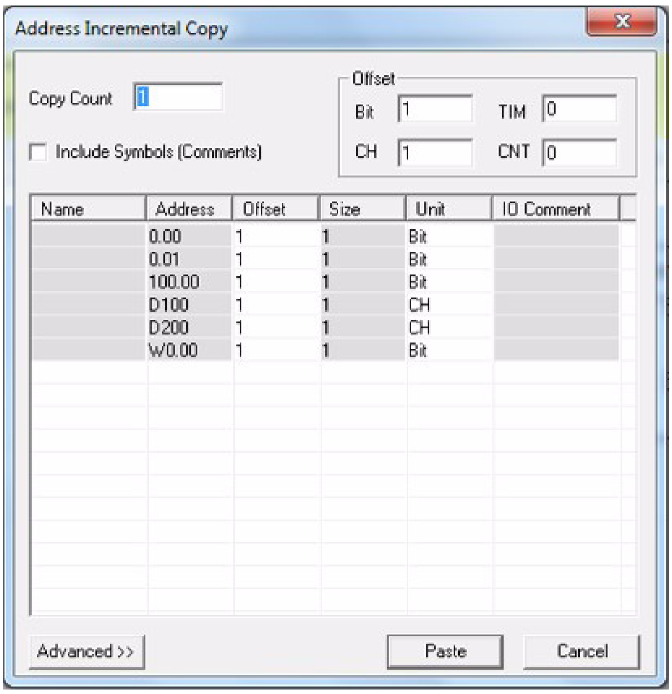
Copying Rungs Using the Automatic Address Increment Function

When rungs are copied and then pasted, it is possible to automatically increment the addresses by the specified number when pasting the rungs.

Example: When the following rung is copied, the bit addresses can be incremented by +16, and the word address can be incremented by +10 when pasting the rung.

1 Select the above rung and then select ***Address Increment Copy*** from the Edit Menu.

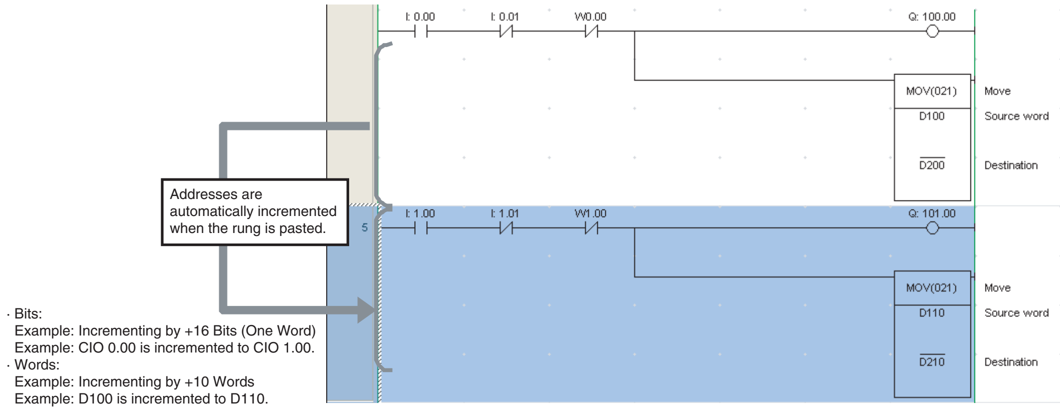
The following dialog box will be displayed.



2 In the Offset Area set the Bit Field to 16 and the CH Field to 10 for this example.

Click the **Paste** Button.

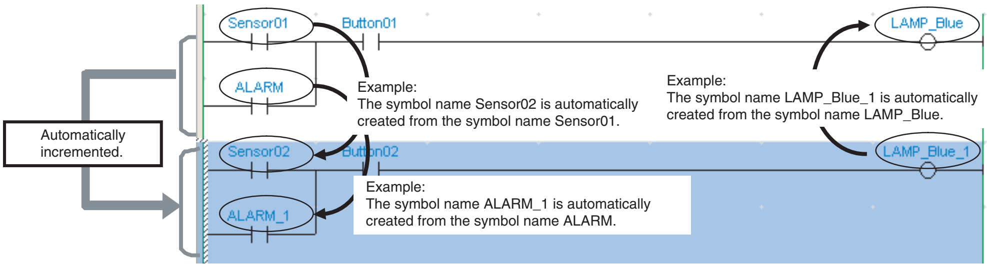
As shown below, the addresses are automatically incremented and the rung is pasted as the next rung.



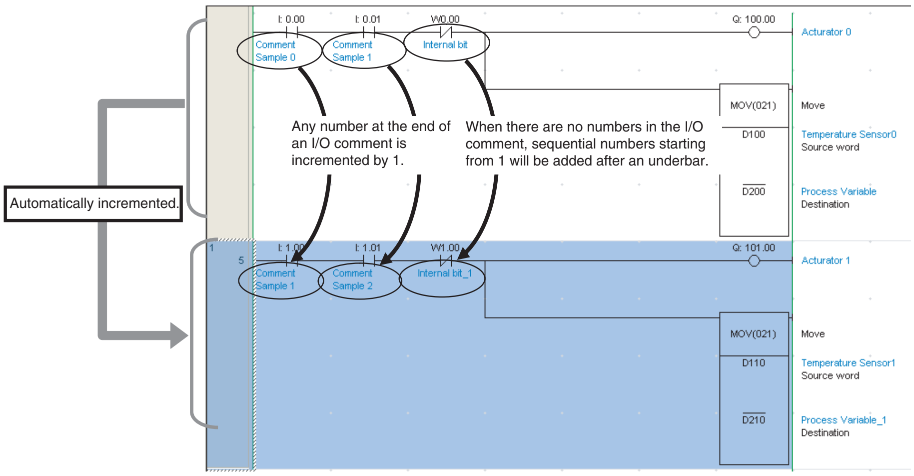
● Automatic Creation of Symbol Names and I/O Comments

If there are symbol names or I/O comments in the rung that was copied, executing the *Address Increment Copy* Command will automatically create symbol names and I/O comments.

- Automatic Symbol Name Creation



- Automatic I/O Comment Creation

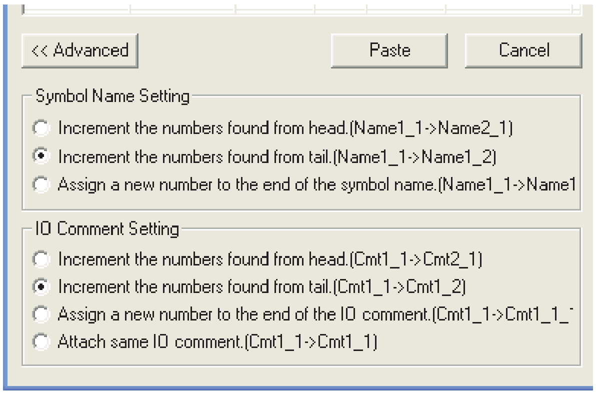


- Automatic Creation Rules
  By default, automatic creation is governed by the following rules.

Target	Automatic creation rule	Description
Symbol names	Increment the numbers found from tail.	The symbol name is searched for a number starting from the end, and any number that is found is incremented by 1. If no number is found, an underbar and a sequential number starting from 1 are appended.
I/O comments	Increment the numbers found from tail.	The I/O comment is searched for a number starting from the end, and any number that is found is incremented by 1. If no number is found, an underbar and a sequential number starting from 1 are appended.

Other rules may also be applicable.

Click the **Advanced** Button to select options. The options are enabled when the **Paste** Button is clicked.



Target	Automatic creation rule	Description
Symbol names	Increment the numbers found from head.	The symbol name is searched for a number starting from the beginning, and any number that is found is incremented by 1. If no number is found, an underbar and a sequential number starting from 1 are appended.
	Assign a new number to the end of the symbol name.	And underbar and sequential number, starting from 1, are appended to the end of the symbol name.
I/O comments	Increment the numbers found from head.	The I/O comment is searched for a number starting from the beginning, and any number that is found is incremented by 1. If no number is found, an underbar and a sequential number starting from 1 are appended.
	Assign a new number to the end of the I/O comment.	An underbar and a sequential number starting from 1 are appended to the end of the I/O comment.
	Attach same I/O comment.	The same I/O comment is used for the copy.

18-3-2 Saving and Reading Ladder Programs

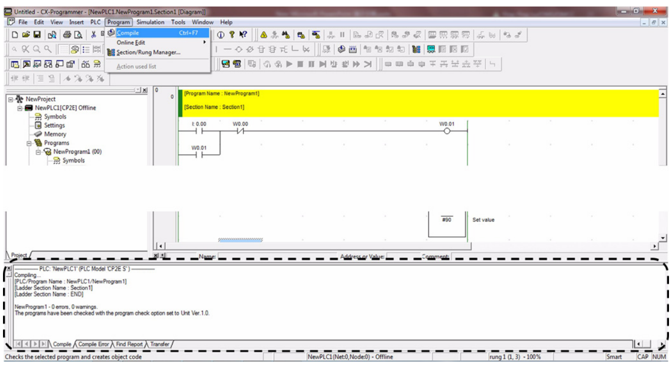
Always save the ladder program that you have created. This section describes how to check, save, and read a ladder program.

Checking a Ladder Program for Errors

You can check for errors in a program by compiling it.

1 Select ***Compile All PLC Programs*** from the Program Menu.

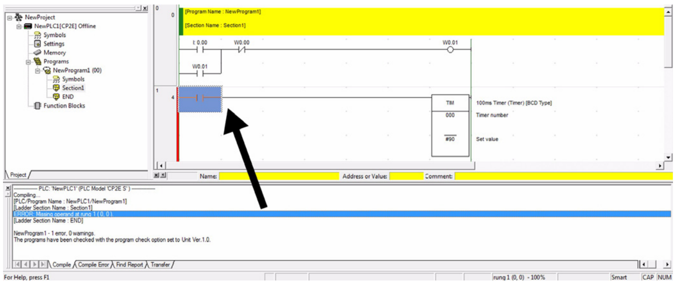
Compiling will start. Once compiling has been completed, the results of the program check will be displayed in the Output Window.



2 If an error was found, double-click the error message displayed in the Output Window.

The cursor will move to the location of the error. Correct the ladder program as required.

Note When there is more than one error, press the **Shift** + **J** keys to search for errors in order.

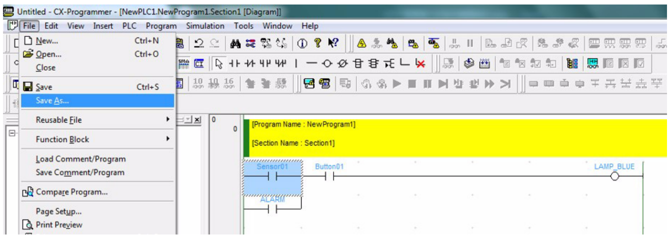


Saving a Ladder Program

Once created, a ladder program must be saved. Ladder programs are saved in projects.

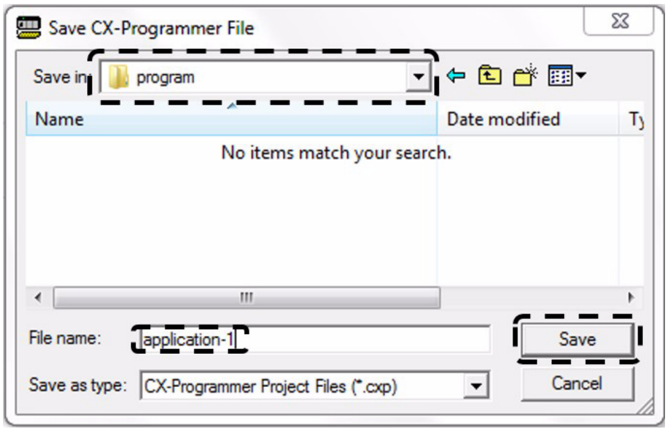
1 Select ***Save As*** from the File Menu.

The Save CX-Programmer File Dialog Box will be displayed.



2 Specify the save location, input the file name, and then click the **Save** Button.

The CX-Programmer project file will be saved.



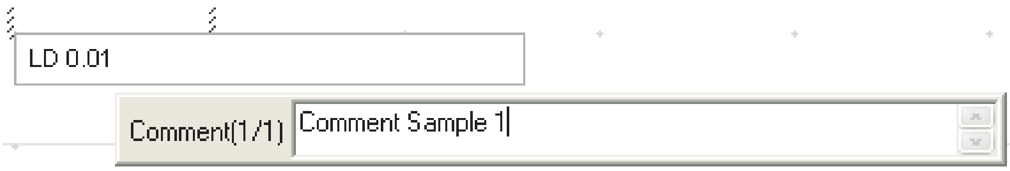
18-3-3 Editing Ladder Programs

A ladder program can be edited in the CX-Programmer. Also, I/O comments and rung comments can be input.

Inputting and Editing I/O Comments

● Inputting an I/O Comment with the Ladder Editor

In Smart Input Mode, an I/O comment can be input after an operand has been input using the Comment Dialog Box.

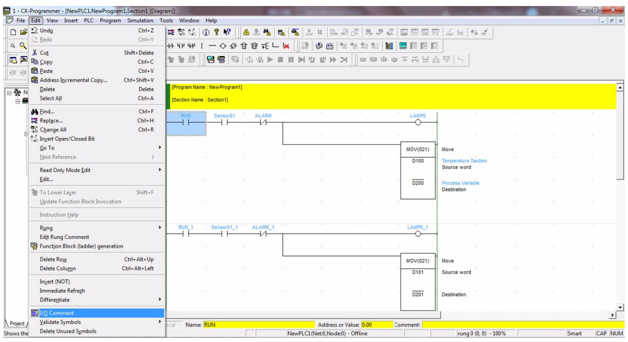


Note The Comment Dialog Box shown above is displayed only when the *Show with comment dialog* Option is selected on the Options - Diagrams Dialog Box. The Options - Diagrams Dialog Box is accessed by selecting ***Options*** from the Tools Menu.

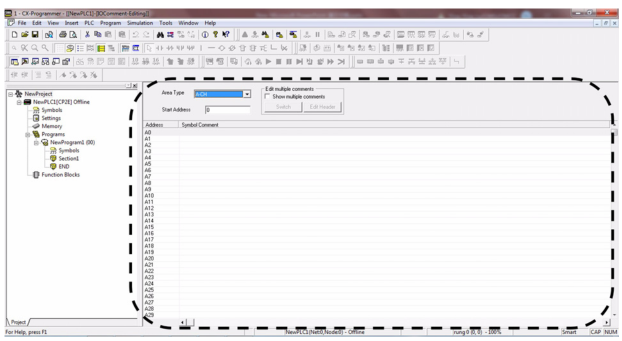
● Inputting by Editing I/O Comments

Multiple I/O comments can be input or changed from an address list.

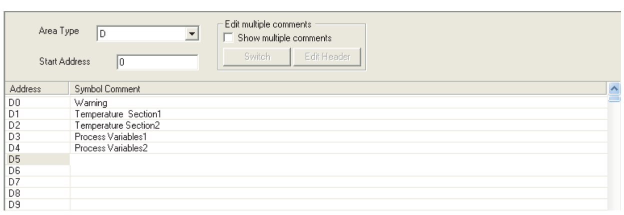
1 Select ***Edit I/O Comment*** from the Edit Menu.



The I/O Comment Editing Window will be displayed.



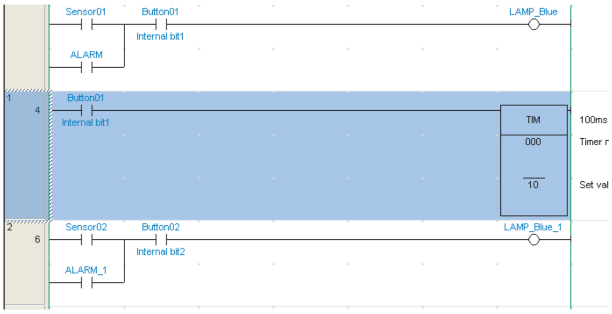
2 Input I/O comments or double-click the address for which the I/O comments are to be changed. Inputting the I/O comment will be enabled, so input the I/O comment.



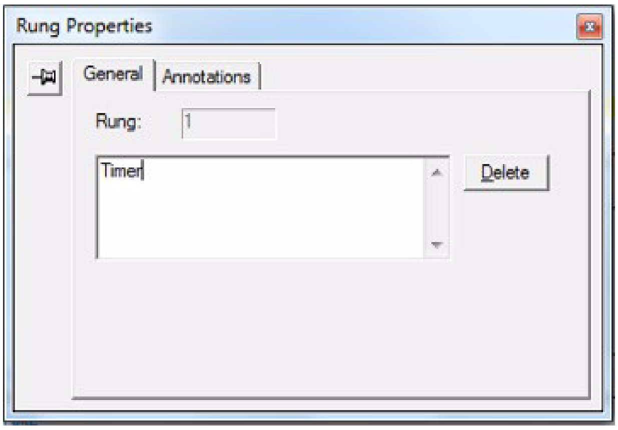
Inputting Rung Comments

Comments can be added to rungs in the program.

1 Double-click the header of the rung to which a comment is to be attached.
The Rung Properties Dialog Box will be displayed.

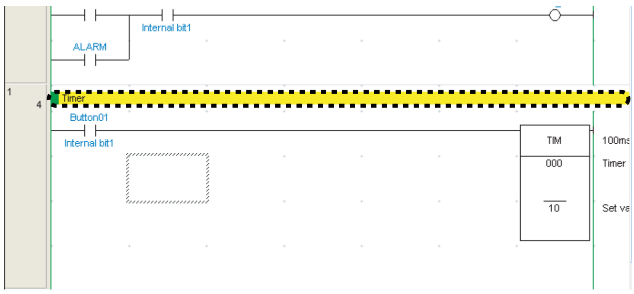


2 Input a comment into the Comment Field on the General Tab Page.



3 Close the Rung Properties Dialog Box.

The input rung comment will be displayed in the ladder program.



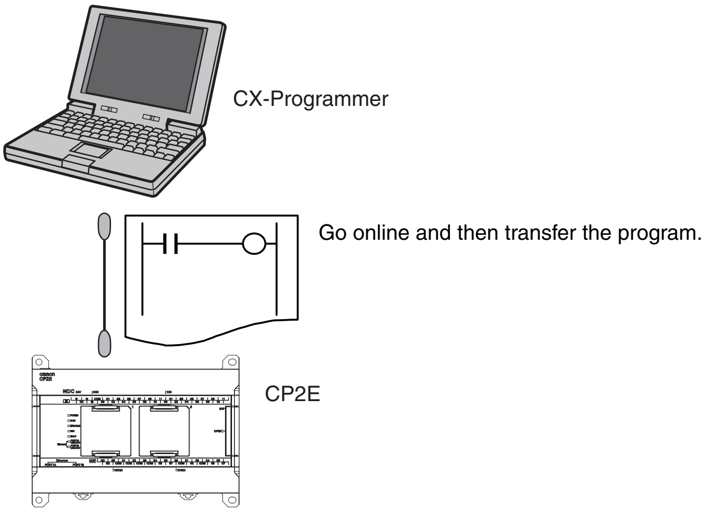
18-4 Connecting Online to the CP2E and Transferring the Program

This section describes how to make an online connection between the CX-Programmer and the CP2E, and then transfer a ladder program to the CP2E.

18-4-1 Connecting Online

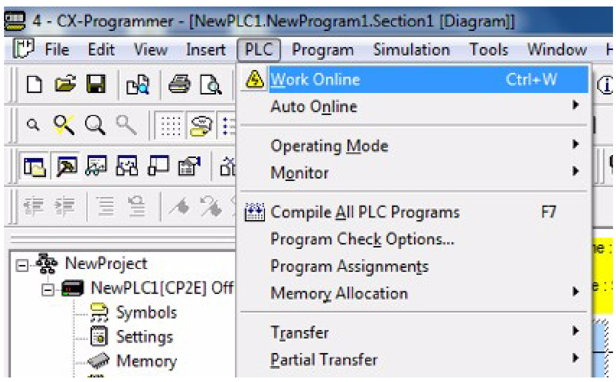
To enable transferring programs from the CX-Programmer to the CP2E, it is first necessary to place the CX-Programmer online with the CP2E.

Online is the state in which communications is possible between the computer and the CP2E.

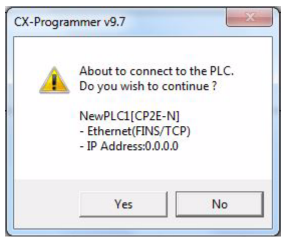


1 Open the project containing the program to be transferred from the CX-Programmer.

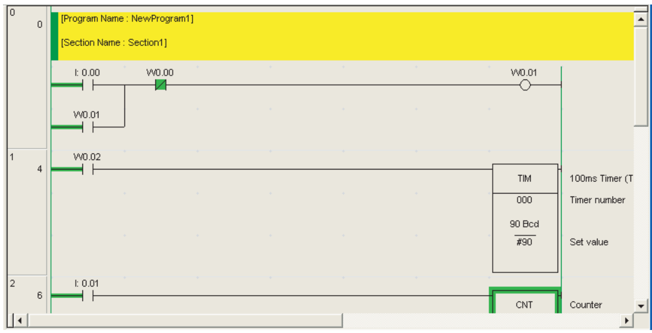
2 Select ***Work Online*** from the PLC Menu of the CX-Programmer.
A dialog box to confirm going online will be displayed.



3 Click the **Yes** Button.



Once the online connection has been established, the color of the Ladder Section Window will change to light gray.



Additional Information

Refer to *Section 4 Programming Device* in the *CP2E CPU Unit Hardware User's Manual* (Cat. No. W613) for the connection method with the CP2E.

If it is not possible to establish an online connection, check the PLC type setting and the communications settings. To check them, double-click *New PLC1 [CP2E] Offline* in the project tree. For details on these settings, refer to *Creating a New Project* in *18-3-1 Inputting a Ladder Program*.

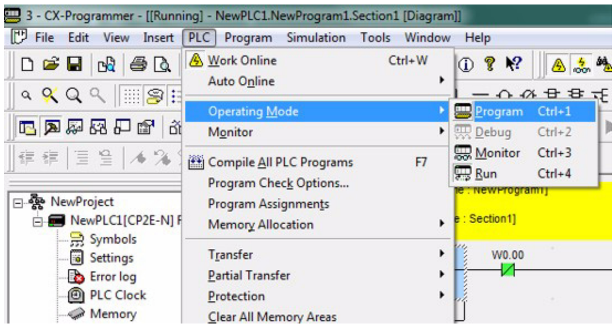
18-4-2 Changing Operating Modes

The operating mode can be changed to PROGRAM mode.

The procedure for changing to PROGRAM mode is given below.

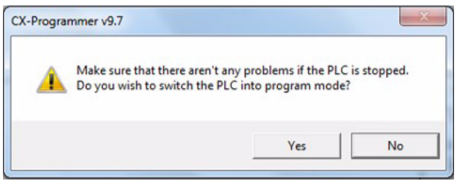
1 Select ***Operating Mode - Program*** from the PLC Menu.

A dialog box to confirm changing the operating mode will be displayed.

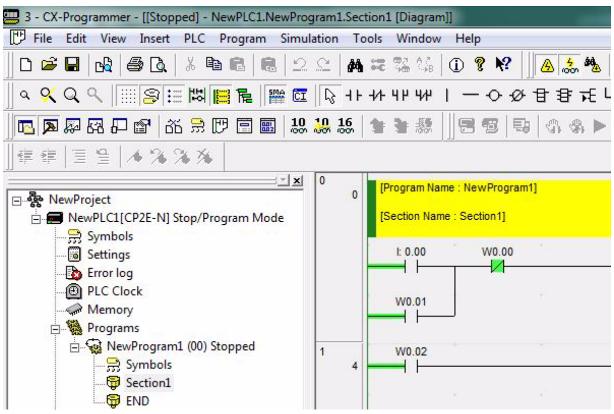


2 Click the **Yes** Button.

The operating mode will be changed.



The operating mode is displayed in the project tree.



Additional Information

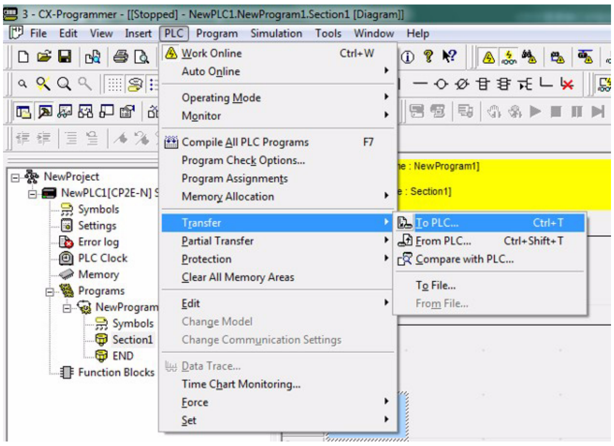
Change to PROGRAM mode before transferring the PLC Setup and ladder program.

18-4-3 Transferring a Ladder Program and the PLC Setup

A ladder program created with the CX-Programmer can be transferred to the CP2E.

1 Change to PROGRAM mode, select ***Operating Mode - Program*** from the PLC Menu, and then click the **Yes** Button.

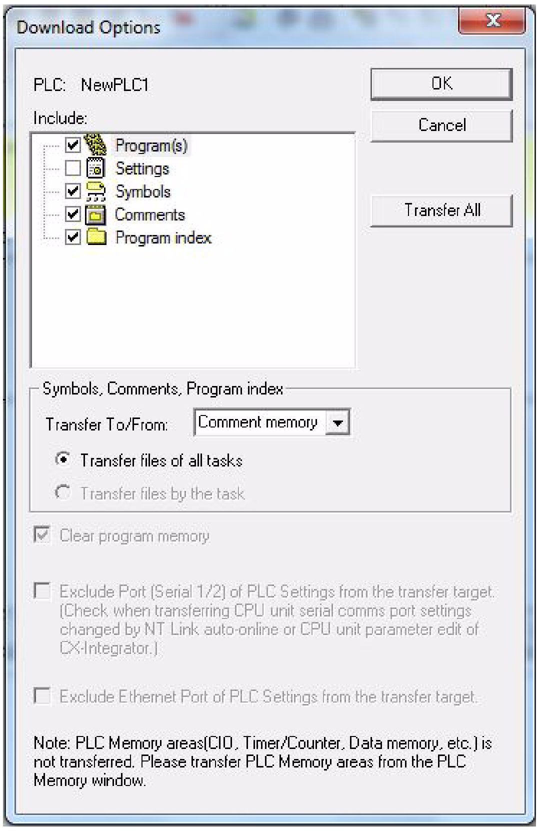
2 Select ***Transfer - Transfer [PC→PLC]*** from the PLC Menu. The Download Options Dialog Box will be displayed.



3 Click the **OK** Button.

A dialog box to confirm the transfer will be displayed.

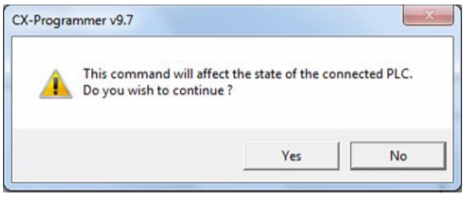
To transfer the PLC Setup, select the Settings Check Box.



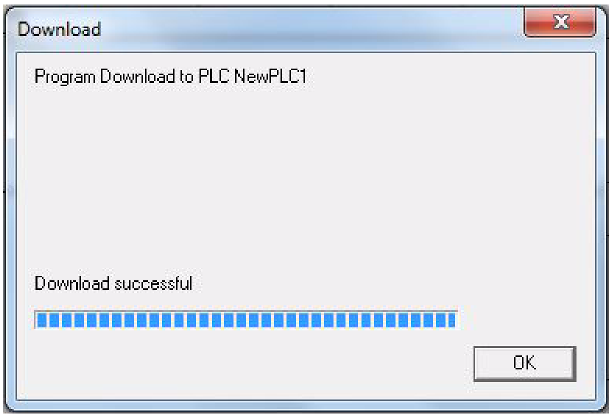
Additional Information

For details on the transfer options, refer to the CX-Programmer Online Help.

4 Click the **Yes** Button.



5 Click the **OK** Button.
This completes transferring the ladder program.



18-4-4 Starting Operation

To start operation, turn ON the power or change the operating mode to RUN mode.

Precautions for Correct Use

To start operation when the power is turned ON, set the operation mode at startup in the PLC Setup to MONITOR or RUN mode.

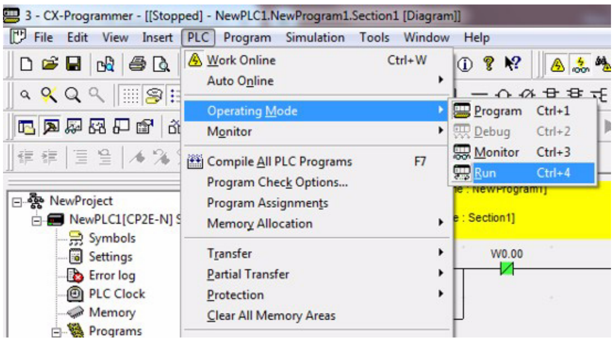
Use the following procedure to change the operating mode to RUN mode. To perform trial operation for debugging or adjustments, change the operating mode to MONITOR mode.

Precautions for Safe Use

Always confirm the safety of the controlled system before changing to MONITOR or RUN mode.

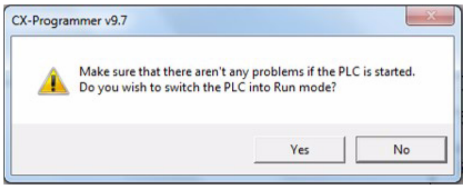
1 Select ***Operating Mode - Run*** from the PLC Menu.

A dialog box to confirm changing the operating mode will be displayed.



2 Click the **Yes** Button.

The CP2E will change to RUN mode, and operation will start.



Additional Information

PROGRAM mode cannot be changed to MONITOR or RUN mode when the user program, PLC Setup settings and DM area data in the CPU Units are being backed up. Change the operating mode after the backup is completed.

18-5 Online Monitoring and Debugging

This section describes how to use CX-Programmer to monitor and debug a ladder program.

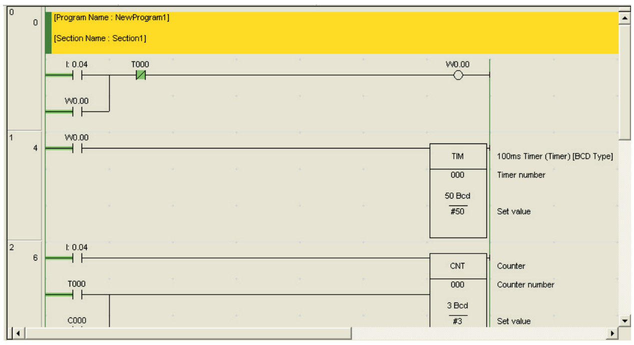
18-5-1 Monitoring Status

Displaying Execution Status

It is possible to display the execution status of a ladder program. This enables checking the execution of the ladder program.

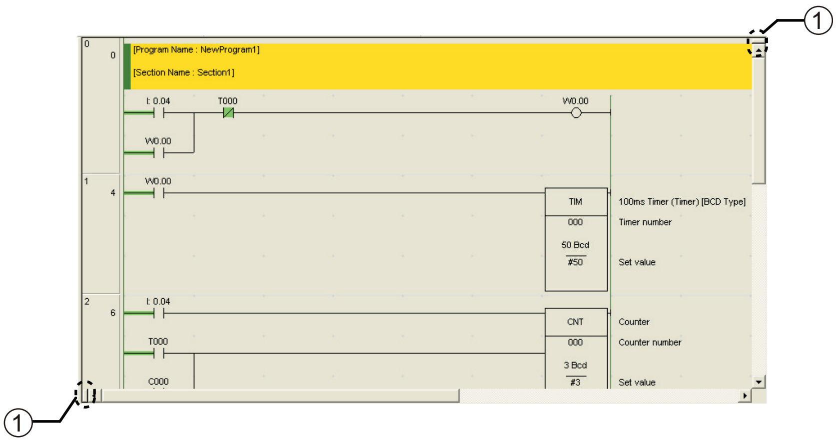
Change the CP2E's operating mode to MONITOR mode to display the execution status.

The execution status of the ladder program will be displayed.



Displaying the Execution Status for More than One Location

The Ladder Section Window can be split. This enables displaying multiple locations within a ladder program so that you can check them at the same time.



18-5 Online Monitoring and Debugging

18

18-5-1 Monitoring Status

● Window Frames

You can drag the frames in the window to display different views of the program in the Ladder Section Window. The window can be split into up to four sections.

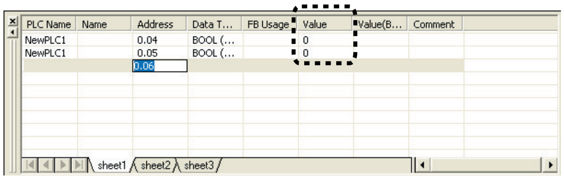
Monitoring Specified Addresses

You can specify addresses to check bit status and word contents.

1 While online, select ***Window - Watch Window*** from the View Menu.

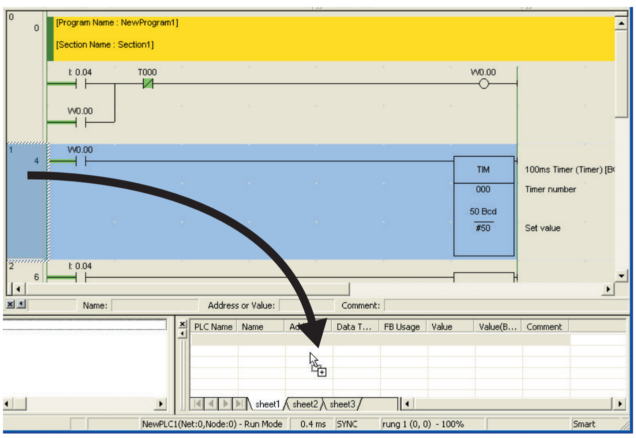
2 Input an address.

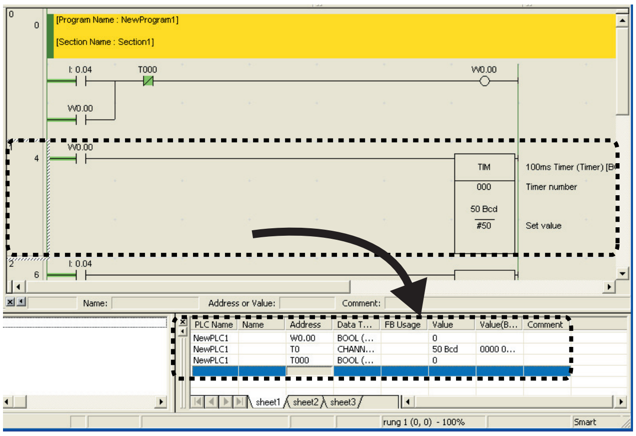
The bit status or word contents will be displayed. For BOOL data, 0 indicates OFF.



Additional Information

- When entering an address, place a period between the word address and bit number. For example, to input the address of bit 04 in CIO 0, input “0.04.”
- An address can be input by dragging it from the Ladder Section Window and dropping it into the Watch Window. By dragging and dropping the header of a rung, all of the addresses on that rung can be input.





18-5-2 Force-set/Reset Bits

Input bits can be controlled from CX-Programmer regardless of input status from the input devices. This is used to establish input and output conditions when performing trial operation, or to see the effect of establishing conditions when debugging.

Bits that can be Force-set/Reset

- I/O bits
- Word Area bits (W)
- Timer Completion Flags
- Holding Area Bits (H)
- Counter Completion Flags

Precautions for Safe Use

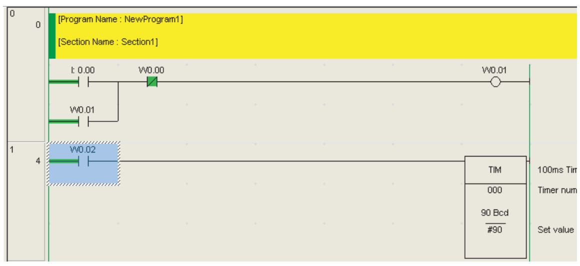
Always check the safety of the system before force-setting or force-resetting a bit and before releasing forced status.

Force-setting

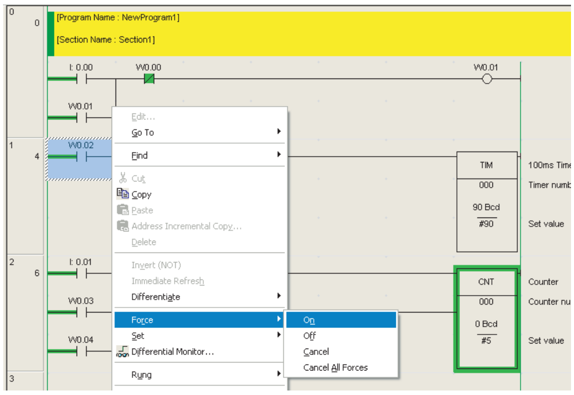
Force-setting a bit.

1 Set the CP2E operating mode to either MONITOR or PROGRAM mode.

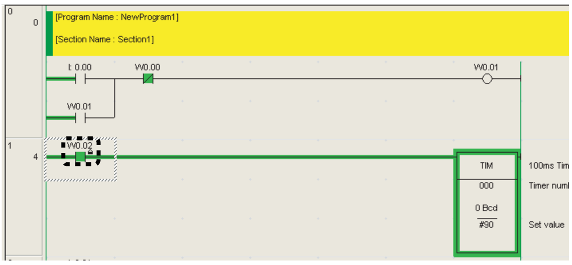
2 Move the cursor to an input condition for the input bit that is to be force-set.



3 Right-click and select ***Force - On***.



The input bit will be force-set. A symbol indicating the force-set status will be displayed at the input condition.



Additional Information

- Select ***On*** to turn ON a bit and ***Off*** to turn OFF a bit.
- To cancel forced status, select ***Cancel***.

18-5-3 Online Editing

About Online Editing

A ladder program running on the CP2E can be edited online.

This can be done while the CP2E is in MONITOR mode or PROGRAM mode.

Using the CX-Programmer, it is possible to either change part of a ladder program running on the CP2E , or make an addition to the program.

Online editing is used to make minor changes to the ladder program without actually stopping the operation of the CP2E.

Precautions for Correct Use

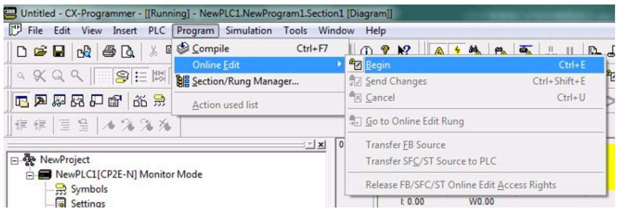
- When a ladder program has been changed using online editing, the cycle time may increase by one or more cycles, or it may not be possible to read an input signal.
- Major changes, such as moving a rung, copying, inserting, or deleting, should be done offline and then the program should be transferred to the CP2E again.
- After completing online editing, the results of editing are backed up to backup memory, resulting in a longer cycle time. While this is being done, the BKUP indicator will be lit, and the CX-Programmer will indicate the progress.
- An increase of one cycle will be 3.5ms maximum daring online editing and 8% of cycle time during back up.
- Do not turn OFF the power supply to the CPU Unit after online editing, until the backup to the backup memory is completed (the BKUP indicator is OFF).

Online Editing Procedure

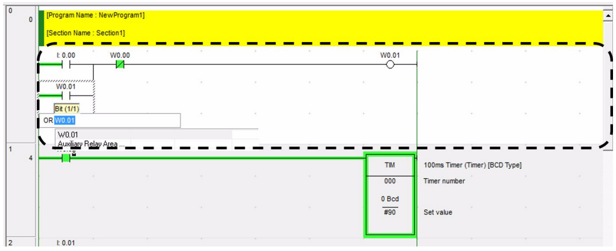
1 Change the CP2E's operating mode to MONITOR or PROGRAM mode.

2 Click the header of the rung to be edited.

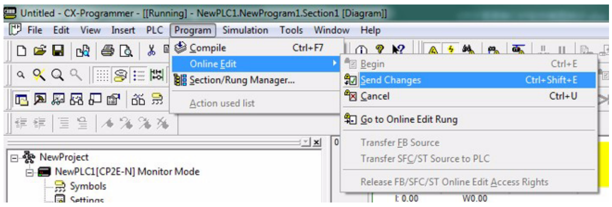
3 Select ***Online Edit - Begin*** from the PLC Menu.
The gray color will be cleared from the Ladder Section Window to indicate that the ladder program can be edited.



4 Edit the ladder program.



5 Select ***Online Edit - Send Changes*** from the PLC Menu.
The edited rung will be transferred to the CP2E.



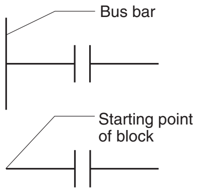
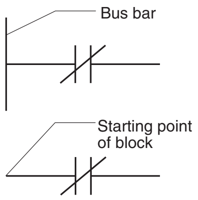
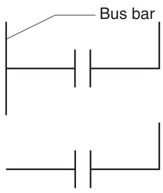
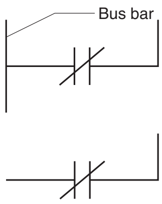
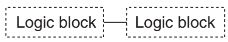
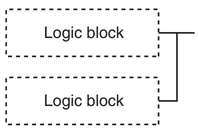
App

Appendices

A-1 Instruction Functions

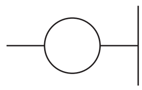
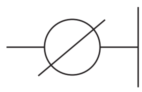
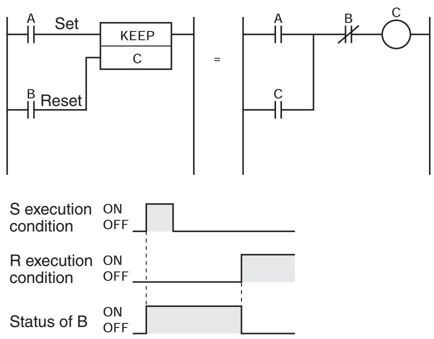
The CP2E CPU Units support the following instructions.
Refer to the *CP1E/CP2E CPU Unit Instructions Reference Manual* (Cat. No. W483) for details.

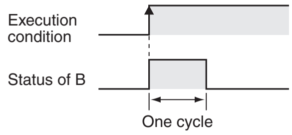
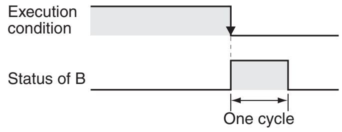
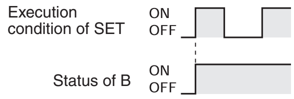
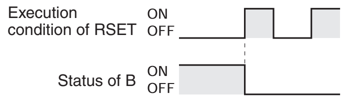
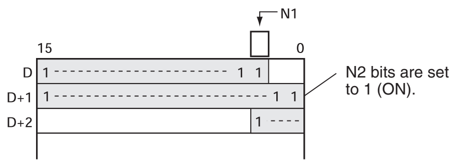
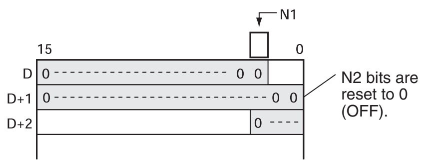
A-1-1 Sequence Input Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
LOAD	LD	@/%/!/!@/!%		Indicates a logical start and creates an ON/OFF execution condition based on the ON/OFF status of the specified operand bit.
LOAD NOT	LD NOT	@/%/!/!@/!%		Indicates a logical start and creates an ON/OFF execution condition based on the reverse of the ON/OFF status of the specified operand bit.
AND	AND	@/%/!/!@/!%	Symbol 	Takes a logical AND of the status of the specified operand bit and the current execution condition.
AND NOT	AND NOT	@/%/!/!@/!%	Symbol 	Reverses the status of the specified operand bit and takes a logical AND with the current execution condition.
OR	OR	@/%/!/!@/!%		Takes a logical OR of the ON/OFF status of the specified operand bit and the current execution condition.
OR NOT	OR NOT	@/%/!/!@/!%		Reverses the status of the specified bit and takes a logical OR with the current execution condition.
AND LOAD	AND LD	---		Takes a logical AND between logic blocks. LD to — Logic block A LD to — Logic block B AND LD ······ Serial connection between logic block A and logic block B.
OR LOAD	OR LD	---		Takes a logical OR between logic blocks. LD to — Logic block A LD to — Logic block B OR LD ····· Parallel connection between logic block A and logic block B.

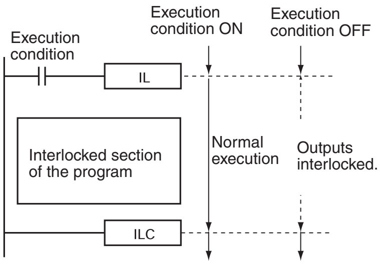
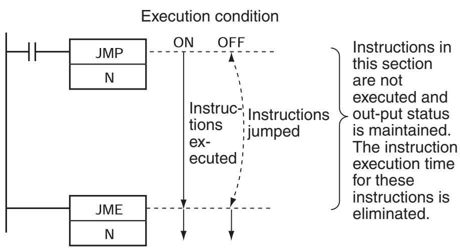
Instruction	Mnemonic	Variations	Symbol/Operand	Function
NOT	NOT	---		Reverses the execution condition.
CONDITION ON	UP	---	UP(521)	UP(521) turns ON the execution condition for one cycle when the execution condition goes from OFF to ON.
CONDITION OFF	DOWN	---	DOWN(522)	DOWN(522) turns ON the execution condition for one cycle when the execution condition goes from ON to OFF.
LOAD BIT TEST	LD TST	---	TST(350) S N **S**: Source word **N**: Bit number	LD TST(350) is used in the program like LD; the execution condition is ON when the specified bit in the specified word is ON, and OFF when the bit is OFF.
LOAD BIT TEST NOT	LD TSTN	---	TSTN(351) S N **S**: Source word **N**: Bit number	LD TSTN(351) is used in the program like LD NOT; the execution condition is OFF when the specified bit in the specified word is ON, and ON when the bit is OFF.
AND BIT TEST	AND TST	---	AND TST(350) S N **S**: Source word **N**: Bit number	AND TST(350) is used in the program like AND; the execution condition is ON when the specified bit in the specified word is ON, and OFF when the bit is OFF.
AND BIT TEST NOT	AND TSTN	---	AND TSTN(351) S N **S**: Source word **N**: Bit number	AND TSTN(351) is used in the program like AND NOT; the execution condition is OFF when the specified bit in the specified word is ON, and ON when the bit is OFF.
OR BIT TEST	OR TST	---	TST(350) S N **S**: Source word **N**: Bit number	OR TST(350) is used in the program like OR; the execution condition is ON when the specified bit in the specified word is ON, and OFF when the bit is OFF.
OR BIT TEST NOT	OR TSTN	---	TSTN(351) S N **S**: Source word **N**: Bit number	OR TSTN(351) is used in the program like OR NOT; the execution condition is OFF when the specified bit in the specified word is ON, and ON when the bit is OFF.

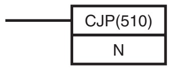
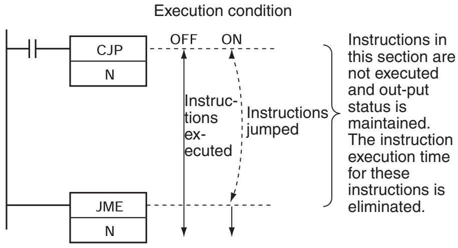
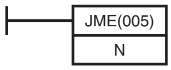
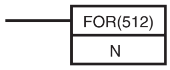
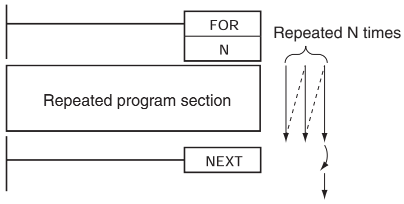
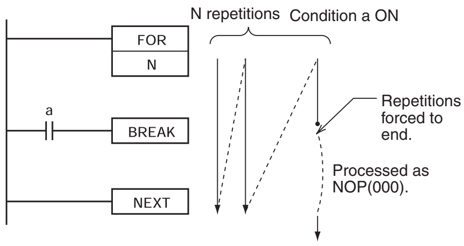
A-1-2 Sequence Output Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
OUTPUT	OUT	!		Outputs the result (execution condition) of the logical processing to the specified bit.
OUTPUT NOT	OUT NOT	!		Reverses the result (execution condition) of the logical processing, and outputs it to the specified bit.
KEEP	KEEP	!	S (Set) R (Reset) KEEP(011) B **B**: Bit	Operates as a latching relay. 

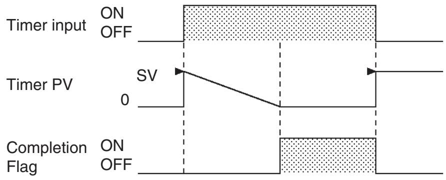
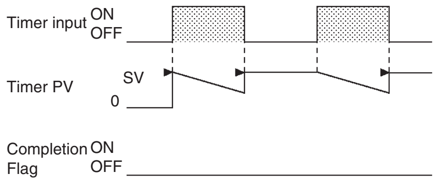
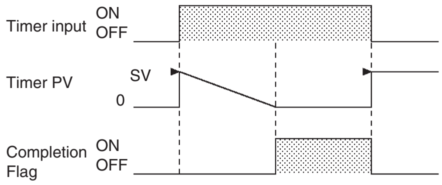
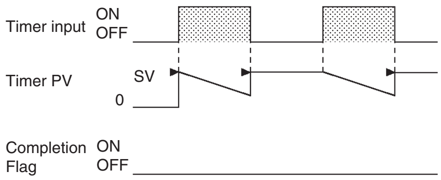
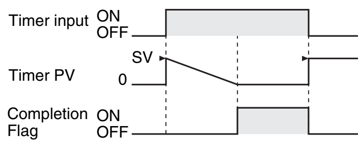
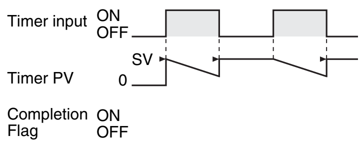
Instruction	Mnemonic	Variations	Symbol/Operand	Function
DIFFERENTI-ATE UP	DIFU	!	DIFU(013) B **B**: Bit	DIFU(013) turns the designated bit ON for one cycle when the execution condition goes from OFF to ON (rising edge). 
DIFFERENTI-ATE DOWN	DIFD	!	DIFD(014) B **B**: Bit	DIFD(014) turns the designated bit ON for one cycle when the execution condition goes from ON to OFF (falling edge). 
SET	SET	@/%/!/!@/!%	SET B **B**: Bit	SET turns the operand bit ON when the execution condition is ON. 
RESET	RSET	@/%/!/!@/!%	RSET B **B**: Bit	RSET turns the operand bit OFF when the execution condition is ON. 
MULTIPLE BIT SET	SETA	@	SETA(530) D N1 N2 **D**: Beginning word **N1**: Beginning bit **N2**: Number of bits	SETA(530) turns ON the specified number of consecutive bits. 
MULTIPLE BIT RESET	RSTA	@	RSTA(531) D N1 N2 **D**: Beginning word **N1**: Beginning bit **N2**: Number of bits	RSTA(531) turns OFF the specified number of consecutive bits. 
SINGLE BIT SET	SETB	@/!/!@	SETB(532) D N **D**: Word address **N**: Bit number	SETB(532) turns ON the specified bit in the specified word when the execution condition is ON. Unlike the SET instruction, SETB(532) can be used to reset a bit in a DM word.
SINGLE BIT RESET	RSTB	@/!/!@	RSTB(533) D N **D**: Word address **N**: Bit number	RSTB(533) turns OFF the specified bit in the specified word when the execution condition is ON. Unlike the RSET instruction, RSTB(533) can be used to reset a bit in a DM word.

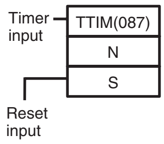
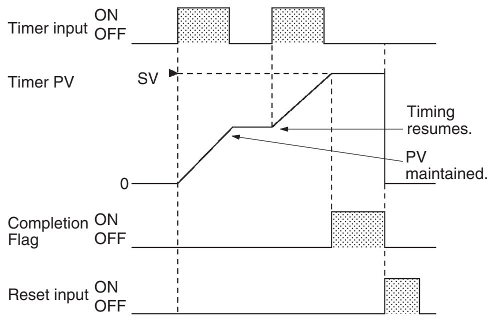
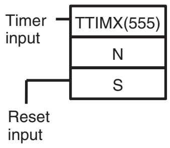
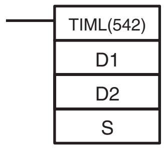
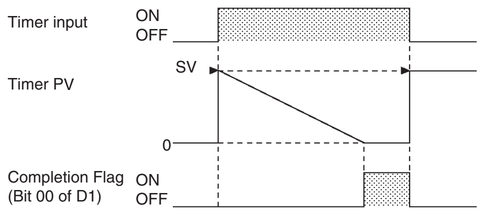
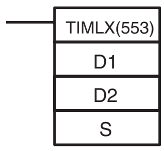
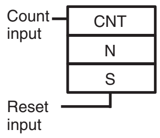
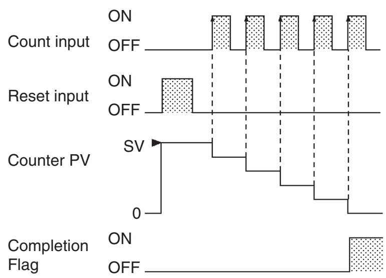
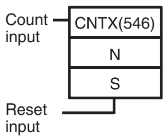
A-1-3 Sequence Control Instructions

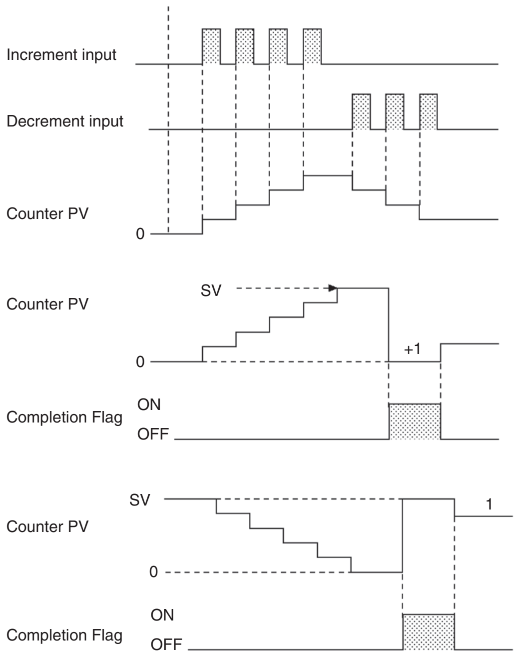
Instruction	Mnemonic	Variations	Symbol/Operand	Function
END	END	---	END(001)	Indicates the end of a program.
NO OPERATION	NOP	---	---	This instruction has no function. (No processing is performed for NOP(000).)
INTERLOCK	IL	---	IL(002)	Interlocks all outputs between IL(002) and ILC(003) when the execution condition for IL(002) is OFF. IL(002) and ILC(003) are normally used in pairs. 
INTERLOCK CLEAR	ILC	---	ILC(003)	All outputs between IL(002) and ILC(003) are interlocked when the execution condition for IL(002) is OFF. IL(002) and ILC(003) are normally used in pairs.
MULTI-INTERLOCK DIFFERENTIATION HOLD	MILH	---	MILH (517) N D **N**: Interlock number **D**: Interlock Status Bit	When the execution condition for MILH(517) is OFF, the outputs for all instructions between that MILH(517) instruction and the next MILC(519) instruction are interlocked. MILH(517) and MILC(519) are used as a pair. MILH(517)/MILC(519) interlocks can be nested (e.g., MILH(517)-MILH(517)-MILC(519)-MILC(519)). If there is a differentiated instruction (DIFU, DIFD, or instruction with a @ or % prefix) between MILH(517) and the corresponding MILC(519), that instruction will be executed after the interlock is cleared if the differentiation condition of the instruction was established.
MULTI-INTERLOCK DIFFERENTIATION RELEASE	MILR	---	MILR (518) N D **N**: Interlock number **D**: Interlock Status Bit	When the execution condition for MILR(518) is OFF, the outputs for all instructions between that MILR(518) instruction and the next MILC(519) instruction are interlocked.MILR(518) and MILC(519) are used as a pair. MILR(518)/MILC(519) interlocks can be nested (e.g., MILR(518)-MILR(518)-MILC(519)-MILC(519)). If there is a differentiated instruction (DIFU, DIFD, or instruction with a @ or % prefix) between MILR(518) and the corresponding MILC(519), that instruction will not be executed after the interlock is cleared even if the differentiation condition of the instruction was established.
MULTI-INTERLOCK CLEAR	MILC	---	MILC (519) N **N**: Interlock number	Clears an interlock started by an MILH(517) or MILR(518) with the same interlock number. All outputs between MILH(517)/MILR(518) and the corresponding MILC(519) with the same interlock number are interlocked when the execution condition for MILH(517)/MILR(518) is OFF.
JUMP	JMP	---	JMP(004) N **N**: Interlock number	When the execution condition for JMP(004) is OFF, program execution jumps directly to the first JME(005) in the program with the same jump number. JMP(004) and JME(005) are used in pairs. 

Instruction	Mnemonic	Variations	Symbol/Operand	Function
CONDITIONAL JUMP	CJP	---	 **N**: Interlock number	The operation of CJP(510) is the basically the opposite of JMP(004). When the execution condition for CJP(510) is ON, program execution jumps directly to the first JME(005) in the program with the same jump number. CJP(510) and JME(005) are used in pairs. 
JUMP END	JME	---	 **N**: Interlock number	Indicates the destination of a jump instruction.
FOR-NEXT LOOPS	FOR	---	 **N**: Number of loops	The instructions between FOR(512) and NEXT(513) are repeated a specified number of times. FOR(512) and NEXT(513) are used in pairs. 
BREAK LOOP	BREAK	---		Programmed in a FOR-NEXT loop to cancel the execution of the loop for a given execution condition. The remaining instructions in the loop are processed as NOP(000) instructions. 
FOR-NEXT LOOPS	NEXT	---		The instructions between FOR(512) and NEXT(513) are repeated a specified number of times. FOR(512) and NEXT(513) are used in pairs.

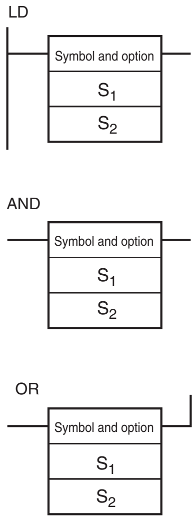
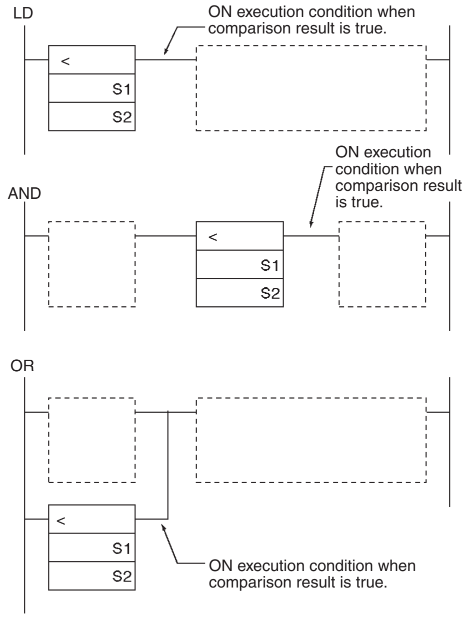
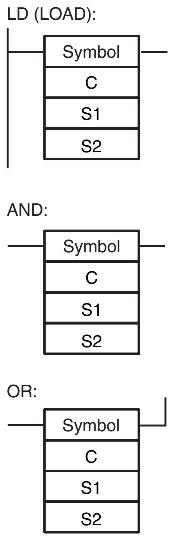
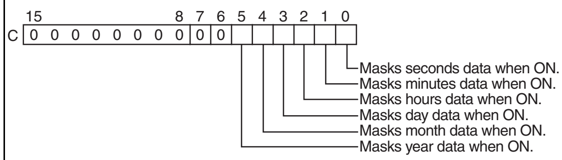
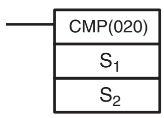
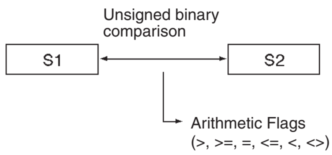
A-1-4 Timer and Counter Instructions

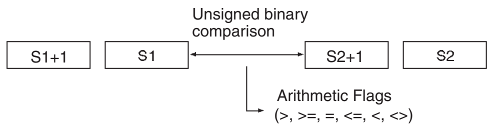
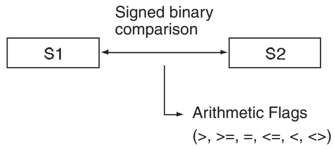
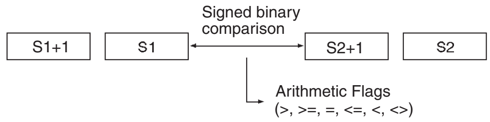
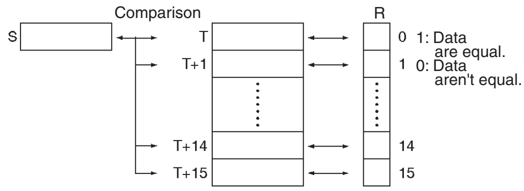
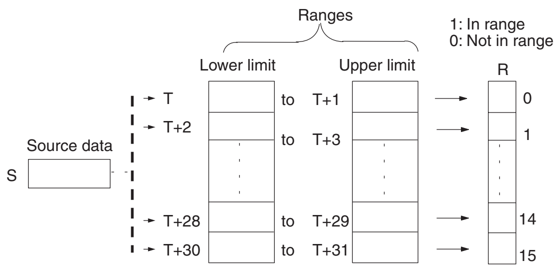
Instruction	Mnemonic	Variations	Symbol/Operand	Function
HUNDRED-MS TIMER	TIM (BCD)	---	TIM N S **N**: Timer number **S**: Set value	TIM/TIMX(550) operates a decrementing timer with units of 0.1-s. The setting range for the set value (SV) is 0 to 999.9 s for TIM(BCD) and 0 to 6,553.5 s for TIMX(Binary).  Timer Input Turned OFF before Completion Flag Turns ON 
	TIMX (Binary)	---	TIMX(550) N S **N**: Timer number **S**: Set value	
TEN-MS TIMER	TIMH (BCD)	---	TIMH(015) N S **N**: Timer number **S**: Set value	TIMH(015)/TIMHX(551) operates a decrementing timer with units of 10-ms. The setting range for the set value (SV) is 0 to 99.99 s for TIMH(BCD) and 0 to 655.35 s for TIMHX(Binary).  Timer Input Turned OFF before Completion Flag Turns ON 
	TIMHX (Binary)	---	TIMHX(551) N S **N**: Timer number **S**: Set value	
ONE-MS TIMER	TMHH (BCD)	---	TMHH(540) N S **N**: Timer number **S**: Set value	TMHH(540)/TMHHX(552) operates a decrementing timer with units of 1-ms. The setting range for the set value (SV) is 0 to 9.999 s for TMHH(BCD) and 0 to 65.535 s for TMHHX(Binary). The timing charts for TMHH(540) and TMHHX(552) are the same as those given above for TIMH(015).  Timer Input Turned OFF before Completion Flag Turns ON 
	TMHHX (BCD)	---	TMHHX(552) N S **N**: Timer number **S**: Set value	

Instruction	Mnemonic	Variations	Symbol/Operand	Function
ACCUMULA-TIVE TIMER	TTIM (BCD)	---	 **N**: Timer number **S**: Set value	TTIM(087)/TTIMX(555) operates an incrementing timer with units of 0.1-s. The setting range for the set value (SV) is 0 to 999.9 s for TTIM(BCD) and 0 to 6,553.5 s for TTIMX(Binary). 
	TTIMX (Binary)	---	 **N**: Timer number **S**: Set value	
LONG TIMER	TIML (BCD)	---	 **D1**: Completion Flag **D2**: PV word **S**: SV word	TIML(542)/TIMLX(553) operates a decrementing timer with units of 0.1-s that can time up to approx. 115 days for TIML(BCD) and 49,710 days for TIMLX(Binary). 
	TIMLX (Binary)	---	 **D1**: Completion Flag **D2**: PV word **S**: SV word	
COUNTER	CNT (BCD)	---	 **N**: Counter number **S**: Set value	CNT/CNTX(546) operates a decrementing counter. The setting range for the set value (SV) is 0 to 9,999 for CNT(BCD) and 0 to 65,535 for CNTX(Binary). 
	CNTX (Binary)	---	 **N**: Counter number **S**: Set value	

Instruction	Mnemonic	Variations	Symbol/Operand	Function
REVERSIBLE COUNTER	CNTR (BCD)	---	Increment input, Decrement input, Reset input → CNTR(012) / N / S **N**: Counter number **S**: Set value	CNTR(012)/CNTRX(548) operates a reversible counter. 
	CNTRX (Binary)	---	Increment input, Decrement input, Reset input → CNTRX(548) / N / S **N**: Counter number **S**: Set value	
RESET TIMER/ COUNTER	CNR (BCD)	@	CNR(545) / N1 / N2 **N1**: 1st number in range **N2**: Last number in range	CNR(545)/CNRX(547) resets the timers or counters within the specified range of timer or counter numbers. Sets the set value (SV) to the maximum of #9999 for CNR(BCD) and #FFFF for CNRX(Binary).
	CNRX (Binary)	@	CNRX(547) / N1 / N2 **N1**: 1st number in range **N2**: Last number in range	

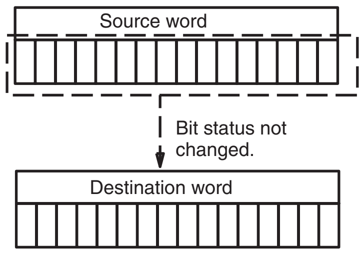
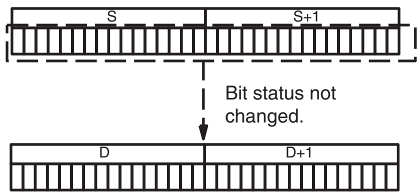
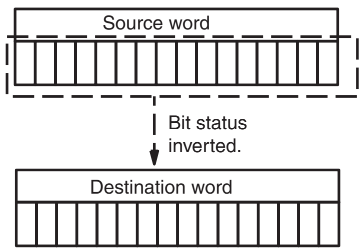
A-1-5 Comparison Instructions

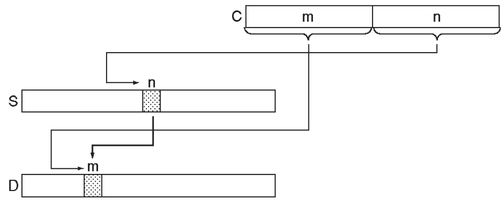
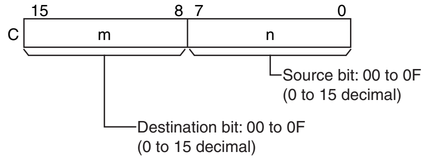
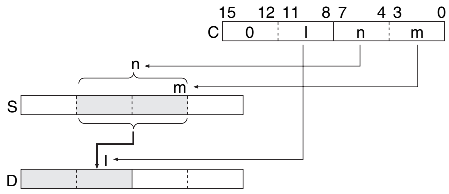
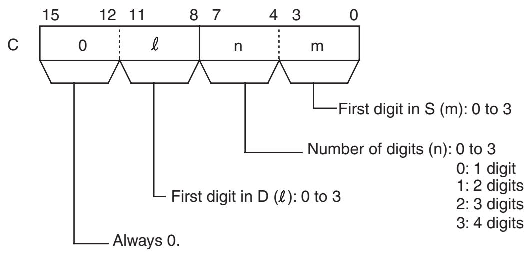
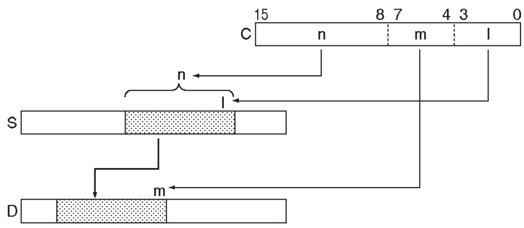
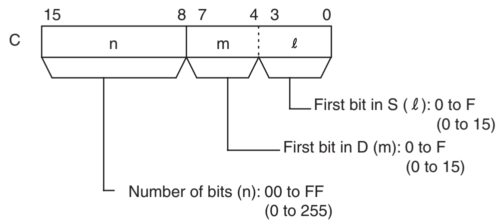
Instruction	Mnemonic	Variations	Symbol/Operand	Function
Symbol Comparison (Unsigned)	LD, AND, OR + =, <>, <, <=, >, >=	---	 **One-word** **S1**: Comparison data 1 **S2**: Comparison data 2 **Double length** **S1**: 1st word of comparison data 1 **S2**: 1st word of comparison data 2	Input comparison instructions compare two values (constants and/or the contents of specified words) and create an ON execution condition when the comparison condition is true. Input comparison instructions are available to compare signed or unsigned data of one-word or double length data. 
Symbol Comparison (Double-word, unsigned)	LD, AND, OR + =, <>, <, <=, >, >= + L	---		
Symbol Comparison (Signed)	LD, AND, OR + =, <>, <, <=, >, >= +S	---		
Symbol Comparison (Double-word, signed)	LD, AND, OR + =, <>, <, <=, >, >= +SL	---		
Time Comparison	LD, AND, OR + = DT, <> DT, < DT, <= DT, > DT, >= DT	---	 **C:** Control word **S1:** 1st word of present time **S2:** 1st word of comparison time	Time comparison instructions compare two BCD time values and create an ON execution condition when the comparison condition is true. There are three types of time comparison instructions, LD (LOAD), AND, and OR. Time values (year, month, day, hour, minute, and second) can be masked/unmasked in the comparison so it is easy to create calendar timer functions. C: Control Word Bits 00 to 05 of C specify whether or not the time data will be masked for the comparison. Bits 00 to 05 mask the seconds, minutes, hours, day, month, and year, respectively. If all 6 values are masked, the instruction will not be executed, the execution condition will be OFF, and the Error Flag will be turned ON. 
UNSIGNED COMPARE	CMP	!	 **S1**: Comparison data 1 **S2**: Comparison data 2	Compares two unsigned binary values (constants and/or the contents of specified words) and outputs the result to the Arithmetic Flags in the Auxiliary Area. 

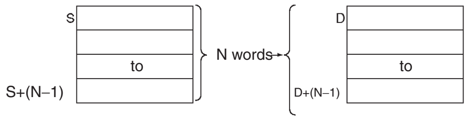
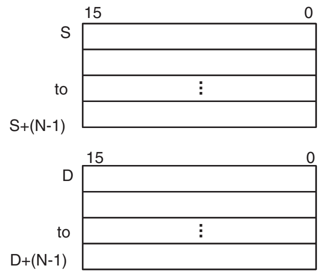
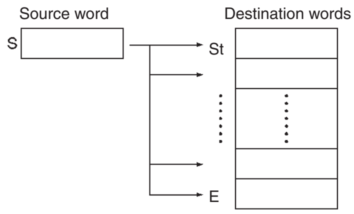
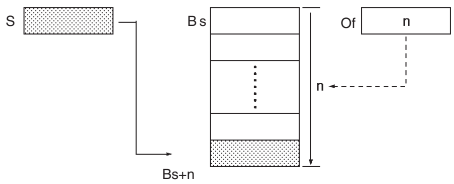
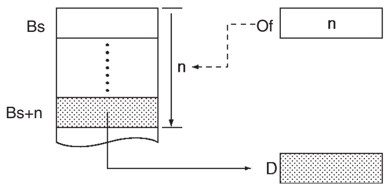
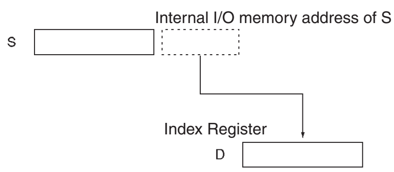
Instruction	Mnemonic	Variations	Symbol/Operand	Function
DOUBLE UNSIGNED COMPARE	CMPL	---	CMPL(060) S_1 S_2 **S1**: Comparison data 1 **S2**: Comparison data 2	Compares two double unsigned binary values (constants and/or the contents of specified words) and outputs the result to the Arithmetic Flags in the Auxiliary Area. 
SIGNED BINARY COMPARE	CPS	!	CPS(114) S_1 S_2 **S1**: Comparison data 1 **S2**: Comparison data 2	Compares two signed binary values (constants and/or the contents of specified words) and outputs the result to the Arithmetic Flags in the Auxiliary Area. 
DOUBLE SIGNED BINARY COMPARE	CPSL	---	CPSL(115) S_1 S_2 **S1**: Comparison data 1 **S2**: Comparison data 2	Compares two double signed binary values (constants and/or the contents of specified words) and outputs the result to the Arithmetic Flags in the Auxiliary Area. 
TABLE COMPARE	TCMP	@	TCMP(085) S T R **S**: Source data **T**: 1st word of table **R**: Result word	Compares the source data to the contents of 16 words and turns ON the corresponding bit in the result word when the contents are equal.  T: First word of table T: Comparison data 0 T+1: Comparison data 1 to: to T+15: Comparison data 15
UNSIGNED BLOCK COMPARE	BCMP	@	BCMP(068) S T R **S**: Source data **T**: 1st word of table **R**: Result word	Compares the source data to 16 ranges (defined by 16 lower limits and 16 upper limits) and turns ON the corresponding bit in the result word when the source data is within the range.  B: First word of block B: Lower limit value 0 B+1: Upper limit value 0 B+2: Lower limit value 1 B+3: Upper limit value 1 to: to B+30: Lower limit value 15 B+31: Upper limit value 15

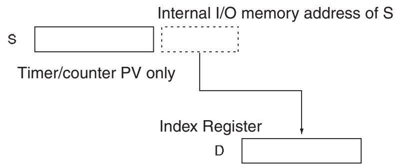
Instruction	Mnemonic	Variations	Symbol/Operand	Function
AREA RANGE COMPARE	ZCP	---	ZCP(088) CD LL UL **CD**: Comparison data (1 word) **LL**: Lower limit of range **UL**: Upper limit of range	Compares the 16-bit unsigned binary value in CD (word contents or constant) to the range defined by LL and UL and outputs the results to the Arithmetic Flags in the Auxiliary Area.
DOUBLE AREA RANGE COMPARE	ZCPL	---	ZCPL(116) CD LL UL **CD**: Comparison data (2 words) **LL**: Lower limit of range **UL**: Upper limit of range	Compares the 32-bit unsigned binary value in CD and CD+1 (word contents or constant) to the range defined by LL and UL and outputs the results to the Arithmetic Flags in the Auxiliary Area.

A-1-6 Data Movement Instructions

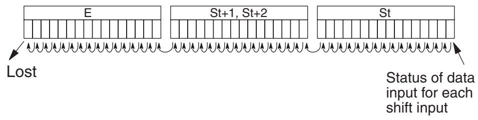
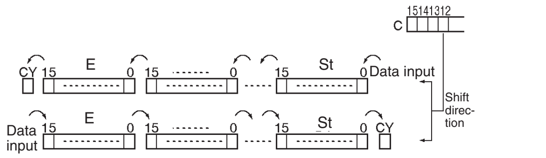
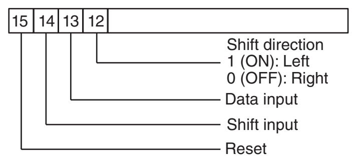
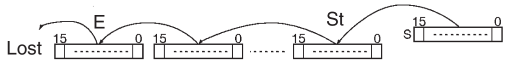
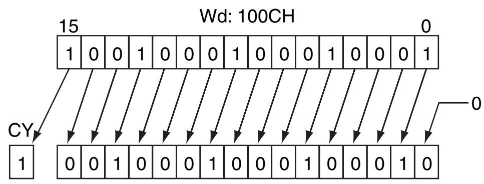
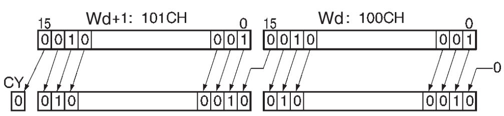
Instruction	Mnemonic	Variations	Symbol/Operand	Function
MOVE	MOV	@/!/!@	MOV(021) S D **S**: Source **D**: Destination	Transfers a word of data to the specified word. 
DOUBLE MOVE	MOVL	@	MOVL(498) S D **S**: 1st source word **D**: 1st destination word	Transfers two words of data to the specified words. 
MOVE NOT	MVN	@	MVN(022) S D **S**: Source **D**: Destination	Transfers the complement of a word of data to the specified word. 

Instruction	Mnemonic	Variations	Symbol/Operand	Function
MOVE BIT	MOVB	@	MOVB(082) S C D **S**: Source word or data **C**: Control word **D**: Destination word	Transfers the specified bit.  C: Control Word 
MOVE DIGIT	MOVD	@	MOVD(083) S C D **S**: Source word or data **C**: Control word **D**: Destination word	Transfers the specified digit or digits. (Each digit is made up of 4 bits.)  C: Control Word 
MULTIPLE BIT TRANSFER	XFRB	@	XFRB(062) C S D **C**: Control word **S**: 1st source word **D**: 1st destination word	Transfers the specified number of consecutive bits.  C: Control Word 

Instruction	Mnemonic	Variations	Symbol/Operand	Function
BLOCK TRANSFER	XFER	@	XFER(070) N S D **N**: Number of words **S**: 1st source word **D**: 1st destination word	Transfers the specified number of consecutive words.  N: Number of Words The possible range for N is 0000 to FFFF (0 to 65,535 decimal). 
BLOCK SET	BSET	@	BSET(071) S St E **S**: Source word **St**: Starting word **E**: End word	Copies the same word to a range of consecutive words. 
DATA EXCHANGE	XCHG	@	XCHG(073) E1 E2 **E1**: 1st exchange word **E2**: Second exchange word	Exchanges the contents of the two specified words. 
SINGLE WORD DIS-TRIBUTE	DIST	@	DIST(080) S Bs Of **S**: Source word **Bs**: Destination base address **Of**: Offset	Transfers the source word to a destination word calculated by adding an offset value to the base address. 
DATA COL-LECT	COLL	@	COLL(081) Bs Of D **Bs**: Source base address **Of**: Offset **D**: Destination word	Transfers the source word (calculated by adding an offset value to the base address) to the destination word. 
MOVE TO REGISTER	MOVR	@	MOVR(560) S D **S**: Source (desired word or bit) **D**: Destination (Index Register)	Sets the PLC memory address of the specified word, bit, or timer/counter Completion Flag in the specified Index Register. 

Instruction	Mnemonic	Variations	Symbol/Operand	Function
MOVE TIMER/ COUNTER PV TO REGISTER	MOVRW	@	MOVRW(561) S D **S**: Source (desired TC number) **D**: Destination (Index Register)	Sets the PLC memory address of the specified timer or counter's PV in the specified Index Register. 

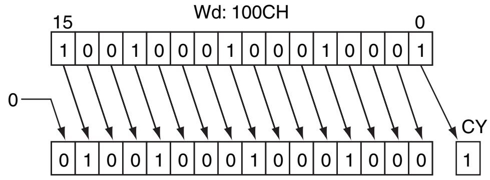
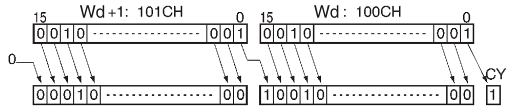
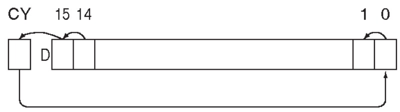
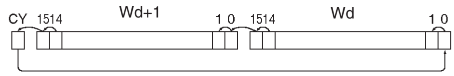
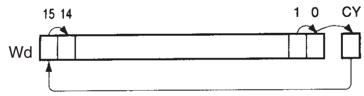
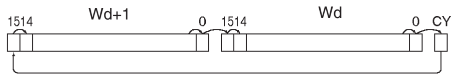
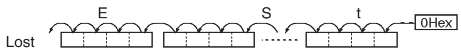
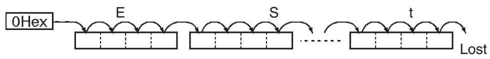
A-1-7 Data Shift Instructions

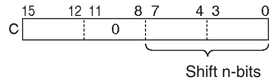
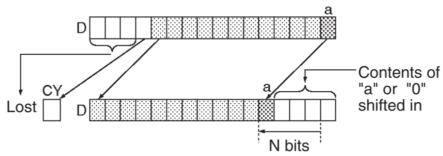
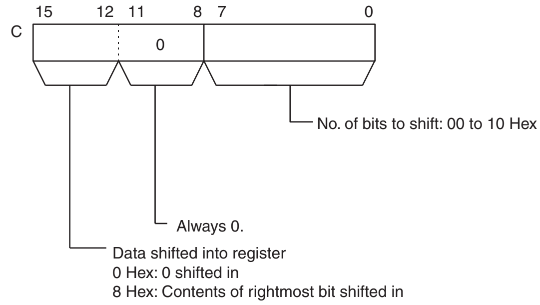
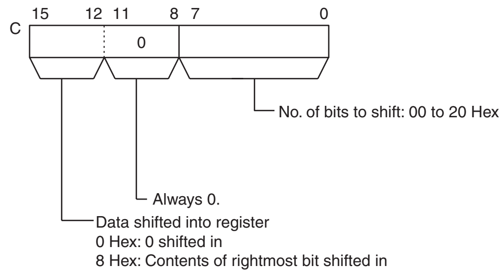
Instruction	Mnemonic	Variations	Symbol/Operand	Function
SHIFT REGISTER	SFT	---	Data input, Shift input, Reset input SFT(010) St E **St**: Starting word **E**: End word	Operates a shift register. 
REVERSIBLE SHIFT REGISTER	SFTR	@	SFTR(084) C St E **C**: Control word **St**: Starting word **E**: End word	Creates a shift register that shifts data to either the right or the left.  C: Control Word 
WORD SHIFT	WSFT	@	WSFT(016) S St E **S**: Source word **St**: Starting word **E**: End word	Shifts data between St and E in word units. 
ARITHMETIC SHIFT LEFT	ASL	@	ASL(025) Wd **Wd**: Word	Shifts the contents of Wd one bit to the left. 
DOUBLE SHIFT LEFT	ASLL	@	ASLL(570) Wd **Wd**: Word	Shifts the contents of Wd and Wd +1 one bit to the left. 

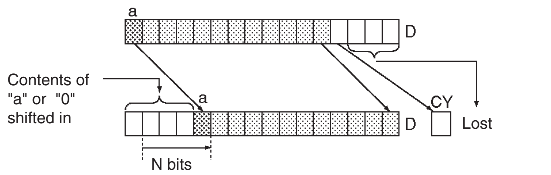
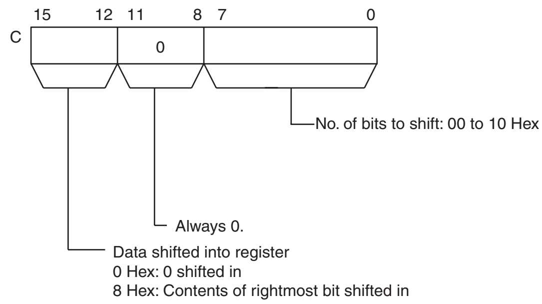
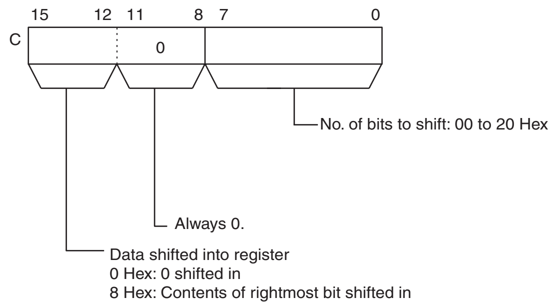
A-1 Instruction Functions

App

A-1-7 Data Shift Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
ARITHMETIC SHIFT RIGHT	ASR	@	ASR(026) Wd **Wd**: Word	Shifts the contents of Wd one bit to the right. 
DOUBLE SHIFT RIGHT	ASRL	@	ASRL(571) Wd **Wd**: Word	Shifts the contents of Wd and Wd +1 one bit to the right. 
ROTATE LEFT	ROL	@	ROL(027) Wd **Wd**: Word	Shifts all Wd bits one bit to the left including the Carry Flag (CY). 
DOUBLE ROTATE LEFT	ROLL	@	ROLL(572) Wd **Wd**: Word	Shifts all Wd and Wd +1 bits one bit to the left including the Carry Flag (CY). 
ROTATE RIGHT	ROR	@	ROR(028) Wd **Wd**: Word	Shifts all Wd bits one bit to the right including the Carry Flag (CY). 
DOUBLE ROTATE RIGHT	RORL	@	RORL(573) Wd **Wd**: Word	Shifts all Wd and Wd +1 bits one bit to the right including the Carry Flag (CY). 
ONE DIGIT SHIFT LEFT	SLD	@	SLD(074) St E **St**: Starting word **E**: End word	Shifts data by one digit (4 bits) to the left. 
ONE DIGIT SHIFT RIGHT	SRD	@	SRD(075) St E **St**: Starting word **E**: End word	Shifts data by one digit (4 bits) to the right. 

Instruction	Mnemonic	Variations	Symbol/Operand	Function
SHIFT N-BITS LEFT	NASL	@	NASL(580) D C **D**: Shift word **C**: Control word	Shifts the specified 16 bits(NASL) or 32 bits(NSLL) of word data to the left by the specified number of bits.   C: Control word • NASL  • NSLL 
DOUBLE SHIFT N-BITS LEFT	NSLL	@	NSLL(582) D C **D**: Shift word **C**: Control word	

Instruction	Mnemonic	Variations	Symbol/Operand	Function
SHIFT N-BITS RIGHT	NASR	@	NASR(581) D C **D**: Shift word **C**: Control word	Shifts the specified 16 bits(NASR) or 32 bits(NSRL) of word data to the right by the specified number of bits.  C: Control word • NASR  • NSRL 
DOUBLE SHIFT N-BITS RIGHT	NSRL	@	NSRL(583) D C **D**: Shift word **C**: Control word	

A-1-8 Increment/Decrement Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
INCREMENT BINARY	++	@	++(590) Wd **Wd**: Word	Increments the 4-digit hexadecimal content of the specified word by 1. Wd +1 → Wd
DOUBLE INCREMENT BINARY	++L	@	++L(591) Wd **Wd**: Word	Increments the 8-digit hexadecimal content of the specified words by 1. Wd+1 Wd +1 → Wd+1 Wd
DECREMENT BINARY	– –	@	– – (592) Wd **Wd**: Word	Decrements the 4-digit hexadecimal content of the specified word by 1. Wd –1 → Wd
DOUBLE DEC-REMENT BINARY	– –L	@	– – L(593) Wd **Wd**: 1st word	Decrements the 8-digit hexadecimal content of the specified words by 1. Wd+1 Wd –1 → Wd+1 Wd
INCREMENT BCD	++B	@	++B(594) Wd **Wd**: Word	Increments the 4-digit BCD content of the specified word by 1. Wd +1 → Wd

Instruction	Mnemonic	Variations	Symbol/Operand	Function
DOUBLE INCREMENT BCD	++BL	@	++BL(595) Wd **Wd**: 1st word	Increments the 8-digit BCD content of the specified words by 1. Wd+1 Wd +1 → Wd+1 Wd
DECREMENT BCD	– –B	@	– – B(596) Wd **Wd**: Word	Decrements the 4-digit BCD content of the specified word by 1. Wd –1 → Wd
DOUBLE DEC-REMENT BCD	– –BL	@	– – BL(597) Wd **Wd**: 1st word	Decrements the 8-digit BCD content of the specified words by 1. Wd+1 Wd –1 → Wd+1 Wd

A-1-9 Symbol Math Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
SIGNED BINARY ADD WITHOUT CARRY	+	@	+(400) Au Ad R **Au**: Augend word **Ad**: Addend word **R**: Result word	Adds 4-digit (single-word) hexadecimal data and/or constants. Au (Signed binary) + Ad (Signed binary) CY R (Signed binary) CY will turn ON when there is a carry.
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L	@	+L(401) Au Ad R **Au**: 1st augend word **Ad**: 1st addend word **R**: 1st result word	Adds 8-digit (double-word) hexadecimal data and/or constants. Au+1 Au (Signed binary) + Ad+1 Ad (Signed binary) CY R+1 R (Signed binary) CY will turn ON when there is a carry.
SIGNED BINARY ADD WITH CARRY	+C	@	+C(402) Au Ad R **Au**: Augend word **Ad**: Addend word **R**: Result word	Adds 4-digit (single-word) hexadecimal data and/or constants with the Carry Flag (CY). Au (Signed binary) Ad (Signed binary) + CY CY R (Signed binary) CY will turn ON when there is a carry.
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL	@	+CL(403) Au Ad R **Au**: 1st augend word **Ad**: 1st addend word **R**: 1st result word	Adds 8-digit (double-word) hexadecimal data and/or constants with the Carry Flag (CY). Au+1 Au (Signed binary) Ad+1 Ad (Signed binary) + CY CY R+1 R (Signed binary) CY will turn ON when there is a carry.

Instruction	Mnemonic	Variations	Symbol/Operand	Function
BCD ADD WITHOUT CARRY	+B	@	+B(404) Au Ad R **Au**: Augend word **Ad**: Addend word **R**: Result word	Adds 4-digit (single-word) BCD data and/or constants. Au (BCD) + Ad (BCD) CY will turn ON when there is a carry. CY R (BCD)
DOUBLE BCD ADD WITH-OUT CARRY	+BL	@	+BL(405) Au Ad R **Au**: 1st augend word **Ad**: 1st addend word **R**: 1st result word	Adds 8-digit (double-word) BCD data and/or constants. Au+1 Au (BCD) + Ad+1 Ad (BCD) CY will turn ON when there is a carry. CY R+1 R (BCD)
BCD ADD WITH CARRY	+BC	@	+BC(406) Au Ad R **Au**: Augend word **Ad**: Addend word **R**: Result word	Adds 4-digit (single-word) BCD data and/or constants with the Carry Flag (CY). Au (BCD) Ad (BCD) + CY CY will turn ON when there is a carry. CY R (BCD)
DOUBLE BCD ADD WITH CARRY	+BCL	@	+BCL(407) Au Ad R **Au**: 1st augend word **Ad**: 1st addend word **R**: 1st result word	Adds 8-digit (double-word) BCD data and/or constants with the Carry Flag (CY). Au+1 Au (BCD) Ad+1 Ad (BCD) + CY CY will turn ON when there is a carry. CY R+1 R (BCD)
SIGNED BINARY SUB-TRACT WITH-OUT CARRY	–	@	– (410) Mi Su R **Mi**: Minuend word **Su**: Subtrahend word **R**: Result word	Subtracts 4-digit (single-word) hexadecimal data and/or constants. Mi (Signed binary) – Su (Signed binary) CY will turn ON when there is a borrow. CY R (Signed binary)
DOUBLE SIGNED BINARY SUB-TRACT WITH-OUT CARRY	– L	@	–L(411) Mi Su R **Mi**: Minuend word **Su**: Subtrahend word **R**: Result word	Subtracts 8-digit (double-word) hexadecimal data and/or con-stants. Mi+1 Mi (Signed binary) – Su+1 Su (Signed binary) CY will turn ON when there is a borrow. CY R+1 R (Signed binary)

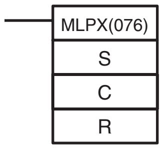
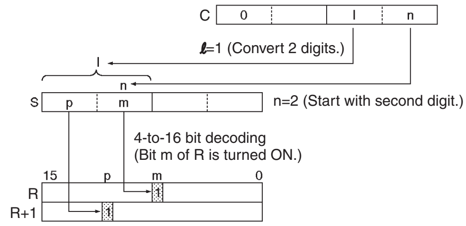
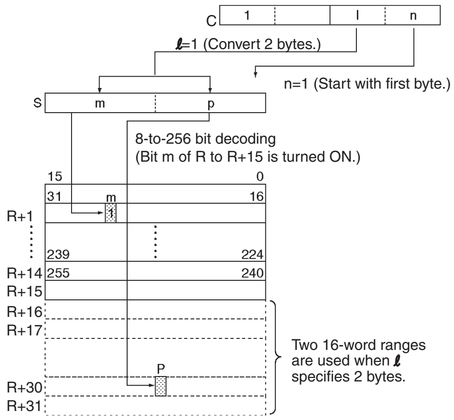
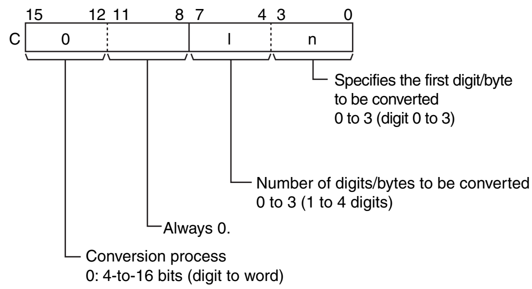
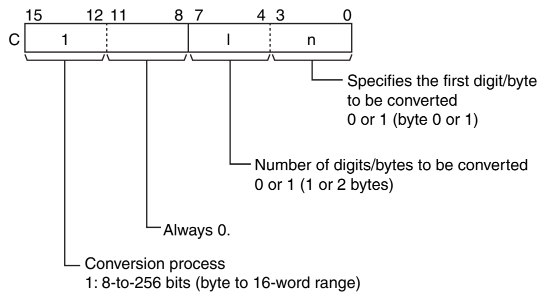
Instruction	Mnemonic	Variations	Symbol/Operand	Function
SIGNED BINARY SUB-TRACT WITH CARRY	– C	@	–C(412) Mi Su R **Mi**: Minuend word **Su**: Subtrahend word **R**: Result word	Subtracts 4-digit (single-word) hexadecimal data and/or constants with the Carry Flag (CY). Mi (Signed binary) Su (Signed binary) – CY CY will turn ON when there is a borrow. CY R (Signed binary)
DOUBLE SIGNED BINARY WITH CARRY	– CL	@	–CL(413) Mi Su R **Mi**: Minuend word **Su**: Subtrahend word **R**: Result word	Subtracts 8-digit (double-word) hexadecimal data and/or con-stants with the Carry Flag (CY). Mi+1 Mi (Signed binary) Su+1 Su (Signed binary) – CY CY will turn ON when there is a borrow. CY R+1 R (Signed binary)
BCD SUB-TRACT WITH-OUT CARRY	– B	@	–B(414) Mi Su R **Mi**: Minuend word **Su**: Subtrahend word **R**: Result word	Subtracts 4-digit (single-word) BCD data and/or constants. Mi (BCD) – Su (BCD) CY will turn ON when there is a carry. CY R (BCD)
DOUBLE BCD SUBTRACT WITHOUT CARRY	– BL	@	–BL(415) Mi Su R **Mi**: 1st minuend word **Su**: 1st subtrahend word **R**: 1st result word	Subtracts 8-digit (double-word) BCD data and/or constants. Mi +1 Mi (BCD) – Su+1 Su (BCD) CY will turn ON when there is a borrow. CY R+1 R (BCD)
BCD SUB-TRACT WITH CARRY	– BC	@	–BC(416) Mi Su R **Mi**: Minuend word **Su**: Subtrahend word **R**: Result word	Subtracts 4-digit (single-word) BCD data and/or constants with the Carry Flag (CY). Mi (BCD) Su (BCD) – CY CY will turn ON when there is a borrow. CY R (BCD)
DOUBLE BCD SUBTRACT WITH CARRY	– BCL	@	–BCL(417) Mi Su R **Mi**: 1st minuend word **Su**: 1st subtrahend word **R**: 1st result word	Subtracts 8-digit (double-word) BCD data and/or constants with the Carry Flag (CY). Mi +1 Mi (BCD) Su+1 Su (BCD) – CY CY will turn ON when there is a borrow. CY R+1 R (BCD)

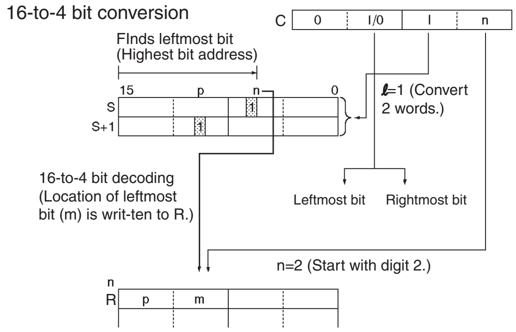
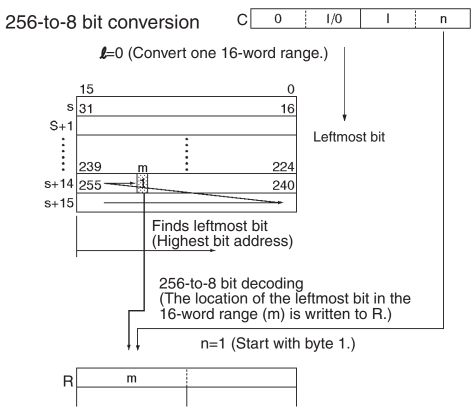
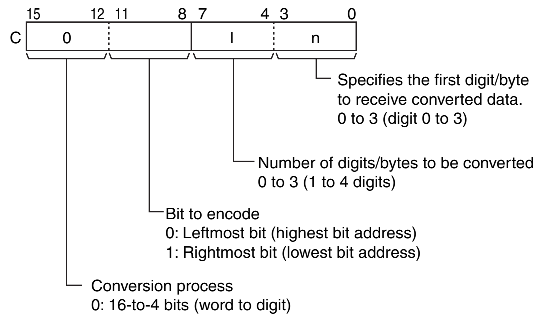
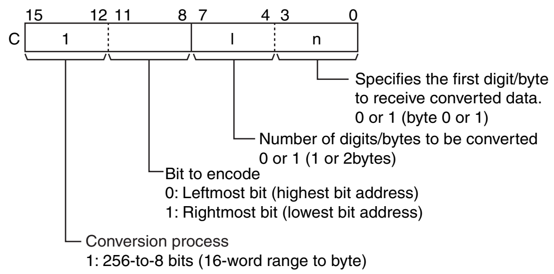
Instruction	Mnemonic	Variations	Symbol/Operand	Function
SIGNED BINARY MUL-TIPLY	*	@	*(420) Md Mr R **Md**: Multiplicand word **Mr**: Multiplier word **R**: Result word	Multiplies 4-digit signed hexadecimal data and/or constants. Md (Signed binary) × Mr (Signed binary) R +1 | R (Signed binary)
DOUBLE SIGNED BINARY MUL-TIPLY	*L	@	*L(421) Md Mr R **Md**: 1st multiplicand word **Mr**: 1st multiplier word **R**: 1st result word	Multiplies 8-digit signed hexadecimal data and/or constants. Md + 1 | Md (Signed binary) × Mr + 1 | Mr (Signed binary) R + 3 | R + 2 | R + 1 | R (Signed binary)
UNSIGNED BINARY MUL-TIPLY	*U	@	*U(422) Md Mr R **Md**: Multiplicand word **Mr**: Multiplier word **R**: Result word	Multiplies 4-digit unsigned hexadecimal data and/or constants. Md (Unsigned binary) × Mr (Unsigned binary) R+1 | R (Unsigned binary)
DOUBLE UNSIGNED BINARY MUL-TIPLY	*UL	@	*UL(423) Md Mr R **Md**: 1st multiplicand word **Mr**: 1st multiplier word **R**: 1st result word	Multiplies 8-digit unsigned hexadecimal data and/or constants. Md + 1 | Md (Unsigned binary) × Mr + 1 | Mr (Unsigned binary) R + 3 | R + 2 | R + 1 | R (Unsigned binary)
BCD MULTI-PLY	*B	@	*B(424) Md Mr R **Md**: Multiplicand word **Mr**: Multiplier word **R**: Result word	Multiplies 4-digit (single-word) BCD data and/or constants. Md (BCD) × Mr (BCD) R +1 | R (BCD)
DOUBLE BCD MULTIPLY	*BL	@	*BL(425) Md Mr R **Md**: 1st multiplicand word **Mr**: 1st multiplier word **R**: 1st result word	Multiplies 8-digit (double-word) BCD data and/or constants. Md + 1 | Md (BCD) × Mr + 1 | Mr (BCD) R + 3 | R + 2 | R + 1 | R (BCD)
SIGNED BINARY DIVIDE	/	@	/(430) Dd Dr R **Dd**: Dividend word **Dr**: Divisor word **R**: Result word	Divides 4-digit (single-word) signed hexadecimal data and/or con-stants. Dd (Signed binary) ÷ Dr (Signed binary) R +1 | R (Signed binary) Remainder Quotient

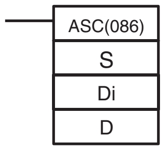
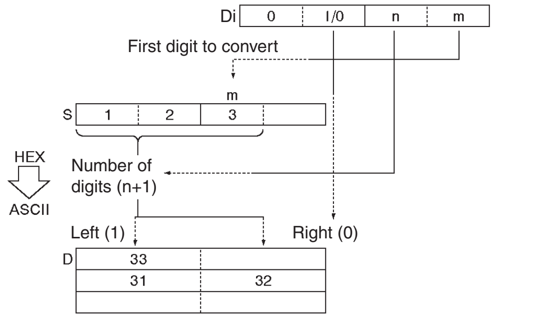
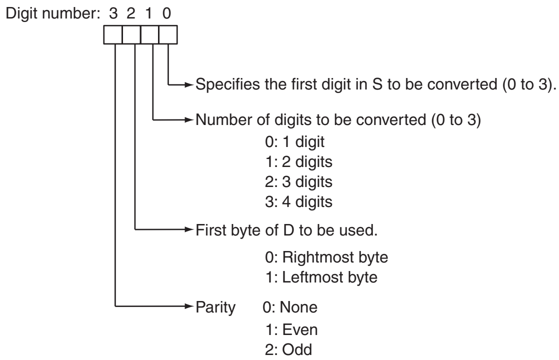
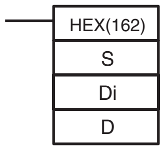
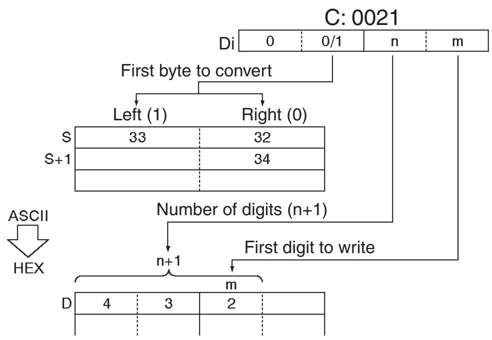
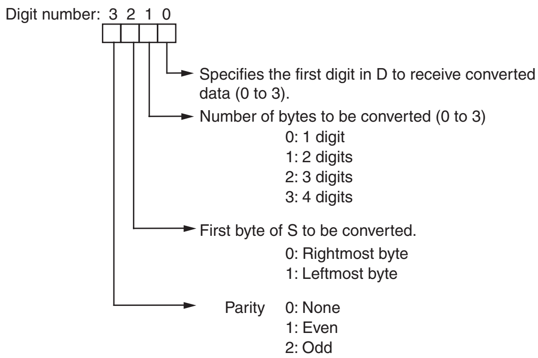
Instruction	Mnemonic	Variations	Symbol/Operand	Function
DOUBLE SIGNED BINARY DIVIDE	/L	@	/L(431) Dd Dr R **Dd**: 1st dividend word **Dr**: 1st divisor word **R**: 1st result word	Divides 8-digit (double-word) signed hexadecimal data and/or constants. Dd + 1 | Dd (Signed binary) ÷ Dr + 1 | Dr (Signed binary) R + 3 | R + 2 | R + 1 | R (Signed binary) Remainder Quotient
UNSIGNED BINARY DIVIDE	/U	@	/U(432) Dd Dr R **Dd**: Dividend word **Dr**: Divisor word **R**: Result word	Function:Divides 4-digit (single-word) unsigned hexadecimal data and/or constants. Dd (Unsigned binary) ÷ Dr (Unsigned binary) R+1 | R (Unsigned binary) Remainder Quotient
DOUBLE UNSIGNED BINARY DIVIDE	/UL	@	/UL(433) Dd Dr R **Dd**: 1st dividend word **Dr**: 1st divisor word **R**: 1st result word	Function:Divides 8-digit (double-word) unsigned hexadecimal data and/or constants. Dd + 1 | Dd (Unsigned binary) ÷ Dr + 1 | Dr (Unsigned binary) R + 3 | R + 2 | R + 1 | R (Unsigned binary) Remainder Quotient
BCD DIVIDE	/B	@	/B(434) Dd Dr R **Dd**: Dividend word **Dr**: Divisor word **R**: Result word	Divides 4-digit (single-word) BCD data and/or constants. Dd (BCD) ÷ Dr (BCD) R +1 | R (BCD) Remainder Quotient
DOUBLE BCD DIVIDE	/BL	@	/BL(435) Dd Dr R **Dd**: 1st dividend word **Dr**: 1st divisor word **R**: 1st result word	Divides 8-digit (double-word) BCD data and/or constants. Dd + 1 | Dd (BCD) ÷ Dr + 1 | Dr (BCD) R + 3 | R + 2 | R + 1 | R (BCD) Remainder Quotient

A-1-10 Conversion Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
BCD-TO-BINARY	BIN	@	BIN(023) S R **S**: Source word **R**: Result word	Converts BCD data to binary data. S (BCD) → R (BIN)
DOUBLE BCD-TO-DOUBLE BINARY	BINL	@	BINL(058) S R **S**: 1st source word **R**: 1st result word	Converts 8-digit BCD data to 8-digit hexadecimal (32-bit binary) data. S (BCD), S+1 (BCD) → R (BIN), R+1 (BIN)
BINARY-TO-BCD	BCD	@	BCD(024) S R **S**: Source word **R**: Result word	Converts a word of binary data to a word of BCD data. S (BIN) → R (BCD)
DOUBLE BINARY-TO-DOUBLE BCD	BCDL	@	BCDL(059) S R **S**: 1st source word **R**: 1st result word	Converts 8-digit hexadecimal (32-bit binary) data to 8-digit BCD data. S (BIN), S+1 (BIN) → R (BCD), R+1 (BCD)
2'S COMPLEMENT	NEG	@	NEG(160) S R **S**: Source word **R**: Result word	Calculates the 2's complement of a word of hexadecimal data. 2's complement (Complement + 1) (S) → (R)

Instruction	Mnemonic	Variations	Symbol/Operand	Function
DATA DECODER	MLPX	@	 **S**: Source word **C**: Control word **R**: 1st result word	Reads the numerical value in the specified digit (or byte) in the source word, turns ON the corresponding bit in the result word (or 16-word range), and turns OFF all other bits in the result word (or 16-word range). 4-to-16 bit conversion  8-to-256 bit conversion  C: Control Word • 4-to-16 bit decoder  • 8-to-256 bit conversion 

Instruction	Mnemonic	Variations	Symbol/Operand	Function
DATA ENCODER	DMPX	@	DMPX(077) S R C **S**: 1st source word **R**: Result word **C**: Control word	FInds the location of the first or last ON bit within the source word (or 16-word range), and writes that value to the specified digit (or byte) in the result word.   C: Control Word • 16-to-4 bit conversion  • 256-to-8 bit conversion 

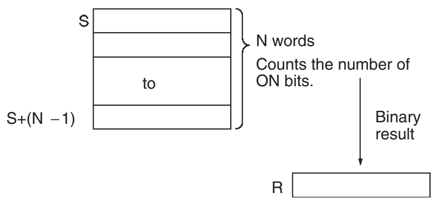
Instruction	Mnemonic	Variations	Symbol/Operand	Function
ASCII CON-VERT	ASC	@	 **S**: Source word **Di**: Digit designator **D**: 1st destination word	Converts 4-bit hexadecimal digits in the source word into their 8-bit ASCII equivalents.  DI: Digit Designator 
ASCII TO HEX	HEX	@	 **S**: 1st source word **Di**: Digit designator **D**: Destination word	Converts up to 4 bytes of ASCII data in the source word to their hexadecimal equivalents and writes these digits in the specified destination word.  DI: Digit Designator 

A-1-11 Logic Instructions

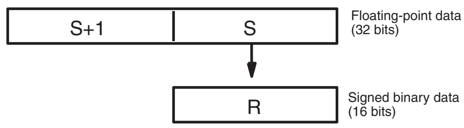
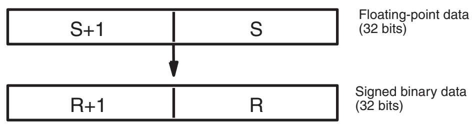
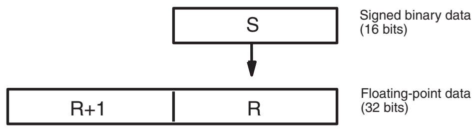
<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr><td>LOGICAL AND</td><td>ANDW</td><td>@</td><td>ANDW(034)
I_1
I_2
R
I1: Input 1
I2: Input 2
R: Result word</td><td>Takes the logical AND of corresponding bits in single words of word data and/or constants.
$I_1 . I_2 \rightarrow R$
<table>
<tr><th>I_1</th><th>I_2</th><th>R</th></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>DOUBLE LOGICAL AND</td><td>ANDL</td><td>@</td><td>ANDL(610)
I_1
I_2
R
I1: Input 1
I2: Input 2
R: Result word</td><td>Takes the logical AND of corresponding bits in double words of word data and/or constants.
$(I_1, I_1+1) . (I_2, I_2+1) \rightarrow (R, R+1)$
<table>
<tr><th>I_1, I_1+1</th><th>I_2, I_2+1</th><th>R, R+1</th></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>LOGICAL OR</td><td>ORW</td><td>@</td><td>ORW(035)
I_1
I_2
R
I1: Input 1
I2: Input 2
R: Result word</td><td>Takes the logical OR of corresponding bits in single words of word data and/or constants.
$I_1 + I_2 \rightarrow R$
<table>
<tr><th>I_1</th><th>I_2</th><th>R</th></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>DOUBLE LOGICAL OR</td><td>ORWL</td><td>@</td><td>ORWL(611)
I_1
I_2
R
I1: Input 1
I2: Input 2
R: Result word</td><td>Takes the logical OR of corresponding bits in double words of word data and/or constants.
$(I_1, I_1+1) + (I_2, I_2+1) \rightarrow (R, R+1)$
<table>
<tr><th>I_1, I_1+1</th><th>I_2, I_2+1</th><th>R, R+1</th></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>EXCLUSIVE OR</td><td>XORW</td><td>@</td><td>XORW(036)
I_1
I_2
R
I1: Input 1
I2: Input 2
R: Result word</td><td>Takes the logical exclusive OR of corresponding bits in single words of word data and/or constants.
$I_1 . I_2 + I_1 . I_2 \rightarrow R$
<table>
<tr><th>I_1</th><th>I_2</th><th>R</th></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>DOUBLE EXCLUSIVE OR</td><td>XORL</td><td>@</td><td>XORL(612)
I_1
I_2
R
I1: Input 1
I2: Input 2
R: Result word</td><td>Takes the logical exclusive OR of corresponding bits in double words of word data and/or constants.
$(I_1, I_1+1) . \overline{(I_2, I_2+1)} + \overline{(I_1, I_1+1)} . (I_2, I_2+1) \rightarrow (R, R+1)$
<table>
<tr><th>I_1, I_1+1</th><th>I_2, I_2+1</th><th>R, R+1</th></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
</table>

Instruction	Mnemonic	Variations	Symbol/Operand	Function
COMPLE-MENT	COM	@	COM(029) Wd **Wd**: Word	Turns OFF all ON bits and turns ON all OFF bits in Wd. $\overline{\text{Wd}} \rightarrow$ Wd: 1 $\rightarrow$ 0 and 0 $\rightarrow$ 1
DOUBLE COMPLE-MENT	COML	@	COML(614) Wd **Wd**: Word	Turns OFF all ON bits and turns ON all OFF bits in Wd and Wd+1. $\overline{\text{(Wd+1, Wd)}} \rightarrow$ (Wd+1, Wd)

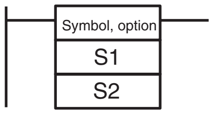
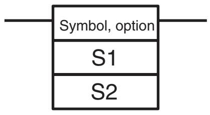
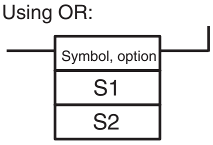
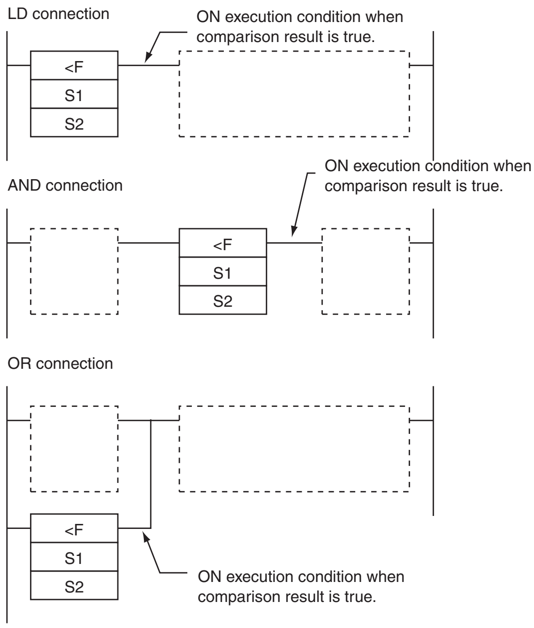
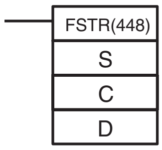
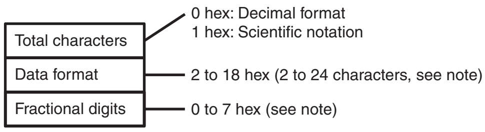
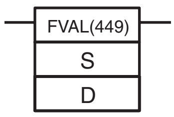
A-1-12 Special Math Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
ARITHMETIC PROCESS	APR	@	APR(069) C S R **C**: Control word **S**: Source data **R**: Result word	Calculates the sine, cosine, or a linear extrapolation of the source data. The linear extrapolation function allows any relationship between X and Y to be approximated with line segments.
BIT COUNTER	BCNT	@	BCNT(067) N S R **N**: Number of words **S**: 1st source word **R**: Result word	Counts the total number of ON bits in the specified word(s).  N: Number of words The number of words must be 0001 to FFFF (1 to 65,535 words).

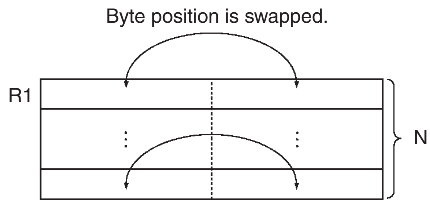
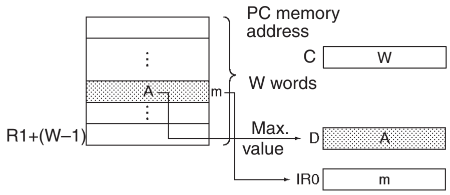
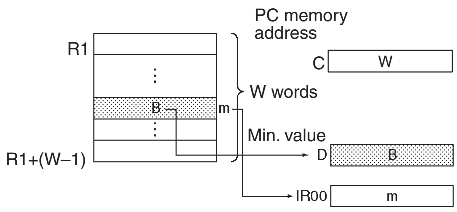
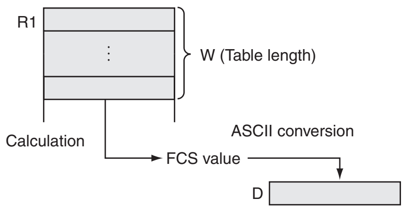
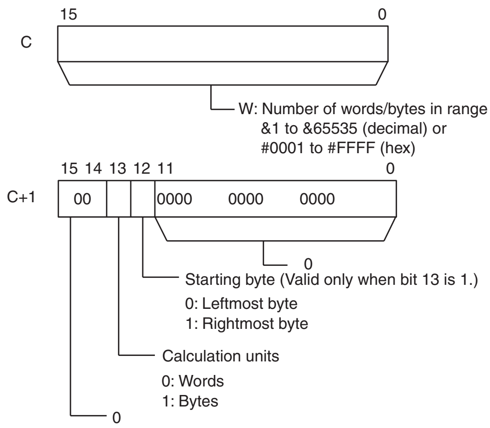
A-1-13 Floating-point Math Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
FLOATING TO 16-BIT	FIX	@	FIX(450) S R **S**: 1st source word **R**: Result word	Converts a 32-bit floating-point value to 16-bit signed binary data and places the result in the specified result word. 
FLOATING TO 32-BIT	FIXL	@	FIXL(451) S R **S**: 1st source word **R**: 1st result word	Converts a 32-bit floating-point value to 32-bit signed binary data and places the result in the specified result words. 
16-BIT TO FLOATING	FLT	@	FLT(452) S R **S**: Source word **R**: 1st result word	Converts a 16-bit signed binary value to 32-bit floating-point data and places the result in the specified result words. 

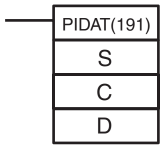
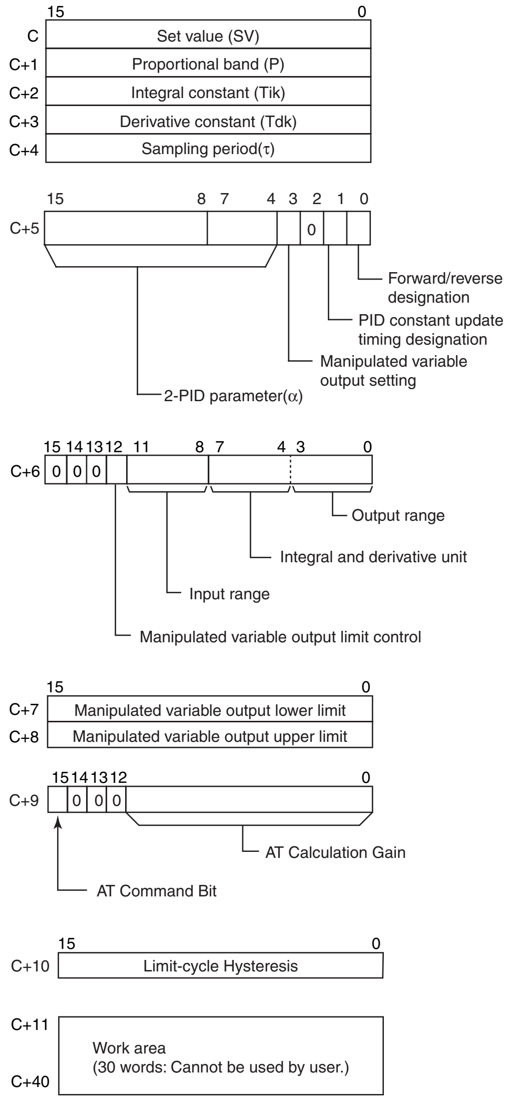
Instruction	Mnemonic	Variations	Symbol/Operand	Function
32-BIT TO FLOATING	FLTL	@	FLTL(453) S R **S**: 1st source word **R**: 1st result word	Converts a 32-bit signed binary value to 32-bit floating-point data and places the result in the specified result words. S+1 | S — Signed binary data (32 bits) ↓ R+1 | R — Floating-point data (32 bits)
FLOATING-POINT ADD	+F	@	+F(454) Au Ad R **Au**: 1st augend word **AD**: 1st addend word **R**: 1st result word	Adds two 32-bit floating-point numbers and places the result in the specified result words. Au+1 | Au — Augend (floating-point data, 32 bits) + Ad+1 | Ad — Addend (floating-point data, 32 bits) R+1 | R — Result (floating-point data, 32 bits)
FLOATING-POINT SUBTRACT	–F	@	–F(455) Mi Su R **Mi**: 1st Minuend word **Su**: 1st Subtrahend word **R**: 1st result word	Subtracts one 32-bit floating-point number from another and places the result in the specified result words. Mi+1 | Mi — Minuend (floating-point data, 32 bits) – Su+1 | Su — Subtrahend (floating-point data, 32 bits) R+1 | R — Result (floating-point data, 32 bits)
FLOATING-POINT MULTIPLY	*F	@	*F(456) Md Mr R **Md**: 1st Multiplicand word **Mr**: 1st Multiplier word **R**: 1st result word	Multiplies two 32-bit floating-point numbers and places the result in the specified result words. Md+1 | Md — Multiplicand (floating-point data, 32 bits) × Mr+1 | Mr — Multiplier (floating-point data, 32 bits) R+1 | R — Result (floating-point data, 32 bits)
FLOATING-POINT DIVIDE	/F	@	/F(457) Dd Dr R **Dd**: 1st Dividend word **Dr**: 1st Divisor word **R**: 1st result word	Divides one 32-bit floating-point number by another and places the result in the specified result words. Dd+1 | Dd — Dividend (floating-point data, 32 bits) ÷ Dr+1 | Dr — Divisor (floating-point data, 32 bits) R+1 | R — Result (floating-point data, 32 bits)

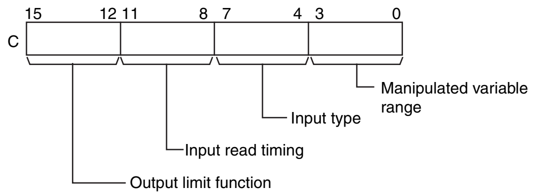
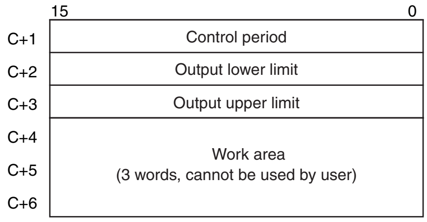
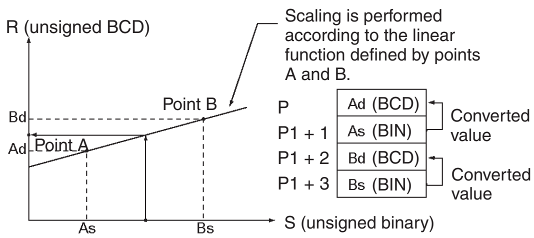
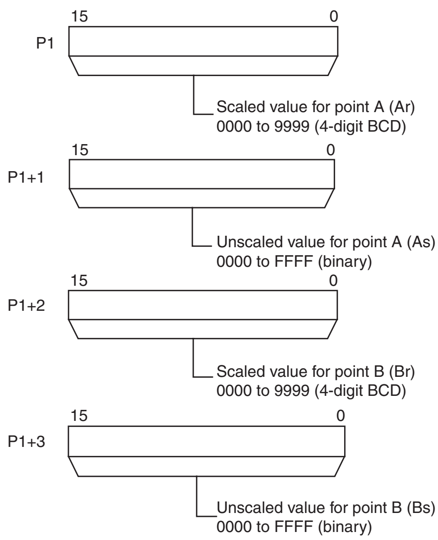
Instruction	Mnemonic	Variations	Symbol/Operand	Function
FLOATING SYMBOL COMPARISON	LD, AND, or OR + =F, <>F, <F, <=F, >F, or >=F	---	Using LD:  Using AND:  Using OR:  **S1**: Comparison data 1 **S2**: Comparison data 2	Compares the specified single-precision data (32 bits) or constants and creates an ON execution condition if the comparison result is true. 
FLOATING-POINT TO ASCII	FSTR	@	 **S:** 1st source word **C**: Control word **D**: Destination word	Converts the specified single-precision floating-point data (32-bit decimal-point or exponential format) to text string data (ASCII) and outputs the result to the destination word. C: First Control Word  **Note** There are limits on the total number of characters and the number of fractional digits.
ASCII TO FLOATING-POINT	FVAL	@	 **S:** Source word **D**: 1st destination word	Converts the specified text string (ASCII) representation of single-precision floating-point data (decimal-point or exponential format) to 32-bit single-precision floating-point data and outputs the result to the destination words.

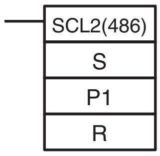
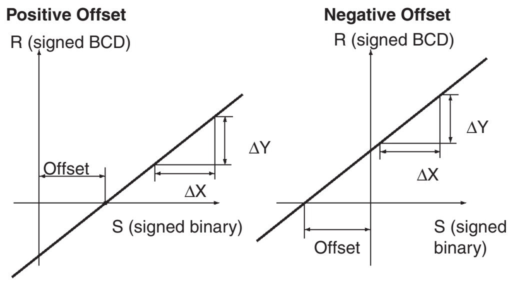
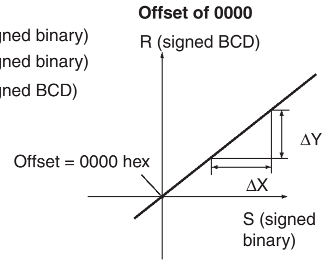
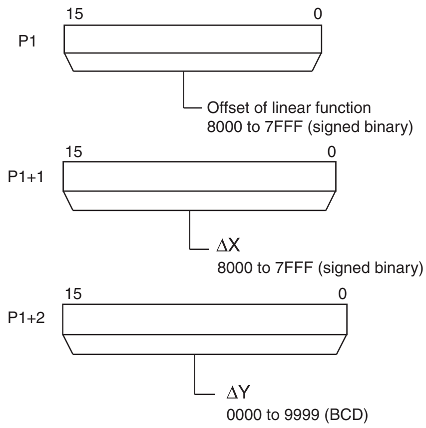
A-1-14 Table Data Processing Instructions

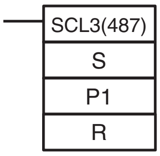
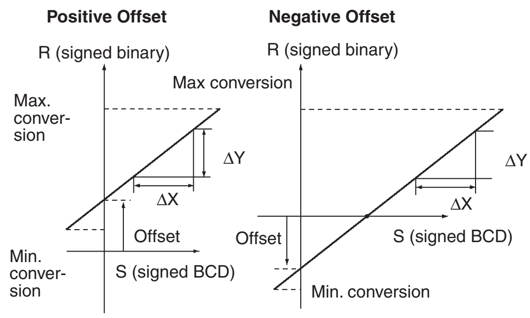
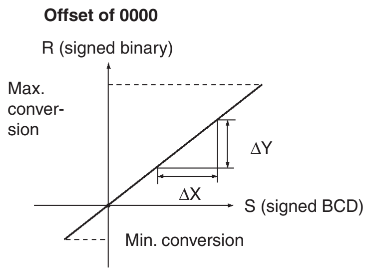
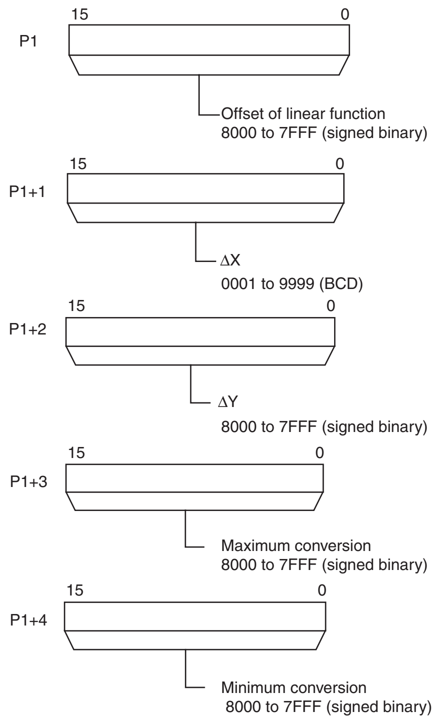
Instruction	Mnemonic	Variations	Symbol/Operand	Function
SWAP BYTES	SWAP	@	SWAP(637) N R1 **N**: Number of words **R1**: 1st word in range	Switches the leftmost and rightmost bytes in all of the words in the range. 
FIND MAXIMUM	MAX	@	MAX(182) C R1 D **C**: First control word **R1**: First word in range **D**: Destination word	Finds the maximum value in the range. 
FIND MINIMUM	MIN	@	MIN(183) C R1 D **C**: First control word **R1**: First word in range **D**: Destination word	Finds the minimum value in the range. 
FRAME CHECKSUM	FCS	@	FCS(180) C R1 D **C**: First control word **R1**: First word in range **D**: First destination word	Calculates the FCS value for the specified range and outputs the result in ASCII.  C: First control word 

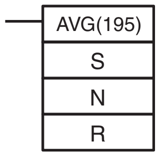
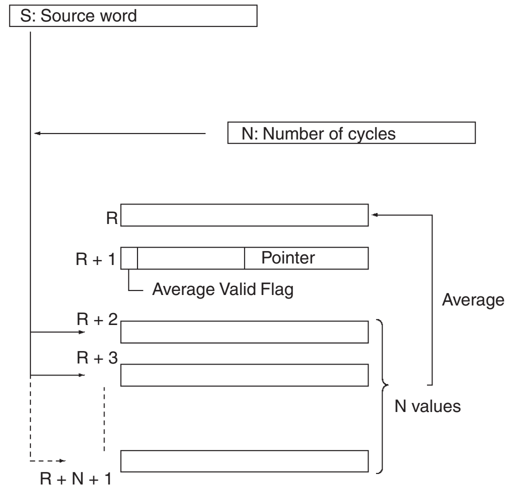
A-1-15 Data Control Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
PID CONTROL WITH AUTOTUNING	PIDAT	---	 **S**: Input word **C**: 1st parameter word **D**: Output word	Executes PID control according to the specified parameters. The PID constants can be auto-tuned with PIDAT(191). C: First Parameter Word 

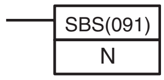
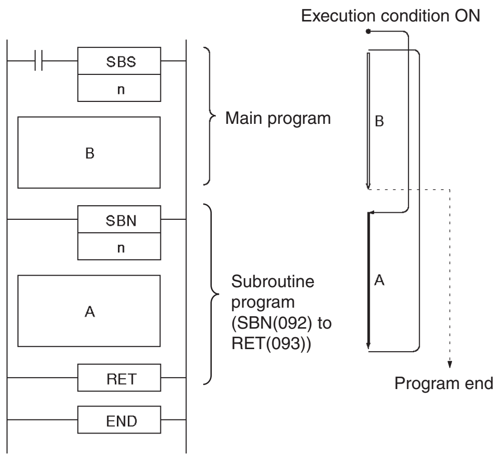
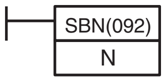
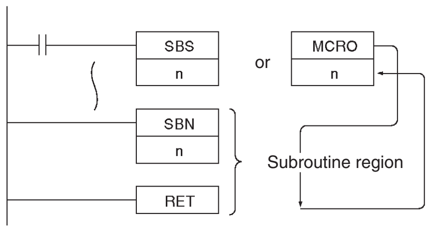
Instruction	Mnemonic	Variations	Symbol/Operand	Function
TIME-PRO- PORTIONAL OUTPUT	TPO	---	TPO (685) S C R **S**: Input word **C**: 1st parameter word **R**: Pulse output bit	Inputs the duty ratio or manipulated variable from the specified word, converts the duty ratio to a time-proportional output based on the specified parameters, and outputs the result from the specified output. C: First Parameter Word Bits 04 to 07 of C specify the input type, i.e., whether the input word contains an input duty ratio or manipulated variable. (Set these bits to 0 hex to specify a input duty ratio or to 1 hex to specify a manipulated variable.) The following diagram shows the locations of the parameter data.   R: Pulse Output Bit Specifies the destination output bit for the pulse output. Normally, specify an output bit allocated to a Transistor Output Unit and connect a solid state relay to the Transistor Output Unit.
SCALING	SCL	@	SCL(194) S P1 R **S**: Source word **P1**: 1st parameter word **R**: Result word	Converts unsigned binary data into unsigned BCD data according to the specified linear function.  P1: First Parameter Word 

Instruction	Mnemonic	Variations	Symbol/Operand	Function
SCALING 2	SCL2	@	 **S**: Source word **P1**: 1st parameter word **R**: Result word	Converts signed binary data into signed BCD data according to the specified linear function. An offset can be input in defining the linear function.  P1 Offset (Signed binary) P1 + 1 ΔY (Signed binary) P1 + 2 ΔX (Signed BCD)  P1: First Parameter Word  **Note** P1 to P1+2 must be in the same area.

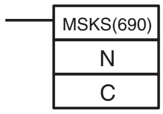
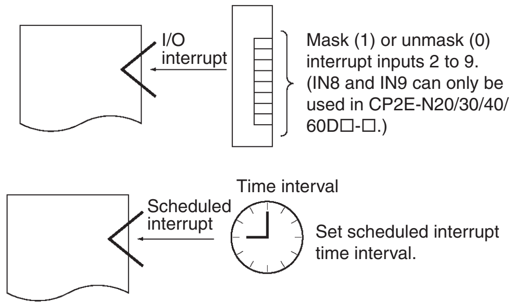
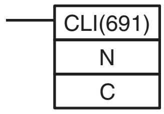
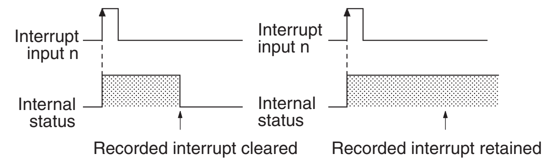
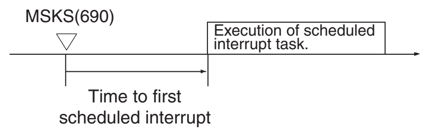
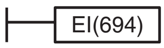
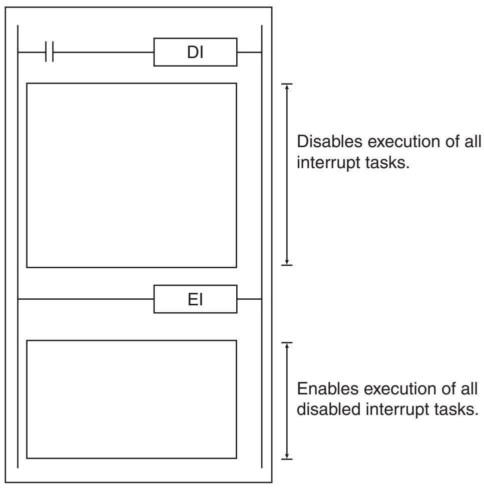
Instruction	Mnemonic	Variations	Symbol/Operand	Function
SCALING 3	SCL3	@	 **S**: Source word **P1**: 1st parameter word **R**: Result word	Converts signed BCD data into signed binary data according to the specified linear function. An offset can be input in defining the linear function.   P1: First Parameter Word  **Note** P1 to P1+4 must be in the same area.

Instruction	Mnemonic	Variations	Symbol/Operand	Function
AVERAGE	AVG	---	 **S**: Source word **N**: Number of cycles **R**: Result word	Calculates the average value of an input word for the specified number of cycles. 

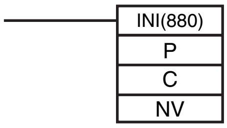
A-1-16 Subroutine Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
SUBROUTINE CALL	SBS	@	 **N**: Subroutine number	Calls the subroutine with the specified subroutine number and executes that program. 
SUBROUTINE ENTRY	SBN	---	 **N**: Subroutine number	Indicates the beginning of the subroutine program with the specified subroutine number. 
SUBROUTINE RETURN	RET	---		Indicates the end of a subroutine program.

A-1-17 Interrupt Control Instructions

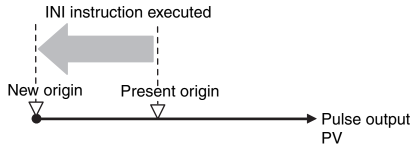
Instruction	Mnemonic	Variations	Symbol/Operand	Function
SET INTER-RUPT MASK	MSKS	@	 **N**: Interrupt number **C**: Control data	Sets up interrupt processing for I/O interrupts or scheduled interrupts. Both I/O interrupt tasks and scheduled interrupt tasks are masked (disabled) when the PC is first turned on. MSKS(690) can be used to unmask or mask I/O interrupts and set the time intervals for scheduled interrupts. 
CLEAR INTERRUPT	CLI	@	 **N**: Interrupt number **C**: Control data	Clears or retains recorded interrupt inputs for I/O interrupts or sets the time to the first scheduled interrupt for scheduled interrupts. N = 102 to 109 (108 and 109 can only be used in CP2E-N20/30/40/60D☐-☐.)  N = 4 
DISABLE INTERRUPTS	DI	@	 	Disables execution of all interrupt tasks.
ENABLE INTERRUPTS	EI	---	 	Enables execution of all interrupt tasks that were disabled with DI(693). 

A-1-18 High-speed Counter/Pulse Output Instructions

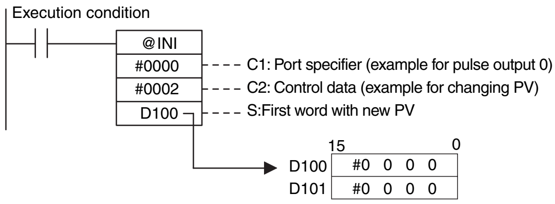
<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr>
<td>MODE CONTROL</td>
<td>INI</td>
<td>@</td>
<td>


P: Port specifier
C: Control data
NV: First word with new PV
</td>
<td>
INI(880) can be used to execute the following operations

- To start or stop comparison of a high-speed counter's PV to the comparison table registered with CTBL(882).
- To change the PV of the high-speed counter.
- To change the PV of the pulse output (origin fixed at 0).
- To stop pulse output.



Example: Setting the Present Position as the Origin



P: Port Specifier

0000 hex	Pulse output 0
0001 hex	Pulse output 1
0002 hex	Pulse output 2*
0003 hex	Pulse output 3*
0010 hex	High-speed counter 0
0011 hex	High-speed counter 1
0012 hex	High-speed counter 2
0013 hex	High-speed counter 3
0014 hex	High-speed counter 4
0015 hex	High-speed counter 5
1000 hex	PWM(891) output 0

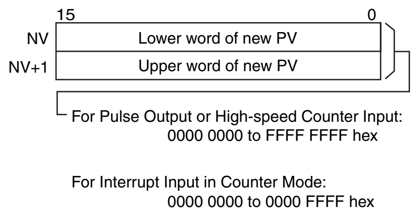
* Can only be used in CP2E-N30/40/60D☐-☐.

C: Control Data

0000 hex	Starts comparison.
0001 hex	Stops comparison.
0002 hex	Changes the PV.
0003 hex	Stops pulse output.

NV: First Word with New PV

If C is 0002 hex (i.e., when changing a PV), NV and NV+1 contain the new PV. Any values in NV and NV+1 are ignored when C is not 0002 hex.



</td>
</tr>
</table>

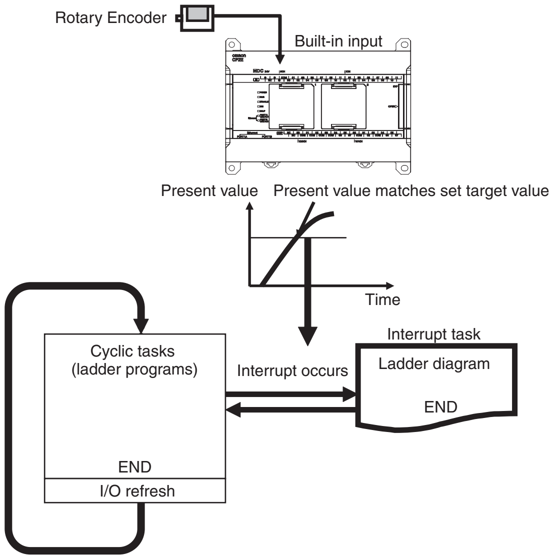
<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr>
<td>HIGH-SPEED COUNTER PV READ</td>
<td>PRV</td>
<td>@</td>
<td>PRV(881)
P
C
D

P: Port specifier
C: Control data
D: First destination word</td>
<td>Reads the High-speed counter PV and pulse output PV.
P: Port Specifier
<table>
<tr><td>0000 hex</td><td>Pulse output 0</td></tr>
<tr><td>0001 hex</td><td>Pulse output 1</td></tr>
<tr><td>0002 hex</td><td>Pulse output 2*</td></tr>
<tr><td>0003 hex</td><td>Pulse output 3*</td></tr>
<tr><td>0010 hex</td><td>High-speed counter 0</td></tr>
<tr><td>0011 hex</td><td>High-speed counter 1</td></tr>
<tr><td>0012 hex</td><td>High-speed counter 2</td></tr>
<tr><td>0013 hex</td><td>High-speed counter 3</td></tr>
<tr><td>0014 hex</td><td>High-speed counter 4</td></tr>
<tr><td>0015 hex</td><td>High-speed counter 5</td></tr>
<tr><td>1000 hex</td><td>PWM(891) output 0</td></tr>
</table>
* Can only be used in CP2E-N30/40/60D□-□.
C: Control Data
<table>
<tr><td>0000 hex</td><td>Reads the PV.</td></tr>
<tr><td>0001 hex</td><td>Reads status.</td></tr>
<tr><td>0002 hex</td><td>Reads range comparison results.</td></tr>
<tr><td>00□3 hex</td><td>P = 0000 or 0001: Reads the output frequency of pulse output 0 or pulse output 1.
C = 0003 hex
P = 0010: Reads the frequency of high-speed counter input 0.
C = 0013 hex: 10-ms sampling method
C = 0023 hex: 100-ms sampling method
C = 0033 hex: 1-s sampling method</td></tr>
</table>
D: First Destination Word

15 0
D: Lower word of PV
D+1: Upper word of PV
2-word PV
Pulse output PV, high-speed counter input PV,
high-speed counter input frequency for high-speed counter input 0

15 0
D: PV
1-word PV
Status, range comparison results</td>
</tr>
</table>

Instruction	Mnemonic	Variations	Symbol/Operand	Function
REGISTER COMPARISON TABLE	CTBL	@	CTBL(882) P C TB **P**: Port specifier **C**: Control data **TB**: First comparison table word	Registers a comparison table and performs comparisons for a PV of high-speed counter 0 to 5. An interrupt task between 0 to 15 will be executed when an execution condition is turned ON.



P: Port specifier

0000 hex	High-speed counter 0
0001 hex	High-speed counter 1
0002 hex	High-speed counter 2
0003 hex	High-speed counter 3
0004 hex	High-speed counter 4
0005 hex	High-speed counter 5

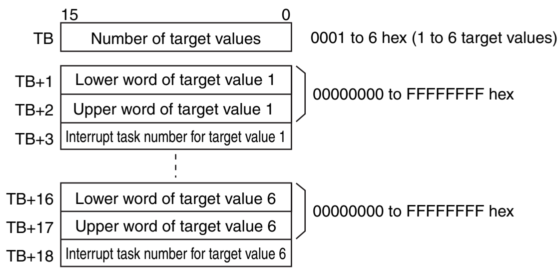
C: Control data

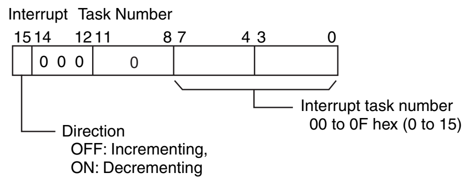
0000 hex	Registers a target value comparison table and starts comparison.
0001 hex	Registers a range comparison table and performs one comparison.
0002 hex	Registers a target value comparison table. Comparison is started with INI(880).
0003 hex	Registers a range comparison table. Comparison is started with INI(880).

TB: First comparison table word

TB is the first word of the comparison table. The structure of the comparison table depends on the type of comparison being performed.

- For target value comparison, the length of the comparison table is determined by the number of target values specified in TB. The table can be between 4 and 19 words long, as shown below.

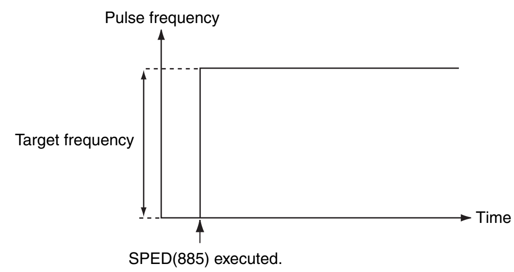


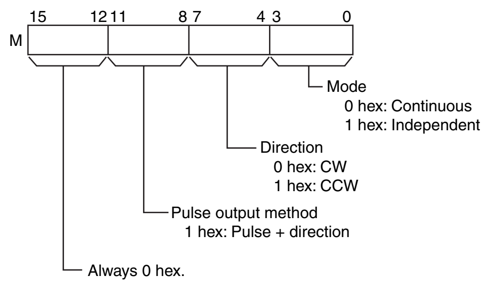


<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr><td>REGISTER COMPARISON TABLE</td><td>CTBL</td><td>@</td><td>CTBL(882)
P
C
TB

P: Port specifier
C: Control data
TB: First comparison table word</td><td>• For range comparison, the comparison table always contains six ranges. The table is 30 words long, as shown below. If it is not necessary to set six ranges, set the interrupt task number to FFFF hex for all unused ranges.
<table>
<tr><th></th><th>15 … 0</th><th></th></tr>
<tr><td>TB</td><td>Lower word of range 1 lower limit</td><td rowspan="2">0000 0000 to FFFF FFFF hex (See note.)</td></tr>
<tr><td>TB+1</td><td>Upper word of range 1 lower limit</td></tr>
<tr><td>TB+2</td><td>Lower word of range 1 upper limit</td><td rowspan="2">0000 0000 to FFFF FFFF hex (See note.)</td></tr>
<tr><td>TB+3</td><td>Upper word of range 1 upper limit</td></tr>
<tr><td></td><td>Range 1 interrupt task number</td><td></td></tr>
<tr><td></td><td>⋮</td><td></td></tr>
<tr><td>TB+25</td><td>Lower word of range 6 lower limit</td><td rowspan="2">0000 0000 to FFFF FFFF hex (See note.)</td></tr>
<tr><td>TB+26</td><td>Upper word of range 6 lower limit</td></tr>
<tr><td>TB+27</td><td>Lower word of range 6 upper limit</td><td rowspan="2">0000 0000 to FFFF FFFF hex (See note.)</td></tr>
<tr><td>TB+28</td><td>Upper word of range 6 upper limit</td></tr>
<tr><td>TB+29</td><td>Range 6 interrupt task number</td><td></td></tr>
</table>
Interrupt task number
0000 to 000F hex: Interrupt task number 0 to 15
AAAA hex: Do not execute interrupt task.
FFFF hex: Ignore the settings for this range.

Note Always set the upper limit greater than or equal to the lower limit for any one range.</td></tr>
<tr><td>SPEED OUT-PUT</td><td>SPED</td><td>@</td><td>SPED(855)
P
M
F

P: Port specifier
M: Output mode
F: First pulse frequency word</td><td>Sets the output pulse frequency for a specific port and starts pulse output without acceleration or deceleration.


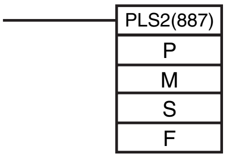
P: Port specifier
<table>
<tr><td>0000 hex</td><td>Pulse output 0</td></tr>
<tr><td>0001 hex</td><td>Pulse output 1</td></tr>
<tr><td>0002 hex</td><td>Pulse output 2*</td></tr>
<tr><td>0003 hex</td><td>Pulse output 3*</td></tr>
</table>
* Can only be used in CP2E-N30/40/60D□-□.
M: Output mode


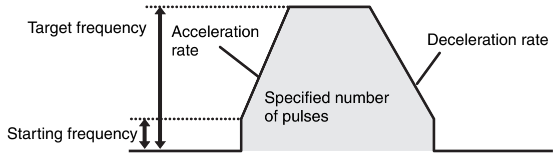
F: First pulse frequency word
<table>
<tr><th></th><th>15 … 0</th><th></th></tr>
<tr><td>F</td><td>Lower word of target frequency</td><td rowspan="2">0 to 100000 Hz
(0000 0000 to 0001 86A0 hex)</td></tr>
<tr><td>F+1</td><td>Upper word of target frequency</td></tr>
</table>
The value of F and F+1 sets the pulse frequency in Hz.</td></tr>
</table>

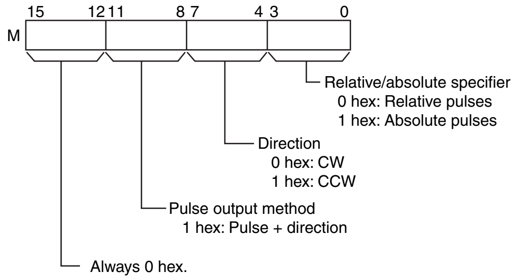
<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr>
<td>SET PULSES</td>
<td>PULS</td>
<td>@</td>
<td>PULS(886)
P
T
N

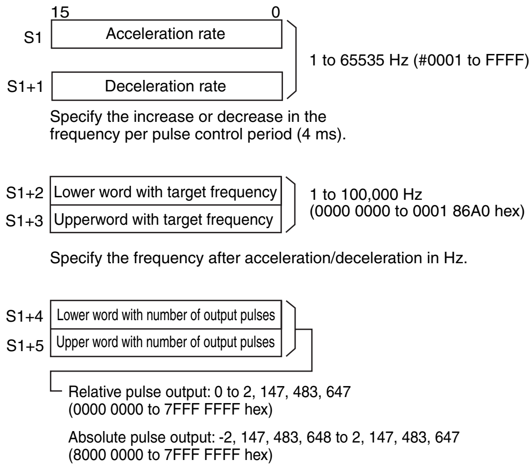
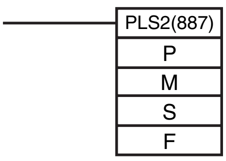
P: Port specifier
T: Pulse type
N: Number of pulses</td>
<td>Sets the number of output pulses.Actual output of the pulses is started later in the program using SPED(885) or ACC(888) in independent mode.
P: Port specifier
<table>
<tr><td>0000 hex</td><td>Pulse output 0</td></tr>
<tr><td>0001 hex</td><td>Pulse output 1</td></tr>
<tr><td>0002 hex</td><td>Pulse output 2*</td></tr>
<tr><td>0003 hex</td><td>Pulse output 3*</td></tr>
</table>
* Can only be used in CP2E-N30/40/60D☐-☐.
T: Pulse type
<table>
<tr><td>0000 hex</td><td>Relative</td></tr>
<tr><td>0001 hex</td><td>Absolute</td></tr>
</table>
N: Number of pulses
<table>
<tr><td></td><td>15   0</td></tr>
<tr><td>N</td><td>Lower word with number of pulses</td></tr>
<tr><td>N+1</td><td>Upper word with number of pulses</td></tr>
</table>
Relative pulse output:
0 to 2, 147, 483, 647, (0000 0000 to 7FFF FFFF hex)

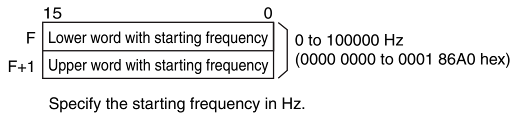
Absolute pulse output:
-2, 147, 483, 648 to 2, 147, 483, 647, (8000 0000 to 7FFF FFFF hex)</td>
</tr>
</table>

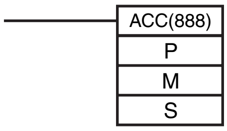
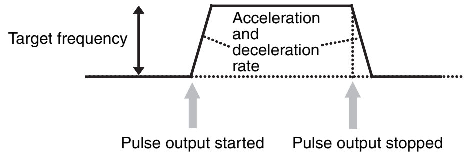
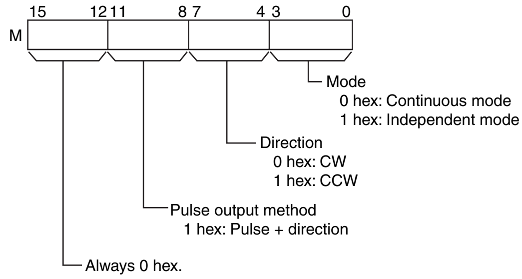
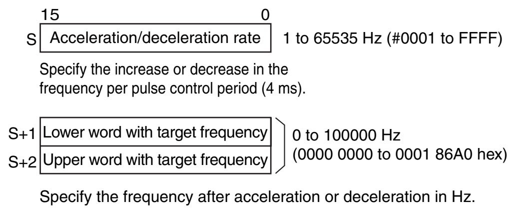
<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr>
<td>PULSE OUT-PUT</td>
<td>PLS2</td>
<td>@</td>
<td>

P: Port specifier
M: Output mode
S: First word of settings table
F: First word of starting frequency</td>
<td>Performs trapezoidal positioning control as the following time chart. Sets the target frequency, starting frequency, acceleration and deceleration rate and direction.


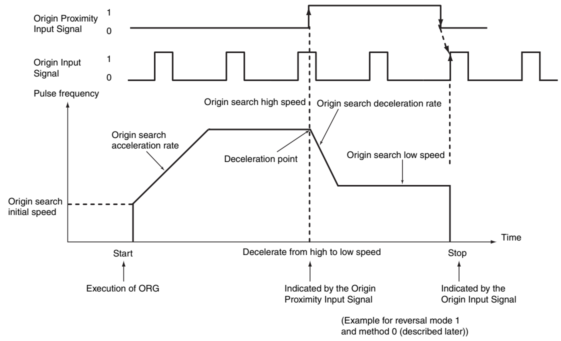
P: Port Specifier
<table>
<tr><td>0000 hex</td><td>Pulse output 0</td></tr>
<tr><td>0001 hex</td><td>Pulse output 1</td></tr>
<tr><td>0002 hex</td><td>Pulse output 2*</td></tr>
<tr><td>0003 hex</td><td>Pulse output 3*</td></tr>
</table>
* Can only be used in CP2E-N30/40/60D□-□.
M: Output Mode


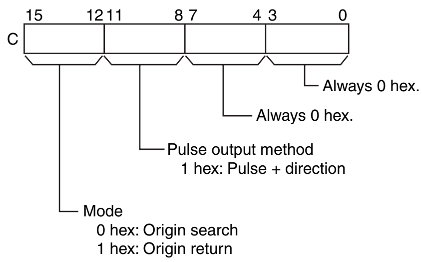
S: First Word of Settings Table

</td>
</tr>
<tr>
<td>PULSE OUT-PUT</td>
<td>PLS2</td>
<td>@</td>
<td>

P: Port specifier
M: Output mode
S: First word of settings table
F: First word of starting frequency</td>
<td>F: First Word of Starting Frequency
The starting frequency is given in F and F+1.

</td>
</tr>
</table>

Instruction	Mnemonic	Variations	Symbol/Operand	Function
ACCELERA-TION CON-TROL	ACC	@	 **P**: Port specifier **M**: Output mode **S**: First word of settings table	Outputs pulses to the specified output port at the specified frequency using the specified acceleration and deceleration rate.  P: Port Specifier 0000 hex: Pulse output 0 0001 hex: Pulse output 1 0002 hex: Pulse output 2* 0003 hex: Pulse output 3* * Can only be used in CP2E-N30/40/60D☐-☐. M: Output Mode  S: First Word of Settings Table 

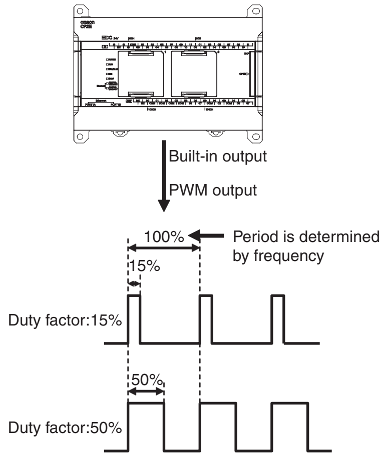
<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr>
<td>ORIGIN SEARCH</td>
<td>ORG</td>
<td>@</td>
<td>ORG(889)
P
C

P: Port specifier
C: Control data</td>
<td>
Performs an origin search or origin return operation.


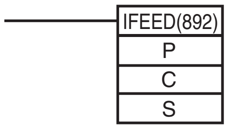
The following parameters must be set in the PLC Setup before ORG(889) can be executed.
<table>
<tr><th>Origin search</th><th>Origin return</th></tr>
<tr><td>
• Origin Search Function Enable/Disable
• Origin Search Operating Mode
• Origin Search Operation Setting
• Origin Detection Method
• Origin Search Direction Setting
• Origin Search/Return Initial Speed
• Origin Search High Speed
• Origin Search Proximity Speed
• Origin Compensation
• Origin Search Acceleration Rate
• Origin Search Deceleration Rate
• Limit Input Signal Type
• Origin Proximity Input Signal Type
• Origin Input Signal Type
• Positioning Monitor Time
</td><td>
• Origin Search/Return Initial Speed
• Origin Return Target Speed
• Origin Return Acceleration Rate
• Origin Return Deceleration Rate
</td></tr>
</table>
P: Port Specifier
<table>
<tr><td>0000 hex</td><td>Pulse output 0</td></tr>
<tr><td>0001 hex</td><td>Pulse output 1</td></tr>
<tr><td>0002 hex</td><td>Pulse output 2*</td></tr>
<tr><td>0003 hex</td><td>Pulse output 3*</td></tr>
</table>
* Can only be used in CP2E-N30/40/60D□-□.
C: Control Data


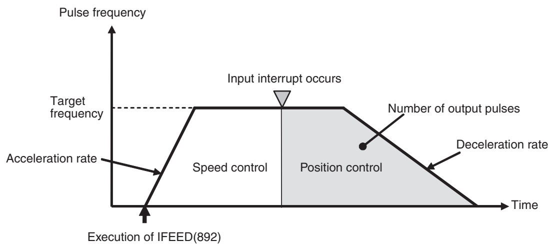
</td>
</tr>
</table>

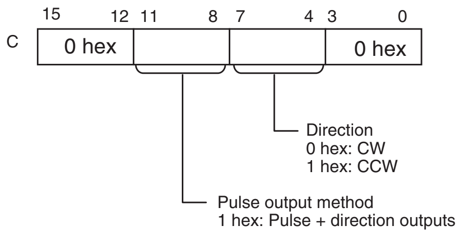
<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr>
<td>PULSE WITH VARIABLE DUTY FACTOR</td>
<td>PWM</td>
<td>@</td>
<td>PWM(891)
P
F
D

P: Port specifier
F: Frequency
D: Duty factor</td>
<td>Outputs pulses with the specified duty factor from the specified port.


P: Port Specifier
<table>
<tr><td>1000 hex</td><td>PWM output 0 (duty factor: in increments of 1%, frequency 0.1 Hz)</td></tr>
<tr><td>1100 hex</td><td>PWM output 0 (duty factor: in increments of 1%, frequency 1 Hz)</td></tr>
</table>
F: Frequency
F specifies the frequency of the PWM output between 2.0 and 6,553.5 Hz (0.1 Hz units, 0014 to FFFF hex), or between 2 and 32,000 Hz (2 Hz units, 0002 to 7D00 hex).
D: Duty Factor
0.0% to 100.0% (0.1% units, 0000 to 03E8 hex)
D specifies the duty factor of the PWM output, i.e., the percentage of time that the output is ON.</td>
</tr>
</table>

<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr>
<td>INTERRUPT FEEDING</td>
<td>IFEED</td>
<td>@</td>
<td>

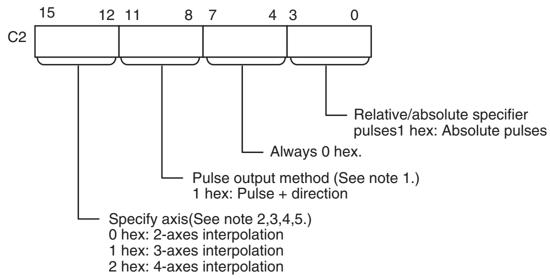
P: Port Specifier
C: Control Data
S: First word of settings table</td>
<td>IFEED(892) uses an input interrupt as a trigger to switch from speed control to position control and move the specified number of pulses.


P: Port Specifier
<table>
<tr><td>0000 hex</td><td>Pulse output 0</td></tr>
<tr><td>0001 hex</td><td>Pulse output 2</td></tr>
<tr><td>0002 hex</td><td>Pulse output 3*</td></tr>
<tr><td>0003 hex</td><td>Pulse output 4*</td></tr>
</table>
* Can only be used in CP2E-N30/40/60D□-□.
C: Control Data


S: First word of settings table
<table>
<tr><td>S</td><td>Acceleration Rate</td><td>1 to 65,535 Hz/4 ms
(0001 to FFFF hex)</td></tr>
<tr><td>S+1</td><td>Deceleration Rate</td><td>1 to 65,535 Hz/4 ms
(0001 to FFFF hex)</td></tr>
<tr><td>S+2</td><td>Lower word of target frequency</td><td rowspan="2">1 to 100,000 Hz
(0000 0001 to 0001 86A0 hex)</td></tr>
<tr><td>S+3</td><td>Upper word of target frequency</td></tr>
<tr><td>S+4</td><td>Lower word with number of pulses</td><td rowspan="2">Number of output pulses:
0 to 2,147,483,647
(0000 0000 to 7FFF FFFF hex)</td></tr>
<tr><td>S+5</td><td>Upper word with number of pulses</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr>
<td>LINEAR INTERPOLA-TION</td>
<td>ITPL</td>
<td>@</td>
<td>ITPL(893)
C1
C2
S1

C1: Port Specifier
C2: Control Data
S1: First word of settings table</td>
<td>
ITPL(893) outputs a 2 to 4 axes linear interpolation to the specified port.

C1: Port Specifier
<table>
<tr><td>0030 hex</td><td>Linear interpolation 0 (Only can be used in CP2E N□□-type CPU Unit)</td></tr>
<tr><td>0031 hex</td><td>Linear interpolation 0 (Only can be used in CP2E N30/40/60D□-□)</td></tr>
</table>
C2: Control Data


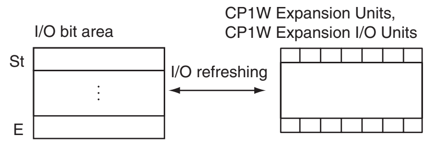
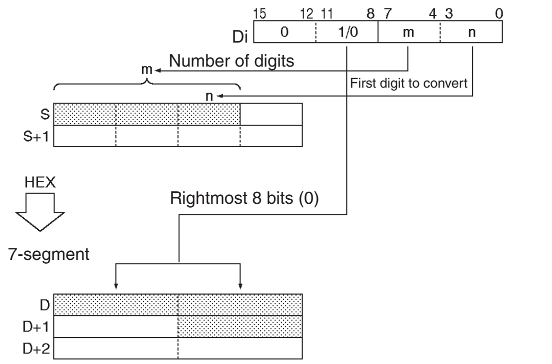
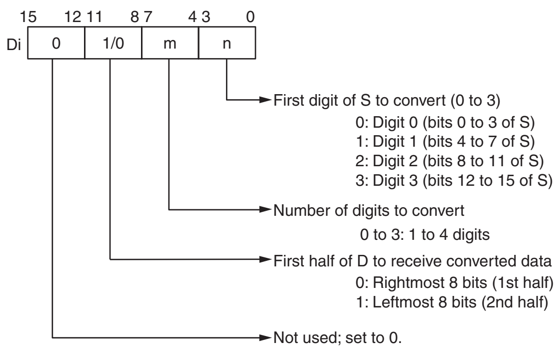
Note 1 Use the same pulse output method when using pulse outputs 1.
2 2-axes interpolation use 0/1 axes in linear interpolation 0 and 2/3 axes in linear interpolation 1 during pulse output.
3 3-axes interpolation can only be executed in linear interpolation 0 and use 0/1/2 axes during pulse output.
4 4-axes interpolation can only be executed in linear interpolation 0 only and use 0/1/2/3 axes during pulse output.
5 CP2E N14/20 CPU Units cannot use 3-axes interpolation or 4-axes interpolation.

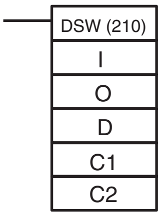
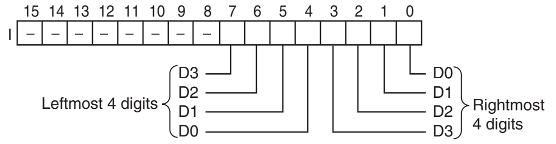
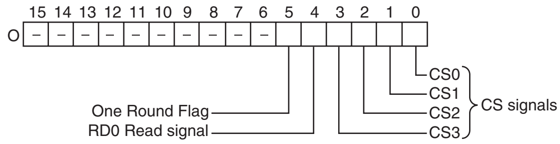
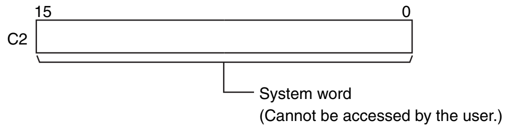
S1: First word of settings table
<table>
<tr><td></td><td>15   0</td><td></td></tr>
<tr><td>S1</td><td>Acceleration rate</td><td rowspan="2">1 to 65535Hz
(0001 to FFFF Hex)</td></tr>
<tr><td>S1+1</td><td>Deceleration rate</td></tr>
<tr><td></td><td colspan="2">Specify the increase or decrease in the frequency per pulse control period (4 ms) in Hz.</td></tr>
<tr><td>S1+2</td><td>Lower word with target frequency</td><td rowspan="2">1 to 100000Hz
(00000001 to 000186A0 Hex)</td></tr>
<tr><td>S1+3</td><td>Upper word with target frequency</td></tr>
<tr><td></td><td colspan="2">Specify the frequency after acceleration in Hz.</td></tr>
<tr><td></td><td>15   0</td><td></td></tr>
<tr><td>S1+4</td><td>Lower word with startup frequency</td><td rowspan="2">0 to 100000Hz
(00000000 to 000186A0 Hex)</td></tr>
<tr><td>S1+5</td><td>Upper word with startup frequency</td></tr>
<tr><td></td><td colspan="2">Specify the frequency during start-up in Hz.</td></tr>
<tr><td>S1+6</td><td>Target position0(L)</td><td rowspan="2">-2147483648 to +2147483647
(80000000 to 7FFFFFFF Hex)</td></tr>
<tr><td>S1+7</td><td>Target position0(H)</td></tr>
<tr><td>S1+8</td><td>Target position1(L)</td><td rowspan="2">-2147483648 to +2147483647
(80000000 to 7FFFFFFF Hex)</td></tr>
<tr><td>S1+9</td><td>Target position1(H)</td></tr>
<tr><td>S1+10</td><td>Target position2(L)</td><td rowspan="2">-2147483648 to +2147483647
(80000000 to 7FFFFFFF Hex)</td></tr>
<tr><td>S1+11</td><td>Target position2(H)</td></tr>
<tr><td>S1+12</td><td>Target position3(L)</td><td rowspan="2">-2147483648 to +2147483647
(80000000 to 7FFFFFFF Hex)</td></tr>
<tr><td>S1+13</td><td>Target position3(H)</td></tr>
</table>
</td>
</tr>
</table>

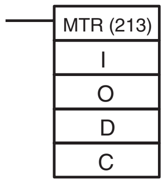
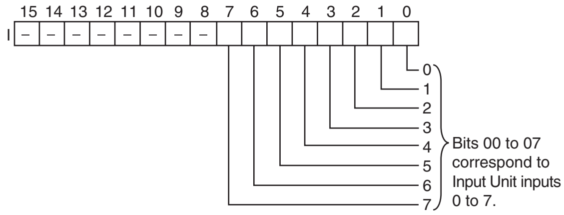
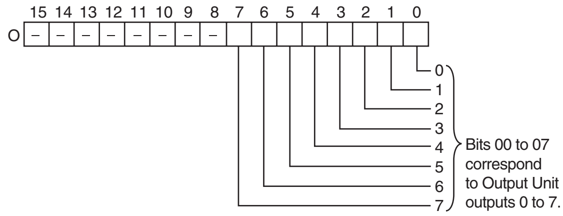
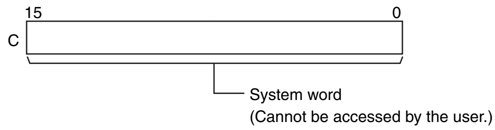
A-1-19 Step Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
STEP DEFINE	STEP	---	When defining the beginning of a step, a control bit is specified as follows: STEP(008) B **B**: Bit When defining the end of a step, a control bit is not specified as follows: STEP(008)	STEP(008) functions in following 2 ways, depending on its position and whether or not a control bit has been specified. (1)Starts a specific step. (2)Ends the step programming area (i.e., step execution).
STEP START	SNXT	---	SNXT(009) B **B**: Bit	SNXT(009) is used in the following three ways: (1)To start step programming execution. (2)To proceed to the next step control bit. (3)To end step programming execution.

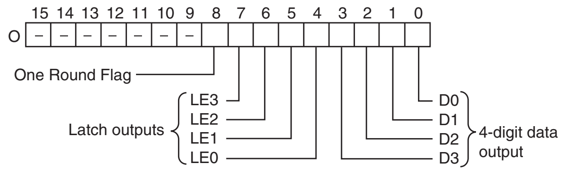
A-1-20 Basic I/O Unit Instructions

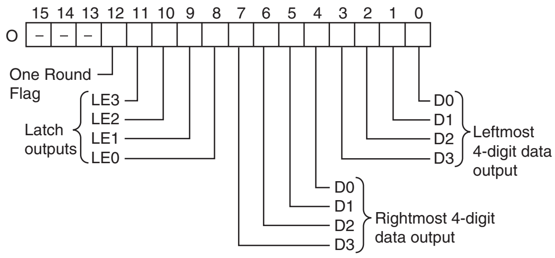
Instruction	Mnemonic	Variations	Symbol/Operand	Function
I/O REFRESH	IORF	@	IORF(097) St E **St**: Starting word **E**: End word	Refreshes the specified I/O words. 
7-SEGMENT DECODER	SDEC	@	SDEC(078) S Di D **S**: Source word **Di**: Digit designator **D**: 1st destination word	Converts the hexadecimal contents of the designated digit(s) into 8-bit, 7-segment display code and places it into the upper or lower 8-bits of the specified destination words.  Di: Digit designator 

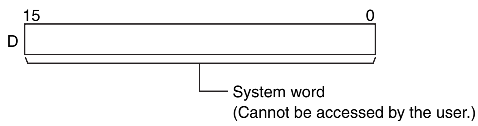
Instruction	Mnemonic	Variations	Symbol/Operand	Function
DIGITAL SWITCH INPUT	DSW	---	 **I**: Data input word (D0 to D3) **O**: Output word **D**: 1st result word **C1**: Number of digits **C2**: System word	Reads the value set on an external digital switch (or thumbwheel switch) connected to an Input Unit or Output Unit and stores the 4-digit or 8-digit BCD data in the specified words. I: Input Word (Data Line D0 to D3 Inputs) Specify the input word allocated to the Input Unit and connect the digital switch's D0 to D3 data lines to the Input Unit as shown in the following diagram.  O: Output Word (CS/RD Control Signal Outputs) Specify the output word allocated to the Output Unit and connect the digital switch's control signals (CS and RD signals) to the Output Unit as shown in the following diagram.  C1: Number of Digits Specifies the number of digits that will be read from the external digital switch. Set C1 to 0000 hex to read 4 digits or 0001 hex to read 8 digits. C2: System Word Specifies a work word used by the instruction. This word cannot be used in any other application. 

Instruction	Mnemonic	Variations	Symbol/Operand	Function
MATRIX INPUT	MTR	---	 **I**: Data input word **O**: Output word **D**: 1st destination word **C**: System word	Inputs up to 64 signals from an 8 × 8 matrix connected to an Input Unit and Output Unit (using 8 input points and 8 output points) and stores that 64-bit data in the 4 destination words. I: Input Word Specify the input word allocated to the Input Unit and connect the 8 input signal lines to the Input Unit as shown in the following diagram.  O: Output Word (Selection Signal Outputs) Specify the output word allocated to the Output Unit and connect the 8 selection signals to the Output Unit as shown in the following diagram.  C: System Word Specifies a work word used by the instruction. This word cannot be used in any other application. 

<table>
<tr><th>Instruction</th><th>Mnemonic</th><th>Variations</th><th>Symbol/Operand</th><th>Function</th></tr>
<tr>
<td>7-SEGMENT DISPLAY OUT-PUT</td>
<td>7SEG</td>
<td>---</td>
<td>7SEG (214)
S
O
C
D

S: 1st source word
O: Output word
C: Control data
D: System word</td>
<td>Converts the source data (either 4-digit or 8-digit BCD) to 7-segment display data, and outputs that data to the specified output word.
O: Output Word (Data and Latch Outputs)
Specify the output word allocated to the Output Unit and connect the 7-segment display to the Output Unit as shown in the following diagram.
• Converting 4 digits


• Converting 8 digits


C: Control Data
The value of C indicates the number of digits of source data and the logic for the Input and Output Units, as shown in the following table. (The logic refers to the transistor output's NPN or PNP logic.)
<table>
<tr><th>Source data</th><th>Display's data input logic</th><th>Display's latch input logic</th><th>C</th></tr>
<tr><td rowspan="4">4 digits (S)</td><td rowspan="2">Same as Output Unit</td><td>Same as Output Unit</td><td>0000</td></tr>
<tr><td>Different from Output Unit</td><td>0001</td></tr>
<tr><td rowspan="2">Different from Output Unit</td><td>Same as Output Unit</td><td>0002</td></tr>
<tr><td>Different from Output Unit</td><td>0003</td></tr>
<tr><td rowspan="4">8 digits (S, S+1)</td><td rowspan="2">Same as Output Unit</td><td>Same as Output Unit</td><td>0004</td></tr>
<tr><td>Different from Output Unit</td><td>0005</td></tr>
<tr><td rowspan="2">Different from Output Unit</td><td>Same as Output Unit</td><td>0006</td></tr>
<tr><td>Different from Output Unit</td><td>0007</td></tr>
</table>
D: System Word
Specifies a work word used by the instruction. This word cannot be used in any other application.

</td>
</tr>
</table>

A-1 Instruction Functions

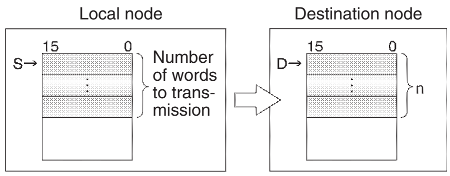
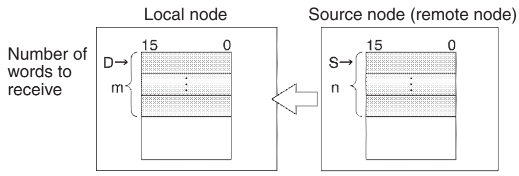
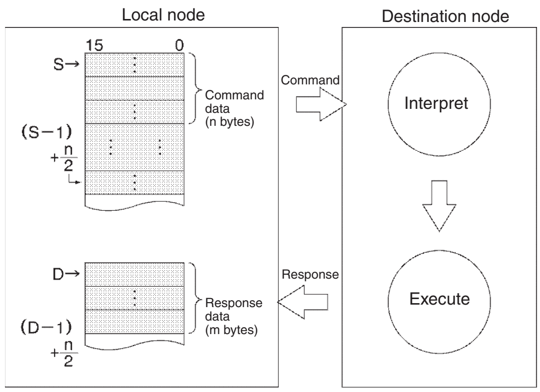
App

A-1-20 Basic I/O Unit Instructions

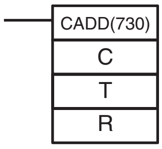
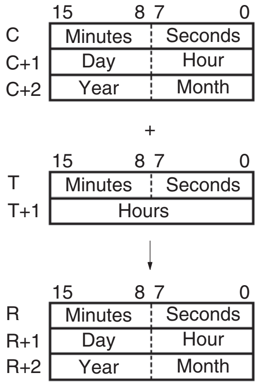
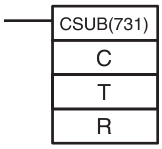
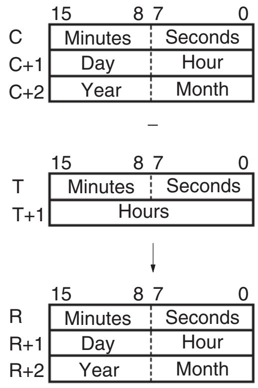
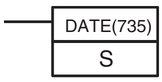
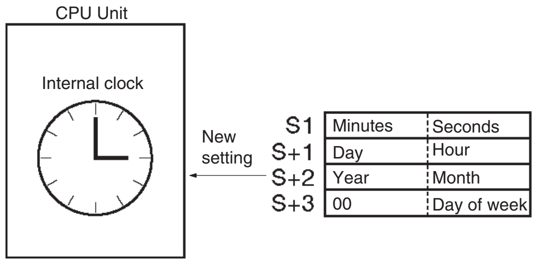
A-1-21 Serial Communications Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
TRANSMIT	TXD	@	TXD(236) S C N **S**: 1st source word **C**: Control word **N**: Number of bytes 0000 to 0100 hex (0 to 256 decimal)	Outputs the specified number of bytes of data without conversion from the RS-232C port or RS-485 port built into the CPU Unit or the serial Option Board according to the start code and end code specified for no-protocol mode in the PLC Setup. C: Control word C: bits 15 12 11 8 7 4 3 0 Byte order 0: Most significant bytes first 1: Least significant bytes first RS and ER signal control 0: No RS and ER signal control 1: RS signal control 2: ER signal control 3: RS and ER signal control Always 0 Serial port specifier 1: Built-in RS-232C port/Serial option port 1 2: Built-in RS485 port/Serial option port 2 3: Serial option port 1 (EX)
RECEIVE	RXD	@	RXD(235) D C N **D**: 1st destination word **C**: Control word **N**: Number of bytes to store 0000 to 0100 hex (0 to 256 decimal)	Reads the specified number of bytes of data starting with the specified first word from the RS-232C port or RS-485 port built into the CPU Unit or the serial Option Board according to the start code and end code specified for no-protocol mode in the PLC Setup. C: Control Word C: bits 15 12 11 8 7 4 3 0 Byte order 0 Hex: Most significant byte to least significant byte 1 Hex: Lest significant byte to most significant byte Always 0 CS and DR signal monitoring 0: No CS and DR signal monitoring 1: CS signal monitoring 2: DR signal monitoring 3: CS and DR signal monitoring Serial port specifier 1: Built-in RS-232C port/Serial option port 1 2: Built-in RS485 port/Serial option port 2 3: Serial option port 1 (EX)

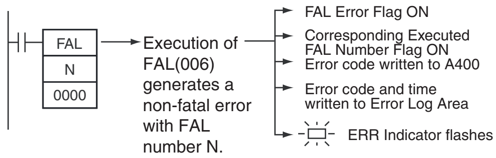
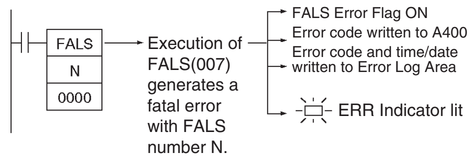
A-1-22 Network Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
NETWORD SEND	SEND	@	SEND(090) S D C **S**: First source word (local node) **D**: First destination word (remote node) **C**: First control word	Sends data to a node in the Ethernet network. 
NETWORD RECEIVE	RECV	@	RECV(098) S D C **S**: First source word (remote node) **D**: First destination word (local node) **C**: First control word	Requests data to be transmitted from a node in the Ethernet network and receives the data. 
DELIVER COMMAND	CMND	@	CMND(490) S D C **S**: First command word **D**: First response word **C**: First control word	Sends an FINS command and receives the response. 

A-1-23 Clock Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
CALENDAR ADD	CADD	@	 **C**: 1st calendar word **T**: 1st time word **R**: 1st result word	Adds time to the calendar data in the specified words. 
CALENDAR SUBTRACT	CSUB	@	 **C**: 1st calendar word **T**: 1st time word **R**: 1st result word	Subtracts time from the calendar data in the specified words. 
CLOCK ADJUSTMENT	DATE	@	 **S**: 1st source word	Changes the internal clock setting to the setting in the specified source words. 

A-1-24 Failure Diagnosis Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
FAILURE ALARM	FAL	@	FAL(006) N S **N**: FAL number **S**: 1st message word or error code to generate	Generates or clears user-defined non-fatal errors. Non-fatal errors do not stop PC operation. Also generates non-fatal errors with the system. 
SEVERE FAILURE ALARM	FALS	---	FALS(007) N S **N**: FALS number **S**: 1st message word or error code to generate	Generates user-defined fatal errors. Fatal errors stop PC operation. Also generates fatal errors with the system. 

A-1-25 Other Instructions

Instruction	Mnemonic	Variations	Symbol/Operand	Function
SET CARRY	STC	@	STC(040)	Sets the Carry Flag (CY).
CLEAR CARRY	CLC	@	CLC(041)	Turns OFF the Carry Flag (CY).
EXTEND MAXIMUM CYCLE TIME	WDT	@	WDT(094) T **T**: Timer setting	Extends the maximum cycle time, but only for the cycle in which this instruction is executed.

A-2 Auxiliary Area Allocations by Address

The following table lists the data provided in the Auxiliary Area in order of the addresses of the data.

A-2-1 Read-only Words

Address Words	Address Bits	Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
A0	---	10-ms Incrementing Free Running Timer	This word contains the system timer used after the power is turned ON. 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 ms), and then continues to be automatically incremented by 1 every 10 ms. **Note** The timer will continue to be incremented when the operating mode is switched to RUN mode. 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.		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. 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 ms), and then continues to be automatically incremented by 1 every 100 ms. **Note** The timer will continue to be incremented when the operating mode is switched to RUN mode. 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.		Retained	Cleared	Every 100 ms after power is turned ON	
A40 to A44	---	Ethernet Communication Error Log	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 most recent errors can be stored. Each error record occupies 5 words. A40: Error code (bits 0 to 15) A41: Error contents (bits 0 to 15) A42: Minutes (upper byte), Seconds (lower byte) A43: Day of month (upper byte), Hours (lower byte) A44: Year (upper byte), Month (lower byte)	For details of the error code and the error contents, refer to *6-1-11 Other Error* in the *CP2E CPU Unit Hardware User's Manual* (Cat. No. W613).	Retained	Retained	Refreshed when error occurs.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A45	14	Ethernet Link Status Flag (N14/20 Ethernet port, N30/40/60 Ethernet PORT1A)	Ethernet port of N14/20 CPU unit and port A of N30/40/60 CPU unit will be ON when the Ethernet link is established.	ON: Ethernet link is established OFF: Ethernet link is terminated	---	Cleared	Refreshed when the Ethernet link status is changed.	
	15	Ethernet Link Status Flag (N30/40/60 Ethernet PORT1B)	Port B of N30/40/60 CPU unit will be ON when the Ethernet link is established.					
A46	02	IP Address Setting Erro Flag	ON when the IP address meets the following conditions: • All bits of the host ID are 0 or 1 • All bits of the net ID are 0 or 1 • All bits of the subnet ID are 1 • IP address starts with 127 (0x7F)	ON: Error OFF: Normal	---	Cleared	Refreshed when error occurs	
	03	IP Address Table Error Flag	ON when the IP address table is abnormal.	ON: Error OFF: Normal	---	Cleared		
	04	IP Router Table Error Flag	ON when the IP router table is abnormal.	ON: Error OFF: Normal	---	Cleared		
	05	DNS Server Error Flag	ON when DNS server meets the following conditions: • The IP address of the server is invalid • Time-out occurs in server communi-cation	ON: Error OFF: Normal	---	Cleared		
	06	Routing Table Error Flag	ON when the routing table setting is abnormal.	ON: Error OFF: Normal	---	Cleared		
	11	SNTP Server Error Flag	ON when SNTP server meets the following conditions: • The IP address of the server or the host name is invalid • Time-out occurs in servecr commu-nication	ON: Error OFF: Normal	---	Cleared		
	14	Address Disagreement Flag	ON when meets the following conditions: • IP address conversion method is set to Automatic • Host ID of your IP address and the FINS node address do not match	ON: Disagree OFF: Agree	---	Cleared		
A47	00	FINS/TCP Connection Establish Flag (No.1)	ON when FINS/TCP connection is established.	ON: Connection Established OFF: Connection Disconnected	---	Cleared	When connection state changes	
	01	FINS/TCP Connection Establish Flag (No.2)						
	02	FINS/TCP Connection Establish Flag (No.3)						

Address: Words	Address: Bits	Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
A50	04	Serial Port 1 (EX) Communication Error Flag	ON when a communication error has occurred at the Serial Port 1 (EX).	ON: Error OFF: Normal	Retained	Cleared	Refreshed when error occurs	
	05	Serial Port 1 (EX) Send Ready Flag (No-protocol Mode)	ON when the Serial Port 1 (EX) is able to send data in no-protocol mode.	ON: Able-to-send OFF: Unable-to-send	Retained	Cleared	Written after transmission.	
	06	Serial Port 1 (EX) Reception Completed Flag (No-protocol Mode)	ON when the Serial Port 1 (EX) has completed the reception in no-protocol mode. • When the number of bytes was specified: ON when the specified number of bytes is received. • 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 transmission.	
	07	Serial Port 1 (EX) Reception Overflow Flag (No-protocol Mode)	• ON when a data overflow occurred during reception through the Serial Port 1 (EX) in no-protocol mode. • When the number of bytes was specified: ON when more data is received after the reception was completed but before RXD was executed. • When the end code was specified: ON when more data is received after the end code was received but before RXD was executed. • ON when 257 bytes are received before the end code.	ON: Overflow OFF: No overflow	Retained	Cleared	Written after transmission.	
A51	00 to 07	Serial Port 1(EX) Polled Unit Communications Flags	The corresponding bit will be ON when the Serial Port 1(EX) is communicating with Serial PLC Link mode. 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.	
	00 to 15	Serial Port 1(EX) Reception Counter (No-protocol mode only)	Indicates (in binary) the number of bytes of data received when the serial port 1(EX) is in no-protocol mode.		Retained	Cleared	Refreshed when data is received.	
A52	---	Pulse Output 2 PV: Lower four digits	Contain the number of pulses output from the corresponding pulse output port. PV range: 8000 0000 to 7FFF FFFF hex (-2,147,483,648 to 2,147,483,647) When pulses are being output in the CW direction, the PV is incremented by 1 for each pulse. When pulses are being output in the CCW direction, the PV is decremented by 1 for each pulse. PV after overflow: 7FFF FFFF hex PV after underflow: 8000 0000 hex **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, ACC, PLS2, ITPL or IFEED) is executed.		Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when the INI instruction is executed (PV change).	
A53		Pulse Output 2 PV: Upper four digits						
A54	---	Pulse Output 3 PV: Lower four digits						
A55		Pulse Output 3 PV: Upper four digits						

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A56	00	Pulse Output 2 Accel/Decel Flag	This flag will be ON when pulses are being output from pulse output 2 according to an ORG, ACC, PLS2, ITPL or IFEED instruction and the output frequency is being changed in steps (accelerating or decelerating).	ON: Accelerating or decelerating OFF: Constant speed	---	Cleared	Refreshed each cycle during the overseeing processes.	
	01	Pulse Output 2 Overflow/Underflow Flag	This flag indicates when an overflow or underflow has occurred in the pulse output 2 PV.	ON: Overflow or underflow OFF: Normal	---	Cleared	• Refreshed when the PV is changed by the INI instruction. • Refreshed when an overflow or underflow occurs.	
	02	Pulse Output 2 Output Amount Set Flag	ON when the number of output pulses for pulse output 2 has been set with the PULS instruction.	ON: Setting made OFF: No setting	---	Cleared	• Refreshed when the PULS instruction is executed. • Refreshed when pulse output stops.	
	03	Pulse Output 2 Output Completed Flag	ON when the number of output pulses set with the PULS, PLS2, ITPL or IFEED instruction has been output through pulse output 2.	ON: Output completed OFF: Output not completed	---	Cleared	Refreshed at the start or completion of pulse output.	
	04	Pulse Output 2 Output In-progress Flag	ON when pulses are being output from pulse output 2.	ON: Outputting pulses OFF: Stopped	---	Cleared	Refreshed when pulse output starts or stops.	
	05	Pulse Output 2 No-origin Flag	ON when the origin has not been determined for pulse output 2 and goes OFF when the origin has been determined.	ON: Origin not established OFF: Origin established	ON	ON	Refreshed each cycle during the overseeing processes.	
	06	Pulse Output 2 At-origin Flag	ON when the pulse output 2 PV matches the origin (0).	ON: Stopped at origin OFF: Not stopped at origin	---	Cleared	Refreshed each cycle during the overseeing processes.	
	07	Pulse Output 2 Output Stopped Error Flag	ON when an error occurred while out putting pulses in the pulse output 2 origin search function. The Pulse Output 2 Output Stop Error code will be written to A438.	ON: Stop error occurred OFF: No error	---	Cleared	• Refreshed when origin search starts. • Refreshed when a pulse output stop error occurs.	A438
	08	Pulse Output 2 Interrupt Feeding Flag	ON between Input Interrupt - decelerating to stop when Interrupt Feeding function (IFEED instruction) is executing.	ON: Feeding OFF: Not feeding	Cleared	Cleared	• Refreshed when feed-ing is com-pleted. • Refreshed when inter-rupt feeding is executing.	
	09	Pulse Output 2 Interrupt Feeding Error Flag	ON if an overflow or underflow error occurs during input interrupt or specified number of pulse output, when Interrupt Feeding function (IFEED instruction) is executing.	ON: Overflow or underflow OFF: Normal	Cleared	Cleared	Refreshed when interrupt feeding is executing.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A57	00	Pulse Output 3 Accel/Decel Flag	This flag will be ON when pulses are being output from pulse output 3 according to an ORG, ACC, PLS2, ITPL or IFEED instruction and the output frequency is being changed in steps (accelerating or decelerating).	ON: Accelerating or decelerating OFF: Constant speed	---	Cleared	Refreshed each cycle during the overseeing processes.	
	01	Pulse Output 3 Overflow/Underflow Flag	This flag indicates when an overflow or underflow has occurred in the pulse output 3 PV.	ON: Overflow or underflow OFF: Normal	---	Cleared	• Refreshed when the PV is changed by the INI instruction. • Refreshed when an overflow or underflow occurs.	
	02	Pulse Output 3 Output Amount Set Flag	ON when the number of output pulses for pulse output 3 has been set with the PULS instruction.	ON: Setting made OFF: No setting	---	Cleared	• Refreshed when the PULS instruction is executed. • Refreshed when pulse output stops.	
	03	Pulse Output 3 Output Completed Flag	ON when the number of output pulses set with the PULS, PLS2, ITPL or IFEED instruction has been output through pulse output3.	ON: Output completed OFF: Output not completed	---	Cleared	Refreshed at the start or completion of pulse output.	
	04	Pulse Output 3 Output In-progress Flag	ON when pulses are being output from pulse output 3.	ON: Outputting pulses OFF: Stopped	---	Cleared	Refreshed when pulse output starts or stops.	
	05	Pulse Output 3 No-origin Flag	ON when the origin has not been determined for pulse output 3 and goes OFF when the origin has been determined.	ON: Origin not established OFF: Origin established	ON	ON	Refreshed each cycle during the overseeing processes.	
	06	Pulse Output 3 At-origin Flag	ON when the pulse output 3 PV matches the origin (0).	ON: Stopped at origin OFF: Not stopped at origin	---	Cleared	Refreshed each cycle during the overseeing processes.	
	07	Pulse Output 3 Output Stopped Error Flag	ON when an error occurred while out putting pulses in the pulse output 3 origin search function. The Pulse Output 3 Output Stop Error code will be written to A439.	ON: Stop error occurred OFF: No error	---	Cleared	• Refreshed when origin search starts. • Refreshed when a pulse output stop error occurs.	A439
	08	Pulse Output 3 Interrupt Feeding Flag	ON between Input Interrupt - decelerating to stop when Interrupt Feeding function (IFEED instrnction) is executing.	ON: Feeding OFF: Not feeding	Cleared	Cleared	• Refreshed when feed-ing is com-pleted. • Refreshed when inter-rupt feeding is executing.	
	09	Pulse Output 3 Interrupt Feeding Error Flag	ON if an overflow or unlerflow error occurrs during input interrupt or specified number of pulse output, when Interrupt Feeding function (IFEED instrnction) is executing.	ON: Overflow or underflow OFF: Normal	Cleared	Cleared	Refreshed when interrupt feeding is executing.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A99	00	UM Read Protection Status	Indicates whether all of the ladder programs in the PLC are read-protected.	ON: UM read-protected. OFF: UM not read-protected	Retained	Retained	When protection is set or cleared	
A100 to A199	---	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. Each error record occupies 5 words; the function of these 5 words is as follows: First word: Error code (bits 0 to 15) First word + 1: Error contents (bits 0 to 15) First word + 2: Minutes (upper byte), Seconds (lower byte) First word + 3: Day of month (upper byte), Hours (lower byte) First word + 4: Year (upper byte), Month (lower byte) **Note 1** Errors generated by FAL(006) and FALS(007) will also be stored in this Error Log. **2** The Error Log Area can be reset from the CX-Programmer. **3** 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. **4** In an E□□-type CPU Unit, the data will be for 1:01.01 on Sunday January 1, 2001.	Error code Error contents: Address of Aux. Area word with details or 0000 hex if there is no related word. 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, BCD	Retained	Retained	Refreshed when error occurs.	A500.14 A300 A400
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	ON	Cleared	---	
	12	Step Flag	ON for one cycle when step execution is started with STEP. This flag can be used for initialization processing at the beginning of a step.	ON for the first cycle after execution of STEP.	Cleared	Cleared	---	
	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. **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.	ON: ON for first cycle (including transitions from WAIT and IN) OFF: Other	Cleared	Cleared	---	
	15	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 or not being executed for the first time.	Cleared	Cleared	---	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A202	00 to 07	Communications Port Enabled Flags	ON when a communication instruction (SEND, RECV, or CMND) can be executed with the corresponding port number. Each bit corresponds to a communication port number. Bits 00 to 07 correspond to communications ports 0 to 7.	ON: Network communication is not being executed OFF: Network communication is being executed	---	Cleared	• Refreshed when communications finished. • Refreshed when instruction is executed.	
	15	Network Communications Port Allocation Enabled Flag	ON when a communication instruction (SEND, RECV, or CMND) can be executed with the automatic allocation function.	ON: Communication port available OFF: Communication port not available	---			
A203 to A210	---	Network Communications Completion Codes	A response code is stored when a network communication instruction (SEND, RECV, or CMND) is executed. Each word corresponds to a port number. Words A203 to A210 correspond to communications ports 0 to 7.	Except 0000: Error code 0000: Normal condition	---	Cleared	Refreshed when instruction is executed.	
A214	00 to 07	First Cycle Flags after Network Communications Finished	Each flag will turn ON for just one cycle when a communications instruction is executed using automatic communication port allocations. Each bit 00 to 07 corresponds to communication port 0 to 7.	ON: First cycle after communications finish only OFF: Other status	---	Cleared	Refreshed when communications finished.	
A215	00 to 07	First Cycle Flags after Network Communications Error	Each flag will turn ON for just one cycle when a communications instruction is executed using automatic communication port allocations and an error occurs at the end of communication. Each bit 00 to 07 corresponds to communication port 0 to 7. If the flag is 1 (ON), please identify the cause of error refer to the network communication response code (A203 to A210).	ON: First cycle after communications error only OFF: Other status	---	Cleared	Refreshed when communications finished.	
A216 to A217	---	Network Communications Completion Code Storage Address	The communication response code is stored and automatically set to any channel address of the auxiliary relay when a communications instruction is executed using automatic communication port allocations.		---	Cleared	Refreshed when instruction is executed.	
A218	---	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	---	Cleared	Refreshed when instruction is executed	
A219	00 to 07	Network Communications Error Flags	ON when an error occurred during execution of a network instruction (SEND, RECV, CMND, or PMCR). Bits 00 to 07 correspond to communications ports 0 to 7.	ON: Error occurred OFF: Normal condition	---	Cleared	Refreshed when instruction is executed	
A262 to A263	---	Maximum Cycle Time	These words contain the maximum cycle time since the start of PLC operation. The cycle time is recorded in 32-bit binary. The upper digits are in A263 and the lower digits are in A262.	0 to FFFFFFFF: 0 to 429,496,729.5 ms (0.1-ms units)	Cleared	Cleared	Refreshed when updating the maximum cycle time.	
A264 to A265	---	Present Cycle Time	These words contain the present cycle time. The cycle time is recorded in 32-bit binary. The upper digits are in A265 and the lower digits are in A264.	0 to FFFFFFFF: 0 to 429,496,729.5 ms (0.1-ms units)	Cleared	Cleared	Refreshed each cycle during the overseeing processes.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A270 to A271	---	High-speed Counter 0 PV	Contains the PV of high-speed counter 0. A271 contains the upper 4 digits and A270 contains the lower 4 digits.		Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read the PV. • Refreshed when INI instruction is executed.	
A272 to A273	---	High-speed Counter 1 PV	Contains the PV of high-speed counter 1. A273 contains the upper 4 digits and A272 contains the lower 4 digits.		Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read the PV. • Refreshed when INI instruction is executed.	
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.	ON: PV in range OFF: PV not in range	Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read the results of range comparison. • Refreshed when range comparison table is registered.	
	01	High-speed Counter 0 Range 2 Comparison Condition Met Flag						
	02	High-speed Counter 0 Range 3 Comparison Condition Met Flag						
	03	High-speed Counter 0 Range 4 Comparison Condition Met Flag						
	04	High-speed Counter 0 Range 5 Comparison Condition Met Flag						
	05	High-speed Counter 0 Range 6 Comparison Condition Met Flag						
	08	High-speed Counter 0 Comparison In-progress Flag	This flag indicates whether a comparison operation is being executed for high-speed counter 0.	ON: Being executed OFF: Stopped	---	Cleared	Refreshed when comparison operation starts or stops.	
	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.)	ON: Overflow or underflow OFF: Normal	Cleared	Cleared	• Refreshed when an overflow or underflow occurs. • Refreshed when PV is changed.	

Address: Words	Address: Bits	Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
A274	10	High-speed Counter 0 Count Direction	This flag indicates whether the high-speed counter 0 is currently being incremented or decremented. The counter PV for the current cycle is compared with the PV in last cycle to determine the direction.	ON: Incrementing OFF: Decrementing	---	Cleared	Setting used for high-speed counter, valid during counter operation.	
A275	00	High-speed Counter 1 Range 1 Comparison Condition Met Flag	These flags indicate whether the PV is within the specified ranges when high-speed counter 1 is being operated in range-comparison mode for upper and lower limits.	ON: PV in range OFF: PV not in range	Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read the results of range comparison. • Refreshed when range comparison table is registered.	
	01	High-speed Counter 1 Range 2 Comparison Condition Met Flag						
	02	High-speed Counter 1 Range 3 Comparison Condition Met Flag						
	03	High-speed Counter 1 Range 4 Comparison Condition Met Flag						
	04	High-speed Counter 1 Range 5 Comparison Condition Met Flag						
	05	High-speed Counter 1 Range 6 Comparison Condition Met Flag						
	08	High-speed Counter 1 Comparison In-progress Flag	This flag indicates whether a comparison operation is being executed for high-speed counter 1.	ON: Being executed OFF: Stopped	---	Cleared	Refreshed when comparison operation starts or stops.	
	09	High-speed Counter 1 Overflow/Underflow Flag	This flag indicates when an overflow or underflow has occurred in the high-speed counter 1 PV. (Used with the linear mode counting range only.)	ON: Overflow or underflow OFF: Normal	Cleared	Cleared	• Refreshed when an overflow or underflow occurs. • Refreshed when PV is changed.	
	10	High-speed Counter 1 Count Direction	This flag indicates whether the high-speed counter 1 is currently being incremented or decremented. The counter PV for the current cycle is compared with the PV in last cycle to determine the direction.	ON: Incrementing OFF: Decrementing	---	Cleared	Setting used for high-speed counter, valid during counter operation.	
A276	---	Pulse Output 0 PV: Lower four digits	Contain the number of pulses output from the corresponding pulse output port. PV range: 8000 0000 to 7FFF FFFF hex (-2,147,483,648 to 2,147,483,647) When pulses are being output in the CW direction, the PV is incremented by 1 for each pulse. When pulses are being output in the CCW direction, the PV is decremented by 1 for each pulse. PV after overflow: 7FFF FFFF hex PV after underflow: 8000 0000 hex **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, ACC, PLS2, ITPL or IFEED) is executed.		Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when the INI instruction is executed (PV change).	
A277		Pulse Output 0 PV: Upper four digits						
A278	---	Pulse Output 1 PV: Lower four digits						
A279		Pulse Output 1 PV: Upper four digits						

Address Words	Address Bits	Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
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 ORG, ACC, PLS2, ITPL or IFEED instruction and the output frequency is being changed in steps (accelerating or decelerating).	ON: Accelerating or decelerating OFF: Constant speed	---	Cleared	Refreshed each cycle during the overseeing processes.	
	01	Pulse Output 0 Overflow/Underflow Flag	This flag indicates when an overflow or underflow has occurred in the pulse output 0 PV.	ON: Overflow or underflow OFF: Normal	---	Cleared	• Refreshed when the PV is changed by the INI instruction. • Refreshed when an overflow or underflow occurs.	
	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 instruction.	ON: Setting made OFF: No setting	---	Cleared	• Refreshed when the PULS instruction is executed. • Refreshed when pulse output stops.	
	03	Pulse Output 0 Output Completed Flag	ON when the number of output pulses set with the PULS, PLS2, ITPL or IFEED instruction has been output through pulse output 0.	ON: Output completed OFF: Output not completed	---	Cleared	Refreshed at the start or completion of pulse output.	
	04	Pulse Output 0 Output In-progress Flag	ON when pulses are being output from pulse output 0.	ON: Outputting pulses OFF: Stopped	---	Cleared	Refreshed when pulse output starts or stops.	
	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.	ON: Origin not established OFF: Origin established	ON	ON	Refreshed each cycle during the overseeing processes.	
	06	Pulse Output 0 At-origin Flag	ON when the pulse output 0 PV matches the origin (0).	ON: Stopped at origin OFF: Not stopped at origin	---	Cleared	Refreshed each cycle during the overseeing processes.	
	07	Pulse Output 0 Output Stopped Error Flag	ON when an error occurred while outputting pulses in the pulse output 0 origin search function. The Pulse Output 0 Output Stop Error code will be written to A444.	ON: Stop error occurred OFF: No error	---	Cleared	• Refreshed when origin search starts. • Refreshed when a pulse output stop error occurs.	A444
	08	Pulse Output 0 Interrupt Feeding Flag	ON between Input Interrupt - decelerating to stop when Interrupt Feeding function (IFEED instruction) is executing.	ON: Feeding OFF: Not feeding	Cleared	Cleared	• Refreshed when feed-ing is com-pleted. • Refreshed when inter-rupt feeding is executing.	
	09	Pulse Output 0 Interrupt Feeding Error Flag	ON if an overflow or underflow error occurs during input interrupt or specified number of pulse output, when Interrupt Feeding function (IFEED instruction) is executing.	ON: Overflow or underflow OFF: Normal	Cleared	Cleared	Refreshed when interrupt feeding is executing.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A281	00	Pulse Output 1 Accel/Decel Flag	This flag will be ON when pulses are being output from pulse output 1 according to an ORG, ACC, PLS2, ITPL or IFEED instruction and the output frequency is being changed in steps (accelerating or decelerating).	ON: Accelerating or decelerating OFF: Constant speed	---	Cleared	Refreshed each cycle during the overseeing processes.	
	01	Pulse Output 1 Overflow/Underflow Flag	This flag indicates when an overflow or underflow has occurred in the pulse output 1 PV.	ON: Overflow or underflow OFF: Normal	---	Cleared	• Refreshed when the PV is changed by the INI instruction. • Refreshed when an overflow or underflow occurs.	
	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 instruction.	ON: Setting made OFF: No setting	---	Cleared	• Refreshed when the PULS instruction is executed. • Refreshed when pulse output stops.	
	03	Pulse Output 1 Output Completed Flag	ON when the number of output pulses set with the PULS, PLS2, ITPL or IFEED instruction has been output through pulse output 1.	ON: Output completed OFF: Output not completed	---	Cleared	Refreshed at the start or completion of pulse output.	
	04	Pulse Output 1 Output In-progress Flag	ON when pulses are being output from pulse output 1.	ON: Outputting pulses OFF: Stopped	---	Cleared	Refreshed when pulse output starts or stops.	
	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.	ON: Origin not established OFF: Origin established	ON	ON	Refreshed each cycle during the overseeing processes.	
	06	Pulse Output 1 At-origin Flag	ON when the pulse output 1 PV matches the origin (0).	ON: Stopped at origin OFF: Not stopped at origin	---	Cleared	Refreshed each cycle during the overseeing processes.	
	07	Pulse Output 1 Output Stopped Error Flag	ON when an error occurred while outputting pulses in the pulse output 1 origin search function. The Pulse Output 1 Output Stop Error code will be written to A445.	ON: Stop error occurred OFF: No error	---	Cleared	• Refreshed when origin search starts. • Refreshed when a pulse output stop error occurs.	A445
	08	Pulse Output 1 Interrupt Feeding Flag	ON between Input Interrupt - decelerating to stop when Interrupt Feeding function (IFEED instruction) is executing.	ON: Feeding OFF: Not feeding	Cleared	Cleared	• Refreshed when feed-ing is com-pleted. • Refreshed when inter-rupt feeding is executing.	
	09	Pulse Output 1 Interrupt Feeding Error Flag	ON if an overflow or underflow error occurs during input interrupt or specified number of pulse output, when Interrupt Feeding function (IFEED instruction) is executing.	ON: Overflow or underflow OFF: Normal	Cleared	Cleared	Refreshed when interrupt feeding is executing.	
A283	00	PWM Output 0 Output In-progress Flag	ON when pulses are being output from PWM output 0.	ON: Outputting pulses OFF: Stopped	Cleared	Cleared	Refreshed when pulse output starts or stops.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A294		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. **Note** A298 and A299 contain the program address where program execution was stopped.	Cyclic tasks: 0000 Interrupt tasks: 8000 to 800F (task 0 to 15)	Cleared	Cleared	When program error occurs.	A298/ A299
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. **Note** 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, A298/ A299 PLC Setup (Operation when instruction error has occurred)
	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. **Note** 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, A298/ A299 PLC Setup (Operation when instruction error has occurred)
	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. The following operations are considered illegal access: • Reading/writing the system area • Indirect DM BCD error (in BCD mode) **Note** 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, A298/ A299 PLC Setup (Operation when instruction error has occurred)
	11	No END Error Flag	ON when there isn't an END instruction in each program within a task. CPU Unit operation will stop and the ERR/ALM indicator will light when this flag goes ON. **Note** 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	When program error occurs.	A294, A298/ A299

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A295	12	Task Error Flag	ON when a task error has occurred. A task error will occur when there is no program allocated to the task. **Note** 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	When program error occurs.	A294, A298/ A299
	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. **Note** 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	When program error occurs.	A294, A298/ A299
	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	When program error occurs.	A294, A298/ A299
	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	When program error occurs.	A294, A298/ A299
A297	---	Program Stop Error Identification information	This word contains FFFF hex when the stop position is in the program except the function block. This word contains the data other than FFFF hex when the stop position is in the function block.	FFFF hex: Stop in the program Except FFFF hex: Stop in the function block	Retained	Cleared	When program error occurs.	A295, A297/ A298
A298	---	Program Address Where Program Stopped (Lower digits)	These words contain the program address of the instruction where program execution was stopped due to a program error. **Note** A294 contains the task number of the task where program execution was stopped.	Lower digits of the program address	Cleared	Cleared	When program error occurs.	A294
A299	---	Program Address Where Program Stopped (Upper digits)		Upper digits of the program address				
A300	---	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). **Note 1** The Error Log Pointer can be cleared to 00 by turning A500.14 (the Error Log Reset Bit) ON. **2** 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 hex	Retained	Retained	Refreshed when error occurs.	A500.14
A310	---	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.	Examples: Lot number 01805 A310 = 0801, A311 = 0005 Lot number 30Y05 A310 =1130, A311 = 0005	Retained	Retained	Fixed value	
A311	---	Manufacturing Lot Number, Upper Digits						

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A313	---	Error Contents for Ethernet Errors	An error content is registered when a built-in Ethernet error or Ethernet setting table error occurs.	For details of the error contents, refer to *6-1-10 Non-fatal Errors* in the *CP2E CPU Unit Hardware User's Manual* (Cat. No. W613).	Cleared	Cleared	Refreshed when a non-fatal error occurs.	A315.10
A315	10	Built-in Etherent Contents/Setup Error Flag	ON when an error occurs in the connection/setting of FINS/TCP, SNTP and DNS server.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when a non-fatal error occurs.	A313 A402.00
	11	Logic Error in Setting Tables	ON when an error occurs in the routing table, IP address table and IP router table. The error code is stored in A313.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when a non-fatal error occurs.	A402.00
	13	Option Board Error Flag	ON when the Option Board is removed while the power is being supplied. ON when two Analog Option Boards are mounted. CPU Unit operation will continue and the ERR/ALM indicator will flash. **Note** OFF when the error has been cleared.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when a non-fatal error occurs.	A402.00, A424
	15	Backup Memory Error Flag	ON when writing to the built-in Flash Memory fails. CPU Unit operation will continue and the ERR/ALM indicator will flash. **Note** OFF when the error has been cleared.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when a non-fatal error occurs.	A402.00
A316 to A317	---	High-speed Counter 2 PV	Contains the PV of high-speed counter 2. A317 contains the upper 4 digits and A316 contains the lower 4 digits.		Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read PV. • Refreshed when INI instruction is executed.	
A318 to A319	---	High-speed Counter 3 PV	Contains the PV of high-speed counter 3. A319 contains the upper 4 digits and A318 contains the lower 4 digits.		Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read PV. • Refreshed when INI instruction is executed.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A320	00	High-speed Counter 2 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 for upper and lower limits.	ON: PV in range OFF: PV not in range	Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read the results of range comparison. • Refreshed when range comparison table is registered.	
	01	High-speed Counter 2 Range 2 Comparison Condition Met Flag						
	02	High-speed Counter 2 Range 3 Comparison Condition Met Flag						
	03	High-speed Counter 2 Range 4 Comparison Condition Met Flag						
	04	High-speed Counter 2 Range 5 Comparison Condition Met Flag						
	05	High-speed Counter 2 Range 6 Comparison Condition Met Flag						
	08	High-speed Counter 2 Comparison In-progress Flag	This flag indicates whether a comparison operation is being executed for high-speed counter 2.	ON: Being executed OFF: Stopped	---	Cleared	Refreshed when comparison operation starts or stops.	
	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.)	ON: Overflow or underflow OFF: Normal	Cleared	Cleared	• Refreshed when an overflow or underflow occurs. • Refreshed when PV is changed.	
	10	High-speed Counter 2 Count Direction	This flag indicates whether the high-speed counter 2 is currently being incremented or decremented. The counter PV for the current cycle is compared with the PV in last cycle to determine the direction.	ON: Incrementing OFF: Decrementing	---	Cleared	Setting used for high-speed counter, valid during counter operation.	
A321	00	High-speed Counter 3 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 for upper and lower limits.	ON: PV in range OFF: PV not in range	Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read the results of range comparison. • Refreshed when range comparison table is registered.	
	01	High-speed Counter 3 Range 2 Comparison Condition Met Flag						
	02	High-speed Counter 3 Range 3 Comparison Condition Met Flag						
	03	High-speed Counter 3 Range 4 Comparison Condition Met Flag						
	04	High-speed Counter 3 Range 5 Comparison Condition Met Flag						
	05	High-speed Counter 3 Range 6 Comparison Condition Met Flag						
	08	High-speed Counter 3 Comparison In-progress Flag	This flag indicates whether a comparison operation is being executed for high-speed counter 3.	ON: Being executed OFF: Stopped	---	Cleared	Refreshed when comparison operation starts or stops.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
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.)	ON: Overflow or underflow OFF: Normal	Cleared	Cleared	• Refreshed when an overflow or underflow occurs. • Refreshed when PV is changed.	
	10	High-speed Counter 3 Count Direction	This flag indicates whether the high-speed counter is currently being incremented or decremented. The counter PV for the current cycle is compared with the PV in last cycle to determine the direction.	ON: Incrementing OFF: Decrementing	---	Cleared	Setting used for high-speed counter, valid during counter operation.	
A322 to A323		High-speed Counter 4 PV	Contains the PV of high-speed counter 4. A323 contains the upper four digits and A322 contains the lower four digits.		Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read PV. • Refreshed when INI instruction is executed.	
A324 to A325		High-speed Counter 5 PV	Contains the PV of high-speed counter 5. A325 contains the upper 4 digits and A324 contains the lower 4 digits.		Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read PV. • Refreshed when INI instruction is executed.	
A326	00	High-speed Counter 4 Range 1 Comparison Condition Met Flag	These flags indicate whether the PV is within the specified ranges when high-speed counter 4 is being operated in range-comparison mode for upper and lower limits.	ON: PV in range OFF: PV not in range	Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read the results of range com-parison. • Refreshed when range comparison table is reg-istered.	
	01	High-speed Counter 4 Range 2 Comparison Condition Met Flag						
	02	High-speed Counter 4 Range 3 Comparison Condition Met Flag						
	03	High-speed Counter 4 Range 4 Comparison Condition Met Flag						
	04	High-speed Counter 4 Range 5 Comparison Condition Met Flag						
	05	High-speed Counter 4 Range 6 Comparison Condition Met Flag						
	08	High-speed Counter 4 Comparison In-progress Flag	This flag indicates whether a comparison operation is being executed for high-speed counter 4.	ON: Being executed OFF: Stopped	---	Cleared	Refreshed when comparison operation starts or stops.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A326	09	High-speed Counter 4 Overflow/Underflow Flag	This flag indicates when an overflow or underflow has occurred in the high-speed counter 4 PV. (Used with the linear mode counting range only.)	ON: Overflow or underflow OFF: Normal	Cleared	Cleared	• Refreshed when an overflow or underflow occurs. • Refreshed when PV is changed.	
	10	High-speed Counter 4 Count Direction	This flag indicates whether the high-speed counter is currently being incremented or decremented. The counter PV for the current cycle is compared with the PV in last cycle to determine the direction.	ON: Incrementing OFF: Decrementing	---	Cleared	Setting used for high-speed counter, valid during counter operation.	
A327	00	High-speed Counter 5 Range 1 Comparison Condition Met Flag	These flags indicate whether the PV is within the specified ranges when high-speed counter 5 is being operated in range-comparison mode.	ON: PV in range OFF: PV not in range	Cleared	Cleared	• Refreshed each cycle during the overseeing processes. • Refreshed when PRV instruction is executed to read the results of range comparison. • Refreshed when range comparison table is registered.	
	01	High-speed Counter 5 Range 2 Comparison Condition Met Flag						
	02	High-speed Counter 5 Range 3 Comparison Condition Met Flag						
	03	High-speed Counter 5 Range 4 Comparison Condition Met Flag						
	04	High-speed Counter 5 Range 5 Comparison Condition Met Flag						
	05	High-speed Counter 5 Range 6 Comparison Condition Met Flag						
	08	High-speed Counter 5 Comparison In-progress Flag	This flag indicates whether a comparison operation is being executed for high-speed counter 5.	ON: Being executed OFF: Stopped	---	Cleared	Refreshed when comparison operation starts or stops.	
	09	High-speed Counter 5 Overflow/Underflow Flag	This flag indicates when an overflow or underflow has occurred in the high-speed counter 5 PV. (Used with the linear mode counting range only.)	ON: Overflow or underflow OFF: Normal	Cleared	Cleared	• Refreshed when an overflow or underflow occurs. • Refreshed when PV is changed.	
	10	High-speed Counter 5 Count Direction	This flag indicates whether the high-speed counter is currently being incremented or decremented. The counter PV for the current cycle is compared with the PV in last cycle to determine the direction.	ON: Incrementing OFF: Decrementing	---	Cleared	Setting used for high-speed counter, valid during counter operation.	
A339 to A340	---	Maximum Differentiation Flag Number	These words contain the maximum value of the differentiation flag numbers being used by differentiation instructions.		---	Cleared	• Written at the start of operation. • Written at the time of online editing.	A295.13

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A351 to A354		Calendar/Clock Area	These words contain the CPU Unit's internal clock data in BCD. The clock can be set from the CX-Programmer, with the DATE instruction, or with a FINS command (CLOCK WRITE, 0702). 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: Hours (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 to 06)(BCD) 00: Sunday 01: Monday 02: Tuesday 03: Wednesday 04: Thursday 05: Friday 06: Saturday **Note** In an E☐☐-type CPU Unit or when time alignment cannot be executed with an N/S☐☐-type CPU Unit, the data will be for 1:01.01 on Sunday January 1, 2001.		Retained	Retained	Written every cycle	
A360 to A391	01 to 15	Executed FAL Number Flags	The flag corresponding to the specified FAL number will be turned ON when FAL is executed. Bits A360.01 to A391.15 correspond to FAL numbers 001 to 511. **Note** The flag will be turned OFF when the error is cleared.	ON: That FAL executed OFF: That FAL not 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	04	Serial Port 1/ Built-in RS-232C Port Error Flag	ON when an error has occurred at the serial port 1 or built-in RS-232C port. (Not valid in NT Link mode.)	ON: Error OFF: Normal	Retained	Cleared	Refreshed when error occurs.	
	05	Serial Port 1/ Built-in RS-232C Port Send Ready Flag (No-protocol mode)	ON when the serial port 1 or built-in RS-232C port is able to send data in no-protocol mode.	ON: Able-to-send OFF: Unable-to-send	Retained	Cleared	Written after transmission	
	06	Serial Port 1/ Built-in RS-232C Port Reception Completed Flag (No-protocol mode)	ON when the serial port 1 or built-in RS-232C port has completed the reception in no-protocol mode. • When the number of bytes was specified: ON when the specified number of bytes is received. • 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	
	07	Serial Port 1/ Built-in RS-232C Port Reception Overflow Flag (No-protocol mode)	ON when a data overflow occurred during reception through the serial port 1 or built-in RS-232C port in no-protocol mode. • When the number of bytes was specified: ON when more data is received after the reception was completed but before RXD was executed. • When the end code was specified: ON when more data is received after the end code was received but before RXD was executed. ON when 257 bytes are received before the end code.	ON: Overflow OFF: No overflow	Retained	Cleared	Refreshed when error occurs.	
	12	Serial Port 2/ Built-in RS-485 Port Communications Error Flag (CP2E N30/40/60 or S□□-type CPU Unit only)	ON when a communications error has occurred at the serial port 2 or built-in RS-485 port. (Not valid in NT Link mode.)	ON: Error OFF: No error	Retained	Cleared	Refreshed when error occurs.	
	13	Serial Port 2/ Built-in RS-485 Port Send Ready Flag (No-protocol Mode) (CP2E N30/40/60 or S□□-type CPU Unit only)	ON when the serial port 2 or built-in RS-485 port is able to send data in no-protocol mode.	ON: Able-to-send OFF: Unable-to-send	Retained	Cleared	Written after transmission	
	14	Serial Port 2/ Built-in RS-485 Port Reception Completed Flag (No-protocol mode) (CP2E N30/40/60 or S□□-type CPU Unit only)	ON when the serial port 2 or built-in RS-485 port has completed the reception in no-protocol mode. • When the number of bytes was specified: ON when the specified number of bytes is received. • 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	
	15	Serial Port 2/ Built-in RS-485 Port Reception Overflow Flag (No-protocol mode) (CP2E N30/40/60 or S□□-type CPU Unit only)	ON when a data overflow occurred during reception through serial port 2 or built-in RS-485 port in no-protocol mode. • When the number of bytes was specified: ON when more data is received after the reception was completed but before RXD was executed. • When the end code was specified: ON when more data is received after the end code was received but before RXD was executed. ON when 257 bytes are received before the end code.	ON: Reception completed OFF: Reception not completed	Retained	Cleared	Refreshed when error occurs.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A393	00 to 07	Serial Port 1/ Built-in RS-232C Port Polled Unit Communications Flags	The corresponding bit will be ON when the serial port 1 or built-in RS-232C port is communicating with NT Link mode or Serial PLC Link mode. 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.	
	00 to 15	Serial Port 1/ Built-in RS-232C Port Reception Counter (No-protocol Mode)	Indicates (in binary) the number of bytes of data received when the serial port 1 or built-in RS-232C port is in no-protocol mode.		Retained	Cleared	Refreshed when data is received.	
A394	00 to 07	Serial Port 2/ Built-in RS-485 Port Polled Unit Communications Flags (CP2E N30/40/60 or S☐☐-type CPU Unit only)	The corresponding bit will be ON when the serial port 2 or built-in RS-485 port is communicating with NT link mode or Serial PLC Link made. 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.	
	00 to 15	Serial Port 2/ Built-in RS-485 Port Reception Counter (No-protocol Mode) (CP2E N30/40/60 or S☐☐-type CPU Unit only)	Indicates (in binary) the number of bytes of data received when the serial port 2 or built-in RS-485 port is in no-protocol mode.		Retained	Cleared	Refreshed when data is received.	
A395	08	Clock Stop Flag	• ON if Clock stop working. • Cleared when Clock set.	ON: Clock stops OFF: Clock is running	Retained	Cleared	Refreshed when power is ON.	
A400	---	Error code	When a non-fatal error (user-defined FALS or system error) or a fatal error (user-defined FALS or system error) occurs, the 4-digit hexadecimal error code is written to this word. **Note** When two or more errors occur simultaneously, the highest error code will be recorded.		Cleared	Cleared	Refreshed when error occurs.	
A401	04	Built-in Ethernet Stop Error Flag	ON when the built-in Ethernet stops. The ERR/ALM indicator on front of the CPU unit will light.	ON: Error OFF: Normal	Retained	Cleared	Refreshed when error occurs.	
	06	FALS Error Flag (fatal error)	ON when a fatal error is generated by the FALS instruction. The CPU Unit will stop operating and the ERR/ALM indicator will light. The corresponding error code will be written to A400. Error codes C101 to C2FF correspond to FALS numbers 001 to 511. **Note** This flag will be turned OFF when the FALS errors are cleared.	ON: FALS executed OFF: FALS not executed	Cleared	Cleared	Refreshed when error occurs.	A400
	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. **Note** This flag will be turned OFF when the error is cleared.	ON: Cycle time over max. OFF: Cycle time under max.	Cleared	Cleared	Refreshed when the cycle time exceeds maximum.	PLC Setup (Cycle time monitorin g time)

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A401	09	Program Error Flag (fatal error)	ON when program contents are incorrect. 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. 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.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when error occurs.	A294, A295, A298 and A299
	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. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when error occurs.	A407
	14	I/O Bus Error Flag (fatal error)	ON in the following cases: • When an error occurs in a data transfer between the CPU Unit and an Expansion Unit or Expansion I/O Unit. If this happens, 0A0A hex will be output to A404. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when error occurs.	A404
	15	Memory Error Flag (fatal error)	ON when an error occurred in memory. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. The location where the error occurred is indicated in A403.00 to A403.14. This flag will be turned OFF when the error is cleared.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when error occurs.	A403.00 to A403.08, A403.09

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A402	00	Other Non-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 A315.	ON: Other non-fatal error OFF: No other non-fatal error	Cleared	Cleared	Refreshed when error occurs.	A315
	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. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. • This flag can be used to control an external warning light or other indicator to indicate that the battery needs to be replaced. • This flag will be turned OFF when the error is cleared.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when error occurs.	PLC Setup (Detect Battery Error)
	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. **Note** This flag will be turned OFF when the error is cleared.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when error occurs.	
	15	FAL Error Flag (non-fatal error)	ON when a non-fatal error is generated by executing FAL. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. The bit in A360 to A391 that corresponds to the FAL number specified in FALS 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). **Note** This flag will be turned OFF when the error is cleared.	ON: FAL error occurred OFF: FAL not executed	Cleared	Cleared	Refreshed when error occurs.	A360 to A391, A400
A403	00 to 14	Memory Error Location	When a memory error occurs, the Memory Error Flag (A401.15) is turned ON and one of the following flags is turned ON to indicate the memory area where the error occurred A403.00: Ladder program A403.04: PLC Setup A403.07: Routing table A403.11: IP address table A403.12: IP router table A403.14: I/O memory 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. **Note** The corresponding flag will be turned OFF when the error is cleared.	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when error occurs.	A401.15
A404		I/O Bus Error Details	Contains information on I/O bus errors. The CPU Unit will stop operating and the ERR/ALM indicator on the front of the CPU Unit will light. **Note** A401.14 (I/O Bus Error Flag) will turn ON.	0A0A hex: Expansion Unit error	Cleared	Cleared	Refreshed when error is detected.	A401.14
A407	13 to 15	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.	010: Too many Expansion Unit and Expansion I/O Unit words	Cleared	Cleared	Refreshed when error occurs.	A401.11

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A424	00 to 15	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). Bit 00: Option slot 1 (Left) Bit 01: Option slot 2 (Right)	ON: Error OFF: Normal	Cleared	Cleared	Refreshed when error occurs.	A353.13
A435	14	I/O Option Board Run State Flag (Option 1/Left)	Turn ON when the I/O Option Board works normally. Turn OFF when the I/O Option Board is in initial processing or an error occurs.	ON: In initial processing or error occurred. OFF: Normal	Cleared	Cleared	When I/O option board state changes.	
	15	I/O Option Board Run State Flag (Option 2/Right)						
A436	00 to 02	Expansion Unit and Expansion I/O Unit Error Flags	ON when an error occurs in a CP-series Expansion Unit or Expansion I/O Unit. A436.00: 1st Unit A436.01: 2nd Unit A436.02: 3rd Unit A436.03: 4th Unit A436.04: 5th Unit A436.05: 6th Unit **Note** CP1W-TS002/TS003/TS102/ AD041/AD042/DA041/DA042/ MAD42/MAD44/32ER/32ET/32E T1 are each counted as two Units.	ON: Error OFF: Normal	Retained	Cleared		
A437	---	Number of Connected Units	Stores the number of Expansion Units and Expansion I/O Units connected as a hexadecimal number. **Note** This information is invalid only when a Too Many I/O Points error has occurred. CP1W-TS002/TS003/TS102/ AD041/AD042/DA041/DA042/ MAD42/MAD44/32ER/32ET/32ET 1 are each counted as two Units.	0000 to 0006 hex	Retained	Cleared	Retained when error occurs.	
A438	---	Pulse Output 2 Stop Error Code	If a Pulse Output Stop Error occurs for pulse output 2, the error code is written to this word.		Retained	Cleared	Retained when error occurs.	
A439	---	Pulse Output 3 Stop Error Code	If a Pulse Output Stop Error occurs for pulse output 3, the error code is written to this word.		Retained	Cleared	Retained when error occurs.	
A440	---	Max. Interrupt Task Processing Time	Contains the Maximum Interrupt Task Processing Time in units of 0.1 ms. **Note** This value is cleared when PLC operation begins.	0000 to FFFF hex	Cleared	Cleared	Written after the interrupt task with the max. processing time is executed.	
A441	---	Interrupt Task with Max. Processing Time	Contains the task number of the interrupt task with the maximum processing time. Hexadecimal values 8000 to 800F correspond to task numbers 00 to 0F. Bit 15 is turned ON when an interrupt has occurred. **Note** This value is cleared when PLC operation begins.	8000 to 800F hex	Cleared	Cleared	Written after the interrupt task with the max. processing time is executed.	
A442	---	Total Interrupt Task Processing Time One Cycle	Contains the Total Interrupt Task Processing Time in one cycle in units of 0.1ms.Sets when the value is bigger than the last one once a cycle by common processing. **Note** This value is cleared when PLC operation begins. The value is unstable for CPU Unit version 1.0 or earlier.	0000 to FFFF hex	Cleared	Cleared	Each cycle	A440

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A444	---	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.		Retained	Cleared	• Refreshed when origin search starts. • Refreshed when a pulse output stop error occurs.	
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.		Retained	Cleared	• Refreshed when origin search starts. • Refreshed when a pulse output stop error occurs.	

A-2-2 Read/Write Words

Address: Words	Address: Bits	Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
A500	11	Ethernet Communication Error Clear Flag	Turn this bit ON to clear Ethernet communication error. After clearing, the system automatically returns to OFF.		Retained	Cleared	---	
	12	IOM Hold Bit	Turn ON this bit to preserve the status of the I/O Memory when shifting from PROGRAM to RUN or MONITOR mode or vice versa.	ON: Retained OFF: Not retained	Retained	Cleared	---	
	13	Forced Status Hold Bit	Turn ON this bit to preserve the status of bits that have been force-set or force-reset when shifting from PROGRAM to MONITOR mode or vice versa. Always use this bit together with the IOM Hold Bit (A500.12), i.e., turn them ON at the same time.	ON: Retained OFF: Not retained	Retained	Cleared	---	
	14	Error Log Reset Bit	Turn this bit ON to reset the Error Log Pointer (A300) to 00. **Note 1** The contents of the Error Log Area itself (A100 to A199) are not cleared. **2** This bit is automatically reset to 0 after the Error Log Pointer is reset.	OFF to ON: Clear	Retained	Cleared	---	A100 to A199, A300
	15	Output OFF Bit	Turn this bit ON to turn OFF all outputs from the CPU Unit, CP-series Expansion Units, and CP-series Expansion I/O Units. The INH indicator on the front of the CPU Unit will light while this bit is ON.	ON: All output OFF OFF: Normal operation	Retained	Retained	---	
A508	09	Differentiate Monitor Completed Flag	ON when the differentiate monitor condition has been established during execution of differentiation monitoring. **Note** This flag will be cleared to 0 when differentiation monitoring starts.	ON: Monitor condition established OFF: Not yet established	Retained	Cleared	Refreshed when differentiate monitor condition is executed.	
A510 to A511	---	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. A510.00 to A510.07: Second (00 to 59) A510.08 to A510.15: Minute (00 to 59) A511.00 to A511.07: Hour (00 to 23) A511.08 to A511.15: Day of month (01 to 31) **Note** In an E☐☐-type CPU Unit or when time alignment cannot be executed with an N/S☐☐-type CPU Unit, the data will be for 1:01.01 on Sunday January 1, 2001.	See Function column.	Retained	See Function column.	Refreshed when power is turned ON.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A512 to A513	---	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. A512.00 to A512.07: Second (00 to 59) A512.08 to A512.15: Minute (00 to 59) A513.00 to A513.07: Hour (00 to 23) A513.08 to A513.15: Day of month (01 to 31) **Note 1** These words are not cleared at startup. **2** In an E☐☐-type CPU Unit or when time alignment cannot be executed with an N/S☐☐-type CPU Unit, the data will be for 1:01.01 on Sunday January 1, 2001.	See Function column.	Retained	Retained	Written at power interruption.	
A514	---	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.	0000 to FFFF hex	Retained	Retained	Refreshed when power is turned ON.	
A515 to A517	---	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. A515.00 to A515.07: Seconds (00 to 59) A515.08 to A515.15: Minutes (00 to 59) A516.00 to A516.07: Hour (00 to 23) A516.08 to A516.15: Day of month (01 to 31) A517.00 to A517.07: Month (01 to 12) A517.08 to A517.15: Year (00 to 99) **Note 1** The previous start time is stored after turning ON the power supply until operation is started. **2** In an E☐☐-type CPU Unit or when time alignment cannot be executed with an N/S☐☐-type CPU Unit, the data will be for 1:01.01 on Sunday January 1, 2001.	See Function column.	Retained	Retained	See Function column.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A518 to A520	---	Operation End Time	The time that operation stopped as a result of changing the operating mode to PROGRAM mode is stored here in BCD. A518.00 to A518.07: Seconds (00 to 59) A518.08 to A518.15: Minutes (00 to 59) A519.00 to A519.07: Hour (00 to 23) A519.08 to A519.15: Day of month (01 to 31) A520.00 to A520.07: Month (01 to 12) A520.08 to A520.15: Year (00 to 99) **Note 1** 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. **2** In an E□□-type CPU Unit or when time alignment cannot be executed with an N/S□□-type CPU Unit, the data will be for 1:01.01 on Sunday January 1, 2001.	See Function column.	Retained	Retained	See Function column.	
A521	00 to 07	Serial Port 1 (EX) Error Flags (CP2E N□□-type CPU Unit only)	These flags indicate what kind of error has occurred at the serial port 1 (EX). • They are automatically turned OFF when the serial port 1 (EX) is restarted. • Only bit 5 (timeout error) is valid in NT Link mode. • Serial PLC Link Polling Unit: Bit 05: ON for timeout error. Serial PLC Link Polled Unit: Bit 02: ON for parity error. Bit 03: ON for framing error. Bit 04: ON for overrun error. These bits can be cleared by the CX-Programmer.	Bits 00 and 01: Not used. Bit 02: ON for parity error. Bit 03: ON for framing error. Bit 04: ON for overrun error. Bit 05: ON for timeout error. Bits 06 and 07: Not used.	Retained	Cleared	Refreshed when communication error occurs.	
A526	00	Serial Port 1/ Built-in RS-232C Port Restart Bit	Turn ON this bit to restart the serial port 1 or built-in RS-232C port. **Note** This bit is turned OFF automatically when the restart processing is completed.	OFF to ON: Restart	Retained	Cleared	---	
	01	Serial Port 2/ Bult-in RS-485 Port Restart Bit (CP2E N30/40/60 or S□□-type CPU Unit only)	Turn ON this bit to restart the serial port 2 or built-in RS-485 port. **Note** This bit is turned OFF automatically when the restart processing is completed.	OFF to ON: Restart	Retained	Cleared	---	
	02	Serial Port 1 (EX) Restart Bit (CP2E N□□-type CPU Unit only)	Turn ON this bit to restart the serial port 1 (EX). **Note** This bit is turned OFF automatically when the restart processing is completed.	OFF to ON: Restart	Retained	Cleared	---	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A528	00 to 07	Serial Port 1/ Built-in RS-232C Port Error Flags	These flags indicate what kind of error has occurred at the serial port 1 or built-in RS-232C port. • They are automatically turned OFF when the serial port 1 or built-in RS-232C port is restarted. • Only bit 5 (timeout error) is valid in NT Link mode. • Serial PLC Link Polling Unit: Bit 05: ON for timeout error. Serial PLC Link Polled Unit: Bit 02: ON for parity error. Bit 03: ON for framing error. Bit 04: ON for overrun error. These bits can be cleared by the CX-Programmer.	Bits 00 and 01: Not used. Bit 02: ON for parity error. Bit 03: ON for framing error. Bit 04: ON for overrun error. Bit 05: ON for timeout error. Bits 06 and 07: Not used.	Retained	Cleared	Refreshed when communication error occurs.	
	08 to 15	Serial Port 2/ Built-in RS-485 Port Error Flags (CP2E N30/40/60 or S□□-type CPU Unit only)	These flags indicate what kind of error has occurred at the serial port 2 or built-in RS-485 port. • They are automatically turned OFF when the serial port 2 or built-in RS-485 port is restarted. • Only bit 5 (timeout error) is valid in NT Link mode. • Serial PLC Link Polling Unit: Bit 13: ON for timeout error. Serial PLC Link Polled Unit: Bit 10: ON for parity error. Bit 11: ON for framing error. Bit 12: ON for overrun error. These bits can be cleared by the CX-Programmer.	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	Refreshed when communication error occurs.	
A529	---	FAL/FALS Number for System Error Simulation	Set a dummy FAL/FALS number to use to simulate the system error using FAL or FALS. **Note** When FAL or FALS 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.	0001 to 01FF hex: FAL/FALS numbers 1 to 511 0000 or 0200 to FFFF hex: No FAL/FALS number for system error simulation. (No error will be generated.)	Retained	Cleared	---	
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 PV will be reset if the phase-Z signal is received while this bit is ON. When the reset method is set to Software reset, the corresponding high-speed counter's PV will be reset in the cycle when this bit turns ON.	OFF to ON: Reset	Retained	Cleared	---	
	01	High-speed Counter 1 Reset Bit						
	02	High-speed Counter 2 Reset Bit						
	03	High-speed Counter 3 Reset Bit						
	04	High-speed Counter 4 Reset Bit						
	05	High-speed Counter 5 Reset Bit						

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
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.	OFF to ON: Cleared	Retained	Cleared	---	A276 and A277
	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	---	
	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	---	
	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	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.	OFF to ON: Cleared	Retained	Cleared	---	A278 and A279
	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	---	
	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	---	
	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	---	
A542	00	Pulse Output 2 Reset Bit	The pulse output 2 PV (contained in A52 and A53) will be cleared when this bit is turned ON.	OFF to ON: Cleared	Retained	Cleared	---	A52 and A53
	08	Pulse Output 2 CW Limit Input Signal Flag	This is the CW limit input signal for pulse output 2, 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	---	
	09	Pulse Output 2 CCW Limit Input Signal Flag	This is the CCW limit input signal for pulse output 2, 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	---	
	10	Pulse Output 2 Positioning Completed Signal	This is the positioning completed input signal used in the origin search for pulse output 2. The input signal from the servo driver is output to this bit from the ladder program to enable using the signal.	---	Retained	Cleared	---	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A543	00	Pulse Output 3 Reset Bit	The pulse output 3 PV (contained in A54 and A55) will be cleared when this bit is turned ON.	OFF to ON: Cleared	Retained	Cleared	---	A54 and A55
	08	Pulse Output 3 CW Limit Input Signal Flag	This is the CW limit input signal for pulse output 3, 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 programand output the result to this flag.	---	Retained	Cleared	---	
	09	Pulse Output 3 CCW Limit Input Signal Flag	This is the CCW limit input signal for pulse output 3, 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 programand output the result to this flag.	---	Retained	Cleared	---	
	10	Pulse Output 3 Positioning Completed Signal	This is the positioning completed input signal used in the origin search for pulse output 3. The input signal from the servo driver is output to this bit from the ladder program to enable using the signal.	---	Retained	Cleared	---	
A566	02	Socket Force-close Switch	OFF → ON: Forced close of all socket is executed. Automatically turns OFF when closing is completed.	OFF to ON: Forced close is executed	---	Cleared	---	
	04	Automatic Clock Information Adjustment Switch	At the rise of OFF to ON, clock information is obtained from SNTP server and the clock information is reflected. Automatically turns OFF after reflection.	OFF to ON: Clock information is obtained	---	Cleared	---	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A567	00	Open Processing Flag for Socket No.1	Flag keeps ON during open processing of socket No.1. Finish open process, and then flag turns OFF.	ON: During open processing OFF: Open completed	---	Cleared	Refreshed at the start or completion of open process	
	01	Reception Progressing Flag for Socker No.1	Flag keeps ON during reception processing of socket No.1. Finish reception process, and then flag turns OFF.	ON: Receiving OFF: Reception completed	---	Cleared	Refreshed at the start or completion of reception process	
	02	Transmission Progressing Flag for Socket No.1	Flag keeps ON during transimission processing of socket No.1. Finish transmission process, and then flag turns OFF.	ON: Sending OFF: Transimission completed	---	Cleared	Refreshed at the start or completion of transmission process	
	03	Close Processing Flag for Socker No.1	Flag keeps ON during close processing of socket No.1. Finish clsoe process, and then flag turns OFF.	ON: During close processing OFF: Close completed	---	Cleared	Refreshed at the start or completion of close process	
	13	Data Received Present/Not Present Flag for Socket No.1	Flag keeps ON when data is received from the other node on an open TCP socket. Flag turns OFF when the reception request is executed.	ON: Data reception completed OFF: Waiting for reception	---	Cleared	• Refreshed after reception • Refreshed when reception process is requested	
	14	Result Storage Error Flag for Socket No.1	Flag keeps ON when any value other than normal end (0000 hex) is stored in the end code of the parameter area of socket No.1. Flag turns OFF when the next request is received.	ON: Code other than normal end is stored OFF: Normal end	---	Cleared	Refreshed when socket request process is completed	
	15	TCP/UDP Open Flag for Socket No.1	When open processing of socket No.1 is completed, the flag turns ON. When closing process is completed or while closing, the flag turns OFF. When open processing ends abnormally, the flag remains OFF.	ON: Close completed OFF: Open completed	---	Cleared	Refreshed when open/close process is completed	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A568	00	Open Processing Flag for Socket No.2	Flag keeps ON during open processing of socket No.2. Finish open process, and then flag turns OFF.	ON: During open processing OFF: Open completed	---	Cleared	Refreshed at the start or completion of open process	
	01	Reception Progressing Flag for Socker No.2	Flag keeps ON during reception processing of socket No.2. Finish reception process, and then flag turns OFF.	ON: Receiving OFF: Reception completed	---	Cleared	Refreshed at the start or completion of reception process	
	02	Transmission Progressing Flag for Socket No.2	Flag keeps ON during transimission processing of socket No.2. Finish transmission process, and then flag turns OFF.	ON: Sending OFF: Transimission completed	---	Cleared	Refreshed at the start or completion of transmission process	
	03	Close Processing Flag for Socker No.2	Flag keeps ON during close processing of socket No.2. Finish clsoe process, and then flag turns OFF.	ON: During close processing OFF: Close completed	---	Cleared	Refreshed at the start or completion of close process	
	13	Data Received Present/Not Present Flag for Socket No.2	Flag keeps ON when data is received from the other node on an open TCP socket. Flag turns OFF when the reception request is executed.	ON: Data reception completed OFF: Waiting for reception	---	Cleared	• Refreshed after recep-tion • Refreshed when reception process is requested	
	14	Result Storage Error Flag for Socket No.2	Flag keeps ON when any value other than normal end (0000 hex) is stored in the end code of the parameter area of socket No.2. Flag turns OFF when the next request is received.	ON: Code other than normal end is stored OFF: Normal end	---	Cleared	Refreshed when socket request process is completed	
	15	TCP/UDP Open Flag for Socket No.2	When open processing of socket No.2 is completed, the flag turns ON. When closing process is completed or while closing, the flag turns OFF. When open processing ends abnormally, the flag remains OFF.	ON: Close com-pleted OFF: Open completed	---	Cleared	Refreshed when open/close process is completed	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A569	00	Open Processing Flag for Socket No.3	Flag keeps ON during open processing of socket No.3. Finish open process, and then flag turns OFF.	ON: During open processing OFF: Open completed	---	Cleared	Refreshed at the start or completion of open process	
	01	Reception Progressing Flag for Socker No.3	Flag keeps ON during reception processing of socket No.3. Finish reception process, and then flag turns OFF.	ON: Receiving OFF: Reception completed	---	Cleared	Refreshed at the start or completion of reception process	
	02	Transmission Progressing Flag for Socket No.3	Flag keeps ON during transimission processing of socket No.3. Finish transmission process, and then flag turns OFF.	ON: Sending OFF: Transimission completed	---	Cleared	Refreshed at the start or completion of transmission process	
	03	Close Processing Flag for Socker No.3	Flag keeps ON during close processing of socket No.3. Finish clsoe process, and then flag turns OFF.	ON: During close processing OFF: Close completed	---	Cleared	Refreshed at the start or completion of close process	
	13	Data Received Present/Not Present Flag for Socket No.3	Flag keeps ON when data is received from the other node on an open TCP socket. Flag turns OFF when the reception request is executed.	ON: Data reception completed OFF: Waiting for reception	---	Cleared	• Refreshed after reception • Refreshed when reception process is requested	
	14	Result Storage Error Flag for Socket No.3	Flag keeps ON when any value other than normal end (0000 hex) is stored in the end code of the parameter area of socket No.3. Flag turns OFF when the next request is received.	ON: Code other than normal end is stored OFF: Normal end	---	Cleared	Refreshed when socket request process is completed	
	15	TCP/UDP Open Flag for Socket No.3	When open processing of socket No.3 is completed, the flag turns ON. When closing process is completed or while closing, the flag turns OFF. When open processing ends abnormally, the flag remains OFF.	ON: Close completed OFF: Open completed	---	Cleared	Refreshed when open/close process is completed	
A571	00	UDP Open Request Switch for Socket No.1	When the flag changes from OFF to ON, UDP open processing is executed. When open process is completed, the flag automatically turns OFF.	OFF to ON: UDP open	---	Cleared	---	
	01	TCP Passive Open Request Switch for Socket No.1	When the flag changes from OFF to ON, TCP passive open processing is executed. When open process is completed, the flag automatically turns OFF.	OFF to ON: TCP passive open	---	Cleared	---	
	02	TCP Active Open Request Switch for Socket No.1	When the flag changes from OFF to ON, TCP active open processing is executed. When open process is completed, the flag automatically turns OFF.	OFF to ON: TCP active open	---	Cleared	---	
	03	Send Request Switch for Socket No.1	When the flag changes from OFF to ON, transmission processing is executed. When transmission process is completed, the flag automatically turns OFF.	OFF to ON: Send	---	Cleared	---	
	04	Receive Request Switch for Socket No.1	When the flag changes from OFF to ON, reception processing is executed. When reception process is completed, the flag automatically turns OFF.	OFF to ON: Receive	---	Cleared	---	
	05	Close Request Switch for Socket No.1	When the flag changes from OFF to ON, close processing is executed. When close process is completed, the flag automatically turns OFF.	OFF to ON: Close	---	Cleared	---	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A571	08	UDP Open Request Switch for Socket No.2	When the flag changes from OFF to ON, UDP open processing is executed. When open process is completed, the flag automatically turns OFF.	OFF to ON: UDP open	---	Cleared	---	
	09	TCP Passive Open Request Switch for Socket No.2	When the flag changes from OFF to ON, TCP passive open processing is executed. When open process is completed, the flag automatically turns OFF.	OFF to ON: TCP passive open	---	Cleared	---	
	10	TCP Active Open Request Switch for Socket No.2	When the flag changes from OFF to ON, TCP active open processing is executed. When open process is completed, the flag automatically turns OFF.	OFF to ON: TCP active open	---	Cleared	---	
	11	Send Request Switch for Socket No.2	When the flag changes from OFF to ON, transmission processing is executed. When transmission process is completed, the flag automatically turns OFF.	OFF to ON: Send	---	Cleared	---	
	12	Receive Request Switch for Socket No.2	When the flag changes from OFF to ON, reception processing is executed. When reception process is completed, the flag automatically turns OFF.	OFF to ON: Receive	---	Cleared	---	
	13	Close Request Switch for Socket No.2	When the flag changes from OFF to ON, close processing is executed. When close process is completed, the flag automatically turns OFF.	OFF to ON: Close	---	Cleared	---	
A572	00	UDP Open Request Switch for Socket No.3	When the flag changes from OFF to ON, UDP open processing is executed. When open process is completed, the flag automatically turns OFF.	OFF to ON: UDP open	---	Cleared	---	
	01	TCP Passive Open Request Switch for Socket No.3	When the flag changes from OFF to ON, TCP passive open processing is executed. When open process is completed, the flag automatically turns OFF.	OFF to ON: TCP passive open	---	Cleared	---	
	02	TCP Active Open Request Switch for Socket No.3	When the flag changes from OFF to ON, TCP active open processing is executed. When open process is completed, the flag automatically turns OFF.	OFF to ON: TCP active open	---	Cleared	---	
	03	Send Request Switch for Socket No.3	When the flag changes from OFF to ON, transmission processing is executed. When transmission process is completed, the flag automatically turns OFF.	OFF to ON: Send	---	Cleared	---	
	04	Receive Request Switch for Socket No.3	When the flag changes from OFF to ON, reception processing is executed. When reception process is completed, the flag automatically turns OFF.	OFF to ON: Receive	---	Cleared	---	
	05	Close Request Switch for Socket No.3	When the flag changes from OFF to ON, close processing is executed. When close process is completed, the flag automatically turns OFF.	OFF to ON: Close	---	Cleared	---	
A580	00 to 03	FB Communications Instruction Retries Count	Automatically stores the number of retries of FB communication instruction settings in the PLC Setup.	0 to F hex	Set the PLC Settings	Cleared	Written at the start of operation.	
A581	---	FB Communications Instruction Response Monitoring Time	Automatically stores the FB communication instruction response monitoring time settings in the PLC Setup.	0001 to FFFF hex (unit 0.1 s: 0.1 to 6553.5 s) 0000 Hex: 2 s	Set the PLC Settings	Cleared	Written at the start of operation.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A583	00	I/O Memory Backup Error Flag	The flag will be ON when the I/O memory is not held at power ON. It will still be ON until the memory area is cleared or the flag is turned OFF manually.	ON: I/O memory backup error occurred OFF: Normal	Retained	Retained	Refreshed when the power is turned ON.	A403.14
A616	00	Serial Port 1 (EX) Communication Settings	Display the present communication settings of the serial port 1 (EX). Reflect the PLC Setup when power is turned ON.	Parity 0: Even 1: Odd	Retained	See Function column.	Refreshed when power is turned ON.	
	01			Parity 0: Yes 1: No	Retained	See Function column.	Refreshed when power is turned ON.	
	02			Stop bit 0: 2 bits 1: 1 bit	Retained	See Function column.	Refreshed when power is turned ON.	
	03			Data length 0: 7 bits 1: 8 bits	Retained	See Function column.	Refreshed when power is turned ON.	
	04			Start bit 0: 1 bit (fixed)	Retained	See Function column.	Refreshed when power is turned ON.	
	08 to 11			Communication speed 0 hex: Default (9600) 3 hex: 1200 4 hex: 2400 5 hex: 4800 6 hex: 9600 7 hex: 19200 8 hex: 38400 9 hex: 57600 A hex: 115200	Retained	See Function column.	Refreshed when power is turned ON.	
	12 to 15			Communication mode 0 hex: Default (Host Link) 3 hex: Non-protocol 5 hex: Host Link 7 hex: Serial PLC Link (Slave) 8 hex: Serial PLC Link (Master) 9 hex: Modbus-RTU Easy Master	Retained	See Function column.	Refreshed when power is turned ON.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A617	00	Serial Port 1/ Built-in RS232C Port Communication Set-tings	Display the present communication settings of the serial port 1 or built-in RS232C port. Reflect the PLC Setup when power is turned ON.	Parity 0: Even 1: Odd	Retained	See Function column.	Refreshed when power is turned ON.	
	01			Parity 0: Yes 1: No	Retained	See Function column.	Refreshed when power is turned ON.	
	02			Stop bit 0: 2 bits 1: 1 bit	Retained	See Function column.	Refreshed when power is turned ON.	
	03			Data length 0: 7 bits 1: 8 bits	Retained	See Function column.	Refreshed when power is turned ON.	
	04			Start bit 0: 1 bit (fixed)	Retained	See Function column.	Refreshed when power is turned ON.	
	08 to 11			Communication speed 0 hex: Default (9600) 3 hex: 1200 4 hex: 2400 5 hex: 4800 6 hex: 9600 7 hex: 19200 8 hex: 38400 9 hex: 57600 A hex: 115200	Retained	See Function column.	Refreshed when power is turned ON.	
	12 to 15			Communication mode 0 hex: Default (Host Link) 2 hex: NT link (1: N) 3 hex: Non-protocol 5 hex: Host Link 7 hex: Serial PLC Link (Slave) 8 hex: Serial PLC Link (Master) 9 hex: Modbus-RTU Easy Master D hex: Modbus-RTU Slave	Retained	See Function column.	Refreshed when power is turned ON.	
A618	00	Serial Port 2/ Built-in RS-485 Port Communication Set-tings	Display the present communication settings of the serial port 2 or built-in RS-485 port. Reflect the PLC Setup when power is turned ON.	Parity 0: Even 1: Odd	Retained	See Function column.	Refreshed when power is turned ON.	
	01			Parity 0: Yes 1: No	Retained	See Function column.	Refreshed when power is turned ON.	
	02			Stop bit 0: 2 bits 1: 1 bit	Retained	See Function column.	Refreshed when power is turned ON.	
	03			Data length 0: 7 bits 1: 8 bits	Retained	See Function column.	Refreshed when power is turned ON.	
	04			Start bit 0: 1 bit (fixed)	Retained	See Function column.	Refreshed when power is turned ON.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A618	08 to 11	Serial Port 2 Built-in RS-485 Port Communication Settings	Display the present communication settings of the serial port 2 or Built-in RS-485 Port. Reflect the PLC Setup when power is turned ON.	Communication speed 0 hex: Default (9600) 3 hex: 1200 4 hex: 2400 5 hex: 4800 6 hex: 9600 7 hex: 19200 8 hex: 38400 9 hex: 57600 A hex: 115200	Retained	See Function column.	Refreshed when power is turned ON.	
	12 to 15			Communication mode 0 hex: Default (Host Link) 2 hex: NT link(1: N) 3 hex: Non-protocol 5 hex: Host Link 7 hex: Serial PLC Link (Slave) 8 hex: Serial PLC Link (Master) 9 hex: Modbus-RTU Easy Master D hex: Modbus-RTU Slave	Retained	See Function column.	Refreshed when power is turned ON.	
A638	00	Serial Port 1(EX) Modbus-RTU Easy Master Execution Flag (CP2E N☐☐-type CPU Unit only)	Turn ON this bit to send a command and receive a response for the serial port 1(EX) using the Modbus-RTU easy master function. **Note** This bit will be turned OFF automatically by the system when communications have been completed.	Turned ON: Execution started ON: Execution in progress. OFF: Not executed or execution completed.	Retained	Cleared	---	DM Area words for serial port 1(EX) Modbus-RTU Easy Master: D1400 to D1499
	01	Serial Port 1(EX) Modbus-RTU Easy Master Normal End Flag (CP2E N☐☐-type CPU Unit only)	ON when one command has been sent and the response received for the serial port 1(EX) using the Modbus-RTU easy master function.	ON: Execution normal. OFF: Execution error or still in progress.	Retained	Cleared	Written after reception.	
	02	Serial Port 1(EX) Modbus-RTU Easy Master Error End Flag (CP2E N☐☐-type CPU Unit only)	ON when an error has occurred in communications for the serial port 1(EX) using the Modbus-RTU easy master function. The error code is output to D1452 in the DM fixed allocation words for Modbus-RTU Easy Master.	ON: Execution error. OFF: Execution normal or still in progress.	Retained	Cleared	Written when error ends.	D1452

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A640	00	Serial Port 1/ Built-in RS-232C Port Modbus-RTU Easy Master Execution Bit	Turn ON this bit to send a command and receive a response for the serial port 1 or built-in RS-232C port using the Modbus-RTU easy master function. **Note** This bit will be turned OFF automatically by the system when communications have been completed.	Turned ON: Execution started ON: Execution in progress. OFF: Not executed or execution completed.	Retained	Cleared	---	DM Area words for serial port 1 or built-in RS-232C port Modbus-RTU Easy Master: D1200 to D1299
	01	Serial Port 1/ Built-in RS-232C Port Modbus-RTU Easy Master Normal End Flag	ON when one command has been sent and the response received for the serial port 1 or built-in RS-232C port using the Modbus-RTU easy master function.	ON: Execution normal. OFF: Execution error or still in progress.	Retained	Cleared	Written after reception.	
	02	Serial Port 1/ Built-in RS-232C Port Modbus-RTU Easy Master Error End Flag	ON when an error has occurred in communications for the serial port 1 or built-in RS-232C port using the Modbus-RTU easy master function. The error code is output to D1252 in the DM fixed allocation words for Modbus-RTU Easy Master.	ON: Execution error. OFF: Execution normal or still in progress.	Retained	Cleared	Written when error ends.	D1252
A641	00	Serial Port 2/ Built-in RS-485 Port Modbus-RTU Master Execution Bit (CP2E N30/40/60 or S□□-type CPU Unit only)	Turn ON this bit to send a command and receive a response for the serial port 2 or built-in RS-485 port using the Modbus-RTU easy master function. **Note** This bit will be turned OFF automatically by the system when communications have been completed.	Turned ON: Execution started ON: Execution in progress. OFF: Not executed or execution completed.	Retained	Cleared	---	DM Area words for serial port 2 or built-in RS-485 port Modbus-RTU Easy Master: D1300 to D1399
	01	Serial Port 2/ Built-in RS-485 Port Modbus-RTU Master Execution Normal Flag (CP2E N30/40/60 or S□□-type CPU Unit only)	ON when one command has been sent and the response received for the serial port 2 or built-in RS-485 port using the Modbus-RTU easy master function.	ON: Execution normal. OFF: Execution error or still in progress.	Retained	Cleared	Written after reception.	
	02	Serial Port 2/ Built-in RS-485 Port Modbus-RTU Master Execution Error Flag (CP2E N30/40/60 or S□□-type CPU Unit only)	ON when an error has occurred in communications for the serial port 2 or built-in RS-485 port using the Modbus-RTU easy master function. The error code is output to D1352 in the DM fixed allocation words for Modbus-RTU Easy Master.	ON: Execution error. OFF: Execution normal or still in progress.	Retained	Cleared	Written when error ends.	D1352
A720 to A722		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. A720.00 to A720.07: Seconds (00 to 59) A720.08 to A720.15: Minutes (00 to 59) A721.00 to A721.07: Hour (00 to 23) A721.08 to A721.15: Day of month (01 to 31) A722.00 to A722.07: Month (01 to 12) A722.08 to A722.15: Year (00 to 99) **Note** In an E□□-type CPU Unit or when time alignment cannot be executed with an N/S□□-type CPU Unit, the data will be for 1:01.01 on Sunday January 1, 2001.	See at left.	Retained	Retained	Written when power is turned ON.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A723 to A725		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. A723.00 to A723.07: Seconds (00 to 59) A723.08 to A723.15: Minutes (00 to 59) A724.00 to A724.07: Hour (00 to 23) A724.08 to A724.15: Day of month (01 to 31) A725.00 to A725.07: Month (01 to 12) A725.08 to A725.15: Year (00 to 99)	See at left.	Retained	Retained	Written when power is turned ON.	
A726 to A728		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. A726.00 to A726.07: Seconds (00 to 59) A726.08 to A726.15: Minutes (00 to 59) A727.00 to A727.07: Hour (00 to 23) A727.08 to A727.15: Day of month (01 to 31) A728.00 to A728.07: Month (01 to 12) A728.08 to A728.15: Year (00 to 99)	See at left.	Retained	Retained	Written when power is turned ON.	
A729 to A731		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. A729.00 to A729.07: Seconds (00 to 59) A729.08 to A729.15: Minutes (00 to 59) A730.00 to A730.07: Hour (00 to 23) A730.08 to A730.15: Day of month (01 to 31) A731.00 to A731.07: Month (01 to 12) A731.08 to A731.15: Year (00 to 99)	See at left.	Retained	Retained	Written when power is turned ON.	
A732 to A734		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. A732.00 to A732.07: Seconds (00 to 59) A732.08 to A732.15: Minutes (00 to 59) A733.00 to A733.07: Hour (00 to 23) A733.08 to A733.15: Day of month (01 to 31) A734.00 to A734.07: Month (01 to 12) A734.08 to A734.15: Year (00 to 99)	See at left.	Retained	Retained	Written when power is turned ON.	
A735 to A737		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. A735.00 to A735.07: Seconds (00 to 59) A735.08 to A735.15: Minutes (00 to 59) A736.00 to A736.07: Hour (00 to 23) A736.08 to A736.15: Day of month (01 to 31) A737.00 to A737.07: Month (01 to 12) A737.08 to A737.15: Year (00 to 99)	See at left.	Retained	Retained	Written when power is turned ON.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A738 to A740		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. A738.00 to A738.07: Seconds (00 to 59) A738.08 to A738.15: Minutes (00 to 59) A739.00 to A739.07: Hour (00 to 23) A739.08 to A739.15: Day of month (01 to 31) A740.00 to A740.07: Month (01 to 12) A740.08 to A740.15: Year (00 to 99)	See at left.	Retained	Retained	Written when power is turned ON.	
A741 to A743		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. A741.00 to A741.07: Seconds (00 to 59) A741.08 to A741.15: Minutes (00 to 59) A742.00 to A742.07: Hour (00 to 23) A742.08 to A742.15: Day of month (01 to 31) A743.00 to A743.07: Month (01 to 12) A743.08 to A743.15: Year (00 to 99)	See at left.	Retained	Retained	Written when power is turned ON.	
A744 to A746		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. A744.00 to A744.07: Seconds (00 to 59) A744.08 to A744.15: Minutes (00 to 59) A745.00 to A745.07: Hour (00 to 23) A745.08 to A745.15: Day of month (01 to 31) A746.00 to A746.07: Month (01 to 12) A746.08 to A746.15: Year (00 to 99)	See at left.	Retained	Retained	Written when power is turned ON.	
A747 to A749		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. A747.00 to A747.07: Seconds (00 to 59) A747.08 to A747.15: Minutes (00 to 59) A748.00 to A748.07: Hour (00 to 23) A748.08 to A748.15: Day of month (01 to 31) A749.00 to A749.07: Month (01 to 12) A749.08 to A749.15: Year (00 to 99)	See at left.	Retained	Retained	Written when power is turned ON.	

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A751	11	DM Backup Restore Failed Flag	ON when DM backup data could not be restored normally. If this flag turns ON, data will not be restored from the built-in Flash Memory to RAM.	ON: Restore failed OFF: Execution normal	Retained	Cleared	Written when fail to restore.	
	14	DM Backup Save Flag	ON when A751.15 is turned ON to start the saving operation. This flag stays ON while data is being saved and turns OFF when finished.	ON: Saving OFF: Not saving	Retained	Cleared	Written when save starts	
	15	DM Backup Save Start Bit	Saving the specified words from the DM Area in RAM to the built-in Flash Memory is started when this bit is turned ON. This bit will not turn OFF automatically even when saving the data has been completed. If this bit is turned ON and OFF while the DM Backup Save Flag (A751.14) is ON, it will be ignored and the data will not be backed up again. **Note** Select the *Restore D0- from backup memory* Check Box and set the number of DM Area words to back up in the *Number of CH of DM for backup* Box in the PLC Setup before using this bit.	ON: Start saving. OFF: Execution normal or still in progress.	Retained	Cleared	---	

A-3 Response Performance

A-3-1 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 ladder programs are 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.
- The cycle time

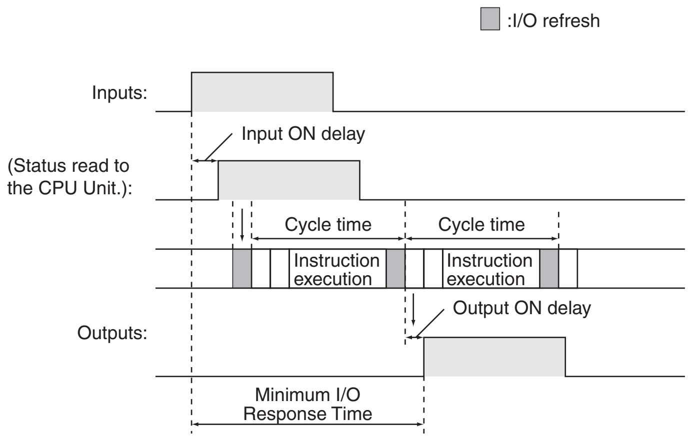
● Minimum I/O Response Time

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.

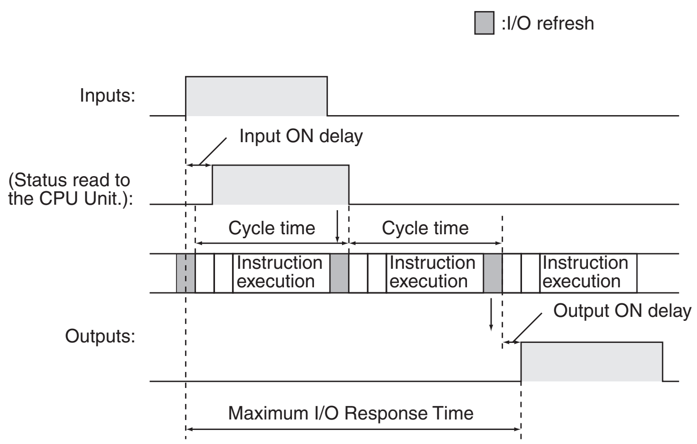


● Maximum I/O Response Time

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 × 2) + Output ON delay



● Calculation Example

Conditions:

Input ON delay: 1 ms (normal input 0.08 to 0.11 with input constant set to 0 ms)

Output ON delay: 0.1 ms (transistor output)

Cycle time: 20 ms

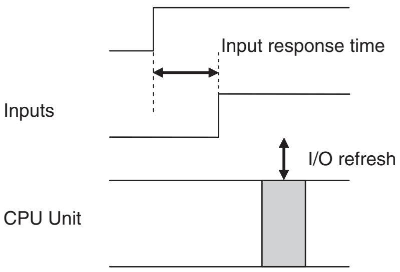
Minimum I/O response time = 1 ms + 20 ms + 0.1 ms = 21.1 ms

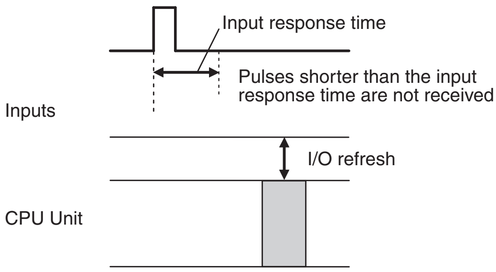
Maximum I/O response time = 1 ms + (20 ms × 2) + 0.1 ms = 41.1 ms

Input Constant Setting

Input constant setting can be set in the PLC Setup.

Increasing the input constants slowers the input response time and reduces the effects of chattering and noise. Decreasing the input constants fasters the input response time and allows reception of shorter input pulses (but the pulse width must be longer than the cycle time).





● PLC Setup

Name	Description	Setting	Default
Input Constant Settings	Input Constants	00 hex: 8 ms 10 hex: No filter (0 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	00 hex (8 ms)

Note The input constants of CP1W-40EDR/EDT/EDT1 are always 16ms regardless of the settings.

A-3-2 Interrupt Response Time

● Interrupt Response Time for Input Interrupt Tasks

The interrupt response time for input interrupt tasks is the time taken from when a built-in input has turned ON (or OFF) until the input interrupt task has actually been executed.

The length of the interrupt response time for input interrupt tasks depends on the total of the hardware interrupt response time and software interrupt response time.

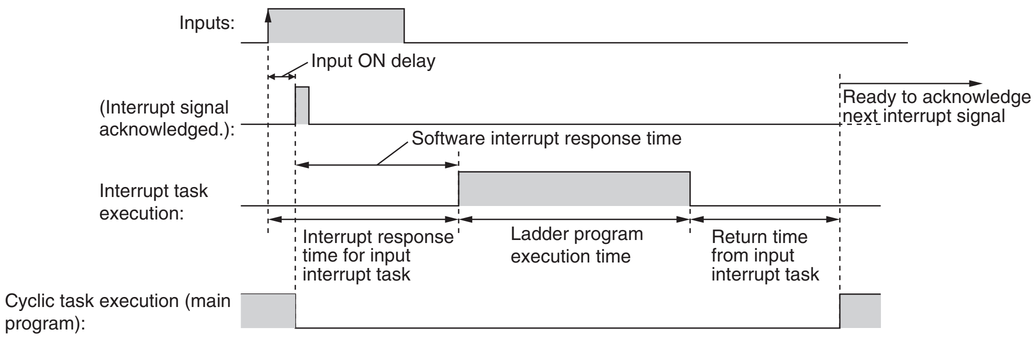
Item	Interrupt response time	Counter 0~5 interrupts
Hardware interrupt response time	Upward differentiation: 50 μs	–
	Downward differentiation: 50 μs	–
Software interrupt response time	Minimum: 23 μs + Wait time*	Minimum: 40 μs + Wait time*

* The wait time occurs when there is competition with other interrupts. As a guideline, the wait time will be 0 to 3 ms.

Note Input interrupt tasks can be executed during execution of the user program, I/O refresh, peripheral servicing, or overseeing. (Even if an instruction is being executed, execution of the instruction will be stopped to execute the interrupt task.)
The interrupt response time is not affected by the above processing operations during which the interrupt inputs turns ON.
Input interrupts, however, are not executed during execution of other interrupt tasks even if the input interrupt conditions are satisfied. Instead, the input interrupts are executed after the current interrupt task has completed execution and the software interrupt response time has elapsed.

The interrupt response time of input interrupt tasks is calculated as follows:

Interrupt response time = Input ON delay + Software interrupt response time



The time from when execution of the input interrupt task is completed until execution of the cyclic task is resumed is 11 μs.

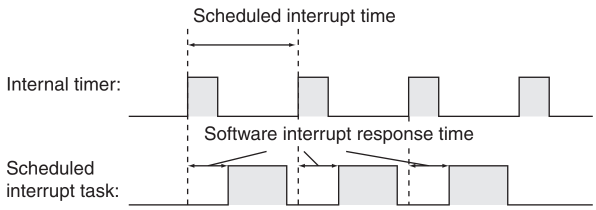
● Interrupt Response Time for Scheduled Interrupt Tasks

The interrupt response time for scheduled interrupt tasks is the time taken from after the scheduled time specified by the MSKS instruction has elapsed until the interrupt task has actually been executed.

The length of the interrupt response time for scheduled interrupt tasks is 0.1 ms max.

There is also an error of 10 μs in the time to the first scheduled interrupt (1.0 ms min.).

Note Scheduled interrupt tasks can be executed during execution of the ladder program (even while an instruction is being executed by stopping the execution of an instruction), I/O refresh, peripheral servicing, or overseeing. The processing operation in which the scheduled interrupt occurs does not affect the interrupt processing time.
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.As a guideline, the wait time will be 0 to 3ms.



Precautions for Correct Use

The scheduled task will not be executed while the CPU Unit suspends operation for online editing.

A-3-3 Serial PLC Link Response Performance

The response times for CPU Units connected via a Serial PLC Link (polling unit to polled unit or polled unit to polling unit) can be calculated as shown below.

Note A PT cannot be used in the Serial PLC Link.

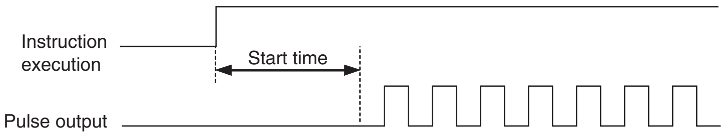
- Maximum I/O response time (not including hardware delay) =
  Polling unit cycle time + Communications cycle time + Polled unit cycle time + 4 ms
- Minimum I/O response time (not including hardware delay) =
  Polled unit communications time + 0.8 ms

Number of participating polled unit nodes	The number of polled units to which links have been established within the maximum unit number set in the polling unit.
Number of non-participating polled unit nodes	The number of polled units not participating in the links within the maximum unit number set in the polling unit.
Communications cycle time (ms)	Polled unit communications time × Number of participating polled unit nodes + 10 × Number of non-participating polled unit nodes + 20 × Number of polled unit nodes
Polled unit communications time (ms)	• Communications time set to *Standard:* 0.4 + 0.286 × [(No. of polled units + 1) × No. of link words × 2 + 12] • Communications time set to *Fast*: 0.4 + 0.0955 × [(No. of polled units + 1) × No. of link words × 2 + 12]

A-3-4 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	Minimum: 500μs + Wait time*
SPED: independent	
ACC: continuous	
ACC: independent, trapezoidal	
ACC: independent, triangular	
PLS2: trapezoidal	
PLS2: triangular	
IFEED: interrupt feeding	
ITPL: linear interpolation	

* The wait time occurs when there is competition with other interrupts. As a guideline, the wait time will be 0 to 3 ms.

A-3-5 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	Minimum: 100μs + Wait time*1 + 1 pulse output time
SPED: immediate stop	
ACC: deceleration stop	1 control cycle (4 ms) minimum, 2 control cycles (8 ms) maximum*2
PLS2: deceleration stop	
SPED: speed change	
ACC: speed change	
PLS2: target position change in reverse direction	
PLS2: target position change in same direction at same speed	
PLS2: target position change in same direction at different speed	

*1 The wait time occurs when there is competition with other interrupts. As a guideline, the wait time will be 0 to 3 ms.

*2 When the frequency of pulse output is lower than 250Hz, 1 control cycle will be equal to the time of pulse output.

Example: The change response time of 100Hz pulse output is above 1 control cycle (10ms) and below 2 control cycles (20ms).

A-4 PLC Operation for Power Interruptions

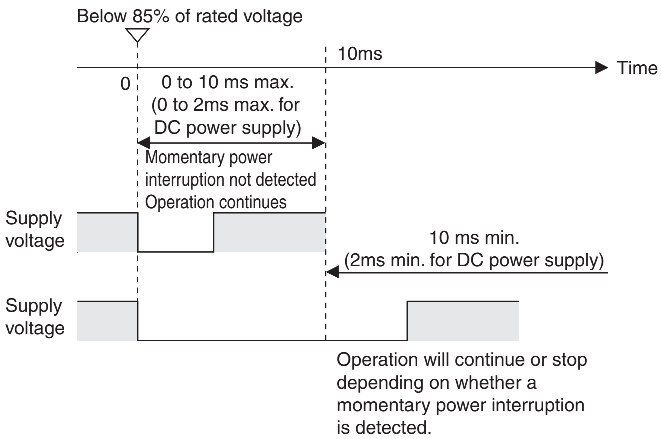
Overview of Operation for Power Interruptions

● Power Supply Voltage Drop

If the power supply voltage falls below the specified value (85% of rated voltage) while the CPU Unit is in RUN or MONITOR mode, operation will be stopped and all outputs will be turned OFF.

● Detection of Momentary Power Interruptions

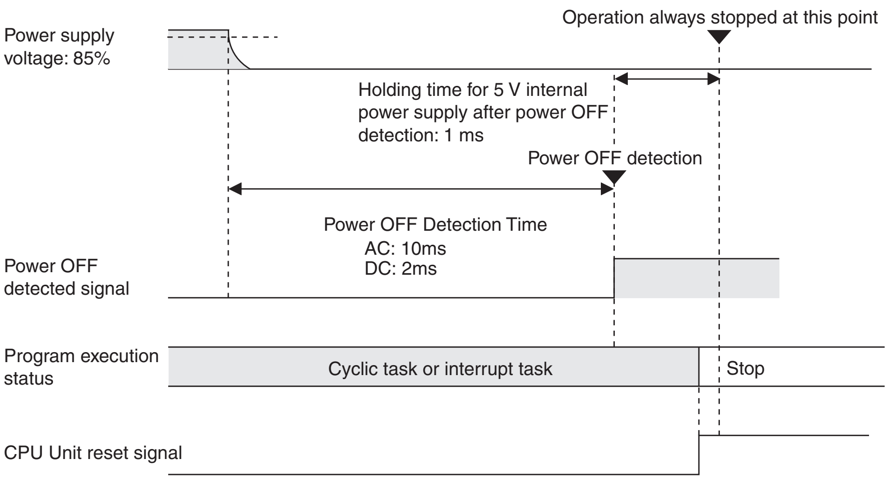
The system will continue to run if the momentary power interruption lasts less than 10 ms (2ms for DC power supply). If power is interruped for longer than 10 ms (2ms for DC power supply), a momentary power interruption will be detected or undetected. If the momentary power interruption is detected, the CPU Unit will be stopped and outputs will be turned OFF.



● Automatic Recovery

Operation is automatically restarted when the power supply voltage is restored.

Power OFF Timing Chart



Power OFF Detection Time:	The time from when the power supply voltage drops to 85% or less the rated voltage until the power interruption is detected.
Power Holding Time:	The maximum amount of time (fixed at 1 ms) that 5 V will be held internally after power shuts OFF.

● Description of Operation

The power interruption will be detected if the 100 to 240 VAC power supply falls below 85% of the minimum rated voltage for the power OFF detection time (10 ms minimum for AC power supply and 2ms minimum for DC power supply, not fixed).

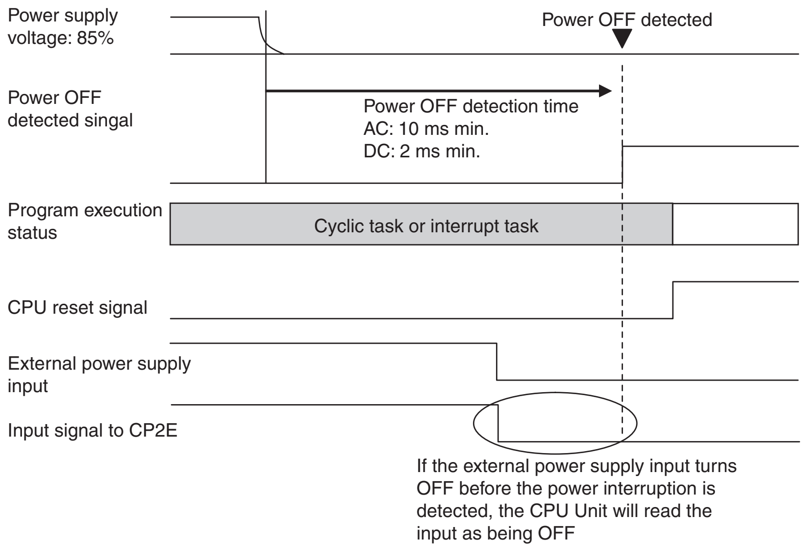
The CPU reset signal will turn ON and the CPU Unit will be reset immediately.

Instruction Execution for Power Interruptions

The power OFF detection time of CP2E CPU Units is 10 ms minimum for AC power supply and 2ms minimum for DC power supply. If the power interruption is detected when operating in RUN or MONITOR mode, the instruction currently being executed will be stopped and then the CPU Unit will be reset.

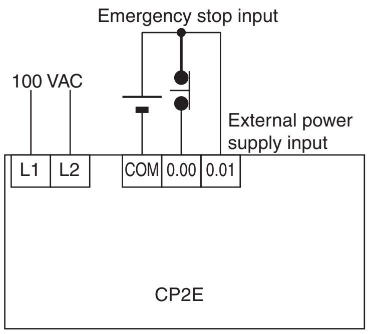
Malfunction Countermeasures

If only a couple of Expansion I/O Units or Expansion Units are connected to the CPU Unit resulting in a light power supply circuit load and a small current consumption, the time required by the CPU Unit to detect a power interruption will be longer. For this reason, inputs may be incorrectly identified as being OFF if external power supply used for an input turns OFF before the power interruption is detected. If an external NC contact input is used or the ladder program counts the number of ON to OFF transitions, a malfunction may occur if the external power supply turns OFF.

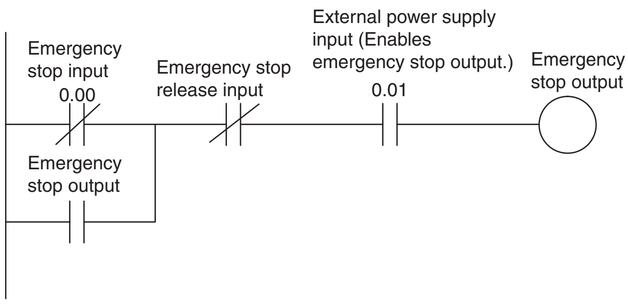


The following diagram shows an example countermeasure for this situation.

- Wiring



- Ladder Program



A-5 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 FIND MAXIMUM (MAX(182)), and FIND MINIMUM (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 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)).

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 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.

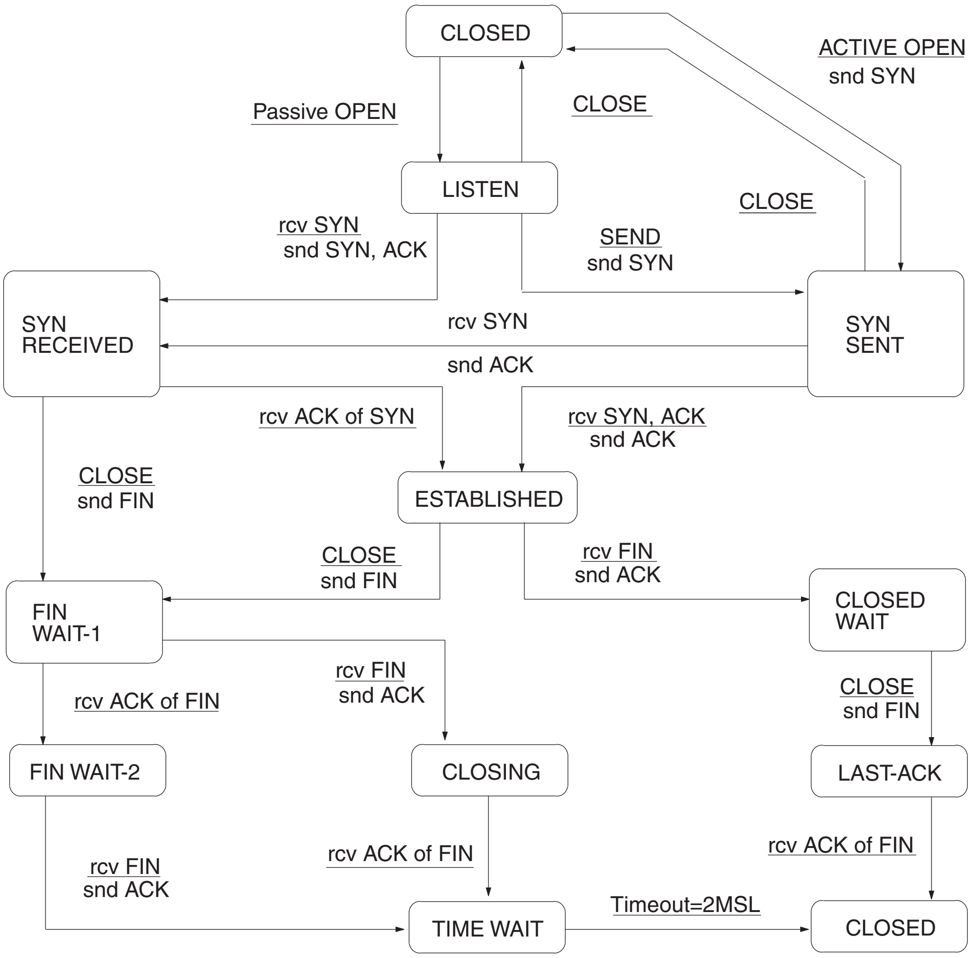
Memory Map

Classification	PLC memory address (hex)	User addresses	Area
I/O memory areas	0 to 0174F	---	Reserved for system
	01750 to 0176F	T00 to T31	Timer Completion Flags
	01770 to 0178F	C00 to C31	Counter Completion Flags
	01790 to 017BF	---	Reserved for system
	017C0 to 0197F	A0 to A447	Read-only Auxiliary Area
	01980 to 01B7F	A448 to A959	Read/Write Auxiliary Area
	01B80 to 01CA1	CIO 0 to 289	CIO Area
	01CA2 to 01CBF	---	Reserved for system
	01CC0 to 01D3F	H0 to H127	Holding Area
	01D40 to 022BF	---	Reserved for system
	022C0 to 0233F	W0 to W127	Work Area
	02340 to 0253F	T000 to T511	Timer PVs
	02540 to 0273F	C000 to C511	Counter PVs
	E□□-type CPU Unit: 02740 to 0373F S□□-type CPU Unit: 02740 to 0473F N□□-type CPU Unit: 02740 to 0673F	E□□-type CPU Unit: D0 to D4095 S□□-type CPU Unit: D0 to D8191 N□□-type CPU Unit: D0 to D16383	DM Area
	E□□-type CPU Unit: 03740 to 0673F S□□-type CPU Unit: 04740 to 0673F	---	Reserved for system

Note Do not access areas reserved for the system.

A-6 Ethernet Functions

A-6-1 TCP Status Transitions

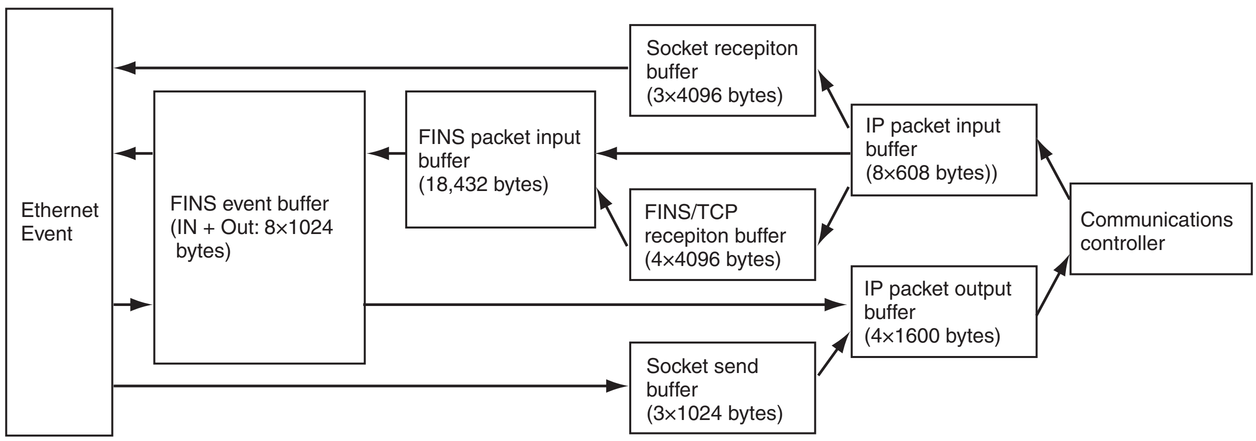


Number	Status	Meaning
00000000	CLOSED	Connection closed.
00000001	LISTEN	Waiting for connection.
00000002	SYN SENT	SYN sent in active status.
00000003	SYN RECEIVED	SYN received and sent.
00000004	ESTABLISHED	Already established.
00000005	CLOSE WAIT	FIN received and waiting for completion.
00000006	FIN WAIT 1	Completed and FIN sent.
00000007	CLOSING	Completed and exchanged FIN. Awaiting ACK.
00000008	LAST ACK	FIN sent and completed. Awaiting ACK.
00000009	FIN WAIT 2	Completed and ACK received. Awaiting FIN.
0000000A	TIME WAIT	After closing, pauses twice the maximum segment life (2MSL).

A-6-2 Ethernet Network Parameters

Parameter	Value	Description
Hold timer	18 s	The hold timer is used for active open processing of TCP sockets. An ETIMEDOUT error will occur if connection is not completed within 18 s.
Resend timer	3 s	The resend timer is used to monitor completion of reception of arrival confirmations when transferring data via socket services. If the timer setting is exceeded before arrival confirmation is received, data is resent. Resends are performed through the 12th timeout (3 s). An ETIMEDOUT error will occur after the 12th timeout.
2MSL timer	120 s	The 2MSL timer starts at the TCP socket that first closes the socket and will run for 120 s in the TIME_WAIT status.
ARP timer	20 min/5 s	If a complete ARP table entry (with an Ethernet address) is not referred to for 20 minutes, it is removed from the table. An incomplete ARP table entry (no response yet returned to the ARP request) is removed from the table after 5 seconds.
Window size	2,144 bytes	The initial value of the maximum capacity used to control the convergence of TCP sockets. Actually, the node negotiates with the remote node and uses the smaller of the values for the two nodes. The window size will fluctuate with the available space in the TCP reception buffers of the remote node when processing communications.
Segment size (MSS)	536 bytes	TCP data is separated into 536-byte units.
TTL (Time to Live)	30 times	Decreases 1 each time to comnunicate through the IP roater.

A-6-3 Buffer Configuration



Index

Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.

Cat. No. W614-E1-01

Revision code

Revision code	Date	Revised content
01	September 2019	Original production

OMRON Corporation **Industrial Automation Company**
Kyoto, JAPAN

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 ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

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 (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:

Cat. No. W614-E1-01 0919