Cat. No. Z901-E1-05

SYSMAC NE1S Series
NE1S-CPU01
Programmable Controller

OPERATION MANUAL

OMRON

NE1S Series
NE1S-CPU01
Programmable Controller
Operation Manual

Revised December 2006

Notice:

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

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

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

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

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

OMRON Product References

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

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

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

Visual Aids

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

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

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

Trademarks and Copyrights

ControlNet and EtherNet/IP are registered trademarks of ControlNet International.

DeviceNet is a registered trademark of the Open DeviceNet Vendors Association.

Microsoft, Windows, Windows 95, Windows 98, Windows Me, Windows NT, Windows 2000, and Windows XP are registered trademarks of the Microsoft Corporation.

The realtime operating system for this product is equipped with Hyper-ITRON by Elmic Systems, Inc.

Other product names and company names in this manual are trademarks or registered trademarks of their respective companies.

The copyrights for the NE1S PLC belong to OMRON Corporation.

Version Upgrade Guide

Functionality Improved in Rev. 3.1

The following tables list the improvements made in the upgrade to NE1S-CPU01 Rev. 3.1.

Online Editing includes Comments

Earlier versions	Rev. 3.1
When using online editing to edit multiple program rungs, the rungs could not be edited if there were rung comments in the selected rungs.	When using online editing to edit multiple program rungs, the rungs can be edited if there are rung comments in the selected rungs. The rung comments themselves cannot be edited, inserted, or deleted.

Connecting Online through the Network without Creating an I/O Table

Earlier versions	Rev. 3.1
When a network connection is established but the I/O table was not created, it was not possible to connect online with network PLCs from the NE Programmer/Network Configurator because the Network Communications Unit itself was not recognized from the CPU Unit.	Even if an I/O table has not been created (no registered I/O table), the CPU Unit can still recognize CPU Bus Units such as Network Communications Units. Consequently, a variety of operations can be performed from a computer-based Programming Device such as the NE Programmer when a network connection is established, even if the I/O table has not been created. These operations include connecting online to PLCs in the network, creating the I/O table (or editing and transferring the I/O table), and transferring the user program, PLC Setup, and CPU Bus Unit Setup.

Operation Start Time and Stop Time Record Function

Earlier versions	Rev. 3.1
The time/date that operation started and stopped could not be recorded in the CPU Unit.	The time and date that operation starts and stops is automatically recorded in the Auxiliary Area. • The time/date (year, month, day, hour, minute, and second) that operation started (by switching to RUN or MONITOR mode) is stored in words A515 to A517. • The time/date (year, month, day, hour, minute, and second) that operation stopped (by a fatal error or switching to PROGRAM mode) is stored in words A518 to A520. This function simplifies time management of the PLC system.

Transferring Screens to NS-series PTs through the USB Port

Earlier versions	Rev. 3.1
Screens could not be transferred to an NS-series PT through the CPU Unit. Screens had to be transferred from the computer to the PT through a direct connection such as serial, USB, or Ethernet. When the CPU Unit was connected through the RS-232C port, the computer had to connect to the PT's serial port B, USB port, or Ethernet in order to transfer screens. Therefore, a separate cable had to be used to connect to the CPU Unit or a single cable had to be switched when transferring screens.	Screens can be transferred from a computer connected to the USB port to an NS-series PT connected to the RS-232C port. Therefore, it is no longer necessary to wire the connection separately or switch cables when connecting an NS-series PT to a CPU Unit by RS-232C.

Serial Gateway (Converting from FINS to CompoWay/F at the Built-in Serial Port)

Earlier versions	Rev. 3.1
If you wanted to send standard CompoWay/F commands from the PLC to access OMRON components such as Temperature Controllers or Panel Meters, it was necessary to the Serial Communications Unit’s protocol macro function and execute a PMCR instruction in the CPU Unit’s ladder program to operate as a standard CompoWay/F Master. Since the protocol macro function was used, devices could not be accessed over the network.	Rev. 3.1 and later versions of the NE1S CPU Units can receive FINS commands (CompoWay/F commands encapsulated in frames by FINS) at the built-in USB port, automatically convert the FINS frame to the CompoWay/F command frame, and transfer it over the RS-232C circuit.

Free-running Counters

Earlier versions	Rev. 3.1
This function was not supported.	Auxiliary Area words (A000 and A001) contain system timers that start when the power is turned ON. The content of A000 is automatically incremented by 1 every 10 ms while the power is ON. The content is returned to 0000 hex after reaching FFFF hex (655,350 ms), and then continues to operate in ring mode. The content of A001 is automatically incremented by 1 every 100 ms while the power is ON. The content is returned to 0000 hex after reaching FFFF hex (6,553,500 ms), and then continues to operate in ring mode.

Write-protecting the CPU Unit from Write Operations over the Network

Earlier versions	Rev. 3.1
There was no special setting to prevent write processing or write operations on the PLC’s CPU Unit from network connections such as Ethernet (connections other than direct serial and USB connections).	The CPU Unit can be protected against write operations and manipulation operations over the network from the NE Programmer, NE OPC Server, and CommLinkServer (connections other than direct serial connections). Read operations are not affected. The FINS Write-protection prevents downloading of data such as the user program, PLC Setup, and I/O memory, changing the operating mode, online editing, and other write processes.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

SECTION 10
Outline of NE Programmer and Its Functions. 263

SECTION 11
Programming 285

SECTION 12
PLC System Configuration 341

SECTION 13
Online Operation 349

TABLE OF CONTENTS

TABLE OF CONTENTS

Appendices

About this Manual:

This manual describes the installation and operation of the NE1S-series Programmable Controller (PLC) and includes the sections described below.

Please read this manual and all related manuals listed in the following table and be sure you understand information provided before attempting to install or use an NE1S-series CPU Unit in a PLC System.

Name	Cat. No.	Contents
SYSMAC NE1S Series NE1S-CPU01 Programmable Controller Operation Manual (this manual)	Z901	Provides an outlines of and describes the design, installation, maintenance, and other basic operations for the NE1S-series PLC. Also provides information on how to use the NE Programmer.
NE1S-CNS21U ControlNet Unit Operation Manual	Z902	Describes the use of the ControlNet Unit.
NE1S-DRM21U DeviceNet Unit Operation Manual	Z903	Describes the use of the DeviceNet Unit.

Please read this manual carefully and be sure you understand the information provided before attempting to install or operate an NE1S-series PLC.

Section 1 introduces the NE1S, provides CPU Unit specifications, and describes the system configuration. It also provides a table outlining the differences between the NE1S and CS/CJ-series PLCs.

Section 2 describes the structure of the programs.

Section 3 describes hardware and software aspects of the CPU Unit.

Section 4 describes the settings in the PLC Setup and how they are used to control CPU Unit operation.

Section 5 describes the structure and functions of the I/O Memory Areas and Parameter Areas.

Section 6 describes how to install a PLC System, including mounting the various Units and wiring the System. Be sure to follow the instructions carefully. Improper installation can cause the PLC to malfunction, resulting in very dangerous situations.

Section 7 introduces the Network Configurator.

Section 8 provides an outline of the functions of the NE Program.

Section 9 provides details on programming.

Section 10 describes the configuration of the PLC system.

Section 11 provides the procedures for online operation.

Section 12 describes the functions used to manipulate file memory.

Section 13 describes the internal operation of the CPU Unit and the cycle used to perform internal processing.

Section 14 provides information on hardware and software errors that occur during PLC operation.

Section 15 provides inspection and maintenance information.

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

Read and Understand this Manual

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

Warranty and Limitations of Liability

WARRANTY

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

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

LIMITATIONS OF LIABILITY

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

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

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

Application Considerations

SUITABILITY FOR USE

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

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

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

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

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

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

PROGRAMMABLE PRODUCTS

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

Disclaimers

CHANGE IN SPECIFICATIONS

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

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

DIMENSIONS AND WEIGHTS

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

PERFORMANCE DATA

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

ERRORS AND OMISSIONS

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

PRECAUTIONS

This section provides general precautions for using the NE1S-series Programmable Controller (PLC) and related devices.

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

1 Intended Audience

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

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

2 General Precautions

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

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

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

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

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

3 Safety Precautions

⚠ WARNING The CPU Unit refreshes I/O even when the program is stopped (i.e., even in PROGRAM mode). Confirm safety thoroughly in advance before changing the status of any part of memory allocated to I/O Units, Special I/O Units, or CPU Bus Units. Any changes to the data allocated to any Unit may result in unexpected operation of the loads connected to the Unit. Any of the following operation may result in changes to memory status.

- Transferring I/O memory data to the CPU Unit from a Programming Device.
- Changing present values in memory from a Programming Device.
- Force-setting/-resetting bits from a Programming Device.
- Transferring I/O memory files from a Memory Card to the CPU Unit.
- Transferring I/O memory from a host computer or from another PLC on a network.

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

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

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

⚠ WARNING Do not touch the Power Supply Unit while power is being supplied or immediately after power has been turned OFF. Doing so may result in electric shock.

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

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

- Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.
- The PLC will turn OFF all outputs when its self-diagnosis function detects any error or when a severe failure alarm (FALS) instruction is executed. As a countermeasure for such errors, external safety measures must be provided to ensure safety in the system.
- The PLC outputs may remain ON or OFF due to deposition or burning of the output relays or destruction of the output transistors. As a countermeasure for such problems, external safety measures must be provided to ensure safety in the system.
- When the 24-V DC output (service power supply to the PLC) is overloaded or short-circuited, the voltage may drop and result in the outputs being turned OFF. As a countermeasure for such problems, external safety measures must be provided to ensure safety in the system.

⚠ WARNING Confirm the safety of the system before transferring data files from file memory (e.g., on a Memory Card) to actual I/O words in the CPU Unit's CIO Area using a Programming Device. Transferring data without confirming the safety of the system may result in malfunctions in devices connected to Output Units, regardless of the CPU Unit's operating mode.

⚠ Caution Fail-safe measures must be taken by the customer to ensure safety in the event of incorrect, missing, or abnormal signals caused by broken signal lines, momentary power interruptions, or other causes. Serious accidents may result from abnormal operation if proper measures are not provided.

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

⚠ **Caution** Confirm safety at the destination node before editing or transferring a program, PLC Setup, I/O table, I/O memory data, or parameter data to another node. Doing either of these without confirming safety may result in unexpected operation and injury.

⚠ **Caution** Tighten the screws on the terminal block of the AC Power Supply Unit to the torque specified in the operation manual. The loose screws may result in burning or malfunction.

4 Operating Environment Precautions

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

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

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

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

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

5 Application Precautions

Observe the following precautions when using the PLC System.

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

- Always connect to a ground of 100 Ω or less when installing the Units. Not connecting to a ground of 100 Ω or less may result in electric shock.
- A ground of 100 Ω or less must be installed when shorting the GR and LG terminals on the Power Supply Unit.
- Always turn OFF the power supply to the PLC before attempting any of the following. Not turning OFF the power supply may result in malfunction or electric shock.
  - Mounting or dismounting Power Supply Units, I/O Units, CPU Units, or any other Units.
  - Assembling the Units.
  - Setting DIP switches or rotary switches.
  - Connecting cables or wiring the system.
  - Connecting or disconnecting the connectors.

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

- Mount the Unit only after checking the connectors and terminal blocks completely.
- Always use the power supply voltage specified in the operation manuals. An incorrect voltage may result in malfunction or burning.
- Take appropriate measures to ensure that the specified power with the rated voltage and frequency is supplied. Be particularly careful in places where the power supply is unstable. An incorrect power supply may result in malfunction.
- Install external breakers and take other safety measures against short circuiting in external wiring. Insufficient safety measures against short-circuiting may result in burning.
- Use crimp terminals for wiring. Do not connect bare stranded wires directly to terminals. Connection of bare stranded wires may result in burning.
- Do not apply voltages to the Input Units in excess of the rated input voltage. Excess voltages may result in burning.
- Do not apply voltages or connect loads to the Output Units in excess of the maximum switching capacity. Excess voltage or loads may result in burning.
- Install the Unit properly as specified in the operation manual. Improper installation of the Unit may result in malfunction.
- Be sure that all the terminal screws and cable connector screws are tightened to the torque specified in the relevant manuals. Incorrect tightening torque may result in malfunction.

- Leave the label attached to the Unit when wiring. Removing the label may result in malfunction.
- Remove the label after the completion of wiring to ensure proper heat dissipation. Leaving the label attached may result in malfunction.
- Be sure that the terminal blocks, connectors, expansion cables, and other items with locking devices are properly locked into place. Improper locking may result in malfunction.
- Disconnect the functional ground terminal when performing withstand voltage tests. Not disconnecting the functional ground terminal may result in burning.
- Wire correctly and double-check all the wiring or the setting switches before turning ON the power supply. Incorrect wiring may result in burning.
- Check that the DIP switches and data memory (DM) are properly set before starting operation.
- Before actual operation, check the parameter settings and user program (such as the ladder program) for proper execution in trial operation. Always check the program before transferring it.
- Resume operation only after transferring to the new CPU Unit and/or Special I/O Units the contents of the DM and HR Areas required for resuming operation. Not doing so may result in an unexpected operation.
- Confirm that no adverse effect will occur in the system before attempting any of the following. Not doing so may result in an unexpected operation.
    - Changing the operating mode of the PLC.
    - Force-setting/force-resetting any bit in memory.
    - Changing the present value of any word or any set value in memory.
    - Restoring the values of variables.
- Do not pull on the cables or bend the cables beyond their natural limit. Doing either of these may break the cables.
- Do not place objects on top of the cables. Doing so may break the cables.
- When replacing parts, be sure to confirm that the rating of a new part is correct. Not doing so may result in malfunction or burning.
- Before touching the Unit, be sure to first touch a grounded metallic object in order to discharge any static built-up. Not doing so may result in malfunction or damage.
- Do not turn OFF the power supply to the Unit while data is being transferred.
- When transporting or storing the product, cover the PCBs with electrically conductive materials to prevent LSIs and ICs from being damaged by static electricity, and also keep the product within the specified storage temperature range.
- Do not touch the mounted parts or the rear surface of PCBs because PCBs have sharp edges such as electrical leads.
- Double-check the pin numbers when assembling and wiring the connectors.
- Wire correctly according to the procedures specified in relevant manuals.
- Confirm that a Compact Flash Card containing the correct contents is inserted before starting operation.
- Be sure to set the network connection settings and network parameters correctly.

- The BKUP indicator lights when data is being written to flash memory. Do not turn OFF the power supply to the CPU Unit when the BKUP indicator is lit. The data may not be written correctly.
- Set the startup mode only after confirming that the controlled facilities will not be adversely affected.
- Do not turn OFF the power supply to the CPU Unit while a Memory Card is being accessed.
- Do not remove a Memory Card while it is being accessed. Press the Memory Card power button and confirm that the BUSY indicator goes out before removing the Memory Card.
- The user program is stored in nonvolatile memory, and operation is possible even if the Battery voltage has dropped or a Battery is not installed. (Operation will not be stopped for a memory error.) Data in the DM and EM Areas, however, will not be stable without a Battery that is fully charged. If data from the DM or EM Areas is used to control outputs from the program, used the Battery Error Flag to control outputs or perform other measures to ensure safety.
- When replacing the Battery, turn ON the power supply to the CPU Unit for at least 1 minute and then complete the Battery replacement operation within 5 minutes. Memory may be corrupted if Battery replacement is not completed within 5 minutes.

6 Conformance to EC Directives

6-1 Applicable Directives

- EMC Directives
- Low Voltage Directive

6-2 Concepts

EMC Directives

OMRON devices that comply with EC Directives also conform to the related EMC standards so that they can be more easily built into other devices or the overall machine. The actual products have been checked for conformity to EMC standards (see the following note). Whether the products conform to the standards in the system used by the customer, however, must be checked by the customer.

EMC-related performance of the OMRON devices that comply with EC Directives will vary depending on the configuration, wiring, and other conditions of the equipment or control panel on which the OMRON devices are installed. The customer must, therefore, perform the final check to confirm that devices and the overall machine conform to EMC standards.

Note Applicable EMC (Electromagnetic Compatibility) standards are as follows:

EMS (Electromagnetic Susceptibility): EN61131-2 or EN61000-6-2
EMI (Electromagnetic Interference): EN61000-6-4
(Radiated emission: 10-m regulations)

Low Voltage Directive

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

6-3 Conformance to EC Directives

The NE1S-series PLCs comply with EC Directives. To ensure that the machine or device in which the NE1S-series PLC is used complies with EC Directives, the PLC must be installed as follows:

1,2,3...
1. The NE1S-series PLC must be installed within a control panel.
2. You must use reinforced insulation or double insulation for the DC power supplies used for the communications power supply and I/O power supplies.
3. NE1S-series PLCs complying with EC Directives also conform to the Common Emission Standard (EN61000-6-4). Radiated emission characteristics (10-m regulations) may vary depending on the configuration of the control panel used, other devices connected to the control panel, wiring, and other conditions. You must therefore confirm that the overall machine or equipment complies with EC Directives.

6-4 Relay Output Noise Reduction Methods

The NE1S-series PLCs conforms to the Common Emission Standards (EN61000-6-4) of the EMC Directives. However, noise generated by relay output switching may not satisfy these Standards. In such a case, a noise filter must be connected to the load side or other appropriate countermeasures must be provided external to the PLC.

Countermeasures taken to satisfy the standards vary depending on the devices on the load side, wiring, configuration of machines, etc. Following are examples of countermeasures for reducing the generated noise.

Countermeasures

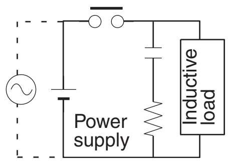
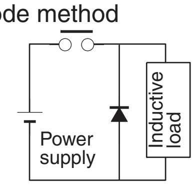
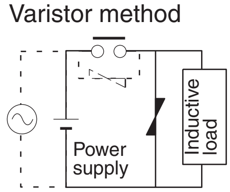
(Refer to EN61000-6-4 for more details.)

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

Countermeasures are required if the frequency of load switching for the whole system with the PLC included is more than 5 times per minute.

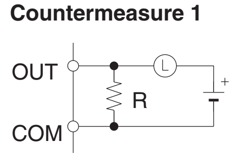
Countermeasure Examples

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

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

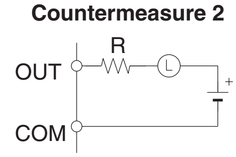
When switching a load with a high inrush current such as an incandescent lamp, suppress the inrush current as shown below.

Countermeasure 1



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

Countermeasure 2



Providing a limiting resistor

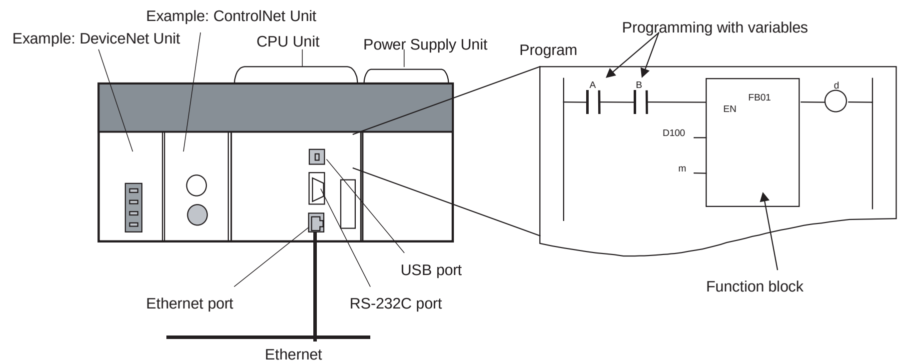
SECTION 1
Introduction

This section introduces the NE1S, provides CPU Unit specifications, and describes the system configuration. It also provides a table outlining the differences between the NE1S and CS/CJ-series PLCs.

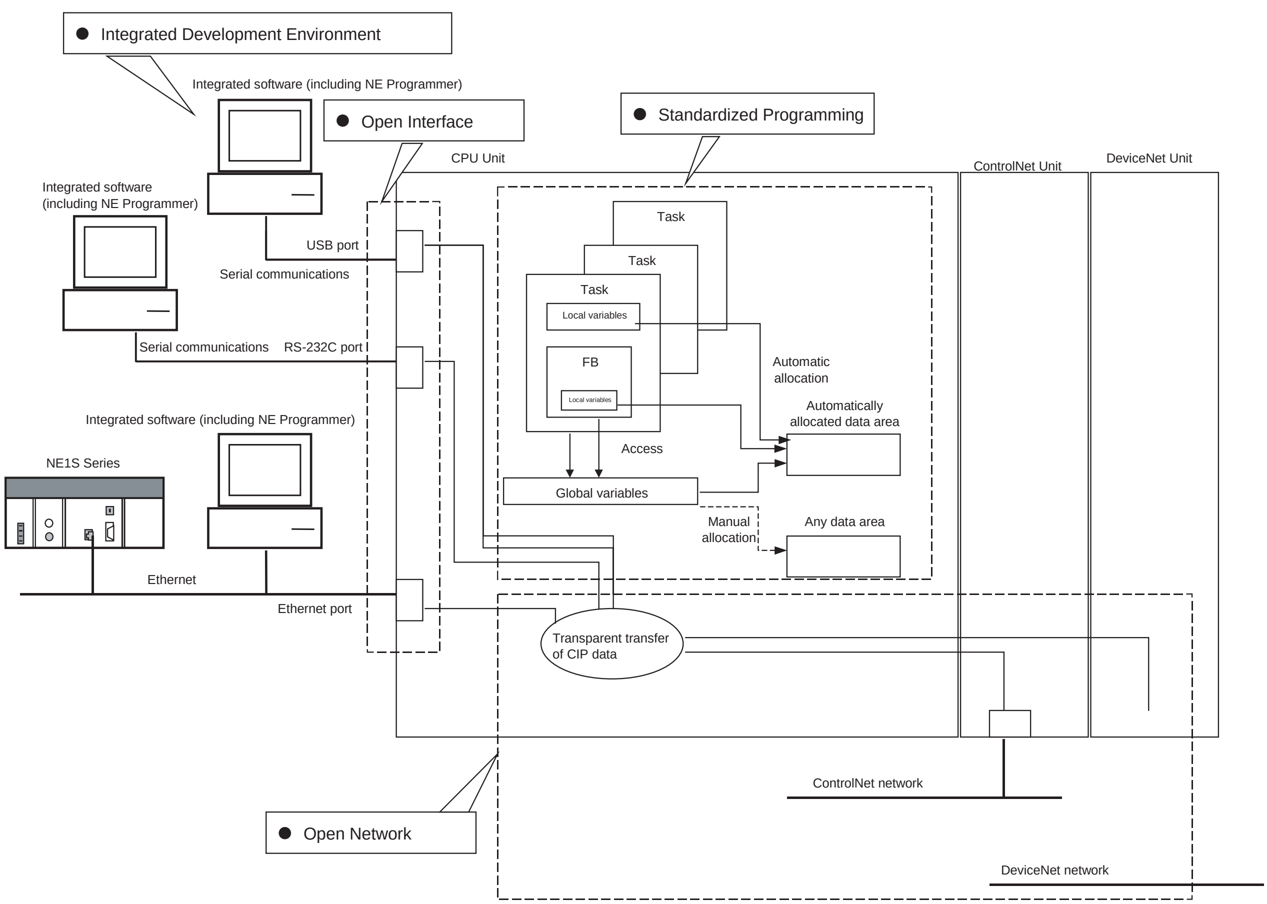
1-1 NE1S Series Features

1-1-1 Overview

The NE1S is a next-generation Programmable Controller and features the latest standardized programming, interface, and networking functions. The NE1S can be used to create and edit programs in a structured text language that conforms to the IEC 61131-3 standard.



Overview of System Configuration



- Program standardization:
  The software conforms to IEC 61131-3 standards, so it can support genuine variable programming and function blocks at the network level. Both ladder language and ST language are supported.

- Open network standardization:
  The CIP (Common Industrial Protocol) provides transparency across networks.
- Open interface standardization:
  An Ethernet port and USB port are built into the CPU Unit as a standard feature.
- Integrated development environment:
  The Programming Device and Network Configurator are integrated in a single software package.

1-1-2 Advantages

Standardized Programs

Internationally Standardized Programming

Programs can be created that conform to the IEC 61131-3 standard.

Programs and function blocks are supported as logical POUs (Program Organization Units). Either LD (ladder/mnemonic language) or ST (structured text) programming can be used in these logical POUs (programs or functions).

Software created in this way is reusable in other systems so it provides true software standardization.

Supports True Variable Programming

After variables are declared in the variable table, those declared variables can be used in the program, providing true variable programming just like other high-level languages. Programs can be designed in the same way with the three kinds of variables (local program-level variables, function block-level variables, and PLC-level variables).

By using variables, programs can be written before physical addresses are assigned, making it easier to reuse programs and make programs modular.

Supports Different Data Type Declarations for Each Variable

The data type can be declared for each variable. Using variables makes programs easier to read and can be reused without data inconsistencies.

True Network-level Programming with the Network Path Function and Variable Function

When another CPU Unit in the network is specified in an NE1S-series CPU Unit, the path to the other CPU Unit can be specified without setting routing tables. This network path function allows the address of another NE1S-series CPU Unit in the network to be specified as a variable with a network path.

The variable function in the NE1S-series CPU Unit that received the variable can determine the physical address from the variable name. These functions provide true variable programming on a network level.

Modularize Program Processes into User-defined Function Blocks

User-defined function blocks can be created. A series of complex processes can be combined in a single block, making it easy to reuse programming resources and experience. It is even possible to almost eliminate programming from the user standpoint.

Automatic Allocation of I/O Memory to Variables

Variables can be broadly divided into two categories: global variables that are shared within a PLC and local variables that are unique to a program or function block.

Physical memory addresses can be allocated to both global variables and local variables automatically with NE Programmer. The automatic allocation of I/O memory allows variables to be used in programming without dealing directly with the variables' addresses.

While it isn't necessary to deal with the variables' addresses, the user can manually specify the physical addresses of global variables if necessary.

Improve Reusability with Library Conversion

Program sections or function blocks that are used repeatedly can be registered as library functions and reused. Using library functions is much more efficient than copying program sections, so more efficient programs can be created.

Sensitive Program Sections Can Be Protected

Program tasks or function blocks can be read or write protected. This function can protect software resources from third parties and convert the processing to a "black box."

Open Interfaces

Standard Ethernet Port (100Base-TX)

All NE1S-series CPU Units are equipped with a built-in 100-Mbps Ethernet port that is TCP/IP compatible. The port uses standard Ethernet shielded, twisted-pair cables (Ethernet/IP specifications).

Standard USB Port and RS-232C Port

All NE1S-series CPU Units are equipped with a built-in USB Port and RS-232C port. A computer running the NE Programmer Programming Software can be connected to the USB port with a standard USB cable.

The USB port supports Programming Device connections (other than Programming Consoles) and Host Link. The RS-232C port supports Programming Device connections (other than Programming Consoles), Host Link, and 1:N NT Links.

NE Programmer Programming Software Connects to All Interfaces

The NE Programmer Programming Software can be connected through RS-232C, USB, ControlNet, or Ethernet. (NE Programmer uses CIP communications for all of the interfaces.)

Open Networks

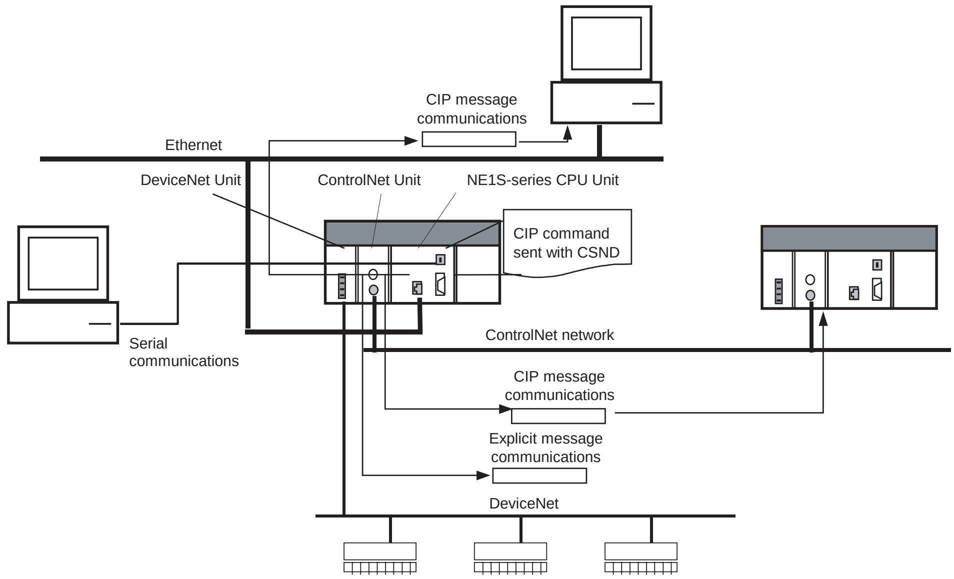
Share Ethernet, ControlNet, DeviceNet, and Serial Communications

Standard CIP (Control and Information Protocol) Communications

The industry's internationally standardized CIP protocol is used for both serial communications (the USB and RS-232C ports) and network communications (the ControlNet and Ethernet multivendor networks). The CIP communications allow a truly open communications system to be constructed.

Transfer CIP Commands through Message Communications

When necessary, any CIP command can be sent to a device connected through serial communications or an Ethernet, ControlNet, or DeviceNet network. Any CIP command, such as a data read/write command, can be sent to the device and the corresponding response can be received just by executing a CSND instruction in the NE1S-series CPU Unit's program (without using a connection).



Transparent Message Communications across Networks

CIP message communications can be sent and received transparently through the following networks: Ethernet (built-in Ethernet port), ControlNet (ControlNet Unit), DeviceNet (DeviceNet Unit), serial communications (built-in USB and RS-232C ports).

Note For DeviceNet, CIP messages can be sent from other networks to DeviceNet, but not from DeviceNet to other networks. For serial communications, CIP messages can be sent from a serial connection to Ethernet, ControlNet, or DeviceNet, but not from these networks to a serial connection.

Specify Paths for Network Routing without Routing Tables

When a target device in the network is being specified from an online NE Programmer/Configurator connection or message communications, the path to the target can be specified (instead of specifying the network address). With the source routing method, it is not necessary to create routing tables and transfer those tables to NE1S-series CPU Units in the network before operation.

Transparent Transfer of TCP/IP or UDP/IP Packets across Ethernet/ControlNet

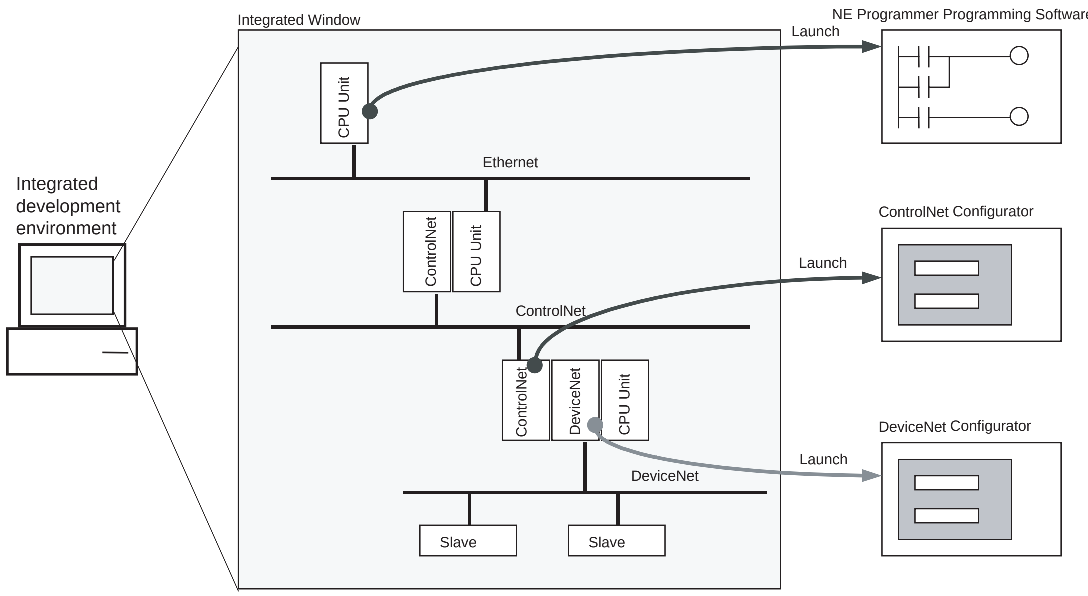
TCP/IP or UDP/IP packets can be sent and received transparently between the Ethernet port and ControlNet network (using the ControlNet Unit's IP over ControlNet service and Ethernet port's IP routing function).

With these functions, communications based on TCP/IP or UDP/IP (such as FTP) can be executed between the Ethernet network connected to the Ethernet port and the ControlNet network connected to the ControlNet Unit.

Integrated Development Environment

Integrated Development Environment

Each Programming Device can be started centrally from a window displaying the NE1S-series CPU Units connected through serial communications (USB and RS-232C), Ethernet, ControlNet, and DeviceNet.



Note It is not possible to display three different network levels simultaneously on in the actual window.

Remote Programming/Monitoring from NE Programmer (Serial Connection)

The networks share the common CIP communications protocol, so that another NE1S-series CPU Unit in an Ethernet, ControlNet, or DeviceNet network can be remotely programmed or monitored from NE Programmer Programming Software connected through a serial connection (USB or RS-232C).

Remote Programming/Monitoring from NE Programmer (Ethernet or ControlNet Connection)

A computer running the NE Programmer Programming Software can also be connected directly to an Ethernet or ControlNet network. In this case, another NE1S-series CPU Unit in an Ethernet, ControlNet, or DeviceNet network can be remotely programmed or monitored from NE Programmer Programming Software connected through ControlNet

Directly Inputting Mnemonics in a Ladder Window

Instructions with text mnemonics such as LD, AND, and MOV can be entered directly by moving the cursor to the desired insertion point in the Ladder Programming Window and entering the mnemonic. Inputs and outputs can also be entered by selecting the input or output icon from the toolbar and advanced instructions can be entered by dragging and dropping the instruction from the instruction list.

Ethernet Network (Built-in Ethernet Port)

Compatible with the Next-generation Industrial Network: EtherNet/IP Network

The Ethernet network is equipped with CIP (application layer) to support EtherNet/IP. This function allows CIP message communications (UCMM communications and Class 3 message communications) to be executed.

Data can be exchanged seamlessly between the IT system and Control system because CIP routing can be used and data can be transmitted transparently through ControlNet and serial communications (USB and RS-232C, see note 2).

Note

(1) EtherNet/IP is an abbreviation for Ethernet Industrial Protocol, which is the Ethernet-based industrial network published by ODVA in June 2001.

(2) Only from a serial connection to the network.

FTP Server Function in Ethernet

The FTP server function (file access from users) can be executed over the Ethernet network. The FTP server function allows individual files in the NE1S-series CPU Unit's Memory Card to be read/written from a host computer through the Ethernet network or Internet. Large amounts of data (such as programs) can be transferred easily between a computer and the NE1S-series CPU Unit.

SMTP Client Function in Ethernet

The SMTP client function (mail transfer) can be executed over the Ethernet network. The SMTP client function allows e-mail to be sent from the NE1S-series CPU Unit to e-mail addresses that are set in the system. User-defined messages, Unit error information, and status information can be sent to computers as e-mail through the Ethernet network or Internet. E-mail can be sent at periodic intervals or when a specified bit goes ON.

BOOTP Client Function in Ethernet

The BOOTP client function (a bootstrap protocol that finds the local MAC address and IP address) can be executed over the Ethernet network. With the BOOTP client function, it isn't necessary to make an IP address setting for the NE1S-series CPU Unit with a rotary switch or Programming Device.

SNTP Client Function in Ethernet

The SNTP client function can be executed over the Ethernet network. The SNTP client function can periodically obtain clock information from the NTP server and synchronize the clock.

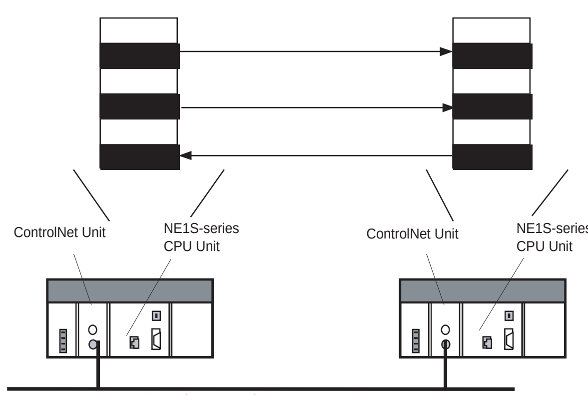
ControlNet Network (ControlNet Unit)

Data Sharing through Cyclic Communications (ControlNet)

The ControlNet Unit can exchange data cyclically with other devices in the ControlNet network. Programming is not required.

By establishing connections between input/output areas in the NE1S-series CPU Unit's data areas and other devices in the ControlNet network, data is exchanged automatically at the specified refreshing period. Up to 128 variables can be distributed in both the input and output areas of the CPU Unit's data area.

Parameters such as variables, target devices, and refreshing schedules must be specified in advance with the ControlNet Configurator.



Transfer Multiple Variable Groups through Cyclic Communications (ControlNet)

Multiple variables can be grouped and sent or received through a single connection. Grouping variables provides more flexibility and allows multiple data items to be transferred through cyclic communications without violating the limit of 126 connections.

Deterministic Scheduled Cyclic Communications

The ControlNet network uses the CTDMA (Concurrent Time Domain Multiple Access) algorithm to regulate communications so that each node transmits in order within its specified time slice. Transmitting devices repeat their turn at exact intervals. This method provides scheduled I/O data communications at exact intervals without a Master Unit. Reserving time slices for cyclic node transmission also prevents message communications from affecting cyclic communications.

Horizontal Distributed Networks

ControlNet cyclic communications do not use a fixed Master/Slave configuration, but a more flexible and efficient Producer/Consumer configuration. With this configuration, nodes in the network are distributed horizontally (on the same level) and data can be transmitted to all of the required devices when necessary (1:1, 1:N, and N:N communications are possible). For example, a producer node can send a multi-cast communication and the consumer nodes can receive the data only when it is needed.

DeviceNet Network (DeviceNet Unit)

Support for CIP Communications

NE1S-series DeviceNet Units are available to provide transparent communications in methods ranging from CIP communications to explicit messaging.

1-2 Specifications

1-2-1 NE1S-series CPU Unit Specifications

System Configuration

Item		Specifications	
Applicable Backplanes		• CPU Backplane: 2, 3, 5, 8, or 10 slots (CS1W-BC022/032/052/082/102) • Expansion Backplanes: 3, 5, 8, or 10 slots (CS1W-BI032/052/082/102)	
Number of Expansion Racks		7 max. (CS-series Long-distance Expansion Racks and C200H I/O Expansion Racks cannot be connected.)	
Applicable Power Units		Four models of AC Power Supply Unit (C200HW-PA204, C200HW-PA204S, C200HW-PA204R and C200HW-209R)	
Applicable Basic I/O Units		• Input Units: CS1W-ID211/231/261/291, -IA111/211, -INT01 • Output Units: CS1W-OD211/212/231/232/261/262/291/292, -OC201/211, -OA201/211 • Mixed I/O Units: CS1W-MD261/262/291/292/561 • High-speed Input Unit: CS1W-IDP01 • Safety Relay Unit: CS1W-SF200 **Note** C200H Basic I/O Units and C200H Group-two High-density I/O Units cannot be used.	
Applicable Special Units	CPU Bus Units	• ControlNet Unit: NE1S-CNS21U • Serial Communications Unit: CS1W-SCU21-V1 • DeviceNet Unit: NE1S -DRM21U • Controller Link Unit: CS1W-CLK21-V1 • Ethernet Unit: CS1W-ETN01/11/21 • FL-net Unit: CS1W-FLN22/02 • Data Storage/Processing Unit: CS1W-SPU01/02 • Motion Control Unit: CS1W-MCH71 • Position Control Unit: CS1W-NCF71	
	Special I/O Units	• Motion Control Units (2 axes or 4 axes): CS1W-MC421-V1/221-V1 • Analog I/O Unit: CS1W-AD041-V1/081-V1, CS1W-DA041/08C/08V, CS1W-MAD44, CS1W-AD161 • High-speed Counter Unit: CS1W-CT021/041 • ID Sensor Unit: CS1W-V600C11/12 • Position Control Unit: CS1W-NC113/133/213/233/413/433 • Isolated-type Thermocouple Input Units (Process Analog I/O Units): CS1W-PTS51/55	
Inner Boards		Cannot be used.	
CPU Unit built-in communications port	Serial communications port	USB port × 1 port	USB1.1, Protocol: Peripheral bus, SYSWAY, CIP, OS: Windows 2000 or XP
		RS-232C port × 1 port	Protocol: Peripheral bus, SYSWAY, 1:N NT Link, CIP
		Note A Programming Console cannot be connected to either port.	
	Ethernet communications port	Ethernet port	100Base-TX or 10Base-T
		Communications functions	CIP message communications (messages: UCMM/Class 3. I/O connections: Class 1 is not supported), FTP server, SMTP client, SNTP client, BOOTP client, IP routing (10 layers max.), IP over ControlNet
Interfaces to integrated development environments		COM (USB)	CIP communications via USB port or RS-232C port
		DeviceNet Interface Card	CIP communications via DeviceNet
		TCP/IP encapsulation	CIP communications via Ethernet
		RS Linx	Communications using RS Linx Rockwell ControlNet Communications Driver via Allen-Bradley Communications Card

Performance

Item		Specifications
Control method		Stored program
I/O control method		Cyclic scan and immediate processing are both possible.
Logic POU		Program and function blocks
Programming language		Program and function blocks: Ladder diagrams or structured text (ST)
CPU processing mode		Synchronous Mode
Instruction length		1 to 7 steps per instruction
Ladder instructions		Approx. 400 (3-digit function codes) Refer to *Appendix G Instruction Support and Operand/Variable Restrictions* for applicable instructions.
Instruction execution times	Basic instructions	0.019 µs min.(19 ns)
	Special instructions	0.06 µs min.
Program capacity		250 Ksteps (Including data automatically generated by NE Programmer.)
I/O capacity		5,120 points
Overhead processing time		0.3 ms
I/O table settings		First rack words and first slot words
Number of tasks		384 (cyclic tasks: 128, interrupt tasks: 256) **Note** Cyclic tasks are executed each cycle and are controlled with TKON and TKOF instructions. **Note** The following 4 types of interrupt tasks are supported. Power OFF interrupt tasks: 1 max. Scheduled interrupt tasks: 2 max. I/O interrupt tasks: 32 max. External interrupt tasks: 256 max. **Note** Indirect address of task numbers can be used in TKON and TKOF instructions.
Interrupt types		• Scheduled Interrupts: Interrupts generated at a time scheduled by the CPU Unit's built-in timer. • I/O Interrupts: Interrupts from Interrupt Input Units. • Power OFF Interrupts: Interrupts executed when the CPU Unit's power is interrupted. • External I/O Interrupts: Interrupts from the Special I/O Units and CPU Bus Units.

Item		Specifications
Variables	No. of registered variables	20,000 max. per project
	Variable types	Local variables in programs, local variables in function blocks, global variables
	Automatic allocations of variables in I/O memory	Automatic allocation in specific area of I/O memory **Note** Automatic allocation is not supported for timer numbers for TMHHX (ONE-MS TIMER).
	Data types	BOOL, WORD/DWORD, INT/UINT/DINT/UDINT, REAL (Rev. 2.1 or later), STRING, TIMER, COUNTER
	Initialization of variables	Supported. Local variables in function blocks (FBs): • Local variables in FBs can be set to default values at the start of operation.
	Array variables	Supported (number of elements: 1 to 255) No. of registered arrays: 20,000, No. of elements per array: 255, Total data volume of elements: 255 bytes
	Direct addresses (AT)	Supported for global variables; manual setting possible in the following areas. CIO (Core I/O Area), A (Auxiliary Area), D (Data Memory Area), E_0 (Extended Data Memory Area, Bank No. 0), H (Holding Area), W (Work Area) Direct addresses are not possible for the following areas. • Index registers and data registers (both direct and indirect specifications) • Indirect DM and EM addresses (for both @ (binary mode) and * (BCD mode) • EM bank No. 1 or higher
	Indirect addresses using variables	Not supported.
	Variable structures	Can be created by the user (each data structure can contain up to 255 bytes). STRING data types cannot be included as members. No. of variables in registered data structures: 4,000 (including structure members), Total data volume in data structures: 255 bytes max.
	Variable groups	Only function block I/O variables can be grouped. • 64 variables per group • Input variables: 256 groups max. • Output variables: 256 groups max.
	Network disclosure of variables	Variables to be disclosed on the network can be set for disclosure during cyclic communications for global variables only.
	Automatic allocation of variables	The maximum number of words for global variables (automatically allocated) and internal variables in tasks is as follows. Area held during power interruptions: 10,000 words max. Area not held during power interruptions: 32,768 words max. **Note** The areas listed above include empty regions that occur when data is allocated. Refer to *Function blocks (FB)* in the following row of this table for details on internal variables used in function blocks.

Item		Specifications
Function blocks (FB)	No. of FB instances	1,024
	No. of FB definitions	1,024
	Restrictions in FB definitions	• There must be no more than 32,767 steps of programming between JMP or CJP and JME. (Note: Each instruction requires from 1 to 7 steps.) • While EN is OFF, differentiated instructions and timer instructions are not initialized. (It is thus necessary to keep the input condition for EN ON and use an input condition inside the function block for differentiated instructions and timer instructions).
	FB nesting levels	8 levels max.
	FB IN and OUT variable restrictions	• IN (No. of VAR_INPUT input variables): 64 max., OUT (No. of VAR_OUTPUT output variables): 64 max. • Data types that can be used for IN and OUT variables (VAR_INPUT and VAR_OUTPUT): INT, DINT, UINT, UDINT, BOOL, WORD, DWORD, data structures, arrays, and STRING **Note** IN and OUT variables (VAR_INPUT and VAR_OUTPUT) can be grouped.
	FB I/O parameter restrictions	• Total data size of IN and OUT parameters per instance: 1,024 words max. • Addresses cannot be passed as parameters (only values can be passed). • Timer and counter variables cannot be used as parameters.
		• The following addresses can be input as parameters: CIO (Core I/O Area), A (Auxiliary Area), D (Data Memory Area), E_0 (Extended Data Memory Area, Bank No. 0), H (Holding Area), W (Work Area) **Note** The following cannot be specified as parameters. • Index registers and data registers (both direct and indirect specifications) • Indirect DM and EM addresses (for both @ (binary mode) and * (BCD mode) • EM bank No. 1 or higher **Note** Program local variables can be input for parameters as long as they are of the same data type as the FB local variables being passed from/to.
	Internal variable restrictions	There are no restrictions to the number of internal variables (VAR) that can be used. There are, however, restrictions to the following area sizes. Area size for internal variables held for power interrupts (regardless of data types of the variables): 65,536 words max. Area size for variables not held for power interruptions: 131,072 words max. **Note** The above sizes include open areas created by data configurations. Instances cannot be created after the specified sizes have been used for internal variables.
	No. of index registers (IRs) in instances	16 max.
Import/export functions for integrated development environment	Library files	Programs (including local variables), function blocks, and ladder diagram circuit parts (one circuit or multiple circuits, including local variables) can be saved as library files and reused. 1 program = 1 file, 1 function block = 1 file, 1 circuit part = 1 file
	Logic POU importing/ exporting	Ladder diagrams: Mnemonic text files can be imported. Structured text: Text data can be pasted.
	Variable table importing/ exporting	Importing: CSV files Exporting: CSV files, CSV files for OPC server

Data Structure in CPU Unit and Backup Methods

CPU Unit internal data	Data backup for simple backup operation	Automatic backup	
		To flash memory	**To Memory Card**
User program area (entire user program)	Yes (entire user program)	Yes	---
Program information	Yes (as program information file)	Yes	---
Variables (excluding variable comments)	Yes (as variable file)	Yes	---
Parameter area (PLC Setup, built-in Ethernet Setup, registered I/O tables, CIP service protection setting)	Yes (as parameter file)	Yes	---
I/O memory areas	Yes (as data file)	---	---
Special Unit data for simple backup operation	Yes (as Unit backup file)	---	---
Data file list	Yes (as data file)	---	---

Memory Card Data

Memory Card data	Data for simple backup operation	Automatic backup to Memory Card (for downloads or online editing)
Program revision history	Yes (as program revision history file)	Yes (as program revision history file)
Comments (variable comments, instruction comments, line comments, data structure member comments)	---	Yes (as comment file)
Structured text program	---	Yes (as structured text program file)

I/O Memory

As a rule, the physical addresses in the I/O memory described below can be used only for direct addresses for global variable address for NE-Programmer.

Note As an exception, physical addresses can be input directly for NE-Programmer.

<table>
<tr><th colspan="2">Item</th><th>Specification</th><th>Remarks</th></tr>
<tr><td rowspan="4">CIO (Core I/O) Area</td><td>I/O Area</td><td>5,120 bits (320 words): CIO 000000 to CIO 031915 (words CIO 0000 to CIO 0319)
The setting of the first word can be changed from the default (CIO 0000) so that CIO 0000 to CIO 0999 can be used.
I/O bits are allocated to Basic I/O Units.</td><td rowspan="4">The CIO Area can be used as work bits if the bits are not used as shown here.</td></tr>
<tr><td>CPU Bus Unit Area</td><td>6,400 bits (400 words): CIO 150000 to CIO 189915 (words CIO 1500 to CIO 1899)
CPU Bus Unit bits store the operating status of CPU Bus Units. (25 words per Unit, 16 Units max.)</td></tr>
<tr><td>Special I/O Unit Area</td><td>15,360 bits (960 words): CIO 200000 to CIO 295915 (words CIO 2000 to CIO 2959)
Special I/O Unit bits are allocated to Special I/O Units. (See Note.) (10 words per Unit, 96 Units max.)</td></tr>
<tr><td>DeviceNet Area</td><td>9,600 bits (600 words): CIO 320000 to CIO 379915 (words CIO 3200 to CIO 3799)
DeviceNet Area bits are allocated to Slaves according to fixed allocations for NE1S-DRM21U DeviceNet Unit remote I/O communications.
<table>
<tr><td>Fixed allocations 1</td><td>Outputs: CIO 3200 to CIO 3263
Inputs: CIO 3300 to CIO 3363</td></tr>
<tr><td>Fixed allocations 2</td><td>Outputs: CIO 3400 to CIO 3463
Inputs: CIO 3500 to CIO 3563</td></tr>
<tr><td>Fixed allocations 3</td><td>Outputs: CIO 3600 to CIO 3663
Inputs: CIO 3700 to CIO 3763</td></tr>
</table>
Note The following words are allocated to the NE1S-DRM21U.
DeviceNet Unit functioning as a slave when fixed allocations are used for the DeviceNet Unit.
<table>
<tr><th>Setting</th><th>Master to slave</th><th>Slave to master</th></tr>
<tr><td>Fixed allocations 1</td><td>Outputs: CIO 3370</td><td>Inputs: CIO 3270</td></tr>
<tr><td>Fixed allocations 2</td><td>Outputs: CIO 3570</td><td>Inputs: CIO 3470</td></tr>
<tr><td>Fixed allocations 3</td><td>Outputs: CIO 3770</td><td>Inputs: CIO 3670</td></tr>
</table></td></tr>
</table>

Item		Specification	Remarks
Internal I/O Area	Core I/O (CIO) Area	4,800 bits (300 words): CIO 120000 to CIO 149915 (words CIO 1200 to CIO 1499) 37,504 bits (2,344 words): CIO 380000 to CIO 614315 (words CIO 3800 to CIO 6143) These bits in the CIO Area are used as work bits in programming to control program execution. They cannot be used for external I/O.	The CIO Area can be used as work bits if the bits are not used as shown here.
	Work Area	8,192 bits (512 words): W00000 to W51115 (words W000 to W511) Used as work bits in programming to control program execution. (I/O from external I/O terminals is not possible.) **Note** When using work bits in programming, use the bits in the Work Area first before using bits from other areas.	
Holding Area		8,192 bits (512 words): H00000 to H51115 (H000 to H511) Holding bits are used to control the execution of the program, and maintain their ON/OFF status when the PLC is turned OFF or the operating mode is changed.	---
Auxiliary Area		Read only: 7,168 bits (448 words): A00000 to A44715 (words A000 to A447) Read/write: 8,192 bits (512 words): A44800 to A95915 (words A448 to A959) Auxiliary bits have specific functions.	---
Temporary Area		16 bits (TR0 to TR15) Temporary bits are used to temporarily store the ON/OFF execution conditions at program branches.	---
Timer Area		4,096 timers: T0000 to T4095 (used for timers only)	---
Counter Area		4,096 counters: C0000 to C4095 (used for counters only)	---
DM Area		32K words: D00000 to D32767 Internal Special I/O Unit DM Area: D20000 to D29599 (100 words x 96 Units); used to set parameters for Special I/O Units. CPU Bus Unit DM Area: D30000 to D31599 (100 words x 16 Units); used to set parameters for CPU Bus Units. Used as a general-purpose data area for reading and writing data in word units (16 bits). Words in the DM Area maintain their status when the PLC is turned OFF or the operating mode is changed. **Note** Indirect DM addresses (for both @ (binary mode) and * (BCD mode)) cannot be used as AT designations for variables. (Physical addresses can be input directly.)	---
EM Area		32K words per bank, (bank No. 0 only): E0_00000 to E0_32767 max. Used as a general-purpose data area for reading and writing data in word units (16 bits). Words in the EM Area maintain their status when the PLC is turned OFF or the operating mode is changed. **Note** The EM Area cannot be converted to file memory. **Note** Indirect EM addresses (for both @ (binary mode) and * (BCD mode)) cannot be used as AT designations for variables. (Can be input directly as physical addresses.)	
Index Registers		Can be used only by direct access though global variables. (Can be input directly as physical addresses.) IR0 to IR15 Store I/O memory addresses for indirect addressing. One register is 32 bits (2 words). Index registers can be used either independently in each task or to share them between tasks.	
Task Flag Area		128 (TK0000 to TK0127) Task Flags are read-only flags that are ON when the corresponding cyclic task is executable and OFF when the corresponding task is not executable or in standby status.	

Function Specifications

Item	Specifications
Trace Memory	4,000 words (trace data: 31 bits, 6 words))
File Memory	Memory Cards: Memory Cards manufactured by OMRON are used (MS-DOS format). EM file memory: EM Area cannot be converted to file memory.
Constant cycle time	Supported. (1 to 32,000 ms, unit: 1 ms) When a parallel processing mode is used, the cycle time for executing instructions can be set to a minimum time.
Maximum cycle time	Supported. (Unit stops operating if the cycle is too long): 10 to 40,000 ms (Unit: 10 ms)
I/O refreshing	Cyclic refreshing, immediate refreshing, refreshing by IORF(097). IORF(097) refreshes I/O bits allocated to Basic I/O Units and Special I/O Units. With the NE1S CPU Units, the CPU BUS UNIT I/O REFRESH (DLNK(226)) instruction can be used to refresh bits allocated to CPU Bus Units in the CIO and DM Areas.

<table>
<tr><th colspan="2">Item</th><th colspan="2">Specifications</th></tr>
<tr><td colspan="2">Timing of special refreshing for CPU Bus Units</td><td colspan="2">ControlNet cyclic communications, remote I/O for DeviceNet Units, and other special refreshing for CPU Bus Units is performed at the following times:
I/O refresh period and when the CPU BUS UNIT I/O REFRESH (DLNK(226)) instruction is executed.</td></tr>
<tr><td colspan="2">I/O memory holding when changing operating modes</td><td colspan="2">Supported. (Depends on the ON/OFF status of the IOM Hold Bit in the Auxiliary Area.)
I/O memory, however, cannot be held for automatically allocated variables when the operating mode is changed.</td></tr>
<tr><td colspan="2">Load OFF function</td><td colspan="2">All outputs on Output Units can be turned OFF when the CPU Unit is operating in RUN, MONITOR, or PROGRAM mode.</td></tr>
<tr><td colspan="2">Online editing</td><td colspan="2">Supported, but the following restrictions apply:
• There restrictions in the number of times and the applicable areas for adding and deleting variables.
• Local variables can be added only for automatic allocation of variables.
• FB instances cannot be added, deleted, or changed.
• I/O variables (VAR_INPUT and VAR_OUTPUT) cannot be added, deleted, or changed.
• Global variables cannot be deleted or changed.
• Existing variables cannot be deleted or changed (including all properties and names).
• Data types cannot be added, deleted, or changed (including increasing, decreasing, and deleting members from data structures, changing array elements, etc.)
• FB I/O groups cannot be added, deleted, or changed.
Note When a FB is edited, the entire FB definition is transferred regardless of the size of the edit.
Note Power to the PLC cannot be turned OFF during backup operations.
Note All building operation must be completed before online editing can be started.</td></tr>
<tr><td colspan="2">Timer/Counter PV refresh mode</td><td colspan="2">Binary mode only.
Note BCD mode cannot be used.</td></tr>
<tr><td colspan="2">Input response time setting</td><td colspan="2">Time constants can be set for inputs from Basic I/O Units. The time constant can be increased to reduce the influence of noise and chattering or it can be decreased to detect shorter pulses on the inputs.</td></tr>
<tr><td colspan="2">Startup mode setting</td><td colspan="2">Supported.
PROGRAM mode is the default setting.</td></tr>
<tr><td colspan="2">Flash memory</td><td colspan="2">The user program, program information, variable information, and parameter area data (e.g., PLC Setup and setting for built-in Ethernet) are always backed up and restored automatically using flash memory.</td></tr>
<tr><td rowspan="4">Memory Card functions</td><td>Memory Card read/write methods</td><td colspan="2">Special instructions in user program, simple backup operation, or FTP transfers using Ethernet.
Note Program replacement during operation is not supported.</td></tr>
<tr><td>Data saved in Memory Cards</td><td>Data specified in user operations (Special instructions in user program or simple backup operation)</td><td>For special instructions in user program:
Data files (extension: .IOM, .TXT, or .CSV)</td></tr>
<tr><td rowspan="2">Data saved in Memory Cards</td><td>User operation</td><td>Simple backup operation:
• Program file
• Parameter file
• Data file
• Unit backup file
• Variable file
• Program history file</td></tr>
<tr><td>Automatic saving by system</td><td>Following files are automatically saved when the program is changed (downloading or online editing):
• Comments (variable comments, instruction comments, line comments, data structure member comments)
• Source code for structured text program
• Program history file</td></tr>
</table>

Item		Specifications
Program protection		Overwrite protection: Set using DIP switch. Read (copy) protection: Password set using Programming Device. Reading function block definitions can be prohibited.
Fault diagnosis functions		User-defined errors (i.e., user can define fatal errors and non-fatal errors) FAL and FALS instructions can be used to simulate errors.
Error log		Up to 20 errors are stored in the error log. Information includes the error code, error details, and the time the error occurred. User-defined FAL errors can be set to be stored or not to be stored in the error log.
Clock		Provided on all models. Accuracy: ±1.5 s/mo at 25°C (accuracy varies with the temperature) **Note** Used to store the time when power is turned ON and when errors occur.
Power OFF detection time		10 to 25 ms (not fixed)
Power OFF detection delay time		0 to 10 ms (user-defined, default: 0 ms)
Memory protection for power interruptions		Held Areas: Holding bits, contents of Data Memory and Extended Data Memory, and status of the counter Completion Flags and present values. **Note** If the IOM Hold Bit in the Auxiliary Area is turned ON and the PLC Setup is set to maintain the IOM Hold Bit status when power to the PLC is turned ON, the contents of the CIO Area, the Work Area, part of the Auxiliary Area, timer Completion Flags, timer PVs, Index Registers, and Data Registers will be held.
Sending unsolicited commands to a computer connected via CIP		Not supported
Remote programming and monitoring		CIP communications can be used for remote programming and remote monitoring through ControlNet and Ethernet networks.
Access across networks	CIP messages	CIP messages can be routed between serial communications (USB and RS-232C), Ethernet, and ControlNet networks. (There is no limit in the protocol to the number of networks through which access is possible.)
	IP routing	Up to 10 gateway addresses can be specified, including Ethernet and ControlNet, in addition to the default gateway addresses.
Program check		Program checks are performed at the beginning of operation for items such as no END instruction and instruction errors. User program checks are also possible.
Control output signals		RUN output: The internal contacts will turn ON (close) while the CPU Unit is operating.
Battery life		CJ1W-BAT01 Battery Set: 5 years max. (Depends on operating environment.)
Self-diagnostics		CPU errors (watchdog timer), I/O verification errors, I/O bus errors, memory errors, and battery errors.
Other functions		Storage of number of times power has been interrupted. (Stored in A514.)

General Specifications

Item	Specification	
Power Supply Unit	C200HW-PA204/204S/204R	C200HW-PA209R
Supply voltage	100 to 120 V AC or 200 to 240 V AC, 50/60 Hz	
Operating voltage range	85 to 132 V AC or 170 to 264 V AC	
Power consumption	120 VA max.	180 VA max.
Inrush current	30A max.	30A max. for 100 to 120 V AC 40A max. for 200 to 240 V AC
Output capacity	4.6 A, 5 V DC (including the CPU Unit power supply)	9 A, 5 V DC (including the CPU Unit power supply)
	26 V DC, 0.625 A 24 V DC, 0.8 A	26 V DC, 1.3 A
	Total: 30 W max.	Total: 45 W max.

Item	Specification	
Output terminal (service supply)	Provided only on the PA204S. (See note 4.) 0.3 A or less current consumption by 24-V DC load: 24 V DC is +17% to –11% 0.3 A current consumption by 24 V DC load: 24 V DC is +10% to –11% (lot No. 0197 or later)	None
RUN output (See note 2.)	Provided only on the PA204S. (See note 5.) Contact form: SPST-NO Switching capacity: 250 V AC, 2 A (resistive load) 250 V AC, 0.5 A (induction load: 24 V DC, 2 A)	Contact configuration: SPST-NO Switch capacity: 240 V AC, 2A (resistive load), 120 V AC, 0.5 A (induction load) 24 V DC, 2A (resistive load) 24 V DC, 2 A (induction load)
Insulation resistance	20 MW min. (at 500 V DC) between AC external and GR terminals (See note 1.)	
Dielectric strength	2,300 V AC 50/60 Hz for 1 min between AC external and GR terminals (See note 1.) Leakage current: 10 mA max.	
	1,000 V AC 50/60 Hz for 1 min between DC external and GR terminals (See note 1.) Leakage current: 10 mA max.	
Noise immunity	2 kV on power supply line (conforming to IEC61000-4-4)	
Vibration resistance	10 to 57 Hz, 0.075-mm amplitude, 57 to 150 Hz, acceleration: 9.8 m/s^2 in X, Y, and Z directions for 80 minutes (Time coefficient: 8 minutes × coefficient factor 10 = total time 80 min.) CPU Unit mounted to a DIN track: 2 to 55 Hz, 2.94 m/s^2 in X, Y, and Z directions for 20 minutes	
Shock resistance	147 m/s^2 3 times each in X, Y, and Z directions (according to JIS C0041)	
Ambient operating temperature	0 to 55°C	
Ambient operating humidity	10% to 90% (with no condensation)	
Atmosphere	Must be free from corrosive gases.	
Ambient storage temperature	–20 to 75°C (excluding battery)	
Grounding	Less than 100 Ω	
Enclosure	Mounted in a panel.	
Weight	Refer to *Section 3*.	
CPU Rack dimensions (mm)	2-slot Rack: 198.5 ×157×123 mm (W×H×D) (See note 3.) 3-slot Rack: 260 × 130 × 123 mm (W×H×D) (See note 3.) 5-slot Rack: 330 × 130 × 123 mm (W×H×D) (See note 3.) 8-slot Rack: 435 × 130 × 123 mm (W×H×D) (See note 3.) 10-slot Rack: 505 × 130 × 123 mm (W×H×D) (See note 3.)	
Safety standards	Conforms to cULus and EC Directives.	

Note

(1) Disconnect the Power Supply Unit's LG terminal from the GR terminal when testing insulation and dielectric strength. Testing the insulation and dielectric strength with the LG terminal and the GR terminals connected will damage internal circuits in the CPU Unit.

(2) Supported only when mounted to CPU Backplane.

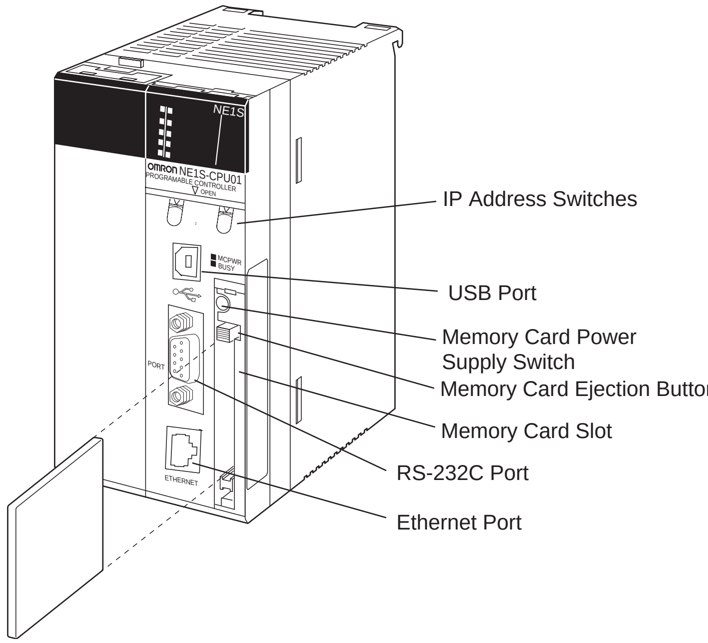
(3) The depth is 153 mm for the C200HW-PA209R Power Supply Unit.

(4) The C200HW-PA204 and C200HW-PA204R do not have an output terminal.

(5) The C200HW-PA204 and C200HW-PA204S do not have a RUN output.

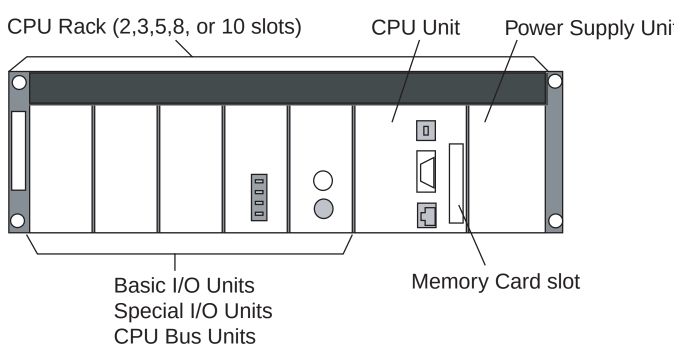
1-3 System Configuration

1-3-1 CPU Unit Components



1-3-2 System Configuration

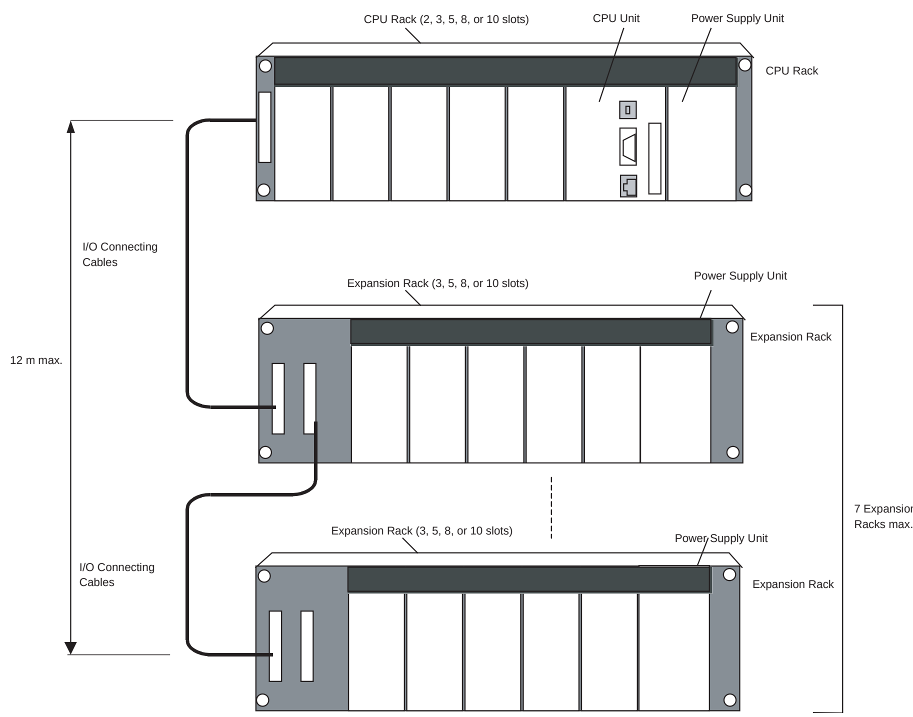
CPU Rack



Name	Model	Specifications
CPU Unit	NE1S-CPU01	I/O bits: 5,120, Program capacity: 250K steps Data Memory: 64K words (DM: 32K words, EM: 32K words)
CPU Backplanes	CS1W-BC022	2 slots
	CS1W-BC032	3 slots
	CS1W-BC052	5 slots
	CS1W-BC082	8 slots
	CS1W-BC102	10 slots

Name	Model	Specifications
Power Supply Units	C200HW-PA204	100 to 120 V AC or 200 to 240 V AC
	C200HW-PA204S	100 to 120 V AC or 200 to 240 V AC (with 0.8 A 24 V DC service power supply) Output capacity: 4.6 A, 5 V DC
	C200HW-PA204R	100 to 120 V AC or 200 to 240 V AC (with RUN output) Output capacity: 4.6 A, 5 V DC
	C200HW-PA209R	100 to 120 V AC or 200 to 240 V AC (with RUN output) Output capacity: 9 A, 5 V DC
Memory Cards	HMC-EF172	Flash memory, 15 MB
	HMC-EF372	Flash memory, 30 MB
	HMC-EF671	Flash memory, 64 MB
	HMC-AP001	Memory Card Adapter
Battery Set (Built in the CPU Unit)	CJ1W-BAT01	5 years max. (depending on the environment)

Expansion Racks



Name	Model	Specifications	Cable Length
Expansion Back-planes	CS1W-BI032	3 slots	---
	CS1W-BI052	5 slots	---
	CS1W-BI082	8 slots	---
	CS1W-BI102	10 slots	---

Name	Model	Specifications	Cable Length
Power Supply Units	C200HW-PA204	100 to 120 V AC or 200 to 240 V AC Output capacity: 4.6 A, 5 V DC	---
	C200HW-PA204S	100 to 120 V AC or 200 to 240 V AC (with service supply: 0.8 A, 24 V DC) Output capacity: 4.6 A, 5 V DC	---
	C200HW-PA204R	100 to 120 V AC or 200 to 240 V AC (with RUN output) Output capacity: 4.6 A, 5 V DC	---
	C200HW-PA209R	100 to 120 V AC or 200 to 240 V AC (with RUN output) Output capacity: 9 A, 5 V DC	---
I/O Connecting Cables	CS1W-CN313	Connect Expansion Backplanes to CPU Backplane or other Expansion Backplane.	0.3 m
	CS1W-CN713		0.7 m
	CS1W-CN223		2 m
	CS1W-CN323		3 m
	CS1W-CN523		5 m
	CS1W-CN133		10 m
	CS1W-CN133B2		12 m

Connectable Units

The following table shows the Units that can be connected to CPU Rack and Expansion Racks.

Unit	CS-series Basic I/O Units (See note.)	C200H Basic I/O Units	C200H Group-2 High-density I/O Units (Basic I/O Units)	CS-series Special I/O Units	C200H Special I/O Units	CPU Bus Units
CPU Rack	Yes	No	No	Yes	No	Yes
Expansion Racks	Yes	No	No	Yes	No	Yes

Note CS1W-INT01 Interrupt Input Units must be mounted to the CPU Rack.

Maximum Number of Units

The maximum number of expansion slots is 80, so the maximum number of Units that can be mounted is 80. There are no restrictions in the number of each type of Unit in terms of mounting location.

Note (1) A maximum of two CS1W-INT01 Interrupt Input Units can be mounted and these must be mounted to the CPU Rack.

(2) Up to 16 CPU Bus Units can be mounted.

1-3-3 Units

Basic I/O Units

Basic Input Units

Name	Specifications	Model	Number of bits allocated (CIO 0000 to CIO 0319)	Mountable Racks	
				CPU Rack	Expansion Racks
AC Input Units	100 to 120 V DC, 100 to 120 V AC, 16 inputs	CS1W-IA111	16	Yes	Yes
	200 to 240 V DC, 16 inputs	CS1W-IA211	16	Yes	Yes
DC Input Units	24 V DC, 16 inputs	CS1W-ID211	16	Yes	Yes
Interrupt Input Units	24 V DC, 16 inputs	CS1W-INT01	16	Yes	Yes (See note.)
High-speed Input Units	24 V DC, 16 inputs	CS1W-IDP01	16	Yes	Yes (See note.)

Name	Specifications	Model	Number of bits allocated (CIO 0000 to CIO 0319)	Mountable Racks	
				CPU Rack	Expansion Racks
DC Input Units	24 V DC, 32 inputs	CS1W-ID231	32	Yes	Yes
	24 V DC, 64 inputs	CS1W-ID261	64	Yes	Yes
	24 V DC, 96 inputs	CS1W-ID291	96	Yes	Yes

Note The interrupt function can be used only when the Unit is mounted to the CPU Rack.

Basic Output Units

Name	Specifications		Model	Number of bits allocated (CIO 0000 to CIO 0319)	Mountable Racks	
					CPU Rack	Expansion Racks
Relay Output Units	250 V AC/ 24 V DC, 2 A; 120 V DC, 0.1A; 8 outputs, independent contacts		CS1W-OC201	16	Yes	Yes
	250 V AC/24 V DC, 2 A; 120 V DC, 0.1 A; 16 outputs		CS1W-OC211	16	Yes	Yes
Triac Output Units	250 V AC, 1.2 A, 8 outputs, with fuse burn-out detection circuit		CS1W-OA201	16	Yes	Yes
	250 V AC, 0.5 A, 16 outputs		CS1W-OA211	16	Yes	Yes
Transistor Output Units	Sinking outputs	12 to 24 V DC, 0.5 A, 16 outputs	CS1W-OD211	16	Yes	Yes
		12 to 24 V DC, 0.5 A, 32 outputs	CS1W-OD231	32	Yes	Yes
		12 to 24 V DC, 0.3 A, 64 outputs	CS1W-OD261	64	Yes	Yes
		12 to 24 V DC, 0.1 A, 96 outputs, with fuse burnout detection circuit	CS1W-OD291	96	Yes	Yes
	Sourcing outputs	24 V DC, 0.5 A, 16 outputs	CS1W-OD212	16	Yes	Yes
		24 V DC, 0.3 A, 32 outputs, load short-circuit protection	CS1W-OD232	32	Yes	Yes
		24 V DC, 0.3 A, 64 outputs, load short-circuit protection	CS1W-OD262	64	Yes	Yes
		24 V DC, 0.1 A, 96 outputs, with fuse burnout detection circuit	CS1W-OD292	96	Yes	Yes

Mixed I/O Units

Name	Specifications		Model	Number of bits allocated (CIO 0000 to CIO 0319)	Mountable Racks	
					CPU Rack	Expansion Racks
DC Input/ Transistor Output Units	Sinking outputs	24 V DC, 32 inputs 12 to 24 V DC, 0.3 A, 32 outputs	CS1W-MD261	32 input bits 32 output bits	Yes	Yes
		24 V DC, 48 inputs 12 to 24 V DC, 0.1 A, 48 outputs, with fuse burnout detection circuit	CS1W-MD291	48 input bits 48 output bits	Yes	Yes
	Sourcing outputs	24 V DC, 32 inputs 24 V DC, 0.3 A, 32 outputs, load short-circuit protection	CS1W-MD262	32 input bits 32 output bits	Yes	Yes
		24 V DC, 48 inputs 12 to 24 V DC, 0.1 A, 48 outputs, with fuse burnout detection circuit	CS1W-MD292	48 input bits 48 output bits	Yes	Yes
TTL I/O Unit		5 V DC, 3.5 mA, 32 inputs 5 V DC, 35 mA, 32 outputs	CS1W-MD561	32 input bits 32 output bits	Yes	Yes

Special I/O Units

Name	Specifications	Model	Number of words allocated (CIO 2000 to CIO 2959)	Number of words allocated (D20000 to D29599)	Mountable Racks		Unit No.	Reference
					CPU Rack	Expansion Racks		
Analog I/O Unit	4 inputs (4 to 20 mA,1 to 5 V, etc.) 4 outputs (1 to 5 V,0 to 10 V, etc.)	CS1W-MAD44	10 words	100 words	Yes	Yes	0 to 95	W345
Analog Input Units	4 inputs (4 to 20 mA, 1 to 5 V, etc.)	CS1W-AD041-V1	10 words	100 words	Yes	Yes	0 to 95	
	8 inputs (4 to 20 mA, 1 to 5 V, etc.)	CS1W-AD081-V1	10 words	100 words	Yes	Yes	0 to 95	
Analog Output Units	4 outputs (1 to 5 V, 4 to 20 mA, etc.)	CS1W-DA041	10 words	100 words	Yes	Yes	0 to 95	
	8 outputs (1 to 5 V, etc.)	CS1W-DA08V	10 words	100 words	Yes	Yes	0 to 95	
	8 outputs (4 to 20 mA)	CS1W-DA08C	10 words	100 words	Yes	Yes	0 to 95	
Motion Control Units	Two axes, analog output for each axis, supports G-language	CS1W-MC221	30 words	None	Yes	Yes	0 to 95 (Uses three unit numbers.)	W359
	Four axes, analog output for each axis, supports G-language	CS1W-MC421-R1	50 words	None	Yes	Yes	0 to 95 (Uses five unit numbers.)	

Name	Specifications	Model	Number of words allocated (CIO 2000 to CIO 2959)	Number of words allocated (D20000 to D29599)	Mountable Racks		Unit No.	Reference
					CPU Rack	Expansion Racks		
High-speed Counter Units	2-axis pulse inputs, counting rate: 500 Kcps max., supports line driver inputs	CJ1W-CT021	40 words	400 words	Yes	Yes	0 to 92 (Uses four unit numbers.)	W401
	4-axis pulse inputs, counting rate: 500 Kcps max., supports line driver inputs	CJ1W-CT041	40 words	400 words	Yes	Yes	0 to 92 (Uses four unit numbers.)	
ID Sensor Units	V600-series RFID System, one head	CS1W-W600C11	10 words	100 words	Yes	Yes	0 to 95	Z174
	V600-series RFID System, two heads	CS1W-V600C12	20 words	200 words	Yes	Yes	0 to 94 (Uses two unit numbers.)	
Position Control Units	1-axis pulse output, 1 to 500 kpps, open collector outputs	CS1W-NC113	10 words	100 words	Yes	Yes	0 to 95	W376
	1-axis pulse output, 1 to 500 kpps, line driver inputs	CS1W-NC133	10 words	100 words	Yes	Yes	0 to 95	
	2-axis pulse output, 1 to 500 kpps, open collector outputs	CS1W-NC213	10 words	100 words	Yes	Yes	0 to 95	
	2-axis pulse output, 1 to 500 kpps, line driver inputs	CS1W-NC233	10 words	100 words	Yes	Yes	0 to 95	
	4-axis pulse output, 1 to 500 kpps, open collector outputs	CS1W-NC413	20 words	200 words	Yes	Yes	0 to 95 (Uses two unit numbers.)	
	4-axis pulse output, 1 to 500 kpps, line driver inputs	CS1W-NC433	20 words	200 words	Yes	Yes	0 to 95 (Uses two unit numbers.)	
Isolated-type Thermocouple Input Units (Process Analog I/O Units)	4 independent thermocouple inputs (R, S, K, J, T, L, or B)	CS1W-PTS51	10 words	100 words	Yes	Yes	0 to 95	W368
	8 independent thermocouple inputs (R, S, K, J, T, L, or B)	CS1W-PTS55	10 words	100 words	Yes	Yes	0 to 95	

CPU Bus Units

Name	Specifications	Model	Number of words allocated CIO 1500 to CIO 1899	Number of words allocated D30000 to D31599	CPU Bus Unit System Settings	Mountable Racks		Unit No.	Reference
						CPU Rack	Expansion Racks		
ControlNet Unit	ControlNet ports A and B	NE1S-CNS21U	25 words	Not used	Not used	Yes	Yes	0 to F	Z902
DeviceNet Unit	DeviceNet remote I/O 2,048 points, also can operate as slave, user settings possible without Configurator	NE1S-DRM21U	25 words	100 words	Not used	Yes	Yes	0 to F	Z903

Name	Specifications	Model	Number of words allocated CIO 1500 to CIO 1899	Number of words allocated D30000 to D31599	CPU Bus Unit System Settings	Mountable Racks		Unit No.	Reference
						CPU Rack	Expansion Racks		
Serial Communications Unit	Two RS-232C ports	CS1W-SCU21-V1	25 words	20 words	Not used	Yes	Yes	0 to F	W336
High-resolution Motion Control Unit	MECHATROLINK II, 30 real axes, 2 virtual axes, special motion control language	CS1W-MCH71	25 words	100 words	Not used	Yes	Yes	0 to F	
MECHATROLINK-II-compatible Position Control Unit	MECHATROLINK-II16 axes max.	CS1W-NCF71	25 words	None	None	Yes	Yes	0 to F	W426
Controller Link Unit	Twisted-pair cable (wire type)	CS1W-CLK21-V1	25 words	100 words	Used (Data link table)	Yes	Yes	0 to F (4 Units max.)	W370
Ethernet Units	10Base-5	CS1W-ETN01	25 words	100 words	Used	Yes	Yes	0 to F (4 Units max.)	W343
	10Base-T	CS1W-ETN11	25 words	100 words	Used	Yes	Yes	0 to F (4 Units max.)	
	100Base-TX (Can be used as 10Base-T.)	CS1W-ETN21	25 words	100 words	Used	Yes	Yes	0 to F (4 Units max.)	W420 and W421
FL-net Units	FL-net (OPCN-2) Ver. 2 specifications, 100Base-TX	CS1W-FLN22	25 words	100 words	Used	Yes	Yes	0 to F (4 Units max.)	W440
	FL-net (OPCN-2) Ver. 2 specifications, 10Base-5	CS1W-FLN02	25 words	100 words	Used	Yes	Yes	0 to F (4 Units max.)	
Data Storage/Processing Units	Type II PC Card × 1 slot, Ethernet 10/100Base-TX × 1 port	CS1W-SPU01	Not used	Not used	Not used	Yes	Yes	0 to F	V229
	Type II PC Card × 1 slot, Ethernet 10/100Base-TX × 2 ports	CS1W-SPU02	Not used	Not used	Not used	Yes	Yes	0 to F	

Note A Programming Device can be used to specify the leading DM word allocated to a CS1W-EIP21, or set the DM words as unused.)

1-3-4 Unit Current Consumption

The amount of current/power that can be supplied to the Units mounted in a Rack is limited by the capacity of the Rack's Power Supply Unit. Refer to the following tables when designing your system so that the total current consumption of the mounted Units does not exceed the maximum current for each voltage group and the total power consumption does not exceed the maximum for the Power Supply Unit.

CPU Racks and Expansion Racks

The following table shows the maximum current and power that can be supplied by a Power Supply Unit in a CPU Rack or Expansion Rack.

When calculating current/power consumption in a CPU Rack, be sure to include the power required by the CPU Backplane and the CPU Unit. Likewise, be sure to include the power required by the Expansion Backplane when calculating current/power consumption in an Expansion Rack.

Power Supply Unit	Max. current consumption			Max. total power consumption
	5-V group (internal logic)	26-V group (relays)	24-V group (service)	
C200HW-PA204	4.6 A	0.625 A	None	30 W
C200HW-PA204S	4.6 A	0.625 A	0.8 A	30 W
C200HW-PA204R	4.6 A	0.625 A	None	30 W
C200HW-PA209R	9 A	1.3 A	None	45 W

Current Consumption Tables

5-V DC Voltage Group

Name	Model	Current consumption (A)
CPU Units	NE1S-CPU01	0.98
CPU Backplanes	CS1W-BC022	0.11
	CS1W-BC032	0.11
	CS1W-BC052	0.11
	CS1W-BC082	0.11
	CS1W-BC102	0.11
Expansion Backplanes	CS1W-BI032	0.23
	CS1W-BI052	0.23
	CS1W-BI082	0.23
	CS1W-BI102	0.23

■ **Basic I/O Units**

Category	Name	Model	Current consumption (A)
Basic Input Units	DC Input Units	CS1W-ID211	0.10
		CS1W-ID231	0.15
		CS1W-ID261	0.15
		CS1W-ID291	0.20
	AC Input Units	CS1W-IA111	0.11
		CS1W-IA211	0.11
	Interrupt Input Unit	CS1W-INT01	0.10
	High-speed Input Unit	CS1W-IDP01	0.10
Basic Output Units	Relay Output Units	CS1W-OC201	0.10
		CS1W-OC211	0.13
	Transistor Output Units	CS1W-OD211	0.17
		CS1W-OD212	0.17
		CS1W-OD231	0.27
		CS1W-OD232	0.27
		CS1W-OD261	0.39
		CS1W-OD262	0.39
		CS1W-OD291	0.48
		CS1W-OD292	0.48
	Triac Output Units	CS1W-OA201	0.23
		CS1W-OA211	0.41

Category	Name	Model	Current consumption (A)
Mixed I/O Units	DC Input/Transistor Output Units	CS1W-MD261	0.27
		CS1W-MD262	0.27
		CS1W-MD291	0.35
		CS1W-MD292	0.35
	TTL I/O Units	CS1W-MD561	0.27

■ Special I/O Units

Name	Model	Current consumption (A)
Analog I/O Unit	CS1W-MAD44	0.20
Analog Input Units	CS1W-AD041-V1	0.13
	CS1W-AD081-V1	
Analog Output Units	CS1W-DA041	0.13
	CS1W-DA08V	
	CS1W-DA08C	
Motion Control Units	CS1W-MC221	0.60 (0.80 with Teaching Box connected)
	CS1W-MC421-R1	0.70 (1.00 with Teaching Box connected)
High-speed Counter Units	CS1W-CT021	0.36
	CS1W-CT041	0.45
ID Sensor Units	CS1W-V600C11	0.26
	CS1W-V600C12	0.32
Position Control Units	CS1W-NC113	0.25
	CS1W-NC133	0.25
	CS1W-NC213	0.25
	CS1W-NC233	0.25
	CS1W-NC413	0.36
	CS1W-NC433	0.36
Isolated-type Thermocouple Input Units (Process Analog I/O Units)	CS1W-PTS51	0.25
	CS1W-PTS55	0.18

CPU Bus Units

Name	Model	Current consumption (A)
ControlNet Unit	NE1S-CNS21U	0.5
DeviceNet Unit	NE1S-DRM21U	0.2
Serial Communications Units	CS1W-SCU21-V1	0.30 (See note.)
High-resolution Motion Control Unit	CS1W-MCH71	0.80
MECHATROLINK-II-compatible Position Control Unit	CS1W-NCF71	0.36
Controller Link Unit	CS1W-CLK21-V1	0.52
Ethernet Units	CS1W-ETN01	0.40
	CS1W-ETN11	0.40
	CS1W-ETN21	0.38
FL-net Units	CS1W-FLN22	0.38
	CS1W-FLN02	0.40
Data Storage/Processing Units	CS1W-SPU01	0.56
	CS1W-SPU02	0.70

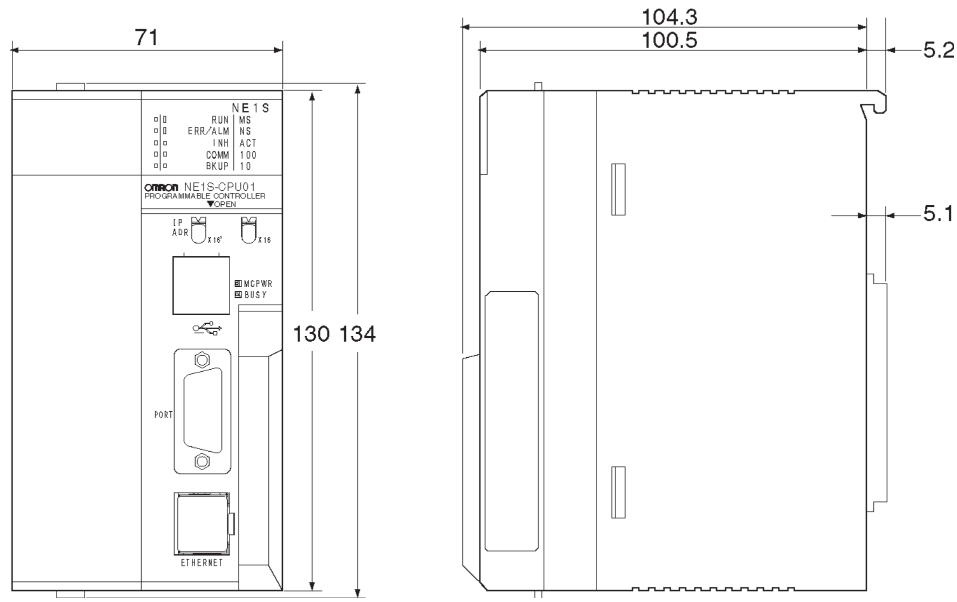
Note NT-AL001 Link Adapters consume 0.15 A/Unit when used.

26-V DC Voltage Group

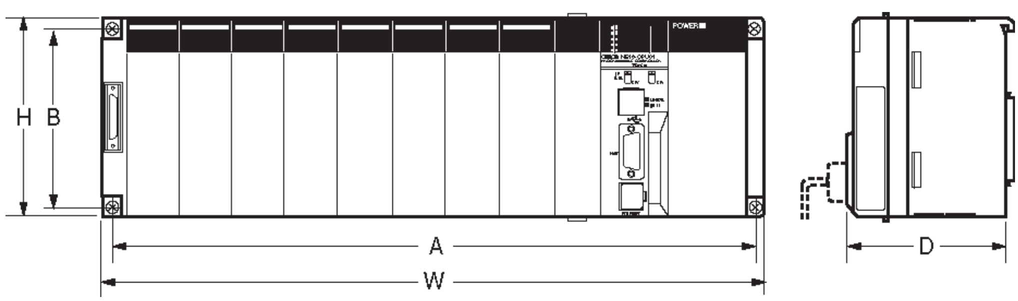
Category	Name	Model	Current consumption (A)
Basic Output Units	Relay Output Units	CS1W-OC201	0.006 per ON output point
		CS1W-OC211	0.006 per ON output point
Special I/O Units	Analog I/O Unit	CS1W-MAD44	0.20
	Analog Input Unit	CS1W-AD041-V1/081-V1	0.10
	Analog Output Unit	CS1W-DA041/08V	0.18
		CS1W-DA08C	0.25

1-3-5 Dimensions

CPU Unit



CPU Rack

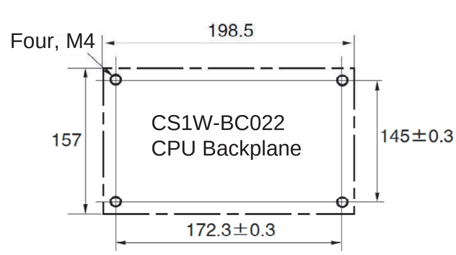


CPU Backplane	A (mm)	B (mm)	W (mm)	H (mm)	D (mm)
CS1W-BC022 (2 slots)	172.3	145	198.5	157	123
CS1W-BC032 (3 slots)	246	118	260	130	
CS1W-BC052 (5 slots)	316		330		
CS1W-BC082 (8 slots)	421		435		
CS1W-BC102 (10 slots)	491		505		

Mounting Dimensions

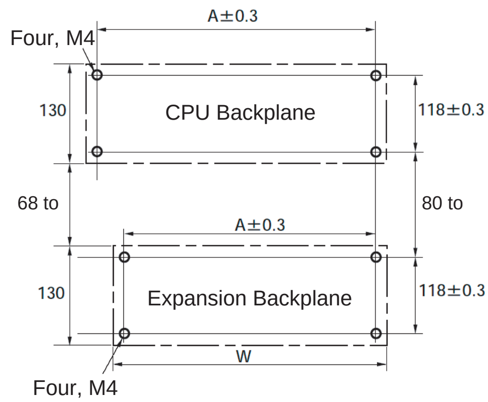
■ CPU Backplanes with 2 Slots

CS1W-BC022



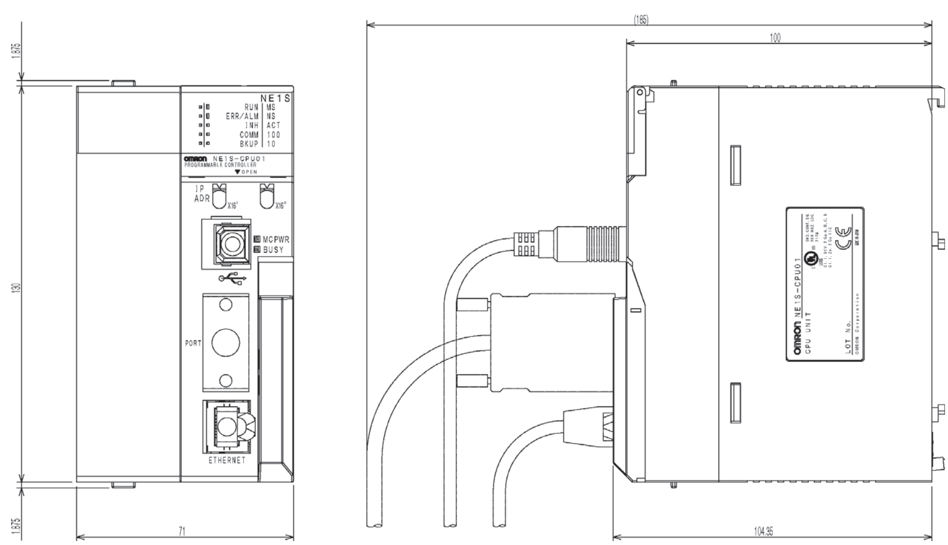
■ CPU Backplanes with 3, 5, 8, or 10 Slots and Expansion Backplanes

CS1W-BC032/52/82/102 or CS1W-BI032/052/082/102



Name	Model	A (mm)	W (mm)
CPU Backplane	CS1W-BC022 (2 slots)	172.3	198.5
	CS1W-BC032 (3 slots)	246	260
	CS1W-BC052 (5 slots)	316	330
	CS1W-BC082 (8 slots)	421	435
	CS1W-BC102 (10 slots)	491	505
Expansion Back-plane	CS1W-BI032 (3 slots)	246	260
	CS1W-BI052 (5 slots)	316	330
	CS1W-BI082 (8 slots)	421	435
	CS1W-BI102 (10 slots)	491	505

Mounting Height



1-4 Main Differences and Restrictions Compared with SYSMAC CS/CJ-series PLCs

1-4-1 Main Functional Improvements

The main differences between OMRON's SYSMAC CS/CJ-series PLCs and NE1S-series PLCs are described in this section.

Hardware

CPU Unit Built-in Ports

- NE1S-series CPU Units have an USB port instead of a peripheral port.
- NE1S-series CPU Units have a built-in Ethernet port with functionality equivalent to the CS1W-ETN11 Ethernet Unit.

Software

Function Blocks

- AND/OR logic for bits can be programmed for function block parameters (with some restrictions).
- With some restrictions, multiple function blocks can be positioned between the bus bars.
- Arrays can be specified for input variables and output variables (VAR_OUTPUT).

Variables

- The CPU Unit contains variable information and a variable name server. Therefore, variables in the CPU Unit can be access from external devices using variable names. (The variable name server accesses variable information to determine the physical address of the variable.)
  Variable information is automatically backed up in built-in flash memory whenever variables are written by the use. The variable information is automatically restored whenever power is turned ON to the PLC.

  Note Variable comments are not included in the variable information. Variable comments are automatically stored in a Memory Card whenever the program is changed (uploaded or edited online).
- Local variables in the program, local variables in function blocks, and global variables in the PLC all have the same structure.
- Structure variables are supported.
- Fewer data types are supported in the NE1S Series. The following data types are supported in the CS1 Series, but not in the NE1S Series:

  LINT, ULINT, LWORD, LREAL, CHANNEL, NUMBER, UINT_BCD, UDINT_BCD, and ULINT_BCD

Programming Language

- Programs and function blocks can be written in the structured text (ST) language.

Network Communications

- CIP (Common Industrial Protocol) message communications are supported instead of FINS message communications.
- Routing is possible with network path specifications without fixed routing tables.

Others

Files Saved in Memory Cards

The following files are automatically saved on Memory Card whenever a program is changed (downloaded or edited online): program history (scheduled function) comments (variable comments, instruction comments, and line comments), and ST programs.

Additional Instruction Functionality

- The following instructions, which are not supported by CS/CJ-series PLCs, are supported by the NE1S-series PLCs.
  SEND CIP COMMAND instruction (CSND) and SEND MAIL instruction (MLSND)
- Indirect word addresses can be used for the task number for TKON/TKOF instructions.

Parameter Area Contents

Settings for the built-in Ethernet power are stored in the parameter area.

Auxiliary Area

The following flags are supported by the NE1S but not by the CS-series PLCs.

Word address	Bit address	Name
A202	A202.13	Information Instruction Execution Enabled Flag 0
A202	A202.14	Information Instruction Execution Enabled Flag 1
A219	A219.13	Information Instruction Execution Error Flag 0
A219	A219.14	Information Instruction Execution Error Flag 1
A295	A295.07	Variable Table Mismatch Flag
A402	A402.01	Communications Interface Error
A403	A403.11	Comment Memory Error Variable Table Error (CNS)
A438	---	Built-in Ethernet Error
A439	A439.00	Built-in Ethernet Communications Enabled Flag

The following bits and words are supported in the CS Series but not by the NE1S Series.

Word address	Bit address	Name
A099	A099.00	UM Read Protection Status
A099	A099.01	Task Read Protection Status
A099	A099.02	Program Write Protection Status when Read Protection Is Set
A099	A099.03	Enable/Disable Status for Backing Up the Program to a Memory Card
A099	A099.14	IR/DR Operation between Tasks
A099	A099.15	Timer/Counter PV Refresh Mode Flag
A213	A213.00 to A213.07	Explicit Communications Error Flags
A214	A214.00 to A214.07	First Cycle Flags after Network Communications Finished
A215	A215.00 to A215.07	First Cycle Flags after Network Communications Error
A216 to A217	---	All Network Communications Completion Code Storage Address
A218	---	All Used Communications Port Numbers
A337	---	Number of Units Recognized at Startup (Racks 4 to 7)
A338	---	EM bank number
A339 to A340	---	Maximum Differentiation Flag Number
A345	A345.00	FB Program Data Flag
A345	A345.01	Program Index File Flag
A345	A345.02	Comment File Flag
A345	A345.03	Symbol Table File Flag
A355 to A359	A355.00 to A359.15	Inner Board Status

Word address	Bit address	Name
A392	A392.08	Peripheral (CP1H: Option 1) Port Status
A392	A392.12	Peripheral (CP1H: Option 1) Port Communications Error Flag
A392	A392.13	Peripheral (CP1H: Option 1) Port Send Ready Flag (No-protocol mode)
A392	A392.15	Peripheral (CP1H: Option 1) Port Reception Overflow Flag (No-protocol mode)
A394	A394.00 to A394.07	Peripheral (CP1H: Option 1) Port PT Communications Flag
A394	A394.08 to A394.15	Peripheral (CP1H: Option 1) Port PT Priority Registered Flags
A394	A394.00 to A394.15	Peripheral (CP1H: Option 1) Port Reception Counter (No-protocol mode only)
A395	A395.00	Fatal Error Flag
A395	A395.01	Non-fatal Error Flag
A401	A401.12	Inner Board Stopped Error Flag (fatal error)
A402	A402.05	SYSMAC BUS Error Flag
A402	A402.08	Inner Board Error Flag (non-fatal error)
A405	A405.00 to A405.01	SYSMAC BUS Master Flags
A405	A405.15	Peripheral Servicing Too Long Flag
A424	A424.00 to A424.15	Inner Board Error Information
A425	A425.04 to A425.06	Slave Number of SYSMAC BUS Error After Start-up
A425	A425.08 to A425.15	Master Number of SYSMAC BUS Error After Start-up
A442	A442.11	PLC Link Operating Level 1 Detection Flag
A442	A442.12	PLC Link Operating Level 0 Detection Flag
A509	A509.00	SYSMAC BUS Slave Number Refresh Bit
A526	A526.01	Peripheral (CP1H: Option 1) Port Restart Bit
A526	A526.14	SYSMAC BUS Master 1 Restart Bit
A526	A526.15	SYSMAC BUS Master 0 Restart Bit
A527	A527.00 to A527.07	Online Editing Disable Bit Validator
A527	A527.09	Online Editing Disable Bit
A528	A528.08 to A528.15	Peripheral (CP1H: Option 1) Port Error Code
A580	---	FB Communications Instruction Retries
A581	---	FB Communications Instruction Response Monitoring Time
A582	---	FB DeviceNet Communications Instruction Response Monitoring Time
A608	A608.00	Inner Board Restart Bit
A608	A608.01 to A608.03	Reserved for additional Inner Board Restart Bits
A609 to A613	A609.00 to A613.15	Inner Board User Interface Area
A619	A619.01	Peripheral (CP1H: Option 1) Port Settings Changing Flag
A650	A650.00 to A650.07	Program Replacement Completion Code
A650	A650.14	Program Replacement Error Flag

Word address	Bit address	Name
A650	A650.15	Program Replacement Start Bit
A651	---	Program Replacement Password
A652	---	Program Replacement Task Number
A654 to A657	---	Program Replacement Source File Name

1-4-2 Functional Restrictions

The following restrictions in functionally apply in comparison with the OMRON's SYSMAC CS/CJ-series PLCs.

Restrictions in Using Instructions

Mainly, the following instructions cannot be used. Refer to *Appendix G Instruction Support and Operand/Variable Restrictions* for details.

Subroutine instructions (SBS, GSBS, and MCRO), data control instructions (AVG, SCL, etc.), FINS communications instructions, FAILURE POINT DETECTION (FPD), ladder step control instructions (SNXT and STEP), some of the special instructions (IORS, IOSP, TOCV, etc.), special math instructions (APR, BCNT, ROOT, etc.), and block programming instructions.

Restrictions in Unit I/O Memory Physical Addresses

Physical addresses in I/O memory can be used to directly specify addresses for global variables. For NE Programmer, physical addresses can be input directly as an exception.

Restrictions in Direct Address Specifications for Global Variables

- Direct address specifications are supported only for the following areas.

  CIO (Core I/O Area), A (Auxiliary Area), D (Data Memory Area), E_0 (Extended Data Memory Area, Bank No. 0), H (Holding Area), W (Work Area)
- Direct address specifications cannot be used for the following portions of I/O memory.

  Indirect EM and DM area addresses (only in binary mode), EM Area bank 1 or higher

Restrictions in Directly Inputting Physical Addresses

EM Area bank 1 and higher are automatically allocated by the system to variables. The user cannot access these banks directly.

Restrictions in System Configuration

- Programming Consoles are not supported.
- Inner Boards are not supported.
- EM file memory is not supported.
- Only the following Special Units can be used.
  - ControlNet Unit: NE1S-CNS21U
  - Serial Communications Unit: CS1W-SCU21-V1
  - DeviceNet Unit: NE1S-DRM21U
  - Motion Control Units: CS1W-MC221-V1, CS1W-MC421-V1, and CS1W-MCH71
  - Ethernet Unit: CS1W-ETN01/11/21
  - FL-net Unit: CS1W-FLN22/12/02/01
  - Data Storage/Processing Unit: CS1W-SPU01/02
  - Controller Link Unit: CS1W-CLK21-V1

    - Analog I/O Units: CS1W-AD041-V1/081-V1/161, CS1W-DA041/08C/08V, and CS1W-MAD44
    - High-speed Counter Unit: CS1W-CT021/041
    - ID Sensor Unit: CS1W-V600C11/12
    - Position Control Unit: CS1W-NC113/133/213/233/413/433, CS1W-NCF71
    - Process Analog I/O Units: CS1W-PTS51/55
- C200H Basic I/O Units, Special I/O Units, and CPU Bus Units for communications cannot be used.

Restrictions in PLC Setup

- There are no settings for EM file memory conversion.
- There are no peripheral servicing mode settings.

Restrictions in Timers/Counters

- Only the binary mode is supported for timer and counter PVs. The BCD mode cannot be used.

Restrictions Related to File Memory

- There is no function to replace the program during operation.
- There is no file memory function for the EM Area.
- The simple backup operation for specific Units can be used only for the NE1S-DRM21U DeviceNet Unit, the CS1W-MCH71 MC Unit (MCH), and the NE1S-CNS21U ControlNet Unit.

Other Restrictions

- Extra cyclic tasks are not supported. There is thus a maximum of 128 tasks.

SECTION 2
Program Structure

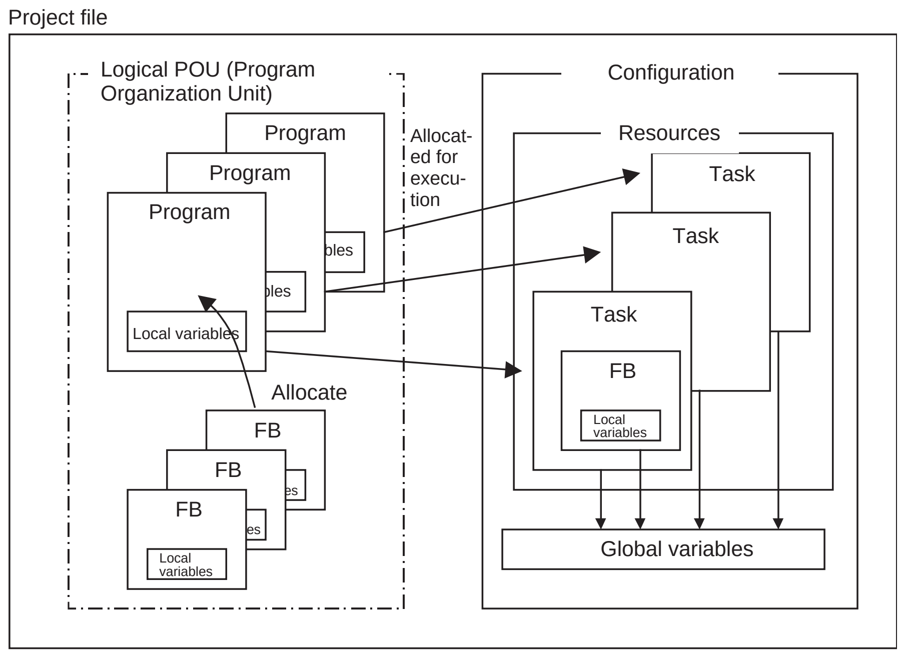
This section describes the structure of the programs.

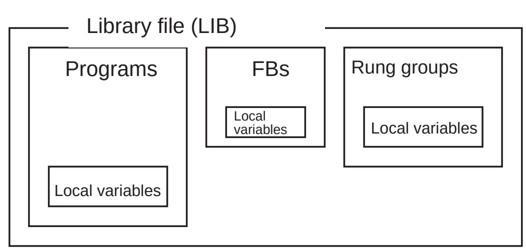
2-1 NE1S-series Structure

2-1-1 Project Structure

The NE1S-series PLCs are next-generation PLC that support programs conforming to the IEC 61131-3 standard.

All of the data is created in a single file, known as a project file, from an integrated development environment. A project is composed of logical POUs (Program Organization Units) and the configuration. The logical POUs become executable when they are allocated in the configuration (PLC system).





Program Organization Units (Logical POUs)

Program organization units (logical POUs) include the following elements.

1,2,3...
1. Programs
2. Function Blocks (FB)

All kinds of POUs are programmed entirely with variables. Variables are automatically used in a specified area in I/O memory and are allocated in the specified area.

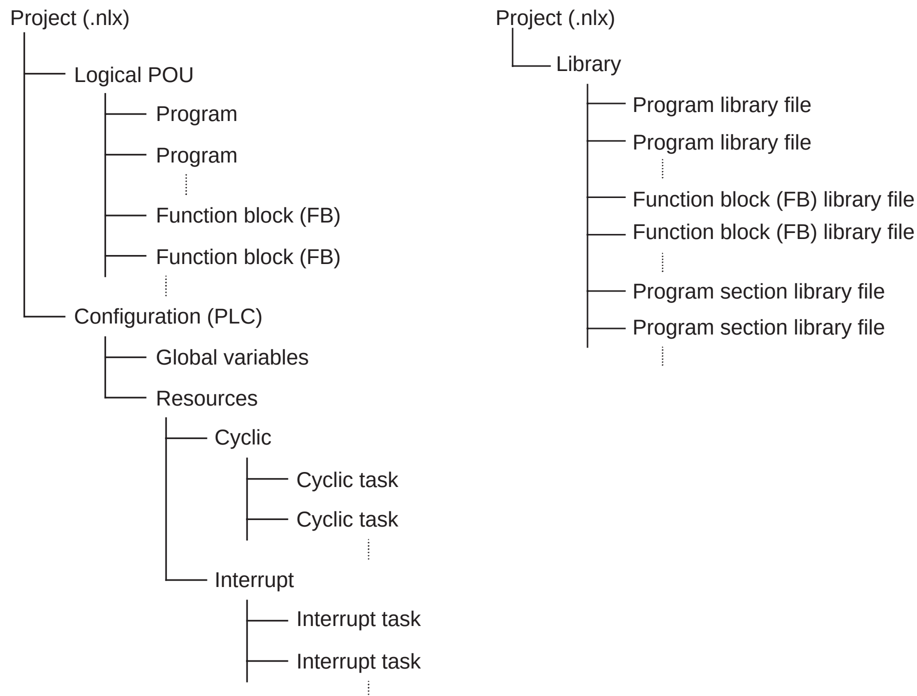
Configuration (PLC System)

The configuration (PLC system) is made up of resources (a task group of executable units) and global variables (variables that apply to all resources).

Libraries

A program (including local variables), function block, or rung groups (one rung or multiple rungs, including local variables) can be saved as a library function and reused. (Each unit is saved as a separate file, i.e., 1 program = 1 file, 1 function block = 1 file, or 1 rung group = 1 file.)

Note NE Programmer projects have the following directory-tree structure.



2-1-2 Logical POU (Program Organization Unit)

Logical POU Structure

A single POU is defined with the following two software elements.

1,2,3...

1. Declaration of memory usage (variable declarations)
2. Logic in an algorithm

The logical POU can be reused when these two elements have been converted to a library file as a set. (It is also possible to reuse multiple rungs that have been converted to a library file with the variables used in the rungs.)

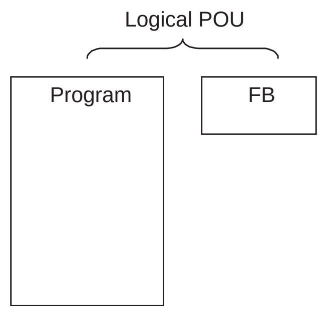
Logical POU Types

There are two types of logical POUs: programs and function blocks.

1,2,3...

1. Programs
   Programs are large functional units or units that must be executed at a specific time. Basically, the unit is entirely declared in the user interface.

2. Function Blocks
   Function blocks are components of programs and are not executed at a specific time. The unit is displayed as a single function in the user interface, so only inputs and outputs are declared.



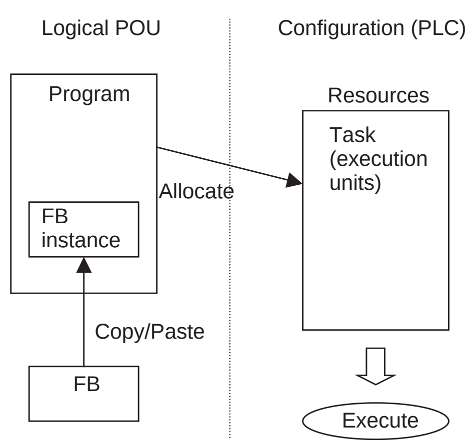
Item	Logical POU	
	Program	Function block (FB)
Unit size	Large functional unit	Functional unit smaller than program
Execution timing	(or) Requires execution at specific time.	Does not require execution at a specific time.
User interface	All disclosed.	I/O only disclosed.

Executing a Logical POU

The logical POU itself is not an executable unit; it is allocated to the configuration (PLC) first and then can be executed.

1,2,3...

1. Programs are allocated to “tasks” in the resources (creating a program instance) to become executable.
2. Function blocks are copied and pasted into a program (creating a function block instance) to become executable.



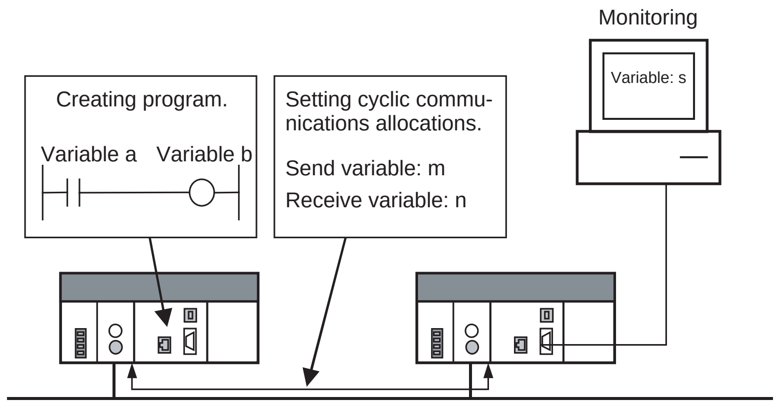
Creating a Logical POU in NE Programmer

After creating a NE Programmer project, select ***File - New*** from the menus, click the **Function Block** Tab, and input the POU name. Set the *POU Type* to either ***Program*** or ***Function Block***.

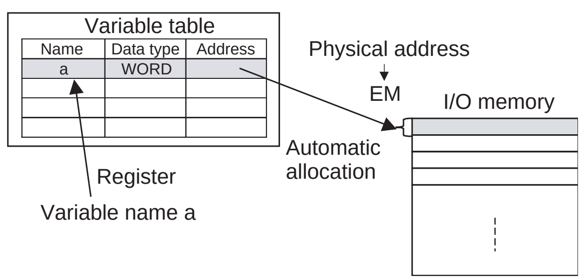
Note In order to paste a created function block in a program (creating an instance), the created function block must be saved. Once the function block has been saved, its icon will appear in the Project Explorer Bar and the icon can be dragged and dropped into the desired program.

2-1-3 Variables

The user does not specify I/O memory addresses directly when identifying a NE1S-series CPU Unit's I/O memory addresses (known as a physical addresses) for programming, making communications settings, or monitoring. Specify the CPU Unit's I/O memory through variables.



If the user has entered a variable in the variable table, the created variable is automatically allocated a physical address in the CPU Unit's I/O memory. It is not necessary for the user to know which physical addresses have been allocated.



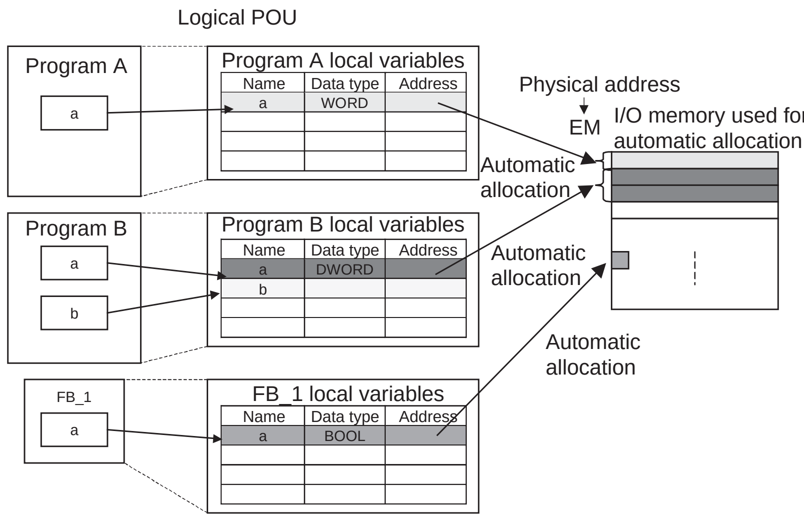
Local Variables and Global Variables

There are two basic variable types. Local variables are defined in individual logical POUs and used just within the local logical POU. Global variables are defined at the configuration level and can be used in all of the logical POUs.

Local Variables

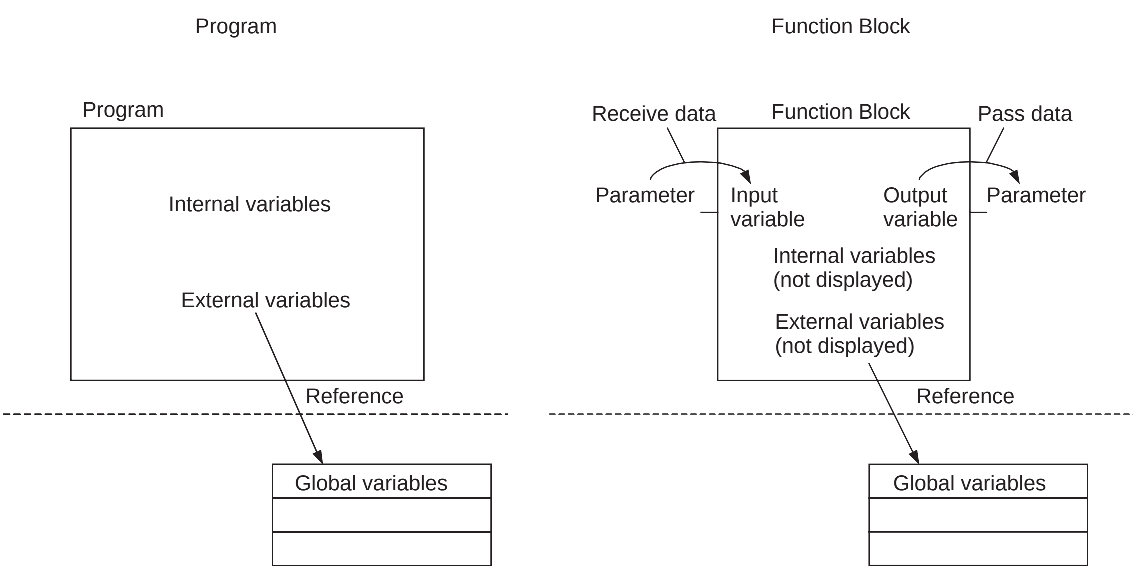
Variables defined in individual logical POUs (programs or function blocks) are known as local variables. These variables are effective only in the local logical POU.

The system automatically allocates local variables (both local program and local FB variables) to a particular memory area (EM area banks other than EM bank 0). The physical addresses are hidden.



The various kinds of local variables are listed below.

- Program: Internal variables (VAR) and external variables (VAR_EXTERNAL)
- Function Block: Internal variables (VAR), input variables (VAR_INPUT), output variables (VAR_OUTPUT), and external variables (VAR_EXTERNAL)



- Internal variables (VAR) are used only in the logical POU, they do not exchange values with external parameters, and cannot reference external global variables.
- External variables (VAR_EXTERNAL) are special variables used within the logical POU to reference global variables outside of the POU.
- Input variables (VAR_INPUT) exist only in function blocks and are used to receive the values of external parameters.
- Output variables (VAR_INPUT) exist only in function blocks and are used to pass values to external parameters.

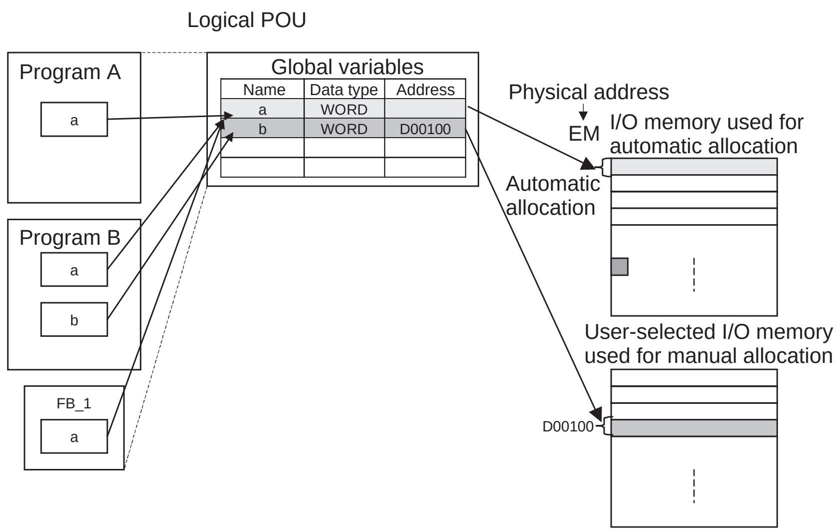
Note Even if local variables are defined with the same name in different logical POUs, they are treated as different variables and allocated different physical addresses.

Global Variables

Variables defined at the configuration level and shared by all logical POUs are known as global variables. Global variables provide an interface between logical POUs.

The system can allocate memory to global variables automatically or specific physical addresses can be allocated manually (direct specification).

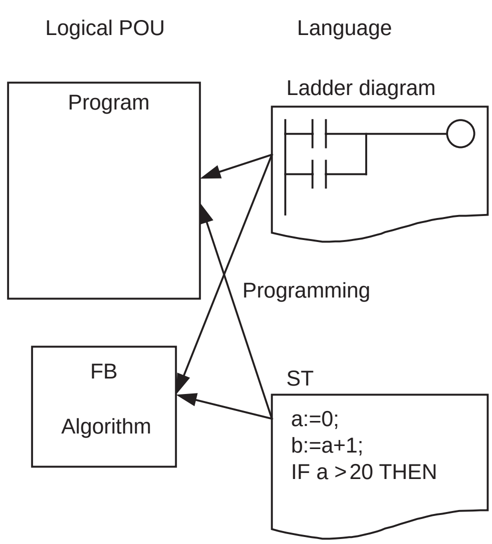
To use global variables within logical POUs, global variables must be referenced from external variables as local variables. For details on external variables, refer to *External Variables (VAR_EXTERNAL)* in *2-2-10 Details on Local Variables*.



Note Physical addresses can also be directly specified without using variables.

2-1-4 Programming Languages

The algorithms in programs and function blocks (FB) can be written as ladder diagrams (LD) or structured text (ST).



Creating Programs in NE Programmer

When creating logical POUs in NE Programmer, select **LD** when creating a ladder diagram or **ST** when creating structured text.

Note

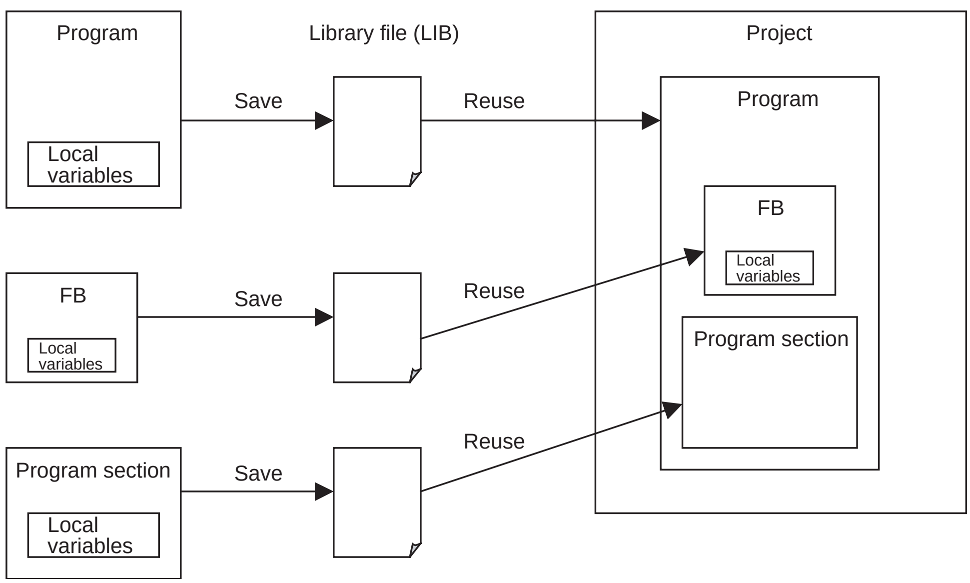
(1) When using ladder diagrams, the program can be input directly in the ladder diagram with mnemonics. Mnemonics can also be input in a mnemonics-only editing screen by selecting ***Edit - Edit Using Mnemonic Editor*** from the menus. It is not possible, however to create logical POUs that call function blocks.

(2) When using ladder diagrams, it is possible to import a text file (.txt) containing the mnemonics. To import a file, 1) Select the logical POU in program view, right-click, and select ***Change View***, and then 2) Right-click in mnemonic view, right-click, and select **Import.**

2-1-5 Libraries

Converting a Logical POU or Rungs to a Library

In the NE1S-series, a logical POU (program or function block) or rungs in a program can be converted to a library file and reused.



Creating Library Files in NE Programmer:
Either select the desired logical POU and select ***Library - POU - Register to Library*** or select the logical POU in project view, right-click, and select ***Register to Library*** from the popup menu. Input the file name and click the **OK** Button.

2-2 Variables

2-2-1 Naming Variables

The following rules apply to variables compliant with the IEC61131-3 standards.

- Number of characters: 48 characters max.
- Usable characters: numbers 0 to 9, letters a to z, letters A to Z, and the underscore character.

- Upper and lower case characters are distinguished but do not make variables different. For example, “aBc” and “AbC” are treated as the same variable, but the variable “aBc” is recorded as “aBc”.
- Variable names cannot begin with a number (0 to 9).
- There cannot be two or more consecutive underscore characters.
- Variable names cannot have the following characters followed by a number, because these combinations are treated as actual data area addresses.
  A, W, H, T, C, D, E0, @D, *D, @E□, *E□, IR, DR, TK, and TR
- Also, the actual address expressions of external variables (preassigned variables) cannot be used.
- Variables with only numbers are treated as actual addresses, e.g., “D00000” is treated as the word address D00000 and “0000” is treated as the word address CIO 0000.

2-2-2 Types of Variable

The following variable types are supported.

- Internal Variables (VAR)
  Internal variables are used only within an instance. They cannot be used pass data directly to or from parameters outside of the instance.
- Input Variables (VAR_INPUT)
  Input variables can input data from parameters outside of the instance. The default input variable is an EN (Enable) variable, which passes input condition data.
- Output Variables (VAR_OUTPUT)
  Output variables can output data to parameters outside of the instance. The default output variable is an ENO (Enable Out) variable, which passes the instance’s execution status.
- External Variables (VAR_EXTERNAL)
  External variables are local variables that are used to access global variables. They include both system variables that are registered in the NE Programmer in advance, as well as user-defined local variables. When handling I/O with variables, always define I/O as global variables and access the global variables through external variables. System variables are registered in advance as external variables. User-defined global variables are automatically registered as external variables when they are used as operands for instructions in logical POUs. If a variable is first created as an external variable, the user must manually register it as a global variable in the External Variable Tab Page of the Variable Editor to use it as a global variable. We thus recommend that the required global variables are defined before starting to program the ladder diagrams.
- Global Variables
  Global variables are defined in the in the configuration and are shared by all logical POUs. Global variables include system variables, such as the Conditions Flags and some Auxiliary Area bits, that are registered in the NE Programmer in advance, as well as user-defined global variables.

2-2-3 Variable Properties

The following table lists the variable properties.

Variable property	Content	Value
Data type	Selects the variable's data type.	BOOL, INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIMER, COUNTER, or STRING
Array elements	Selects whether the variable is an array and sets the number of array elements when it is an array.	When not specifying an array, leave this setting at 0. When specifying an array, set the number of array elements between 1 and 255.
Initial value	With a program, this property sets the variable's value at the start of operation. With a function block, this property sets the variable's value when an instance is executed.	Set an initial value as follows, according to the variable's data type: • BOOL, WORD, or DWORD: Input the value in unsigned hexadecimal after "16#". • INT or DINT: Input the value in unsigned decimal after "+10#" or "–10#". • UINT or UDINT: Input the value in unsigned decimal after "10#". • REAL: Input the value as a real number after the sign.
Address	This property sets a fixed address when a fixed address is being set manually (direct specification). Fixed addresses can be set for global variables only.	---
Network variable setting (disclosed to the network)	When using cyclic communications, select **Input** when presenting the variable as an input from the network, select **Output** when presenting the variable as an output to the network, and select **None** when the variable will not be presented.	None, Input, or Output
Retain/nonretain	This property specifies whether the variable's value is retained when operation starts and the power is reset.	Retained or not retained.
Comment	This is the variable comment, which can be up to 256 bytes (256 ASCII characters) long. **Note** 1. The variable comment can be displayed in the Ladder Diagram Screen. To display the comment, select ***View - Option*** from the NE Programmer menu and select the *Show variable comment* Option in the *Display Settings* Tab. 2. The comments are stored in the Memory Card as a comment file.	---

The properties listed in the table are described in detail below.

2-2-4 Data Type

The variable's data format is called the data type. The following data types are supported in the NE1S Series.

Data type

- Basic types
  - INT (Integer)
  - DINT (Double integer)
  - UINT (Unsigned integer)
  - UDINT (Unsigned double integer)
  - BOOL (1 bit)
  - WORD (16 bits)
  - DWORD (32 bits)
  - REAL (Real number)
  - TIMER (timer)
  - COUNTER (counter)
  - STRING (Character string)

  Array variables can be created (except for STRING).
- User-set types
  - Structure

  Arrays can be created.
  Structure variables can be set as arrays.

Note Arrays and Structures
In the NE1S Series, one-dimensional array variables and user-defined structured variables can be created.

- One-dimensional array variables:
  A variable can be set as a one-dimensional array by setting the number of elements to an integer value between 1 and 255.
- Structured variables:
  A structure is a set of variables containing several variables (members) with different data types. The user can configure the structure freely. Members can be specified by specifying the variable name and member name.

Data Ranges of Data Types

Data type	Contents	Size	Data range
BOOL	Bit data	1 bit	16#0 or 16#1
INT	Integer	16 bits	–10#32768 to +10#32767
UINT	Unsigned integer	16 bits	10#0 to 10#65535
DINT	Double integer	32 bits	–10#2147483648 to +10#2147483647
UDINT	Double unsigned integer	32 bits	10#0 to 10#4294967295
WORD	16-bit data	16 bits	16#0000 to 16#FFFF
DWORD	32-bit data	32 bits	16#00000000 to 16#FFFFFFFF
REAL	Real number	32 bits	Conforms to IEEE754
STRING	Character string data (ASCII data)	128 bytes	---

Data type	Contents	Size	Data range
TIMER	Timer (See note 1.)	1 bit or 16 bits	16#0, 16#1, or 16#0000 to 16#FFFF
COUNTER	Counter (See note 2.)	1 bit or 16 bits	16#0, 16#1, or 16#0000 to 16#FFFF

Note (1) When a variable is entered in the timer number (0 to 4095) operand of a timer instruction, such as TIM or TIMH, the data type will be TIMER. When this variable is used as an operand in another instruction, it will be treated as the timer Completion Flag if the operand takes 1-bit data or as a timer PV if the operand takes 16-bit data. The timer PVs are 16-bit binary data because NE Programmer uses only the binary format for the PVs.
The TIMER data type cannot be used in ST language function blocks.

(2) When a variable is entered in the counter number (0 to 4095) operand of a counter instruction, such as CNT or CNTR, the data type will be COUNTER. When this variable is used as an operand in another instruction, it will be treated as a counter Completion Flag if the operand takes 1-bit data or as a counter PV if the operand takes 16-bit data. The counter PVs are 16-bit binary data because NE Programmer uses only the binary format for the PVs.
The COUNTER data type cannot be used in ST language function blocks.

2-2-5 Array Elements (Array Specification)

A set of data with the same properties can be handled in a single structure (array). Input the array's maximum number of elements in this property to define the variable as an array.

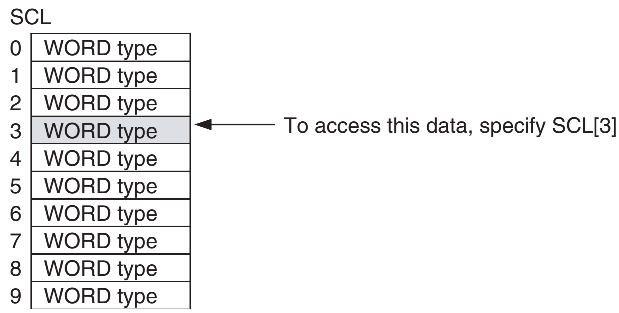
NE Programmer supports one-dimensional arrays only.

- The number of elements in an array can be set between 1 and 255.
- The array specification can be used for internal variables (VAR), input variables (VAR_INPUT), output variables (VAR_OUTPUT), and external variables (VAR_EXTERNAL).
- When entering the variable name, specify the subscript in square brackets after the variable name.
- The subscript (for example, the subscript in variable a[ ]) can be specified in the following three ways.
  a. Directly specify the subscript number (ladder or ST language).
  Example: a[2]
  b. Specify a local variable as the subscript (ladder or ST language).
  Example: a[n], where n is a local variable
  c. Specify an arithmetic expression as the subscript (ST language only).
  Example: a[b+c], where b and c are local variables

  Note The four arithmetic operators (+, -, *, and /) can be used in expressions.

Note An array is a group of data items with the same data type. Each variable element is specified by the local variable name and a subscript. (The subscript indicates the position of an element within the array.) With a one-dimensional array, the subscript's number indicates the element in the array.
Example) An array with 10 elements and local variable name SCL

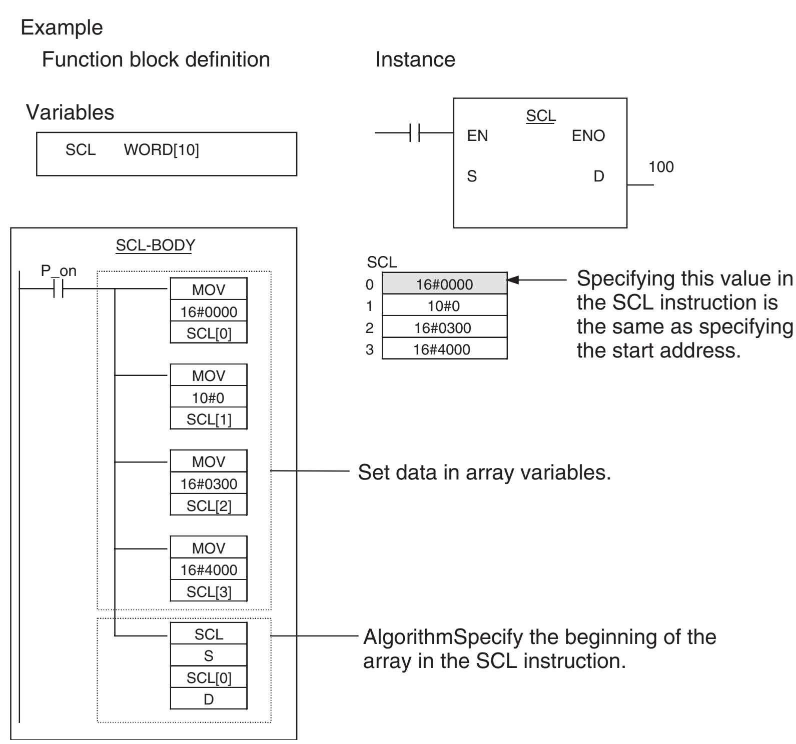
The following 10 local variables can be used: SCL[0], SCL[1], SCL[2], SCL[3], SCL[4], SCL[5], SCL[6], SCL[7], SCL[8], and SCL[9].



Note When specifying the starting address (or end address) of multiple words in an instruction operand (see note), the address cannot be passed to a local variable by an input parameter or output parameter.

Note For example, multiple words are accessed when specifying an instruction's control data or the starting and end words of the BLOCK SET instruction (BSET(071)).

In this case, prepare arrays or structured variables with the required number of elements, set the data in the arrays or structures in the function block definition, and specify the start (or end) of the array or structure in the operand. This method effectively specifies the starting address (or end address) of multiple words.



(For details, refer to *Appendix A Variable Applications Guidelines*.)

Setting the Property

- Input the number of elements (1 or more) in the *Array Size* Field in the Edit Variables Dialog Box. If the *Array Size* Field is set to 0, the variable will not be an array.
- When entering the local variable name, specify the subscript in square brackets after the local variable name.
  Example: When BOOL variable “a” is defined as an array with 8 array elements, the 7th element of variable “a” can be input as the operand bit by inputting LD a[7].

Note Basically, use arrays when specifying the starting address (or end address) of multiple words in an instruction operand. The data cannot be passed from a parameter because parameters pass values.
Prepare a local array variable of the required size, set the data in the array in the function block definition, and specify the start (or end) of the local array variable in the operand. For details, refer to *Appendix A Variable Applications Guidelines*. For details on using arrays to specify the starting/end addresses of multiple-word operands, refer to *Appendix G Instruction Support and Operand/Variable Restrictions*.

2-2-6 Initial Value

For programs, this property specifies the variable's value when operation starts.

For function blocks, this property specifies the variable's value when the instance is executed. When an instance is first executed, internal variables (VAR) and output variables (VAR_OUTPUT) are set to the initial value. Later, the variable's value may change as the instance is executed.

Setting the Property

Input the desired value in the *Initial Value* Field in the *Edit Variables* Dialog Box. Input a value consistent with the variable's data type, as shown in the following table.

Data type	Contents	Initial value	Initial value input method
BOOL	Bit data	16#0 or 16#1	Input as unsigned hexadecimal. Input the value after “16#”.
INT	Integer	–10#32768 to +10#32767	Input as signed decimal. Input the value after “+10#” or “–10#”. (If “10#” is input, a + sign will be attached automatically.)
UNIT	Unsigned integer	10#0 to 10#65535	Input as unsigned decimal. Input the value after “10#”.
DINT	Double integer	–10#2147483648 to +10#2147483647	Input as signed decimal. Input the value after “+10#” or “–10#”. (If “10#” is input, a + sign will be attached automatically.)
UDINT	Double unsigned integer	10#0 to 10#4294967295	Input as unsigned decimal. Input the value after “10#”.
WORD	16-bit data	16#0000 to 16#FFFF	Input as unsigned hexadecimal. Input the value after “16#”.
DWORD	32-bit data	16#00000000 to 16#FFFFFFFF	Input as unsigned hexadecimal. Input the value after “16#”.
REAL	Real number	0.0	Input a signed numerical value. Input the value after the sign. E.g., +1.0, –0.23, +9.87E-3
STRING	Character string data (ASCII data)	---	---

2-2-7 Address (Direct Allocation of a Physical Address)

A physical address cannot be directly allocated to a local variable.

With external variables, a global variable with a direct address designation can be referenced. When specifying a physical address, input the physical address directly.

When using ST language, a physical address cannot be directly allocated.

For example, if “100” is input in an ST program, it is interpreted as the decimal value 100 (10#100). If “100” is input in a ladder program, it is interpreted as CIO address CIO 0100.

2-2-8 Retain/Nonretain

This property specifies whether the variable will retain its data when the PLC is turned form OFF to ON, operation starts, or a fatal error occurs. (If the CPU Unit is not equipped with a battery, the data will be cleared regardless of the Retain setting.)

Effect of IOM Hold Bit (A50012) Setting

When a global variable is directly allocated, the variable's data will be retained when the IOM Hold Bit (A50012) is ON, even if the Retain setting is not set to retain the variable's data. (If the CPU Unit is not equipped with a battery, the data will be cleared regardless of the IOM Hold Bit and Retain settings.)

The following tables show whether variable data is retained or cleared for the various Retain and IOM Hold Bit (A50012) settings when the power is turned ON, operation starts, or a fatal error occurs.

Local and Global Variables

Retain/ Nonretain setting	Address	Initial value	IOM Hold Bit (A50012)	
			0 (OFF)	1 (ON)
Retain	Automatic allocation	Cannot be set	Retained (See note.)	Retained (See note.)
Nonretain	Automatic allocation	Set	Initial value	Initial value
		Not set	Cleared (0)	Cleared (0)

Effect of Forced Status Hold Bit (A50013) Setting

Operation of the Retain setting is not related to the Forced Status Hold Bit setting.

The following tables show whether variable data is retained or cleared for the various Retain and Forced Status Hold Bit settings when the power is turned ON, operation starts, or a fatal error occurs.

When global variables are directly allocated, the forced status will be retained only when the Forced Status Hold bit is ON. Otherwise, the forced status will be cleared.

Global Variables

Retain setting	Address	Forced Status Hold Bit (A50013) setting	
		0 (OFF)	1 (ON)
Retain or Do not retain	Direct allocation	Forced status cleared.	Forced status retained.
	Automatic allocation		Forced status cleared.

Local Variables

Retain/Nonretain setting	Address	Forced Status Hold Bit (A50013) setting	
		0 (OFF)	1 (ON)
Retain or Do not retain	Automatic allocation	Forced status cleared.	Forced status cleared.

Setting the Property

Select ***Retain*** or ***Nonretain*** in the *Retain/Nonretain* Field of the *Edit Variables* Dialog Box.

When allocating address to global variables, they will be set to either *Retain* or *Nonretain* and cannot be changed.

2-2-9 Local Variable Properties and Types of Local Variables

The following table shows which properties must be set, can be set, and cannot be set, based on the local variable usage.

Property	Variable usage			
	Internal	Input	Output	External
Name	Must be set.	Must be set.	Must be set.	Must be set.
Type	Must be set.	Must be set.	Must be set.	Must be set.
Initial Value	Can be set.	Can be set.	Can be set.	Can be set.
Array specification	Can be set.	Can be set.	Can be set.	Can be set.
Retain	Can be set.	Can be set.	Can be set.	Can be set.
AT (direct allocation to a specific physical address, global variables only)	Cannot be set.	Cannot be set.	Cannot be set.	Cannot be set.

2-2-10 Details on Local Variables

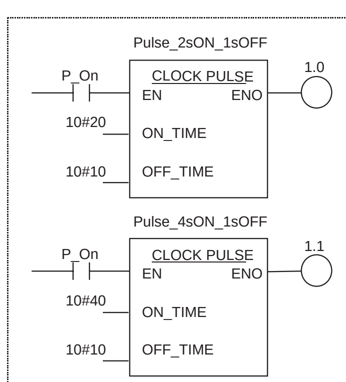
■ **Internal Variables (VAR)**

Internal variables are used within an instance. These variables are internal to each instance. They cannot be referenced from outside of the instance and are not displayed in the instance.

The values of internal variables are retained until the next time the instance is executed. Consequently, even if instances of the same function block definition are executed with the same I/O parameters, the result will not necessarily be the same.

Example:

The internal variable tim_a in instance Pulse_2sON_1sOFF is different from internal variable tim_a in instance Pulse_4sON_1sOFF, so the instances cannot reference and will not affect each other's tim_a value.



Variable table

Usage	Name	Type
Internal	tim_a	WORD
Internal	tim_b	WORD
Input	ON_TIME	INT
Input	OFF_TIME	INT

Note In NE Programmer, internal variables are created with a default variable name of “FI”, which is a local variable that is ON the first time that an instance is executed. (The “FI” internal variable can be used to initialize the instance.)

Retaining Data through Power Interruptions and Start of Operation

Internal variables (VAR) retain the value from the last time that the instance was called. In addition, the Retain Option can be selected so that an internal variable will also retains its value when the power is interrupted or operation starts (the mode is switched from PROGRAM mode to RUN or MONITOR mode).

Initial Value

An initial value can be set for an internal variable (VAR) that is not being retained (i.e., when the Retain Option not selected). An initial value cannot be set for an internal variable if the Retain Option is selected.
Internal variables that are not being retained will be initialized to 0.

The initial value will be set, regardless of the value of the IOM Hold Bit (A50012).

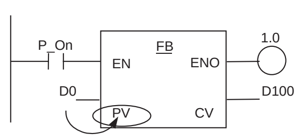
Auxiliary Area control bit		Initial value
IOM Hold Bit (A50012)	ON	Initial value is set.
	OFF	Initial value is set.

Local Variable Usage (Types)

■ Input Variables (VAR_INPUT)

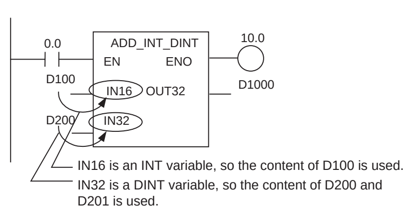
Input variables pass arguments to the instance from the outside. The input variables are displayed on the left side of the instance.

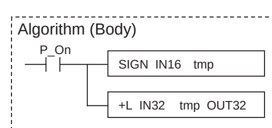
When an instance is called, the value of the input source (data contained in the specified parameter just before the instance was called) will be passed to the input variable.



The value of the parameter specified as the input (value of D0) is passed to the instance’s input variable (PV).

Example





Variable table

Usage	Name	Type
Internal	tmp	DINT
Input	EN	BOOL
Input	IN16	INT
Input	IN32	DINT
Output	ENO	BOOL
Output	OUT32	DINT

Note

(1) The same variable name cannot be assigned to an input variable (VAR_INPUT) and output variable (VAR_OUTPUT). If it is necessary to have the same variable as an input variable and output variable, register the variables with different names and transfer the value of the input variable to the output variable in the function block with an instruction such as MOV.

(2) When the instance is executed, input values are passed from parameters to input variables before the algorithm is processed. Consequently, val-

ues cannot be read from parameters to input variables while the algorithm is being processed. If it is necessary to read a value within the execution cycle of the algorithm, do not pass the value from a parameter. Assign the value to an internal variable and use an AT setting (direct allocation of a physical address).

Initial Value

When you set an initial value for an input variable (VAR_INPUT), that value will be written to the variable when the parameter for input variable EN goes ON and the instance is executed for the first time (and that one time only). If an initial value has not been set for an input variable, the input variable will be set to 0 when the instance is first executed.

EN (Enable) Variable

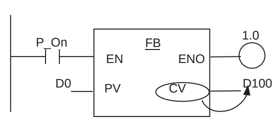
When an input variable (VAR_INPUT) is created, the default input variable is the EN variable. The instance will be executed when the parameter for input variable EN is ON.

The initial value will be set, regardless of the value of the IOM Hold Bit (A50012).

■ Output Variables (VAR_OUTPUT)

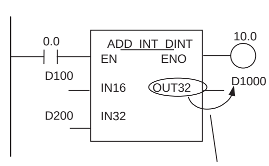
Output variables pass return values from the instance to external applications. The output variables are displayed on the right side of the instance.

After the instance is executed, the value of the output variable is passed to the specified parameter.

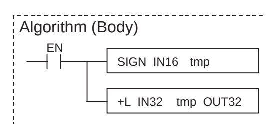


The value of the output variable (CV) is passed to the parameter specified as the output destination, which is D100 in this case.

Example



OUT32 is a DINT variable, so the variable's value is passed to D1000 and D1001.



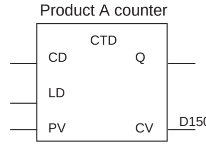
Variable table

Usage	Name	Type
Internal	tmp	DINT
Input	EN	BOOL
Input	IN16	INT
Input	IN32	DINT
Output	ENO	BOOL
Output	OUT32	DINT

Like internal variables (VAR), the values of output variables (VAR_OUTPUT) are retained until the next time the instance is executed.

Example:
In the following example, the value of output variable CV will be retained until the next time the instance is executed.

Product A counter



Note (1) The same variable name cannot be assigned to an input variable (VAR_INPUT) and output variable (VAR_OUTPUT). If it is necessary to have the same variable as an input variable and output variable, register the variables with different names and transfer the value of the input variable to the output variable in the function block with an instruction such as MOV.

(2) Output variable values (data) are passed to the corresponding parameters after the algorithm is processed. Consequently, values cannot be written from output variables to parameters while the algorithm is being executed. If it is necessary to write a value within the execution cycle of the algorithm, do not write the value to a parameter. Assign the value to an internal variable and use an AT setting (direct allocation of a physical address).

Initial Value

An initial value can be set for an output variable (VAR_OUTPUT) that is not being retained, i.e., when the Retain Option is not selected. An initial value cannot be set for an output variable if the Retain Option is selected.

The initial value will be set, regardless of the value of the IOM Hold Bit (A50012).

ENO (Enable Output) Variable

When a local output variable is created, the default variable is the ENO variable. The ENO output variable will be turned ON when the instance is called. The user can change this value. The ENO output variable can be used as a flag to check whether or not instance execution has been completed normally.

■ **External Variables (VAR_EXTERNAL)**

External variables are local variables that are used to access global variables. They include both system variables that are registered in the NE Programmer in advance, as well as user-defined local variables. When handling I/O with variables, always define I/O as global variables and access the global variables through external variables. System variables are registered in advance as external variables. User-defined global variables are automatically registered as external variables when they are used as operands for instructions in logical POUs. If a variable is first created as an external variable, the user must manually register it as a global variable in the External Variable Tab Page of the Variable Editor to use it as a global variable. We thus recommend that the required global variables are defined before starting to program the ladder diagrams.

2-2-11 Creating Variables in NE Programmer

In NE Programmer, variables are not declared by inputting a declarative statement. To create variables, insert the variables in the variable tables and input the properties.

Internal Variables

When a logical POU (program or function block) is created, the unit's individual variable table is displayed automatically. Variables can be registered in the table with either of the following methods.

1,2,3...

1. Inputting the instructions first:
   If a new variable name is input in an operand when inputting the instruction, the variable will be registered automatically in the variable table's internal variable sheet.
2. Inputting the variable table first:
   Right-click the variable table, select ***Insert***, and register the variable in the variable table. When the instructions are input later, the registered variable name can be input in the operands.

Physical addresses **cannot** be input directly for local variables by inputting the physical address in the *Edit Variables* Dialog Box.

Global Variables

After creating a NE Programmer project, select ***File - New*** from the menus, change the configuration name in the *Configuration* Tab if necessary, and create a new configuration. In the project workspace, double-click the global variables under the configuration. Right-click the variable table, select ***Insert Variable***, and register the variable in the variable table. When the instructions are input later, the registered variable name can be input in the operands.

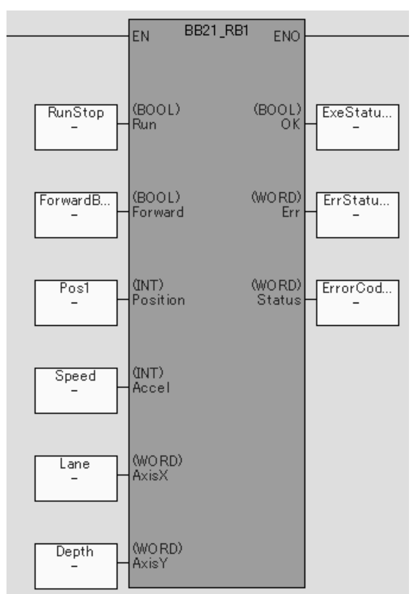
Physical addresses can be input directly for global variables by inputting the physical address in the *Address* Field in the *Edit Variables* Dialog Box.

2-2-12 Grouping Variables

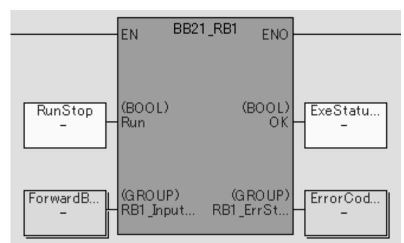
In function blocks, the input variables (VAR_INPUT) and output variables (VAR_OUTPUT) can be placed in separate groups and displayed with input and output group names.

When grouping input or output variables, the group name is displayed on the input and output side when the function block is pasted into a program.

Display Example of Ungrouped Variables



Display Example of Grouped Variables



Grouping FB I/O Variables in NE Programmer:
Select the input or output variables to be grouped, right-click, and select ***Group Input/Output Variables - Group*** from the popup menu.

2-2-13 Importing and Exporting Variables

Exporting Variables

In the NE1S Series, variable data can be exported in the following file formats.

- OPC Server CSV file (for use with SCADA Software)
- Text file for use with the CX-Designer
- Text file for use with the SPU-Console
- CSV file (for use with external Programming Devices or programs such as Excel)

Exporting with NE Programmer:

- OPC Server, CX-Designer, or SPU-Console Format:
  Select ***File - Export Variable*** and select the desired file format.

- CSV Format:
  Right-click in the variable table and select ***Export***.

Importing Variables

The following data can also be imported.

- CSV files (for use with external Programming Devices or programs such as Excel)

Importing with NE Programmer:
Right-click the local or global variable table and select ***Import - CSV Format***.

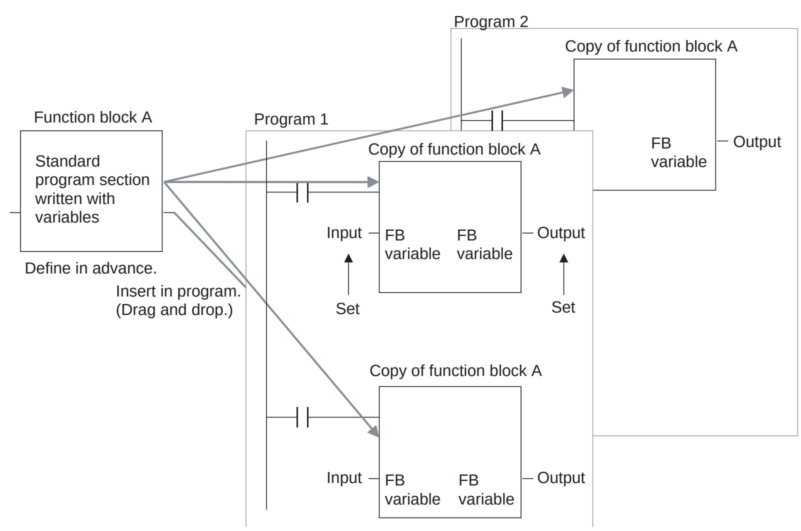
2-3 Function Blocks

2-3-1 Function Block Features and Operation

Overview

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 (constants, local program variables, or physical addresses) in those variables to use the function block.



Advantages of Function Blocks

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

Structured Programming

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

■ **Easy-to-read "Black Box" Design**

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

■ Easily Create Different Processes from a Single Function Block

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

■ Reduce Coding Errors

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

■ Protect Data

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.)

■ Programming with Variables provides Improved Reusability

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.

Compatible with Multiple Languages

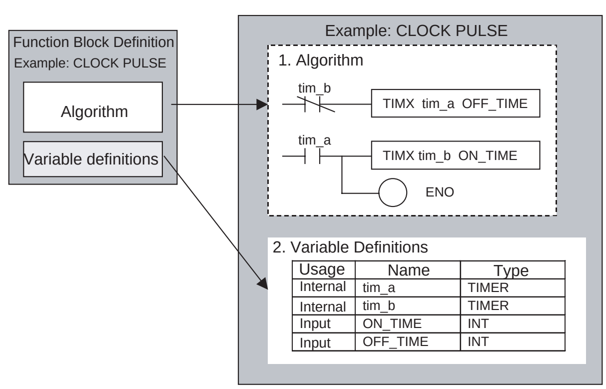
Mathematical expressions that are difficult to enter in ladder language can be entered easily in structured text (ST) language.

Function Block Structure

A function block consists of the function block definition that is created in advance and the function block instances that are actually inserted in the program.

Function Block Definition

The basic, reusable part of the function block is called the "function block definition." Each function block definition contains the algorithm and local variable definitions, as shown in the following diagram.



Usage	Name	Type
Internal	tim_a	TIMER
Internal	tim_b	TIMER
Input	ON_TIME	INT
Input	OFF_TIME	INT

1. Algorithm

The algorithm is standard programming written with variable names rather than physical I/ O memory addresses. In NE Programmer, algorithms can be written in either ladder programming or ST (structured text).

2. Local Variable Definitions

The variable table lists each variable's usage (input, output, or internal) and properties (data type, etc.). For details on local variables, refer to *2-3-2 Restrictions in Variables in Function Blocks*.

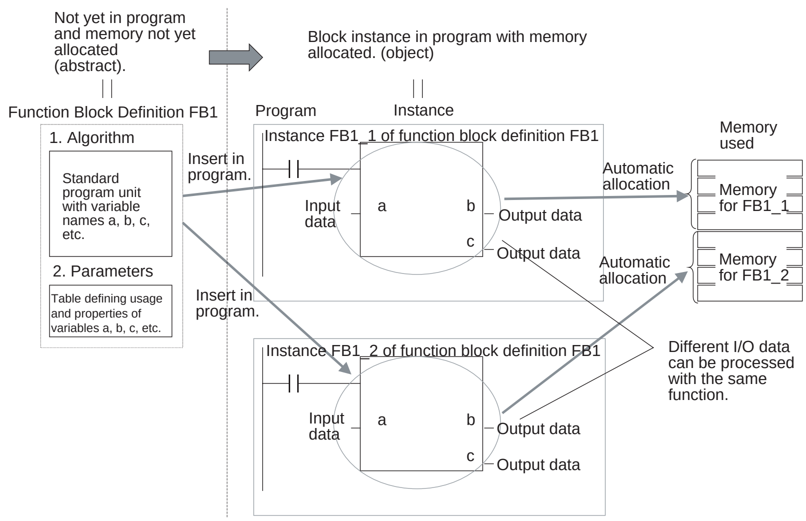
■ Number of Function Block Definitions

Up to 1,024 function block definitions can be created for one PLC (CPU Unit).

Instances

When a function block definition is inserted in a program, the function block uses a particular memory region for its local variables. Each function block definition that is inserted in the program and allocated a different memory region is called an "instance." Each instance is assigned an identifier called an "instance name."

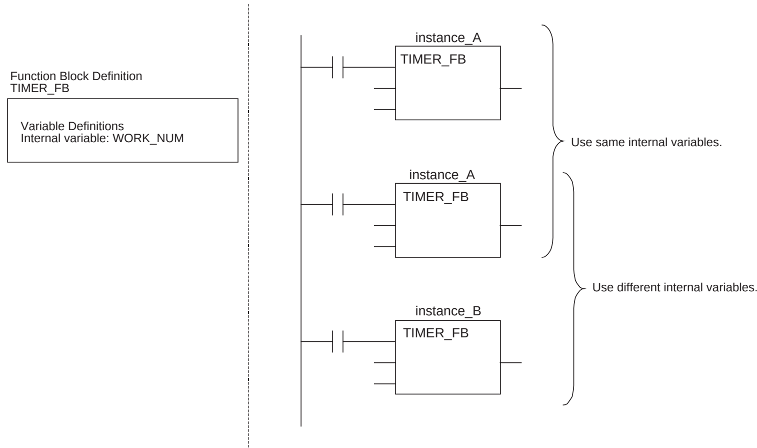
By generating instances, a single function block definition can be used to process different I/O data with the same function.



Note Instances are managed by names. More than one instance with the same name can also be inserted in the program. If two or more instances have the same name and in the same POU, they will use the same internal variables (VAR). If the instances are in different POUs, they will use different internal variables.

For example, if a function block that uses a timer as an internal variable (VAR) is inserted at several points in the POU, each instance must be given a different name. If two or more of the instances have the same name, the timer would be duplicated, which is not allowed.

If, however, internal variables are not used or they are used only temporarily and initialized the next time an instance is executed, the same instance name can be used to save memory.



■ Creating Instances

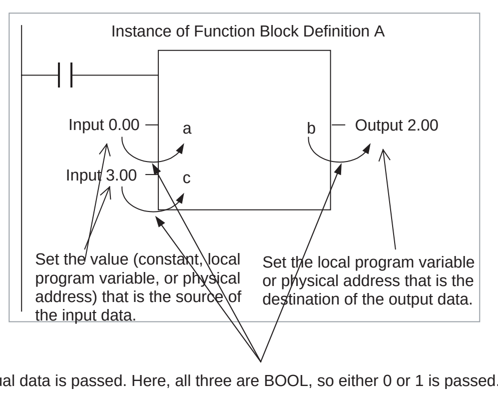
Instances can be created just by dragging and dropping the function block icon. Drag the function block icon from the logical POU in the NE Programmer workspace and drop it in the program.

■ Number of Instances

Multiple instances can be created from a single function block definition. Up to 1,024 instances can be created for a single PLC (CPU Unit). The allowed number of instances is not related to the number of function block definitions or the number of tasks in which the instances are inserted.

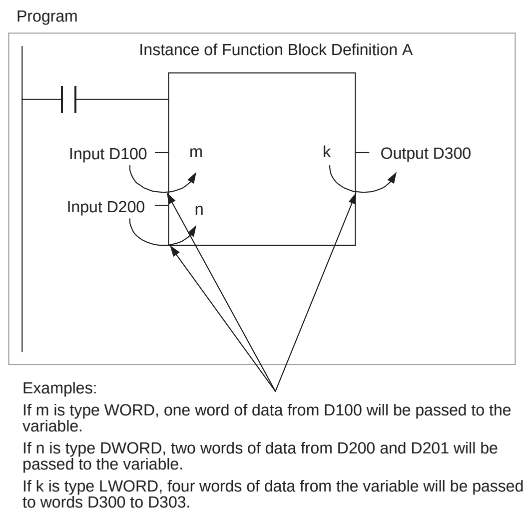
Parameters

Each time an instance is created, set the values (constants, local program variables, or physical I/O memory addresses) used to exchange data with the function block's local I/O variables. These values are called parameters.



The data passed from the input parameter to the function block is not the source address, but the data within the source address (with the data format and size specified by the variable's data type). In a similar fashion, the data passed from the function block to the output parameter is not an address, but the actual data for the output address (with the data format and size specified by the variable's data type).

Even if the input parameter or output parameter is a word address, the data that is passed will be the data (with the format and size specified by the variable's data type) starting from the specified word address.



Note

(1) Only addresses in the following data areas can be used as parameters: CIO Area, Auxiliary Area, DM Area, EM Area (bank 0), Holding Area, and Work Area.
The following cannot be used:
a) Index and Data Registers (neither direct nor indirect specification)
b) Indirect addressing of DM or EM Area (neither binary nor BCD mode)

(2) Local program variables can also be specified as parameters, provided that the local program variable's data size is the same as the local variable's data size.

■ Inputting Parameters

1,2,3...

1. Select ***Mode - Variable mode*** from the *Ladder* Menu and click the **AB** Button.
2. The cursor will change to a cursor with “ABC” and an arrow.
3. Click the left side for an input variable or the right side for an output variable.
4. Input a constant (such as a decimal number after 10#), a local program variable name, or a physical address.

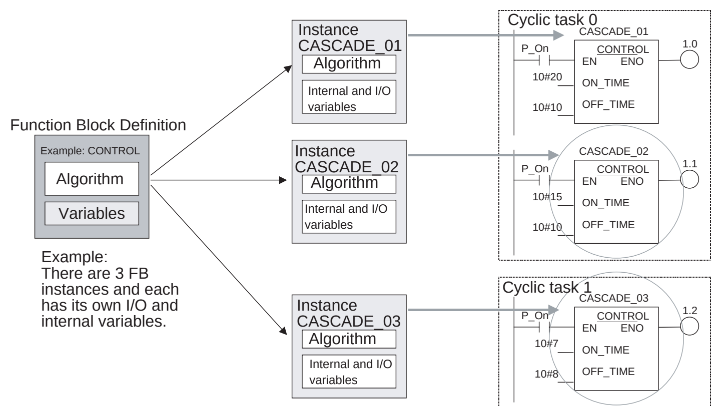
⚠ Caution Input values are passed from parameters to the input variables (VAR_INPUT) when an instance is executed but before the algorithm is processed. Consequently, a parameter cannot be used to pass a value if it is necessary to read the value during processing (within the execution cycle of the algorithm). Instead, directly input a physical address to pass the value.

Likewise, output values (VAR_OUTPUT) are passed from output variables to parameters just after the algorithm is processed. Consequently, a parameter cannot be used to pass a value if it is necessary to write the value during processing (within the execution cycle of the algorithm). Instead, directly input a physical address to pass the value.

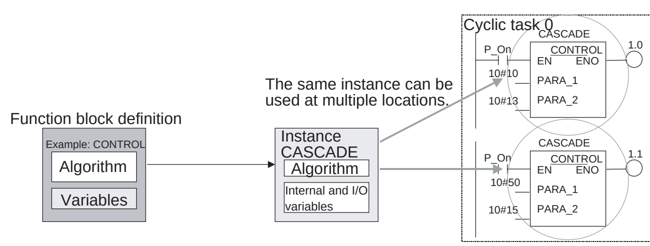
■ Reference Information

A variety of processes can be created easily from a single function block by using parameter-like elements (such as fixed values) as input variables (VAR_INPUT) and changing the values passed to the input variables for each instance.

Example: Creating 3 Instances from a Single Function Block Definition



The same instance name can be used at multiple locations in the program if internal variables (VAR) are not used or processing will not be affected even if internal variables (VAR) are used in the multiple locations.



In this case, the same memory area will be used, so some precautions are required. For example, if an instance containing a timer instruction is used in more than one program location, the same timer number will be used causing coil duplication, and the timers will not function properly if both instructions operate at the same time.

2-3-2 Restrictions in Variables in Function Blocks

The following table shows the number of local variables that can be used and the kind of variable that is created by default for each of the variable usages.

Variable usage	Allowed number	Variable created by default
Input (VAR_INPUT)	Up to 64 per function block (not including EN)	EN (Enable): Receives an input condition. The instance is executed when the variable is ON. The instance is not executed when the variable is OFF.
Output (VAR_OUTPUT)	Up to 64 per function block (not including ENO)	ENO (Enable Output): Outputs the function block's execution status. The variable is turned ON when the instance starts being executed. It can be turned OFF by the algorithm. The variable remains OFF when the instance is not executed.
Internal (VAR)	Unlimited	FI: A local variable that turns ON the first time the instance is executed. (It can be used for initialization the first time an instance is executed.)
External variables	---	System variables

Initial Values of Input Variables (VAR_INPUT) in Function Blocks

When you set an initial value for an input variable (VAR_INPUT), that value will be written to the variable when the parameter for input variable EN goes ON and the instance is executed for the first time (and that one time only). If an initial value has not been set for an input variable, the input variable will be set to 0 when the instance is first executed.

Initial Values of Output Variables (VAR_OUTPUT) in Function Blocks

An initial value can be set for an output variable (VAR_OUTPUT) that is not being retained, i.e., when the Retain Option is not selected. An initial value cannot be set for an output variable if the Retain Option is selected.
The initial value will not be written to the output variable if the IOM Hold Bit (A50012) is ON.

Auxiliary Area control bit		Initial value
IOM Hold Bit (A50012)	ON	The initial value will not be set.

Initial Values of Internal Variables (VAR) in Function Blocks

An initial value can be set for an internal variable (VAR) that is not being retained (i.e., when the Retain Option not selected). An initial value cannot be set for an internal variable if the Retain Option is selected. Internal variables that are not being retained will be initialized to 0.

The initial value will be set, regardless of the value of the IOM Hold Bit (A50012).

2-3-3 Function Block Specifications

Function Block Specifications

Item	Description
Number of function block definitions	CPU01: 1,024 max. per CPU Unit
Number of instances	CPU01: 1,024 max. per CPU Unit
Number of instance nesting levels	8 levels max.
Number of input variables (VAR_INPUT) per function block	64 variables max.
Number of output variables (VAR_OUTPUT) per function block	64 variables max.

Function Block Elements

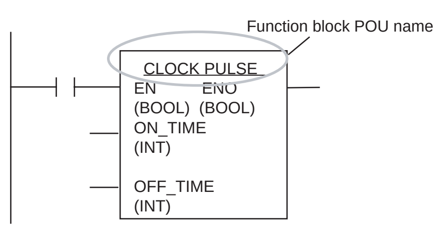
The following table shows the items that must be entered by the user when defining function blocks.

Item	Description
Function block POU name	The name of the function block definition (logical POU name)
Language	The programming language used in the function block definition. Select ladder programming or structured text
Local variable definitions	Variable settings, such as operands and return values, required when the function block is executed • Type (usage) of the variable • Name of the variable • Data type of the variable • Initial value of the variable
Algorithm	Enter the programming logic in ladder or structured text.
Comment	Function blocks can have comments.

Function Block Definition Name

This is the name of a function block, which is one kind of logical POU.

- Number of characters: 24 characters max. for variable or instance names
- Usable characters: Numbers 0 to 9, letters a to z, letters A to Z, and the underscore character
- Upper and lower case characters are distinguished but do not make variables different. For example, “aBc” and “AbC” are treated as the same variable, but the variable “aBc” is recorded as “aBc”.
- Variable names cannot begin with a number (0 to 9).
- There cannot be two or more consecutive underscore characters (_).



Language

Select either ladder or structured text.

Local Variable Definitions

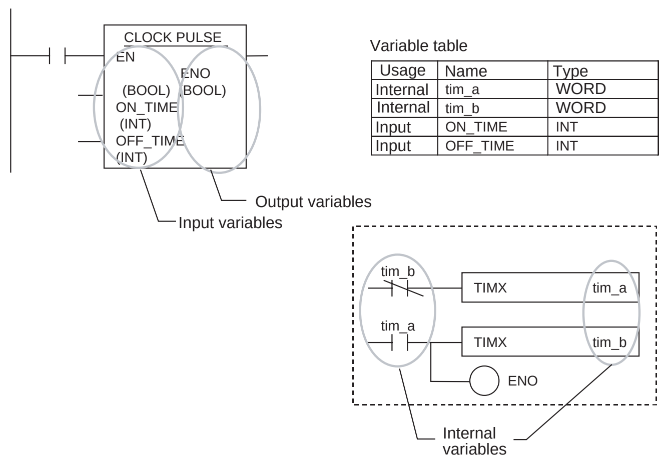
Define the arguments and local variables used in the function block definition.

■ **Local Variable Names**

- Number of characters: 48 characters max.
- Usable characters: Numbers 0 to 9, letters a to z, letters A to Z, and the underscore character

- Upper and lower case characters are distinguished but do not make variables different. For example, “aBc” and “AbC” are treated as the same variable, but the variable “aBc” is recorded as “aBc”.
- Variable names cannot begin with a number (0 to 9).
- There cannot be two or more consecutive underscore characters (_).
- Variable names cannot have the following characters followed by a number, because these combinations are treated as actual data area addresses: A, W, H, T, C, D, E0, @D, *D, @E□, *E□, IR, DR, TK, and TR.
- The actual address expressions of external variables (preassigned variables) cannot be used.

■ Variable Notation



2-3-4 Instance Specifications

Composition of an Instance

The following table lists the items that the user must set when registering an instance.

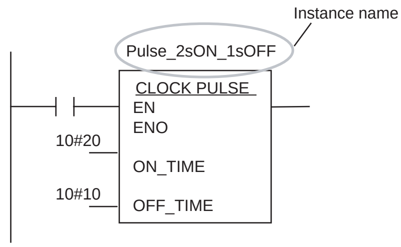
Item		Description
Instance name		Name of the instance
Parameters	Input parameters	Pass data to input variables (constants, local program variables, or physical addresses)
	Output parameters	Receive data from output variables (local program variables or physical addresses)

Instance Name

This is the name of the instance.

- Number of characters: 48 characters max.
- Usable characters: Numbers 0 to 9, letters a to z, letters A to Z, and the underscore character
- Upper and lower case characters are distinguished but do not make variables different. For example, “aBc” and “AbC” are treated as the same variable, but the variable “aBc” is recorded as “aBc”.
- Instance names cannot begin with a number (0 to 9).
- There cannot be two or more consecutive underscore characters (_).

The instance name is displayed above the instance in the diagram.



Function Block Instance Areas

■ Calling an Instance from Multiple Locations

A single instance can be called from multiple locations. In this case, the internal variables (VAR) will be shared.

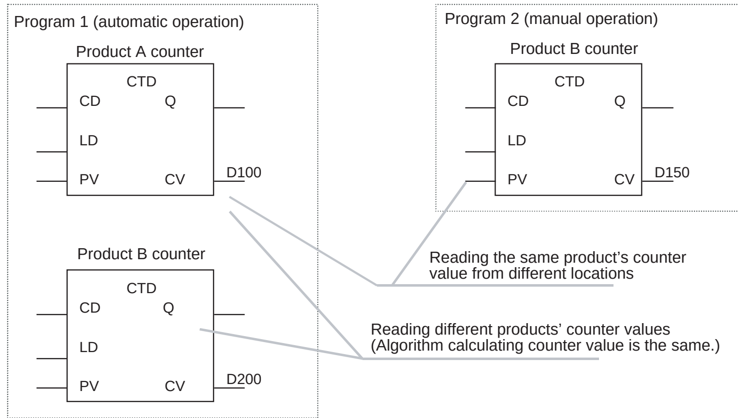
■ Making Multiple Instances

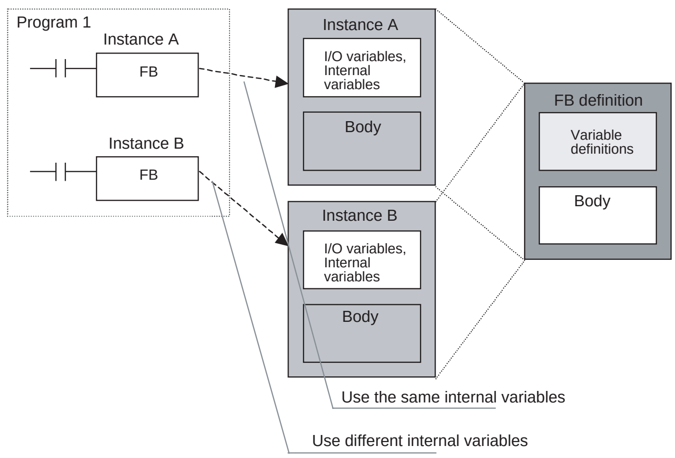
Multiple instances can be created from a single function block definition. In this case, the values of internal variables (VAR) will be different in each instance.

Example: Counting Product A and Product B

Prepare a function block definition called Down Counter (CTD) and set up counters for product A and product B. There are two types of programs, one for automatic operation and another for manual operation. The user can switch to the appropriate mode of operation.

In this case, multiple instances will be created from a single function block. The same instance must be accessible from multiple locations.

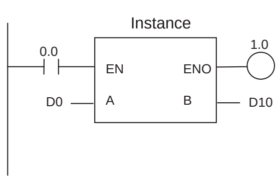




Operating Specifications

Calling Instances

The user can call an instance from any location. The instance will be executed when the input to EN is ON.

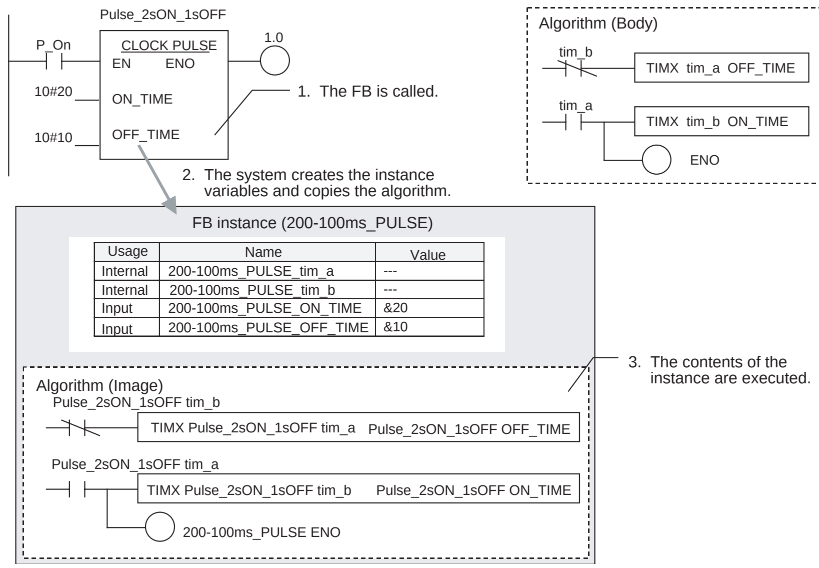


In this case, the input to EN is bit 0.0 at the left of the diagram.

- When the input to EN is ON, the instance is executed and the execution results are reflected in bit 1.0 and word D10.
- When the input to EN is OFF, the instance is not executed, bit 1.0 is turned OFF, and the content of D10 is not changed.

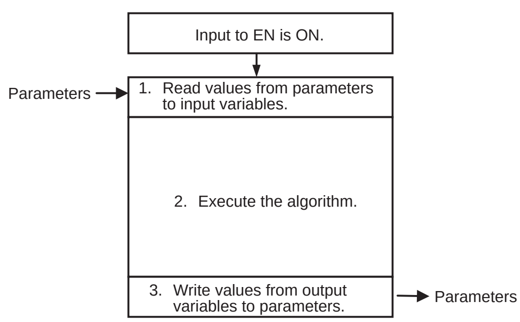
Operation when the Instance Is Executed

The system calls a function block when the input to the function block's EN input variable is ON. When the function block is called, the system creates the instance's local variables and copies the algorithm registered in the function block. The instance is then executed.



The order of execution is as follows:

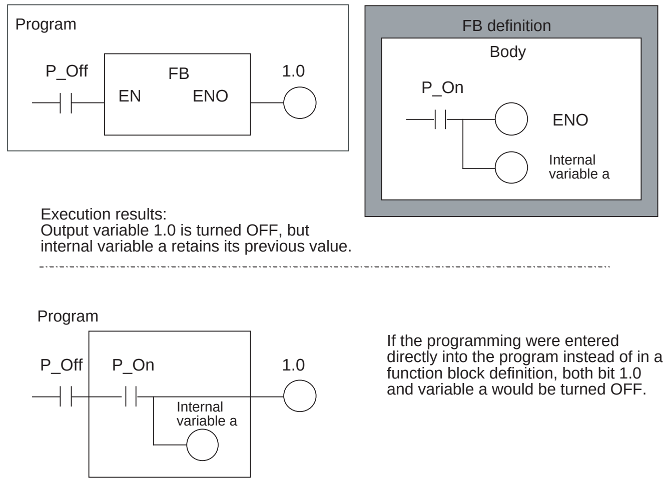
1. Read data from parameters to input variables (VAR_INPUT).
2. Execute the algorithm.
3. Write data from output variables (VAR_OUTPUT) to parameters.



Note Data cannot be exchanged with parameters while the algorithm is being executed. In addition, if an output variable is not changed by the execution of the algorithm, the output parameter will retain its previous value.

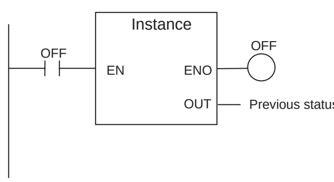
Operation when the Instance Is Not Executed

When the input to the function block's EN input variable is OFF, the function block is not called, so the internal variables of the instance do not change.



Output Variables when a FB Is Not Executed

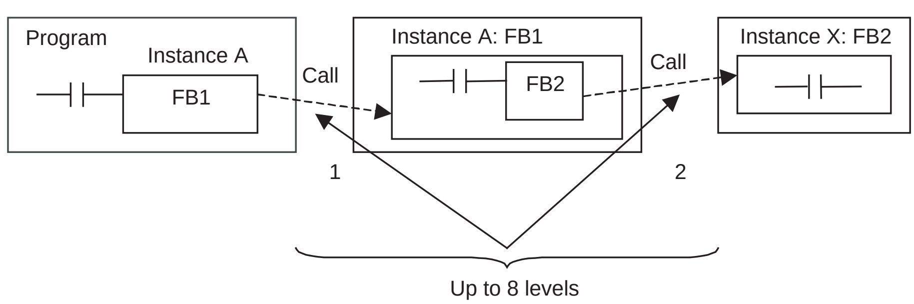
NEO will have the status of the power flow (P.F.: execution condition). User-defined output variables will retain their previous status.



⚠ Caution An instance will not be executed while its EN input variable is OFF, so Differentiation and Timer instructions will not be initialized while EN is OFF. If Differentiation or Timer instructions are being used, use the Always ON Flag (P_On) for the EN input condition and include the instruction's input condition within the function block definition.

Nesting

A function block can be called from another function block, i.e., nesting is supported (up to 8 levels).

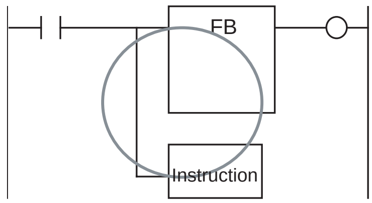


Placement of Instances in Program

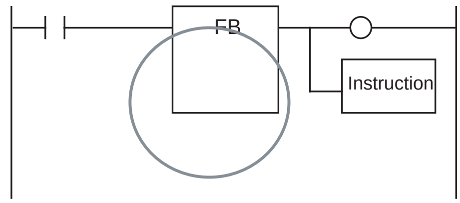
Branches Allowed on the Left and Right Sides of the Instance

Branches can be placed on the left side of the instance (EN input side) as well as the right side of the instance (to the left of ENO).

Correct



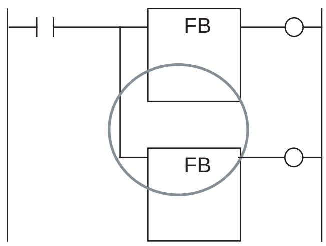
Correct



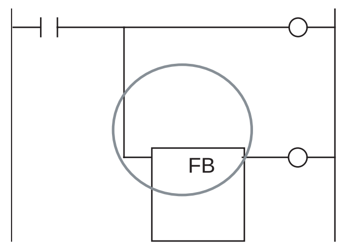
Multiple Instances in a Single Rung

A single program rung can have more than one instance.

Correct



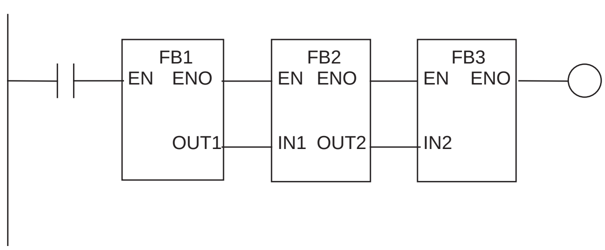
Correct



Connections between Instances are Allowed

An instance's ENO output can be input directly to another instance's EN input. Furthermore, an instance's output parameter can be input directly to another instance's input parameter.

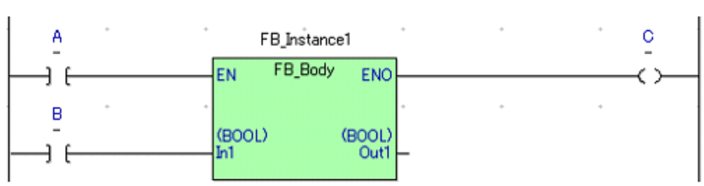
Note The data type of the input parameter must match the data type of the output parameter. As long as the data types match, other data types can be used besides BOOL-BOOL, including WORD-WORD and DWORD-DWORD.



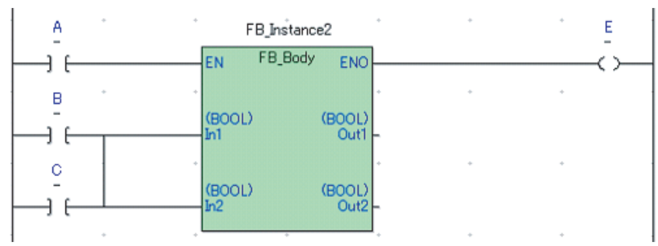
Up to 4 instances can be connected between the left and right bus bars in the ladder diagram. If more than 4 are entered, the additional instructions will wrap around. There can be up to 150 wrap-around lines per program section. Function block connections are allowed within that range.

AND and OR Conditions Allowed on Input Side

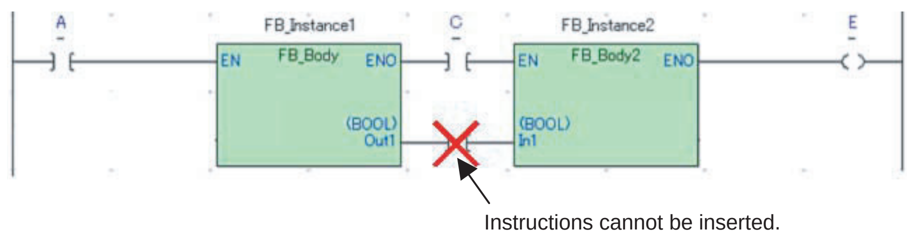
AND or OR conditions can be entered at the instance's left side (either the EN input side or the input parameter side).



Input parameters can also be connected together.



An AND or OR condition can be inserted between function blocks, as shown in the following diagram. On the other hand, instructions cannot be inserted between parameters, as explained in *Restrictions on Placement of Instances*.

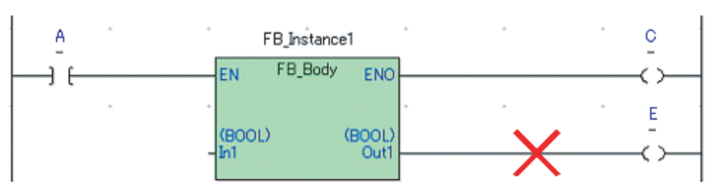


Restrictions on Placement of Instances

There are some restrictions on the placement of instances in the program.

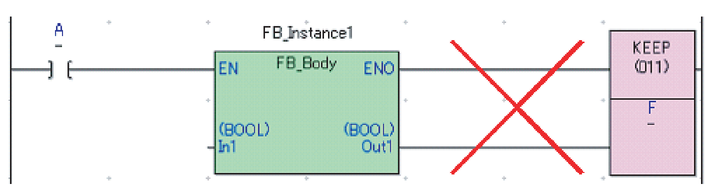
Connecting an Output Bit to an Output Parameter

An output bit cannot be connected to an output parameter.



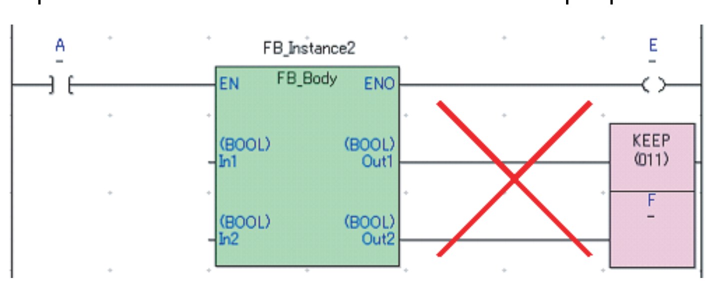
Connecting an Instruction to an ENO Output and Output Parameter

A special instruction cannot be connected to an ENO output and output parameter.



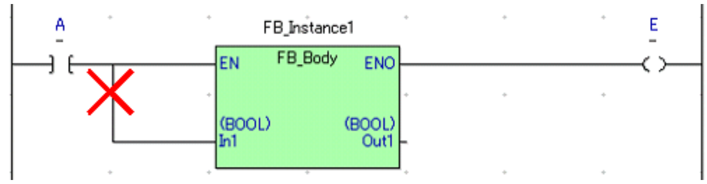
Connecting an Instruction to Output Parameters

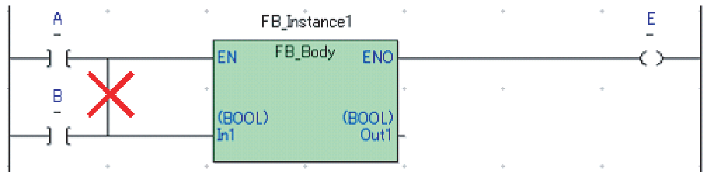
A special instruction cannot be connected to output parameters.



Connecting an EN Input with an Input Parameter

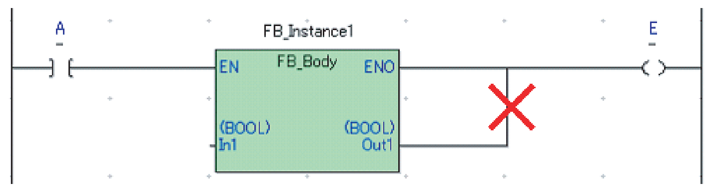
A EN input cannot be connected to an input parameter.

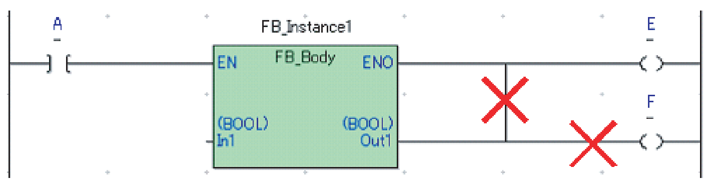




Connecting an ENO Output with an Output Parameter

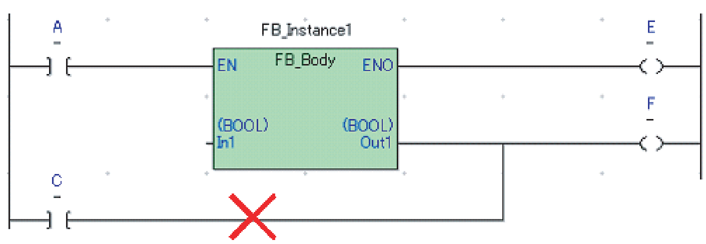
A ENO output cannot be connected to an output parameter.





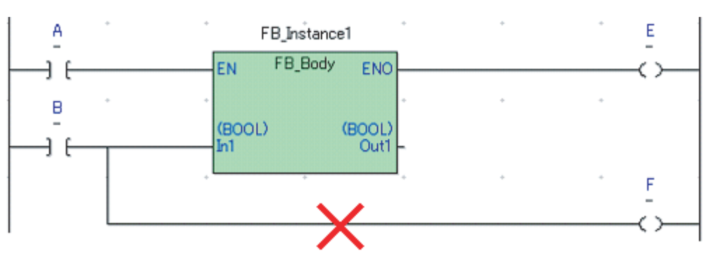
Connecting an Input to an Output Parameter

An input cannot be connected directly to an output parameter.



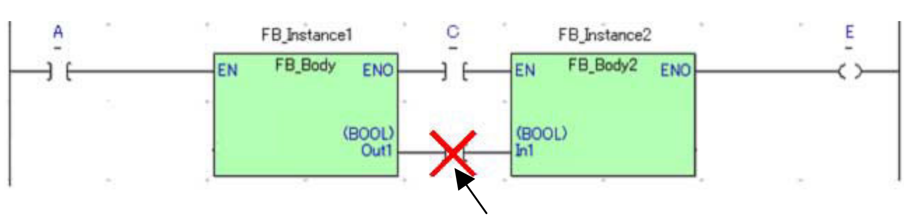
Connecting an Input Parameter to an Output Bit

An input parameter cannot be connected directly to an output.



Inserting an AND or OR Condition between Parameters

An AND or OR condition cannot be inserted between parameters.



Instructions cannot be inserted.

Parameter Specifications

Allowed Parameter Inputs

The following data can be set in input and output parameters.

Input Parameters:

- Constants (See note.)
- Local program variables (for example: aa)
  (The data type must match the corresponding input variable.)
- Physical addresses (for example: D00100 or 0000.00)

Output Parameters:

- Local program variables (for example: aa)
  (The data type must match the corresponding output variable.)
- Physical addresses (for example: D00100 or 0000.00)

Note The input range of the constant depends on the data type, as shown in the following table.

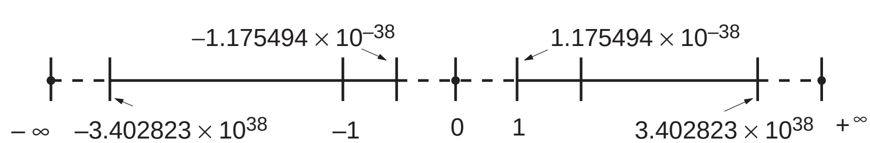
Input the desired value in the *Initial Value* Field in the *Edit Variables* Dialog Box. Input a value consistent with the variable's data type, as shown in the following table.

Data type	Content	Range of values	Initial value input method
BOOL	Bit data	16#0 or 16#1	Input as unsigned hexadecimal after “16#”.
INT	Integer	–10#32768 to +10#32767	Input as signed decimal after “+10#” or “–10#”. (If “10#” is input, a + sign will be attached automatically.)
UINT	Unsigned integer	10#0 to 10#65535	Input as unsigned decimal after “10#”.
DINT	Double integer	–10#2147483648 to +10#2147483647	Input as signed decimal after “+10#” or “–10#”. (If “10#” is input, a + sign will be attached automatically.)
UDINT	Unsigned double integer	10#0 to 10#4294967295	Input as unsigned decimal after “10#”.
WORD	16-bit data	16#0000 to16#FFFF	Input as unsigned hexadecimal after “16#”.
DWORD	32-bit data	16#00000000 to16#FFFFFFFF	Input as unsigned hexadecimal after “16#”.
REAL	Real number	Refer to the details on floating-point data below.	Input a signed numerical value. Input the value after the sign. E.g., +1.0, –0.23, +9.87E-3
STRING	Character string data (ASCII data)	127 characters	---

Floating-point Data

The following data can be expressed by floating-point data:

- $-\infty$
- $-3.402823 \times 10^{38} \leq$ value $\leq -1.175494 \times 10^{-38}$
- 0
- $1.175494 \times 10^{-38} \leq$ value $\leq 3.402823 \times 10^{38}$
- $+\infty$
- Not a number (NaN)



Special Numbers

The formats for NaN, ±∞, and 0 are as follows:

NaN*: e = 255, f ≠ 0
+∞: e = 255, f = 0, s = 0
–∞: e = 255, f = 0, s = 1
0: e = 0

*NaN (not a number) is not a valid floating-point number. Executing floating-point calculation instructions will not result in NaN.

Restrictions on Variable Data Types

Parameters cannot be input to TIMER or COUNTER variables or output from TIMER or COUNTER variables.

Timing of Data Transfers to/from Variables

Input data is passed from parameters to the corresponding input variables before the algorithm is processed. Output variable data is passed to the corresponding parameters after algorithm processing is completed. Consequently, values cannot be passed from parameters to input variables while the algorithm is being executed.

Restrictions on Parameter Data Quantity

The maximum amount of a parameter I/O data is 1,024 words per instance.

Connecting a Bit Control Instruction to a Parameter

For details on connecting a parameter to a Bit Control Instruction, refer to the sections *Placement of Instances in Program* and *Restrictions on Placement of Instances* above.

2-3-5 Restrictions on Function Blocks

Instructions Restricted in Ladder Programs

Instructions Prohibited in Function Block Definitions

Instructions that can be used in the program can also be used in function blocks.

I/O Variable Restrictions (Unsupported Data Areas)

Addresses in the following data areas cannot be used as parameters for input and output variables.

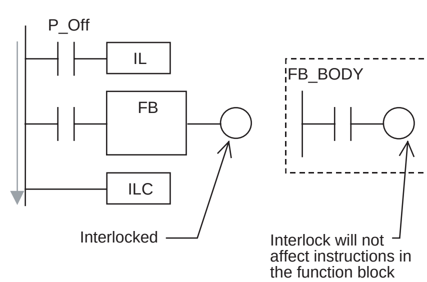
- Index Registers and Data Registers (Neither indirect nor direct addressing is supported.)
- Indirect addressing of DM or EM Area addresses (Neither binary-mode nor BCD-mode indirect addressing is supported.)

Refreshing Timer and Counter PVs

Timer and counter PVs are always stored in binary mode, so PVs of all Timer and Counter Instructions must be treated as binary data whether or not the instructions are in function blocks.

Interlocks

When a function block is called from an interlocked program section, the contents of the function block definition will not be executed. The interlocked function block will behave just like an interlocked subroutine.

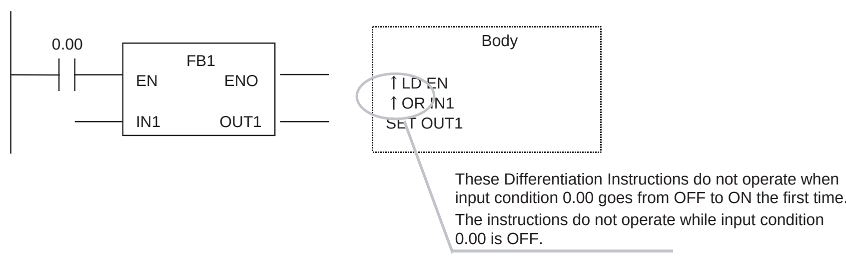


Differentiation Instructions in Function Block Definitions

An instance will not be executed while its EN input variable is OFF, so the following precautions are essential when using a Differentiation Instruction in a function block definition. (Differentiation Instructions include DIFU, DIFD, and any instruction with an @ or % prefix.)

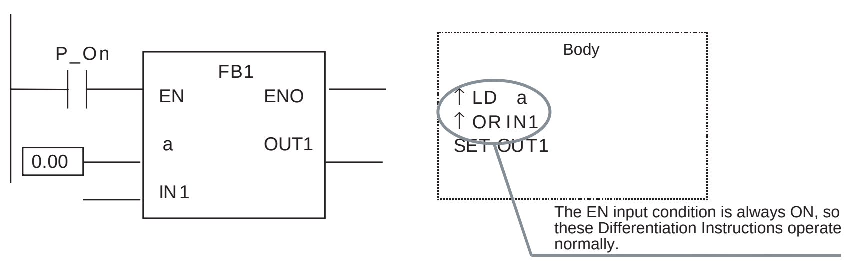
- As long as the instance's EN input variable is OFF, the execution condition will retain its previous status (the last status when the EN input variable was ON) and the Differentiation Instruction will not operate.
- When the instance's EN input variable goes ON, the present execution condition status will not be compared to the last cycle's status. The present execution condition will be compared to the last condition when the EN input variable was ON, so the Differentiation Instruction will not operate properly. (If the EN input variable remains ON, the Differentiation Instruction will operate properly when the next rising edge or falling edge occurs.)

Example:



If Differentiation Instructions are being used, always use the Always ON Flag (P_On) for the EN input condition and include the instruction's input condition within the function block definition.

Example:

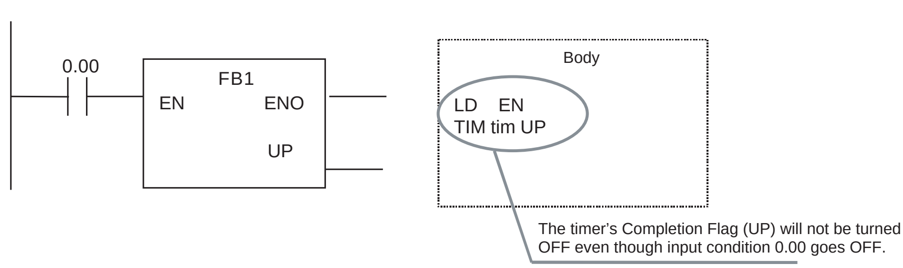


Timer Instructions in Function Block Definitions

An instance will not be executed while its EN input variable is OFF, so the following precautions are essential when using a Timer Instruction in a function block definition.

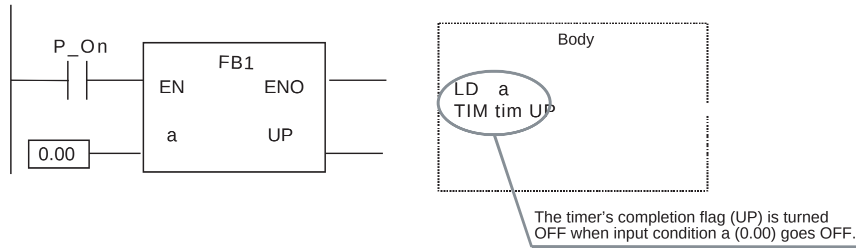
The Timer Instruction will not be initialized even though the instance's EN input variable goes OFF. Consequently, the timer's Completion Flag will not be turned OFF if the EN input variable goes OFF after the timer started operating.

Example:



If Timer Instructions are being used in the function block definition, always use the Always ON Flag (P_On) for the EN input condition and include the instruction's input condition within the function block definition.

Example:



- If the same instance containing a timer is used in multiple locations at the same time, the timer will be duplicated.

ST Programming Restrictions

- Only the following statements and operators are supported.
  - Assignment statements
  - Selection statements (CASE and IF statements)
  - Iteration statements (FOR, WHILE, and REPEAT statements)
  - Arithmetic operators
  - Logical operators
  - Comparison operators
  - Comments
- The TIMER and COUNTER data types cannot be used.
- Use parentheses to indicate the priority of arithmetic operations.
  Example: D:= (A+B) *C
- Tabs and spaces can be used to indent text.

Online Editing Restrictions

The following online editing operations cannot be performed on the user program in the CPU Unit.

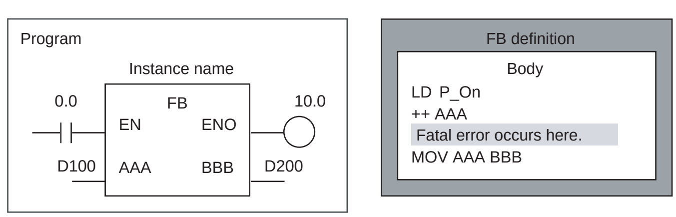
- I/O variables cannot be added, deleted, or changed.
- Internal variables cannot be deleted or changed.
- FB body names cannot be changed.
- FB instance names cannot be changed.
- FB instances cannot be added.
- Comments cannot be edited.

Note (1) Internal variables can be added, but there are restrictions on the variables that can be added.

(2) Global variables can be added.

Error-related Restrictions

If a fatal error occurs in the CPU Unit while a function block definition is being executed, ladder program execution will stop at the point where the error occurred.



In this case, the MOV AAA BBB instruction will not be executed and output variable D200 will retain the same value that it had before the function block was executed.

2-4 Programming Notes

2-4-1 Tasks

Tasks control the timing of program execution. Logical POU programs are assigned to tasks to execute them.

There are two types of tasks: cyclic tasks and interrupt tasks (power OFF interrupt, scheduled interrupt, I/O interrupt, and external interrupt tasks). Extra cyclic tasks are not supported.

Up to 128 cyclic tasks (task numbers 0 to 127) can be used. When specifying the task number in the operand of a TASK ON (TKON) or TASK OFF (TKOF) instruction, the task number can be specified directly by inputting the number in decimal after “10#” or it can be specified indirectly by inputting an I/O memory address containing the task number.

Example: When the instruction “TKON 3” is input, the task number is the value in CIO 0003.

2-4-2 Inputting Constants

When inputting a constant in the following cases, always include the prefix code corresponding to the data type (16#, 10#, or -10# for integers, or decimal point or □E□ for real numbers).

- Inputting an instruction operand (other than timer/counter number inputs)
- Inputting I/O memory data
- Inputting a constant for an initial value
- Inputting function block parameters

Unsigned Hexadecimal: Input 16#

Input the prefix 16# before a hexadecimal constant.
For example, input “16#1A” to input the hexadecimal value 1A.

Positive or Zero Decimal: Input 10#

Input the prefix 10# before a positive decimal constant (or 0).
For example, input “10#123” to input the decimal value 123.

Negative Decimal: Input -10#

Input the prefix -10# before a negative decimal constant.
For example, input “-10#123” to input the decimal value -123.

Note A space can be inserted after the 16#, 10#, or -10# prefix code.
For example, the input “16# 1A” is acceptable.

Real Number: Decimal or □E□

Real numbers can be input with decimal points or as □E□. For example, the input “-0.123” or “-1.23E-1” is acceptable.

Binary (ST Program Only): Input 2#

Binary numbers can be input in ST language only. Input the prefix 2# before a binary constant.
For example, input “2#010” to input the binary value 010.

Octal (ST Program Only): Input 8#

Octal numbers can be input in ST language only. Input the prefix 8# before an octal constant.
For example, input “8#123” to input the octal value 123.

⚠ Caution If a number is input without a 16#, 10#, or -10# prefix code, it is treated as a word address in the CIO Area.
For example, an input of “100” specifies CIO 0100 in the CIO Area.

2-4-3 Inputting Operand Numbers

Inputting Timer/Counter Numbers

Input timer and counter numbers (and only these numbers) as a plain decimal value without a 10# prefix code. (A prefix such as 16# or 10# cannot be input.)

For example, to enter a timer with timer number 2 and decimal SV of 200, input TIMX 2 10#200.

Inputting Jump Numbers

Always input jump numbers with a 16# or 10# prefix code. If the prefix is omitted, the number indicates a CIO Area word address and the content of that address is used as the jump number.

For example, to enter a JMP instruction with jump number 2, input JMP 10#2 or JMP 16#2. (If JMP 2 is input, the content of CIO 0002 is used as the jump number.)

Inputting Task Numbers

Always input a decimal task numbers with the 10# prefix code. (A prefix code of 16# cannot be used.) If the prefix is omitted, the number indicates a CIO Area word address and the content of that address is used as the task number.

For example, to enter a TASK ON instruction with task number 3, input TKON 10#3. (If TKON 3 is input, the content of CIO 0003 is used as the task number.)

2-4-4 Inputting a Network Path

The network path indicates the path from the local CPU Unit to another PLC in the network. The NE1S-series PLCs can use the network path for routing.

When a Network Path is Required

A network path can be specified in the following case.

- SEND CIP COMMAND Instruction (CSND)
  When reading or writing the I/O memory of another node in the network with the SEND CIP COMMAND Instruction (CSND), the network path can be specified in the instruction’s control data.

The I/O memory of another PLC in the network can be accessed through a variable by specifying the network path and variable name.

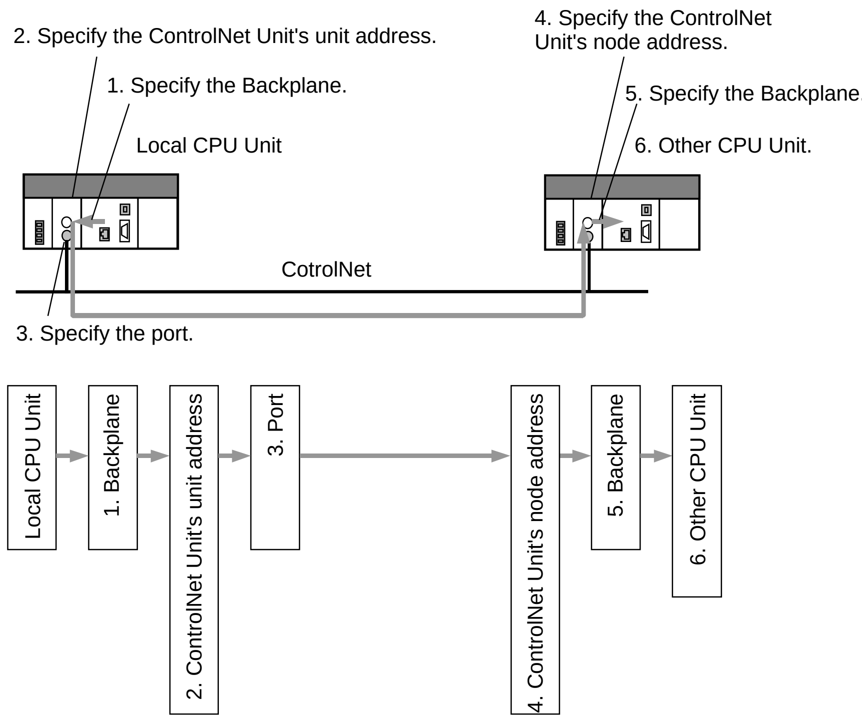
Specifying the Network Path

Specify the network as shown in the following table.

Target device	Specifying the network path for a SEND CIP COMMAND Instruction (CSND)
Backplane	01 Hex
Specifying a Unit mounted to the local CPU Unit	Specify the unit address in hexadecimal. For a CPU Bus Unit, specify the unit number + 10 Hex.
Specifying the network port	Port A: 01 Hex Port B: 02 Hex, etc.
Specifying a Unit in the network	Specify the network node address in hexadecimal.
Specifying the other PLC’s CPU Unit	00 Hex (unit address)

Example:

Specifying the path from the local CPU Unit to a local variable in another CPU Unit in a ControlNet Network



Example		Specifying the network path for a SEND CIP COMMAND Instruction (CSND)
Specification method	1. Backplane	16#01
	2. ControlNet Unit's unit address (Example: Unit address = 3)	16#13
	3. Port (Example: Port = Port B)	16#02
	4. ControlNet Unit's node address (Example: Node address = 5)	16#5
	5. Backplane	16#01
	6. Other CPU Unit	16#00

SECTION 3
CPU Unit

This section describes hardware and software aspects of the CPU Unit.

3-1 CPU Unit

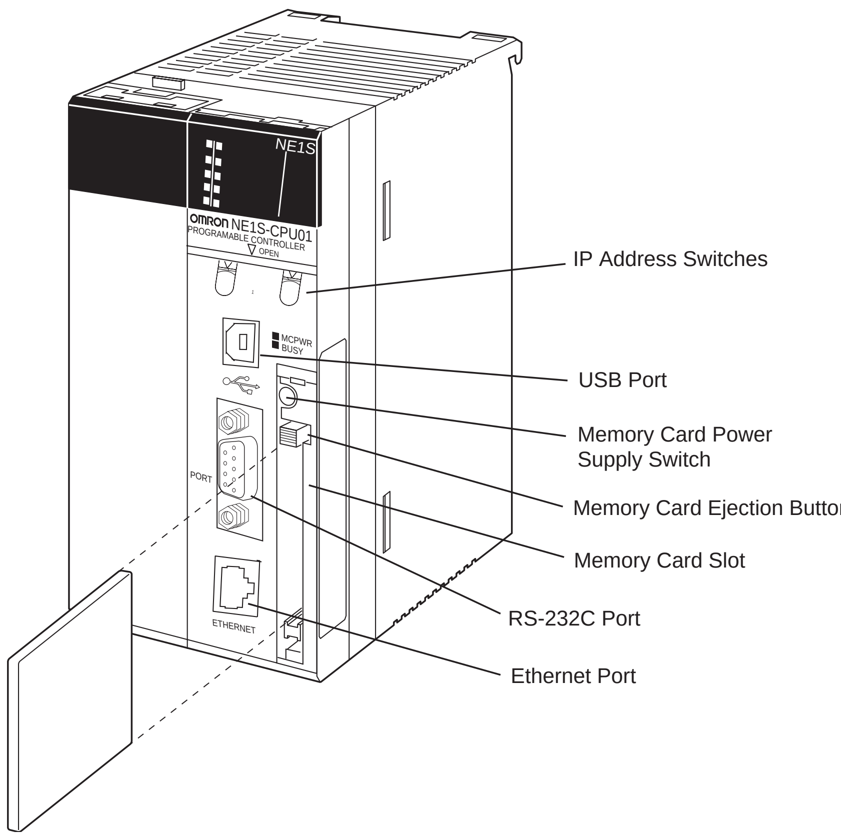
3-1-1 Model

NE1S CPU Unit

Model	Expansion Racks	I/O points	Programming	Data Memory (DM + EM)	LD instruction processing time
NE1S-CPU01	7 max. (Long-distance Expansion Racks cannot be connected.)	5,120	250K steps (including portion automatically generated by Programming Device)	64K words	0.019 µs

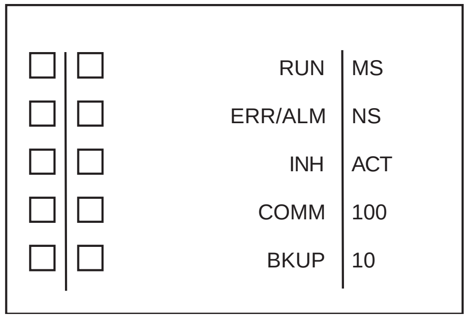
3-1-2 Component Names and Functions

Component Names



Indicators

The following table describes the LED indicators located on the front panel of the CPU Unit.



Indicator	Color	Status	Meaning
RUN	Green	ON	CPU Unit is executing the program in MONITOR or RUN mode.
		Flashing	DIP switch setting error
		OFF	PLC has stopped operating while in PROGRAM mode, or has stopped operating due to a fatal error.
ERR/ALM	Red	ON	A fatal error has occurred (including FALS instruction execution), or a hardware error (watchdog timer error) has occurred. The CPU Unit stops operating, and the outputs from all Output Units will be turned OFF.
		Flashing	A non-fatal error has occurred (including FAL instruction execution) The CPU Unit will continue operating.
		OFF	CPU Unit is operating normally.
INH	Yellow	ON	Output OFF Bit (A50015) is ON. The outputs from all Output Units will be OFF.
		OFF	Output OFF Bit (A50015) is OFF.
COMM	Yellow	Flashing	CPU Unit is communicating (sending or receiving) via the RS-232C port.
		OFF	CPU Unit is not communicating via the RS-232C port.
BKUP	Yellow	ON	User program, program information, variable information, and parameter area data is being backed up to built-in flash memory in the CPU Unit or being restored from flash memory (as specified when turning ON the PLC power supply). **Note** Do not turn OFF the power supply to the PLC while this indicator is lit.
		OFF	Data is not being written to flash memory.
MS (CPU Unit error indicator)	Green	ON	CPU Unit is operating normally.
	Red	ON	A serious error has occurred in the CPU Unit. (The ERR/ALM indicator will light if the error is a WDT error.)
		Flashing	A minor error has occurred in the CPU Unit. (The ERR/ALM indicator will flash.)
NS (Network status indicator)	Green	ON	A CIP connection established.
		Flashing	No CIP connection established.
	Red	ON	IP address duplication
		Flashing	CIP I/O connection error (not supported)
ACT	Yellow	ON	Ethernet communications are in progress.
		OFF	Ethernet communications are not in progress.
100	Orange	ON	100Base-TX Ethernet link established (100 Mbps).
		OFF	No 100Base-TX Ethernet link established.
10	Orange	ON	10Base-TX Ethernet link established (10 Mbps).
		OFF	No 10Base-TX Ethernet link established.
MCPWR (Memory Card power indicator)	Green	Flashing	Power is being supplied to the Memory Card.
		OFF	Power is not being supplied to the Memory Card.

Indicator	Color	Status	Meaning
BUSY	Yellow	Flashing	Memory Card is being accessed.
		OFF	Memory Card is not being accessed.

DIP Switch Settings

Pin no.	Setting	Function	Usage	Default
1	ON	Writing disabled for user program memory. (See note 1.)	Used to prevent programs from being accidently overwritten from NE Programmer.	OFF
	OFF	Writing enabled for user program memory.		
2	Not used. Keep set to OFF.			OFF
3	Not used. Keep set to OFF.			OFF
4	ON	USB port communications parameters set in the PLC Setup are used.	Turn OFF to use the USB port as a peripheral bus port.	ON
	OFF	USB port communications parameters set for the peripheral bus.		
5	ON	RS-232C port communications parameters set using a Programming Device (peripheral bus only) are used.	Turn ON to use the RS-232C port as a peripheral bus port.	OFF
	OFF	RS-232C port communications parameters set in the PLC Setup are used.		
6	ON	User-defined pin. Turns OFF the User DIP Switch Pin Flag (A39512).	Set pin 6 to ON or OFF and use A39512 in the program to create a user-defined condition without using an I/O Unit.	OFF
	OFF	User-defined pin. Turns ON the User DIP Switch Pin Flag (A39512).		
7	ON (Normally, this pin should be left turned OFF.)	Writing from the CPU Unit to the Memory Card.	Press and hold the Memory Card Power Supply Switch for three seconds.	OFF
		Restoring from the Memory Card to the CPU Unit.	To read from the Memory Card to the CPU Unit, turn ON the PLC power. This operation is given priority over automatic transfer (pin 2 is ON) when power is ON.	
	OFF	Verifying contents of Memory Card.	Press and hold the Memory Card Power Supply Switch for three seconds.	
8	Always OFF	---	---	OFF

Note (1) The following data cannot be overwritten when pin 1 is ON:

- All parts of the user program (programs in all tasks)
- Program information
- Variable information
- All data in the parameter area (such as the PLC Setup, built-in Ethernet setup, and registered I/O tables)

When pin 1 is ON, the user program, program information, variable information, and parameter area will not be cleared when the memory clear operation is performed from NE Programmer.

(2) The CPU Unit will not enter any mode except PROGRAM mode after backing up data to a Memory Card using DIP switch pin 7. To enter RUN or MONITOR mode, turn OFF the power supply, turn OFF pin 7, and then restart the PLC. This will enable changing the operating mode from the NE Programmer.

Rotary Switch Settings

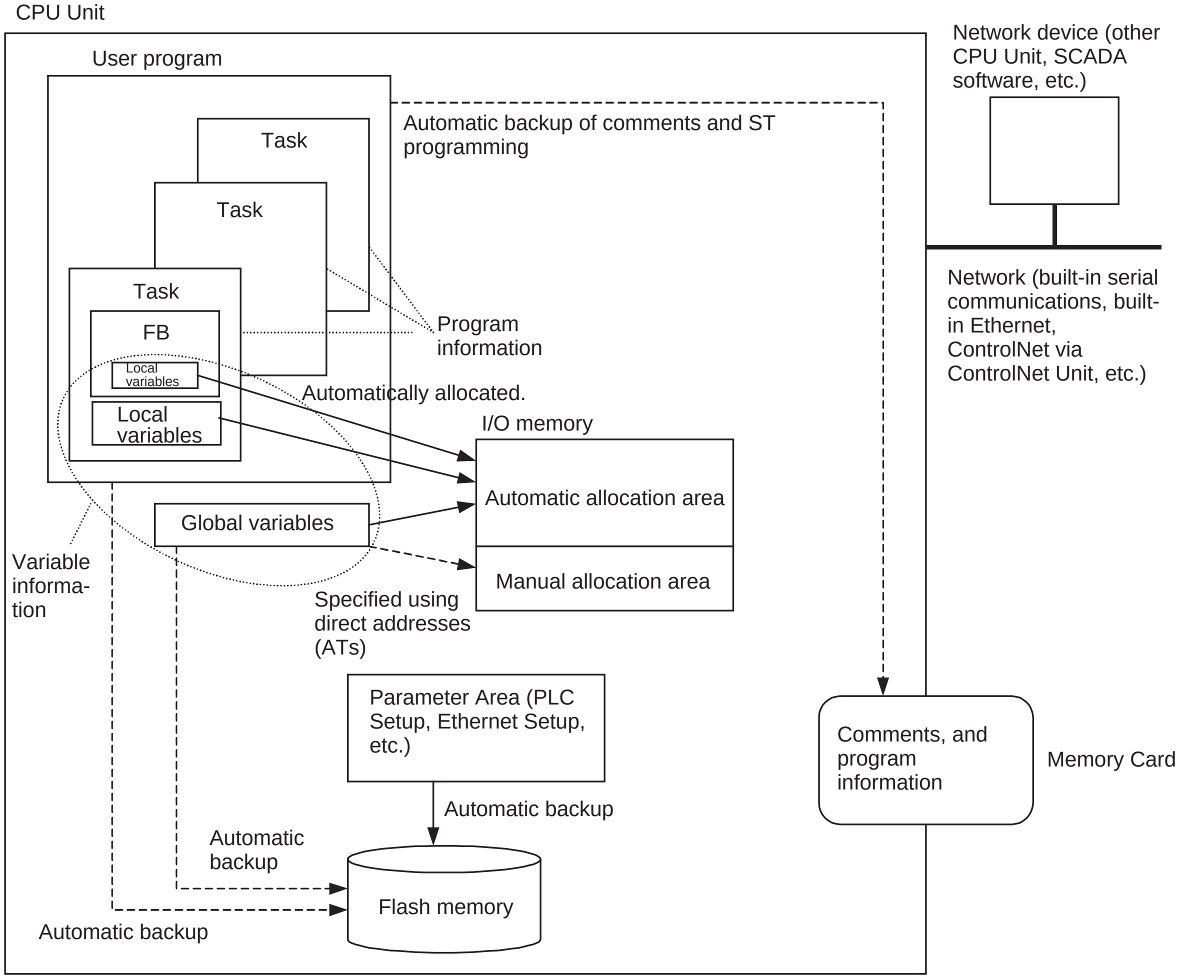
The rotary switch settings are listed in the following table. Any changes to the rotary switch settings or the Ethernet Setup are valid from the next time the CPU Unit is started.

Setting	Meaning
00	• Start with IP address saved in the flash memory in the CPU Unit. • The following default values are stored in flash memory. IP address: 192.168.200.200, network mask: 255.255.255.0, default gateway: 0.0.0.0 (not set). • If the IP address is changed from the Ethernet Setup on NE Programmer and then downloaded tot he CPU Unit, the new IP address will be stored in flash memory and it will be used the next time the CPU Unit is started. **Note** A setting of 00 is particularly convenient when setting and managing the IP addresses for all CPU Units from the NE Programmer.
00 to FE	• Only the rightmost 8 bits (i.e., the host portion) of the IP address of the CPU are changed. • The default IP address will be set to 192.168.200.01 to 192.168.200.254 (network mask: 255.255.255.0, default gateway: 0.0.0.0). **Note** When any of these settings are used, the rightmost 8 bits of the IP address will be ignored when setting the IP address from the Ethernet Setup on NE Programmer. **Note** As a rule, set the rightmost 8 bits on the rotary switch and then set the Ethernet Setup on NE Programmer to the same setting. This will enable reading the IP address off the rotary switch onsite. • The network portion of the IP address is set or changed from the Ethernet Setup on NE Programmer.
FF	The CPU Unit will use BOOTP. Any settings made from the Ethernet Setup on NE Programmer will be ignored if this setting is used.

Note Any changes made to the rotary switch setting or Ethernet Setup are effective the next time the CPU Unit is started.

3-1-3 Internal Workings of the CPU Unit

The internal workings of the CPU Unit are outlined in the following figure.



Overall User Program

The overall user program consists of more than one program and can include up to 384 programs, including interrupt programs. Programs are assigned to tasks to use them. (Function blocks are pasted into a program in advance.)

Instructions in a program are executed in order from the beginning of the program, and I/O memory is written through variables. When all cyclic tasks have been executed, I/O refreshing is performed for the I/O Units. Program execution is then repeated starting from the cyclic task with the smallest number. This process is called a cyclic scan.

Tasks

Tasks determine the timing for program execution. Tasks that are executed once each cycle are called cyclic tasks. Up to 128 cyclic tasks can be used. Cyclic tasks are executed in order of the task numbers. Tasks that are executed when an interrupt condition is met are called interrupt tasks. Up to 256 interrupt tasks can be used.

Program Information

Program information provides file names and other information associated with programs. (This data does not exist in the CS/CJ Series.)

Variable Information

Variable information is automatically backed up in flash memory whenever it is changed by the user.

■ Local Variables

Local variables are maintained in the local variable table for each logic POU (program or function block).

■ Global Variables

Global variables are shared by all programming within one PLC. They are automatically allocated in I/O memory or the physical addresses are specified by the user using direct addressing.

I/O Memory

I/O memory consists to two areas: automatic allocation area, which is allocated automatically for local variables and global variables, and manual allocation area, which is allocated manually using direct addresses for global variables.

■ Automatic Allocation Area

Internally, the system automatically allocates global variables, local variables for programs, and local variables for each function block instance in a reserved portion of the EM (expanded data memory) Area. It is not necessary for the user to consider the physical addresses that are used.

■ Manual Allocation Area

Direct address specifications can be manually made for global variables to specify fixed addresses in the following areas: CIO (core I/O) Area, Work Area, Holding Area, Auxiliary Area, DM (data memory) Area, and EM (Expanded data memory) Area bank 0 only (CPU01) or banks 0 to 3 (CPU 02).

Note Index registers (IR), data registers (DR), and indirectly addressed DM addresses cannot be specified.

Parameter Area

The Parameter Area is inside the CPU Unit and cannot be accessed using instructions. The Parameter Area consists of the following and is accessible only from the NE Programmer.

- PLC Setup
- Built-in Ethernet Setup (This data does not exist in the CS/CJ Series.)
- Registered I/O tables

Flash Memory

The following data is automatically backed up to flash memory when the user performs a read or write operation for the CPU Unit.

- Any part of the user program
- Program information (This data does not exist in the CS/CJ Series.)
- Variable information (This data does not exist in the CS/CJ Series.)
- Parameter Area

When the PLC is turned ON, the data contained in the flash memory is automatically restored. This enables battery-free operation without a Memory Card.

Note (1) Comments and program information are automatically backed up in a Memory Card when data is downloaded or online editing is performed. Comments include variable comments, instruction comments, line comments, and data structure member comments.

(2) DM Area and other I/O memory data is not backed up unless a battery is mounted.

Memory Card

Comments and program information are automatically backed up in a Memory Card when data is downloaded or online editing is performed. Comments include variable comments, instruction comments, line comments, and data structure member comments.

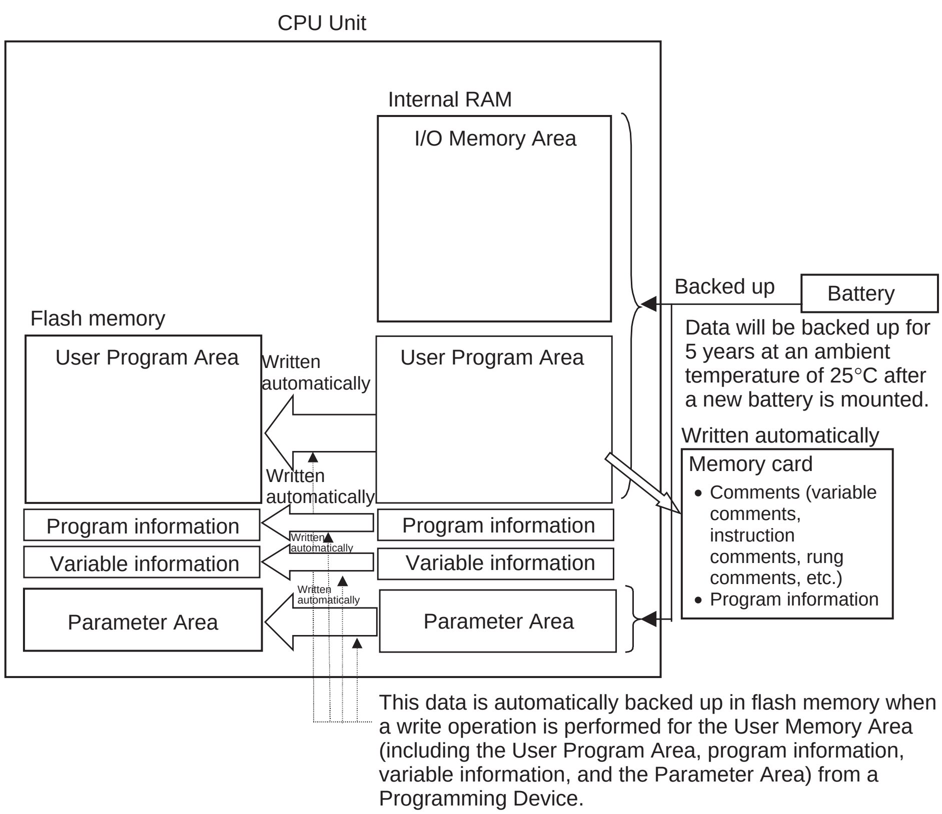
Files on the Memory Card can also be manipulated using the simple backup operation, FREAD/FWRIT instructions in the user program, or FTP operations via Ethernet.

3-1-4 CPU Unit Memory Block Map

The memory of NE1S CPU Unit consists of the following blocks.

- User program area (all user programming)
- Program information
- Variable information
- Parameter Area (PLC Setup, built-in Ethernet Setup, registered I/O tables, routing tables, and CPU Bus Unit System Setup Area)
- I/O Memory Area

Of the above data, the user program area, program information, variable information, and Parameter Area are backed up in flash memory. The I/O Memory Area is backed up using a CJ1W-BAT01 Battery. If the battery voltage is low, the data in this area will be lost.



Note Writing to the following areas will be disabled if pin 1 on the DIP switch on the front of the CPU Unit is turned ON: User Program Area, program information, variable information, and Parameter Area. Collectively, this data is called “user memory.”

⚠ Caution A CJ1W-BAT01 Battery is installed in the NE1S CPU Unit when it is shipped from the factory. It is not necessary to clear memory or set the internal clock.

⚠ Caution The BKUP indicator on the front of the NE1S CPU Unit will light when flash memory is being written. Do not turn OFF the power supply to the PLC when the BKUP indicator is lit.

3-1-5 Memory Card Functions

User File Operations

The user can perform the following file operations for Memory Cards.

- Simple backup operation
- File operations using FREAD/FWRIT instructions in the user program
- FTP transfers via Ethernet
- Automatic transfers at startup

Note The NE1S does not support replacing user programming during operation using Auxiliary Area bit operations or file memory operations with FINS commands.

Simple Backup Operation

The following files are created for the simple backup operation. (See note.)

- Program files
- Data files
- Parameter files
- Unit backup files
- Variable information files
- Program history file

Note If pin 7 on the DIP switch on the front of the CPU Unit is turned ON and the Memory Card power button is pressed for 3 seconds, data is backed up from the CPU Unit to a Memory Card. If the power supply to the PLC is turned ON with pin 7 on the DIP switch on the front of the CPU Unit turned ON, data will be restored from the Memory Card to the CPU Unit.

Details on the files are listed in the following table.

File type	File names for simple backup operation
Program files	• BACKUP.OBJ = All user programming (UM) • BACKUP.PFL = Program information file (file names and other program associated information)
Parameter files	• BACKUP.STD = PLC Setup and registered I/O tables • BACKUP.ETP = Built-in Ethernet Setup • BACKUP.PRT = CIP service protection settings (Rev. 3.1 and later revisions only)
Data files	• BACKUP.IOM = D20000 to D32767 • BACKUPIO.IOR = CIO 0000 to CIO 6143, W000 to W511, H000 to H511, A000 to A959, T0000 to T4095, C0000 to C4095 • BACKUPDM.IOM = D0000 to D19999 • BACKUPE0.IOM = E□_0000 to E0_32767 • BACKUPDM.FSR = D00000 to D32767 forced set/reset data • BACKUPE0.FSR = E0_0000 to E0_32767 forced set/reset data
Unit backup files	• BACKUP□□.PRM = Backup data for Unit with unit number indicated in the boxes.
Variable information files	• BACKUP.USV, BACKUP.TNS, and BACKUP.VTP

File type	File names for simple backup operation
Program history file	• BACKUP.TRK = Program history file
Data file list	• BACKUP.LST = Data file list (Rev. 3.1 and later revisions only)

File Operations Using FREAD/FWRIT Instructions

The FREAD and FWRIT instructions can be used in the user program to read and write the specified I/O memory data (using file extensions of .IOM, .TXT, or .CSV) between the CPU Unit and a Memory Card.

System File Operations

Comment Data and Program Information

When a project is downloaded from the NE Programmer to an NE1S CPU Unit or when a project is edited online, the following data is downloaded to the Memory Card instead of to the CPU Unit. (This data is also uploaded from the Memory Card.)

Note ST programs cannot be edited online.

File type	File names
Comment data	File names are created automatically by the system.
Program Information (including function block definitions)	

3-1-6 CPU Unit Operating Modes

Operating Modes

An NE1S CPU Unit has the following operating modes. These mode control the overall user programming and apply to all tasks.

PROGRAM Mode

Program execution is stopped in PROGRAM mode. The RUN indicator will not be lit. This mode is used for the following preparations before starting program execution.

- Creating the I/O tables
- Setting the PLC Setup and other settings
- Transferring and checking the program
- Checking wiring by force-setting/resetting I/O bits
- Other operations

All cyclic tasks will be in disabled status (INI) in PROGRAM mode. All interrupt tasks will be stopped.

I/O refreshing, however, is still performed in PROGRAM mode.

MONITOR Mode

The program is executed in MONITOR mode and the RUN indicator will be lit. The following setting operations are possible from the NE Programmer in this mode.

- Online editing
- Force-setting/resetting bits
- Changing present values in I/O memory
- Other operations

MONITOR mode is used for system adjustments during trial operation.

In MONITOR mode, the cyclic tasks set for execution at startup and any cyclic tasks enabled by the TASK ON instruction (TKON) will be executed. Interrupt tasks will be executed when their interrupt conditions have been met.

RUN Mode

The program is executed in RUN mode and the RUN indicator will be lit. In this mode, online editing, force-setting/resetting bits, changing present values, and other setting operations are not possible from the NE Programmer. The NE Programmer, however, can be used to monitor program status and I/O memory status.

RUN mode is used for normal system operation. The operation of tasks is the same as that for MONITOR mode.

Operating Mode at Startup

The operating mode when power is turned ON to the CPU Unit will always be PROGRAM mode.

Relation between Changes in Operating Mode and Physical Addresses in I/O Memory

Mode change	I/O memory physical addresses	
	Areas that are not held (CIO, Work, Timer (PVs and Completion Flags), Index Register, Data Register, and Task Flag Areas) **Note** In the Auxiliary Area, some portions are held, some are not.	**Areas that are held (Holding, DM, EM, and Counter (PVs and Completion Flags) Areas)**
RUN or MONITOR to PROGRAM	Cleared (See note.)	Held
PROGRAM to RUN or MONITOR	Cleared (See note.)	Held
RUN to MONITOR or MONITOR to RUN	Held	Held

Note These areas are held if the I/O Memory Hold Bit is turned ON. If operation stops for a fatal error (including FAL execution), however, I/O memory in the CPU Unit is held, but all outputs from Output Units are turned OFF.

Relation between Changes in Operating Mode and Specification for Holding/Clearing Automatically Allocated Variables for Power Interruptions

Mode change	Automatically allocated variables	
	Clearing for power interruptions specified	**Holding for power interruptions specified**
RUN or MONITOR to PROGRAM	Cleared	Held
PROGRAM to RUN or MONITOR	Cleared	Held
RUN to MONITOR or MONITOR to RUN	Held	Held

3-1-7 I/O Allocations

Words must be allocated to the Units mounted to an NE1S PLC. This is called I/O allocation. The mounting location, allocated words, and other information on the Units is called the I/O tables and registering this information in the CPU Unit is called I/O table registration.

When the power supply is turned ON, the CPU Unit will verify the mounted Units against those registered in the I/O tables before it starts operation.

Unit Classifications and Allocation Methods

There are three classifications of Unit, as listed below, and the I/O allocation method for each class of Unit is different.

- Basis I/O Units
- Special I/O Units
- CPU Bus Units

Unit classification	Area allocated	Allocation method
Basic I/O Units	CIO 0000 to CIO 0319	Using the Mounted Units The number of words required by each Unit (in word (16 bits) units) are allocated to all Basic I/O Units from left to right on the CPU Rack starting from the Basic I/O Unit closest to the CPU Unit. When this method is used, allocations of I/O memory to the Basic I/O Units is automatically based on the order the Units are mounted (right to left on each Rack and in order from Rack 0 to Rack 7).
		Not Using the Mounted Units With this method, some of the allocations can be made without regard to the mounting position and CIO 0000 to CIO 0999 can be allocated. • Assigning First Rack Words: Order of Racks can be specified regardless of the order they are connected, words are allocated left to right starting at slot 00. • Assigning First Slot Words: Not supported.
Special I/O Units	Special I/O Unit Area CIO 2000 to CIO 2959	10 words/Unit: The words that are allocated are determined by the unit number.
CPU Bus Units	CPU Bus Unit Area CIO 1500 to CIO 1899	25 words/Unit: The words that are allocated are determined by the unit number.

3-1-8 CPU Execution Mode

The program is executed parallel processing mode while syncing I/O memory access. Program execution and peripheral servicing are executed sequentially.

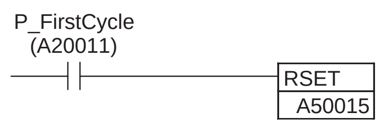
The peripheral servicing time can be set in the PLC Setup to 8%, 16%, or 32% of the cycle time.

3-1-9 Precautions for Operations without a Battery

The NE1S-series CPU Units can be operated without a battery (i.e., without a battery installed or without being affected by low battery power). The following precautions, however, must be observed when operating the CPU Unit without using a battery.

- When operating the CPU Unit without a battery, set the detect low battery setting in the PLC Setup to not detect low-battery errors. (Using the NE Programmer, click the **General** Tab in the PLC Setup, select ***CPU Settings***, and select ***Nondetect*** for the *Detect Low Battery* setting.)

- A memory error will occur when the power is turned ON if literal text strings are used for Text String Processing Instructions.
  Example:
  MOV$'ABCD'StringVar (*Transfers ABCD to *StringVar* using STRING variable.*)
  If a literal text string of the STRING data type is used for ABCD and the battery is low or not mounted, a memory error will occur when the power is turned ON.
- HR, DM, and EM words and variables with holding attributes will all become undefined values. Make sure that the program does not require values to be held in these areas.
  Also, when a battery is not mounted, set the IOM Hold Bit (A50012) and Forced Status Hold Bit (A50013) to not retain settings when the power is turned ON.
- The value of the Output OFF Bit (A50015) will be undefined when the battery is not mounted or the battery power is low. All the Output Unit's outputs will turn OFF when the Output OFF Bit turns ON. Incorporate the following programming in the ladder program to prevent all the outputs on the Output Units from turning OFF when the power is turned ON.



- The calendar and clock functions cannot be used. Therefore, the Calendar/Clock Area words (A351 to A354) and Startup Time (A510 and A511) values will be undefined. The date and time for the file created by the CPU Unit in the Memory Card will also be undefined.
- The bits in Auxiliary Area words Total Power ON Time (A523), Power Interruption Time (A512 and A513), and Number of Power Interruptions (A514) will be set to 0 when the power is turned ON.
- The error log data in the Error Log Area (A100 to A199) will not be retained when the power is turned ON.

3-1-10 Battery Mounting Method

Purpose of Mounting Battery

The battery backs up the clock and the contents of RAM in the CPU Unit when the power supply is turned OFF.

- User program
- PLC Setup
- Portion of I/O memory to be retained

If a battery is not mounted or the battery voltage had exceeded its useful life, the clock will stop and the above data will be lost when power is turned OFF.

Battery Model

Name: Battery Set
Model: CJ1W-BAT01

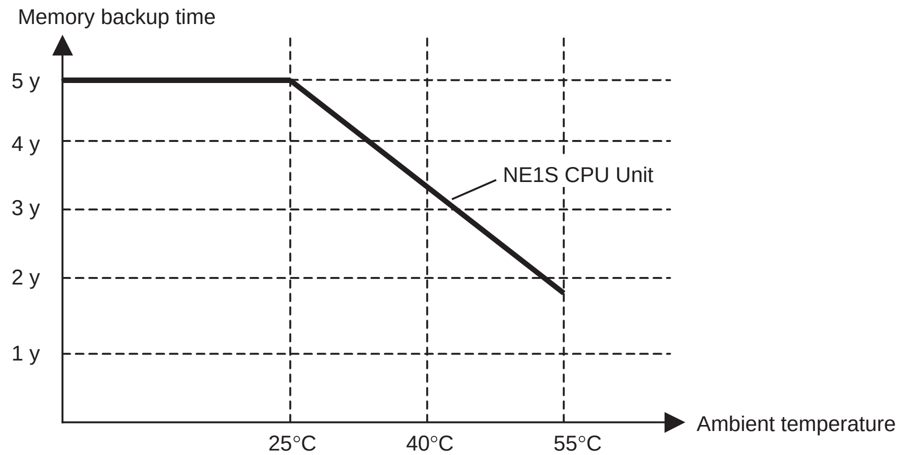
Battery Life and Replacement Period

The effective battery life (i.e., the maximum life) is 5 years at 25°C regardless of whether power is supplied to the CPU Unit. The life will be shorter at higher temperatures.

The minimum and typical backup times of the battery are given in the following table.

Effective life (i.e., max. life)	Minimum backup time (See note.)	Typical backup time
5 years	13,000 hours (approx. 1.7 years)	43,000 hours (approx. 5 years)

Note Minimum life: Memory backup time at an ambient temperature of 55°C
Typical life: Memory backup time at an ambient temperature of 25°C.



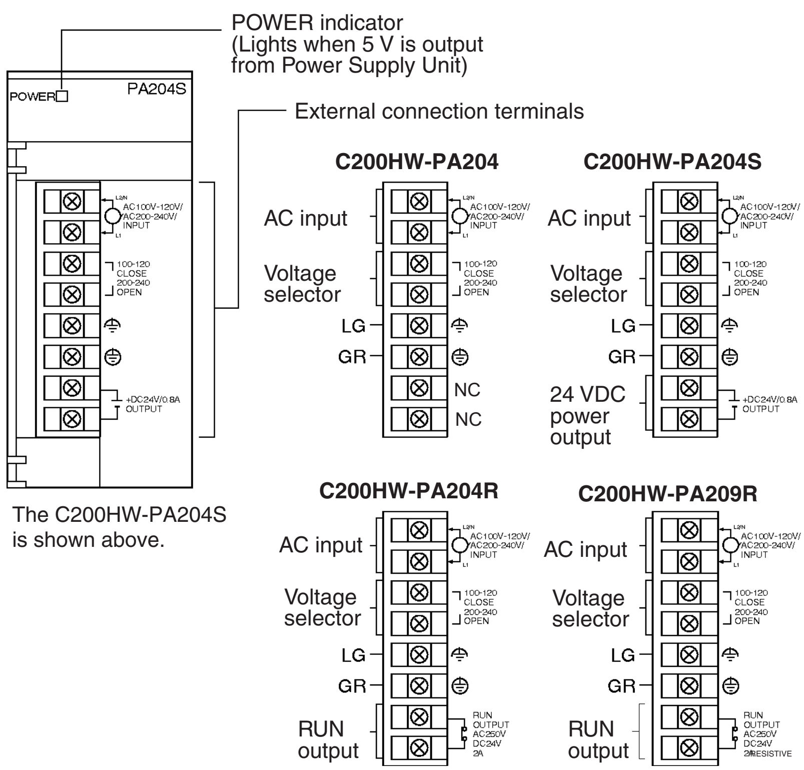
Note The above graph gives reference values for the memory backup time.

3-2 Power Supply Units

3-2-1 Power Supply Units

Power supply voltage	Output	Power output terminals	RUN output	Model	Weight
100 to 120 V AC or 200 to 240 V AC (selectable using jumper)	4.6 A at 5 V DC, 30 W	No	No	C200HW-PA204	500 g max.
		Yes 0.8 A at 24 V DC	No	C200HW-PA204S	500 g max.
		No	Yes	C200HW-PA204R	500 g max.
	9 A at 5 V DC, 45 W	No	Yes	C200HW-PA209R	1,000 g max.

3-2-2 Components and Switch Settings



The C200HW-PA204S is shown above.

Note 100 to 120 V AC: Closed circuit, 200 to 240 V AC: Open circuit
Always open the circuit (remove the metal jumper) before applying a voltage of 200 to 240 V AC.

AC Input

Either a power supply of 100 to 120 V AC or 200 to 240 V AC can be selected.

Voltage Selector

Before applying a voltage of 100 to 120 V AC, close the circuit using the metal jumper.

Note Always remove the metal jumper before applying a voltage of 200 to 240 V AC. Not doing so will damage the Unit.

LG

Ground to a resistance of 100 Ω or less to increase noise resistance and avoid electric shock.

GR

Ground to a resistance of 100 Ω or less to avoid electric shock.

24-V DC Power Output

This terminal outputs a service voltage of 24 V DC. Use this terminal to supply power to DC Input Units (C200HW-PA204S only). The total current consumption of the 5-V and 24-V outputs must be 30 W max.

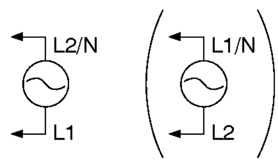
DC Input

24-V DC power is supplied from this terminal.

RUN Output

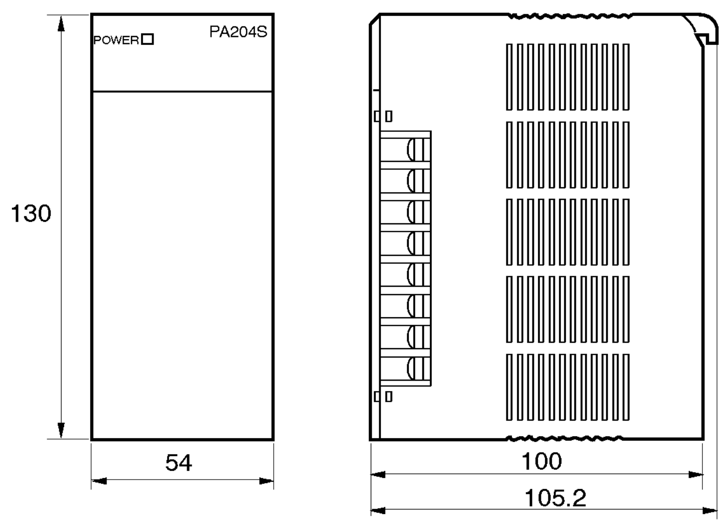
The internal contact turns ON when the CPU Unit is operating (RUN or MONITOR mode). (The RUN output does not turn ON on Expansion Racks.)

Note The L2/N and L1 display on the AC power supply terminal is L1/N and L2 on some products, however, the function and performance of the terminals are the same.

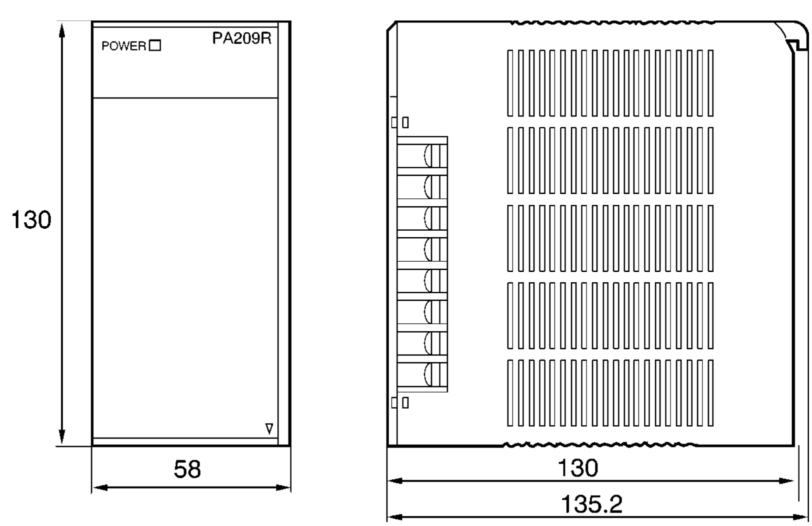


3-2-3 Dimensions

C200HW-PA204
C200HW-PA204S
C200HW-PA204R



C200HW-PA209R



3-2-4 Selecting a Power Supply Unit

After determining what power supply voltage is required and whether power output terminals and a RUN output are required, calculate the current and power requirements for each Rack.

Condition 1: Current Requirements

There are three voltage groups for internal power consumption: 5 V DC, 26 V DC, and 24 V DC.

Current Consumption at 5 V DC (Internal Logic Power Supply)

The following table shows the current that can be supplied to Units (including the CPU Unit) and Backplanes that use 5-V DC power.

Power Supply Unit	Maximum current at 5 V DC
C200HW-PA204/204S/204R	4.6 A
C200HW-PA209R	9 A

Current Consumption at 26 V DC (Relay Driving Power Supply)

The following table shows the current that can be supplied to Units that use 26-V DC power.

Power Supply Unit	Maximum current at 26 V DC
C200HW-PA204/204S/204R	0.625 A
C200HW-PA209R	1.3 A

Current Consumption at 24 V DC (Power Output Terminals)

The C200HW-PA204S Power Supply Unit can supply up to 0.8 A at 24 V DC through its power output terminals.

Condition 2: Power Requirements

The following table shows the maximum total power that can be supplied at 5 V DC, 26 V DC, and 24 V DC.

Power Supply Unit	Maximum total power output
C200HW-PA204/204S/204R	30 W
C200HW-PA209R	45 W

Refer to *1-3-4 Unit Current Consumption* for tables showing the current consumed by each particular Unit as well as example calculations.

3-3 Backplanes

3-3-1 CPU Backplanes

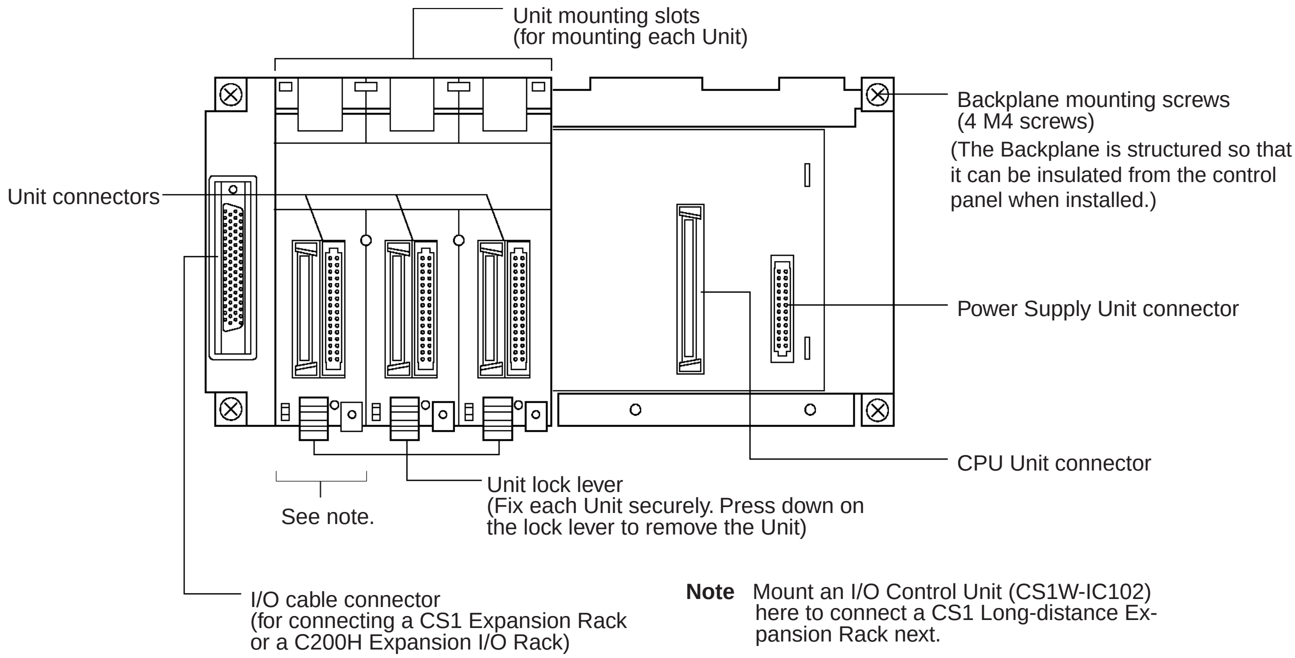
CPU Backplane Models

CPU Backplanes

Number of slots	Model
2 slots	CS1W-BC022
3 slots	CS1W-BC032
5 slots	CS1W-BC052
8 slots	CS1W-BC082
10 slots	CS1W-BC102

Note C200H Units (C200H Basic I/O Units, C200H Group-2 High-density I/O Units, and C200H Special I/O Units) cannot be used.

Components and Switch Settings

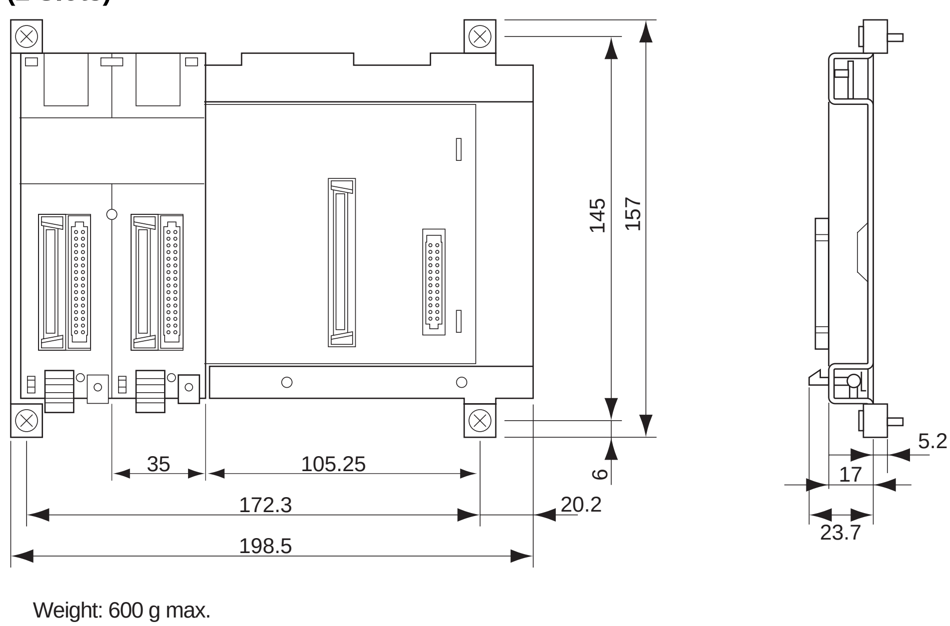


Note Always cover the connectors that are not being used with Connector Covers (sold separately) as a measure against dust.

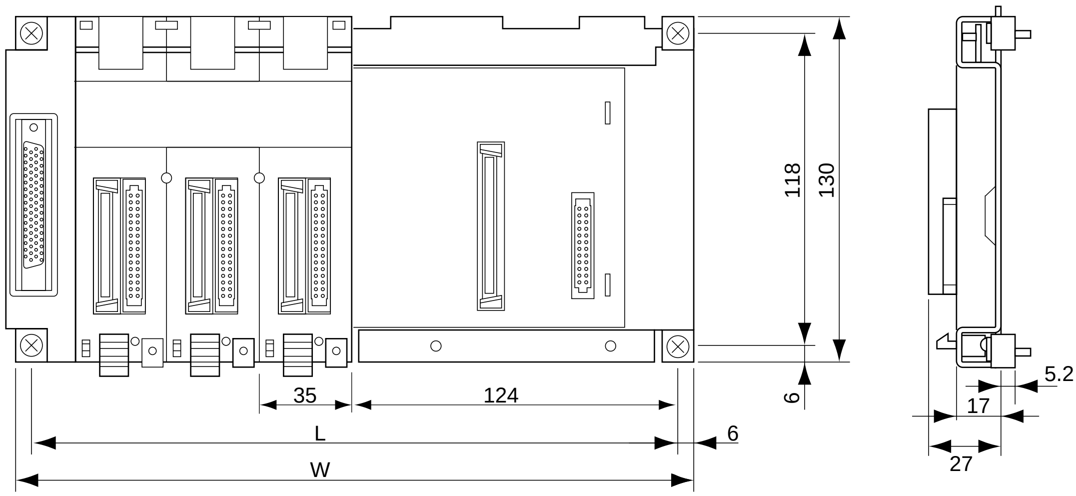
Name	Model
Special I/O Unit Connector Cover	CV500-COV01

Dimensions and Weights

CS1W-BC022 (2 Slots)



CS1W-BC□□□
(3, 5, 8, or 10 Slots)



Model	Number of slots	L (mm)	W (mm)	Weight (max.)
CS1W-BC032	3	246	260	750 g
CS1W-BC052	5	316	330	900 g
CS1W-BC082	8	421	435	1,200 g
CS1W-BC102	10	491	505	1,400 g

3-3-2 Expansion Backplanes

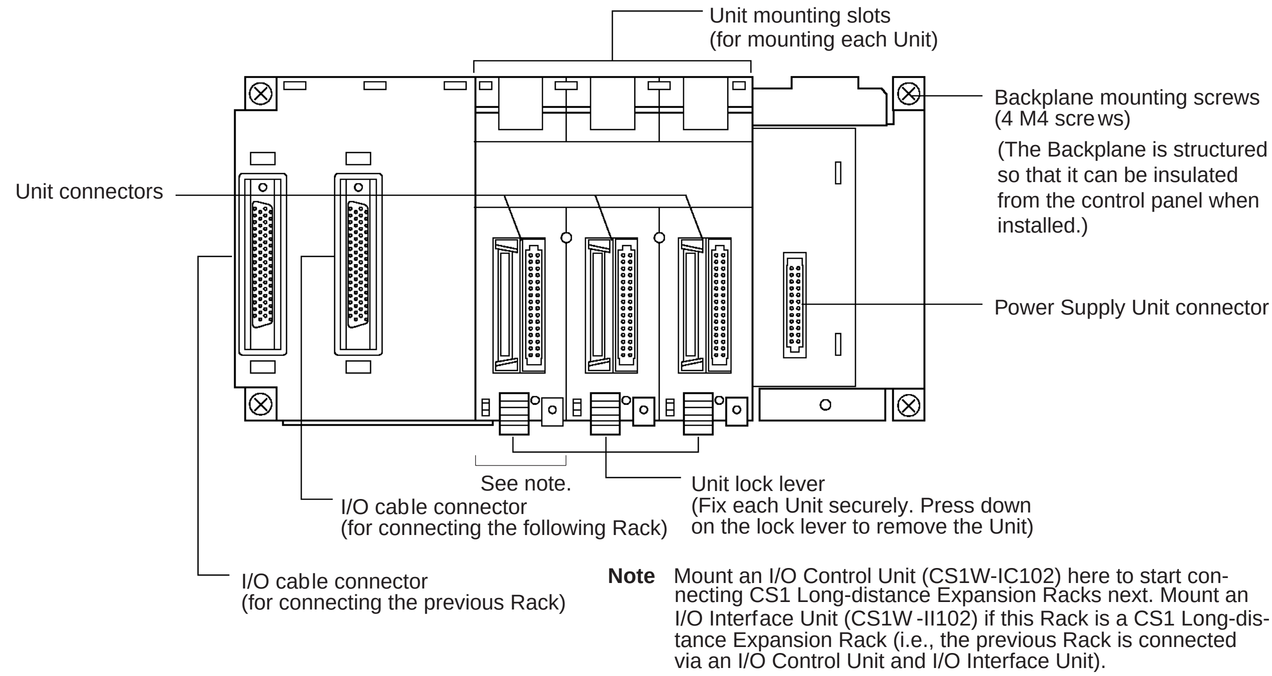
Expansion Backplanes are used for both Expansion Racks and Long-distance Expansion Racks.

Expansion Backplane Models

Number of slots	Model
3 slots	CS1W-BI032
5 slots	CS1W-BI052
8 slots	CS1W-BI082
10 slots	CS1W-BI102

Note C200H Units (C200H Basic I/O Units, C200H Group-2 High-density I/O Units, and C200H Special I/O Units) cannot be used.

Components and Switch Settings

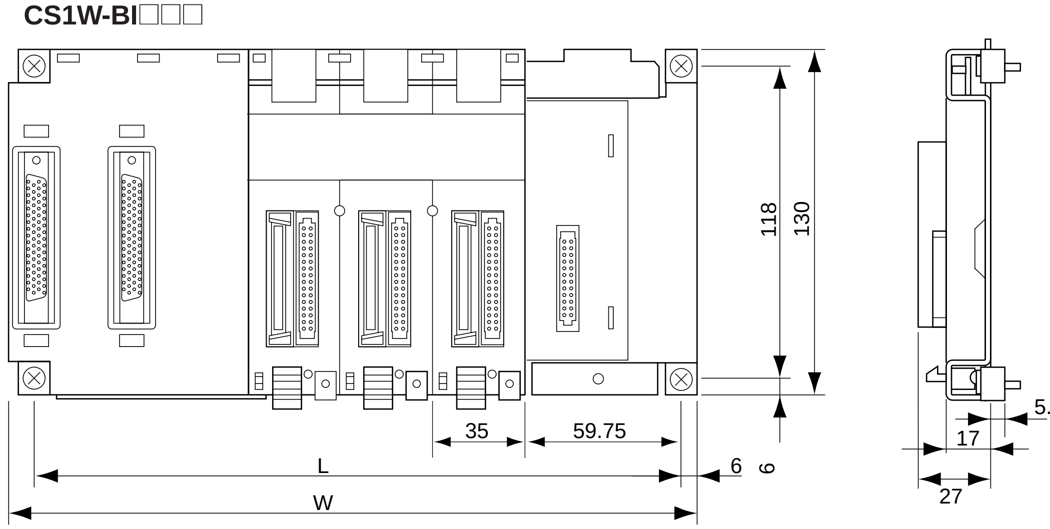


Note Always cover the connectors that are not being used with Connector Covers (sold separately) as a measure against dust.

Name	Model
CPU Bus Unit Connector Cover	CV500-COV01

Dimensions

CS1W-BI□□□

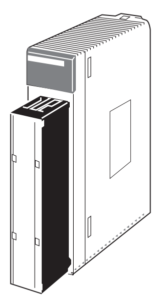


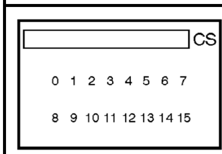
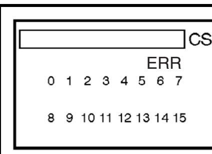
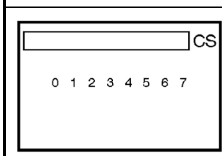
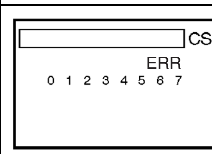
Model	Number of slots	L (mm)	W (mm)	Weight (max.)
CS1W-BI032	3	246	260	750 g
CS1W-BI052	5	316	330	900 g
CS1W-BI082	8	421	435	1,200 g
CS1W-BI102	10	491	505	1,400 g

3-4 Basic I/O Units

3-4-1 Basic I/O Units with Terminal Blocks

Name		Specifications	Model
Basic Input Units with terminal blocks	AC Input Units	100 to 120 V AC, 100 to 120 V DC, 16 inputs	CS1W-IA111
		200 to 240 V AC, 16 inputs	CS1W-IA211
		24 V DC, 16 inputs	CS1W-ID211
	Interrupt Input Unit	24 V DC, 16 inputs	CS1W-INT01
	High-speed Input Unit	24 V DC, 16 inputs	CS1W-IDP01
Basic Output Units with terminal blocks	Relay Output Units	2 A at 250 V AC/24 V DC max., 0.1 A at 120 V DC, independent contacts, 8 outputs	CS1W-OC201
		2 A at 250 V AC/24 V DC max., 0.1 A at 120 V DC, 16 outputs	CS1W-OC211
	Triac Output Units	1.2 A at 250 V AC max., 8 outputs, with fuse burnout detection circuit	CS1W-OA201
		0.5 A at 250 V AC max., 16 outputs	CS1W-OA211
	Transistor Output Unit with Sinking Outputs	0.5 A at 12 to 24 V DC, 16 outputs	CS1W-OD211
	Transistor Output Unit with Sourcing Outputs	0.5 A at 24 V DC, load short-circuit protection, 16 outputs	CS1W-OD212

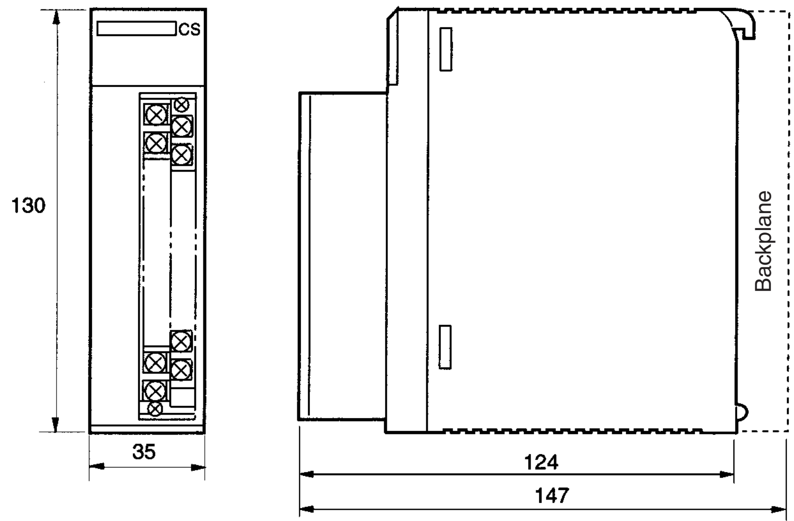


Terminal block					
	16-point Unit	CS1W-ID211 INT01 IDP01 OD211 IA111 IA211 OC211 OA211		16-point Units with ERR indicator (load short-circuit)	CS1W-OD212
	8-point Unit	CS1W-OC201		8-point Units with ERR indicator (fuse burnout)	CS1W-OA201

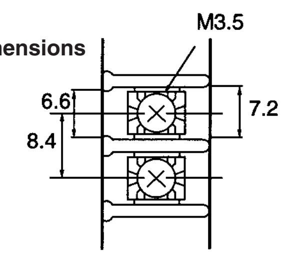
Dimensions

Units with 20-point Terminal Blocks

CS1W-IA111
CS1W-IA211
CS1W-ID211
CS1W-INT01
CS1W-IDP01
CS1W-OD211
CS1W-OD212
CS1W-OA201
CS1W-OA211
CS1W-OC201
CS1W-OC211



Terminal Dimensions



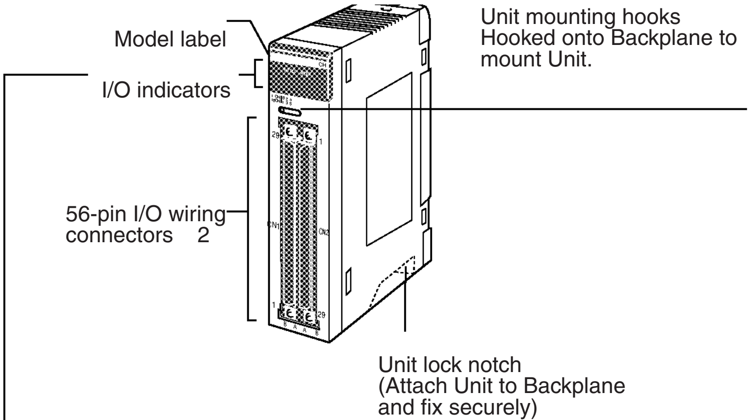
3-4-2 Basic I/O Units with Connectors (32-, 64-, and 96-pt Units)

Basic I/O Units are classified as Basic I/O Units.

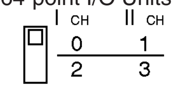
Models

Name	Specifications	Model
DC Input Unit	24 V DC, 32 inputs	CS1W-ID231
	24 V DC, 64 inputs	CS1W-ID261
	24 V DC, 96 inputs	CS1W-ID291
Transistor Output Unit, Sinking	0.5 A at 12 to 24 V DC, 32 outputs	CS1W-OD231
	0.3 A at 12 to 24 V DC, 64 outputs	CS1W-OD261
	0.1 A at 12 to 24 V DC, with fuse burnout detection circuit, 96 outputs	CS1W-OD291
Transistor Output Unit, Sourcing	0.5 A at 24 V DC, load short-circuit protection, 32 outputs	CS1W-OD232
	0.3 A at 24 V DC, load short-circuit protection, 64 outputs	CS1W-OD262
	0.1 A at 24 V DC, with fuse burnout detection circuit, 96 outputs	CS1W-OD292
DC Input/Transistor Output Unit, Sinking	24 V DC input, 0.3 A output at 12 to 24 V DC, 32 inputs/32 outputs	CS1W-MD261
	24 V DC input, 0.1 A output at 12 to 24 V DC, with fuse burnout detection circuit, 48 inputs/48 outputs	CS1W-MD291
DC Input/Transistor Output Unit, Sourcing	24 V DC input, 0.3 A output at 24 V DC, load short-circuit protection, 32 inputs/32 outputs	CS1W-MD262
	24 V DC input, 0.1 A output at 24 V DC, with fuse burnout detection circuit, 48 inputs/48 outputs	CS1W-MD292
TTL I/O Unit	3.5 mA at 5 V DC, 32 inputs 35 mA at 5 V DC, 32 outputs	CS1W-MD561

Note Immediate refreshing (!) or refreshing using IORF(097) is possible for Basic I/O Units with Connectors (32-, 64-, and 96-pt Units).

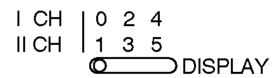


Display Switch
for 64-point I/O Units



	Display switch	
	0, 1	2, 3
Area 1	m	m + 2
Area 2	m + 1	m + 3

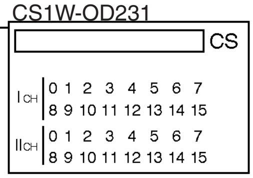
Display Switch
for 96-point I/O Units



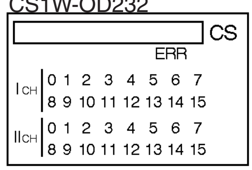
Display switch (3-level selection)

	Display switch		
	0, 1	2, 3	4, 5
Area 1	m	m + 2	m + 4
Area 2	m + 1	m + 3	m + 5

32-point Units
CS1W-ID231
CS1W-OD231

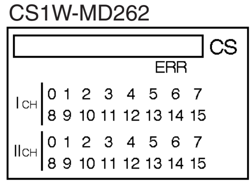


32-point Units with ERR indicator
(load short-circuit indicator)
CS1W-OD232

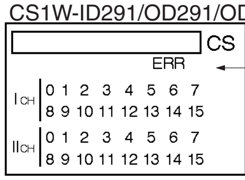


64-point Units
CS1W-ID261
CS1W-OD261
CS1W-MD261

64-point Units with ERR indicator
(load short-circuit indicator)
CS1W-OD262
CS1W-MD262

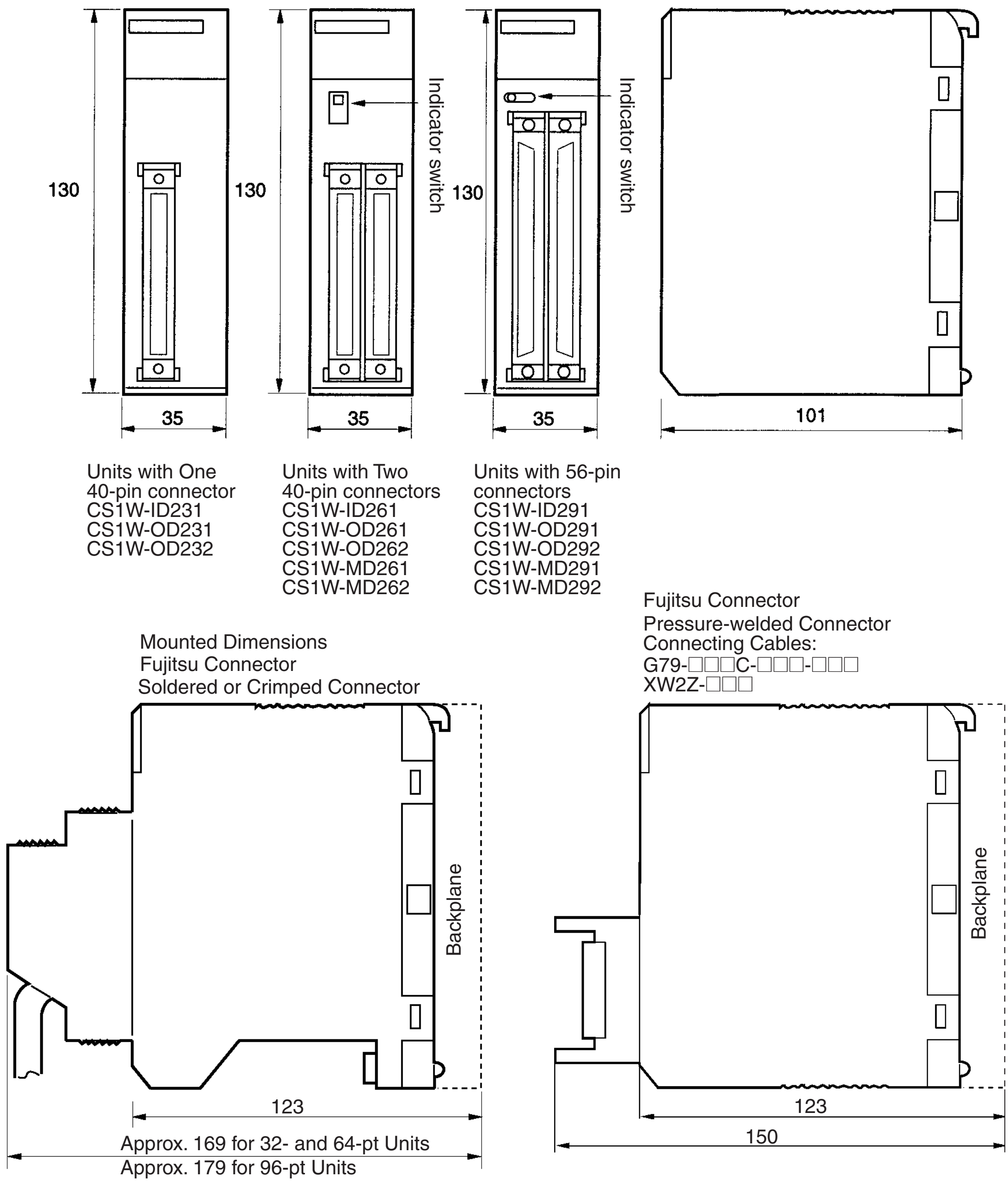


CS1W-ID291/OD291/OD292/MD291/MD292



F (fuse burnt out) indicator
Available on Output Units.
Lights when one or more fuses in the Unit blows.
Lights when external power is OFF.

Dimensions

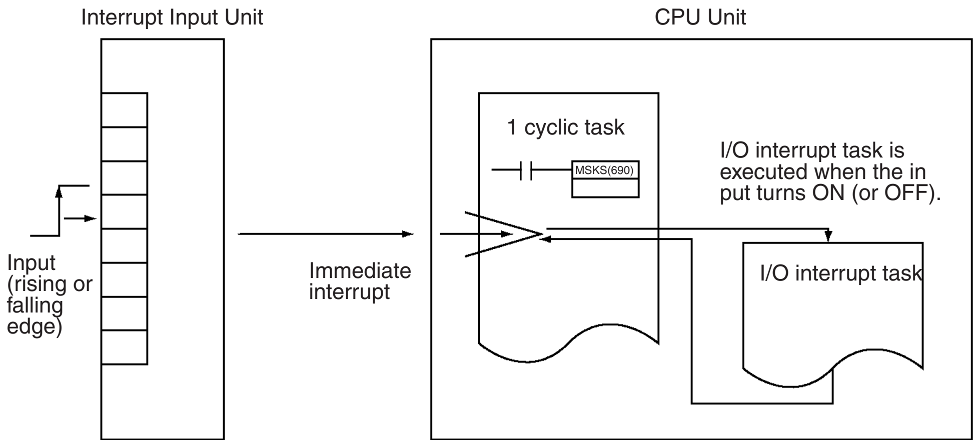


3-4-3 Interrupt Input Units

Functions

Interrupt Input Units are used to execute interrupt programs on the rising or falling edge of an input signal (See note.). When the specified interrupt input turns ON (or OFF), execution of the cyclic program in the CPU Unit is interrupted and an I/O interrupt task (task number 100 to 131) is executed. When execution of the I/O interrupt task has been completed, the cyclic program is again executed starting from the instruction after which it was interrupted.

Note Only the CS1W-INT01 can detect falling edges. The interrupt control instructions, however, can be used to switch between upward and downward differentiation.



Applicable Units

Either of the following Interrupt Input Units can be used.

Model	Specifications	No. of Units mountable to CPU Rack
CS1W-INT01	24 V DC 16 inputs	2 max.

Application Precautions

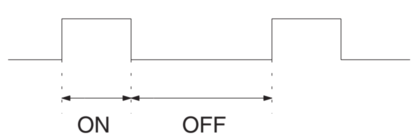
All Interrupt Input Units must be mounted to the CPU Rack. The interrupt input function will not be supported if an Interrupt Input Unit is mounted to an Expansion Rack. If mounted to an Expansion Rack, the Unit can be used as a normal I/O Unit.

There are limits to the number of Interrupt Input Units that can be mounted. (See table, above.)

The input response time cannot be changed for the CS1W-INT01, and the related portions of the Basic I/O Unit input time constants in the PLC Setup, and the setting status in A220 to A259 will not be valid.

Input Signal Width

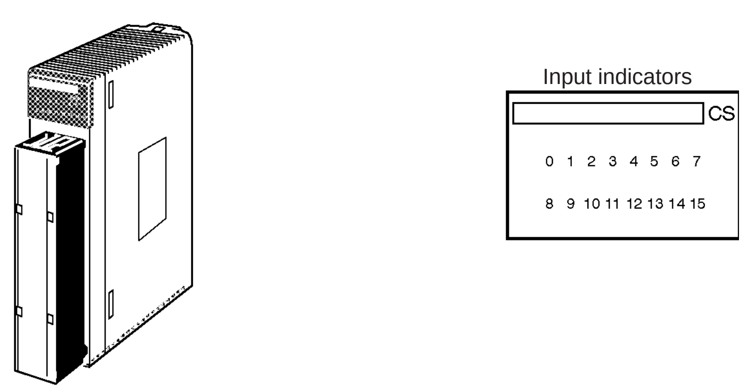
Input signals must meet the following conditions.



Unit	ON time	OFF time
CS1W-INT01	0.1 ms min.	0.5 ms min.

Components

CS1W-INT01



Using I/O Interrupts

1,2,3...

1. Mount the Input Interrupt Unit to the CPU Rack and create the I/O tables.
2. Create the I/O task (see note 1).
3. Use the SET INTERRUPT MASK Instruction (MSKS(690)) in the cyclic program to enable the required interrupt input numbers (see note 1).
4. Turn ON or OFF the inputs on the Interrupt Input Unit for the interrupt input numbers that have been enabled. (The default is for an interrupt on the rising edge.)

Note

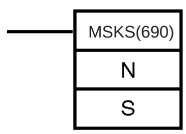
1. The relationship between Interrupt Input Unit numbers, interrupt input numbers, and I/O interrupt tasks is shown in the following table.

Model	Interrupt Input Unit number	Interrupt input number	I/O interrupt task number
CS1W-INT01	0	0 to 15	100 to 115
	1		116 to 131

2. The CS1W-INT01 can set to detect either rising or falling edges.

MSKS(690) Instruction

Specifying Rising/Falling Edge for CS1W-INT01



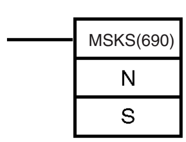
N: Control data 1 (Interrupt Input Unit No.)
S: Control data 2 (interrupt mask data)

The MSKS(690) instruction is used to set rising edge or falling edge detection for each interrupt input number.

- The value of N will determine which Interrupt Input Unit will perform the I/O interrupt processing.
- The value of S will determine rising or falling edge detection for each interrupt input number.

Operand	Value	Details
N	2, 3	Interrupt Input Unit Number Numbers 2 and 3 are assigned to the Units in order from left to right. 2:Interrupt Input Unit 0 (interrupt task numbers 100 to 115) 3:Interrupt Input Unit 1 (interrupt task numbers 116 to 131)
S	0000 to FFFF hex	Rising/Falling Edge Designation Bits 00 to 15 correspond to the interrupt input numbers (interrupt tasks 100 to 115 or 116 to 131). 0: Rising edge 1: Falling edge

Specifying I/O Interrupt Processing



N: Control data 1 (Interrupt Input Unit No.)
S: Control data 2 (interrupt mask data)

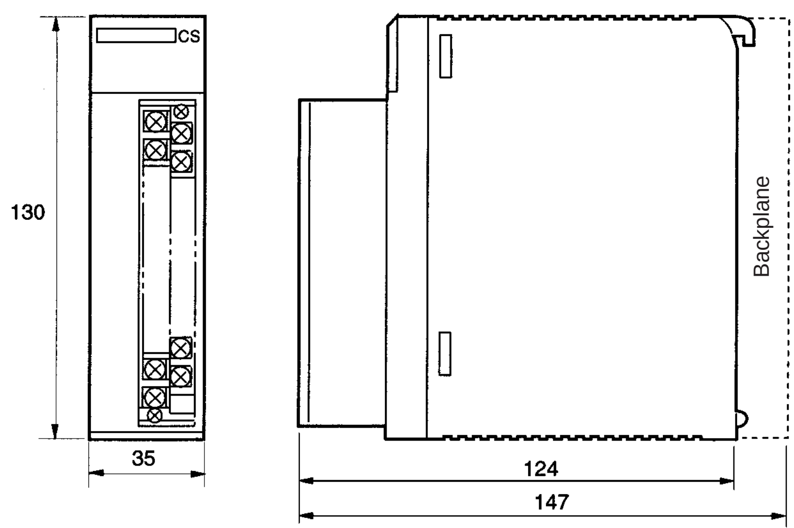
The MSKS(690) instruction is used to set I/O interrupt processing or scheduled interrupt processing.

- The value of N will determine which Interrupt Input Unit will perform the I/O interrupt processing.
- The value of S will determine which interrupt number will be enabled.

Operand	Value	Details
N	0, 1	Interrupt Input Unit Number Numbers are assigned to the Units in order from left to right. 0:Interrupt Input Unit 0 (interrupt task numbers 100 to 115) 1:Interrupt Input Unit 1 (interrupt task numbers 116 to 131)
S	0000 to FFFF hex	Interrupt Mask Data Bits 00 to 15 for the CS1W-INT01 are used for the Interrupt Input Unit's interrupt input number. 1:Interrupt masked (interrupt input disabled) 0:Interrupt valid (interrupt input enabled)

Dimensions

Unit with 20-point Terminal Block
CS1W-INT01



Note The height of the Unit including the Backplane is 5 mm greater on the CPU Backplane and Expansion Backplane than the Unit by itself.

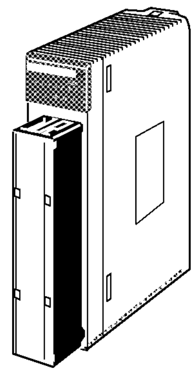
3-4-4 Units with High-speed Inputs

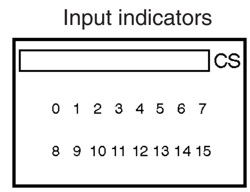
Functions

The CS1W-IDP01 enables inputting pulse signals that are shorted than the cycle time of the CPU Unit.

Components

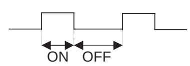
CS1W-IDP01





Input Signal Width

High-speed input signals must meet the following conditions for the ON time.



Model	ON time
CS1W-IDP01	0.1 ms min.

Dimensions

The High-speed Input Unit has the same dimensions as the Interrupt Input Unit. Refer to page 106.

SECTION 4
PLC Setup

This section describes the settings in the PLC Setup and how they are used to control CPU Unit operation.

4-1 General Tab Page

4-1-1 Startup Settings

Startup IOM Hold

Settings	Function	Related flags and words	New setting's effectiveness
Invalid Valid Default: Invalid	This setting determines whether or not the status of the IOM Hold Bit (A50012) is retained at startup. When you want all of the data in I/O Memory to be retained when the power is turned ON, turn ON the IOM Hold Bit and set this setting to 1 (ON).	A50012 (IOM Hold Bit)	At startup

Startup Force Status

Settings	Function	Related flags and words	New setting's effectiveness
Invalid Valid Default: Invalid	This setting determines whether or not the status of the Forced Status Hold Bit (A50013) is retained at startup. When you want all of the bits that have been force-set or force-reset to retain their forced status when the power is turned ON, turn ON the Forced Status Hold Bit and set this setting to *Retained*.	A50013 (Forced Status Hold Bit)	At startup

Startup Operating Mode

Settings	Function	Related flags and words	New setting's effectiveness
PROGRAM MONITOR RUN Default: RUN	This setting determines whether the Startup Mode will be the mode set on the Programming Console's mode switch or the mode set here in the PLC Setup.	---	At startup

4-1-2 CPU Settings

Detect Low Battery

Settings	Function	Related flags and words	New setting's effectiveness
Detect Nondetect Default: Detect	This setting determines whether CPU Unit battery errors are detected. If this setting is set to *Detect* and a battery error is detected, the ERR/ALM indicator on the CPU Unit will flash and the Battery Error Flag (A40204) will be turned ON, but CPU Unit operation will continue.	A40204 (Battery Error Flag)	Next cycle

Detect Interrupt Task Error

Settings	Function	Related flags and words	New setting's effectiveness
Detect Nondetect Default: Detect	This setting determines whether interrupt task errors are detected. If this setting is set to *Detect* and an interrupt task error is detected, the ERR/ALM indicator on the CPU Unit will flash and the Interrupt Task Error Flag (A40213) will be turned ON, but CPU Unit operation will continue.	A40213 (Interrupt Task Error Flag)	Next cycle

Register FAL to Error Log

Settings	Function	Related flags and words	New setting's effectiveness
Register Unregister Default: Register	This setting determines if user-defined FAL errors created with FAL(006) and time monitoring for FPD(269) will be recorded in the error log (A100 to A199).	---	Whenever FAL(006) is executed (every cycle)

Table Data Process Instructions

Settings	Function	Related flags and words	New setting's effectiveness
Disable Enable Default: Disable	This setting determines if Table Data Instructions will be processed over multiple cycle times (i.e., processed in the background).	---	Start of operation

String Data Process Instructions

Settings	Function	Related flags and words	New setting's effectiveness
Disable Enable Default: Disable	This setting determines if Text String Data Instructions will be processed over multiple cycle times (i.e., processed in the background).	---	Start of operation

Data Shift Process Instructions

Settings	Function	Related flags and words	New setting's effectiveness
Disable Enable Default: Disable	This setting determines if Data Shift Instructions will be processed over multiple cycle times (i.e., processed in the background).	---	Start of operation

Com Port Number

Settings	Function	Related flags and words	New setting's effectiveness
0 to 7: Communications ports 0 to 7 (internal logical ports) Default: 0	The communications port number (internal logical port) that will be used for background execution. The same port number is used for all instructions.	---	Start of operation

Stop CPU on Instruction Error

Settings	Function	Related flags and words	New setting's effectiveness
Continue Stop Default: Continue	This setting determines whether instruction errors (instruction processing errors (ER) and illegal access errors (AER)) are treated as non-fatal or fatal errors. When this setting is set to *Stop*, CPU Unit operation will be stopped if the ER or AER Flags is turned ON (even when the AER Flag is turned ON for an indirect DM/EM BCD error). Related Flags: A29508 (Instruction Processing Error Flag) A29509 (Indirect DM/EM BCD Error Flag) A29510 (Illegal Access Error Flag)	A29508, A29509, A29510 (If this setting is set to *Continue*, these flags won't be turned ON even if an instruction error occurs.)	Start of operation

4-1-3 Timings Settings

Schedule Interrupt Interval

Settings	Function	Related flags and words	New setting's effectiveness
10 ms 1.0 ms Default: 10 ms	Sets the time interval for the scheduled interrupt task.	---	Start of operation. . (Can't be changed during operation.)

Cycle Monitor Time (Minimum Cycle Time)

Settings	Function	Related flags and words	New setting's effectiveness
1 to 32,000 ms (1-ms increments) Default: 0 ms	Set to the desired value to specify a minimum cycle time. If the cycle time is less than this setting, it will be extended until this time passes. Leave this setting at 0 ms for a variable cycle time.	---	Start of operation. (Can't be changed during operation.)

Cycle Time

Settings	Function	Related flags and words	New setting's effectiveness
Disabled Enabled Default: Disabled (1,000 ms)	Set to *Enabled* to use a cycle monitor time setting other than the default value of 1 s.	A264 and A265 (Present Cycle Time)	Start of operation. (Can't be changed during operation.)

Set Cycle Monitor Time

Settings	Function	Related flags and words	New setting's effectiveness
10 to 4,000 ms Default: 1,000 ms	This setting is valid only when *Enable Minimum Cycle Time Setting* is set to *Enable*. The Cycle Time Too Long Flag (A40108) will be turned ON if the cycle time exceeds this setting.	A40108 (Cycle Time Too Long Flag)	Start of operation. (Can't be changed during operation.)

Power OFF Interrupt

Settings	Function	Related flags and words	New setting's effectiveness
Invalid Valid Default: Invalid	When this setting is set to *Enabled*, the power OFF interrupt task will be executed when power is interrupted.	---	At startup or at start of operation. (Can't be changed during operation.)

Power OFF Detection Time

Settings	Function	Related flags and words	New setting's effectiveness
0 to 10 ms (1-ms increments) Default: 0	This setting determines how much of a delay there will be from the detection of a power interruption (approximately 10 to 25 ms after the power supply voltage drops below 85% of the rated value) to the confirmation of a power interruption. The default setting is 0 ms. When the power OFF interrupt task is enabled, it will be executed when the power interruption is confirmed. If the power OFF interrupt task is disabled, the CPU will be reset and operation will be stopped.	---	At startup or at start of operation. (Can't be changed during operation.)

4-1-4 SIOU Refresh Settings

Item	Settings	Function	Related flags and words	New setting's effectiveness
Cyclic Refreshing of Units 0 to 95	Enabled Disabled Default: Enabled	These settings determine whether data will be exchanged between the specified Unit and the Special I/O Unit's allocated words (10 words/Unit) during cyclic refreshing for Special I/O Units.	---	Start of operation.

4-1-5 Unit Settings

Item	Settings	Function	Related flags and words	New setting's effectiveness
Rack 0, Slot 0	No filter 0.5 ms 1 ms 2 ms 4 ms 8 ms 16 ms 32 ms Default: 8 ms	Sets the input response time (ON response time = OFF response time) for Basic I/O Units. The default setting is 8 ms and the setting range is 0.5 ms to 32 ms. This value can be increased to reduce the effects of chattering and noise, or it can be reduced to allow reception of shorter input pulses.	A220 to A259: Actual input response times for Basic I/O Units	At startup.
Rack 0, Slot 1				
Rack 0, Slot 2				
Rack 0, Slot 3				
Rack 0, Slot 4				
Rack 0, Slot 5				
Rack 0, Slot6				
Rack 0, Slot 7				
Rack 0, Slot 8				
Rack 0, Slot 9				
Rack 1, Slots 0 to 9				
Rack 2, Slots 0 to 9				
Rack 3, Slots 0 to 9				
Rack 4, Slots 0 to 9				
Rack 5, Slots 0 to 9				
Rack 6, Slots 0 to 9				
Rack 7, Slots 0 to 9				

4-1-6 Communications Settings

USB Port Port Settings

Settings	Function	Related flags and words	New setting's effectiveness
Default* Manual Default: Default	*The default settings are for 1 start bit, 7 data bits, even parity, 2 stop bits, and a baud rate of 9,600 bps.		Next cycle. (Also can be changed with STUP (237).)

USB Port Port Mode

Settings	Function	Related flags and words	New setting's effectiveness
Default (SYSWAY) Tool bus SYSWAY	This setting determines which serial communications mode will be used for the USB port.		Next cycle.

USB Port Data Length

Settings	Function	Related flags and words	New setting's effectiveness
7 bits 8 bits Default: 7 bits	These settings are valid only when the communications mode is set to *Host link*. These settings are also valid only when *USB Port Port Settings* is set to *Manual*.		Next cycle.

USB Port Stop Bits

Settings	Function	Related flags and words	New setting's effectiveness
2 bits 1 bit Default: 2 bits	These settings are valid only when the communications mode is set to Host link. These settings are also valid only when *USB Port Port Settings* is set to *Manual*.		Next cycle.

USB Port Parity

Settings	Function	Related flags and words	New setting's effectiveness
Parity - Even Parity - Odd Parity - None Default: Parity - Even	These setting is valid only when the communications mode is set to Host link. These settings are also valid only when *USB Port Port Settings* is set to *Manual*.		Next cycle.

USB Port Baud Rate (bps)

Settings	Function	Related flags and words	New setting's effectiveness
300 600 1,200 2,400 4,800 9,600 19,200 38,400 57,600 115,200 Default: 9,600	This setting is valid only when the communications mode is set to the SYSWAY mode. These settings are also valid only when *USB Port Settings* is set to *Manual*.		Next cycle.

USB Port SYSWAY Mode Unit Number

Settings	Function	Related flags and words	New setting's effectiveness
0 to 31 Default: 0	This setting determines the CPU Unit's unit number when it is connected in a 1-to-N (N=2 to 32) Host Link.		Next cycle.

RS232C Port Port Settings

Settings	Function	Related flags and words	New setting's effectiveness
Default* Manual Default: Default	*The default settings are for 1 start bit, 7 data bits, even parity, 2 stop bits, and a baud rate of 9,600 bps.		Next cycle.

RS232C Port Port Mode

Settings	Function	Related flags and words	New setting's effectiveness
NT Link (1:N) Tool Bus SYSWAY Default: SYSWAY	This setting determines which serial communications mode will be used for the RS232C port.		Next cycle.

RS232C Port Data Length

Settings	Function	Related flags and words	New setting's effectiveness
7 bits 8 bits Default: 7 bits	These settings are valid only when the communications mode is set to SYSWAY.		Next cycle.

RS232C Port Stop Bits

Settings	Function	Related flags and words	New setting's effectiveness
2 bits 1 bit Default: 2 bits	These settings are valid only when the communications mode is set to SYSWAY.		Next cycle.

RS232C Port Parity

Settings	Function	Related flags and words	New setting's effectiveness
Parity - Even Parity - Odd Parity - None Default: Parity - Even	These settings are valid only when the communications mode is set to SYSWAY.		Next cycle.

RS232C Port Baud Rate (bps)

Settings	Function	Related flags and words	New setting's effectiveness
300 600 1,200 2,400 4,800 9,600 19,200 38,400 57,600 115,200 Default: 9,600	These settings are valid only when the communications mode is set to host link.		Next cycle.

RS232C Port SYSWAY Mode Unit Number

Settings	Function	Related flags and words	New setting's effectiveness
0 to 31 Default: 0	This setting determines the CPU Unit's unit number when it is connected in a 1-to-N (N=2 to 32) Host Link.		Next cycle.

RS232C Port NT Link Max.)

Settings	Function	Related flags and words	New setting's effectiveness
0 to 7 Default: 1	This setting determines the highest unit number of PT that can be connected to the PLC.		Next cycle.

4-1-7 FINS Write Protection

FINS Write Protection Settings

Settings	Function	Related flags and words	New setting's effectiveness
Enable or disable FINS write protection Default: Disable	This setting specifies whether or not to enable write protection on FINS commands sent to the CPU Unit through the network. (The write protection does not apply to direct serial connections.)	---	As soon as changed.
The following settings can be used to allow write commands from up to specified nodes in the specified networks. (Write-protection will not apply to the specified nodes.) If the following settings are not made, write-protection will apply to all nodes other than the local node.			
Network address: 0 to 127 Node address: 1 to 255 (255 = all nodes) Default: Network address 0, node address 1	Specifies the network address and node address of the 1st node that is not write-protected.	---	As soon as changed.
:	:	:	:
Network address: 0 to 127 Node address: 1 to 255 (255 = all nodes) Default: Network address 0, node address 1	Specifies the network address and node address of the 32nd node that is not write-protected.	---	As soon as changed.

4-1-8 General Settings

Peripheral Servicing

Settings	Function	Related flags and words	New setting's effectiveness
8%, 16%, 32% Default: 8%	This setting sets the time slice for peripheral servicing as a percentage of the cycle time.	---	Start of operation. (Can't be changed during operation.)

4-2 Ethernet Settings

4-2-1 List of Settings

Group	Setting	Default
TCP/IP Settings	BOOTP	Unused
	IP Address	192.168.200.200
	Network Mask	255.255.255.0
	Default Gateway	0.0.0.0
	Host Name	
	MAC Address	
Unit TCP/IP Settings	Unit ☐ IP Address	0.0.0.0
	Unit ☐ Network Mask	0.0.0.0
IP Routing Settings	Routing Info ☐ Network Address	0.0.0.0
	Routing Info ☐ Network Mask	0.0.0.0
	Routing Info ☐ Gateway IP Address	0.0.0.0
FTP Server Settings	FTP Server	Unused
	Login Name	
	Password	

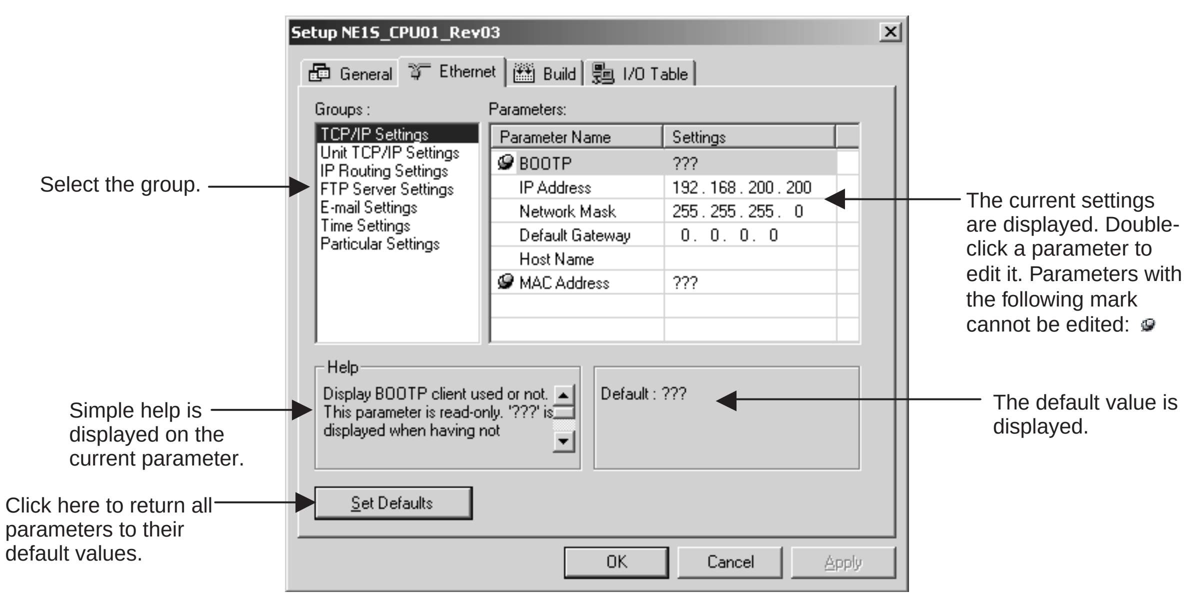
Group	Setting	Default
Email Settings	SMTP Server Address	
	Source Mail Address	
	Destination Mail Address	
	Send Trigger (At Changing)	Disable
	Send Trigger (At Error)	Disable
	Send Trigger (At periodic interval)	Disable
	Send Data (User Information)	Send
	Send Data (Change Log)	Send
	Send Data (Error Log)	Send
	Send Data (Status)	Send
	User Information	
	Posting Interval (Day)	0 Day
	Posting Interval (Hour)	0 Hour
	Posting Interval (Minute)	10 Minute
Time Settings	Use SNTP	Unused
	NTP Server IP Address	0.0.0.0
	Server Access Interval Time (Day)	1 Day
	Server Access Interval Time (Hour)	0 Hour
	Server Access Interval Time (Minute)	0 Minute
Particular settings	Ethernet	Enable
	Communications Zone Control	Enable
	Amount of Communications Zones	Low

4-2-2 Basic Setting Procedures for Ethernet Settings

Ethernet Settings are made on the Ethernet Tab Page of the Configuration Setting Window of NE Programmer. The Ethernet Tab Page is shown below.

1,2,3...

1. Right-click the configuration name (i.e., the PLC name) in the Project Workspace and select ***System Configuration*** from the popup menu. The Configuration Setting Window will be displayed.
2. Click the **Ethernet** Tab in the Configuration Setting Window. The following Ethernet Setting Window will be displayed.

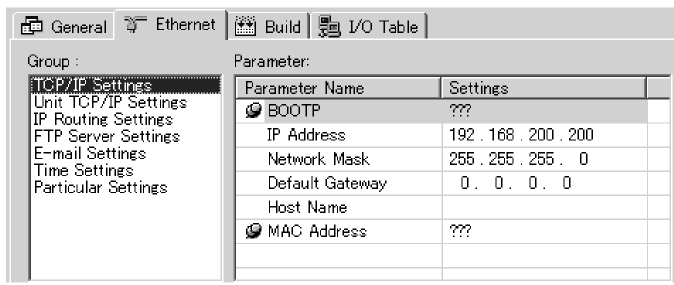


3. Select the group and set the parameters.
4. After completing all settings, click the **OK** Button.
5. Download the settings to the CPU Unit. Select ***Controller - Download to Controller***, select ***Ethernet Setting***, and then click the **OK** Button.
   Refer to *SECTION 8 Software Installation* for information on downloading and other online operations for the NE Programmer.
6. Restart the CPU Unit.

The Ethernet Settings and setting methods are described in the following section by group.

4-2-3 TCP/IP Settings

The settings in the TCP/IP group are shown below.



These settings are described individually below.

BOOTP

This window only displays whether the BOOTP function is enabled or not. The BOOTP setting cannot be changed here.

To use the BOOTP function, set the rotary switches on the font of the CPU Unit to FF.

Note BOOTP
The BOOTP function enables automatically setting the IP address, network mask, and default gateway by obtaining IP configuration information from the BOOTP server on the network.
A BOOTP server must be installed separately on the network.

IP Address

- As a rule, the IP address is set on the rotary switches on the CPU Unit first and then the setting on the Ethernet Setting Tab Page from the NE Programmer is set to match the setting on the rotary switches.
- The network portion of the IP address, however, must be changed using the Ethernet Settings on the NE Programmer.
- If the rotary switches are set to 00, the IP address set in the Ethernet Settings will be used. Set the IP address switches on all CPU Units to 00 when you want to set and manage IP addresses from the NE Programmer.
- If the rotary switches are set to FF, the CPU Unit will use BOOTP.

CPU Unit IP Address Switch Settings

Setting	Meaning
00	• The IP address saved in the flash memory of the CPU Unit will be used at startup. • The defaults stored in flash memory are an IP address of 192.168.200.200, a network mask of 255.255.255.0, and a default gateway of 0.0.0.0 (no gateway). • If the IP address is changed on the Ethernet Setting Tab Page of the NE Programmer and then downloaded to the CPU Unit, the changed settings will be stored in flash memory. The IP address saved in the flash memory of the CPU Unit will be used at startup. **Note** Set the IP address switches on all CPU Units to 00 when you want to set and manage IP addresses from the NE Programmer.
01 to FE	• Only the rightmost 8 bits of the IP address is set on the IP address switches on the CPU Unit. • The default IP address will be from 192.168.200.01 to 192.168.200.254, a network mask of 255.255.255.0, and a default gateway of 0.0.0.0 (no gateway). **Note** 1. The rightmost 8 bits of the IP address set on the Ethernet Settings Tab Page of the NE Programmer will be ignored. 2. As a rule, set the rightmost 8 bits of the IP address on the rotary switches and then set the remaining portion of the IP address from the Ethernet Settings Tab Page of the NE Programmer. The enables reading IP addresses directly off the rotary switches on the CPU Units onsite. • The network portion of the IP address, however, must be changed using the Ethernet Settings Tab Page of the NE Programmer.
FF	The CPU Unit will use BOOTP. **Note** The IP address set from the NE Programmer will be ignored if FF is set.

Note Changes to the Ethernet Settings are effective the next time the CPU Unit is started.

Network Mask

A subnet mask can be set.

Class	Subnet mask
Class A	255.0.0.0
Class B	255.255.0.0
Class C	255.255.255.0

Set the same subnet mask for all nodes on the same subnetwork.

Default Gateway

Select the IP address of the default gateway.

Do not set anything if a default gateway is not being used.

Host Name

Set a host name for the IP address of the CPU Unit.

Do not set anything if a host name is not being used.

MAC Address

The MAC Address of the CPU Unit is displayed.

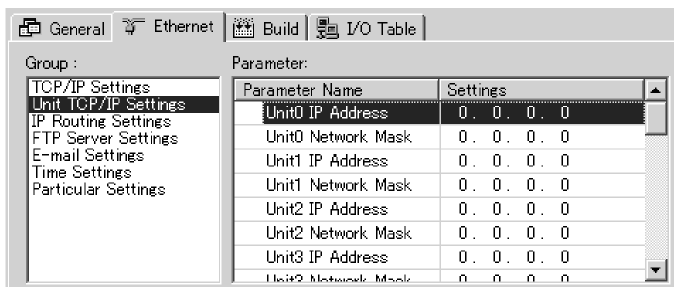
4-2-4 Unit TCP/IP Settings

The Unit TCP/IP group sets IP addresses and network masks for Communications Units.

Note (1) The only Unit that currently uses this setting is the NE1S-CNS21U ControlNet Unit.

(2) The IP Over ControlNet function can be implemented by making these settings for ControlNet Units mounted on the CPU Rack. This will enable FTP transfers and other IP packet communications from a personal computer connected by Ethernet to the CPU Unit for devices that support IP over ControlNet on a ControlNet network.

(3) If these settings are used, IP routing does not have to be set for devices on the same network at the ControlNet Unit. IP routing must be set, however, to communicate with devices on other networks beyond the local network.



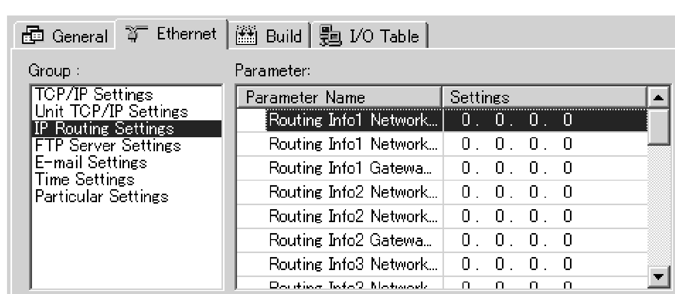
The Unit TCP/IP settings are listed in the following table.

Setting	Meaning	Default
Unit □ IP Address	IP address of the Communications Unit	0.0.0.0
Unit □ Network Mask	Network mask of the Communications Unit	0.0.0.0

4-2-5 IP Routing Settings

Up to 10 sets of IP routing information can be set in the IP routing settings.

Note IP Routing Settings
These settings enable IP packets to be routed according to the IP routing information set in the CPU Unit (routing tables).
These settings are required to communicate with remote networks.



The following addresses must be set for each IP routing path.

- Final IP network address
- Final IP network mask
- IP address of gateway

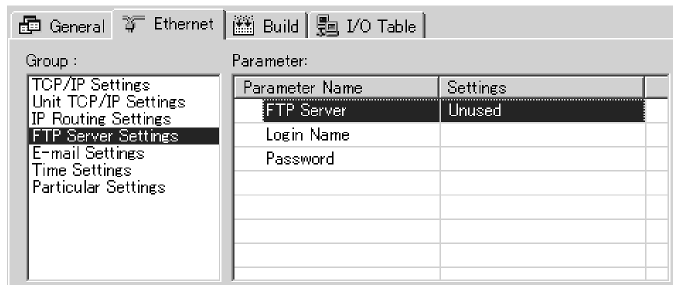
Note (1) IP routing is possible via IP over ControlNet.

(2) An error will occur is any of the following settings are made.
If an IP address not on the local network is set.
If the same setting is made twice.
If two or more gateways are set for the same destination.

(3) If the number of hops used by RIP is not supported.

4-2-6 FTP Server Settings

The settings in the FTP server group are shown below.



These settings are described individually below.

FTP Server

Set the FTP server function to *Used* or *Unused*.

If the FTP server function is set to be used, FTP client software on a personal computer of other device connect to Ethernet can use FTP commands to upload and download data files on Memory Cards.

Note (1) The NE1S CPU Unit does not support an FTP client function. The CPU Unit thus cannot upload or download files at other nodes.

(2) The root directory of the Memory Card is the initial directory when connecting with FTP.

The FTP server supports only one client. The FTP server thus cannot provide services to more than one client at the same time. If the FTP server is already being used, the message *530 Not logged in* will be returned to any other clients requesting connection and the connection will be refused.

Login Name and Password

Set the login name and password for the FTP server.

The following characters can be used for the login name and password.

- Alphanumeric characters, hyphens, and underscores.
- Log names and passwords are not case sensitive.
- The login name must be 16 characters maximum.
- The password must be 16 characters maximum.

Note The default login name is *CONFIDENTIAL*. A password is not set for the default. Logging in is possible merely by inputting the login name *CONFIDENTIAL*.

- FTP can be used via IP over ControlNet.

Supported FTP Commands

The FTP commands that can be sent from the FTP client to the FTP server are listed in the following table.

Command	Function
open	Connects to the FTP server of the specified host.
user	Specifies the user name for the remote FTP server.
ls	Displays a list of files at the remote host.
dir	Displays file names and other information at the remote host.

Command	Function
rename	Changes a file name.
mkdir	Creates a directory in the working directory at the remote host.
rmdir	Deletes a directory in the working directory at the remote host.
cd	Changes the working directory at the remote host to the specified directory.
cdup	Makes the parent directory the working directory.
pwd	Displays the working directory at the remote host.
type	Changes the file transfer type to a specified type.
get	Transfers the specified remote file to the local host.
mget	Transfers the specified remote files to the local host.
put	Transfers the specified local file to the remote host.
mput	Transfers the specified local files to the remote host.
delete	Deletes the specified file at the remote host.
mdelete	Deletes the specified files at the remote host.
close	Disconnects the FTP server.
bye	Ends the FTP client.
quit	Ends the FTP client.

Note
(1) "Remote host" indicates the NE1S.
(2) "Remote file" indicates a file in the NE1S.
(3) "Local host" indicates the host computer (i.e., the FTP client).
(4) "Local file" indicates a file on the host computer (i.e., the FTP client).

4-2-7 Email Settings

Email Settings

The mail settings include the SMTP server address, the destination mail address, send triggers, send data, etc.

Note Email:
The SMTP client function is used to send email from the CPU Unit. An SMTP server must be installed separately on the network.

This function can be used, for example, to send mail to a specified address when a specified condition was satisfied in the CPU Unit.

The following mail send triggers can be used.

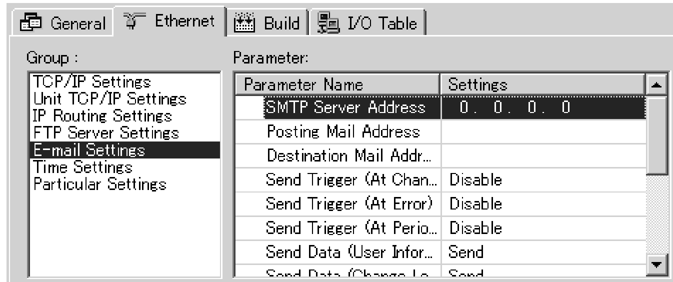
- When executing the SEND MAIL ladder diagram instruction
- When a change in memory occurs
- When an error occurs
- Periodically

The following data can be set as email. Except for data set for the SEND MAIL ladder diagram instruction, each of these can be set to be sent or not sent.

- User-set data (256 characters maximum)
- Changes log (5 records maximum)
- Error log (20 records maximum)
- Status data (CPU Unit status and communications status)
- Data set for SEND MAIL ladder diagram instruction (128 characters maximum when instruction is executed)

Note
(1) Notification of errors recorded in the CPU Unit error log can be sent by email. For details on error log notifications, refer to *14-2-3 Transferring Files by FTP over an Ethernet Network*.

(2) It is not necessary to restart the CPU Unit after downloading email settings. New settings can be enabled by selecting ***Controller - Restart Service***.



The email settings are listed in the following table.

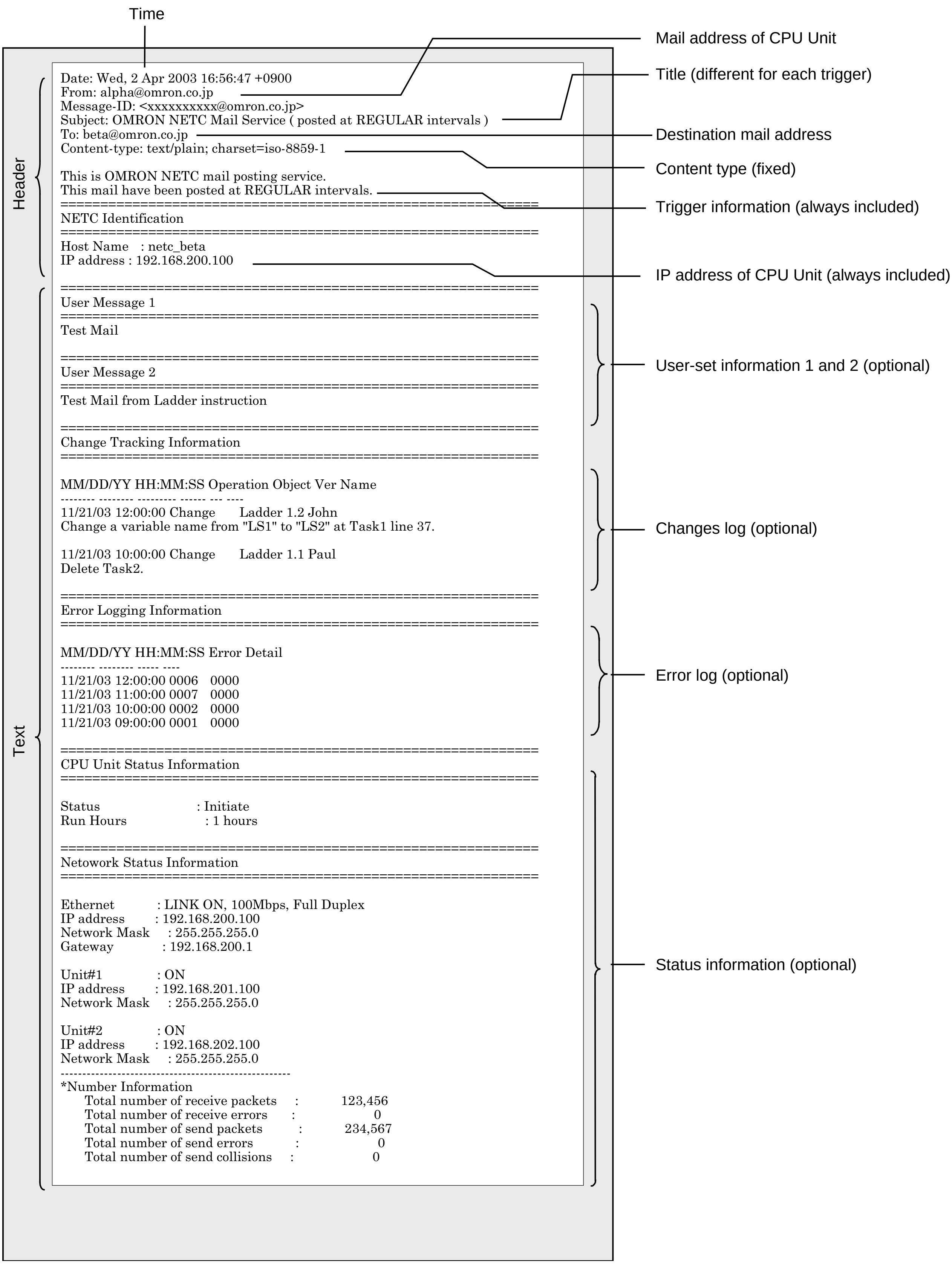
Setting	Contents	Default
SMTP Server Address	IP address of SMTP server	0.0.0.0
Source Mail Address	The mail address of the sender (64 characters maximum) The domain name after @ is required.	
Destination Mail Address	The mail addresses of up to three recipients (200 characters maximum total) Separate addresses with commas. Commas are also counted in the maximum number of characters.	
Send Trigger (At Changing)	More than one trigger can be specified. No trigger needs to be set for the SEND MAIL instruction.	Disable
Send Trigger (At Error)		Disable
Send Trigger (At Periodic Interval)		Disable
Send Data (User Information)	More than one type of data can be specified.	Send
Send Data (Change Log)		Send
Send Data (Error Log)		Send
Send Data (Status)		Send
User Information	256 characters max. If the text string contains 00 hex (the null code), data will be sent only up to just before the null code.	
Posting Interval (Day)	The setting range is from 1 minute to 10 days, 23 hours, 59 minutes The setting interval is 1 minute If 0 min is set, 1 min will be used.	0 Day
Posting Interval (Hour)		0 Hour
Posting Interval (Minute)		10 Minute

Email Format

The email format is described in the following table.

Item		Contents
Header	Date	Year, month, day of month, day of week
	From	This address will be used as the destination address for any response returned to mail sent using the SMTP function. For example, the address of the system administrator can be set so that responses are returned to the administrator.
	Message-ID	An ID attached by the SMTP client
	To	The destination address
	Subject	“OMRON NE1S Mail Service” and the send trigger information
	Content-type	“text/plain; charset=iso-8859-1” (fixed)
Text	Title	“This is OMRON NE1S mail Service.” and the send trigger information
	NE1S ID information	The host name and IP address of the CPU Unit
	User-set information 1 (optional)	User-set information in the SMTP client settings
	User-set information 2 (optional)	User-set information set in the SEND MAIL instruction
	Changes log (optional)	Time of change, change operation, item changed, version, comment, and person making the change (5 records max.)
	Error log (optional)	Time of error and error code (20 records maximum)
	Status (optional)	CPU Unit status and communications status (Ethernet Unit)

Email Example

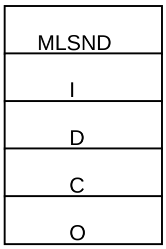



```
Date: Wed, 2 Apr 2003 16:56:47 +0900
From: alpha@omron.co.jp
Message-ID: <xxxxxxxxxx@omron.co.jp>
Subject: OMRON NETC Mail Service ( posted at REGULAR intervals )
To: beta@omron.co.jp
Content-type: text/plain; charset=iso-8859-1

This is OMRON NETC mail posting service.
This mail have been posted at REGULAR intervals.
============================================================
NETC Identification
============================================================
Host Name   : netc_beta
IP address : 192.168.200.100

============================================================
User Message 1
============================================================
Test Mail

============================================================
User Message 2
============================================================
Test Mail from Ladder instruction

============================================================
Change Tracking Information
============================================================

MM/DD/YY HH:MM:SS Operation Object Ver Name
-------- -------- --------- ------ --- ----
11/21/03 12:00:00 Change      Ladder 1.2 John
Change a variable name from "LS1" to "LS2" at Task1 line 37.

11/21/03 10:00:00 Change      Ladder 1.1 Paul
Delete Task2.

============================================================
Error Logging Information
============================================================

MM/DD/YY HH:MM:SS Error Detail
-------- -------- ----- ----
11/21/03 12:00:00 0006   0000
11/21/03 11:00:00 0007   0000
11/21/03 10:00:00 0002   0000
11/21/03 09:00:00 0001   0000

============================================================
CPU Unit Status Information
============================================================

Status                     : Initiate
Run Hours                   : 1 hours

============================================================
Netowork Status Information
============================================================

Ethernet          : LINK ON, 100Mbps, Full Duplex
IP address        : 192.168.200.100
Network Mask      : 255.255.255.0
Gateway           : 192.168.200.1

Unit#1            : ON
IP address        : 192.168.201.100
Network Mask      : 255.255.255.0

Unit#2            : ON
IP address        : 192.168.202.100
Network Mask      : 255.255.255.0
--------------------------------------------------
*Number Information
     Total number of receive packets    :          123,456
     Total number of receive errors     :                0
     Total number of send packets       :          234,567
     Total number of send errors        :                0
     Total number of send collisions    :                0
```

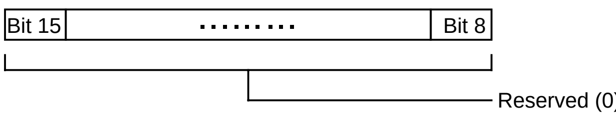
SEND MAIL Instruction (MLSND) Specifications

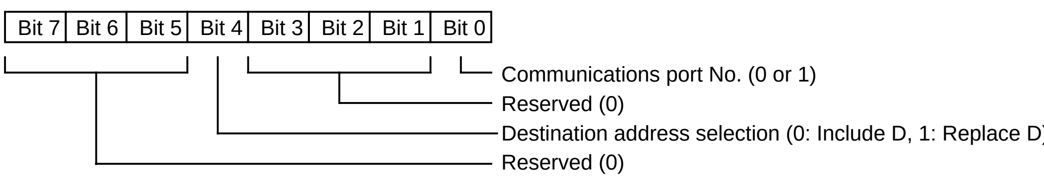
The specifications of the SEND MAIL instruction (MLSND) are given below.

Item	Specifications
Instruction name	SEND MAIL
Mnemonic	MLSND, @MLSND
Function code	795



I: User-set information (128 characters max., STRING data)
D: Destination address (one only, 64 characters max., STRING data)
C: Control data (WORD data)
O: Execution status output (WORD data)
Word addresses can be specified for any of the operands.





- SEND MAIL sends email according to the SMTP client settings and control data (C).
- The following settings can be made using the control data.
  - a. Communications Port
    Communications port 0 or 1
  - b. Destination Address Selection
    0: D is added to the destination address in the SMTP client settings.
    1: The destination address in the SMTP client settings is replaced with D.
- SEND MAIL uses a communications port for information communications instructions (internal logical ports A20213 and A20214).
- The following codes will be output to internal memory to indicate execution status.

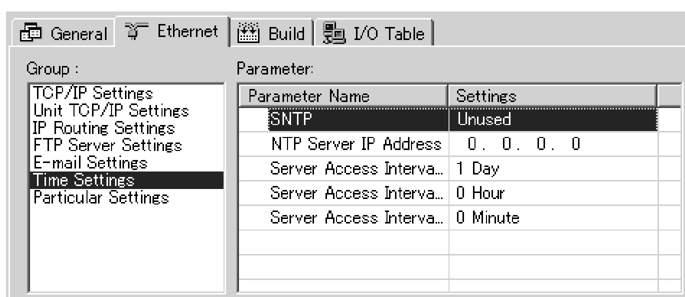
Normal end:	0000 hex
Parameter error:	0100 hex
Failed to obtain communications port:	0101 hex
SMTP setting error:	0102 hex
SMTP communications error:	0103 hex

The following table shows the details of the execution status codes.

Code	Description
0000 hex (normal end)	The instruction was completed normally. **Note** The normal end code will still be returned if the destination address does not exist in another domain (different from the domain set in the SMTP Server Address setting). In this case, another SMTP server manages the non-existent email address error, not the SMTP server set in the SMTP Server Address setting.
0100 hex (parameter error	The length of the text string in operand S1 or S2 exceeds the maximum allowed length.
0101 hex (failed to obtain communications port)	The communications port specified in control word C (A20213 or A20214) is OFF.
0102 hex (SMTP setting error)	There is an invalid setting in the Ethernet email settings. 1. The SMTP server's IP address is invalid. This can occur in the following cases: •Class setting is a class other than A, B, or C. •The loopback address (127.xxx) is specified. •The rightmost 8 bits of the address are set to 0x00 or 0xFF. 2. The sender's mail address does not contain a domain name (after the @ character). 3. There are more than three destination mail addresses.

4-2-8 Time Settings

The settings in the Time group are shown below.



Note Time Settings
With this function, time information is obtained from the NTP server on the network and automatically set in the internal clock in the CPU Unit. An NTP server must be installed separately on the network.
The CPU Unit does not provide a SNTP (Simple Network Time Protocol) client. It uses the **NTP (Network Time Protocol) server**.

The Time settings are listed in the following table.

Setting	Meaning	Default
SNTP	Used or Unused	Unused
NTP Server IP Address	The IP address of the NTP server	0.0.0.0
Server Access Interval Time (Day)	Setting range: 1 min to 1 day	1 Day
Server Access Interval Time (Hour)		0 Hour
Server Access Interval Time (Minute)		0 Minute

The SNTP client operates as follows:

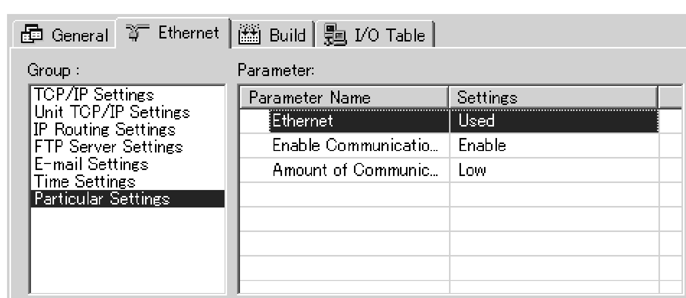
- An SNTP request is sent when power is turned ON to the CPU Unit or when the service is restarted. (The service can be restarted using ***Controller - Restart Service***.).
- Thereafter, an SNTP request is sent periodically according to the set access interval.
- Two retries will be made for each send. The total timeout time is 6 seconds.

Note It is not necessary to restart the CPU Unit after downloading email settings. New settings can be enabled by selecting ***Controller - Restart Service***.

4-2-9 Particular Settings

Particular settings can be used restrict TCP/IP and Ethernet activity to limit the affect of Ethernet communications on the execution cycle of ladder programming in the CPU Unit. Ethernet can be turned ON or OFF and the Ethernet communications processing time can be changed.

The settings in the Particular Setting group are shown below.



The Particular settings are listed in the following table.

Setting	Meaning	Default
Ethernet	Used or Unused	Used
Communication Zone Control	Enable or Disable Can be set only when *Ethernet* is set to *Used*.	Used
Amount of Communication Zones	Low, Middle, or High Can be set only when *Enable communication zone control* is set to *Used*.	Middle

Setting Example for Communications Zone Control

1,2,3...

1. Use the following setting to not use Ethernet and thus reduce the ladder program execution time.

   Ethernet: Set to *Unused*.

2. Use the following setting to use Ethernet but to also reduce the ladder program execution time.

   Ethernet: Set to *Used.*
   Enable communications zone control: Set to *Used*.
   Amount of communication zones: Set to *Low*.

3. Use the following setting to use Ethernet without concern for the affect on the ladder program execution time.

   Ethernet: Set to *Used.*
   Enable communications zone control: Set to *Used*.
   Amount of communication zones: Set to *Medium* or *High*.

4. Use the following setting to use Ethernet with maximum communications performance.

   Ethernet: Set to *Used*.
   Enable communications zone control: Set to *Unused*.

- These settings do not apply to communications via IP over ControlNet.
- The *Amount of communication zones* settings have the following affects on the ladder program cycle time.
  Low: 400 µs max.
  Medium: 1.6 ms max.
  High: 3.2 ms max.

4-2-10 Ethernet Error Codes

Ethernet Error Status

Error code	Error factor code	Error status	Meaning	Indicators
Communi-cations interface error 00B0	0001	IP address setting error	The IP address or network mask in the CPU Unit is wrong. BOOTP could not be accessed.	MS flashing red
	0004	ControlNet Unit IP address setting error	The IP address or network mask in the ControlNet Unit is wrong.	
	0008	FTP server setting error	There is an illegal character in FTP server login name or password.	
	0020	SMTP client setting error	The IP address or destination mail address set for the SMTP server is wrong. More than 3 mail destinations have been specified.	
	0040	SNTP client setting error	The IP address of the NTP server is wrong.	
	0080	ControlNet Unit setting error	The IP over ControlNet settings in the ControlNet Unit exceed specifications.	
	0100	IP routing setting error	An IP address that is not on the local network was set. The same setting was made twice. Two or more gateways were set for the same destination.	
	1000	Duplicated IP address	Another node with the same IP address exists on the same network.	
	2000	IP address was changed during operation	The IP address switches were changed after the NE1S was started.	
	8000	Ethernet communications error	Refer to the following table under *Ethernet Communications Errors.*	

Ethernet Communications Errors

Error code	Error factor code	Detailed error code	Error status	Meaning	Indicators
Communi-cations interface error 00B0	8000	0001	SMTP communications error	Could not connect or communicate with SMTP server.	MS flashing red
		0002	SNTP communications error	Could not connect or communicate with SNTP server.	
		0004	FINS/TCP connection error	Access was attempted such that the maximum number of connections was exceeded.	
		0008	FINS/TCP send error	Could not send an FINS/TCP frame.	
		0010	FINS/TCP receive error	Received an incorrect FINS/TCP frame.	
		0020	FINS/UDP send error	Could not send an FINS/UDP frame.	
		0040	FINS/UDP receive error	Received an incorrect FINS/UDP frame.	

4-3 Build Settings

Area	Setting	Default
Area for Global/Program	Timer Start Address	1024
	Timer Number	1024
	Counter Start Address	1024
	Counter Number	1024
Area for FB	Timer Start Address for FB	2048
	Timer Number for FB	2048
	Number of Timer Variables Reserved for Online Editing	1
	Counter Start Address for FB	2048
	Counter Number for FB	2048
	Number of Counter Variables Reserved for Online Editing	1
General Settings	IR/DR Area Shared Between Tasks	Independent

SECTION 5
Memory Areas

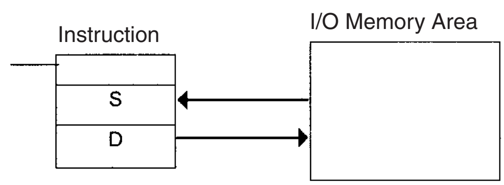
This section describes the structure and functions of the I/O Memory Areas and Parameter Areas.

5-1 Introduction

The CPU Unit's memory (RAM with battery back-up) can be divided into three parts: the User Program Memory, I/O Memory Area, and Parameter Area. This section describes the I/O Memory Area and Parameter Area.

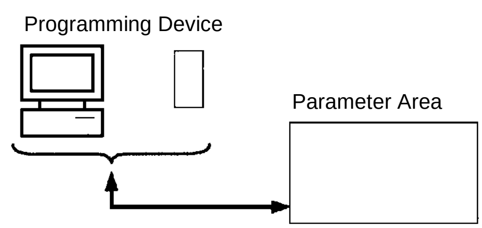
I/O Memory Area

This region of memory contains the data areas which can be accessed by instruction operands. The data areas include the CIO Area, Work Area, Holding Area, Auxiliary Area, DM Area, EM Area, Timer Area, Counter Area, Task Flag Area, Data Registers, Index Registers, Condition Flag Area, and Clock Pulse Area.



Parameter Area

This region of memory contains various settings that cannot be specified by instruction operands; they can be specified from a Programming Device only. The settings include the PLC Setup, Registered I/O Tables, and Ethernet Setup.



5-2 I/O Memory Areas

5-2-1 I/O Memory Area Structure

The following table shows the basic structure of the I/O Memory Area.

Area		Size	Range	External I/O allocation	Bit access	Word access	Access		Change from Programming Device	Status at startup or mode change	Forcing bit status
							Read	Write			
CIO Area	I/O Area	5,120 bits (320 words)	CIO 0000 to CIO 0319 (See note 1.)	Basic I/O Units	OK	OK	OK	OK	OK	Cleared	OK
	CPU Bus Unit Area	6,400 bits (400 words)	CIO 1500 to CIO 1899	CPU Bus Units	OK	OK	OK	OK	OK		OK
	Special I/O Unit Area	15,360 bits (960 words)	CIO 2000 to CIO 2959	Special I/O Units	OK	OK	OK	OK	OK		OK
	DeviceNet Area	9,600 bits (600 words)	CIO 3200 to CIO 3799	DeviceNet Slaves	OK	OK	OK	OK	OK		OK
	Internal I/O Areas	37,504 bits (2,344 words) 4,800 bits (300 words)	CIO 1200 to CIO 1499 CIO 3800 to CIO 6143	---	OK	OK	OK	OK	OK		OK
Work Area		8,192 bits (512 words)	W000 to W511	---	OK	OK	OK	OK	OK	Cleared	OK
Holding Area		8,192 bits (512 words)	H000 to H511	---	OK	OK	OK	OK	OK	Maintained	OK
Auxiliary Area		15,360 bits (960 words)	A000 to A447	---	OK	OK	OK	No	No	Varies from address to address.	No
			A448 to A959					OK	OK		
TR Area		16 bits	TR0 to TR15	---	OK	---	OK	OK	No	Cleared	No
DM Area		32,768 words	D00000 to D32767	---	No (See note 2.)	OK	OK	OK	OK	Maintained	No
EM Area		32,768 words (per bank, 1 banks max.)	E0_00000 to E0_32767	---	No (See note 2.)	OK	OK	OK	OK	Maintained	No
Timer Completion Flags		4,096 bits	T0000 to T4095	---	OK	---	OK	OK	OK	Cleared	OK
Counter Completion Flags		4,096 bits	C0000 to C4095	---	OK	---	OK	OK	OK	Maintained	OK
Timer PVs		4,096 words	T0000 to T4095	---	---	OK	OK	OK	OK	Cleared	No (See note 4.)
Counter PVs		4,096 words	C0000 to C4095	---	---	OK	OK	OK	OK	Maintained	No (See note 5.)
Task Flag Area		32 bits	TK00 to TK127	---	OK	---	OK	No	No	Cleared	No
Index Registers (See note 3.)		16 registers	IR0 to IR15	---	OK	OK	Indirect addressing only	Specific instructions only	No	Cleared	No
Data Registers (See note 3.)		16 registers	DR0 to DR15	---	No	OK	OK	OK	No	Cleared	No

Note

(1) The I/O Area can be expanded to CIO 0000 to CIO 0999 by changing the first words allocated to Racks.

(2) Bits can be manipulated using TST(350), TSTN(351), SET, SETB(532), RSTB(533), OUTB(534).

(3) Index registers and data registers can be used either individually by task or they can be shared by all the tasks.

(4) Timer PVs can be refreshed indirectly by forced setting/resetting Timer Completion Flags.

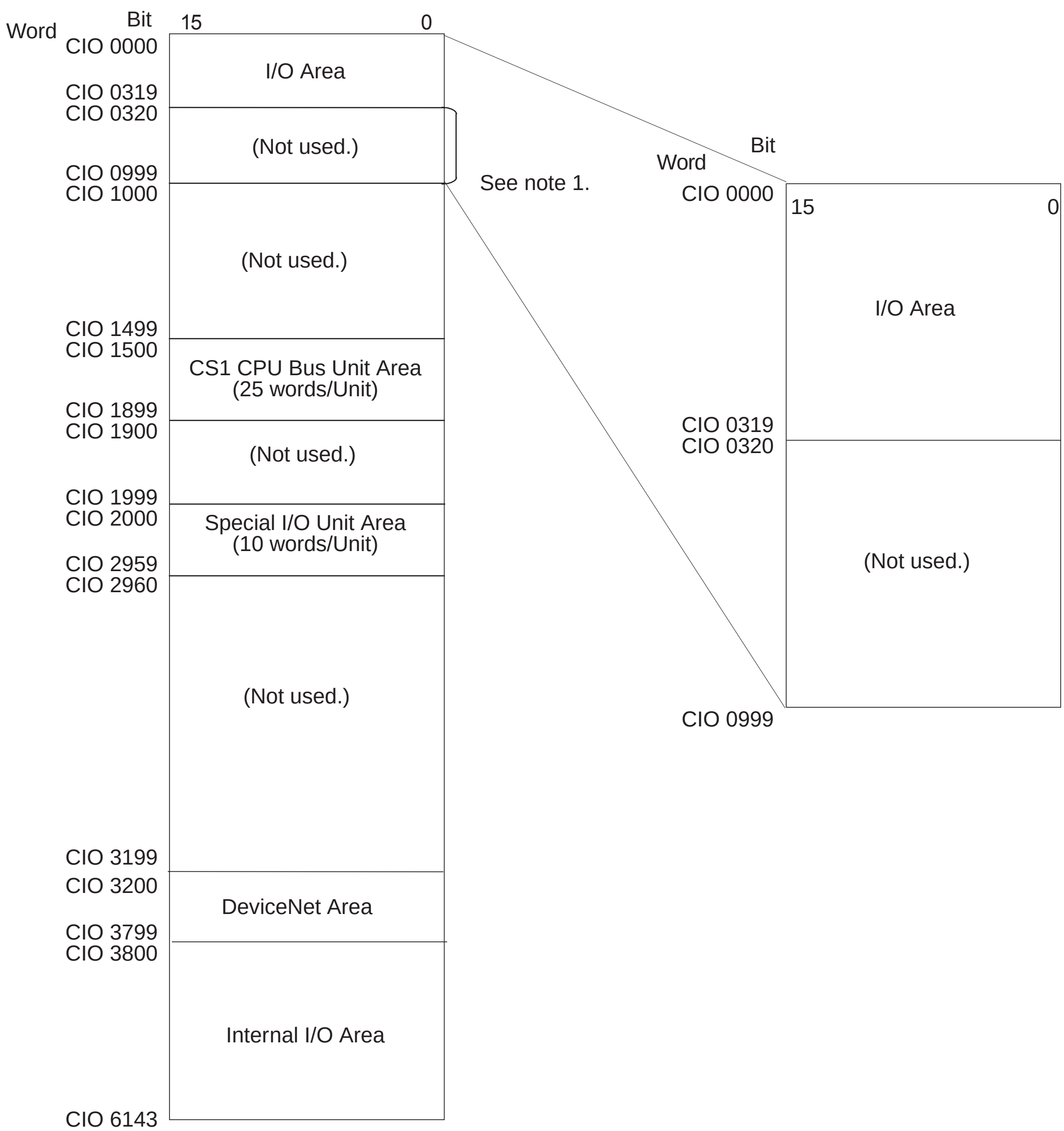
(5) Counter PVs can be refreshed indirectly by forced setting/resetting Counter Completion Flags.

5-2-2 Overview of the Data Areas

The data areas in the NE1S-series I/O Memory Area are described in detail below.

CIO Area

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



Note (1) It is possible to use CIO 0320 to CIO 0999 for I/O words by making the appropriate settings for the first words on the Racks. Settings for the first words on the Racks can be made using the NE Programmer to set the first Rack addresses in the I/O tables. The setting range for the first Rack addresses is from CIO 0000 to CIO 0900.

(2) The parts of the CIO Area that are labelled “Not used” are generally not used, but may be used in programming as work bits. In the future, however, unused CIO Area bits may be used when expanding functions. Always use Work Area bits first.

I/O Area

These words are allocated to external I/O terminals on Basic I/O Units. Words that aren’t allocated to external I/O terminals may be used only in the program.

CPU Bus Unit Area

These words are allocated to CPU Bus Units to transfer status information. Each Unit is allocated 25 words and up to 16 Units (with unit numbers 0 to 15) can be used. Words that aren’t used by CPU Bus Units may be used only in the program.

Special I/O Unit Area

These words are allocated to NE1S-series Special I/O Units. Each Unit is allocated 10 words and up to 96 Units (unit numbers 0 to 95) can be used. Words that aren’t used by Special I/O Units may be used only in the program.

DeviceNet Area

These words are allocated to Slaves for DeviceNet Remote I/O Communications for DeviceNet Units (NE1S-DRM21U). Allocations are fixed and cannot be changed.

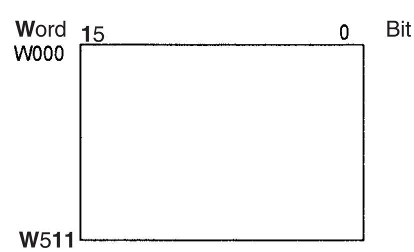
Internal I/O Area

These words can be used only in the program; they cannot be used for I/O exchange with external I/O terminals. Be sure to use the work words provided in the Work Area (WR) before allocating words in the Internal I/O Area or other unused words in the CIO Area.

Note It is possible that these words will be assigned to new functions in future versions of NE1S-series CPU Units, so the program may have to be changed before being used in a new NE1S-series PLC if CIO Area words are used as work words in the program. The function of words listed as “Not used” is basically the same.

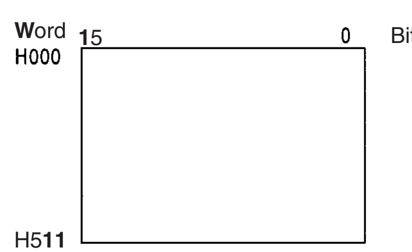
Work Area (WR)

Words in the Work Area can be used only in the program; they cannot be used for I/O exchange with external I/O terminals. No new functions will be assigned to this area in future versions of NE1S-series PLCs, so use this area for work words and bits before any words in the CIO Area.



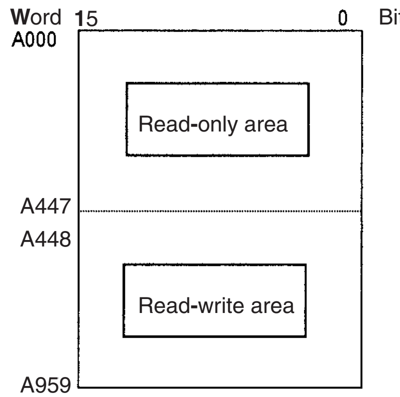
Holding Area (HR)

Words in the Holding Area can be used only in the program. 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.



Auxiliary Area (AR)

The Auxiliary Area contains flags and control bits used to monitor and control PLC operation. This area is divided into two parts: A000 to A447 are read-only and A448 to A959 can be read or written. Refer to *5-9 Auxiliary Area* for details on the Auxiliary Area.

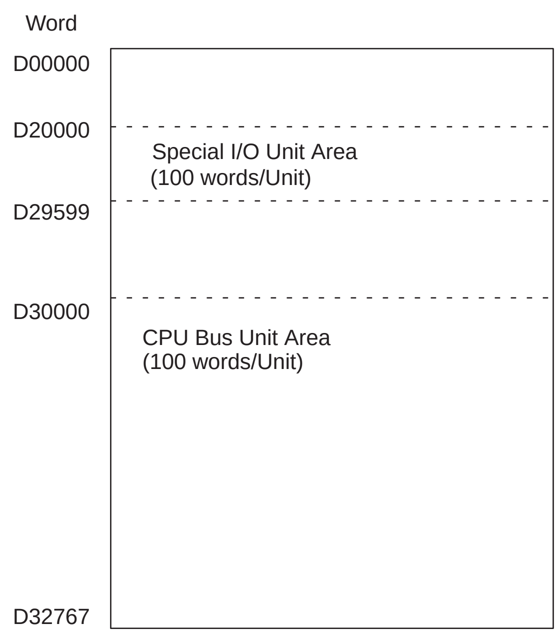


Temporary Relay Area (TR)

The TR Area contains bits that record the ON/OFF status of program branches. The TR bits are used with mnemonics only.

Data Memory Area (DM)

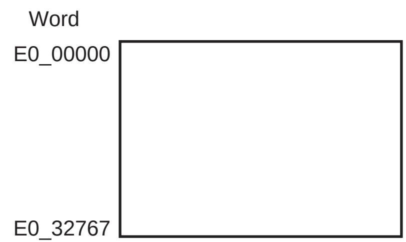
The DM Area is a multi-purpose data area that can be accessed in word-units only. These words retain their content when the PLC is turned on or the operating mode is switched between PROGRAM mode and RUN or MONITOR mode.



Extended Data Memory Area (EM)

The EM Area is a multi-purpose data area that can be accessed in word-units only. 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 EM Area is divided into 32,767-word regions called banks. The number of banks that can be used depends on the CPU Unit model. Refer to *1-2 Specifications* for details.



Timer Area

There are two timer data areas, the Timer Completion Flags and the Timer Present Values (PVs). Up to 4,096 timers with timer numbers T0000 to T4095 can be used. The same number is used to access a timer's Completion Flag and PV.

Timer Completion Flags

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

Timer PVs

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

Counter Area

There are two counter data areas, the Counter Completion Flags and the Counter Present Values (PVs). Up to 4,096 counters with counter numbers C0000 to C4095 can be used. The same number is used to access a counter's Completion Flag and PV.

Counter Completion Flags

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

Counter PVs

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

Condition Flags

These flags include the Arithmetic Flags such as the Error Flag and Equals Flag which indicate the results of instruction execution as well as the Always ON and Always OFF Flags. The Condition Flags are specified with labels (symbols) rather than addresses.

Clock Pulses

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

Task Flag Area (TK)

Task Flags range from TK00 to TK127 and correspond to cyclic tasks 0 to 127. A Task Flag will be ON when the corresponding cyclic task is in executable (RUN) status and OFF when the cyclic task hasn't been executed (INI) or is in standby (WAIT) status.

Index Registers (IR)

These registers (IR0 to IR15) are used to store PLC memory addresses (absolute memory addresses in RAM) to indirectly address words in I/O memory. The Index Registers can be used separately in each task or they can be shared by all tasks.

Data Registers (DR)

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

5-2-3 Data Area Properties

Content after Fatal Errors, Forced Set/Reset Usage

Area		External allocation	Fatal Error Generated				Forced Set/ Forced Reset Functions Usable?
			Execution of FALS (007)		Other Fatal Error		
			IOM Hold Bit OFF	IOM Hold Bit ON	IOM Hold Bit OFF	IOM Hold Bit ON	
CIO Area	I/O Area	Basic I/O Units	Retained	Retained	Cleared	Retained	Yes
	CPU Bus Units	CPU Bus Units					
	Special I/O Unit Area	Special I/O Units					
	DeviceNet Area	DeviceNet Slaves or Master					
	Internal I/O Area	None					
Work Area (W)		None	Retained	Retained	Cleared	Retained	Yes
Holding Area (H)			Retained	Retained	Retained	Retained	Yes
Auxiliary Area (A)			Status varies from address to address.				No
Data Memory Area (D)			Retained	Retained	Retained	Retained	No
Extended Data Memory Area (E)			Retained	Retained	Retained	Retained	No
Timer Completion Flags (T)			Retained	Retained	Cleared	Retained	Yes
Timer PVs (T)			Retained	Retained	Cleared	Retained	No
Counter Completion Flags (C)			Retained	Retained	Retained	Retained	Yes
Counter PVs (C)			Retained	Retained	Retained	Retained	No
Task Flags (TK)			Cleared	Cleared	Retained	Retained	No
Index Registers (IR)			Retained	Retained	Cleared	Retained	No
Data Registers (DR)			Retained	Retained	Cleared	Retained	No

Content after Mode Change or Power Interruption

Area		Mode Changed[1]		PLC Power OFF to ON			
				IOM Hold Bit Cleared[2]		IOM Hold Bit Held[2]	
		IOM Hold Bit OFF	IOM Hold Bit ON	IOM Hold Bit OFF	IOM Hold Bit ON	IOM Hold Bit OFF	IOM Hold Bit ON
CIO Area	I/O Area	Cleared	Retained	Cleared	Cleared	Cleared	Retained
	CPU Bus Units						
	Special I/O Unit Area						
	DeviceNet Area						
	Internal I/O Area						
Work Area (W)		Cleared	Retained	Cleared	Cleared	Cleared	Retained
Holding Area (H)		Retained	Retained	Retained	Retained	Retained	Retained
Auxiliary Area (A)		Status varies from address to address.					
Data Memory Area (D)		Retained	Retained	Retained	Retained	Retained	Retained
Extended Data Memory Area (E)		Retained	Retained	Retained	Retained	Retained	Retained
Timer Completion Flags (T)		Cleared	Retained	Cleared	Cleared	Cleared	Retained
Timer PVs (T)		Cleared	Retained	Cleared	Cleared	Cleared	Retained
Counter Completion Flags (C)		Retained	Retained	Retained	Retained	Retained	Retained
Counter PVs (C)		Retained	Retained	Retained	Retained	Retained	Retained
Task Flags (TK)		Cleared	Cleared	Cleared	Cleared	Cleared	Cleared
Index Registers (IR)		Cleared	Retained	Cleared	Cleared	Cleared	Retained
Data Registers (DR)		Cleared	Retained	Cleared	Cleared	Cleared	Retained

Note (1) Mode changed from PROGRAM to RUN/MONITOR or vice-versa.

(2) The PLC Setup's "IOM Hold Bit Status at Startup" setting determines whether the IOM Hold Bit's status is held or cleared when the PLC is turned ON.

5-3 CIO Area

I/O Area addresses range from CIO 0000 to CIO 0319 (CIO bits 000000 to 031915), but the area can be expanded to CIO 0000 to CIO 0999 by changing the first Rack word with any Programming Device. The maximum number of bits that can be allocated for external I/O will still be 5,120 (320 words) even if the I/O Area is expanded.

Words in the I/O Area can be allocated to I/O terminals on Basic I/O Units.

Words are allocated to Basic I/O Units based on the slot position (left to right) and number of words required. The words are allocated consecutively and empty slots are skipped. Words in the I/O Area that aren't allocated to Basic I/O Units can be used only in the program.

I/O Area Initialization

The contents of the I/O Area will be cleared in the following cases:

1,2,3...

1. The operating mode is changed from PROGRAM to RUN or MONITOR mode or vice-versa and the IOM Hold Bit is OFF.
(See the following explanation of IOM Hold Bit Operation.)
2. The PLC's power supply is cycled and the IOM Hold Bit is OFF or not protected in the PLC Setup.
(See the following explanation of IOM Hold Bit Operation.)
3. The I/O Area is cleared from a Programming Device.
4. PLC operation is stopped when a fatal error other than an FALS(007) error occurs. (The contents of the I/O Area will be retained if FALS(007) is executed.)

IOM Hold Bit Operation

If the IOM Hold Bit (A50012) is ON, the contents of the I/O Area won't be cleared when a fatal error occurs (including those produced with the FALS(007) instruction) or the operating mode is changed from PROGRAM mode to RUN or MONITOR mode or vice-versa. The I/O Area will be cleared when the power is interrupted.

If the IOM Hold BIt (A50012) is ON and the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, the contents of the I/O Area won't be cleared when the PLC's power supply is cycled. All I/O bits, including outputs, will retain the status that they had before the PLC was turned OFF.

Note If the I/O Hold Bit is turned ON, the outputs from the PLC will not be turned OFF and will maintain their previous status when the PLC is switched from RUN or MONITOR mode to PROGRAM mode. Make sure that the external loads will not produce dangerous conditions when this occurs. (When operation stops for a fatal error, including those produced with the FALS(007) instruction, all outputs from Output Unit will be turned OFF and only the internal output status will be maintained.)

Forcing bit Status

Bits in the I/O Area can be force-set and force-reset.

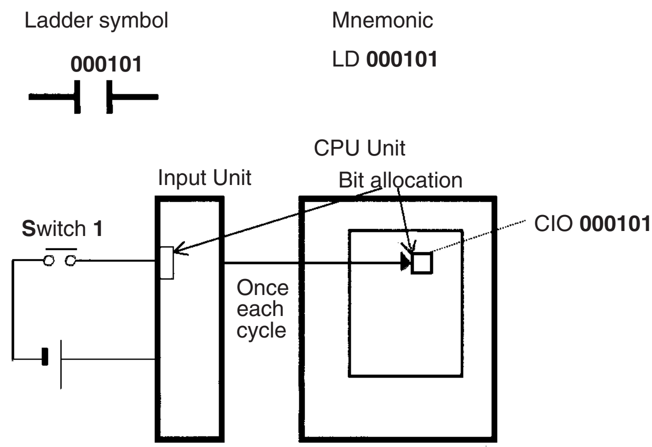
Input Bits

A bit in the I/O Area is called an input bit when it is allocated to an Input Unit. Input bits reflect the ON/OFF status of devices such as pushbutton switches, limit switches, and photoelectric switches. There are three ways for the status of input points to be refreshed in the PLC: normal I/O refreshing, immediate refreshing, and IORF(097) refreshing.

Normal I/O Refreshing

The status of I/O points on external devices is read once each cycle after program execution.

In the following example, CIO 000101 is allocated to switch 1, an external switch connected to the input terminal of an Input Unit. The ON/OFF status of switch 1 is reflected in CIO 000101 once each cycle.



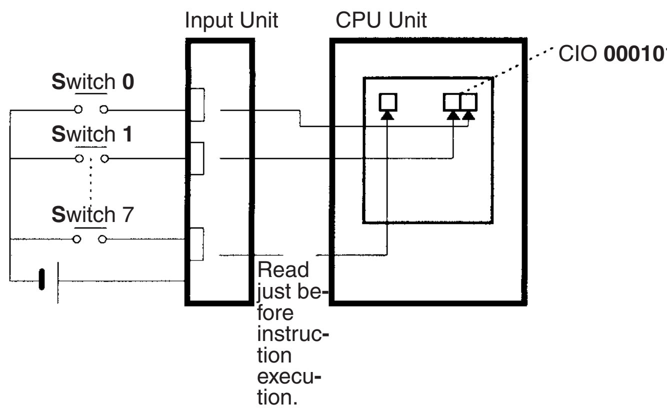
Immediate Refreshing

When the immediate refreshing variation of an instruction is specified by inputting an exclamation point just before the instruction, and the instruction's operand is an input bit or word, the word containing the bit or the word itself will be refreshed just before the instruction is executed. This immediate refreshing is performed in addition to the normal I/O refreshing performed once each cycle.

1,2,3...

1. Bit Operand
   Just before the instruction is executed, the ON/OFF status of the 16 I/O points allocated to the word containing the specified bit will be read to the PLC.
2. Word Operand
   Just before the instruction is executed, the ON/OFF status of the 16 I/O points allocated to the specified word will be read to the PLC.

In the following example, CIO 000101 is allocated to switch 1, an external switch connected to the input terminal of an Input Unit. The ON/OFF status of switch 1 is read and reflected in CIO 000101 just before !LD 000101 is executed.



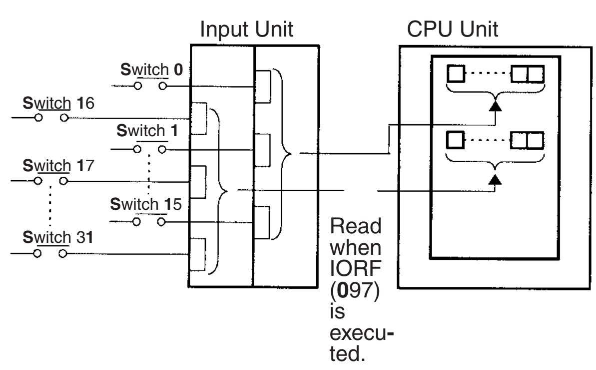
IORF(097) Refreshing

When IORF(097) (I/O REFRESH) is executed, the input bits in the specified range of words are refreshed. This I/O refreshing is performed in addition to the normal I/O refreshing performed once each cycle.

The following IORF(097) instruction refreshes the status of all I/O points in I/O Area words CIO 0000 to CIO 0003. The status of input points is read from the Input Units and the status of output bits is written to the Output Units.

IORF
0000
0003

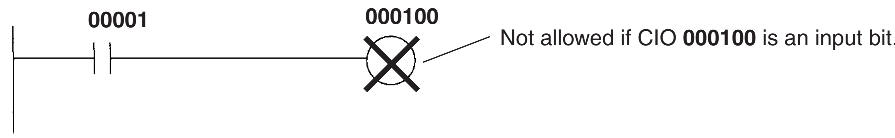
In the following example, the status of input points allocated to CIO 0000 and CIO 0001 are read from the Input Unit. (CIO 0002 and CIO 0003 are allocated to Output Units.)



Limitations on Input Bits

There is no limit on the number of times that input bits can be used as normally open and normally closed conditions in the program and the addresses can be programmed in any order.

An input bit cannot be used as an operand in an Output instruction.

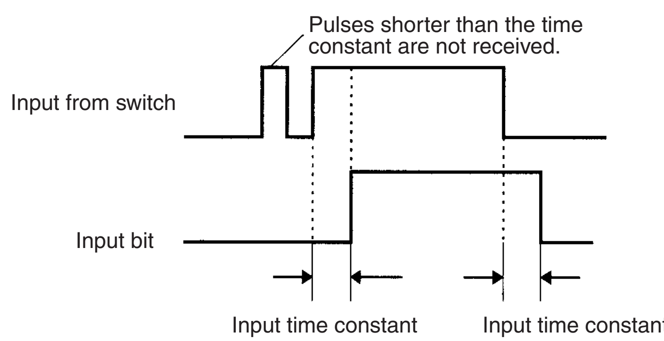


Input Response Time Settings

The input response times for each NE1S-series Basic Input Unit can be set in the PLC Setup. Increasing the input response time will reduce chattering and the effects of noise and decreasing the input response time allows higher speed input pulses to be received (as long as the ON or OFF time is longer than the cycle time).

The default value for input response times is 8 ms and the setting range is 0 to 32 ms.

Note If the time is set to 0 ms, there will still be an ON delay time of 20 µs max. and an OFF delay time of 300 µs due to delays caused by internal elements.



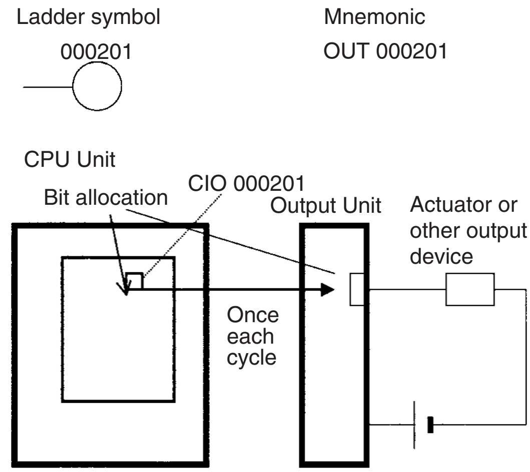
Output Bits

A bit in the I/O Area is called an output bit when it is allocated to an Output Unit. The ON/OFF status of an output bits are output to devices such as actuators. There are three ways for the status of output bits to be refreshed to an Output Unit: normal I/O refreshing, immediate refreshing, and IORF(097) refreshing.

Normal I/O Refreshing

The status of output bits are output to external devices once each cycle after program execution.

In the following example, CIO 000201 is allocated to an actuator, an external device connected to an output terminal of an Output Unit. The ON/OFF status of CIO 000201 is output to that actuator once each cycle.



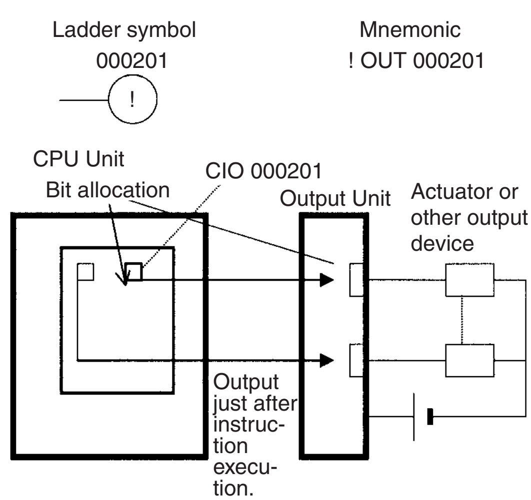
Immediate Refreshing

When the immediate refreshing variation of an instruction is specified by inputting an exclamation point just before the instruction, and the instruction's operand is an output bit or word, the content of the word containing the bit or the word itself will be output just after the instruction is executed. This immediate refreshing is performed in addition to the normal I/O refreshing performed once each cycle.

1,2,3...

1. Bit Operand
   Just after the instruction is executed, the ON/OFF status of the 16 I/O points allocated to the word containing the specified bit will be output to the output device(s).
2. Word Operand
   Just after the instruction is executed, the ON/OFF status of the 16 I/O points allocated to the specified word will be output to the output device(s).

In the following example, CIO 000201 is allocated to an actuator, an external device connected to the output terminal of an Output Unit. The ON/OFF status of CIO 000201 is output to the actuator just after !OUT 000201 is executed.

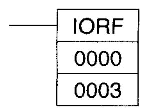


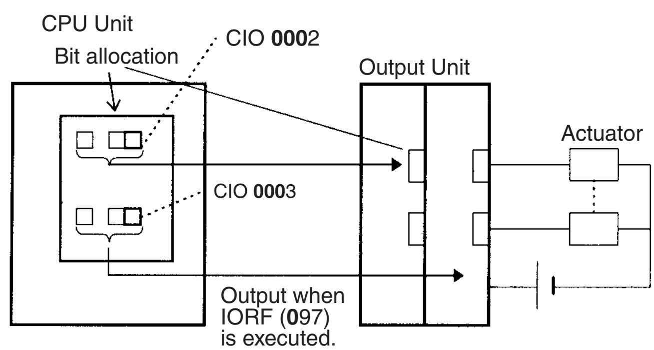
IORF(097) Refreshing

When IORF(097) (I/O REFRESH) is executed, the ON/OFF status of output bits in the specified range of words is output to their external devices. This I/O refreshing is performed in addition to the normal I/O refreshing performed once each cycle.

The following IORF(097) instruction refreshes the status of all I/O points in I/O Area words CIO 0000 to CIO 0003. The status of input points is read from the Input Units and the status of output bits is written to the Output Units.

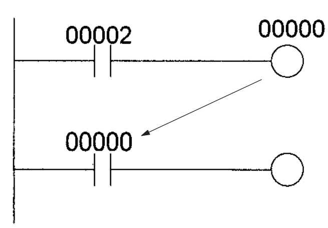
In this example, the status of input points allocated to CIO 0002 and CIO 0003 are output to the Output Unit. (CIO 0000 and CIO 0001 are allocated to Input Units.)



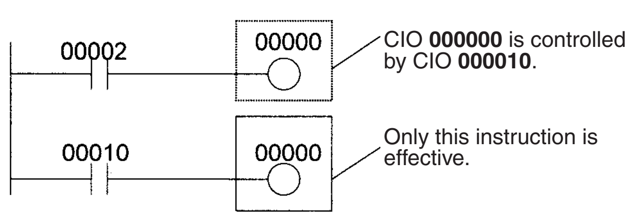


Limitations on Output Bits

Output bits can be programmed in any order. Output bits can be used as operands in Input instructions and there is no limit on the number of times that an output bit is used as a normally open and normally closed condition.



An output bit can be used in only one Output instruction that controls its status. If an output bit is used in two or more Output instructions, only the last instruction will be effective.



Note All outputs on Basic I/O Units and Special I/O Units can be turned OFF by turning ON the Output OFF Bit (A50015). The status of the output bits won't be affected even though the actual outputs are turned OFF.

5-4 DeviceNet Area

The DeviceNet Area addresses run from CIO 3200 to CIO 3799 (600 words).

Words in the DeviceNet Area are used for fixed allocations to Slaves for DeviceNet remote I/O communications for the DeviceNet Unit (NE1S-DRM21U).

The Fixed Allocation Setting Switches 1 to 3 (Software Switches) in the CIO Area words allocated to the DeviceNet Unit determine which fixed allocation words are used.

Area	Master to Slave (Output Words)	Slave to Master (Input Words)
Fixed Allocation Area 1	CIO 3200 to CIO 3263	CIO 3300 to CIO 3363
Fixed Allocation Area 2	CIO 3400 to CIO 3463	CIO 3500 to CIO 3563
Fixed Allocation Area 3	CIO 3600 to CIO 3663	CIO 3700 to CIO 3763

Note If the DeviceNet Unit is set to use the I/O slave function, the following words are also allocated.

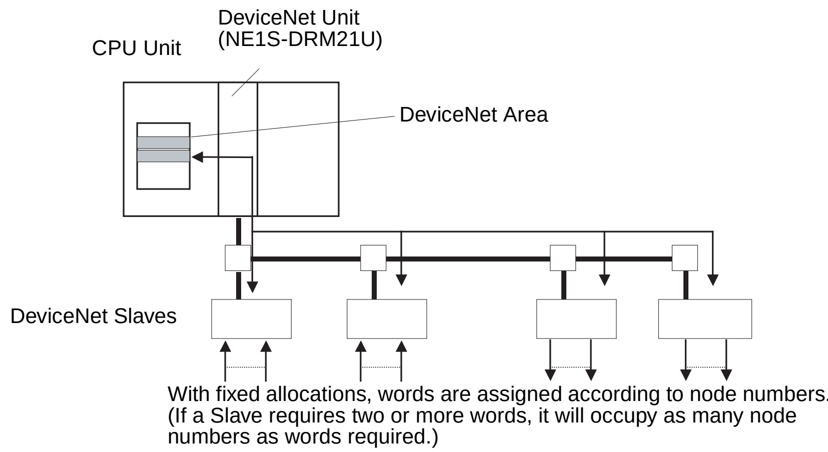
Area	Master to Slave (Output Word)	Slave to Master (Input Word)
Fixed Allocation Area 1	CIO 3370	CIO 3270
Fixed Allocation Area 2	CIO 3570	CIO 3470
Fixed Allocation Area 3	CIO 3770	CIO 3670

Data is exchanged regularly to Slaves in the network (independent of the program) through the DeviceNet Unit (NE1S-DRM21U) mounted in the CPU Rack.

Words can be allocated to Slaves in two ways: fixed allocation (words allocated by node number) or free allocation (user-set word allocation).

- With fixed allocations, words in the DeviceNet Area are allocated automatically in node-number order in one of the fixed allocation areas (1 to 3).
- With user-set allocations, the user can allocate words to Slaves from the following words.
  CIO Area:
  CIO 0000 to CIO 0235, CIO 0300 to CIO 0511, CIO 1000 to CIO 1063
  Work Area: W000 to W511
  Holding Area: H000 to H511
  DM Area: D00000 to D32767
  EM Area: E00000 to E32767 (banks 0 to 3)

For details on word allocations, refer to the *DeviceNet Unit Operation Manual* (Z903).



Forcing Bit Status

Bits in the DeviceNet Area can be force-set and force-reset.

DeviceNet Area Initialization

The contents of the DeviceNet Area will be cleared in the following cases:

1,2,3...

1. The operating mode is changed between PROGRAM and RUN or MONITOR mode and the IOM Hold Bit is OFF.
2. The PLC's power supply is cycled and the IOM Hold Bit is OFF or not protected in the PLC Setup.
3. The DeviceNet Area is cleared from the NE Programmer.
4. PLC operation is stopped when a fatal error other than an FALS(007) error occurs. (The contents of the DeviceNet Area will be retained when FALS(007) is executed.)

IOM Hold Bit Operation

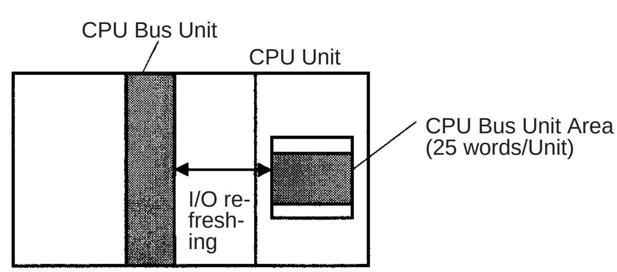
If the IOM Hold BIt (A50012) is ON, the contents of the DeviceNet Area won't be cleared when a fatal error occurs or the operating mode is changed from PROGRAM mode to RUN or MONITOR mode or vice-versa.

If the IOM Hold BIt (A50012) is ON and the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, the contents of the DeviceNet Area won't be cleared when the PLC's power supply is cycled.

5-5 CPU Bus Unit Area

The CPU Bus Unit Area contains 400 words with addresses ranging from CIO 1500 to CIO 1899. Words in the CPU Bus Unit Area can be allocated to CPU Bus Units to transfer data such as the operating status of the Unit. Each Unit is allocated 25 words based on the Unit's unit number setting.

Data is exchanged with CPU Bus Units once each cycle during I/O refreshing, which occurs after program execution. (Words in this data area cannot be refreshed with immediate-refreshing or IORF(097).)



Each CPU Bus Unit is allocated 25 words based on its unit number, as shown in the following table.

Unit number	Allocated words
0	CIO 1500 to CIO 1524
1	CIO 1525 to CIO 1549
2	CIO 1550 to CIO 1574
3	CIO 1575 to CIO 1599
4	CIO 1600 to CIO 1624
5	CIO 1625 to CIO 1649
6	CIO 1650 to CIO 1674
7	CIO 1675 to CIO 1699
8	CIO 1700 to CIO 1724
9	CIO 1725 to CIO 1749
A	CIO 1750 to CIO 1774
B	CIO 1775 to CIO 1799
C	CIO 1800 to CIO 1824
D	CIO 1825 to CIO 1849
E	CIO 1850 to CIO 1874
F	CIO 1875 to CIO 1899

The function of the 25 words depends upon the CPU Bus Unit being used. For details, refer to the Unit's operation manual.

Words in the CPU Bus Unit Area that aren't allocated to CPU Bus Units can be used only in the program.

CPU Bus Unit Area Initialization

The contents of the CPU Bus Unit Area will be cleared in the following cases:

1,2,3...

1. The operating mode is changed from PROGRAM to RUN or MONITOR mode or vice-versa and the IOM Hold Bit is OFF.
2. The PLC's power supply is cycled and the IOM Hold Bit is OFF or not protected in the PLC Setup.
3. The CPU Bus Unit Area is cleared from a Programming Device.
4. PLC operation is stopped when a fatal error other than an FALS(007) error occurs. (The contents of the CPU Bus Unit Area will be retained when FALS(007) is executed.)

IOM Hold Bit Operation

If the IOM Hold BIt (A50012) is ON, the contents of the NE1S-series CPU Bus Unit Area won't be cleared when a fatal error occurs or the operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa.

If the IOM Hold BIt (A50012) is ON and the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, the contents of the NE1S-series CPU Bus Unit Area won't be cleared when the PLC's power supply is cycled.

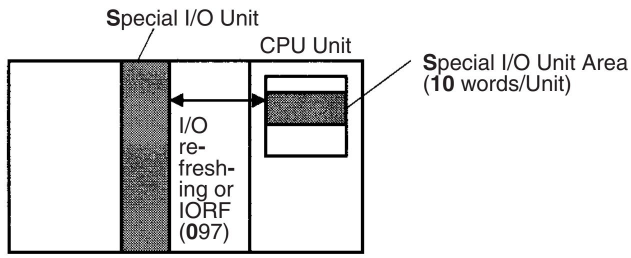
Forcing Bit Status

Bits in the CPU Bus Unit Area can be force-set and force-reset.

5-6 Special I/O Unit Area

The Special I/O Unit Area contains 960 words with addresses ranging from CIO 2000 to CIO 2959. Words in the Special I/O Unit Area are allocated to Special I/O Units to transfer data such as the operating status of the Unit. Each Unit is allocated 10 words based on its unit number setting.

Data is exchanged with Special I/O Units once each cycle during I/O refreshing, which occurs after program execution. The words can also be refreshed with IORF(097).



Each Special I/O Unit is allocated 25 words based on its unit number, as shown in the following table.

Unit number	Allocated words
0	CIO 2000 to CIO 2009
1	CIO 2010 to CIO 2019
2	CIO 2020 to CIO 2029
3	CIO 2030 to CIO 2039
4	CIO 2040 to CIO 2049
5	CIO 2050 to CIO 2059
6	CIO 2060 to CIO 2069
7	CIO 2070 to CIO 2079
8	CIO 2080 to CIO 2089
9	CIO 2090 to CIO 2099
10 (A)	CIO 2100 to CIO 2109
11 (B)	CIO 2110 to CIO 2119
12 (C)	CIO 2120 to CIO 2129
13 (D)	CIO 2130 to CIO 2139
14 (E)	CIO 2140 to CIO 2149
15 (F)	CIO 2150 to CIO 2159
16	CIO 2160 to CIO 2169
17	CIO 2170 to CIO 2179
95	CIO 2950 to CIO 2959

The function of the 10 words allocated to a Unit depends upon the Special I/O Unit being used. For details, refer to the Unit's Operation Manual.

Words in the Special I/O Unit Area that aren't allocated to Special I/O Units can be used only in the program.

Special I/O Unit Area Initialization

The contents of the Special I/O Unit Area will be cleared in the following cases:

1,2,3...

1. The operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa and the IOM Hold Bit is OFF.

2. The PLC's power supply is cycled and the IOM Hold Bit is OFF or not protected in the PLC Setup.
3. The Special I/O Unit Area is cleared from a Programming Device.
4. PLC operation is stopped when a fatal error other than an FALS(007) error occurs. (The contents of the Special I/O Unit Area will be retained when FALS(007) is executed.)

IOM Hold Bit Operation

If the IOM Hold BIt (A50012) is ON, the contents of the Special I/O Unit Area won't be cleared when a fatal error occurs or the operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa.

If the IOM Hold BIt (A50012) is ON and the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, the contents of the Special I/O Unit Area won't be cleared when the PLC's power supply is cycled.

Forcing Bit Status

Bits in the Special I/O Unit Area can be force-set and force-reset.

5-7 Work Area

The Work Area contains 512 words with addresses ranging from W000 to W511. These words can be used only in the program as work words.

There are unused words in the CIO Area (CIO 1200 to CIO 1499 and CIO 3800 to CIO 6143) that can also be used in the program, but use any available words in the Work Area first because the unused words in the CIO Area may be allocated to new functions in future versions of NE1S-series CPU Units.

Work Area Initialization

The contents of the Work Area will be cleared in the following cases:

1,2,3...

1. The operating mode is changed from PROGRAM to RUN or MONITOR mode or vice-versa and the IOM Hold Bit is OFF.
2. The PLC's power supply is cycled and the IOM Hold Bit is OFF or not protected in the PLC Setup.
3. The Work Area is cleared from a Programming Device.
4. PLC operation is stopped when a fatal error other than an FALS(007) error occurs. (The contents of the Work Area will be retained when FALS(007) is executed.)

IOM Hold Bit Operation

If the IOM Hold BIt (A50012) is ON, the contents of the Work Area won't be cleared when a fatal error occurs or the operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa.

If the IOM Hold BIt (A50012) is ON and the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, the contents of the Work Area won't be cleared when the PLC's power supply is cycled.

Forcing Bit Status

Bits in the Work Area can be force-set and force-reset.

5-8 Holding Area

The Holding Area contains 512 words with addresses ranging from H000 to H511 (bits H00000 to H51115). These words can be used only in the program.

Holding Area bits can be used in any order in the program and can be used as normally open or normally closed conditions as often as necessary.

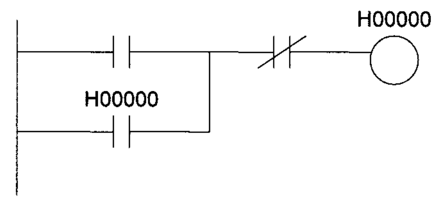
Holding Area Initialization

Data in the Holding Area is not cleared when the PLC's power supply is cycled or the PLC's operating mode is changed from PROGRAM mode to RUN or MONITOR mode or vice-versa.

A Holding Area bit will be cleared if it is programmed between IL(002) and ILC(003) and the execution condition for IL(002) is OFF. To keep a bit ON even when the execution condition for IL(002) is OFF, turn ON the bit with the SET instruction just before IL(002).

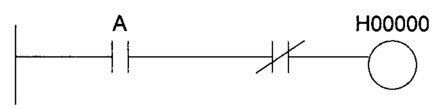
Self-maintaining Bits

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



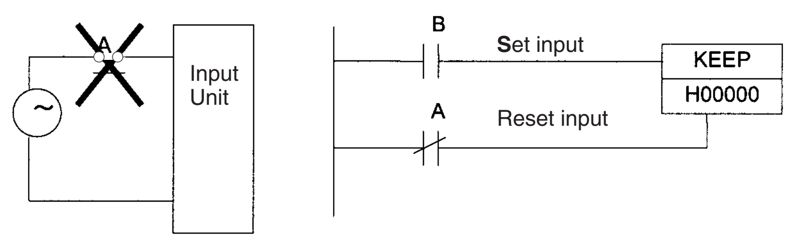
Note

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

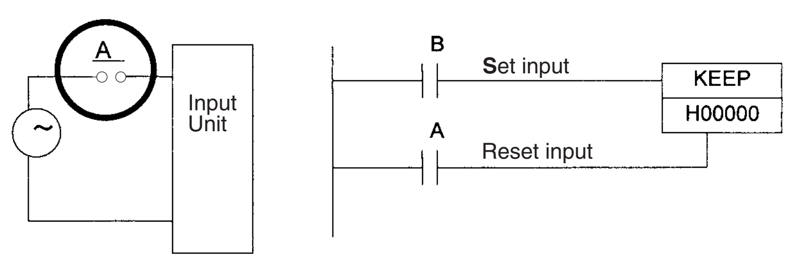


Precautions

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



Instead, use a configuration like the one shown below.



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

5-9 Auxiliary Area

The Auxiliary Area contains 960 words (ranging from A000 to A959) and the words are preassigned as flags and control bits to monitor and control operation. Some bits are automatically set by the system and others are set by the user to control various operations. The data includes error flags set by the self-diagnostic functions, initial settings flags, control bits, and status data.

The Auxiliary Area can be accessed from the program or Programming Devices.

Forcing Bit Status

Bits in the Auxiliary Area cannot be force-set and force-reset continuously.

Functions

For details on the functions of individual words and bits, refer to the following tables in *5-9-1 Read-only Area (Set by System)* and *5-9-2 Read/Write Area (Set by User)*. The tables are organized according to the addresses. A000 through A447 are read-only, but A448 through A959 can be read or written from the program or a Programming Device.

5-9-1 Read-only Area (Set by System)

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A000	00 to 15	10-ms Incrementing Free Running Timer	This word contains a system timer used after the power is turned ON. The timer is reset to 0000 hex 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. This timer can be used to time a process (in 10-ms units) without using a timer instruction. For example, to time process A, read A000 at the start of process A, read A000 again at the start of process B, and subtract to determine the difference. **Note** The timer will continue to be incremented when the operating mode is switched to RUN mode.	---	Retained	Cleared	Every 10 ms after power is turned ON	---
A001	00 to 15	100-ms Incrementing Free Running Timer	This word contains a system timer used after the power is turned ON. The timer is reset to 0000 hex 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. This timer can be used to time a process (in 100-ms units) without using a timer instruction. For example, to time process A, read A000 at the start of process A, read A000 again at the start of process B, and subtract to determine the difference. **Note** The timer will continue to be incremented when the operating mode is switched to RUN mode.	---	Retained	Cleared	Every 100 ms after power is turned ON	---
A050	A05000 to A05007	Basic I/O Unit Information, Rack 0 Slot 0	A bit will turn ON to indicate when a fuse has blows. The bit numbers correspond to the fuse number on the Unit.	1: Fuse blown 0: Normal	---	---	Every cycle	
	A05008 to A05015	Basic I/O Unit Information, Rack 0 Slot 1			---	---		---
A051 to A089	A05100 to A08915	Basic I/O Unit Information, Racks 2 to 7			---	---		---

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A090 to A093	A09000 to A09315	User Program Date	These words contain in BCD the date and time that the user program was last overwritten. A09000 to A09007: Seconds (00 to 59) A09008 to A09015: Minutes (00 to 59) A09100 to A09107: Hour (00 to 23) A09108 to A09115: Day of month (01 to 31) A09200 to A09207: Month (01 to 12) A09208 to A09215: Year (00 to 99) A09300 to A09307: Day of the week (00: Sunday, 01: Monday, 02: Tuesday, 03: Wednesday, 04: Thursday, 05: Friday, 06: Saturday)	---	Retained	Retained	---	---
A094 to A098	A09400 to A09815	Parameter Date	These words contain in BCD the date and time that the parameters were last overwritten. The format is the same as above This date applies only to changes in the PLC Setup, I/O table, routing table, or CPU Bus Unit Setup.	---	Retained	Retained	---	---
A099	A09900	UM Read Protection Status	Indicates whether the entire user program in the PLC is read-protected.	0: UM not read-protected. 1: UM read-protected.	Retained	Retained	When protection is set or cleared	---
	A09901	Task Read Protection Status	Indicates whether read protection is set for individual tasks.	0: Tasks not read-protected. 1: Tasks read-protected.	Retained	Retained	When protection is set or cleared	---
	A09902	Program Write Protection Status when Read Protection Is Set	Indicates whether the program is write-protected.	0: Write-enabled. 1: Write-protected.	Retained	Retained	When protection is set or cleared	---
	A09903	Enable/Disable Status for Backing Up the Program to a Memory Card	Indicates whether creating a backup program file (.OBJ) is enabled or disabled.	0: Enabled. 1: Disabled.	Retained	Retained	When protection is set or cleared	---
	A09914	IR/DR Operation between Tasks	Turn ON this bit to share index and data registers between all tasks. Turn OFF this bit to use separate index and data registers between in each task.	0: Independent 1: Shared (default)	Retained	Retained	---	---
	A09915	Timer/Counter PV Refresh Mode Flag	Indicates whether the CPU Unit is storing timer/counter PV in BCD or binary.	0: BCD mode 1: Binary mode	Retained	Retained	---	---

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A100 to A199	A10000 to A19915	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: 1) Error code (bits 0 to 15) 2) Error contents (bits 0 to 15) 3) Minutes (bits 8 to 15), Seconds (bits 0 to 7) 4) Day of month (bits 8 to 15), Hours (bits 0 to 7) 5) Year (bits 8 to 15), Month (bits 0 to 7) Errors generated by FAL(006) and FALS(007) will also be stored in this Error Log. The Error Log Area can be reset from a Programming Device. If the Error Log Area is full (20 records) and another error occurs, the oldest record in A100 to A104 will be cleared, the other 19 records are shifted down, and the new record is stored in A195 to A199.	Error code Error contents: Address of Aux. Area word with details or 0000. Seconds: 00 to 59, BCD Minutes: 00 to 59, BCD Hours: 00 to 23, BCD Day of month: 01 to 31, BCD Year: 00 to 99, BCD	Retained	Retained	Written when error occurs	A50014 A300 A400
A200	A20011	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	---	---	---	---
	A20012	Step Flag	ON for one cycle when step execution is started with STEP(008). This flag can be used for initialization processing at the beginning of a step.	ON for the first cycle after execution of STEP(008).	Cleared	---	---	---
	A20014	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. The only difference between this flag and A20015 is that this flag also turns ON when the task switches from WAIT to RUN status.	ON for first cycle (including transitions from WAIT and IN)	Cleared	Cleared	---	---
	A20015	First Task Startup Flag	ON when a task is executed for the first time. This flag can be used to check whether the current task is being executed for the first time so that initialization processing can be performed if necessary.	1: First execution 0: Not executable for the first time or not being executed.	Cleared	---	---	---
A201	A20110	Online Editing Wait Flag	ON when an online editing process is waiting. (If another online editing command is received while waiting, the other command won't be recorded and an error will occur.)	1: Waiting for online editing 0: Not waiting for online editing	Cleared	Cleared	---	A527
	A20111	Online Editing Flag	ON when an online editing process is being executed.	1: Online editing in progress 0: Online editing not in progress	Cleared	Cleared	---	A527

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A202	A20200 to A20207	Communi-cations Port Enabled Flags	ON when network instruction CSND or background execution can be executed with the corresponding port number. Bits 00 to 07 correspond to communications ports 0 to 7. **Note** When the simple backup operation is used to perform a write or compare operation for a Memory Card, a communications port will be automatically allocated, and the corresponding flag will be turned ON during execution, and will be turned OFF at completion.	1: Communications enabled 0: Communications disabled.	Cleared	---	---	---
	A20213 and A20214	Network Instruction Executable Flags	ON when network instructions can be executed.	1: Network instructions can be executed. 0: Network instructions cannot be executed.	Cleared	---	---	A21913 and A21914
A203 to A210	A20300 to A21015	Communi-cations Port Completion Codes	Store the response codes when network instruction CSND has been executed. The code is cleared to 0000 hex when background processing has been completed. Words A203 to A210 correspond to communications port numbers 0 to 7. **Note** When the simple backup operation is used to perform a write or compare operation for a Memory Card, a communications port will be automatically allocated, and the completion code will be stored in the corresponding word.	Non-zero: Error code 0000 hex: Normal condition	Cleared	---	---	---
A213CH	A21300 to A21307	Not used.						
A219	A21900 to A21907	Communi-cations Port Error Flags	ON when an error occurred during execution of the network instruction CSND. The flags turn OFF when completion is normal. Bits 00 to 07 correspond to communications ports 0 to 7. **Note** When the simple backup operation is used to perform a write or compare operation for a Memory Card, a communications port will be automatically allocated, and the corresponding flag will be turned ON if an error occurs, or remain OFF for normal completion.	1: Error occurred 0: Normal condition	Cleared	---	---	---
	A21913 and A21914	Network Instruction Error Flags	Turn ON when an error has occurred for the network instruction.	1: Error 0: Normal	Cleared	---	---	A20213 and A20214
A220 to A259	A22000 to A25915	Basic I/O Unit Input Response Times	These words contain the actual input response times for CS-series Basic I/O Units. When the Basic I/O Unit input response time setting is changed in the PLC Setup while the PLC is in PROGRAM mode, the setting in the PLC Setup will not match the actual value in the Basic I/O Unit unless the power is turned OFF and then ON again. In that case, the actual value can be monitored in these words.	0 to 17 hexadecimal	Retained	See function column.	---	PLC Setup (Basic I/O Unit Input response time settings)

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A261	A26100	I/O Table Errors (CS1-H CPU Units only)	CPU Bus Unit Setup Area Initialization Error Flag ON when an I/O table creation error is caused by a CPU Bus Unit Setup Error.	1: Error 1→0: I/O table generated normally.	Retained	Cleared	Written when the I/O table is created	---
	A26102		I/O Overflow Flag ON when an I/O table creation error is caused by a Too Many I/O Points Error.	1: Error 1→0: I/O table generated normally.				A40111 (Too Many I/O Points Error Flag)
	A26103		Duplication Error Flag ON when an I/O table creation error is caused by a unit number duplication error.	1: Error 1→0: I/O table generated normally				A40113 (Duplication Error Flag)
	A26104		I/O Bus Error Flag Turned ON when an I/O table creation error is caused by an I/O Bus Error.	1: Error 1→0: I/O table generated normally				A40114 I/O Bus Error Flag
	A26106		SYSMAC BUS Recognition Error Flag Turned ON when an I/O table creation error is caused by a SYSMAC BUS detection error.	1: Error 1→0: I/O table generated normally.				---
	A26107		Special I/O Unit Error Flag Turned ON when an I/O table creation error is caused by a Special I/O Unit Error.	1: Error 1→0: I/O table generated normally				---
	A26109		I/O Unconfirmed Error Flag Turned ON when an I/O table creation error occurs because I/O detection has not been completed.	1: Error 1→0: I/O table generated normally				---
A262 and A263	A26200 to A26315	Maximum Cycle Time	These words contain the maximum cycle time since the start of PLC operation (the program execution cycle time for a Parallel Processing Mode). The cycle time is recorded in 8-digit hexadecimal with the leftmost 4 digits in A263 and the rightmost 4 digits in A262.	0 to FFFFFFFF: 0 to 429,496,729.5 ms (0.1ms units)	---	---	---	---
A264 and A265	A26400 and A26515	Present Cycle Time	These words contain the present cycle time in 8-digit hexadecimal with the leftmost 4 digits in A265 and the rightmost 4 digits in A264 (the program execution cycle time for a Parallel Processing Mode).	0 to FFFFFFFF: 0 to 429,496,729.5 ms	---	---	---	---
A266 to A293	A26600 to A29315	Not used.						
A294	A29400 to A29415	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. (A298 and A299 contain the program address where program execution was stopped.)	Normal tasks: 0000 to 007F (task 0 to 127) Interrupt tasks: 8000 to 80FF (task 0 to 255)	Cleared	Cleared	---	A298/ A299

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A295	A29507	Variable Information Inconsistent Error	Turns ON when a project is downloaded to the CPU Unit and the variable information in flash memory is inconsistent with the program.	1: Inconsistent 0: Consistent				
	A29508	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. (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.)	1: Error Flag ON 0: Error Flag OFF	Cleared	Cleared	---	A294, A298/ A299 PLC Setup (Operation when instruction error has occurred)
	A29509	Indirect DM/EM BCD Error Flag	This flag and the Access Error Flag (AER) will be turned ON when an indirect DM/EM BCD error has occurred and the PLC Setup has been set to stop operation an indirect DM/EM BCD error. (This error occurs when the content of an indirectly addressed DM or EM 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. (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.)	1: Not BCD 0: Normal	Cleared	Cleared	---	A294, A298/ A299 PLC Setup (Operation when instruction error has occurred)

Address: Words	Address: Bits	Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
A295	A29510	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 access 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: 1) Reading/writing the system area 2) Reading/writing EM File Memory 3) Writing to a write-protected area 4) Indirect DM/EM BCD error (in BCD mode) (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.)	1: Illegal access occurred 0: Normal condition	Cleared	Cleared	---	A294, A298/ A299 PLC Setup (Operation when instruction error has occurred)
	A29511	No END Error Flag	ON when there isn't an END(001) instruction in each program within a task. CPU Unit operation will stop and the ERR/ALM indicator will light when this flag goes ON. (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.)	1: No END 0: Normal condition	Cleared	Cleared	---	A294, A298/ A299
	A29512	Task Error Flag	ON when a task error has occurred. The following conditions generate a task error. There isn't even one regular task that is executable (started). •There isn't a program allocated to the task. •(The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.)	1: Error 0: Normal	Cleared	Cleared	---	A294, A298/ A299
	A29513	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. (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.)	1: Error 0: Normal	Cleared	Cleared	---	A294, A298/ A299
	A29514	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. (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.)	1: Error 0: Normal	Cleared	Cleared	---	A294, A298/ A299
	A29515	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.	1: Error 0: Normal	Cleared	Cleared	---	A294, A298/ A299
A296 and A297	A29600 to A29715	Not used.						

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A298	A29800 to A29815	Program Address Where Program Stopped (Rightmost 4 digits)	These words contain the 8-digit binary program address of the instruction where program execution was stopped due to a program error.	Right 4 digits of the program address	Cleared	Cleared	---	A294
A299	A29900 to A29915	Program Address Where Program Stopped (Leftmost 4 digits)	(A294 contains the task number of the task where program execution was stopped.)	Left 4 digits of the program address	Cleared	Cleared	---	
A300	A30000 to A30015	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). The Error Log Pointer can be cleared to 00 by turning A50014 (the Error Log Reset Bit) from OFF to ON. When the Error Log Pointer has reached 14 (20 decimal), the next record is stored in A195 to A199 when the next error occurs.	00 to 14 hexadecimal	Retained	Retained	Written when error occurs	A50014
A302	A30200 to A30215	CPU Bus Unit Initializing Flags	These flags are ON while the corresponding CPU Bus Unit is initializing after its CPU Bus Unit Restart Bit (A50100 to A50115) is turned from OFF to ON or the power is turned ON. Bits 00 to 15 correspond to unit numbers 0 to 15. Use these flags in the program to prevent the CPU Bus Unit's refresh data from being used while the Unit is initializing. IORF(097) cannot be executed while an CPU Bus Unit is initializing. These bits are turned OFF automatically when initialization is completed.	0: Not initializing 1: Initializing (Reset to 0 automatically after initialization.)	Retained	Cleared	Written during initialization	A50100 to A50115
A330 to A335	A33000 to A33515	Special I/O Unit Initializing Flags	These flags are ON while the corresponding Special I/O Unit is initializing after its Special I/O Unit Restart Bit (A50200 to A50715) is turned from OFF to ON or the power is turned ON. The bits in these words correspond to unit numbers 0 to 95 as follows: A33000 to A33015: Units 0 to 15 A33100 to A33115: Units 16 to 31 ---- A33500 to A33515: Units 80 to 95 Use these flags in the program to prevent the Special I/O Unit's refresh data from being used while the Unit is initializing. Also, IORF(097) cannot be executed while a Special I/O Unit is initializing. These bits are turned OFF automatically when initialization is completed.	0: Not initializing 1: Initializing (Reset to 0 automatically after initialization.)	Retained	Cleared	---	A50200 to A50715
A336 and A342	A33600 to A34215	Not used.						

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A343	A34300 to A34302	Memory Card Type	Indicates the type of Memory Card, if any, installed. This information is recorded when the PLC power is turned ON or the Memory Card power switch is turned ON.	0: None 4: Flash ROM	Retained	See Function column.	See Function column.	---
	A34307	Memory Card Format Error Flag	ON when the Memory Card is not formatted or a formatting error has occurred. (The flag is turned OFF when formatting is completed normally.) This flag is written when the PLC power is turned ON or the Memory Card power switch is turned ON.	1: Format error 0: No format error	Retained	See Function column.	See Function column.	
	A34308	File Transfer Error Flag	ON when an error occurred while writing data to file memory. (The flag is turned OFF when PLC operation begins or data is written successfully.)	1: Error 0: No error	Retained	Cleared	Written when file data is written	---
	A34309	File Write Error Flag	ON when data cannot be written to file memory because it is write-protected or the data exceeds the capacity of the file memory. (The flag is turned OFF when PLC operation begins or data is written successfully.)	1: Write not possible 0: Normal condition	Retained	Cleared	Written when file data is written	---
	A34310	File Read Error	ON when a file could not be read because of a malfunction (file is damaged or data is corrupted). (The flag is turned OFF when PLC operation begins or data is read successfully.)	1: Read not possible 0: Normal condition	Retained	Cleared	Written when file data is read	---
	A34311	File Missing Flag	ON when an attempt is made to read a file that doesn't exist, or an attempt is made to write to a file in a directory that doesn't exist. (The flag is turned OFF when PLC operation begins or data is read successfully.)	1: Specified file or directory is missing 0: Normal condition	Retained	Cleared	Written when file data is read	---
	A34313	File Memory Operation Flag	ON while any of the following operations is being executed. OFF when none of them are being executed. CMND instruction sending a FINS command to the local CPU Unit. FREAD/FWRIT instructions. Program replacement using the control bit in the Auxiliary Area. Simple backup operation. (The flag is turned OFF when PLC operation begins.)	1: Instruction being executed. 0: Instruction not being executed.	Retained	Cleared	Written when file memory instruction is executed	---
	A34314	Accessing File Data Flag	ON while file data is being accessed. Use this flag to prevent two file memory instructions from being executed at the same time. (The flag is turned OFF when PLC operation begins.)	1: File being accessed 0: File not being accessed	Retained	Cleared	---	---
	A34315	Memory Card Detected Flag	ON when a Memory Card has been detected. OFF when a Memory Card has not been detected.	1: Memory Card detected 0: Memory Card not detected	Retained	Cleared	Written when Memory Card is inserted, or the power is turned ON.	
A344 and A345	A34400 to A34515	Not used.						

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	**Bits**							
A346 and A347	A34600 to A34715	Number of Remaining Words to Transfer	These words contain the 8-digit hexadecimal number of words remaining to be transferred by FREAD(700) or FWRIT(701). When one of these instructions is executed, the number of words to be transferred is written to A346 and A347. While the data is being transferred, the value in these words is decremented. A326 contains the rightmost 4-digits and A347 contains the leftmost 4-digits. Check the content of these words to determine whether or not the planned number of words have been transferred successfully.	Data remaining in transfer	Retained	Cleared	Written as FREAD or FWRIT is being executed. Decremented as data is actually transferred.	---
A348 to A350	A34800 to A35015	Not used.						
A351 to A354	---	Calendar/Clock Area	These words contain the CPU Unit's internal clock data in BCD. The clock can be set from a Programming Device such as a Programming Console, with the DATE(735) instruction, or with a FINS command (CLOCK WRITE, 0702).		Retained	Retained	Written every cycle	---
	A35100 to A35107		Seconds (00 to 59) (BCD)					
	A35108 to A35115		Minutes (00 to 59) (BCD)					
	A35200 to A35207		Hours (00 to 23) (BCD)					
	A35208 to A35215		Day of the month (01 to 31) (BCD)					
	A35300 to A35307		Month (01 to 12) (BCD)					
	A35308 to A35315		Year (00 to 99) (BCD)					
	A35400 to A35407		Day of the week (00 to 06) (BCD) 00: Sunday, 01: Monday, 02: Tuesday, 03: Wednesday, 04: Thursday, 05: Friday, 06: Saturday					
A355 to A359	A35500 to A35915	Not used.						
A360 to A391	A36001 to A39115	Executed FAL Number Flags	The flag corresponding to the specified FAL number will be turned ON when FAL(006) is executed. Bits A36001 to A39115 correspond to FAL numbers 001 to 511. The flag will be turned OFF when the error is cleared.	1: That FAL was executed 0: That FAL wasn't executed	Retained	Cleared	Written when error occurs	A40215

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A392	A39204	RS-232C Port Error Flag	ON when an error has occurred at the RS-232C port. (Do not read this bit when operating in peripheral bus mode or NT Link mode.)	1: Error 0: No error	Retained	Cleared	Written when error occurs	A528
	A39205	RS-232C Port Send Ready Flag (No-protocol mode)	ON when the RS-232C port is able to send data in no-protocol mode.	1: Able-to-send 0: Unable-to-send	Retained	Cleared	Written after transmission	---
	A39206	RS-232C Port Reception Completed Flag (No-protocol mode)	ON when the 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.	1: Reception completed 0: Reception not completed	Retained	Cleared	Written after reception	---
	A39207	RS-232C Port Reception Overflow Flag (No-protocol mode)	ON when a data overflow occurred during reception through the 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(235) was executed. • When the end code was specified: ON when more data is received after the end code was received but before RXD(235) was executed. ON when 257 bytes are received before the end code.	1: Overflow 0: No overflow	Retained	Cleared	---	---
A393	A39300 to A39307	RS-232C Port PT Communications Flag	The corresponding bit will be ON when the RS-232C port is communicating with a PT in NT link mode. Bits 0 to 7 correspond to units 0 to 7.	1: Communicating 0: Not communicating	Retained	Cleared	Written when there is a normal response to the token	---
	A39308 to A39315	RS-232C Port PT Priority Registered Flags	The corresponding bit will be ON for the PT that has priority when the RS-232C port is communicating in NT link mode. Bits 0 to 7 correspond to units 0 to 7. These flags are written when the priority registration command is received.	1: Priority registered 0: Priority not registered	Retained	Cleared	See *Function* column.	---
	A39300 to A39315	RS-232C Port Reception Counter (No-protocol mode)	Indicates (in binary) the number of bytes of data received when the RS-232C port is in no-protocol mode.		Retained	Cleared	Written when data is received	---
A394	---	Not used.						
A395	A39506	File Deleted Flags	The system deleted the remainder of a Memory Card file that was being updated when a power interruption occurred.	1: File deleted 0: No files deleted	Cleared	Cleared	Written when the system deletes the file.	---
	A39510	ER/AER Flag for Background Execution	Turns ON if an error or illegal access occurs during background execution. Turns OFF when power is turned ON or operation is started.	1: Error 0: No error (Cleared when background execution is started.)	Cleared	Cleared	---	---

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A395	A39511	Memory Corruption Detected Flag	ON when memory corruption is detected when the power supply is turned ON.	1: Memory corruption 0: Normal operation	Retained	See Function column.	Written when power is turned ON.	---
	A39512	DIP Switch Pin 6 Status Flag	The status of pin 6 on the DIP switch on the front of the CPU Unit is written to this flag every cycle.	1: Pin 6 ON 0: Pin 6 OFF	Retained	See Function column.	Written every cycle	---
A396 and A399	A39600 to A39915	Not used.						
A400	A40000 to A40015	Error code	When a non-fatal error (user-defined FALS(006) or system error) or a fatal error (user-defined FALS(007) or system error) occurs, the 4-digit hexadecimal error code is written to this word. When two or more errors occur simultaneously, the highest error code will be recorded.	Error code (Refer to *Appendix F Details on Auxiliary Area Operation* for details.)	Cleared	Cleared	Written when error occurs	---
A401	A40106	FALS Error Flag (Fatal error)	ON when a non-fatal error is generated by the FALS(006) instruction. The CPU Unit will continue operating and the ERR/ALM indicator will flash. The corresponding error code will be written to A400. Error codes C101 to C2FF correspond to FALS numbers 001 to 511. This flag will be turned OFF when the FALS errors are cleared.	1: FALS(006) executed 0: FALS(006) not executed	Cleared	Cleared	Written when error occurs	A400
	A40108	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). The program execution cycle time is used in a Parallel Processing Mode. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. This flag will be turned OFF when the error is cleared.	0: Cycle time under max. 1: Cycle time over max.	Cleared	Cleared	Written when the cycle time exceeds max.	PLC Setup (Cycle time monitoring time)

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A401	A40109	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 bits 8 to 15 of A295. Refer to A295 for details on program errors. The task type and number where program execution stopped due to a program error will be stored in A294. This flag will be turned OFF when the error is cleared.	1: Error 0: No error	Cleared	Cleared	---	A294, A295, A298 and A299
	A40110	I/O Setting Error Flag (Fatal error)	ON when an Input Unit has been installed in an Output Unit's slot or vice versa, so the Input and Output Units clash in the registered I/O table. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. This flag will be turned OFF when the error is cleared.	1: Error 0: No error	Cleared	Cleared	---	---
	A40111	Too Many I/O Points Flag (Fatal error)	ON when the number of I/O points being used in Basic I/O Units exceeds the maximum allowed for the PLC. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. This flag will be turned OFF when the error is cleared.	1: Error 0: No error	Cleared	Cleared	---	A407
	A40113	Duplication Error Flag (Fatal error)	ON in the following cases: • Two CPU Bus Units have been assigned the same unit number. • Two Special I/O Units have been assigned the same unit number. • Two Basic I/O Units have been allocated the same data area words. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. The duplicated unit number is indicated in A409 to A416. (This flag will be turned OFF when the error is cleared.)	1: Duplication error 0: No duplication	Cleared	Cleared	---	A410 to A416

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A401	A40114	I/O Bus Error Flag (Fatal error)	ON when an error occurs in a data transfer between the CPU Unit and a Unit mounted to a slot. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. The slot number (00 to 09) where the I/O Bus Error occurred is written to A40400 to A40407 in binary and the rack number (00 to 07) is written to A40408 to A40415 in binary (This flag will be turned OFF when the error is cleared.)	1: Error 0: No error	Cleared	Cleared	---	A404
	A40115	Memory Error Flag (Fatal error)	ON when an error occurred in memory or there was an error in automatic transfer from the Memory Card when the power was turned ON. 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 A40300 to A40308, and A40309 will be turned ON if there was an error during automatic transfer at start-up. This flag will be turned OFF when the error is cleared. (The automatic transfer at start-up error cannot be cleared without turning off the PLC.)	1: Error 0: No error	Cleared	Cleared	---	A40300 to A40308, A40309
A402	A40201	Ethernet Communications Error Flag	Turns ON when an error occurs in communications for the built-in Ethernet port, such as an IP address setting error.	1: Error 0: Normal	Retained	Cleared	---	A40312, A438
	A40202	Special I/O Unit Setting Error Flag (Non-fatal error)	ON when an installed Special I/O Unit does not match the Special I/O Unit registered in the I/O table. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. The unit number of the Unit where the setting error occurred is indicated in A428 to A433. (This flag will be turned OFF when the error is cleared.)	1: Setting error detected 0: No setting error	Cleared	Cleared	---	A428 to A433
	A40203	CPU Bus Unit Setting Error Flag (Non-fatal error)	ON when an installed CPU Bus Unit does not match the CPU Bus Unit registered in the I/O table. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. The unit number of the Unit where the setting error occurred is written to A427. (This flag will be turned OFF when the error is cleared.)	1: Setting error detected 0: No setting error	Cleared	Cleared	---	A427
	A40204	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.)	1: Error 0: No error	Cleared	Cleared	---	PLC Setup (Detect Battery Error)

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A402	A40206	Special I/O Unit Error Flag (Non-fatal error)	ON when an error occurs in a data exchange between the CPU Unit and a Special I/O Unit (including an error in the Special I/O Unit itself). The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. The Special I/O Unit where the error occurred will stop operating and the unit number of the Unit where the data exchange error occurred is indicated in A418 through A423. (This flag will be turned OFF when the error is cleared.)	1: Error in one or more Units 0: No errors in any Unit	Cleared	Cleared	---	A418 to A423
	A40207	CPU Bus Unit Error Flag (Non-fatal error)	ON when an error occurs in a data exchange between the CPU Unit and an CPU Bus Unit (including an error in the CPU Bus Unit itself). The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. The CPU Bus Unit where the error occurred will stop operating and the unit number of the Unit where the data exchange error occurred is indicated in A417. (This flag will be turned OFF when the error is cleared.)	1: Error in one or more Units 0: No error in any Unit	Cleared	Cleared	---	A417
	A40209	I/O Verification Error Flag (Non-fatal error)	ON when a Basic I/O Unit registered in the I/O Table does not match the Basic I/O Unit actually installed in the PLC because a Unit was added or removed. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. (This flag will be turned OFF when the error is cleared.)	1: Mismatch 0: No mismatch	Cleared	Cleared	---	---
	A40210	PLC Setup Error Flag (Non-fatal error)	ON when there is a setting error in the PLC Setup. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. The location of the error will be written to A406. (This flag will be turned OFF when the error is cleared.)	1: Error 0: No error	Cleared	Cleared	---	A406

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A402	A40212	Basic I/O Unit Error Flag (Non-fatal error)	ON when an error has occurred in a Basic I/O Unit. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. The location of the error will be written to A408. (This flag will be turned OFF when the error is cleared.)	1: Error 0: No error	Cleared	Cleared	---	A408
	A40213	Interrupt Task Error Flag (Non-fatal error)	ON when the Detect Interrupt Task Errors setting in the PLC Setup is set to “Detect” and an attempt is made to refresh a Special I/O Unit’s I/O from an interrupt task with IORF(097) while the Unit’s I/O is being refreshed by cyclic I/O refreshing (duplicate refreshing). The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. (This flag will be turned OFF when the error is cleared.)	1: Interrupt task error 0: No error	Cleared	Cleared	---	A426, PLC Setup (Detect Interrupt Task Errors setting)
	A40215	FAL Error Flag (Non-fatal error)	ON when a non-fatal error is generated by executing FAL(006). 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(006) will be turned ON and the corresponding error code will be written to A400. Error codes 4101 to 42FF correspond to FAL numbers 001 to 2FF (0 to 511). (This flag will be turned OFF when the error is cleared.)	1: FALS(006) error occurred 0: FALS(006) not executed	Cleared	Cleared	Written when error occurs	A360 to A391, A400

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	Bits							
A403	A40300 to A40308	Memory Error Location	When a memory error occurs, the Memory Error Flag (A40115) is turned ON and one of the following flags is turned ON to indicate the memory area where the error occurred A40300: User program A40304: PLC Setup A40305: Registered I/O Table A40307: Routing Table A40308: CPU Bus Unit Settings 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. (The corresponding flag will be turned OFF when the error is cleared.)	1: Error 0: No error	Cleared	Cleared	---	A40115
	A40309	Memory Card Startup Transfer Error Flag	ON when automatic transfer at startup has been selected and an error occurs during automatic transfer. An error will occur if there is a transfer error, the specified file does not exist, or the Memory Card is not installed. (This flag will be turned OFF when the error is cleared by turning the power off. The error cannot be cleared without turning the power off.)	1: Error 0: No error	Cleared	Cleared	Written when power is turned ON	---
	A40310	Flash Memory Error Flag	ON when the flash memory has been physically damaged.	1: Error 0: No error	Cleared	Cleared	Written an error occurs	---
	A40311	Variable Information Error Flag	ON if an error occurs when variable information that was automatically backed up in flash memory is read.	1: Error 0: No error	Cleared	Cleared	---	
	A40312	Built-in Ethernet Setting Error Flag	ON if an error occurs when built-in Ethernet settings that were automatically backed up in flash memory are read.	1: Error 0: No error	Cleared	Cleared	---	A40201, A438
A404	A40400 to A40407	I/O Bus Error Slot Number	Contains the 8-bit binary slot number (00 to 09) where an I/O Bus Error occurred. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. The I/O Bus Error Flag (A40114) will be ON. (This flag will be turned OFF when the error is cleared.)	00 to 09 hex (slot No. 0 to 9)	Cleared	Cleared	---	A40114
	A40408 to A40415	I/O Bus Error Rack Number	Contains the 8-bit binary rack number (00 to 07) where an I/O Bus Error occurred. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. The I/O Bus Error Flag (A40114) will be ON. (This flag will be turned OFF when the error is cleared.)	00 to 03 hex (Rack No. 0 to 3)	Cleared	Cleared	---	A40114
A405	A40500 to A40515	Not used.						

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	**Bits**							
A406	A40600 to A40615	PLC Setup Error Location	When there is a setting error in the PLC Setup, the location of that error is written to A406 in 4-digit hexadecimal. The location is given as the address displayed on a Programming Console. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. (A406 will be cleared when the cause of the error is eliminated.)	0000 to 01FF hexadecimal	Cleared	Cleared	Written when error occurs	A40210
A407	A40700 to A40712	Too Many I/O Points, Details	The 6 possible causes of the Too Many I/O Points Error are listed below. The 3-digit binary value in A40713 to A40715 indicates the cause of the error (values 0 to 5 correspond to causes 1 to 6, below). The 13-bit binary value in A40700 to A40712 indicates the details: the excessive value or the duplicated unit number. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. 1) The number of I/O points will be written here when the total number of I/O points set in the I/O Table (excluding Slave Racks) exceed the maximum allowed for the CPU Unit. 2) The number of interrupt inputs will be written here when there are more than 32 interrupt inputs. 3) The unit number of the Slave Unit will be written here when a unit number is duplicated or the number of I/O points on a C500 Slave Unit exceeds 320. 4) The unit number of the I/O Terminal (excluding Slave Racks) will be written here when a unit number is duplicated 5) The unit number of the Master Unit will be written here when a unit number is duplicated or the unit number is outside of the allowed setting range. 6) The number of Racks will be written here when the number of Expansion I/O Racks exceeds the maximum. (The relevant value will be written here (A40700 to A40712) when the error occurs. These bits will be cleared when the error is cleared.)	0000 to 1FFF hexadecimal	Cleared	Cleared	Written when error occurs	A40111, A40713 to A40715

Address: Words	Address: Bits	Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
A407	A40713 to A40715	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 and shows the meaning of the value written to bits A40700 to A40712. Values of 000 to 101 (0 to 5) correspond to causes 1 through 6 described in "Too Many I/O Points, Cause 1," above. (These bits will be cleared when the error is cleared.)	000: Too many I/O total 001: Too many interrupt I/O 010: Duplicate Remote I/O Slave Unit unit numbers or too many I/O on C500 Remote I/O Slave Units (more than 320) 011: Duplicate I/O Terminal unit numbers 100: Duplicate Remote I/O Master Unit unit numbers or undefined unit numbers (not 0 or 1) 101: Too many Racks	Cleared	Cleared	Written when error occurs	---
A408	A40800 to A40807	Basic I/O Unit Error, Slot Number	When an error has occurred in a Basic I/O Unit, A40212 will be turned ON and the slot number where the error occurred will be written here in binary. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. (These bits will be cleared when the error is cleared.)	00 to 09 hexadecimal (Slots 0 to 9)	Cleared	Cleared	---	A40212
	A40808 to A40815	Basic I/O Unit Error, Rack Number	When an error has occurred in a Basic I/O Unit, A40212 will be turned ON and the Rack number where the error occurred will be written here in binary. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. (These bits will be cleared when the error is cleared.)	00 to 07 hexadecimal (Racks 0 to 7)	Cleared	Cleared	---	A40212
A409	A40900 to A40907	Expansion I/O Rack Number Duplication Flags	The corresponding flag will be turned ON when an Expansion I/O Rack's starting word address was set from a Programming Device and two Racks have overlapping word allocations or a Rack's starting address exceeds CIO 0901. Bits 00 to 07 correspond to Racks 0 to 7. (The corresponding flag will be cleared when the error is cleared.)	1: Error 0: No error	Cleared	Cleared	---	---
A410	A41000 to A41015	CPU Bus Unit Number Duplication Flags	The Duplication Error Flag (A40113) and the corresponding flag in A410 will be turned ON when an CPU Bus Unit's unit number has been duplicated. Bits 00 to 15 correspond to unit numbers 0 to F. CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light.	1: Duplication detected 0: No duplication	Cleared	Cleared	---	A40113

Address: Words	Address: Bits	Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
A411 to A416	A41100 to A41615	Special I/O Unit Number Duplication Flags	The Duplication Error Flag (A40113) and the corresponding flag in A411 through A416 will be turned ON when a Special I/O Unit's unit number has been duplicated. Bits 00 to 15 correspond to unit numbers 0 to F. (Bits A41100 to A41615 correspond to unit numbers 000 to 05F (0 to 95).) CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. The corresponding bit will also be turned ON when the Special I/O Unit's words are also allocated to a Basic I/O Unit on an Expansion I/O Rack because of the Expansion I/O Rack's starting word setting.	1: Duplication detected 0: No duplication	Cleared	Cleared	---	A40113
A417	A41700 to A41715	CPU Bus Unit Error, Unit Number Flags	When an error occurs in a data exchange between the CPU Unit and an CPU Bus Unit, the CPU Bus Unit Error Flag (A40207) is turned ON and the bit in A417 corresponding to the unit number of the Unit where the error occurred is turned ON. Bits 00 to 15 correspond to unit numbers 0 to F. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash.	1: Error 0: No error	Cleared	Cleared	---	A40207
A418 to A423	A41800 to A42315	Special I/O Unit Error, Unit Number Flags	When an error occurs in a data exchange between the CPU Unit and a Special I/O Unit, the Special I/O Unit Error Flag (A40206) will be turned ON. Each bit corresponds to a unit number. Bit 00 in A418 to bit 15 in A423 correspond to unit numbers 0 to 95. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. (Bits A41800 to A42315 correspond to unit numbers 000 to 05F (0 to 95).) The unit number of the Unit where the error occurred is indicated in A417. If the unit number of the Unit is uncertain, none of the flags will be turned ON. (The flag will be turned OFF when the error is cleared.)	1: Error 0: No error	Cleared	Cleared	---	A40206
A424 and A425	A42400 to A42515	Not used.						

Address Words	Address Bits	Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
A426	A42600 to A42611	Interrupt Task Error, Task Number	When A40213 is ON, the content of these bits depends upon the status of A42615 (the Interrupt Task Error Cause Flag). A42615 ON: An attempt was made to refresh a Special I/O Unit's I/O from an interrupt task with IORF(097) while the Unit's I/O is being refreshed by cyclic I/O refreshing (duplicate refreshing). A42600 to A42611: contain the Special I/O Unit's unit number. These bits will be cleared when the error is cleared.	Task number: 000 to 0FF (0 to 255) Unit number: 000 to 05F (0 to 95)	Cleared	Cleared	---	A40213 A42615
	A42615	Interrupt Task Error Cause Flag	When A40213 (the Interrupt Task Error Flag) is ON, this flag indicates the cause of the error. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. A42615 will be ON if a Special I/O Unit was refreshed from the interrupt task while it was already being refreshed.	1: Duplicated refreshing 0: Interrupt task executed over 10 ms	Cleared	Cleared	---	A40213, A42600 to A42611
A427	A42700 to A42715	CPU Bus Unit Setting Error, Unit Number Flags	When an CPU Bus Unit Setting Error occurs, A40203 and the bit in this word corresponding to the Unit's unit number are turned ON. Bits 00 to 15 correspond to unit numbers 0 to F. The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash.	1: Setting error 0: No setting error	Cleared	Cleared	Written when power is turned ON or I/O is recognized	A40203
A428 to A433	A42800 to A43315	Special I/O Unit Setting Error, Unit Number Flags	When a Special I/O Unit Setting Error occurs, A40202 and the bit in these words corresponding to the Unit's unit number are turned ON. Bits 00 to 15 correspond to unit numbers 0 to F. (Bits A42800 to A43315 correspond to unit numbers 000 to 05F (0 to 95).) The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash.	1: Setting error 0: No setting error	Cleared	Cleared	Written when power is turned ON or I/O is recognized	A40202
A434 to A437	A43400 to A43715	Not used.						
A438	A43800 to A43815	Ethernet Communications Error Flags	These flags turn ON to indicate errors in communications for the built-in Ethernet port. A43800: IP Address Setting Error A43802: Unit IP Address Setting Error A43803: FTP Server Setting Error A43805: SMTP Setting Error A43806: SNTP Setting Error A43808: IP Routing Setting Error A43812: IP Address Duplication Error A43813: IP Address Changed during Operation Error A43815: Ethernet Communications Error	1: Error 0: No error	Retained	Cleared	---	A40201
A439	A43900	Ethernet Communications Enabled Flag	Turns ON when it becomes possible to use communications on the built-in Ethernet port after turning ON the power supply (i.e., after initialization).	1: Communications possible 0: Communications not possible	---	Cleared	Written after power is turned ON	---

Address		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related flags, settings
Words	**Bits**							
A440	A44000 to A44015	Max. Interrupt Task Processing Time	Contains the Maximum Interrupt Task Processing Time in units of 0.1 ms. (This value is written after the interrupt task with the max. processing time is executed and cleared when PLC operation begins.)	0000 to FFFF hexadecimal	Cleared	Cleared	See Function column.	---
A441	A44100 to A44115	Interrupt Task With Max. Processing Time	Contains the task number of the interrupt task with the maximum processing time. Hexadecimal values 8000 to 80FF correspond to task numbers 00 to FF. Bit 15 is turned ON when an interrupt has occurred. (This value is written after the interrupt task with the max. processing time is executed and cleared when PLC operation begins.)	8000 to 80FF hexadecimal	Cleared	Cleared	See Function column.	---
A442 to A499	A44200 to A49915	Not used.						

5-9-2 Read/Write Area (Set by User)

Addresses: Word	Addresses: Bit	Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related Flags, Settings
A500	A50012	IOM Hold Bit	Turn this bit ON to preserve the status of the I/O Memory when shifting from PROGRAM to RUN or MONITOR mode or vice versa. The I/O Memory includes the CIO Area, Transition Flags, Timer Flags and PVs, Index Registers, Data Registers, and the Current EM Bank Number. (If the status of the IOM Hold Bit itself is preserved in the PLC Setup (IOM Hold Bit Status), the status of the I/O Memory Area will be retained when the PLC is turned ON or power is interrupted.)	1: Retained 0: Not retained	Retained	See Function column.	See Function column.	PLC Setup (IOM Hold Bit Status setting)
	A50013	Forced Status Hold Bit	Turn this bit ON to preserve the status of bits that have been force-set or force-reset when shifting from PROGRAM to MONITOR mode or vice versa. Bits that have been force-set or force-reset will always return to their default status when shifting to RUN mode. (If the status of the Forced Status Hold Bit itself is preserved in the PLC Setup (Forced Status Hold Bit Status), the status of force-set and force-reset bits will be retained when the PLC is turned ON or power is interrupted.)	1: Retained 0: Not retained	Retained	See Function column.	See Function column.	PLC Setup (Forced Status Hold Bit Status setting)
	A50014	Error Log Reset Bit	Turn this bit ON to reset the Error Log Pointer (A300) to 00. The contents of the Error Log Area itself (A100 to A199) are not cleared. (This bit is automatically reset to 0 after the Error Log Pointer is reset.)	0→1: Clear	Retained	Cleared	---	A100 to A199, A300
	A50015	Output OFF Bit	Turn this bit ON to turn OFF all outputs from Basic I/O Units and Special I/O Units. The INH indicator on the front of the CPU Unit will light while this bit is ON. (The status of the Output OFF Bit is retained through power interruptions.)		Retained	Retained	---	---
A501	A50100 to A50115	CPU Bus Unit Restart Bits	Turn these bits ON to restart (initialize) the CPU Bus Unit with the corresponding unit number. Bits 00 to 15 correspond to unit numbers 0 to F. When a restart bit is turned ON, the corresponding CPU Bus Unit Initializing Flag (A30200 to A30215) will be turned ON. Both the restart bit and initializing flag will be turned OFF automatically when initialization is completed.	0 to 1: Restart 1 to 0: Restart completed Turned OFF by the system when the Unit has been restarted.	Retained	Cleared	---	A30200 to A30215
A502 to A507	A50200 to A50715	Special I/O Unit Restart Bits	Turn these bits ON to restart (initialize) the Special I/O Unit with the corresponding unit number. Bits A50200 to A50715 correspond to unit numbers 0 to 95. When a restart bit is turned ON, the corresponding Special I/O Unit Initializing Flag (A33000 to A33515) will be turned ON. Both the restart bit and initializing flag will be turned OFF automatically when initialization is completed.	0 to 1: Restart 1 to 0: Restart completed Turned OFF by the system when the Unit has been restarted.	Retained	Cleared	---	A33000 to A33515

Addresses		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related Flags, Settings
Word	Bit							
A508	A50809	Differentiate Monitor Completed Flag	ON when the differentiate monitor condition has been established during execution of differentiation monitoring. (This flag will be cleared to 0 when differentiation monitoring starts.)	1: Monitor condition established 0: Not yet established	Retained	Cleared	---	---
	A50811	Trace Trigger Monitor Flag	ON when a trigger condition is established by the Trace Start Bit (A50814). OFF when the next Data Trace is started by the Sampling Start bit (A50815).	1: Trigger condition established 0: Not yet established or not tracing	Retained	Cleared	---	---
	A50812	Trace Completed Flag	ON when sampling of a region of trace memory has been completed during execution of a Trace. OFF when the next time the Sampling Start Bit (A50815) is turned from OFF to ON.	1: Trace completed 0: Not tracing or trace in progress	Retained	Cleared	------	---
	A50813	Trace Busy Flag	ON when the Sampling Start Bit (A50815) is turned from OFF to ON. OFF when the trace is completed.	1: Trace in progress 0: Not tracing (not sampling)	---	---	---	---
	A50814	Trace Start Bit	Turn this bit from OFF to ON to establish the trigger condition. The offset indicated by the delay value (positive or negative) determines which data samples are valid.	1: Trace trigger condition established 0: Not established	---	---	---	---
	A50815	Sampling Start Bit	When a data trace is started by turning this bit from OFF to ON from a Programming Device, the PLC will begin storing data in Trace Memory by one of the three following methods: 1) Data is sampled at regular intervals (10 to 2,550 ms). 2) Data is sampled when TRSM(045) is executed in the program. 3) Data is sampled at the end of every cycle. The operation of A50815 can be controlled only from a Programming Device.	0 to 1: Starts data trace (sampling) Turned ON from Programming Device.	---	---	---	---
A509	A50900 to A50915	Not used.						
A510 and A511	A51000 to A51115	Start-up 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. A51000 to A51007: Second (00 to 59) A51008 to A51015: Minute (00 to 59) A51100 to A51107: Hour (00 to 23) A51108 to A51115: Day of month (01 to 31)	See Function column.	Retained	See Function column.	Written when power is turned ON	---
A512 and A513	A51200 to A51315	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. A51200 to A51207: Second (00 to 59) A51208 to A51215: Minute (00 to 59) A51300 to A51307: Hour (00 to 23) A51308 to A51315: Day of month (01 to 31) (These words are not cleared at startup.)	See Function column.	Retained	Retained	Written at power interruption	---

Addresses		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related Flags, Settings
Word	Bit							
A514	A51400 to A51415	Number of Power Interrup-tions	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. (This word is not cleared at start-up, but it is cleared when the Memory Corrup-tion Detected Flag (A39511) goes ON.)	0000 to FFFF hexadecimal	Retained	Retained	Written when power is turned ON	A39511
A515 to A517		Operation Start Time	These words contain the time that oper-ation was started by changing the oper-ating mode to RUN or MONITOR mode. A51500 to A51507: Seconds (00 to 59) A51508 to A51515: Minutes (00 to 59) A51600 to A51607: Hour (00 to 23) A51608 to A51615: Day (01 to 31) A51700 to A51707: Month (01 to 12) A51708 to A51715: Year (00 to 99) **Note** These words contain the previ-ous start time from power ON until operation is started.	See function description at left.	Retained	Retained	See func-tion descrip-tion at left.	---
A518 to A520		Operation End Time	These words contain the time that oper-ation was stopped by changing the operating mode to PROGRAM mode. A51800 to A51807: Seconds (00 to 59) A51808 to A51815: Minutes (00 to 59) A51900 to A51907: Hour (00 to 23) A51908 to A51915: Day (01 to 31) A52000 to A52007: Month (01 to 12) A52008 to A52015: Year (00 to 99) **Note** If an error occurs in operation, the time of the error will be stored. If the operating mode is then changed to PROGRAM mode, that time will be stored.	See function description at left.	Retained	Retained	See func-tion descrip-tion at left.	---
A518 to A520	A51500 to A52215	Not used.						
A523	A52300 to A52315	Total Power ON Time	Contains the total time that the PLC has been on in 10-hour units. The data is stored in binary and it is updated every 10 hours. To reset this value, overwrite the current value with 0000. (This word is not cleared at start-up, but it is cleared to 0000 when the Memory Corruption Detected Flag (A39511) goes ON.)	0000 to FFFF hexadecimal	Retained	Retained	---	---
A526	A52600	RS-232C Port Restart Bit	Turn this bit ON to restart the RS-232C port. (Do not use this bit when the port is operating in peripheral bus mode.) This bit is turned OFF automatically when the restart processing is com-pleted.	0 to 1: Restart	Retained	Cleared	---	---
A527	A52700 to A52715	Not used.						

Addresses		Name	Function	Settings	Status after mode change	Status at startup	Write timing	Related Flags, Settings
Word	Bit							
A528	A52800 to A52807	RS-232C Port Error Flags	These flags indicate what kind of error has occurred at the RS-232C port; they are automatically turned OFF when the RS-232C port is restarted. (These flags are not valid in peripheral bus mode and only bit 5 is valid in NT Link mode.) Bits 0 and 1: Not used. Bit 2: ON when there was a parity error. Bit 3: ON when there was a framing error. Bit 4: ON when there was an overrun error. Bit 5: ON when there was a timeout error. Bits 6 and 7: Not used.	See Function column.	---	---	---	---
A529	A52900 to A52915	FAL/FALS Number for System Error Simulation	Set a dummy FAL/FALS number to use to simulate the system error using FAL(006) or FALS(007). When FAL(006) or FALS(007) is executed and the number in A529 is the same as the one specified in the operand of the instruction, the system error given in the operand of the instruction will be generated instead of a user-defined error.	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	---	---
A530	A53000 to A53015	Power Interruption Disable Setting	Set to A5A5 hex to disable power interrupts (except the Power OFF Interrupt task) between DI(693) and EI(694) instructions.	A5A5 hex: Masking power interruption processing enabled Other: Masking power interruption processing not enabled.	Cleared	Cleared	---	---
A531 to A594	A53100 to A59415	Not used.						
A595 and A596	A59500 to A59615	IR00 Output for Background Execution	When an index register is specified as the output for an instruction processed in the background, A595 and A596 receive the output instead of IR00.	0000 0000 to FFFF FFFF hex (A596 contains the leftmost digits.)	Cleared	Cleared	---	---
A597	A59700 to A59715	DR00 Output for Background Execution	When a data register is specified as the output for an instruction processed in the background, A597 receives the output instead of DR00.	0000 to FFFF hex	Cleared	Cleared	---	---
A598	A59801	Equals Flag for Background Execution	Turns ON if matching data is found for an SRCH(181) instruction executed in the background.	1: Search data found in table 0: Search data not found	Cleared	Cleared	---	---
A599 to A619	A59900 to A61915	Not used.						
A620 to A959	A62000 to A95915	Not used.						

5-10 TR (Temporary Relay) Area

The TR Area contains 16 bits with addresses ranging from TR0 to TR15. These temporarily store the ON/OFF status of an instruction block for branching. TR bits are useful when there are several output branches and interlocks cannot be used.

It is not necessary to consider TR bits when displaying ladder diagrams on the NE Programmer.

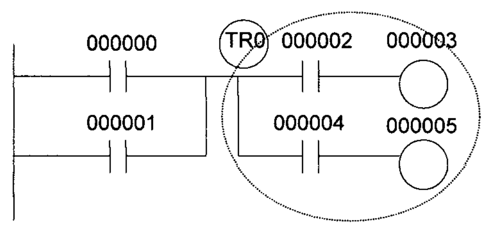
The TR bits can be used as many times as required and in any order required as long as the same TR bit is not used twice in the same instruction block.

TR bits can be used only with the OUT and LD instructions. OUT instructions (OUT TR0 to OUT TR15) store the ON OFF status of a branch point and LD instructions recall the stored ON OFF status of the branch point.

TR bits cannot be changed from a Programming Device.

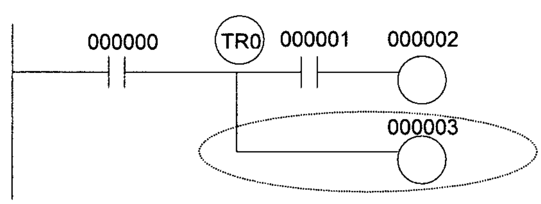
Examples

In this example, a TR bit is used when two outputs have been directly connected to a branch point.



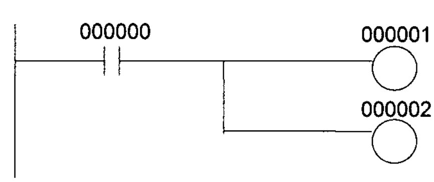
Instruction	Operand
LD	**000000**
OR	**000001**
OUT	TR **0**
AND	**00000**2
OUT	**00000**3
LD	TR **0**
AND	**00000**4
OUT	**00000**5

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

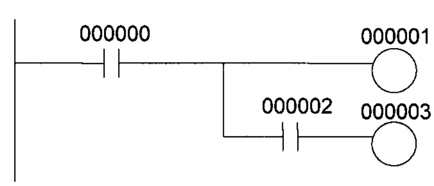


Instruction	Operand
LD	**000000**
OUT	TR **0**
AND	**000001**
OUT	**00000**2
LD	TR **0**
OUT	**00000**3

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



Instruction	Operand
LD	000000
OUT	000001
OUT	000002



Instruction	Operand
LD	000000
OUT	000001
AND	000002
OUT	000003

5-11 Timer Area

The 4,096 timer numbers (T0000 to T4095) are shared by the TIMX, TIMHX(551), TMHHX(552), and TTIMX(555) instructions. Timer Completion Flags and present values (PVs) for these instructions are accessed with the timer numbers. (The TIMLX(553) and MTIMX(554) instructions do not use timer numbers.)

When a timer number is used in an operand that requires bit data, the timer number accesses the Completion Flag of the timer. When a timer number is used in an operand that requires word data, the timer number accesses the PV of the timer. Timer Completion Flags can be used as often as necessary as normally open and normally closed conditions and the values of timer PVs can be read as normal word data.

The refresh method for timer PVs is binary.

Precautions for TIMER Variables

Compatible Instructions

TIMER variables can be used in the TIMX and TIMHX(551) instructions. TIMER variables cannot be used in the TMHHX(552), TIMLX(553), and MTIMX(554) instructions.

Automatic Allocation

The NE Programmer can set timers that are automatically allocated.

Usage in Function Blocks

If a timer number is specified in a function block, all instances of the function block will used the same timer, so the timer may not operate correctly.

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

The following table shows when timer PVs and Completion Flags will be reset.

Instruction name	Effect on PV and Completion Flag			Operation in Jumps and Interlocks	
	Mode change[1]	PLC start-up[1]	CNR(545) or CNRX(547)	Jumps (JMP-JME) or Tasks on standby	Interlocks (IL-ILC)
TIMER: TIMX	PV → 0 Flag → OFF	PV → 0 Flag → OFF	PV → 9999 Flag → OFF	PVs refreshed in operating timers	PV → SV (Reset to SV.) Flag → OFF
HIGH-SPEED TIMER: TIMHX(551)					
ONE-MS TIMER: TMHHX(552)					
ACCUMULATIVE TIMER: TTIMX(555)				PV Maintained	PV Maintained

Note

1. If the IOM Hold BIt (A50012) is ON, the PV and Completion Flag will be retained when a fatal error occurs or the operating mode is changed from PROGRAM mode to RUN or MONITOR mode or vice-versa. The PV and Completion Flag will be cleared when power is cycled.
2. If the IOM Hold BIt (A50012) is ON and the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, the PV and Completion Flag will be retained when the PLC's power is cycled.
3. Since the TIMLX(553) and MTIMX(554) instructions do not use timer numbers, they are reset under different conditions. Refer to the descriptions of these instructions for details.

4. The present value of TIMX, TIMHX(551), TMHHX(552), and MHHX(552) timers programmed with timer numbers 0000 to 2047 will be updated even when jumped between JMP and JME instructions or when in a task that is on standby. The present value of timers programmed with timer numbers 2048 to 4095 will be held when jumped or when in a task that is on standby.

Forcing Bit Status

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.

5-12 Counter Area

The 4,096 counter numbers (C0000 to C4095) are shared by the CNTX and CNTRX(548) instructions. Counter Completion Flags and present values (PVs) for these instructions are accessed with the counter numbers.

When a counter number is used in an operand that requires bit data, the counter number accesses the Completion Flag of the counter. When a counter number is used in an operand that requires word data, the counter number accesses the PV of the counter.

The refresh method for counter PVs is binary.

Precautions for COUNTER Variables

Setting Counter Numbers for Automatic Allocation

The NE Programmer can set the counter numbers that are automatically allocated. (Set the counter numbers in the Configuration Setting Window - Build Tab Page.)

Usage in Function Blocks

If a counter number is specified in a function block, all instances of that function block will use the same counter, so the counter may not operate correctly.

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

The following table shows when counter PVs and Completion Flags will be reset.

Instruction name	Effect on PV and Completion Flag					
	Reset	Mode change	PLC startup	Reset Input	CNR(545) or CNRX(548)	Interlocks (IL-ILC)
COUNTER: CNTX	PV → 0000 Flag → OFF	Maintained	Maintained	Reset	Reset	Maintained
REVERSIBLE COUNTER: CNTRX(548)						
COUNTER WAIT: CNTWX(818)						

5-13 Data Memory (DM) Area

The DM Area contains 32,768 words with addresses ranging from D00000 to D32767. This data area is used for general data storage and manipulation and is normally accessible only by word.

Data in the DM Area is retained when the PLC's power is cycled or the PLC's operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa.

Although bits in the DM Area cannot be accessed directly, the status of these bits can be accessed with the BIT TEST instructions, TST(350) and TSTN(351).

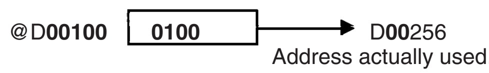
Bits in the DM Area cannot be force-set or force-reset.

Indirect Addressing

Words in the DM Area can be indirectly addressed in two ways: binary-mode and BCD-mode.

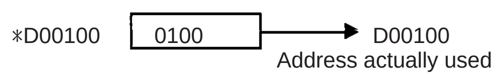
Binary-mode Addressing (@D)

When a "@" character is input before a DM address, the content of that DM word is treated as binary and the instruction will operate on the DM word at that binary address. The entire DM Area (D00000 to D32767) can be indirectly addressed with hexadecimal values 0000 to 7FFF.



BCD-mode Addressing (∗D)

When a "∗" character is input before a DM address, the content of that DM word is treated as BCD and the instruction will operate on the DM word at that BCD address. Only part of the DM Area (D00000 to D09999) can be indirectly addressed with BCD values 0000 to 9999.



DM Area Allocation to Special Units

Parts of the DM Area are allocated to Special I/O Units and CPU Bus Units for functions such as initial Unit settings. The timing for data transfers is different for these Units, but may occur at any of the three following times.

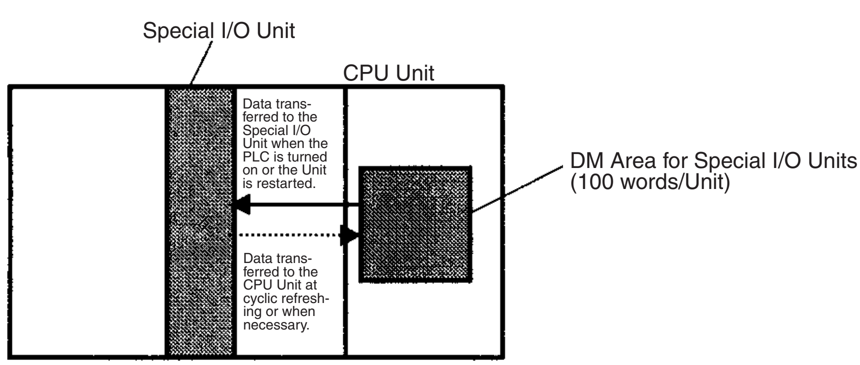
1,2,3...

1. Transfer data when the PLC's power is turned on or the Unit is restarted.
2. Transfer data once each cycle.
3. Transfer data when required.

Refer to the Unit's Operation Manual for details on data transfer timing.

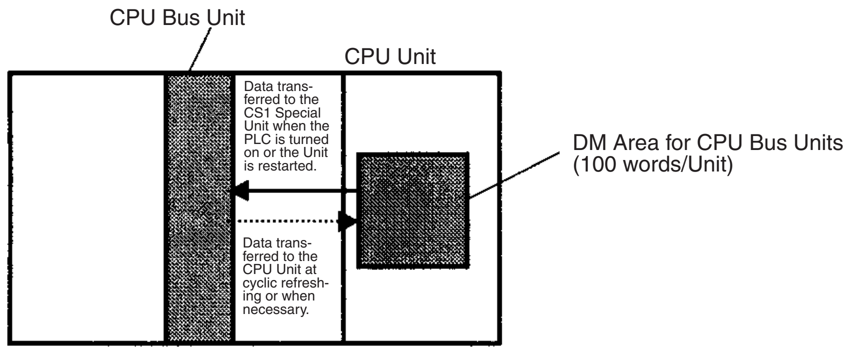
Special I/O Units (D20000 to D29599)

Each Special I/O Unit is allocated 100 words (based on unit numbers 0 to 95). Refer to the Unit's Operation Manual for details on the function of these words.



CPU Bus Units (D30000 to D31599)

Each CPU Bus Unit is allocated 100 words (based on unit numbers 0 to F). Refer to the Unit’s Operation Manual for details on the function of these words. With some CPU Bus Units such as Ethernet Units, initial settings must be registered in the CPU Unit’s Parameter Area; this data can be registered with a Programming Device other than a Programming Console.



5-14 Extended Data Memory (EM) Area

EM Area addresses range from E0_00000 to E3_32767. This data area is used for general data storage and manipulation and is normally accessible only by word.

Data in the EM Area is retained when the PLC’s power is cycled or the PLC’s operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa.

Although bits in the EM Area cannot be accessed directly, the status of these bits can be accessed with the BIT TEST instructions, TST(350) and TSTN(351).

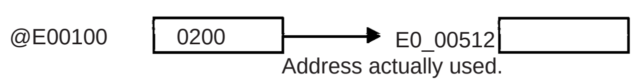
Bits in the EM Area cannot be force-set or force-reset.

Indirect Addressing

Words in the EM Area can be indirectly addressed in two ways: binary-mode and BCD-mode.

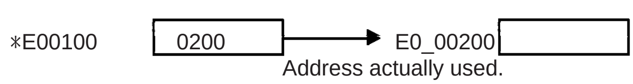
Binary-mode Addressing (@E)

When a “@” character is input before a EM address, the content of that EM word is treated as binary and the instruction will operate on the EM word in the same bank at that binary address.



BCD-mode Addressing (*E)

When a “*” character is input before a EM address, the content of that EM word is treated as BCD and the instruction will operate on the EM word in the same bank at that BCD address. Only part of the EM bank (E00000 to E09999) can be indirectly addressed with BCD values 0000 to 9999.



5-15 Index Registers

The sixteen Index Registers (IR0 to IR15) are used for indirect addressing. Each Index Register can hold a single PLC memory address, which is the absolute memory address of a word in I/O memory. Use MOVR(560) to convert a regular data area address to its equivalent PLC memory address and write that value to the specified Index Register. (Use MOVRW(561) to set the PLC memory address of a timer/counter PV in an Index Register.)

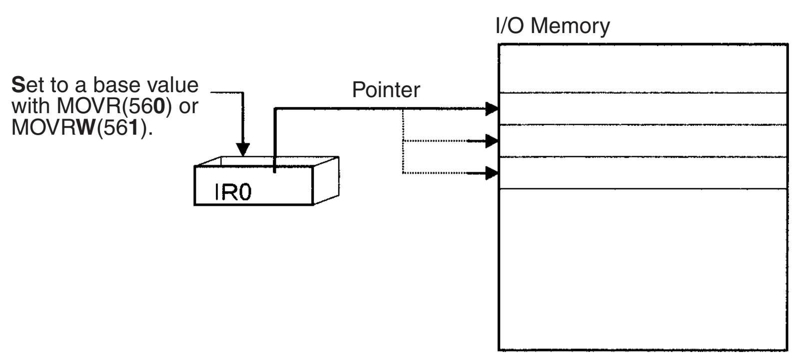
Note Refer to *Appendix E Memory Map of PLC Memory Addresses* for more details on PLC memory addresses.

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.
- In addition to basic indirect addressing, the PLC memory address in an Index Register can be offset with a constant or Data Register, auto-incremented, or auto-decremented. These functions can be used in loops to read or write data while incrementing or decrementing the address by one each time that the instruction is executed.

With the offset and increment/decrement variations, the Index Registers can be set to base values with MOVR(560) or MOVRW(561) and then modified as pointers in each instruction.



Note It is possible to specify regions outside of I/O memory and generate an Illegal Access Error when indirectly addressing memory with Index Registers. Refer to *Appendix E Memory Map of PLC Memory Addresses* for details on the limits of PLC memory addresses.

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.)

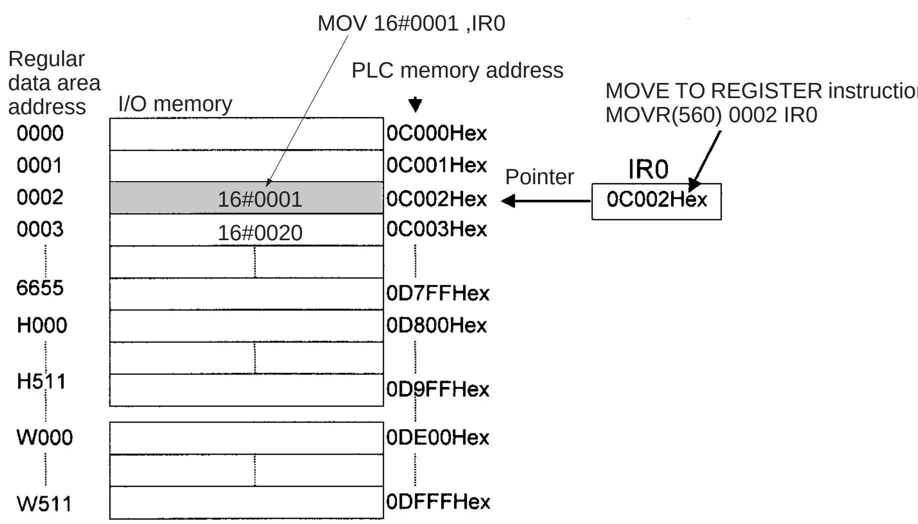
Variation	Function	Syntax	Example	
Indirect addressing	The content of IR☐ is treated as the PLC memory address of a bit or word.	,IR☐	LD ,IR0	Loads the bit at the PLC memory address contained in IR0.
Indirect addressing with constant offset	The constant prefix is added to the content of IR☐ and the result 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.

Variation	Function	Syntax	Example	
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 0002) in an Index Register (IR0), use the Index Register in an instruction, and use the auto-increment variation.

MOVR(560)	0002	IR0	Stores the PLC memory address of CIO 0002 in IR0.
MOV(021)	16#0001	,IR0	Writes #0001 to the PLC memory address contained in IR0.
MOV(021)	16#0020	+1,IR0	Reads the content of IR0, adds 1, and writes #0020 to that PLC memory address.



Note 1. Auto-incrementing and auto-decrementing for index registers are performed whenever an instruction using them is executed. Care is required in application, particularly with instructions like OUT, which are executed every cycle. Refer to *1-1-5 Inputting Data in Operands* in the *Instructions Reference* for details.

Example
MOVR 000013 IR0
LD P_Off
OUT ,IR0+
With the above programming, OUT will turn OFF CIO 000013 and IR0 will be incremented to point to CIO 000014.

MOVR 000013 IR0
LD P_Off
SET ,IR0+

SET is executed only when the input condition is ON. With the above programming, SET will not be executed and IR0 will not be incremented.

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

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

1,2,3...

1. Word Operand:
   MOVR(560) 0000 IR2
   MOV(021) D00000 ,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(560) sets the PLC memory address of CIO 0002 in IR2 and the MOV(021) instruction copies the contents of D00000 to CIO 0002.
2. Bit Operand:
   MOVR(560) 000013 ,IR2
   SET +5 , IR2

   When the operand is treated as a bit, the leftmost 7 digits of the Index Register specify the word address and the rightmost digit specifies the bit number. In this example, MOVR(560) sets the PLC memory address of CIO 000013 (0C000D hex) in IR2. The SET instruction adds +5 from bit 13 to this PLC memory address, so it turns ON bit CIO 000102.

Index Register Initialization

The Index Registers will be cleared in the following cases:

1,2,3...

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

IOM Hold Bit Operation

If the IOM Hold Bit (A50012) is ON, the Index Registers won't be cleared when a FALS error occurs or the operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa.

If the IOM Hold Bit (A50012) is ON, and the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, and if the Index Registers are not set to be shared between tasks (default setting), Index Registers will be held in the following way when power is interrupted. For tasks that were completed before power was interrupted, the values for the cycle during which power was interrupted will be held. For tasks that were not completed before power was interrupted, the values for the cycle before the cycle during which power was interrupted will be held. For example, in a program with three tasks, tasks 0, 1, and 2, if power is interrupted in the nth cycle during execution of task 1, then the execution result for the nth cycle of task 0 and the execution results for the (n–1)th cycle of tasks 1 and 2 will be held.

If the IOM Hold Bit (A50012) is ON, the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, and the Index Registers are set to be shared between tasks, Index Registers will not be held when the PLC's power supply is reset (ON →OFF →ON). The Index Registers may take undefined values. Be sure to set the values before continuing.

Forcing Bit Status

Bits in Index Registers cannot be force-set and force-reset.

Direct Addressing

When an Index Register is used as an operand without a "," prefix, the instruction will operate on the contents of the Index Register itself (a two-word or "double" value). Index Registers can be directly addressed only in the instructions shown in the following table. Use these instructions to operate on the Index Registers as pointers.

The Index Registers cannot be directly addressed in any other instructions, although they can usually be used for indirect addressing.

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

The SRCH(181), MAX(182), and MIN(183) instructions can output the PLC memory address of the word with the desired value (search value, maximum, or minimum) to IR0. In this case, IR0 can be used in later instructions to access the contents of that word.

Precautions

Do not use Index Registers until a PLC memory address has been set in the register. The pointer operation will be unreliable if the registers are used without setting their values.

The values in Index Registers are unpredictable at the start of an interrupt task. When an Index Register will be used in an interrupt task, always set a PLC memory address in the Index Register with MOVR(560) or MOVRW(561) before using the register in that task.

Each Index Register task is processed independently, so they do not affect each other. For example, IR0 used in Task 1 and IR0 used in Task 2 are different. Consequently, each Index Register task has 16 Index Registers.

Limitations when Using Index Registers

1,2,3...

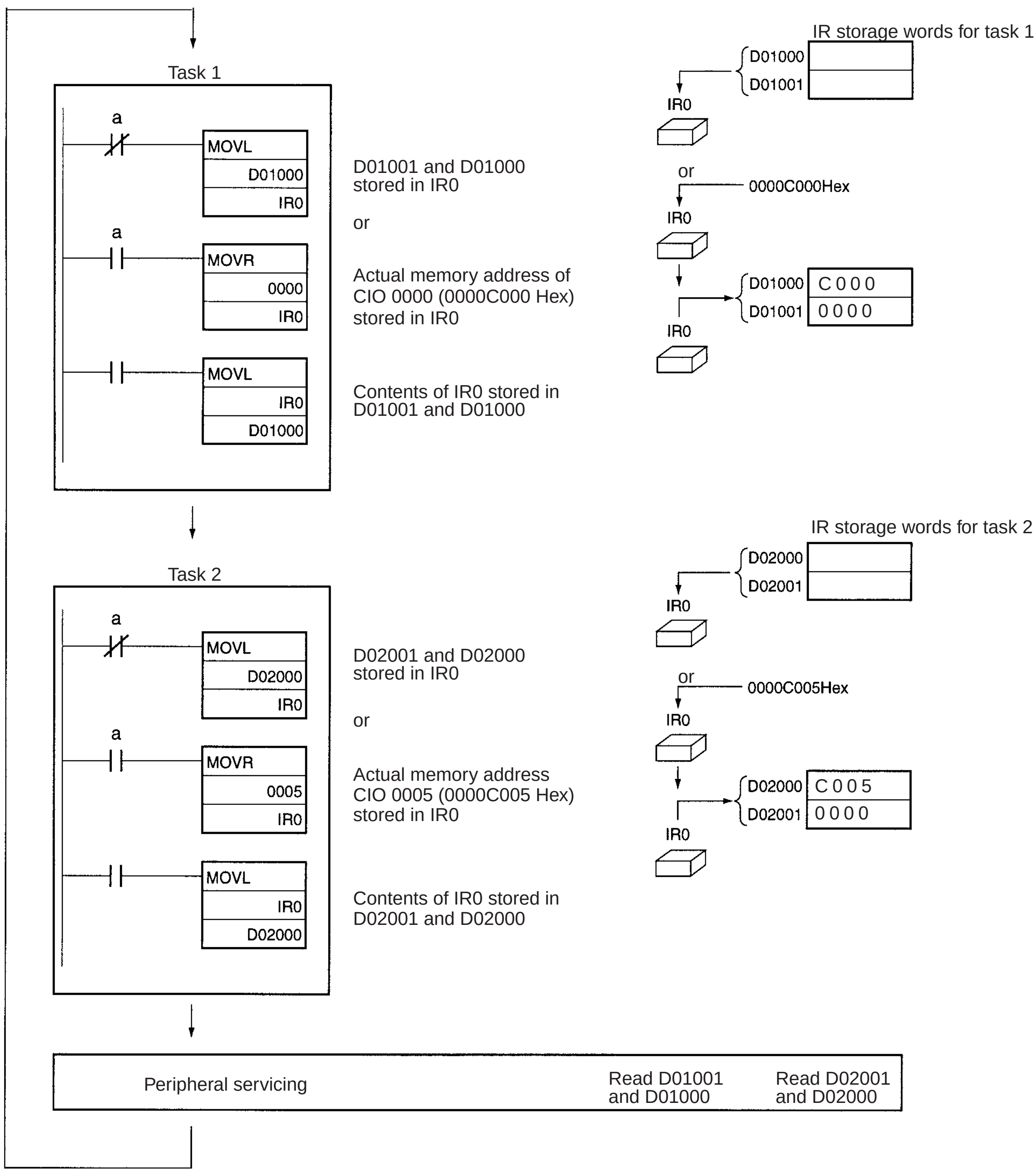
1. It is only possible to read the Index Register for the last task executed within the cycle from the Programming Devices. If using Index Registers with the same number to perform multiple tasks, it is only possible with the Programming Devices to read the Index Register value for the last task performed within the cycle from the multiple tasks. It is not possible to write the Index Register value from the Programming Devices.
2. It is not possible to either read from or write to the Index Registers using CIP messages.
3. The Index Registers can be shared between tasks.

Monitoring and Sharing Index Registers

It is possible to monitor or share Index Registers as follows:

To use the Programming Devices to monitor the final Index Register values for each task, or to monitor the Index Register values using CIP messages, 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 Programming Devices or CIP messages.

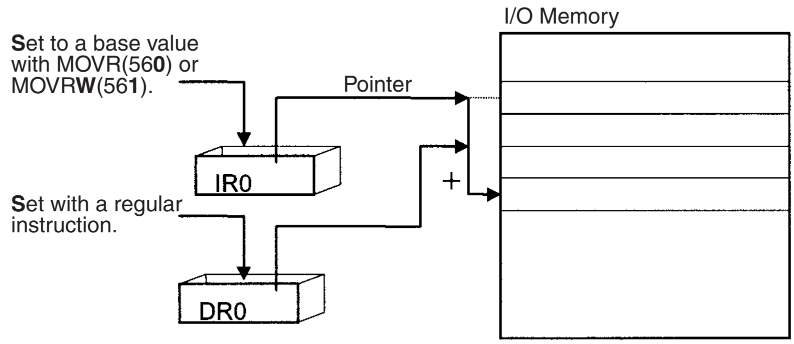
Note Be sure to use PLC memory addresses in Index Registers.



5-16 Data Registers

The sixteen Data Registers (DR0 to DR15) are used to offset the PLC memory addresses in Index Registers when addressing words indirectly.

The value in a Data Register can be added to the PLC memory address in an Index Register to specify the absolute memory address of a bit or word in I/O memory. Data Registers contain signed binary data, so the content of an Index Register can be offset to a lower or higher address.



Examples

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(021) #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

Data Register Initialization

The Data Registers will be cleared in the following cases:

1,2,3...
1. The operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa and the IOM Hold Bit is OFF.
2. The PLC's power supply is cycled and the IOM Hold Bit is OFF or not protected in the PLC Setup.

IOM Hold Bit Operation

If the IOM Hold Bit (A50012) is ON, the Data Registers won't be cleared when a FALS error occurs or the operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa. Data Registers are cleared when power is interrupted.

If the IOM Hold Bit (A50012) is ON, and the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, and if the Data Registers are not set to be shared between tasks (default setting), Data Registers will be held in the following way when power is interrupted. For tasks that were completed before power was interrupted, the values for the cycle during which power was interrupted will be held. For tasks that were not completed before power was interrupted, the values for the cycle before the cycle during which power was interrupted will be held. For example, in a program with three tasks, tasks 0, 1, and 2, if power is interrupted in the nth cycle during execution of task 1, then the execution result for the nth cycle of task 0 and the execution results for the (n–1)th cycle of tasks 1 and 2 will be held.

If the IOM Hold Bit (A50012) is ON, the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, and the Data Registers are set to be shared between tasks, Data Registers will not be held when the PLC's power supply is reset (ON →OFF →ON). The Data Registers may take undefined values. Be sure to set the values before continuing.

Forcing Bit Status

Bits in Data Registers **cannot** be force-set and force-reset.

Precautions

Data Registers are normally local to each task. For example, DR0 used in task 1 is different from DR0 used in task 2.

The content of Data Registers cannot be accessed (read or written) from a Programming Device.

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.

5-17 Task Flags

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

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

Task Flag Initialization

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

1,2,3...

1. The operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa.
2. The PLC's power supply is cycled.

Forcing Bit Status

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

5-18 Condition Flags

These flags include the Arithmetic Flags such as the Error Flag and Equals Flag which indicate the results of instruction execution. In earlier PLCs, these flags were in the SR Area.

The Condition Flags are specified with labels, such as CY and ER, or with symbols, such as P_Carry and P_Instr_Error, rather than addresses. The status of these flags reflects the results of instruction execution, but the flags are read-only; they cannot be written directly from instructions or Programming Devices.

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

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

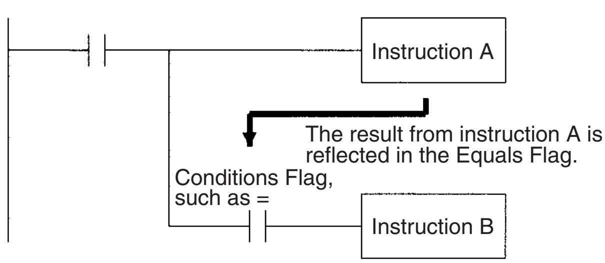
Summary of the Condition Flags

The following table summarizes the functions of the Condition Flags, although the functions of these flags will vary slightly from instruction to instruction. Refer to the description of the instruction for complete details on the operation of the Condition Flags for a particular instruction.

Name	Symbol	Function
Error Flag	P_ER	Turned ON when the operand data in an instruction is incorrect (an 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 (A29508) 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 (A429510) 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 the result of a calculation is 0.
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 (sign 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.
Always ON Flag	P_On	Always ON. (Always 1.)
Always OFF Flag	P_Off	Always OFF. (Always 0.)

Using the Condition Flags

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



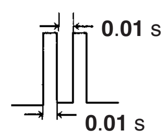
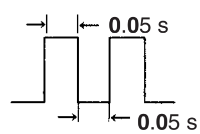
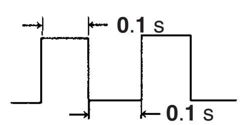
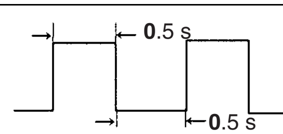
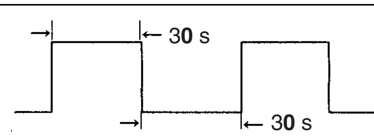
Instruction	Operand
LD	
Instruction A	
AND	=
Instruction B	

Since the Condition Flags are shared by all of the instructions, program operation can be changed from its expected course by interruption of a single task. Be sure to consider the effects of interrupts when writing the program. For details, refer to *2-2-1 Condition Flags* under *2-2 Precautions* in the *SYSMAC CS/CJ-series Programmable Controllers Programming Manual* (W394).

The Condition Flags are cleared when the program switches tasks, so the status of a Condition Flag cannot be passed to another task. For example the status of a flag in task 1 cannot be read in task 2. (The flag's status must be transferred to a bit.)

5-19 Clock Pulses

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

Name	Label	Symbol	Operation	
0.02 s Clock Pulse	0.02s	P_0_02s		ON for 0.01 s OFF for 0.01 s
0.1 s Clock Pulse	0.1s	P_0_1s		ON for 0.05 s OFF for 0.05 s
0.2 s Clock Pulse	0.2s	P_0_2s		ON for 0.1 s OFF for 0.1 s
1 s Clock Pulse	1s	P_1s		ON for 0.5 s OFF for 0.5 s
1 min Clock Pulse	1min	P_1min		ON for 30 s OFF for 30 s

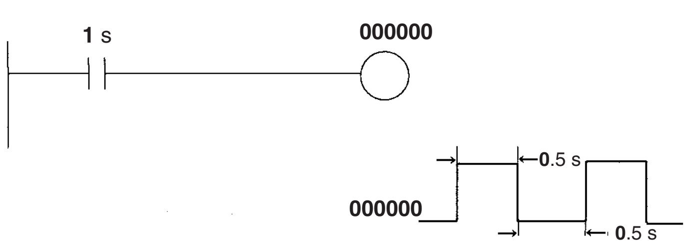
The Clock Pulses are specified with labels (or symbols) rather than addresses.

The Clock Pulses are read-only; they cannot be overwritten from instructions or Programming Devices.

The Clock Pulses are cleared at the start of operation.

Using the Clock Pulses

The following example turns CIO 000000 ON and OFF at 0.5 s intervals.



Instruction	Operand
LD	**1** s
OUT	**000000**

5-20 Parameter Areas

Unlike the data areas in I/O memory which can be used in instruction operands, the Parameter Area can be accessed only from a Programming Device. The Parameter Area is made up of the following parts.

- PLC Setup
- Registered I/O Tables
- Built-in Ethernet Setup
- Routing Table (FINS)
- CPU Bus Unit Setup Area

5-20-1 PLC Setup

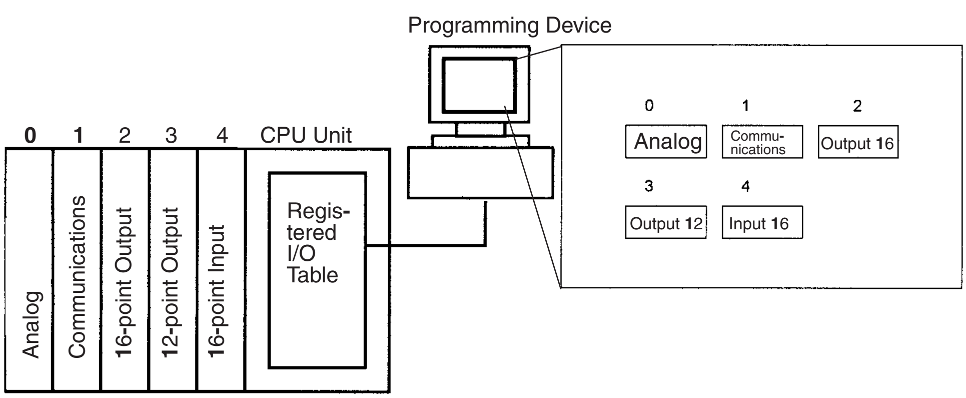
The user can customize the basic specifications of the CPU Unit with the settings in the PLC Setup. The PLC Setup contains settings such as the serial port communications settings and minimum cycle time setting.

Refer to *4-1 General Tab Page* for details on the PLC Setup settings and refer to the Programming Device's Operation Manual for details on changing these settings.

5-20-2 Registered I/O Tables

The Registered I/O Tables are tables in the CPU Unit that contain the information on the model and slot location of all of the Units mounted to the CPU Rack, Expansion I/O Racks, and Slave Racks. The I/O Tables are written to the CPU Unit with a Programming Device operation.

The CPU Unit allocates I/O memory to actual I/O points (on Basic I/O Units or Remote I/O Units) and CPU Bus Units based on the information in the Registered I/O Tables. Refer to the Programming Device's Operation Manual for details on registering the I/O Tables.



The I/O Verification Error Flag (A40209) will be turned ON if the models and locations of the Units actually mounted to the PLC (CPU Rack, Expansion I/O Racks, and Slave Racks) do not match the information in the Registered I/O Table.

5-20-3 Built-in Ethernet Setup

The Built-in Ethernet Setup contains the settings for the Ethernet port built into the CPU Unit. These setting can be changed from the NE Programmer.

5-20-4 Routing Table (FINS)

When transferring data between FINS networks, it is necessary to create a table in each CPU Unit that shows the communications route from the local PLC's Communications Unit to each destination PLC in the other networks. These tables of communications routes are called “Routing Tables.”

Each Routing Table is made up of a relay network table and a local network table.

The Routing Tables are not normally used with NE1S Series PLCs. The tables are used when making a network connection to CS1 Series PLCs.

5-20-5 CPU Bus Unit Setup Area

This area contains system settings for CPU Bus Units managed by the CPU Unit. The contents of the system settings are different for each kind of CPU Bus Unit. Refer to the CPU Bus Unit’s Operation Manual for details on these settings.

Example 1: Controller Link Unit

When the CPU Bus Unit is a Controller Link Unit, this area contains data link parameters for custom data links and network parameters.

Example 2: Ethernet Unit

When the CPU Bus Unit is an Ethernet Unit, this area contains the settings required for the Unit to operate as an Ethernet node, such as the IP address table settings.

SECTION 6
Operating Procedures

This section outlines the steps required to assemble and operate a CS-series PLC system.

6-1 Introduction

The following procedure outlines the recommended steps to follow when preparing CS-series PLCs for operation.

1,2,3...

1. Installation

   Set the DIP switches on the front of each Unit as required.
   Mount the CPU Unit, Power Supply Unit, and other Units to the Backplane.
   Install the Memory Card if required.

   See *7-2 Installation* for details.

2. Wiring

   Connect the power supply wiring, I/O wiring, and Programming Device (NE Programmer). Connect communications wiring as required.

   See *7-3 Wiring* for details on power supply and I/O wiring.
   See *6-2 Examples* for details on connecting Programming Devices.

3. Initial Settings (Hardware)

   Set the DIP switches an Rotary switches on the front of the CPU Unit and other Units.

   See *SECTION 3 CPU Unit* for details.

4. Checking Initial Operation

   Turn the power on after checking the power supply wiring and voltage. Check the Power Supply Unit's POWER indicator.

5. Registering the I/O Table

   Check the Units to verify that they are installed in the right slots. With the PLC in PROGRAM mode, register the I/O table from the Programming Device (NE Programmer). (Another method is to create the I/O table in NE Programmer and transfer it to the CPU Unit.)

   See *13-5 Online Operations for I/O Tables* for details.

6. PLC Setup Settings

   With the PLC in PROGRAM mode, change the settings in the PLC Setup as necessary from the Programming Device (NE Programmer). (Another method is to change the PLC Setup in NE Programmer and transfer it to the CPU Unit.)

   See *6-2 Examples* for details.

7. DM Area Settings for CPU Bus Units and Special I/O Units

   a) Use a Programming Device (NE Programmer) to make any necessary settings in the parts of the DM Area that are allocated to Special I/O Units and CPU Bus Units.

   b) Reset the power (ON → OFF → ON) or toggle the Restart Bit for each Unit. See the Unit's Operation Manual for details.

8. Writing the Program

   Write the program with the NE Programmer.

9. Transferring the Program (NE Programmer Only)

   With the PLC in PROGRAM mode, transfer the program from NE Programmer to the CPU Unit.

10. Testing Operation
    a) Checking I/O Wiring

Output wiring	With the PLC in PROGRAM mode, force-set output bits and check the status of the corresponding outputs.
Input wiring	Activate sensors and switches and either check the status of the indicators on the Input Unit or check the status of the corresponding input bits with the Programming Device's Bit/Word Monitor operation.

    b) Auxiliary Area Settings (As Required)
    Check operation of special Auxiliary Area Settings such as the following:

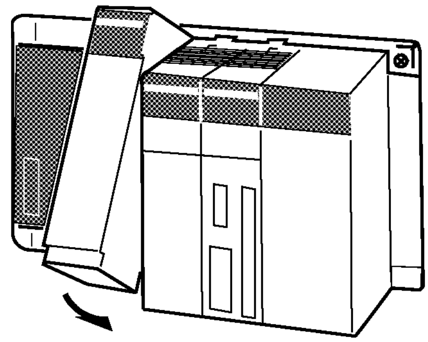
Output OFF Bit	When necessary, turn ON the Output OFF Bit (A50015) from the program and test operation with the outputs forced OFF.
Hot Start Settings	When you want to start operation (switch to RUN mode) without changing the contents of I/O memory, turn ON the IOM Hold Bit (A50012).

    c) Trial Operation
    Test PLC operation by switching the PLC to MONITOR mode.
    d) Monitoring and Debugging
    Monitor operation from the Programming Device. Use functions such as force-setting/force-resetting bits, tracing, and online editing to debug the program.
    See *SECTION 7 Program Transfer, Trial Operation, and Debugging* in the *CS/CS Series Programmable Controllers Programming Manual* (W394) for details.
11. Saving and Printing the Program
12. Running the Program
    Switch the PLC to RUN mode to run the program.

6-2 Examples

1. Installation

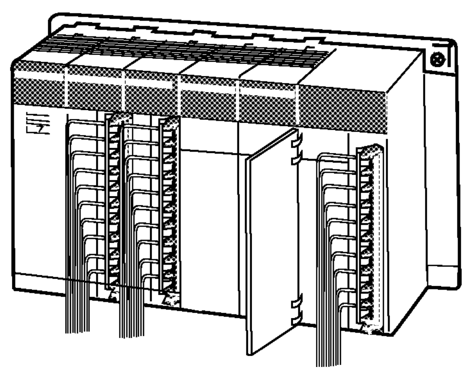
Mount the Backplane and install each Unit. When necessary, install the Memory Card.



Make sure that the total power consumption of the Units is less than the maximum capacity of the Power Supply Unit.

2. Wiring

Connect the power supply and I/O wiring.



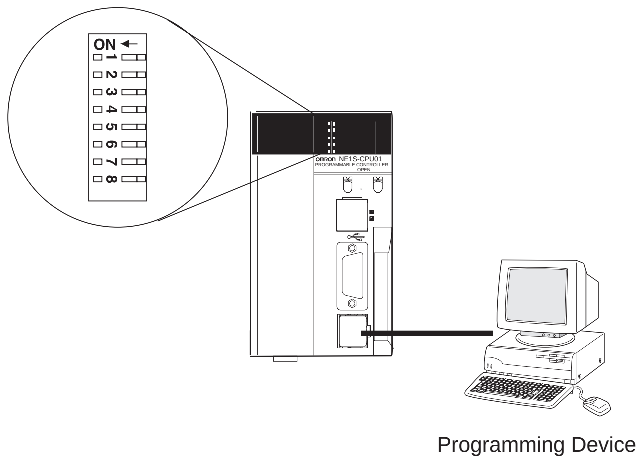
Note When 220-V AC power (200 to 240 V AC) is being supplied, be sure to remove the jumper bar that shorts the voltage selector terminals. The Power Supply Unit will be damaged if 220 V AC is supplied with the jumper bar connected.

3. Initial Settings (Hardware)

Make necessary hardware settings such as the DIP switch settings on the CPU Unit. In particular, be sure that the settings for the USB port and RS-232C port are correct.

For example, when a Programming Device is connected to the USB port, pin 4 is turned OFF and when a Programming Device is connected to the RS-232C port, pin 5 is turned ON.

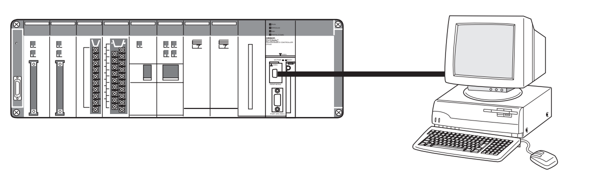
Note When devices other than Programming Devices are connected to the USB port and RS-232C port, turn ON pin 4 and turn OFF pin 5.



4. Checking Initial Operation

Use the following procedure to turn on the PLC and check initial operation.

1,2,3...
1. Connect the Programming Device to the CPU Unit's USB port.



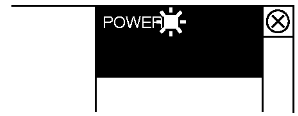
2. Check the power supply wiring and voltage and turn on the power.

   In particular, check that the voltage selector terminals (just below the power input terminals on the Power Supply Unit) are open when 220 V AC power is being supplied. These terminals should be connected only when 110 V AC power is being supplied.

⚠ **Caution** The Unit will be damaged if 220 V AC power is supplied and the voltage selector terminals are connected.

Note The CPU Unit is set by default to start in PROGRAM mode when the power is turned ON. This is the default setting for the operating mode in the PLC Setup. To start operation in RUN mode when the power is turned ON, set the startup mode in the PLC Setup to RUN mode. When RUN mode is specified as the startup mode in the PLC Setup, the CPU Unit will start operating in RUN mode when power is turned ON.

3. Check that the Power Supply Unit's POWER indicator is lit.



4. Check that the CPU Unit's MS indicator is lit green and the ERR/ALM indicator is not lit.

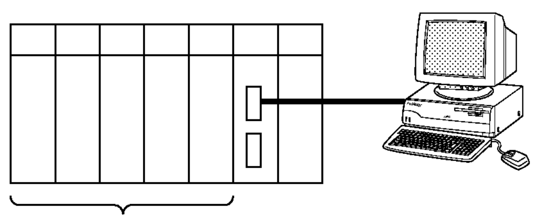
5. Registering the I/O Tables

Registering the I/O tables allocates I/O memory to the Units actually installed in the PLC. This operation is required in NE1S-series PLCs.

Note The user program and parameter area data in NE1S CPU Units is backed up in the built-in flash memory. The BKUP indicator will light on the front of the CPU Unit when the backup operation is in progress. Do not turn OFF the power supply to the CPU Unit when the BKUP indicator is lit. The data will not be backed up if power is turned OFF.

Using the NE Programmer Online

Use the following procedure to register the I/O tables with the CX-Programmer that is connected to the PLC.

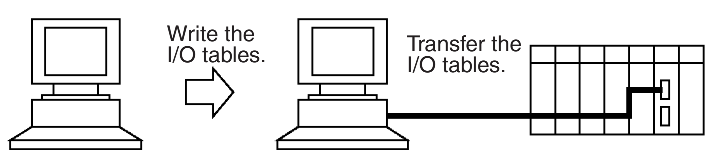


1,2,3...
1. Install all of the Units in the PLC.
2. Connect the host computer to the USB port or RS-232C port. (The power must be OFF.)

   Note If the host computer is being connected to the RS-232C port, pin 5 of the CPU Unit's DIP switch must be set to ON.
3. Select the port to be connected under ***Controller - Auto Upload - Select Port***. Connect online by selecting ***Controller - Auto Upload - Auto Upload from Controller***.
4. Select ***Controller - I/O Table - Create***. The type and location of Unit will be written to the CPU unit as registered I/O tables.

Using the NE Programmer Offline

Use the following procedure to create the I/O tables offline with the NE Programmer and later transfer the I/O tables from to the CPU Unit.



1,2,3...
1. Create a new project by selecting ***File - New - Project***.
2. Create a new configuration by selecting ***File - New - Configuration***.
3. With the configuration selected in the Project Window, select ***Controller - System Configuration***.

4. Click the I/O Table Tab and edit the I/O tables. After editing, click the **OK** Button and close the window.
5. Select ***Controller - Connect*** and connect the applicable PLC.
6. Select ***Controller - Download to Controller*** and transfer the I/O tables.

Note Detailed I/O table error information is stored in A261 whenever I/O tables cannot be created for any reason. This information can be used to facilitate troubleshooting if an error occurs.

6. PLC Setup Settings

These settings are the CPU Unit's software configuration. See *4-1 General Tab Page* for details on the settings.

Installing and Setting the Units

1,2,3...

1. Install the Unit.
2. With the CPU Unit power OFF, connect the NE Programmer to the RS-232C port or USB port.

   Note When connecting to the RS-232C port, turn ON pin 5 of the DIP switch on the front panel of the CPU Unit.
3. Connect the NE Programmer online:
   Select the port to be connected by selecting ***Controller - Auto Upload - Select Port***.
   Connect online by selecting ***Controller - Auto Upload - Auto Upload from Controller***.
4. Select ***Controller - System Configuration - General***. Change the settings and click the **OK** Button to save the changes.
5. Select ***Controller - Download to Controller***, check the system setup, and download the settings.

The PLC Setup will be written to the CPU Unit.

Transferring Settings to the CPU Unit Online without the PLC

1,2,3...

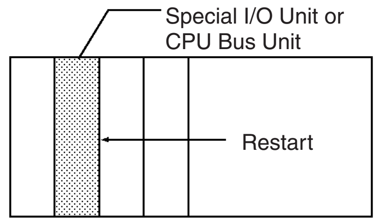
1. Create a new project by selecting ***File - New - Project***.
2. Create a new configuration by selecting ***File - New - Configuration***.
3. With the configuration in the Project Window selected, select ***Controller - System Configuration***.
4. Click the **General** Tab and edit the PLC Setup.
5. Select ***Controller - Connect*** and connect the applicable CPU Unit.
6. Select ***Controller - Download from Controller***, and transfer the system settings.

7. DM Area Initial Settings

The following table shows the parts of the DM Area that are allocated to Special I/O Units and CPU Bus Units for initial settings. The actual settings depend on the model of Unit being used.

Unit/Board	Allocated words
Special I/O Units	D20000 to D29599 (100 words × 96 Units)
CPU Bus Units	D30000 to D31599 (100 words × 16 Units)

After writing the initial settings to the DM Area, be sure to restart the Units by turning the PLC OFF and then ON again or toggling the Restart Bits for the affected Units.



8. Writing the Program

Write the program with the NE Programmer.

The NE1S-series PLC's program can be divided into independently executable tasks. Function blocks can also be used to create easily reusable programs. A single cyclic task can be written for program execution like earlier PLCs or several cyclic tasks can be written for a more flexible and efficient program.

Programming Device	Relationship between Tasks and Program	Writing a new program		Editing an existing program	
		Cyclic tasks	Interrupt tasks	Cyclic tasks	Interrupt tasks
NE Programmer	Specify the type of task and task number for each program.	All can be written. (Cyclic tasks 0 to 127)	All can be written. (Interrupt tasks 0 to 255)	All can be edited.	All can be edited.

9. Transferring the Program

The program that has been created must be transferred to the PLC's CPU Unit.

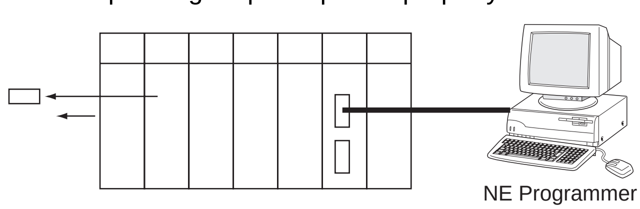
10. Testing Operation

■ Checking I/O Wiring

Before performing a Trial Operation in MONITOR mode, check the I/O wiring.

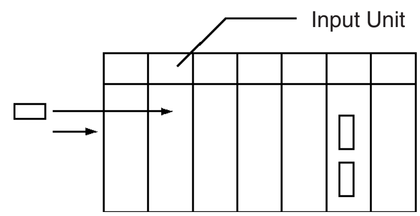
Check Output Wiring

With the PLC in PROGRAM mode, force-set and force-reset output bits and verify that the corresponding outputs operate properly.



Check Input Wiring

Activate input devices such as sensors and switches and verify that the corresponding indicators on the Input Units light. Also, use the Bit/Word Monitor operation in the Programming Device to verify the operation of the corresponding input bits.

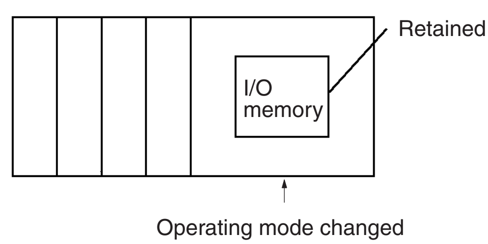


■ Auxiliary Area Settings

Make any required Auxiliary Area settings, such as the ones shown below. These settings can be made from a Programming Device or instructions in the program.

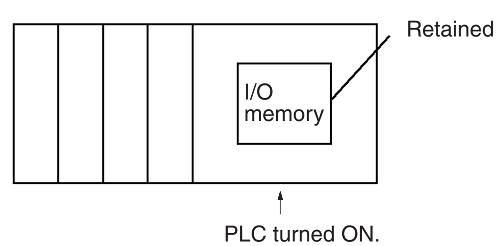
IOM Hold Bit (A50012)

Turning ON the IOM Hold Bit protects the contents of I/O memory (the CIO Area, Work Area, Timer Completion Flags and PVs, Index Registers, and Data Registers) that would otherwise be cleared when the operating mode is switched from PROGRAM mode to RUN/MONITOR mode or vice-versa.



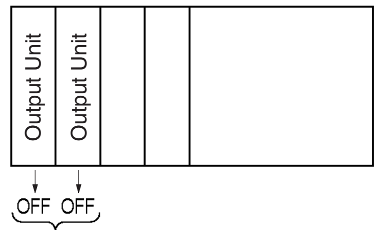
IOM Hold Bit Status at Startup

When the IOM Hold Bit is ON, the PLC Setup is set to protect the status of the IOM Hold Bit at startup, and the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to 1 (ON), the contents of I/O memory that would otherwise be cleared will be retained when the PLC is turned ON.



Output OFF Bit (A50015)

Turning ON the Output OFF Bit causes all outputs on Basic I/O Units and Special I/O Units to be turned OFF. The outputs will be turned OFF regardless of the PLC's operating mode.

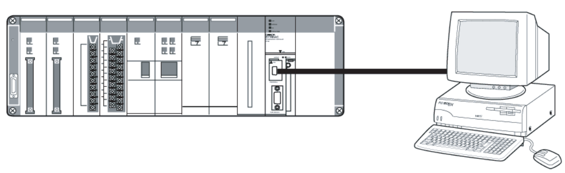


■ Trial Operation

Use the NE Programmer to switch the CPU Unit to MONITOR mode.

Using NE Programmer

The PLC can be put into MONITOR mode with a host computer running CX-Programmer.



NE Programmer

Trial Operation

Select ***Controller - Operating Mode - Monitor***.

Actual operation

Select ***Controller - Operating Mode - Run***.

■ Monitoring and Debugging

There are several ways to monitor and debug PLC operation, including the force-set and force-reset operations, differentiation monitoring, monitoring, data tracing, and online editing.

Force-Set and Force-Reset

When necessary, the force-set and force-reset operations can be used to force the status of bits and check program execution. Use the following procedure from the NE Programmer.

1,2,3...
1. Click the bit to be force-set or force-reset.
2. Select ***Set - Force On*** or ***Force Off*** from the Ladder menu.

Differentiation Monitor

The differentiation monitor operation can be used to monitor the up or down differentiation of particular bits. Use the following procedure from the NE Programmer.

1,2,3...
1. Click the bit for differential monitoring.
2. Click ***Differential Monitor*** from the Ladder Menu.
3. Click ***Rising*** or ***Falling.***
4. Click the **Start** Button.

Data Tracing

The NE Programmer's data trace operation can be used to check and debug program execution.

Online Editing

When a few lines of the program in the CPU Unit have to be modified, they can be edited online with the PLC in MONITOR mode or PROGRAM mode.

When more extensive modifications are needed, upload the program from the CPU Unit to the host computer, make the necessary changes, and transfer the edited program back to the CPU Unit.

11. Save and Print the Program

To save the program, select ***File*** and then ***Save*** (or ***Save As***) from the NE Programmer menus.

To print the program, select ***File*** and then ***Print*** from the NE Programmer menus***.***

12. Run the Program

Use the following procedure from the NE Programmer to switch the CPU Unit's operating mode to RUN mode.

1. Select ***Controller - Operating Mode - Run***.

SECTION 7
Installation and Wiring

This section describes how to install a PLC System, including mounting the various Units and wiring the System. Be sure to follow the instructions carefully. Improper installation can cause the PLC to malfunction, resulting in very dangerous situations.

7-1 Fail-safe Circuits

Be sure to set up safety circuits outside of the PLC to prevent dangerous conditions in the event of errors in the PLC or external power supply.

Supply Power to the PLC before Outputs

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

Managing PLC Errors

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

- Operation of the Power Supply Unit's overcurrent protection circuit
- A CPU error (watchdog timer error) or CPU on standby
- A fatal error* (memory error, I/O bus error, duplicate number error, Inner Board stopped error, too many I/O points error, program error, cycle time too long error, or FALS(007) error)

Be sure to add any circuits necessary outside of the PLC to ensure the safety of the system in the event of an error that stops PLC operation.

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

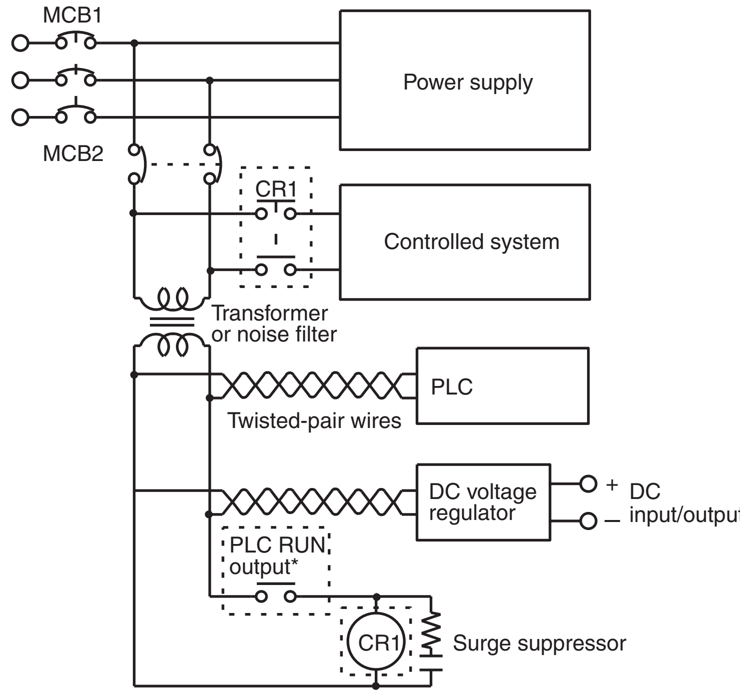
Managing Output Malfunctions

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

Emergency Stop Circuit

The following emergency stop circuit controls the power supply to the controlled system so that power is supplied to the controlled system only when the PLC is operating and the RUN output is ON.

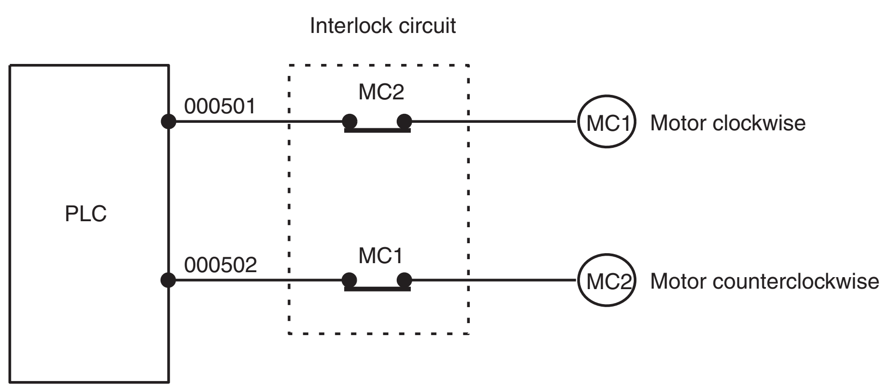
An external relay (CR1) is connected to the RUN output from the Power Supply Unit as shown in the following diagram.



Note *This configuration is possible with the C200HW-PA204R and C200HW-PA209R Power Supply Units only. When a Power Supply Unit without a RUN output is used, program the Always ON Flag (A1) as the execution condition for an output point from an Output Unit.

Interlock Circuits

When the PLC controls an operation such as the clockwise and counterclockwise operation of a motor, provide an external interlock such as the one shown below to prevent both the forward and reverse outputs from turning ON at the same time.



This circuit prevents outputs MC1 and MC2 from both being ON at the same time even if both CIO 000500 and CIO 000501 are both ON, so the motor is protected even if the PLC is programmed improperly or malfunctions.

7-2 Installation

7-2-1 Installation and Wiring Precautions

Be sure to consider the following factors when installing and wiring the PLC to improve the reliability of the system and make the most of the PLC's functions.

Ambient Conditions

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

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

Be sure to enclose or protect the PLC sufficiently in the following locations.

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

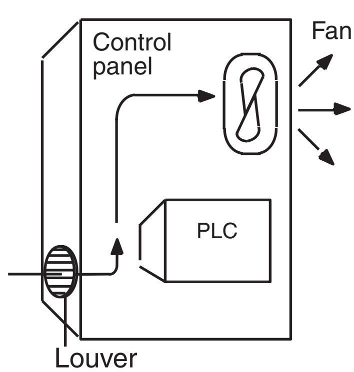
Installation in Cabinets or Control Panels

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

Temperature Control

The ambient temperature within the enclosure must be within the operating range of 0°C to 55°C. When necessary, take the following steps to maintain the proper temperature.

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

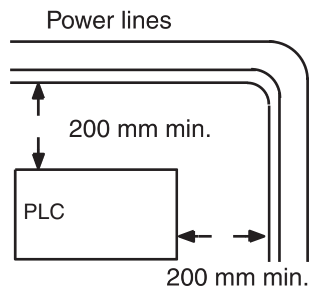


Accessibility for Operation and Maintenance

- To ensure safe access for operation and maintenance, separate the PLC as much as possible from high-voltage equipment and moving machinery.
- The PLC will be easiest to install and operate if it is mounted at a height of about 1.3 m (4 feet).

Improving Noise Resistance

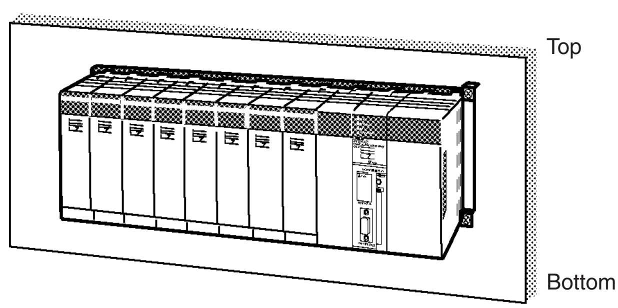
- Do not mount the PLC in a control panel containing high-voltage equipment.
- Install the PLC at least 200 mm (6.5 feet) from power lines.



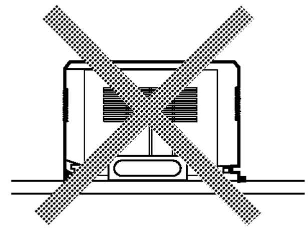
- Ground the mounting plate between the PLC and the mounting surface.
- When I/O Connecting Cables are 10 m or longer, connect the control panels in which Racks are mounted with heavier power wires (3 wires at least 2 mm^2 in cross-sectional area).

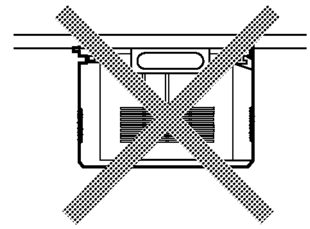
PLC Orientation

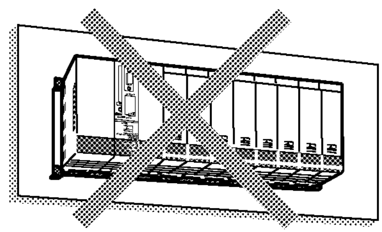
- Each Rack must be mounted in an upright position to provide proper cooling.

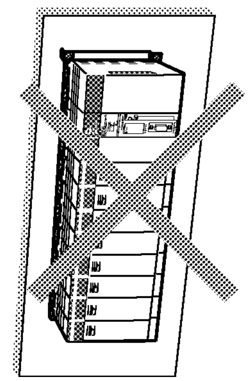


- Do not install a Rack in any of the following positions.





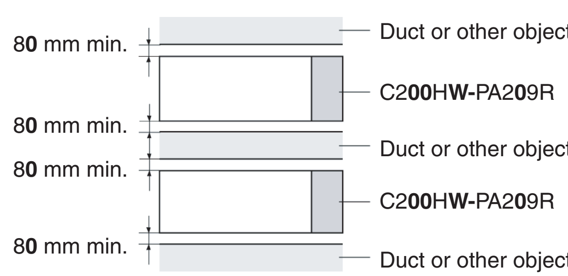




7-2-2 Installation in a Control Panel

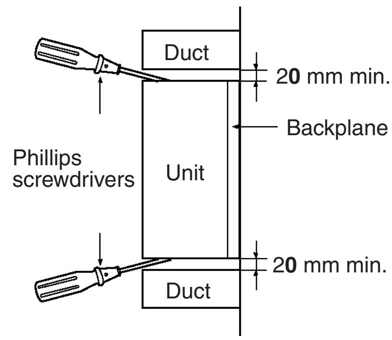
- A typical installation is a CPU Rack mounted above an Expansion Rack on a mounting plate in the control panel.
- The spacing between the CPU Rack and Expansion Rack (or between two Expansion Racks) should be sufficient to allow space for a wiring duct, wiring, air circulation, and replacement of Units in the Racks.

Note If the C200HW-PA209R Power Supply Unit is to be used at an ambient temperature of 50 °C or higher, improve ventilation by providing a minimum space of 80 mm between the top of the Unit and any other objects, e.g., ceiling, wiring ducts, structural supports, devices, etc.



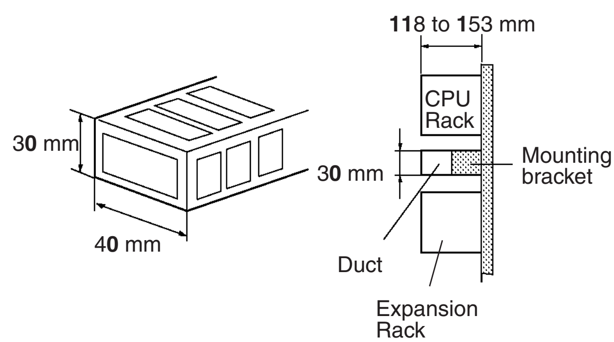
- Up to 7 Expansion Racks can be connected.
  Each I/O Connecting Cable can be up to 12 m long, but the sum total of all cables between the CPU Rack and Expansion Racks must be 12 m or less.
- The mounting plate should be grounded completely and we recommend using a mounting plate that has been plated with a good conductor to improve noise resistance.

- If all of the Racks cannot be mounted to the same mounting plate, the individual plates should be securely connected together using 3 wires of at least 2 mm^2 in cross-sectional area.
- The Backplanes are mounted to the plate(s) with four M4 screws each.
- Whenever possible, route I/O wiring through wiring ducts or raceways. Install the duct so that it is easy to fish wire from the I/O Units through the duct. It is handy to have the duct at the same height as the Racks.



Wiring Ducts

The following example shows the proper installation of wiring duct.



Note Tighten the Unit mounting screws, PLC Rack mounting screws, terminal block screws, and cable screws to the following torques.

Unit Mounting Screws

CPU Unit:	0.9 N·m
Power Supply Unit:	0.9 N·m
I/O Units:	0.4 N·m

Backplane Mounting Screws: 0.9 N·m

Terminal Screws

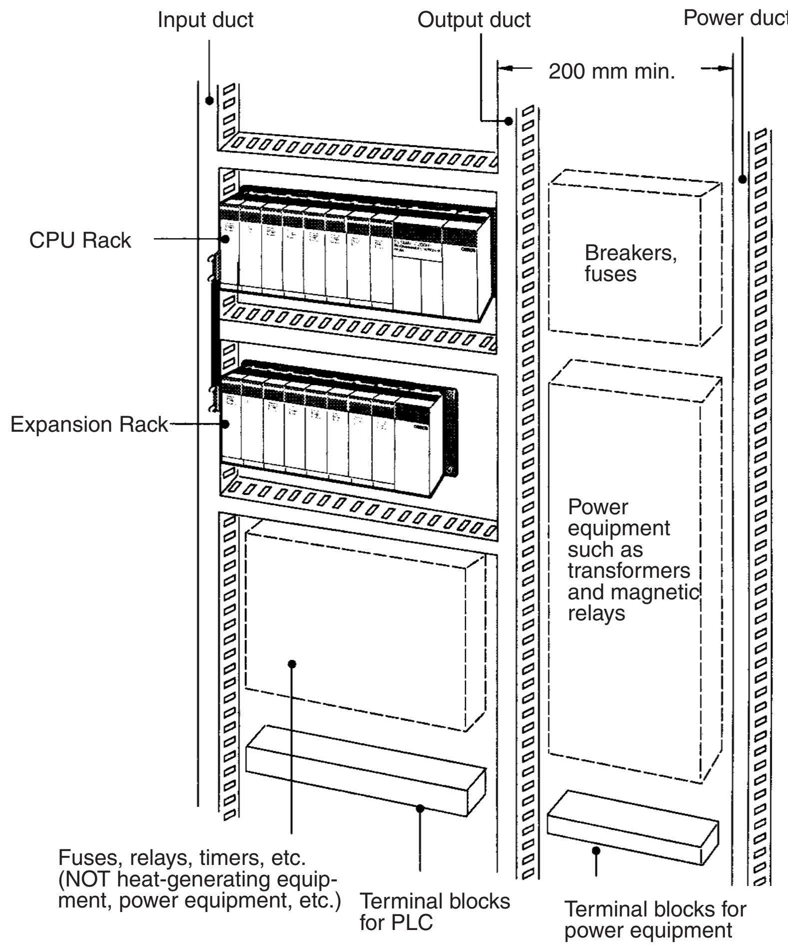
M3.5:	0.8 N·m
M3:	0.5 N·m

Cable Connector Screws

M2.6:	0.2 N·m

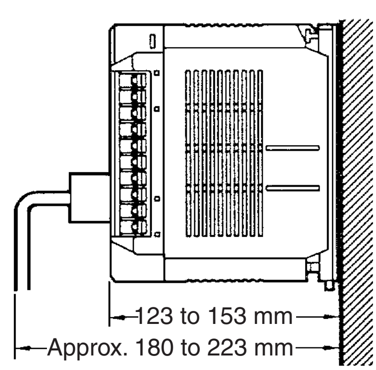
Routing Wiring Ducts

Install the wiring ducts at least 20 mm between the tops of the Racks and any other objects, (e.g., ceiling, wiring ducts, structural supports, devices, etc.) to provide enough space for air circulation and replacement of Units. If the C200HW-PA209R Power Supply Unit is to be used at an ambient temperature of 50 °C or higher, provide a minimum space of 80 mm.



7-2-3 Mounting Height

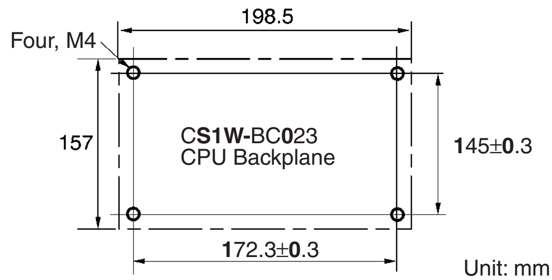
The mounting height of the CPU Rack or Expansion Racks is 123 to 153 mm, depending on I/O Units mounted. If Programming Devices or connecting cables are attached, the additional dimensions must be taken into account. Allow sufficient clearance in the control panel in which the PLC is mounted.



7-2-4 Mounting Dimensions

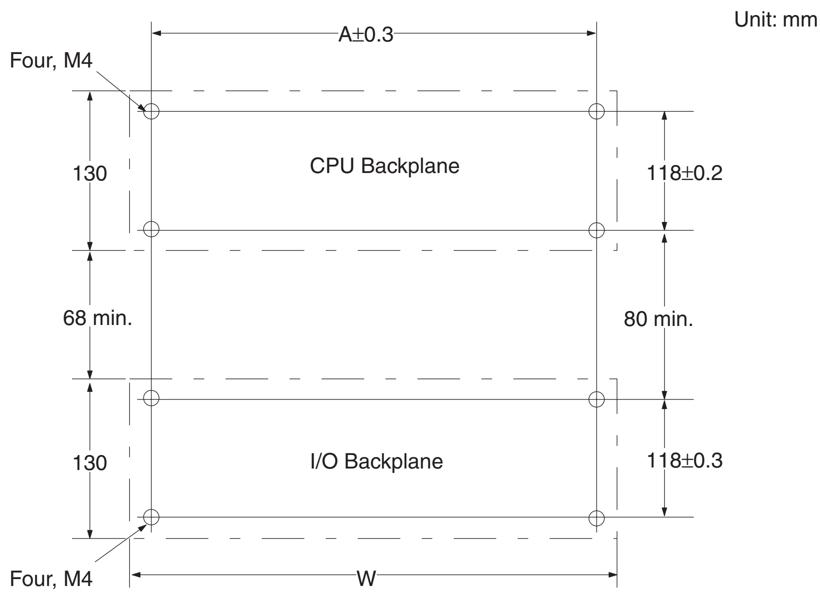
Backplanes

CPU Backplane with 2 Slots



Note Expansion Backplanes cannot be connected to 2-slot CPU Backplanes.

CPU Backplane with 3, 5, 8, or 10 Slots



Backplane	Model	A	W
CPU Backplanes	CS1W-BC032	246 mm	260 mm
	CS1W-BC052	316 mm	330 mm
	CS1W-BC082	421 mm	435 mm
	CS1W-BC102	491 mm	505 mm
Expansion Backplanes	CS1W-BI032	246 mm	260 mm
	CS1W-BI052	316 mm	330 mm
	CS1W-BI082	421 mm	435 mm
	CS1W-BI102	491 mm	505 mm

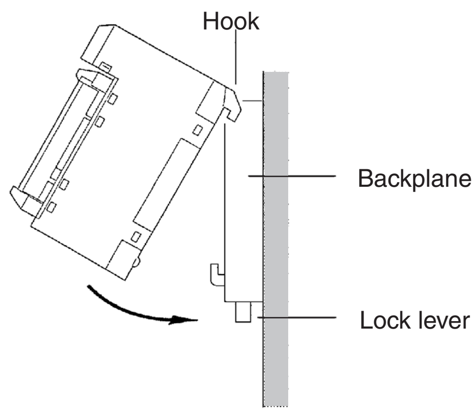
7-2-5 Mounting Units to the Backplane

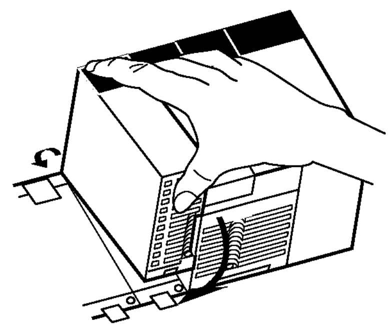
There are two methods for mounting or removing Units in the Backplane. The following table shows which method to use for each type of Unit.

Unit type	Installation method	Removal method
CPU Units, Power Supply Units, Basic I/O Units, Special I/O Units, and CPU Bus Units,	Hook the top of the Unit into the slot on the Backplane and tighten the screw on the bottom of Unit.	Loosen the screw on the bottom of the Unit and rotate the Unit upward.

1,2,3...

1. Mount the Unit to the Backplane by hooking the top of the Unit into the slot on the Backplane and rotating the I/O Unit downwards.

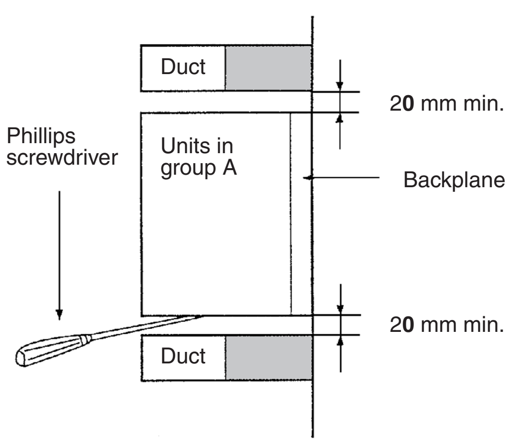


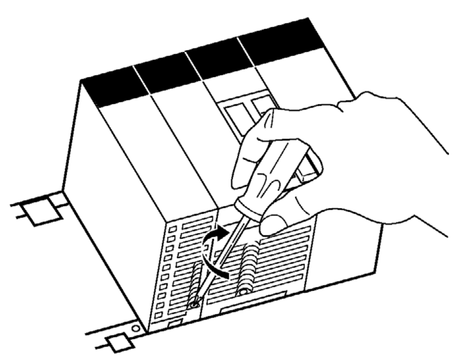


2. Make sure that the connector on the back of the Unit is properly inserted into the connector in the Backplane.
3. Use a Phillips-head screwdriver to tighten the screw on the bottom of Unit. The screwdriver must be held at a slight angle, so be sure to leave enough space below each Rack.

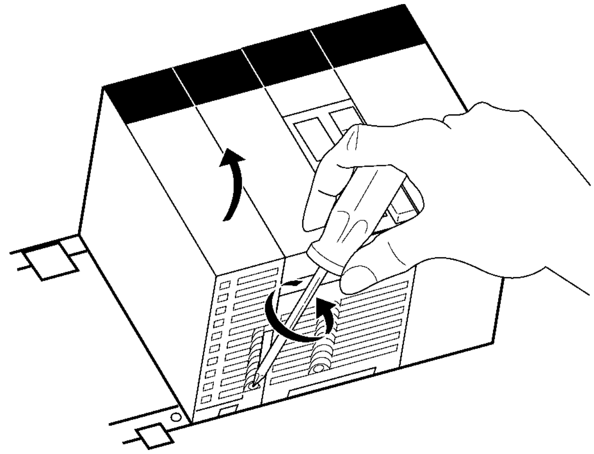
Note The screws at the bottoms of the Units must be tightened to the following torques.

CPU Unit:	0.9 N·m
Power Supply Unit:	0.9 N·m
I/O Units:	0.4 N·m





4. To remove a Unit, use a phillips-head screwdriver to loosen the screw at the bottom of the Unit, rotate the Unit upward, and remove it.



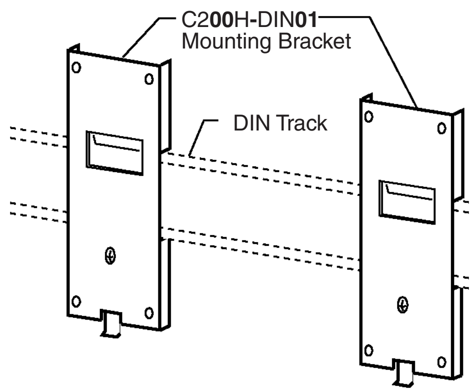
7-2-6 DIN Track Mounting

Do not use DIN Track to mount Backplane in locations subject to vibration; use locking screws to attach the Backplane directly.

Mount the DIN Track in the control panel with M4 screws in at least three places. Tighten the mounting screws to a torque of 1.2 N·m.

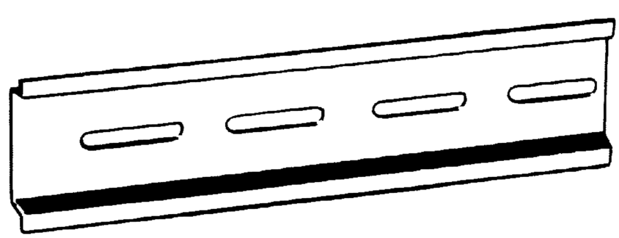
DIN Track Mounting Bracket

Use DIN Track Mounting Brackets to mount Racks to the DIN Track.



DIN Track

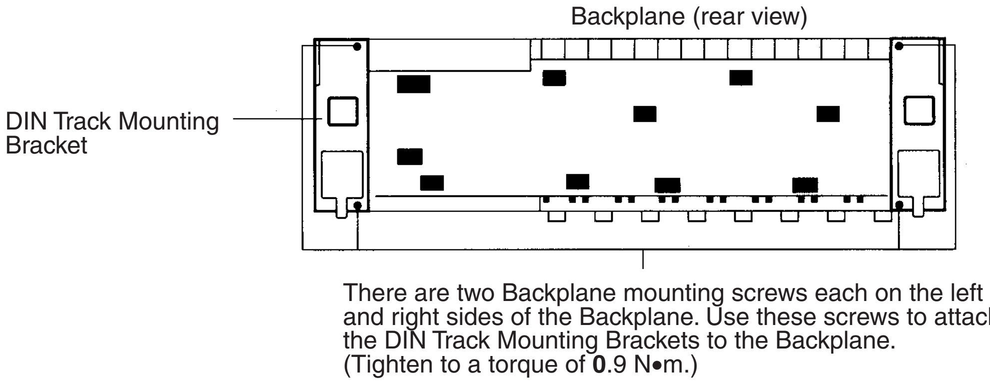
The following DIN Tracks are available.



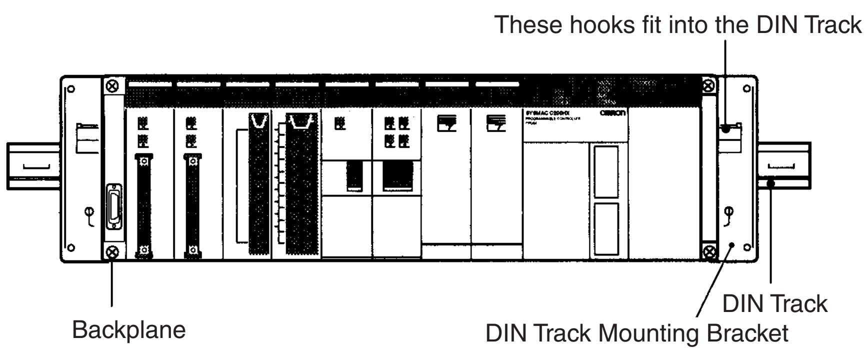
Model	Specification
PFP-50N	50 cm long, 7.3 mm high
PFP-100N	1 m long, 7.3 mm high
PFP-100N2	1 m long, 16 mm high

DIN Track Installation

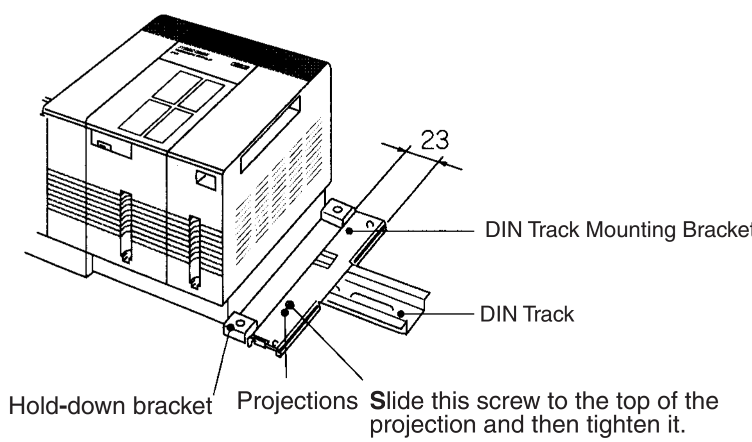
1,2,3... 1. Attach Mounting Brackets to each side (left and right) of the Backplane as shown below.



2. Mount the Backplane to the DIN Track so that the hooks on the Mounting Brackets fit into the upper portion of the DIN Track as shown below.



3. Loosen the hold-down bracket's screws and slide the Backplane upward so that the Mounting Bracket and Backplane clamp securely onto the DIN Track. Tighten the screws to a torque of 0.5 N·m.



7-2-7 I/O Connecting Cables

I/O Connecting Cables are used to connect the CPU Rack and Expansion Racks. There are four types of I/O Connecting Cables.

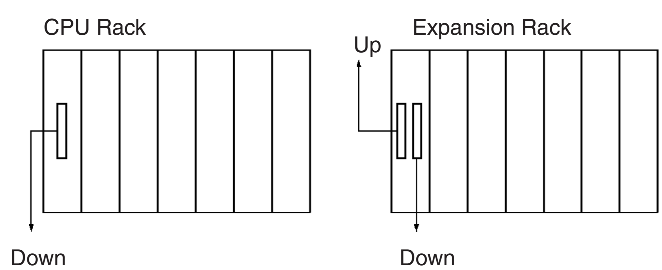
Type	Model number	Connectors		Usage
		CPU Rack end	Expansion Rack end	
I/O Connecting Cables	CS1W-CN□□3	Simple lock connector	Simple lock connector	CPU Rack → Expansion Rack
				Expansion Rack → Expansion Rack

Available Models

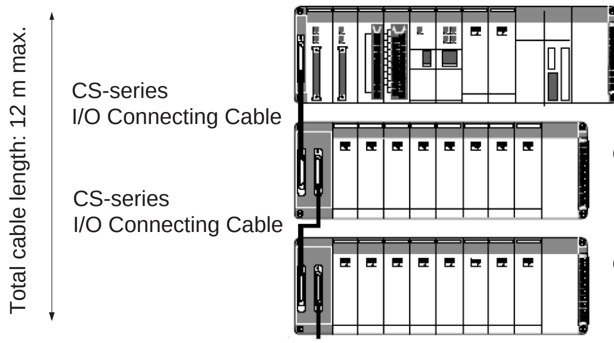
I/O Connecting Cables

Model number	Cable length
CS1W-CN313	0.3 m (See note.)
CS1W-CN713	0.7 m (See note.)
CS1W-CN223	2 m
CS1W-CN323	3 m
CS1W-CN523	5 m
CS1W-CN133	10 m
CS1W-CN133B2	12 m

- Install the Racks and select I/O Connecting Cables so that the total length of all I/O Connecting Cables does not exceed 12 m.
- The following diagram shows where each I/O Connecting Cable must be connected on each Rack. The Rack will not operate if the cables aren't connected properly. (The "up" direction is towards the CPU Unit and "down" is away from the CPU Unit.)



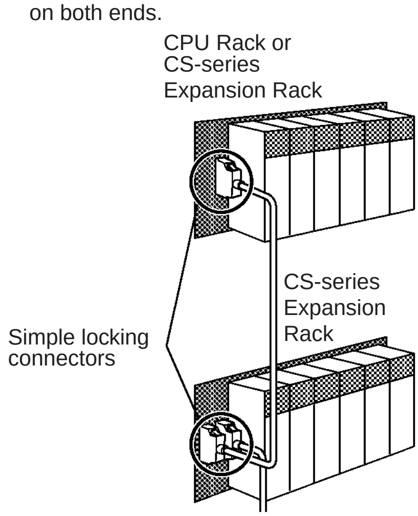
- The following diagram shows an example of proper Rack connections.



Cable Connections

Simple locking connectors are used for the CPU and Expansion Backplanes.

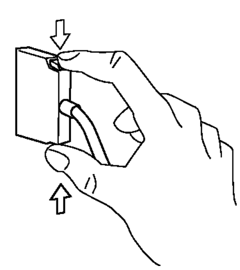
The connectors can be inserted only one way; they cannot be inserted upside down. Be sure that the connectors fit properly as they are inserted.



Connecting the Simple Locking Connectors

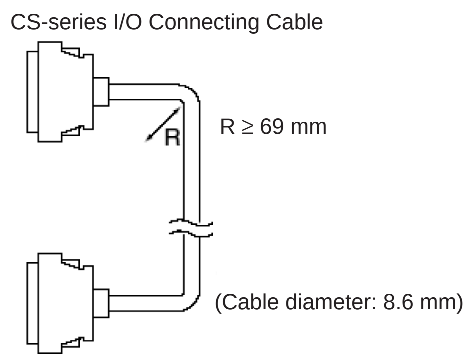
Press the tabs on the end of the connector and insert the connector until it locks in place. The PLC will not operate properly if the connector isn't inserted completely. Press in on both sides of the connector to remove it.

Note When using an I/O Connecting Cable with a locking connector, be sure that the connector is firmly locked in place before using it.



- Do not route the I/O Connecting Cables through ducts that contain the I/O or power wiring.
- Always turn OFF the power supply to the PLC before connecting Cables.

- An I/O bus error will occur and the PLC will stop if an I/O Connecting Cable's connector separates from the Rack. Be sure that the connectors are secure.
- Do not cut or reconnect I/O Connecting Cables. Handle them with caution when mounting the PLC or when using wiring ducts.
- Do not pulse on cables with excessive force or bend them paste their natural bending radius.
- The cables can withstand a pulling force of 49 N (11 lbs) max.
- The minimum bending radius of the I/O Connecting Cables is shown in the following diagram.

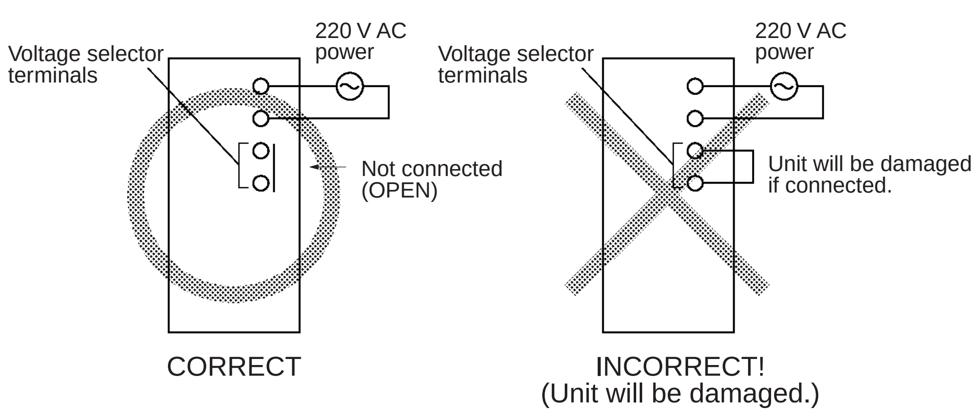


7-3 Wiring

7-3-1 Power Supply Wiring

AC Power Supply Units

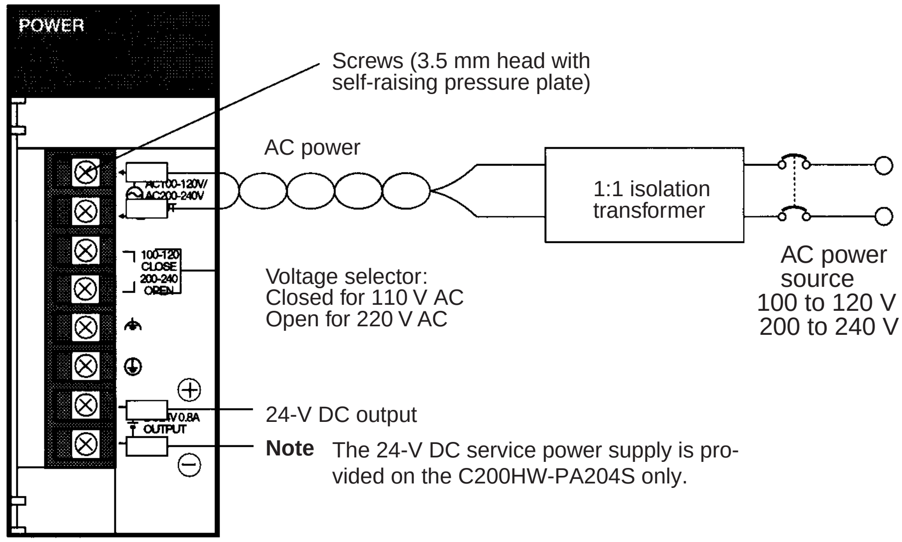
When 200 to 240 V AC power is being supplied, be sure to remove the jumper bar that shorts the voltage selector terminals. The Unit will be damaged if 200 to 240 V AC is supplied with the jumper bar connected.



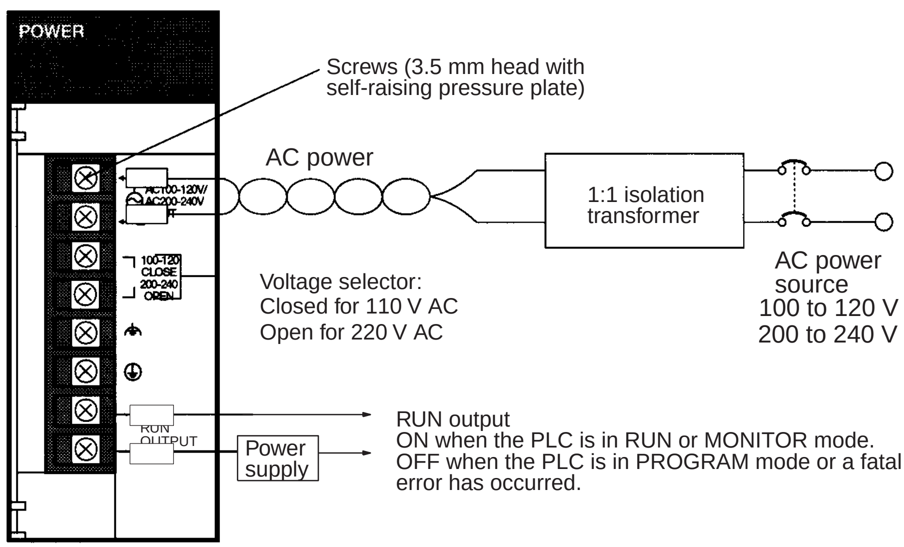
Note (1) If 110 V AC power is supplied but the jumper bar has been removed to select 220 V AC, the Unit will not operate because the power supply voltage will be below the 85% minimum level.

(2) Do not remove the protective label from the top of the Unit until wiring has been completed. This label prevents wire strands and other foreign matter from entering the Unit during wiring procedures.

C200HW-PA204 or C200HW-PA204S Power Supply Unit



C200HW-PA204R or C200HW-PA209R Power Supply Unit



Note If a Power Supply Unit without a RUN output is being used, an output that acts as a RUN output can be created by programming the Always ON Flag (A1) as the execution condition for an output from an Output Unit.

AC Power Source

- Supply 100 to 120 V AC or 200 to 240 V AC.
- Keep voltage fluctuations within the specified range:

Supply voltage	Allowable voltage fluctuations
100 to 120 V AC	85 to 132 V AC
200 to 240 V AC	170 to 264 V AC

- The terminal block indicator L2/N-L1 may appear as L1/N-L2 in some Units, but the function of the terminals is the same.
- If one power supply phase of the equipment is grounded, connect the grounded phase side to the L2/N (or L1/N if so indicated) terminal.

Voltage Selector

Shorted:100 to 120 V AC
Open: 200 to 240 V AC

Short-circuit the voltage selector terminals with the jumper bar to select 100 to 120 V AC supply voltage. For 200 to 240 V AC leave them open.

Note The Power Supply Unit will be damaged if 200 to 240 V AC power is supplied and the voltage selector terminals are connected with the jumper bar.

Isolation Transformer

The PLC's internal noise isolation circuits are sufficient to control typical noise in power supply lines, but noise between the PLC and ground can be significantly reduced by connecting a 1-to-1 isolation transformer. Do not ground the secondary coil of the transformer.

Power Consumption

The power consumption will be 120 VA max. per Rack, but there will be a surge current of at least 5 times the max. current when power is turned ON.

24-V DC Output (C200HW-PA204S Only)

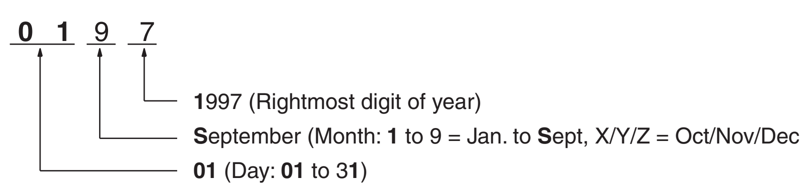
Use these terminals as the power supply for 24-V DC Input Units. Never externally short these terminals; PLC operation will stop if these terminals are shorted.

Although the 24-V DC output can supply up to 0.8 A, the combined power consumption for both 5 V DC and 24 V DC must be 30 W or less, i.e., the capacity of the 24-V DC output will be reduced if the Units mounted to the Rack consume a lot of current. Refer to *1-3-4 Unit Current Consumption* for the power consumption of each Unit.

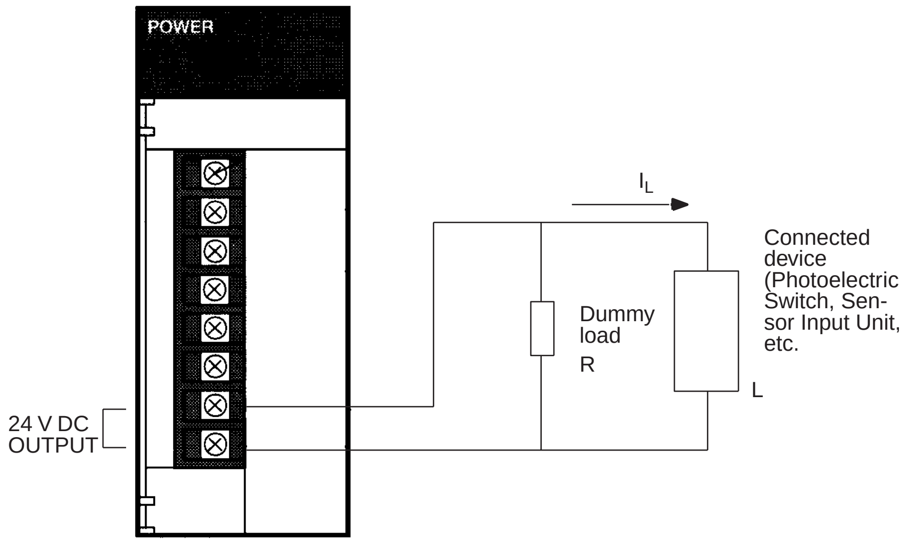
The output voltage of the 24-V DC output will vary with the current consumption of the load as shown in the following table. Be sure to check the current consumption and allowable voltage ranges of the devices connected before using these terminals.

Load current on 24-V DC output	Less than 0.3 A	0.3 A or higher
Accuracy of 24-V DC output for lot No. 0197 or later	+17% –11%	+10% –11%
Accuracy of 24-V DC output for lot No. 3187 or earlier	+10% –20%	

Note Lot numbers are as shown in the following diagram.



We recommend connecting a dummy load as shown in the following diagram if the maximum operating voltage of the connected device is 26.4 V (24 V +10%).



Resistance of the dummy load: 120 Ω when I_L = 0.1 A
240 Ω when I_L = 0.2 A
Not necessary when I_L = 0.3 A
(I_L: Total current of connected devices)

$$R = \frac{24}{0.3 - I_L}$$

Power rating of the dummy load (with a safety factor of 5):

$$W = (0.3 - I_L) \times 26.4 \times 5$$

30 W (120 Ω) when I_L = 0.1 A
15 W (240 Ω) when I_L = 0.2 A

Note Since the dummy load will generate heat, be careful not to allow any combustible materials to come in contact with the resistor.

RUN Output (C200HW-PA204R/209R)

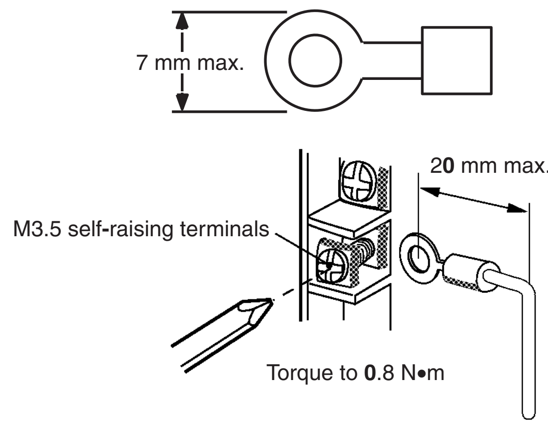
This output is ON whenever the CPU Unit is operating in RUN or MONITOR mode; it is OFF when the CPU Unit is in PROGRAM mode or a fatal error has occurred.

The RUN output can be used to control external systems, such as in an emergency stop circuit that turns off the power supply to external systems when the PLC is not operating. (See *7-1 Fail-safe Circuits* for more details on the emergency stop circuit.)

	C200HW-PA204R	**C200HW-PA209R**
Contact form	SPST-NO	SPST-NO
Maximum switching capacity	250 V AC:2 A for resistive loads 0.5 A for inductive loads 24 V DC:2 A	240 V AC:2 A for resistive loads 120 V AC:0.5 A for inductive loads 24 V DC:2 A for resistive loads 2 A for inductive loads

Crimp Terminals

The terminals on the Power Supply Unit are M3.5, self-raising terminals with screws. Use crimp terminals for wiring. Do not connect bare stranded wires directly to terminals. Tighten the terminal block screws to the torque of 0.8 N•m. Use round-type crimp terminals (M3.5) having the dimensions shown below.



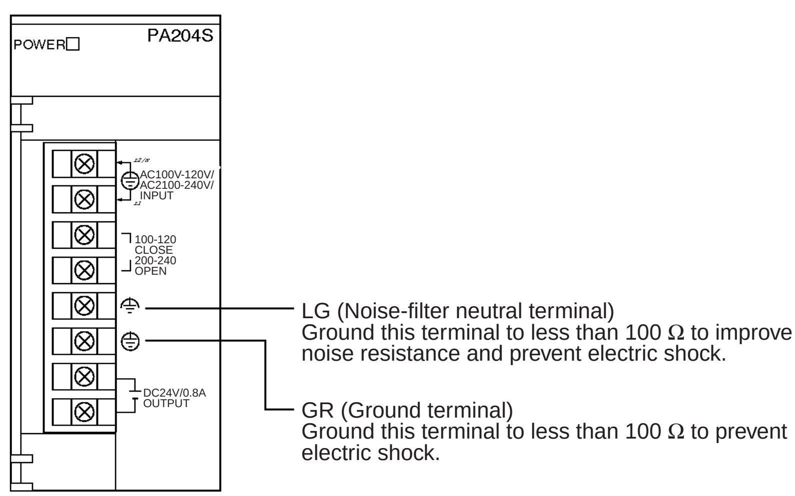
⚠ Caution Tighten the AC power supply terminal block screws to the torque of 0.8 N•m. Loose screws may result in short-circuit, malfunction, or fire.

Note

1. Supply power to all of the Power Supply Units from the same source.
2. Be sure to check the setting of the voltage selector before supplying power.
3. Do not forget to remove the label from the top of the Power Supply Unit after wiring the Unit. The label will block air circulation needed for cooling.

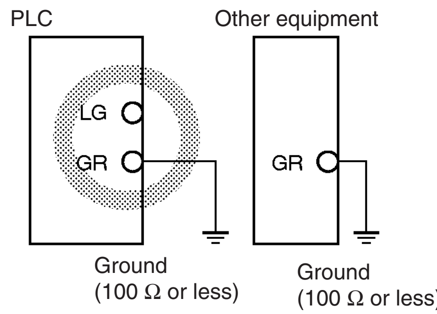
Grounding

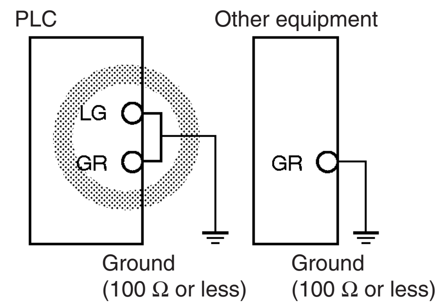
The diagram below shows the location of the ground and line ground terminals.



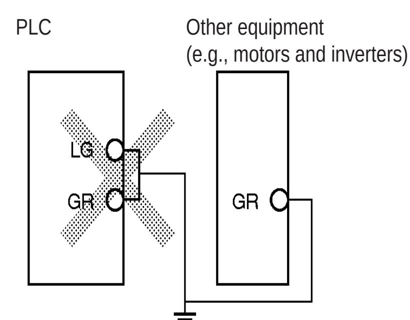
- To help prevent electrical shock, ground the ground terminal (GR: ⏚) with a ground resistance of less than 100 Ω using a 14-gauge wire (minimum cross-sectional area of 2 mm^2).
- The line ground terminal (LG: ⏚) is a noise-filtered neutral terminal. If noise is a significant source of errors or electrical shocks are a problem, connect the line ground terminal to the ground terminal and ground both with a ground resistance of less than 100 Ω.
- The ground wire should not be more than 20 m long.

- The following grounding configurations are acceptable.
- The CS-series Backplanes are designed to be mounted so that they are isolated (separated) from the mounting surface to protect them from the effects of noise in the installation environment (e.g., the control panel).





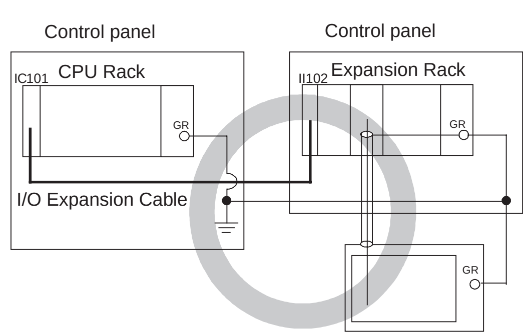
- Do not share the PLC's ground with other equipment, such as motors and inverters, or ground the PLC to the metal structure of a building. The configuration shown in the following diagram may worsen operation.



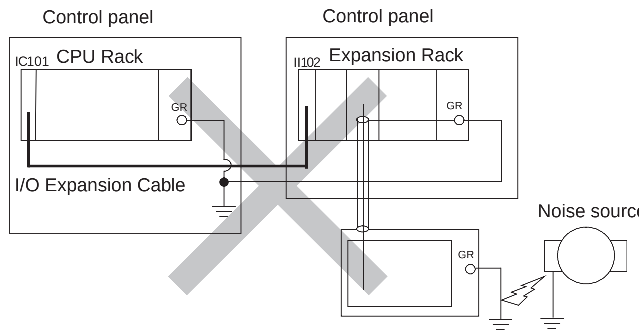
Wiring Communications Lines

When using communications from one or more Rack in the system, ground the entire system so that only one point is grounded. (Refer to user documentation for the devices connected.) For detailed connection methods, refer to the *Operation Manual* for the Communications Unit.

Recommended Wiring



Wiring Susceptible to Noise



Crimp Terminals

The terminals on the Power Supply Unit are M3.5, self-raising terminals with screws. Use crimp terminals for wiring. Do not connect bare stranded wires directly to terminals. Tighten the terminal block screws to the torque of 0.8 N• m. Use crimp terminals (M3.5) having the dimensions shown below.

7-3-2 Wiring Basic I/O Units with Terminal Blocks

I/O Unit Specifications

Double-check the specifications for the I/O Units. In particular, do not apply a voltage that exceeds the input voltage for Input Units or the maximum switching capacity for Output Units. Doing so may result in breakdown, damage, or fire.

When the power supply has positive and negative terminals, be sure to wire them correctly.

Electric Wires

The following wire gauges are recommended.

Terminal Block Connector	Wire Size
10-terminal	AWG 22 to 18 (0.32 to 0.82 mm^2)
19-terminal/20-terminal	AWG 22 (0.32 mm^2)

Note The current capacity of electric wire depends on factors such as the ambient temperature and insulation thickness as well as the gauge of the conductor.

Crimp Terminals

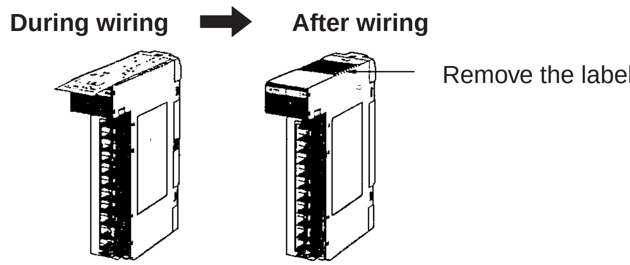
The terminals on the Power Supply Unit are M3.5, self-raising terminals with screws. Use crimp terminals for wiring. Do not connect bare stranded wires directly to terminals. Tighten the terminal block screws to the torque of 0.8 N•m. Use crimp terminals (M3.5) having the dimensions shown below.

10-terminal or 19-terminal Terminal Blocks

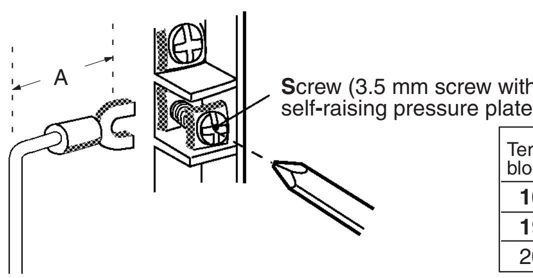
20-terminal Terminal Blocks

Wiring

Do not remove the protective label from the top of the Unit until wiring has been completed. This label prevents wire strands and other foreign matter from entering the Unit during wiring procedures. (Remove the label after wiring has been completed to allow air circulation needed for cooling.)



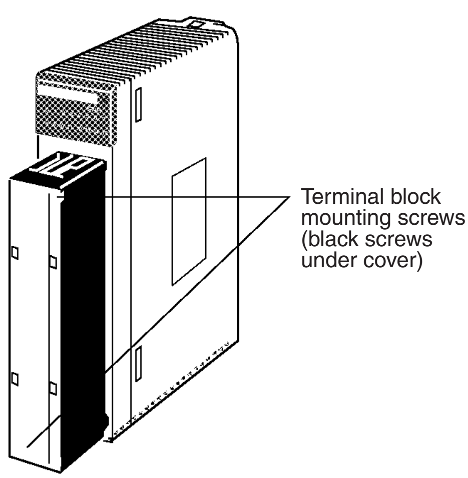
- Wire the Units so that they can be easily replaced. In addition, make sure that the I/O indicators are not covered by the wiring.
- Do not place the wiring for I/O Units in the same duct or raceway as power lines. Inductive noise can cause errors in operation.
- Tighten the terminal screws to the torque of 0.8 N·m.
- The terminals have screws with 3.5-mm diameter heads and self-raising pressure plates. Connect the lead wires to the terminals as shown below.



Terminal block	A
10P	23 mm
19P	14 mm
20P	18 mm

Terminal Blocks

The I/O Units are equipped with removable terminal blocks. The lead wires do not have to be removed from the terminal block to remove it from an I/O Unit. The terminal block can be removed by taking out the terminal block mounting screws.



Basic I/O Units (including Interrupt Input Units and High-speed Input Units)

7-3-3 Wiring Basic I/O Units with Connectors

This section describes wiring for the following Units:

- Basic I/O Units with Connectors (32-, 64-, and 96-point Units)

Basic I/O Units with Connectors use special connectors to connector to external I/O devices. The user can combine a special connector with cable or use a preassembled OMRON cable to connect a High-density I/O Unit to a terminal block or Relay Terminal. The available OMRON cables are described later in this section.

- Be sure not to apply a voltage that exceeds the input voltage for Input Units or the maximum switching capacity for Output Units.
- When the power supply has positive and negative terminals, be sure to wire them correctly. Loads connected to Output Units may malfunction if the polarity is reversed.
- Use reinforced insulation or double insulation on the DC power supply connected to DC I/O Units when required by EC Directives (low voltage).
- When connecting the connector to the I/O Unit, tighten the connector screws to a torque of 0.2 N·m.
- Turn ON the power only after checking the connector's wiring carefully.
- Do not pull the cable. Doing so will damage the cable.
- Bending the cable too sharply can damage or break wiring in the cable.

Available Connectors

Use the following connectors when assembling a connector and cable.

Basic 32- and 64-point I/O Units

The following connectors are recommended for attachment to Basic 32- and 64-point I/O Units.

Connection	Pins	OMRON set	Fujitsu parts
Solder-type (included with Unit)	40	C500-CE404	Socket: FCN-361J040-AU Connector bar: FCN-360C040-J2
Crimp-type	40	C500-CE405	Socket: FCN-363J040 Connector bar: FCN-360C040-J2 Contacts: FCN-363J-AU
Crimp-type	40	C500-CE403	FCN-367J040-AU

Note Solder-type connectors are included with each Unit.

Basic 96-point I/O Units

The following connectors are recommended for attachment to Basic 96-point I/O Units.

Connection	Pins	OMRON set	Fujitsu parts
Solder-type (included with Unit)	56	CS1W-CE561	Socket: FCN-361J056-AU Connector bar: FCN-360C056-J3
Crimp-type	56	CS1W-CE562	Socket: FCN-363J056 Connector bar: FCN-360C056-J3 Contacts: FCN-363J-AU
Crimp-type	56	CS1W-CE563	FCN-367J056-AU

Note Solder-type connectors are included with each Unit.

Wire

We recommend using cable with wire gauges of AWG 24 or AWG 26 (0.2 mm^2 to 0.08 mm^2). Use cable with external wire diameters of 1.61 mm max.

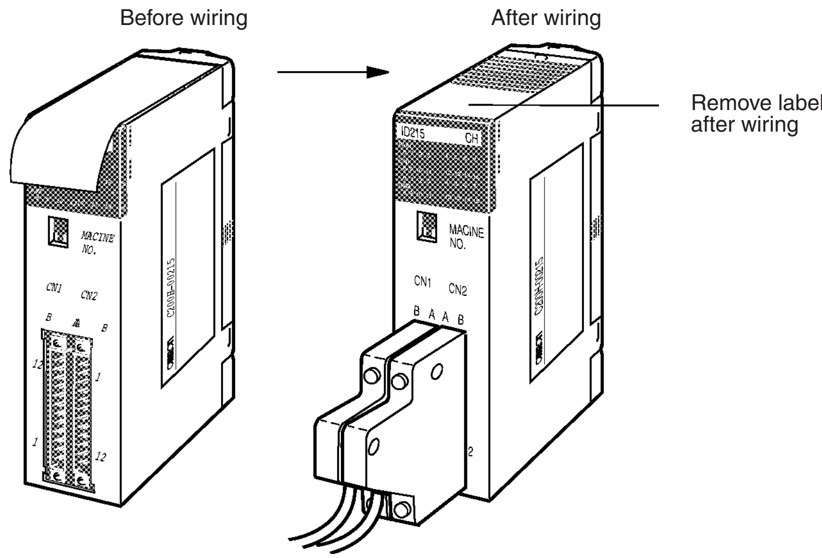
Wiring Procedure

1,2,3...

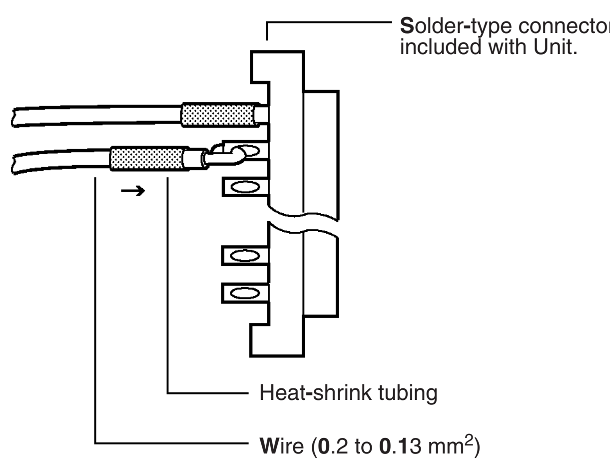
1. Check that each Unit is installed securely.

   Note Do not apply excessive force to the cables.

2. Do not remove the protective label from the top of the Unit until wiring has been completed. This label prevents wire strands and other foreign matter from entering the Unit during wiring. (Remove the label after wiring has been completed to allow air circulation needed for cooling.)

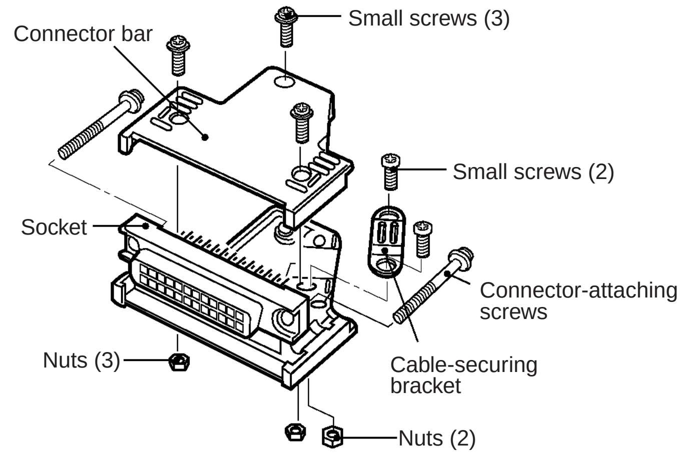


3. When solder-type connectors are being used, be sure not to accidentally short adjacent terminals. Cover the solder joint with heat-shrink tubing.

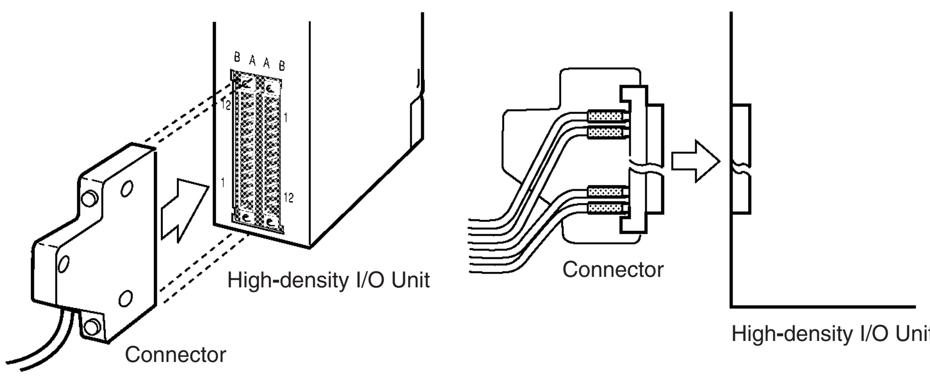


Note Double-check to make sure that the Output Unit's power supply leads haven't been reversed. If the leads are reversed, the Unit's internal fuse will blow and the Unit will not operate.

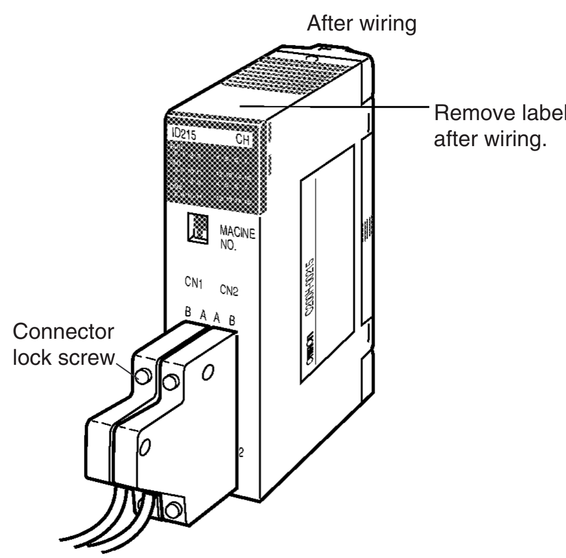
4. Assemble the connector (included or purchased separately) as shown in the following diagram.



5. Insert the wired connector.



6. Remove the protective label after wiring has been completed to allow air circulation needed for cooling.



Tighten the connector-attaching screws to a torque of 0.2 N·m.

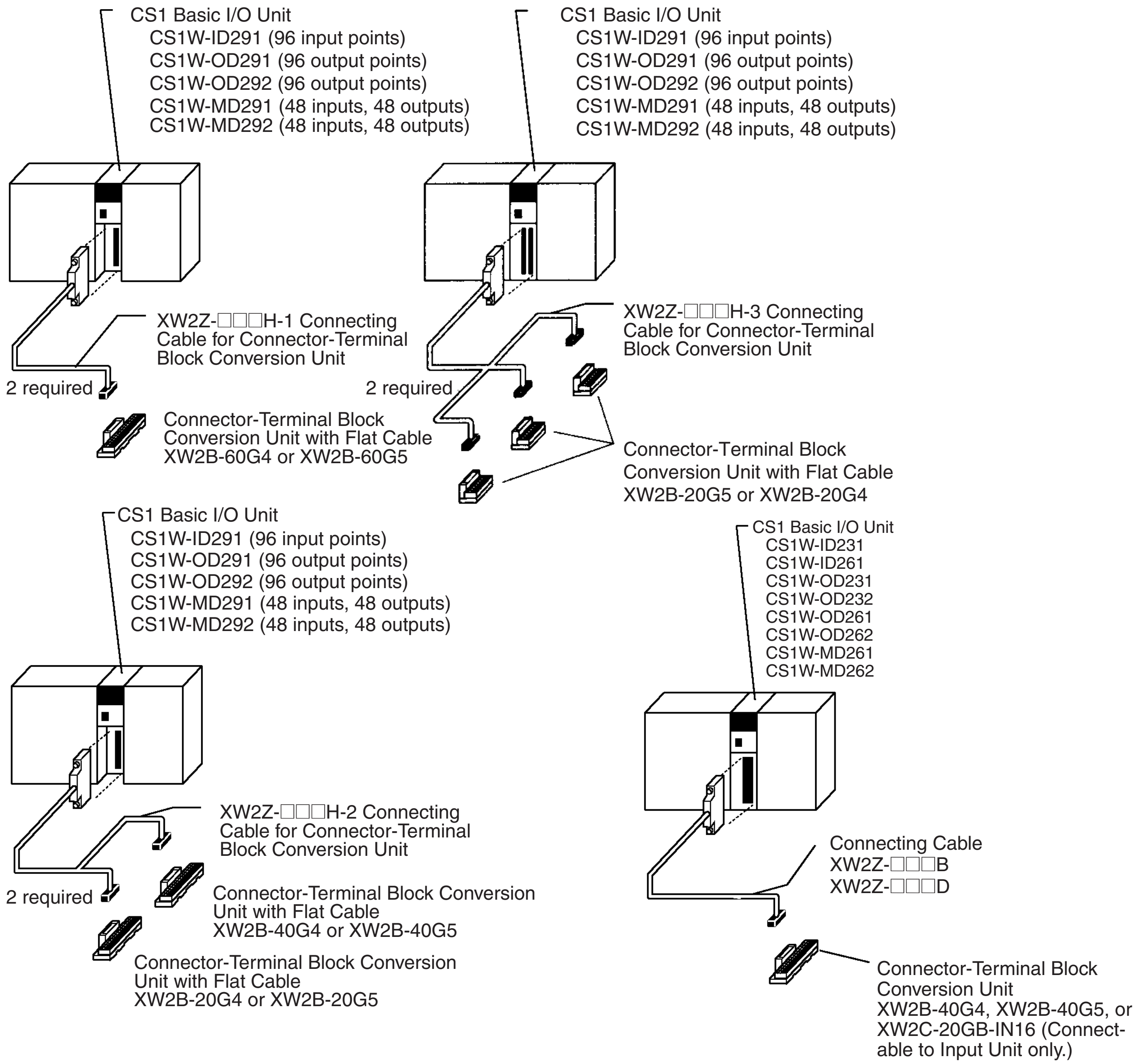
Preassembled Cables

The following examples show applications for preassembled OMRON Cables. Contact your OMRON dealer for more details.

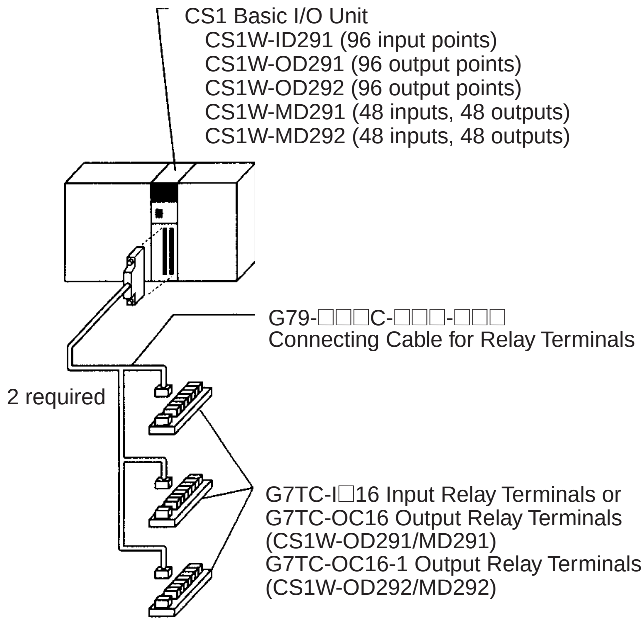
Basic I/O Units with Connectors

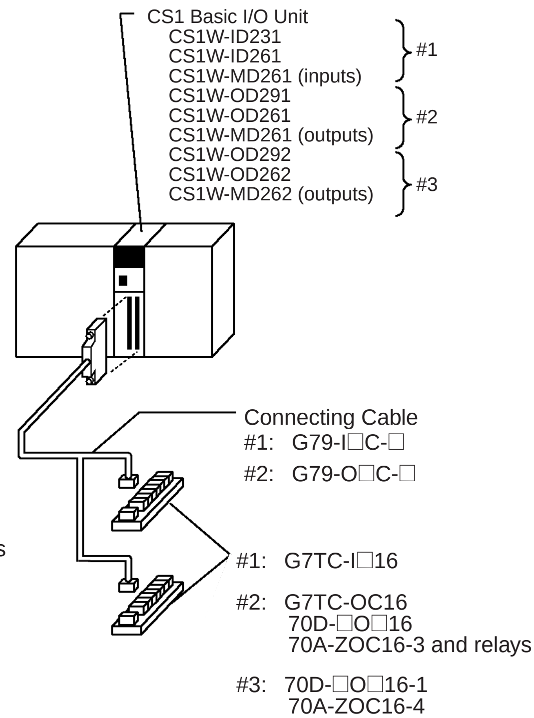
The following cables are compatible with High-density I/O Units.

1,2,3... 1. Connecting to a terminal block. (Two of the following Cables and Conversion Units are required.)



2. Connecting to a Relay Terminal. (Two of the following Cables and Relay Terminals are required.)





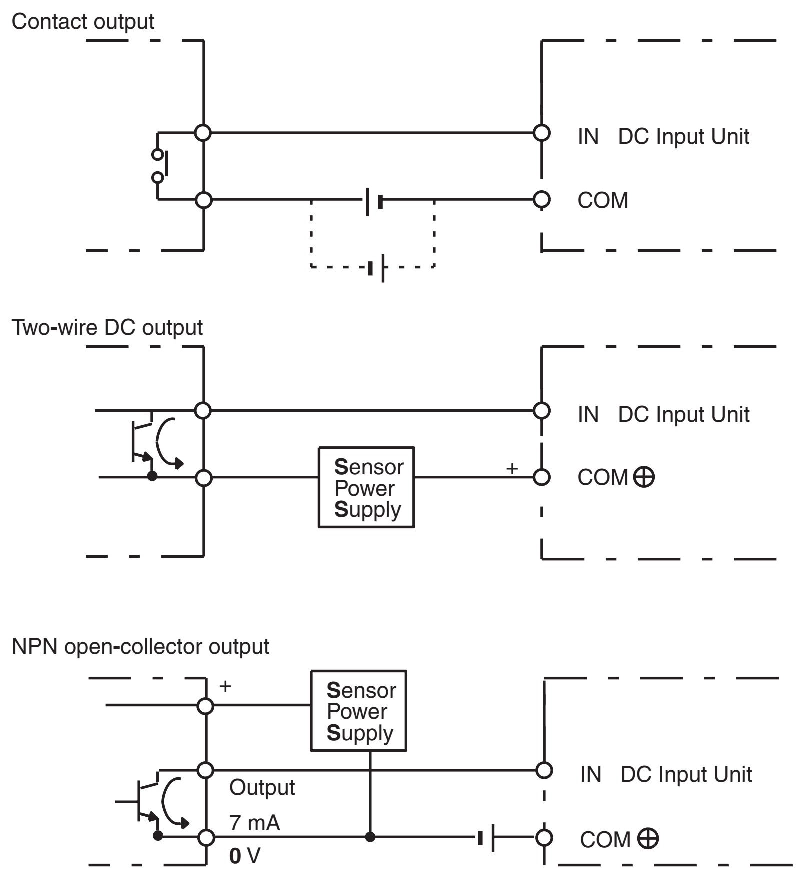
7-3-4 Connecting I/O Devices

Input Devices

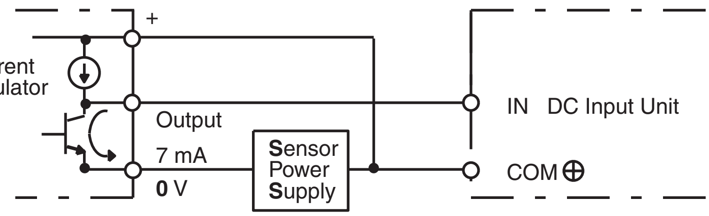
Use the following information for reference when selecting or connecting input devices.

DC Input Units

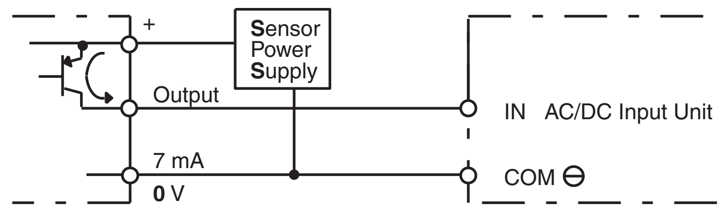
The following types of DC input devices can be connected.



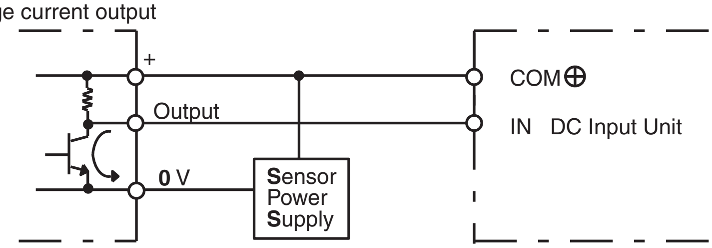
NPN current output



PNP current output

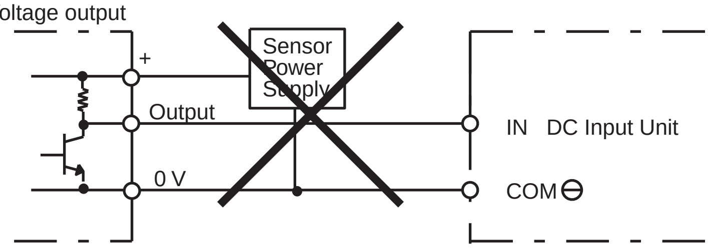


Voltage current output



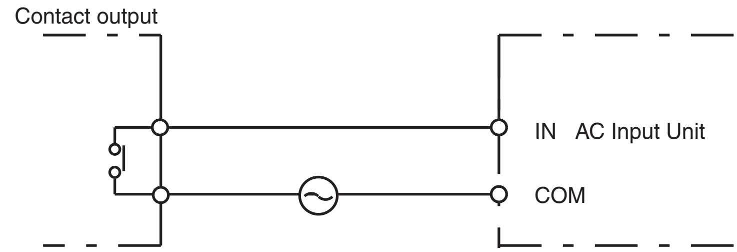
The circuit below should **NOT** be used for I/O devices having a voltage output.

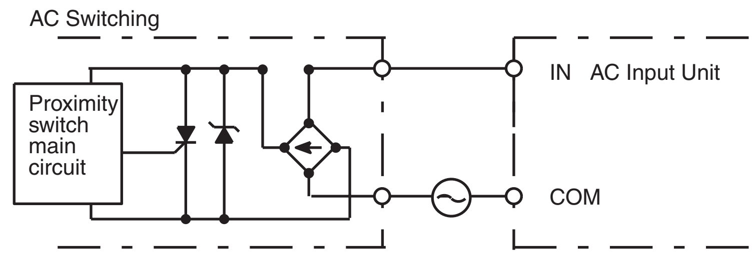
Voltage output



AC Input Units

The following types of AC input devices can be connected.





Note When using a reed switch as the input contact for an AC Input Unit, use a switch with an allowable current of 1 A or greater. If Reed switches with smaller allowable currents are used, the contacts may fuse due to surge currents.

Precautions when Connecting a Two-wire DC Sensor

When using a two-wire sensor with a 12-V DC or 24-V DC input device, check that the following conditions have been met. Failure to meet these conditions may result in operating errors.

1,2,3...

1. Relation between voltage when the PLC is ON and the sensor residual voltage:

   $V_{ON} \leq V_{CC} - V_R$

2. Relation between voltage when the PLC is ON and sensor control output (load current):

   I_{OUT} (min) $\leq I_{ON} \leq I_{OUT}$ (max.)
   $I_{ON} = (V_{CC} - V_R - 1.5$ [PLC internal residual voltage]$)/R_{IN}$

   When I_{ON} is smaller than I_{OUT} (min), connect a bleeder resistor R. The bleeder resistor constant can be calculated as follows:

   $R \leq (V_{CC} - V_R)/(I_{OUT}$ (min.) $- I_{ON})$

   Power $W \geq (V_{CC} - V_R)^2/R \times 4$ [allowable margin]

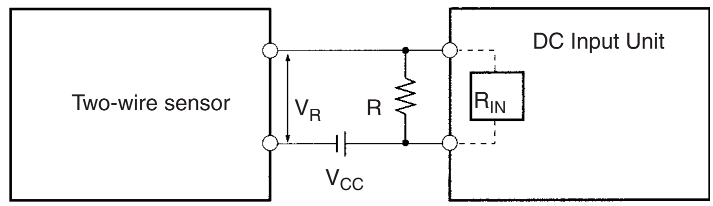
3. Relation between current when the PLC is OFF and sensor leakage current:

   $I_{OFF} \geq I_{leak}$

   If I_{leak} is larger than I_{OFF}, connect a breeder resistor. The breeder resistor constant can be calculated as follows:

   $R \leq R_{IN} \times V_{OFF}/(I_{leak} \times R_{IN} - V_{OFF})$

   Power $W \geq (V_{CC} - V_R)^2/R \times 4$ [allowable margin]



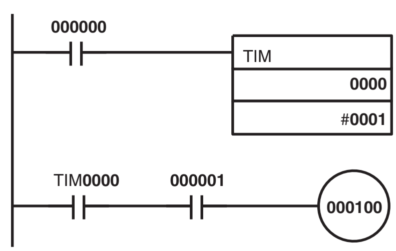
V_{CC}: Power voltage
V_{ON}: PLC ON voltage
I_{ON}: PLC ON current
I_{OFF}: PLC OFF current
R_{IN}: PLC input impedance

V_R: Sensor output residual current
I_{OUT}: Sensor control current (load current)
I_{leak}: Sensor leakage current
R: Bleeder resistance

4. Precautions on Sensor Surge Current

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

   Example
   In this example, the sensor's power supply voltage is used as the input to CIO 000000 and a 100-ms timer delay (the time required for an OMRON Proximity Sensor to stabilize) is created in the program. After the Completion Flag for the timer turns ON, the sensor input on CIO 000001 will cause output bit CIO 000100 to turn ON.



Output Wiring Precautions

Output Short-circuit Protection

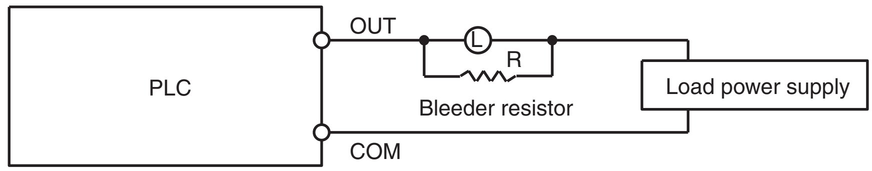
If a load connected to the output terminals is short-circuited, output components and the and printed circuit boards may be damaged. To guard against this, incorporate a fuse in the external circuit. Use a fuse with a capacity of about twice the rated output.

Transistor Output Residual Voltage

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

Output Leakage Current

If a Triac Output Unit is used to drive a low-current load, the leakage current may prevent the output device from turning OFF. To prevent this, connect a bleeder resistor in parallel with the load as shown in the following diagram.



Use the following formula to determine the resistance and rating for the bleeder resistor.

$$R < \frac{V_{ON}}{I}$$

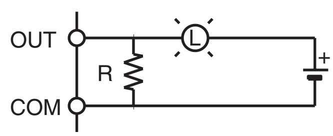
V_{ON}: ON voltage of the load (V)
I: Leakage current (mA)
R: Bleeder resistance (KΩ)

Output Surge Current

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

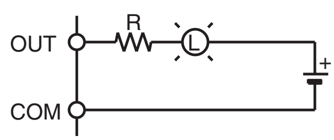
Method 1

Add a resistor that draws about 1/3 of the current consumed by the bulb.



Method 2

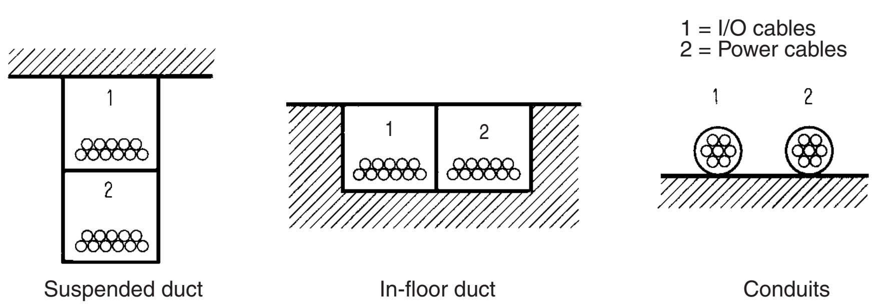
Add a control resistor as shown in the following diagram.



7-3-5 Reducing Electrical Noise

I/O Signal Wiring

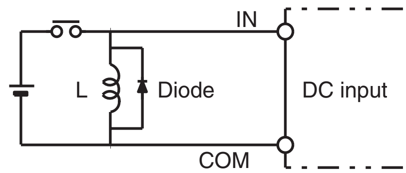
Whenever possible, place I/O signal lines and power lines in separate ducts or raceways both inside and outside of the control panel.

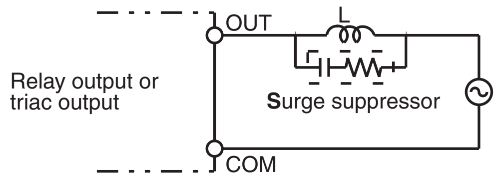


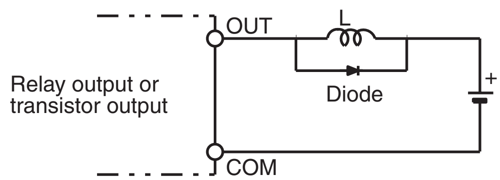
If the I/O wiring and power wiring must be routed in the same duct, use shielded cable and connect the shield to the GR terminal to reduce noise.

Inductive Loads

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







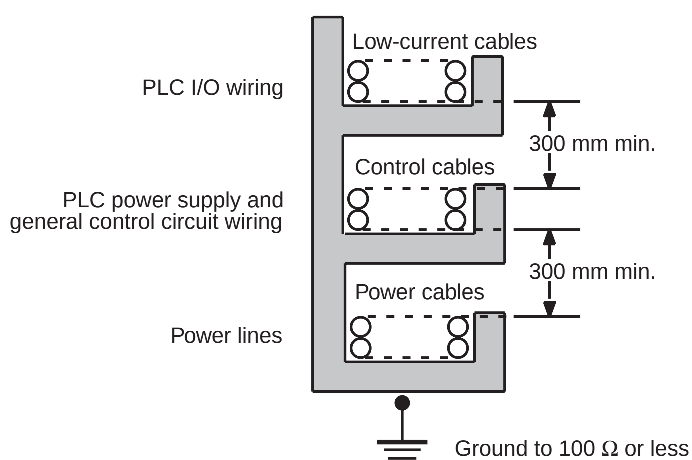
Note Use surge suppressors and diodes with the following specifications.

Surge suppressor specifications	Diode specifications
Resistor: 50 Ω Capacitor: 0.47 μF Voltage: 200 V	Breakdown voltage: 3 times load voltage min. Mean rectification current: 1 A

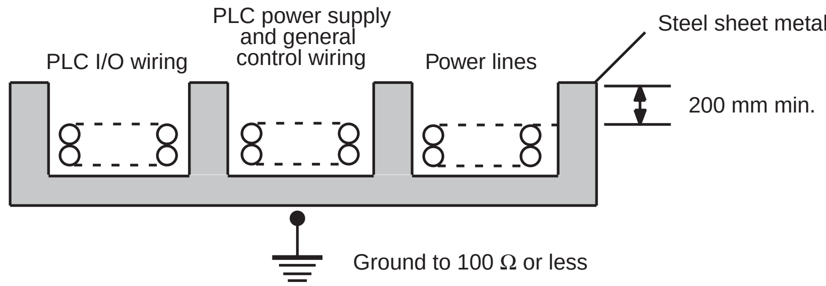
External Wiring

Observe the following precautions for external wiring.

- When multi-conductor signal cable is being used, avoid combining I/O wires and other control wires in the same cable.
- If wiring racks are parallel, allow at least 300 mm (12 inches) between the racks.



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



7-4 Network Installation

7-4-1 Basic Installation Precautions

- Take the greatest care when installing the Ethernet System, being sure to follow ISO 8802-3 specifications. You must obtain a copy of these specifications and be sure you understand them before attempting to install an Ethernet System. Unless you are already experienced in installing communications systems, we strongly recommend that you employ a professional to install your system.
- Do not install Ethernet equipment near sources of noise. If noise-prone environments are unavoidable, be sure to take adequate measures against noise interference, such as installing network components in grounded metal cases, using optical links in the system, etc.

7-4-2 Recommended Products

The following products are recommended for use with the CS1W-ETN21 Ethernet Unit.

Part	Maker	Model number	Specifications	Inquires
Hub	100BASE-TX			
	PHOENIX CONTACT	SWITCH 5TX	10/100 Mbit/s 5-port hub Designed for FA environments	Phoenix 045-931-5602 (in Japan only)
	Allied Telesis	RH509E	9-port hub	Allied Telesis (0120) 86-0442 (in Japan only)
		MR820TLX	9-port hub with 10Base-5 backbone port	
Twisted-pair cable	100BASE-TX			
	Fujikura	F-LINK-E 0.5mm x 4P	STP (shielded twisted-pair) cable: Category 5, 5e **Note:** Impedance is limited to 100 Ω.	---
Connectors (Modular plug)	STP Plug			
	Panduit Corp	MPS588	---	
Boots	Tsuko	MK Boots (IV) LV	---	

7-4-3 Precautions

Precautions on Laying Twisted-pair Cable

Basic Precautions

- Press the cable connector in firmly until it locks into place at both the hub and the Ethernet Unit.
- After laying the twisted-pair cable, check the connection with a 10Base-T cable tester.

Environment Precautions

- The UTP cable is not shielded, and the hub is designed for use in OA environments. Construct a system with shielded twisted-pair (STP) cable and hubs suitable for an FA environment.
- Do not lay the twisted-pair cable together with high-voltage lines.
- Do not lay the twisted-pair cable near devices that generate noise.
- Do not lay the twisted-pair cable in locations subject to high temperatures or high humidity.
- Do not lay the twisted-pair cable in locations subject to excessive dirt and dust or to oil mist or other contaminants.

Hub Installation Environment Precautions

- Do not install the hub near devices that generate noise.
- Do not install the hub in locations subject to high temperatures or high humidity.
- Do not install the hub in locations subject to excessive dirt and dust or to oil mist or other contaminants.

Hub Connection Methods

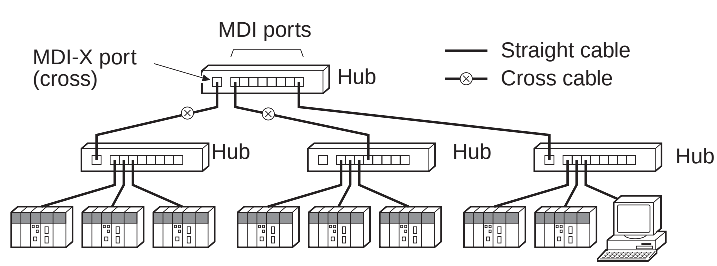
If more hub ports are required, they can be added by connecting more than one hub. There are two possible connection methods for hubs: Cascade and stacked.

Cascade Connections

- Connect two hubs to each other with twisted-pair cable as follows:
  Use a straight cable to connect an MDI port to an MDI-X port.
  Use a cross cable to connect two MDI ports together or two MDI-X ports together.

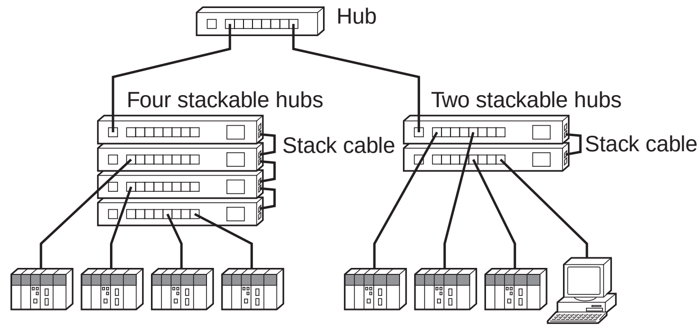
Note It is very difficult to distinguish cross cables and straight cables by appearance. Incorrect cables will cause communications to fail. We recommend using cascade connections with straight cables whenever possible.

- With cascade connections, up to 5 segments can be connected using up to 4 repeaters (i.e., 4 hubs).



Stack Connections

- Connect the hubs using special cables or special racks.
- Normally there is no limit to the number of hubs in a stack, and each stack is treated as one hub. Some hubs, however, are limited in the number of hubs per stack.

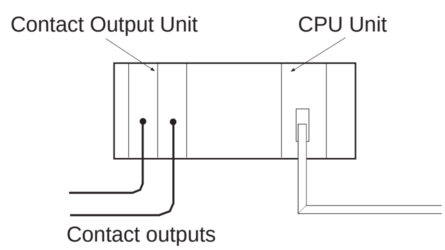


7-4-4 Using Contact Outputs (Common to All Units)

Communications errors can occur when Contact Output Units are mounted to the same Rack as the CPU Unit due to noise generated by the contact outputs. Use one or more of the following measures when installing Contact Output Units and Ethernet Units on the same Rack.

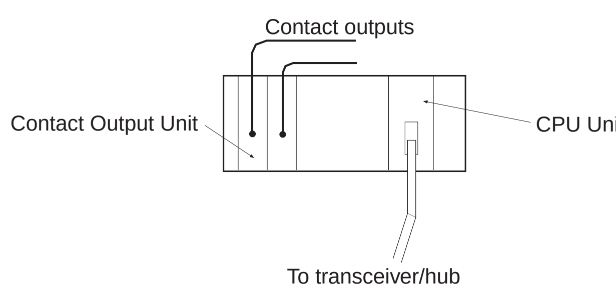
Mounting Location

Mount (or connect) any Contact Output Units as far away from the CPU Unit as possible.



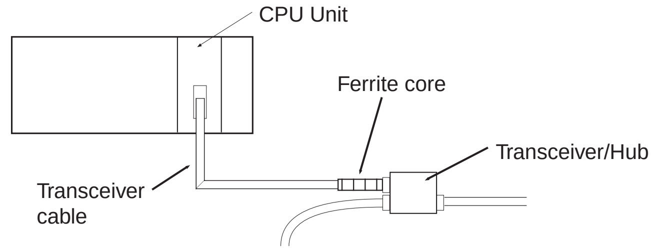
Cable Location

Separate the transceiver cable or twisted-pair cable connecting the Ethernet Unit as far from the wiring to the Contact Output Units as possible. The coaxial cable must also be placed as far away from the Contact Output Units and their wiring as possible.



Hub Measures

Attach several ferrite cores designed for EMI countermeasures on the transceiver cable or twisted-pair cable near the hub. The transceiver must also be placed as far away from the Contact Output Units and their wiring as possible.

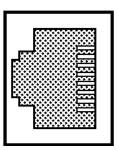


7-5 Connecting to the Network

7-5-1 Ethernet Connectors

The following standards and specifications apply to the connectors for the Ethernet twisted-pair cable.

- Electrical specifications: Conforming to IEEE802.3 standards.
- Connector structure: RJ45 8-pin Modular Connector (conforming to ISO 8877)

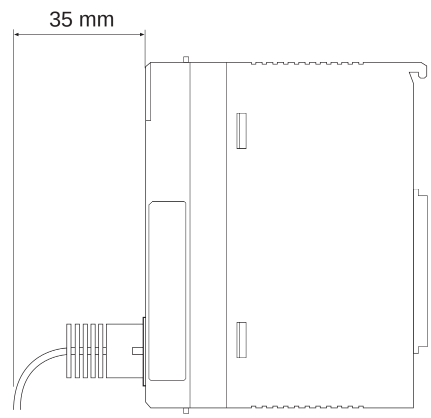


Connector pin	Signal name	Abbr.	Signal direction
1	Transmission data +	TD+	Output
2	Transmission data –	TD–	Output
3	Reception data +	RD+	Input
4	Not used.	---	---
5	Not used.	---	---
6	Reception data –	RD–	Input
7	Not used.	---	---
8	Not used.	---	---
Hood	Frame ground	FG	---

7-5-2 Connecting the Cable

⚠ Caution Turn OFF the PLC's power supply before connecting or disconnecting twisted-pair cable.

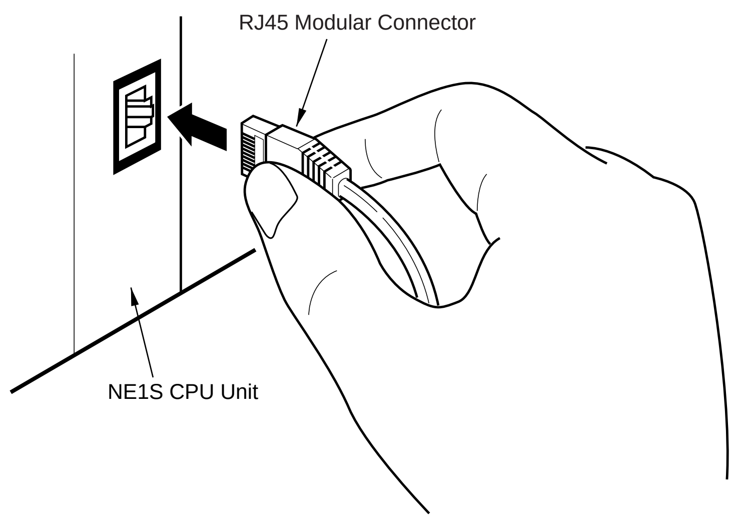
⚠ Caution Allow enough space for the bending radius of the twisted-pair cable as shown in below.



1,2,3...

1. Lay the twisted-pair cable.
2. Connect the cable to the hub. Be sure to press in the cable until it locks into place.

   Request cable installation from a qualified professional.
3. Connect the cable to the connector on the Ethernet Unit. Be sure to press in the cable until it locks into place.



SECTION 8
Software Installation

This section describes software installation.

8-1 Installation Preparations

8-1-1 System Requirements

A computer with the following specifications is required to run the Network Configurator and NE Programmer.

Item	Specification
Computer	IBM PC/AT or compatible
CPU	Pentium 300 MHz min.
OS	Microsoft Windows 2000 or Microsoft Windows XP
Supported languages	Japanese/English
Memory	128 MB min.
HDD	40 MD min. of available space
Monitor	S-VGA or better
CD-ROM	1 min.
Communications port to connect to the NETS	One or more of the following: USB port, RS-232C port, or Ethernet port Or one of the following: DeviceNet Interface Card (see note) or RA Communications Card (using RS Linx communications driver) **Note** Connections with NE1S CPU Units on the network is possible via USB port, RS-232C ports, or Ethernet ports.

To use the Network Configurator and NE Programmer, the personal computer must satisfy the following specifications.

8-1-2 Installation Types

Installing the NE Programmer and Network Configurator

For details on installing the NE Programmer and Network Configurator, refer to *8-2 Installing the NE Programmer and Network Configurator*. Both or either one of NE Programmer and Network Configurator can be installed with one installation procedure.

Installing the USB Driver

For details on connecting online using USB, refer to *8-3 Installing the USB Driver*.

8-2 Installing the NE Programmer and Network Configurator

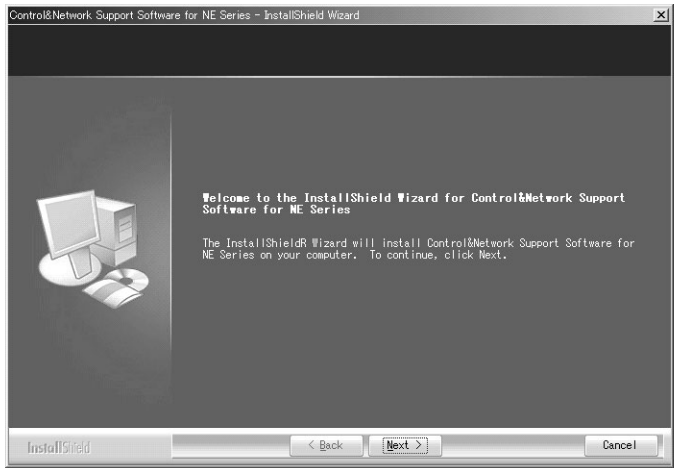
8-2-1 Installation Procedure

Use the following procedure to install the NE Programmer and Network Configurator.

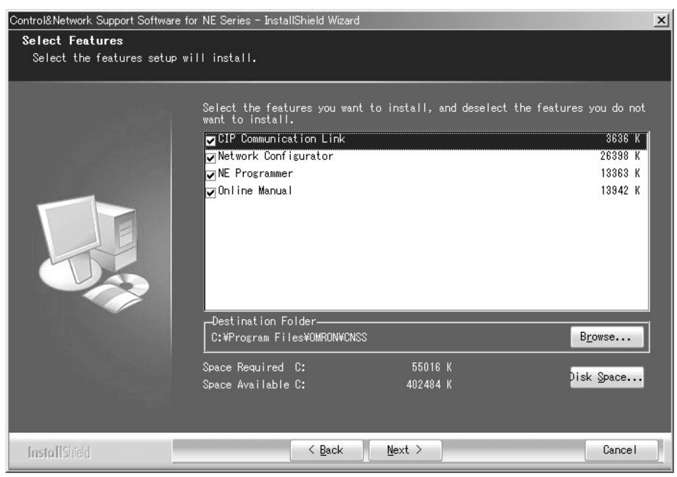
Note Administrator privileges are required to install the software.

1,2,3...

1. Insert the installation CD in the CD-ROM drive.
2. Execute setup.exe from the CD-ROM using either of the following methods.
   - Click the icon to access the CD-ROM and double-click the setup.exe file.
   - Select ***Run*** from the Start menu, browse the CD-ROM for the setup.exe file and then click the **OK** Button. The following window will be displayed.



3. Click the **Next** Button. The License Agreement Window will be displayed.
4. Select the ***I accept the terms of the License Agreement*** option and then click the **Next** Button. The Customer Information Window will be displayed.
5. Enter the *User Name, Company Name,* and *Serial Number,* and then click the **Next** Button. The following Select Features Window will be displayed.



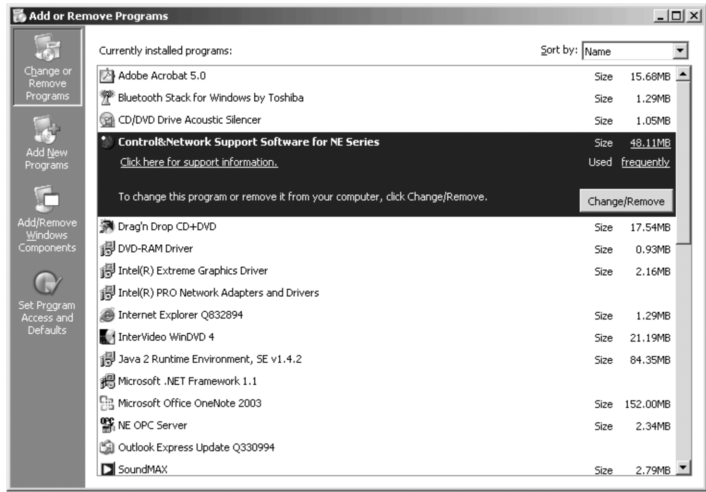
6. Select ***CIP Communication Link*** and any other programs to be installed (*CIP Communication Link* must be selected), specify the installation destination, and click the **Next** Button. The Ready to Install the Program Window will be displayed.
7. Click the **Install** Button to start installation. When installation completes normally, the InstallShield Wizard Complete Window will be displayed.
8. Click the **Finish** Button.

8-2-2 Uninstallation Procedure

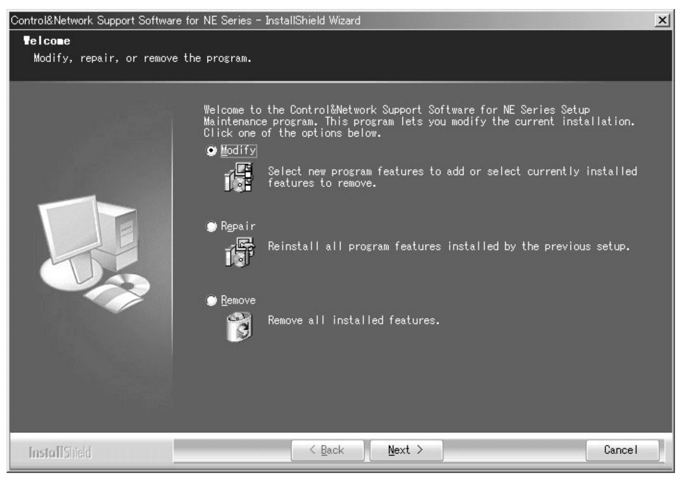
Use the following procedure to uninstall the NE Programmer or Network Configurator.

1,2,3...
1. Select ***Settings - Control Panel - Add/Remove Programs*** from the Start menu.

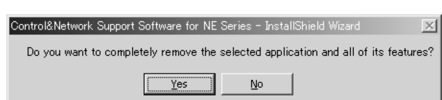
2. Select ***Control&Network Support Software for NE Series*** in the Add/Remove Programs Dialog Box, as shown below, and click the **Change/Remove** Button.



- The following window will be displayed.



3. Click the **Remove** Button, and then click the **Next** Button.



4. Click the **Yes** Button.
   The software will be uninstalled. When uninstallation is completed normally, the Maintenance Complete Window will be displayed.
5. Click the **Finish** Button.

8-2-3 Upgrading Software Versions

Use the following procedure to upgrade the version of the NE Programmer or Network Configurator.

Installing an Update

Install the updated version without uninstalling the existing software.

1,2,3...

1. The updated version is provided as an executable (.exe) file, so either of the following methods can be used to execute the update file.
   - Double-click the file and click the **OK** Button.
   - Select ***Run*** from the Start menu, browse and select the update file, and click the **OK** Button.

Installing a Release

1,2,3...

1. Uninstall the software to be upgraded (refer to *8-2-2 Uninstallation Procedure*).
2. Install the new software version (refer to *8-2-1 Installation Procedure*).

8-3 Installing the USB Driver

Note Administrator privileges are required to install the software.

Use the following procedure to install the USB driver.

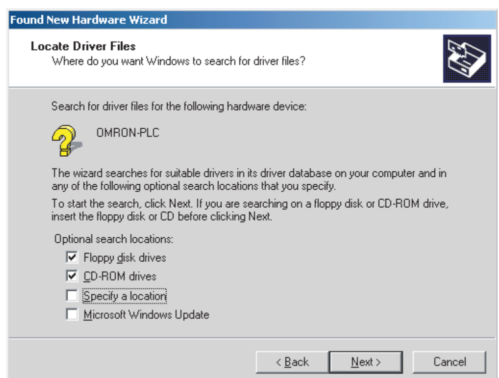
1,2,3...

1. Connect the CPU Unit and the computer with a USB cable.
2. Turn ON the power supply to the CPU Unit.
   The following dialog box will be displayed.

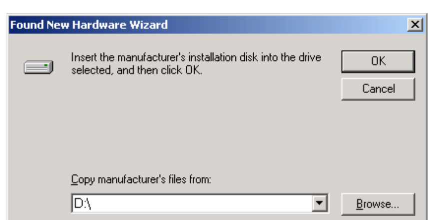
After a few moments, the following dialog box will be displayed.

3. Click the **Next** Button.
   The following dialog box will be displayed.

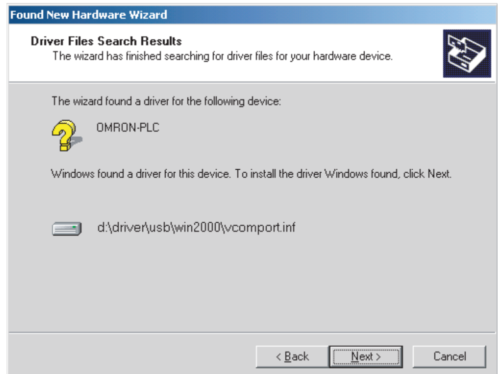
4. Select the Search for a suitable driver for my device (recommended) option and click the **Next** Button.
   The following dialog box will be displayed.



5. Select only the *Specify a location* option (do not select any other options), and click the **Next** Button.
   The following dialog box will be displayed.



6. Click the **Browse** Button, specify the *CD-ROM\driver\usb\win2000* folder in the dialog box that is displayed, and click the **OK** Button.
   The following dialog box will be displayed.



7. Click the **Next** Button.
   The following dialog box will be displayed if the USB driver is installed normally.

8. Click the **Finish** Button.
   This completes the installation of the USB driver.

SECTION 9
Overview of the Network Configurator

This section introduces the Network Configurator.

9-1 What is the Network Configurator?

The Network Configurator is used to build, set, and manage DeviceNet, ControlNet, and EtherNetIP networks on graphic window operations. Virtual DeviceNet and ControlNet networks can be built on the Network Configuration Windows of the Network Configurator to enable setting and monitoring the actual networks.

The NE Program can be started from the Network Configurator and network variables registered with the NE Programmer can be shared between the Network Configurator and NE Programmer.

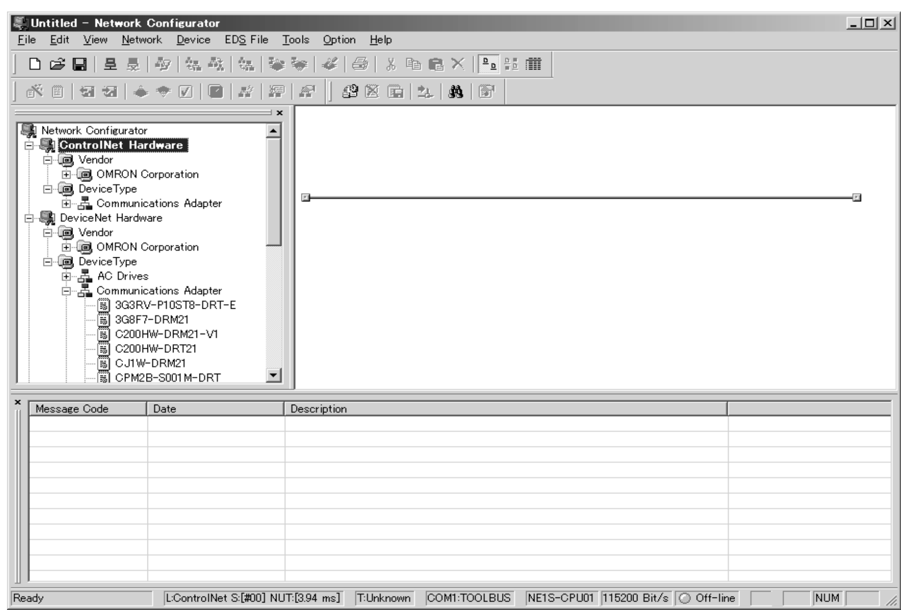
Note This section introduces the Network Configurator and describes basic operations, such as starting the Network Configurator and selecting networks. Refer to the *NE1S DeviceNet Unit Operation Manual* and the *NE1S ControlNet Unit Operation Manual* for details on settings and operating methods.

9-2 Starting the Network Configurator

Starting Procedure

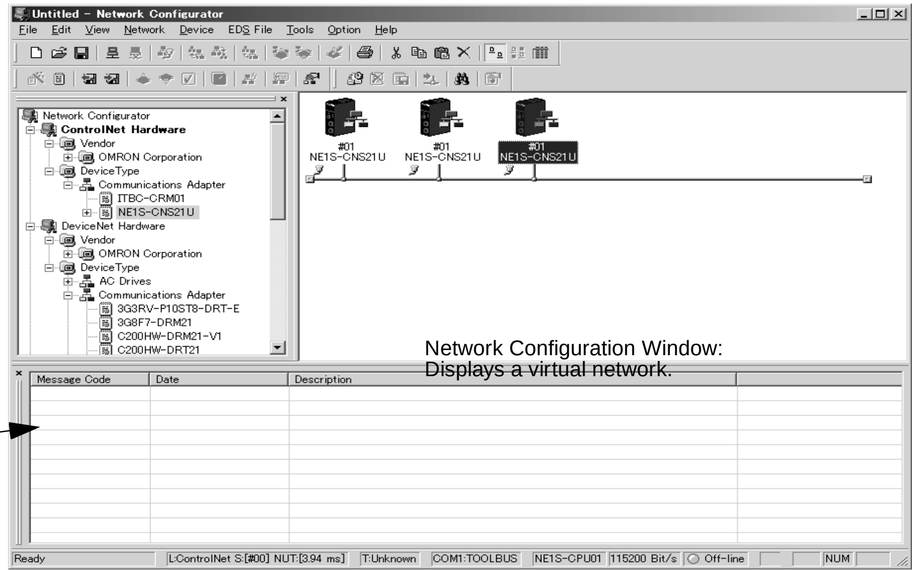
To start the Network Configurator, select ***Start - Program - OMRON Control&Network Support Software - Network Configurator - Network Configurator***.

The initial window will be displayed. DeviceNet will be selected as the network when the Network Configurator is first started. Refer to *9-3* for the procedure to select the network.



Main Window

The Main Window consists of a Hardware List and a Network Configuration Window, as shown below.



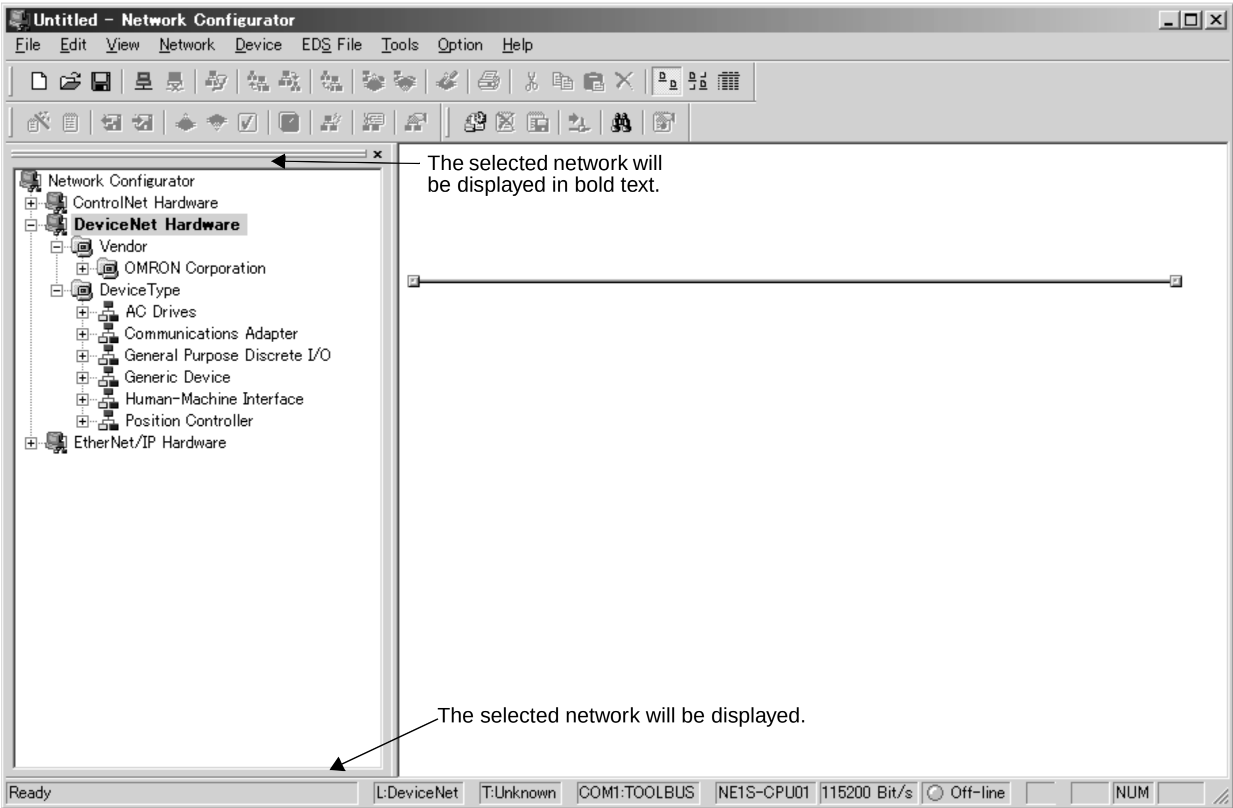
Hardware List:
Displays the devices that can be added to the network.

9-3 Selecting Networks

One and only network configuration can be saved in each project file (*.ncf). After starting the Network Configurator use the following procedure to select the desired type of network.

To select a network, select ***Network - New*** and then select ***DeviceNet, ControlNet***, or ***EtherNetIP***.

A Network Configuration Window will be displayed for the selected network type. The following window shows an example for a ControlNet network.

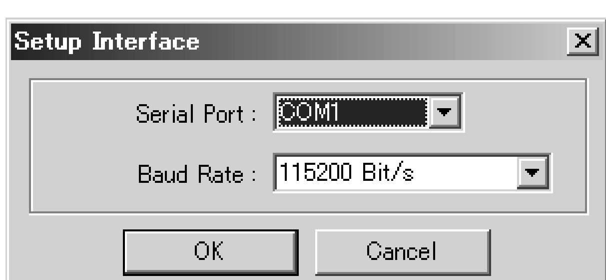


9-4 Network Connections

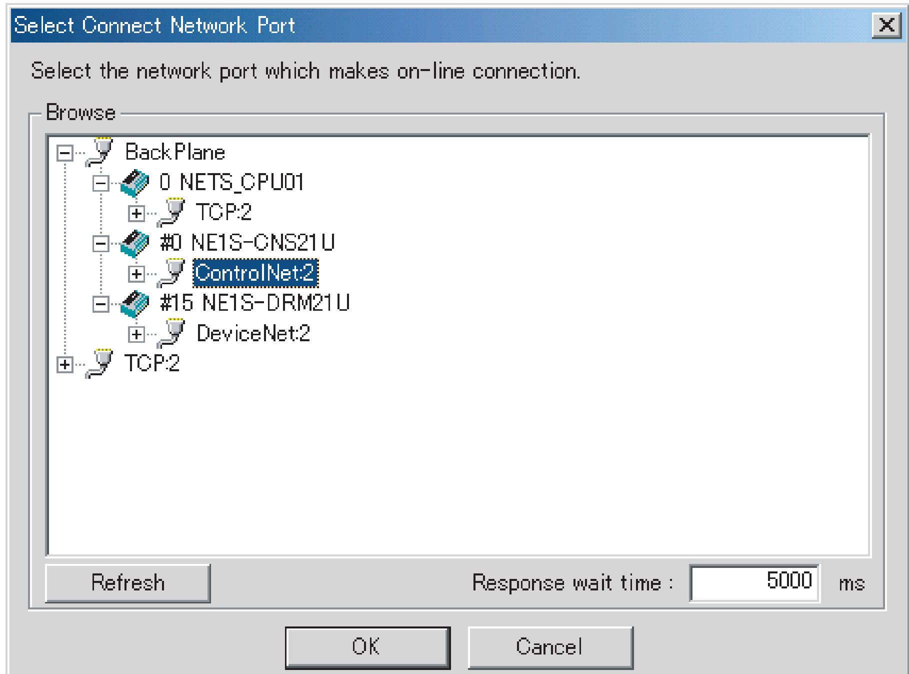
Use the following procedure to connect to a device on the network. This example is written to connect to a device on a ControlNet network (i.e., ***Network - New - ControlNet*** has already been selected).

1,2,3...

1. Select the communications interface.
   Select ***Option - Select Interface***, and then select ***NETS Serial PORT, DeviceNet I/F, Ethernet I/F, ControlNet***, or ***RSLinx***.
   In this example, ***NETS Serial PORT*** and ***USB port*** are selected.
2. Connect to the device.
   - Select ***Network - Connect***.
     (The following example is for when NETS Serial PORT has been selected.)
   - The Select Interface Dialog Box will be displayed.



- Select the ***COM Port*** (USB port) and the ***Baud Rate*** and then click the **OK** Button.
- The Select Network Port Dialog Box will be displayed.



- Select the network port to connect to (***ControlNet2*** is selected above) and then click the **OK** Button.

On-line will be displayed at the bottom of the screen if an online connection is made normally.

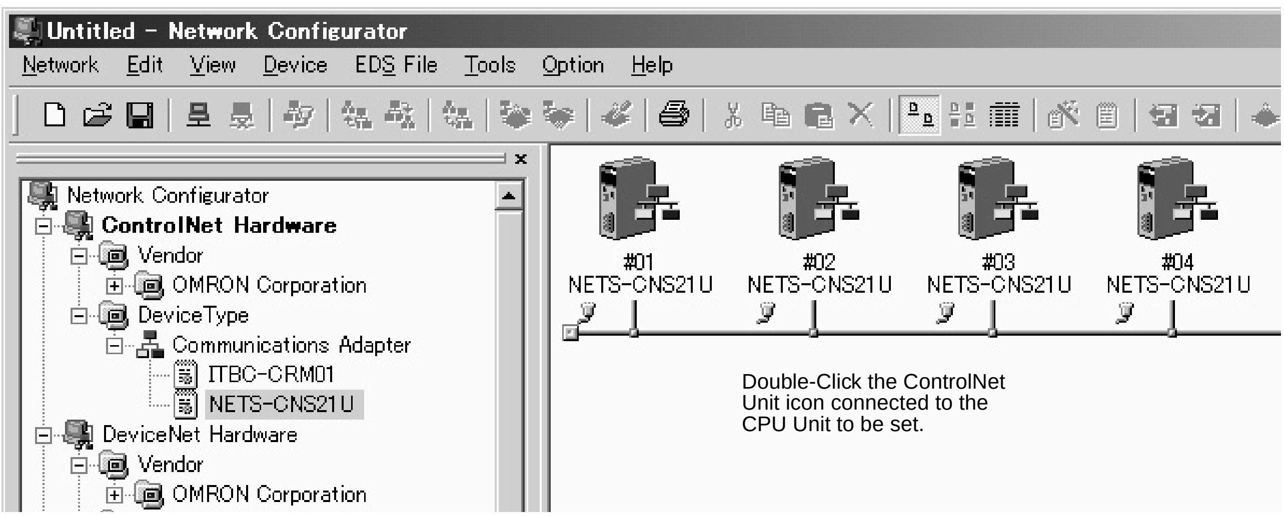
9-5 Starting NE Programmer

9-5-1 Starting NE Programmer from Network Configurator

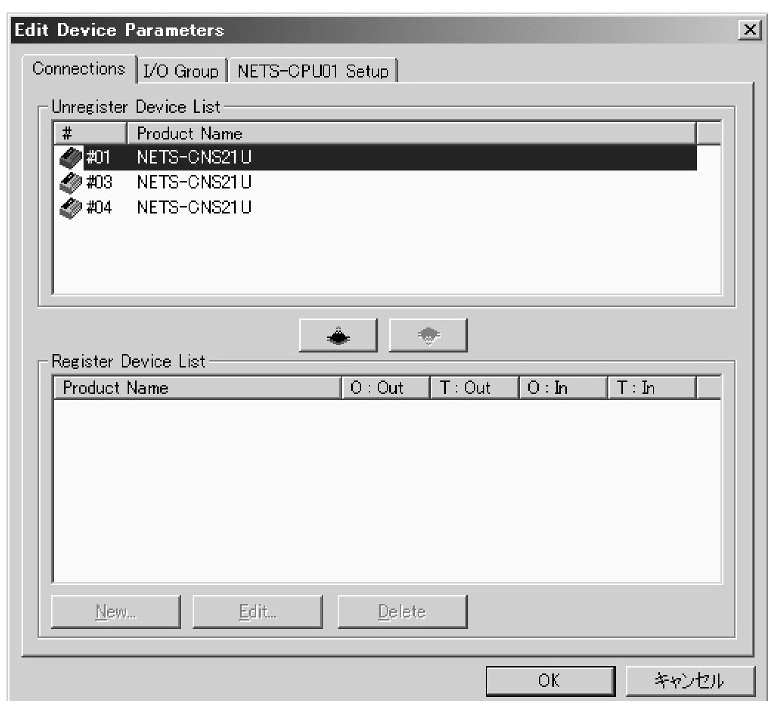
Use the following procedure to start the NE Programmer from the Network Configurator. It is necessary to start the NE Programmer from the Network Configurator to import network variables created on the NE Programmer into the Network Configurator. Refer to *SECTION 10 Outline of NE Programmer and Its Functions* for the procedure to start the NE Programmer by itself.

The following procedure using a ControlNet network as an example for starting the NE Programmer from the Network Configurator.

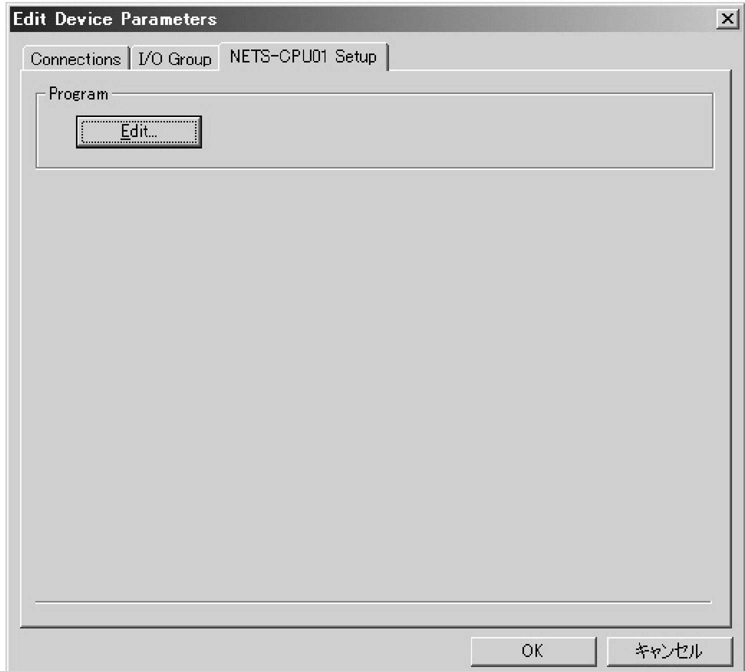
1,2,3... 1. In the Network Configuration Window, double-click the ControlNet Unit icon connected to the CPU Unit to be set.



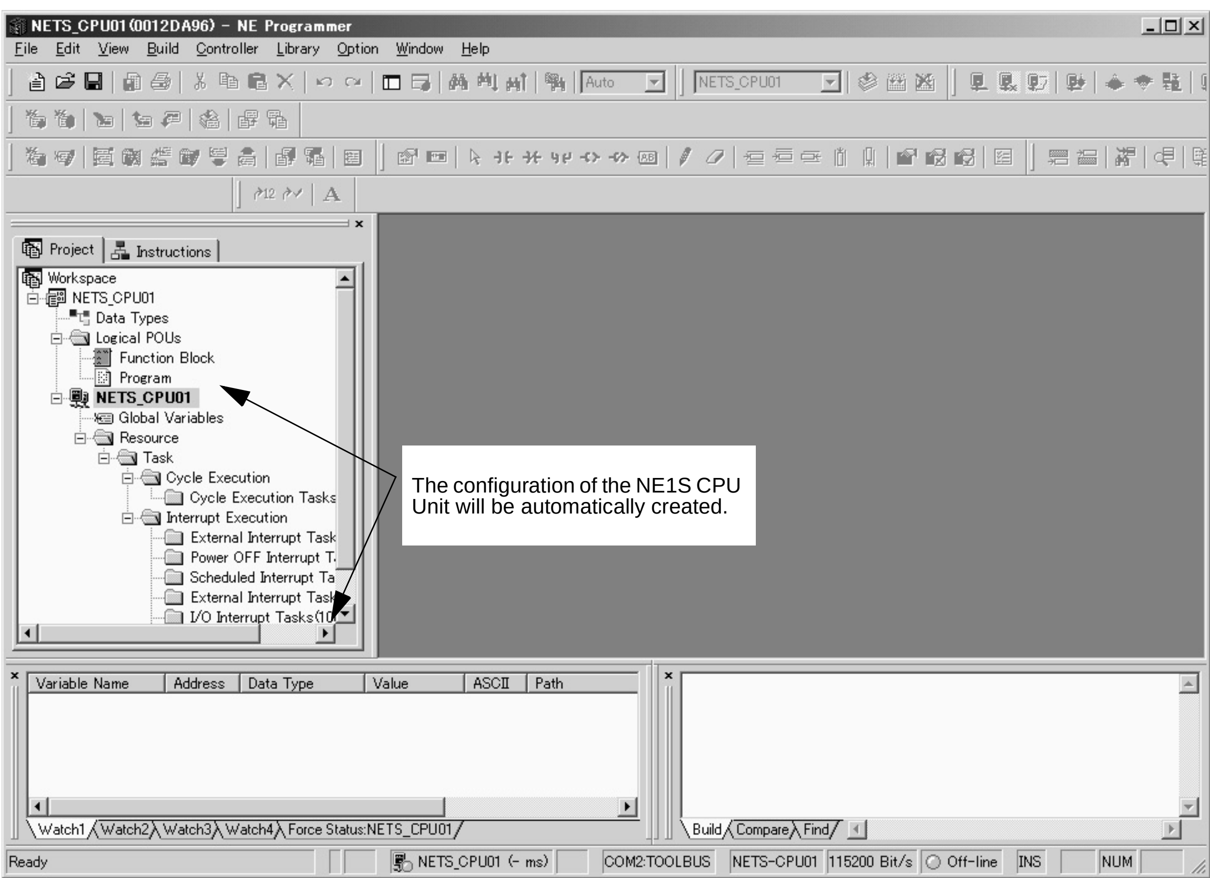
The Device Parameter Dialog Box will be displayed.



2. Click ***NETS-CPU01 Setup*** Tab.
   The ***NETS-CPU01 Setup*** Tab Page will be displayed.



3. Click the **Edit** Button.
   The NE Programmer will be started with a new project created.



Ladder programming can be input, variables can be created, and other operations can be performed on the NE Programmer. Refer to *SECTION 10 Outline of NE Programmer and Its Functions* to *SECTION 13 Online Operation* for an overview and operating procedures for the NE Programmer.

9-5-2 Precautions in Starting NE Programmer from Network Configurator

There are differences between when the NE Programmer is started from the Network Configurator and when it is started by itself, as described below.

Refer to *SECTION 10 Outline of NE Programmer and Its Functions* for the procedure to start the NE Programmer by itself. Refer to *SECTION 11 Programming* to *SECTION 13 Online Operation* for operating procedures.

Quitting the NE Programmer

After starting the NE Programmer from the Network Configurator, always exit the NE Programmer when returning to the Network Configurator. It will not be possible to use the Network Configurator unless you end the NE Programmer.

Saving

When saving from the NE Programmer, an NE Programmer project file (*nlx) will not be saved; rather, the data will be saved as part of the Network Configurator project (*.ncf). Once the Network Configurator project file (*.ncf) has been saved, the project file will be automatically opened the next time the NE Programmer is started from the Network Configurator.

NE Programmer Online Operations

Online connections, disconnections, changing connection locations, and automatic uploading are not possible.

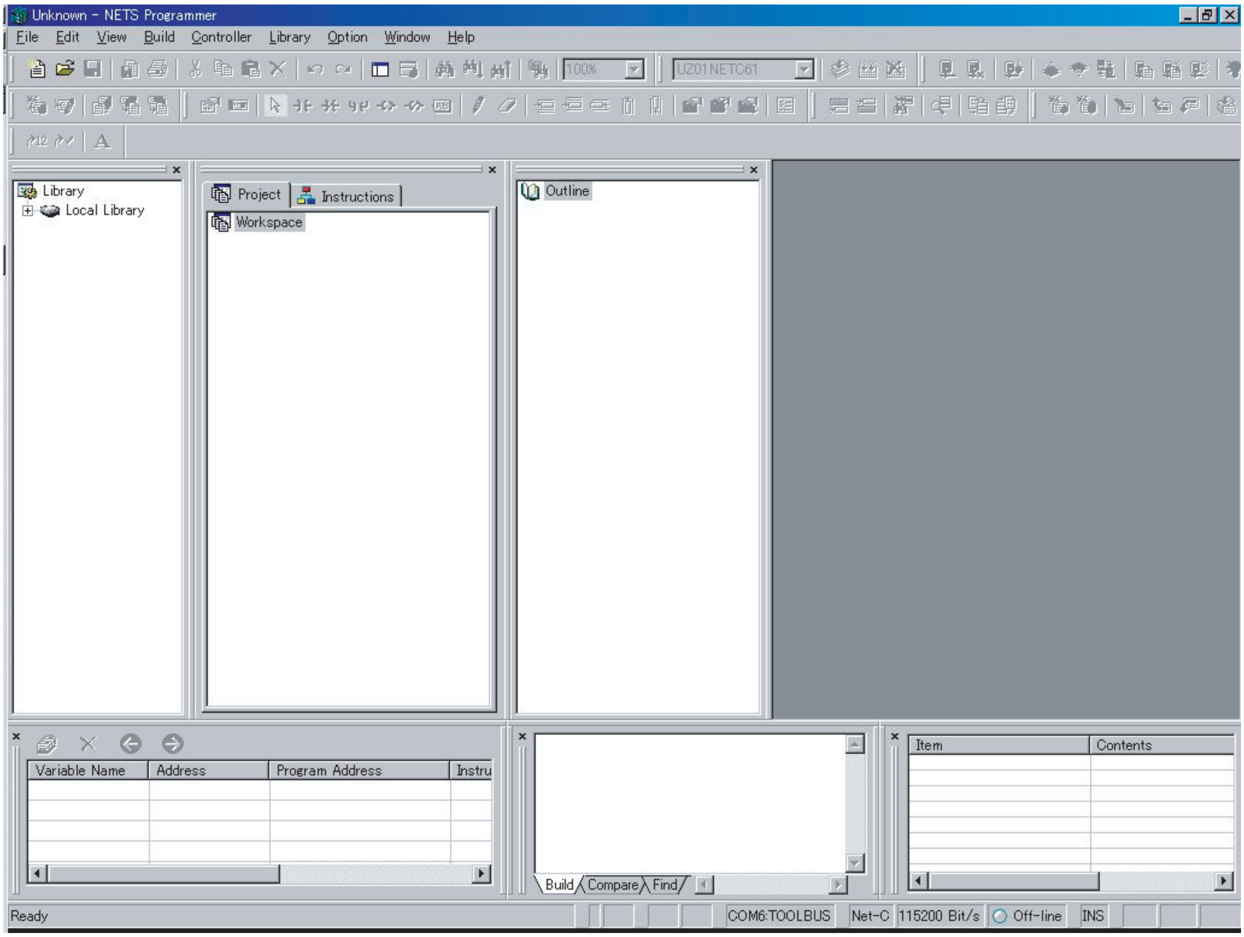
SECTION 10
Outline of NE Programmer and Its Functions

This section provides an outline of the functions of the NE Programmer.

10-1 Starting the NE Programmer

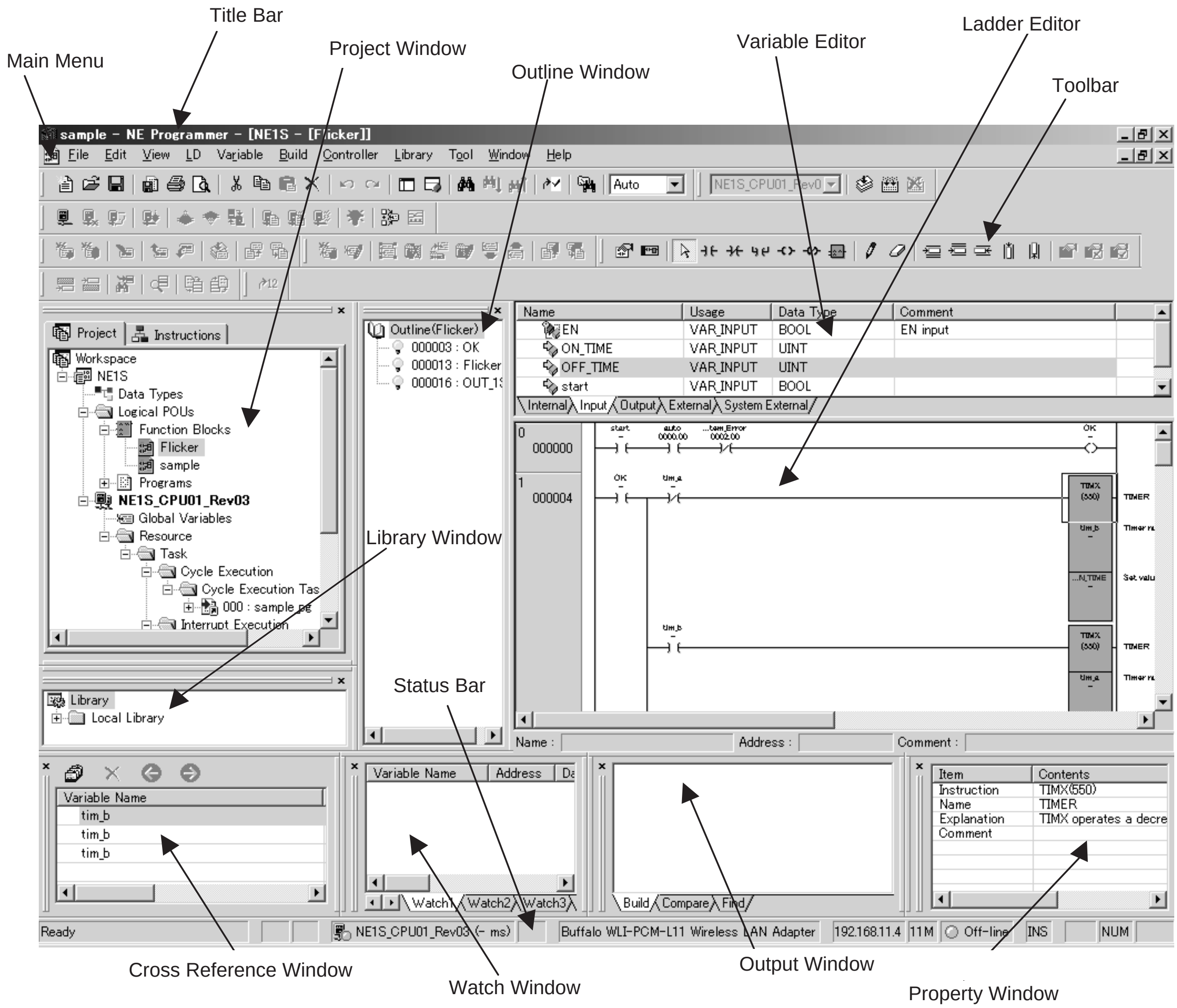
To start the Network Configurator, select ***Start - Program - OMRON Network Tools - NE Programmer - NE Programmer***.

The following initial window will be displayed.



10-2 Main Window

This section describes the main functions of the Main Window.



Name	Function
Title Bar	Displays the file name.
Main Menu	Commands are selected from this menu.
Tool Bar	Functions are selected by clicking the icons.
Project Window	Used to manage programs and data.
Outline Window	Displays an outline of the ladder program currently being edited. If any part of the outline is clicked, a jump will be made to the corresponding location in the program.
Variable Editor	Used to registered and edit variables.
Ladder Editor	Used to create and edit ladder programs. Up to 15 Ladder Editor Windows can be displayed at the same time.
Library Window	Used to manage user-defined libraries (programs, function blocks, and program parts).
Cross Reference Window	Displays other instructions that use the variable at the cursor location. If any cross-referenced instruction is clicked, a jump will be made to the corresponding location in the program.
Watch Window	Displays the present values of registered variables and physical addresses.
Output Window	Displays various information, such as compiling error information and the results of program comparison.

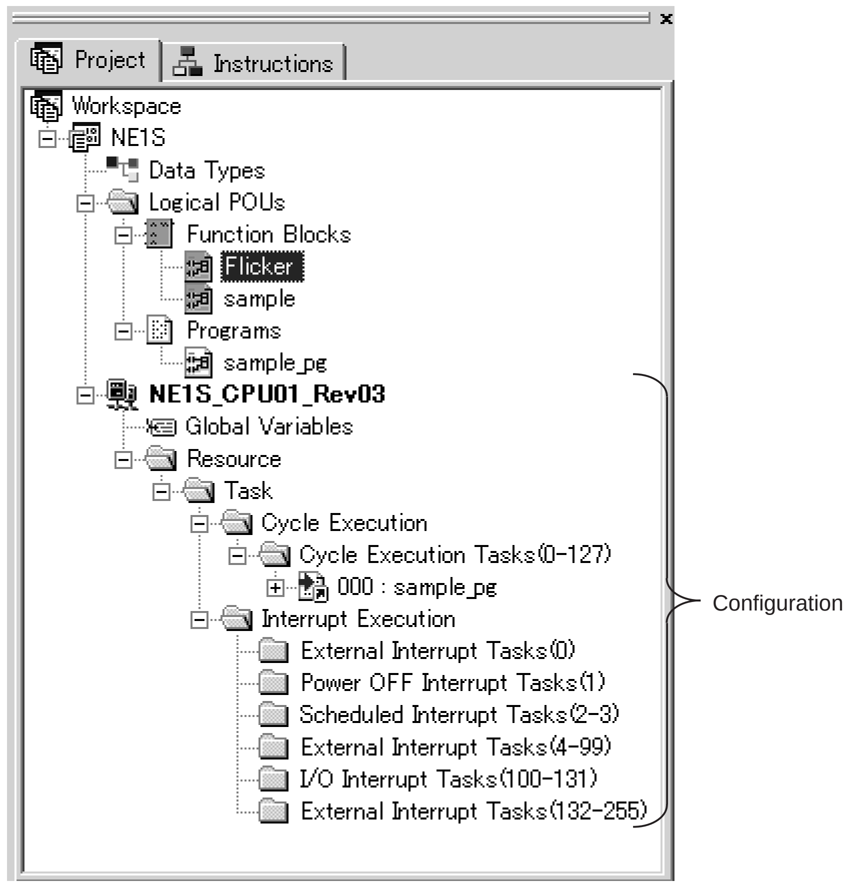
Name	Function
Property Window	Displays the properties of the instruction at the cursor position.
Status Bar	Displays information, such as the configuration name and online/offline status.

10-3 Project Window

This section describes the parts of the Project Window.

In the initial status, there are no elements created in the Project Window except for the NE Programmer workspace.

Refer to *SECTION 5 Memory Areas* for the procedures to create elements.



Name	Description
Workspace	Used as a workspace for the NE Programmer.
Project	A project is a unit of data consisting of the library objects, data types, logical POUs, and configurations that are being edited.
Data Types	A folder that displays the data types that can be used in the project and is used to declare data types.
Logical POUs	Logical POUs (program organization units) are units used to configure programs. POUs are the units by which software can be reused. POUs can be one of two types of element, function blocks and programs. Function blocks are called from a program and programs are allocated to tasks for execution.
Configuration	A configuration is the element that corresponds to an entire PLC system. Global variables can be declared for a configuration.
Resource	Resources consist of programs allocated to tasks and local variables. With the NE1S, one resource folder exists in the configuration.

10-4 Menu Item Lists

This section lists the functions on the main and submenus in the Main Window of the NE Programmer.

10-4-1 File Menu

Menu	Shortcut	Function
New - Project	Ctrl + N	Creates a new project.
New - Configuration		Creates a new configuration.
New - POU		Creates a new logical POU.
Open	Ctrl + O	Opens an existing project.
Save	Ctrl + S	Saves the project being edited (overwrites current project data).
Save As		Saves the project being edited under a new name.
Save Changes to project	Ctrl + Shift + S	Updates the project for the program or function block that is being edited.
Edit Data types		Opens the Data Type Editor.
Configuration - Edit Global Variables	Alt + G	Edits global variables.
Configuration - Replace physical addresses in programs	Ctrl + Shift + G	Replaces physical addresses in the program with global variables specified by that physical address.
Task - Create		Adds a task.
Task - Allocate - (*program name)*		Allocates a program to the selected task.
Task - Release		Releases a task allocation.
Task - Execute on Startup		Sets the task to be active (i.e., to be executed) at startup.
Task - Standby on Startup		Sets the task to be inactive (i.e., not to be executed) at startup.
Logical POU - Edit		Displays the Ladder Editor and Variable Editor for the currently selected program or function block.
Logical POU - Delete		Deletes the currently selected program or function block.
Logical POU - Protect		Sets read protection for a program or function block.
Print	Ctrl + P	Prints the specified items.
Print Preview		Displays a preview for printing.
Setup Printer		Sets the printer to use for printing.
Page Setup		Sets the margins, title, header, and footer to use for printing.
Export Variable - OMRON OPC Server File		Exports variables in NE OPC Server, CX-Designer, or SPU-COnsole format.
Exit		Exists the application.

10-4-2 Edit Menu

Menu	Shortcut	Function
Undo	Ctrl + Z	Undoes the previously performed operation.
Redo	Ctrl + Y	Redoes the operation that was undone with *Undo*.
Cut	Ctrl + X	Cuts the specified range of data.
Copy	Ctrl + C	Copies the specified range of data.
Paste	Ctrl + V	Pastes the contents of the clipboard.
Delete	Del	Deletes the specified range of data.
Rename	F2	Used to change the name of a program or function block.
Select All	Ctrl + A	Selects all data.
Find	Ctrl + F	Searches for a text string.
Find Next	F3	Finds the next instance of a text string.
Find Prev	Shift + F3	Finds the previous instance of a text string.
Replace	Ctrl + H	Replaces a text string.

Menu	Shortcut	Function
Bookmark	Ctrl + B	Registers a bookmark or jumps to a bookmark.
Find in Programs	Ctrl +Shift+ F	Searches mnemonics, variables, physical addresses, instance names, line comments, and instruction comments in all POUs. The specific items to be searched can be specified. The search results are displayed in the Output Window and relevant locations can be jumped to by double-clicking on the search results.

Additional Menu Items when Editing Ladder Programs

Menu	Shortcut	Function
Jump - Jump	Ctrl + G	Jumps to a specified step number or rung number.
Jump - Bit Address Reference	Tab	Searches for contacts and coils that affect each other. If a contact is specified, a search is made backward for a coil with the same bit address. If a coil is specified, a search is made foreword for a contact with the same bit address.
Jump - Next Operand Reference	Shift + Alt + N	Jumps to the next operand.
Jump - Next Input	Shift + Alt + I	Jumps to the next input.
Jump - Next Output	Shift + Alt + O	Jumps to the next output.
Jump - Previous Jump Point	Shift + Alt + B	Returns to the previous instruction found with *Next Address Reference*.
Jump - Jump Variable Define		Jumps to the position in the Variable Editor where the selected instruction operand's variable is defined (declared).
Edit Instruction		Used to edit the currently selected instruction or function block instance name.
Edit Variable	F8	Used to edit the selected instruction operand's variable.
Edit Comment		Used to edit the comment for the currently selected instruction or function block.
Insert Line Comment		Inserts a line comment.
Copy Operand	Crtl + Shift + C	Used to copy the operand only.
Update Function Block Instance		Used to update changes in the input and output variables when the input and output variables of the logical POU's function block were changed after the function block was pasted as an instance. (The ⚠: icon will appear next to an instance when there has been a change.)
Edit Using Mnemonic Editor		Used to edit one rung of a ladder program with the Mnemonic Editor.

Additional Menu Items when Editing Mnemonic Programs

Menu	Shortcut	Function
Jump - Jump	Ctrl + G	Jumps to a specified step number.
Jump - Jump Variable Define		Jumps to the position in the Variable Editor where the selected instruction operand's variable is defined (declared).
Insert upward	Ctrl + G	Inserts an instruction above the current row.
Insert downward		Inserts an instruction below the current row.

Additional Menu Items when Editing ST Programs

Menu	Shortcut	Function
Jump - Go To Line	Ctrl + G	Jumps to a specified line number.
Jump - Jump Variable Define		Jumps to the position in the Variable Editor where the selected ST program variable is defined (declared).

10-4-3 View Menu

Menu	Shortcut	Function
Toolbars: General, Data Type, Build, Controller, Variable, LD, Mnemonic, ST		Displays and hides the toolbars in the Project Window.
Status Bar		Displays/hides the status bar.
Window - Workspace	Alt + 1	Displays/hides the Workspace Window.
Window - Output	Alt + 2	Displays/hides the Output Window.
Window - Watch	Alt + 3	Displays/hides the Watch Window.
Window - Cross Reference	Alt + 4	Displays/hides the Cross Reference Window.
Window - Library	Alt + 5	Displays/hides the Library Window.
Window - Property	Alt + 6	Displays/hides the Property Window.
Window - Outline	Alt + 7	Displays/hides the Outline Window.
Window - Initialize window position		Returns the window arrangement to its initial status (the status after the NE Programmer was first installed).
Variable Editor - Upper		Changes the Variable Editor’s position to the top (default position), effective the next time the Variable Editor starts.
Variable Editor - Right		Changes the Variable Editor’s position to the right side, effective the next time the Variable Editor starts.
Variable Editor - Lower		Changes the Variable Editor’s position to the bottom, effective the next time the Variable Editor starts.
Variable Editor - Left		Changes the Variable Editor’s position to the left side, effective the next time the Variable Editor starts.
Variable Editor - Visible		Displays/hides the Variable Editor.

10-4-4 Ladder Menu

The Ladder Menu is displayed only when the Ladder Editor is active.

Menu	Shortcut	Function
Mode - Select		Switches the editing mode to Select Mode.
Mode - Normally Open Contact		Switches the editing mode to NO Contact Mode.
Mode - Normally Closed Contact		Switches the editing mode to NC Contact Mode.
Mode - Contact OR		Switches the editing mode to Contact OR Mode.
Mode - Coil		Switches the editing mode to Coil Mode.
Mode - Negated Coil		Switches the editing mode to Negative Coil Mode.
Mode - Function Block		Switches the editing mode to Function Block Variable Input Mode.
Mode - Line		Switches the editing mode to Draw Line Mode.
Mode - Erase		Switches the editing mode to Erase Line Mode.
Rung - Select		Selects a rung.
Rung - Insert Row Above	Ctrl + I	Inserts an open row above the cursor position.
Rung - Insert Row Below	Ctrl + Shift + I	Inserts an open row below the cursor position.
Rung - Delete Row	Ctrl + D	Deletes the selected row.
Rung - Insert Column		Inserts a column at the cursor position.
Rung - Delete Column		Deletes the column at the cursor position.
Draw Line	Ctrl + L	Switches the editing mode to Draw Line Mode, so the cursor becomes the starting point for line drawing.
Erase Line	Ctrl + Shift + L	Switches the editing mode to Erase Line Mode, so the cursor becomes the starting point for line erasing.
Change Variable Usage - Input/ Output/Internal		Changes a variable for a function block to an input variable, output variable, or internal variable.
Immediate Refresh		Changes an instruction between a immediate refresh instructions and a normally refreshed instruction.

Menu	Shortcut	Function
Invert (NOT)	Ctrl + R	Switches between A (NO) contacts and B (NC) contacts and between output contacts and negative output contacts.
Transition Sensing - Nothing/Positive/Negative		Sets or releases a transition (differential) condition for an input contact.
Online Edit - Begin		Starts online editing.
Online Edit - Cancel		Cancels online editing.
Online Edit - Finish		Ends online editing.
Add to Watch		Adds an instruction operand to the Watch Window.
View Program		Opens a function block pasted in a program.

10-4-5 ST Menu

This menu is displayed only when the ST editor is active.

Menu	Shortcut	Function
Add Variable	Ctrl + R	Adds the selected text string to the variable.

10-4-6 Mnemonic Menu (Displayed in Mnemonic Editor Only)

Menu	Shortcut	Function
Change Variable Usage - Input/Output/Internal		Changes a variable for a function block to an input variable, output variable, or internal variable.
Immediate Refresh		Changes an instruction between a immediate refresh instructions and a normally refreshed instruction.
Invert (NOT)	Ctrl + R	Switches between A (NO) contacts and B (NC) contacts and between output contacts and negative output contacts.
Differentiate - Nothing/Positive/Negative		Sets or releases a transition (differential) condition for an input contact.
Add to Watch		Adds an instruction operand to the Watch Window.
Import		Imports mnemonic data from a CSV file.
Export		Exports mnemonic data to a CSV file.

10-4-7 Variable Menu

Menu	Shortcut	Function
Add		Adds new variables.
Edit		Deletes existing variables.
Group Input/Output Variables - Group	Ctrl + G	Creates a group of I/O variables for a function block to make them the function block easier to understand.
Group Input/Output Variables - Release Group	Ctrl + Shift + M	Ungroups the I/O variables for a function block.
Group Input/Output Variables - Release Member	Ctrl + M	Removes the selected members from the group.
Rename	Ctrl + E	Changes the group name.
Change Variable Usage - Input/Output/Internal/External		Changes the variable to an input variable, output variable, internal variable, or external variable
External Variable Consistency Check		Checks whether an external variable and global variable match.
Up	Ctrl + ↑	Shifts the selected member up one rung.
Down	Ctrl + ↓	Shifts the selected member down one rung.
Add to Watch		Adds a variable to the Watch Window.
Import - CSV Format		Imports variables from a CSV-format file.
Export - CSV Format		Exports variables to a CSV-format file.

10-4-8 Data Type Menu

The Data Type Menu is displayed only when the Data Type Editor is active.

Menu	Shortcut	Function
Insert - Struct		Inserts a data structure.
Insert - Element		Inserts an element into a data structure.
Edit		Edits a user-defined data structure.
Move Upward		Moves a user-defined data structure upward.
Move Downward		Moves a user-defined data structure downward.
Check		Checks a data structure for errors.
Import - CSV File		Imports a data structure from a CSV-format file.
Export - CSV File		Exports a data structure to a CSV-format file.

10-4-9 Build Menu

Menu	Shortcut	Function
Compile	Ctrl + F7	Compiles and performs a program check on the active program or function block.
Build	F7	Builds the entire program.
Stop build		Stops building a program.

10-4-10 Controller Menu

Menu	Shortcut	Function
Connect	Ctrl + W	Used to select the communications port and then connect online to the CPU Unit.
Disconnect	Ctrl + Shift + W	Disconnects from the CPU Unit.
Change Controller		Changes the device to which the connection is made (eliminates the need to disconnect).
Change Controller		Changes the controller type.
Auto Upload	Ctrl + Shift + A	Automatically finds the connected PLC model and communications conditions, connects online, and uploads the program.
Auto Upload - Select Port		Used to set the default COM port to use for automatic online connection with a USB interface.
Upload from Controller	Ctrl + Shift + T	Used to upload the program, PLC Setup, TCP/IP settings, and I/O tables from the CPU Unit to the computer.
Download to Controller	Ctrl + T	Used to download the program, PLC Setup, TCP/IP settings, and I/O tables from the computer to the CPU Unit.
Compare with Controller		Compares the programs on the computer and in the CPU Unit.
System Configuration		Used to set up the PLC.
Operating Mode - Program/Monitor/Run	Ctrl + 1/Ctrl + 3/Ctrl + 4	Changes the PLC's operating mode.
Monitor	Ctrl + M	Starts the monitor.
Backup value of variables		Saves the present variable values to a CSV file.
Restore value of variables		Restores saved variable values from a CSV file.
I/O Table - Create		Creates the real I/O tables.
I/O Table - Delete		Deletes the registered I/O tables. The CPU Unit will operate with the real I/O tables.
I/O Table - Compare		Compares the real I/O tables and registered I/O tables.
Clear Error	F4	Clears an error.
Clear Memory		Clears the CPU Unit memory, including the user program, parameter area, and I/O memory.
Restart Service		Starts the SMTP server and SNMP server.
Set - On		Sets (turns ON) a bit.

Menu	Shortcut	Function
Set - Off		Resets (turns OFF) a bit.
Set - Force On	Ctrl + J	Force-sets (forces ON) a bit.
Set - Force Off	Ctrl + K	Force-resets (forces OFF) a bit.
Set - Force Cancel		Clears a force-set or force-reset bit.
Set - Cancel All Force		Clears all force-set and force-reset bits.
Set - Value		Changes the PV of the selected variable or physical address.
Set - Timer/Counter Setting Value		Changes the PV of the timer or counter.
Differential Monitor		Executes differential monitoring.
Error Log		Displays the PLC error log.
Change Log - Enable Mode		Enables the change log.
Change Log - Disable Mode		Disables the change log.
Change Log - Change Log List		Reads the change log list.
Cycle Time		Displays the cycle time.
Data Trace		Used to execute a data trace.
Variable Reference		Displays a list showing the usage of variables.
Set Clock		Used to set the clock in the CPU Unit.

10-4-11 Library Menu

Menu	Shortcut	Function
POU - Register to Library		Registers a POU in the library.
POU - Add to Project		Adds a POU registered in the library to a project.
Rung - Register to Library		Registers a rung group in the library.
Rung - Insert to Program		Adds a rung group registered in the library to a program.
Map Folder		Changes the library folder allocated in the computer.
Unmap Folder		Clears the library folder allocation.
Create Folder		Creates a folder.
Property		Displays information on the library.
Update		Redisplays the library folder tree.
Edit		Used to edit library properties.
Delete		Deletes a library object or folder.
Option		Sets the operation used for the library function.

10-4-12 Tool Menu

Menu	Shortcut	Function
Select Interface - NE1S Serial Port / DeviceNet I/F/ Ethernet I/F/ RSLinx I/F		Used to select the communications interface. • NE1S Serial Port: USB port • DeviceNet Interface: DeviceNet • Ethernet Interface: Ethernet port • RSLinx interface: RSLinx port
Key Customize		Used to change shortcut key allocations.
Option		Sets various options for NE Programmer displays and operations.

10-4-13 Window Menu

Menu	Shortcut	Function
Next Docked Window	Alt + 0	Switches to the next window as the active window.
Toggle Split Window	F6	Switches between the Variable Editor and Program Editor.
Close All		Closes all windows.
Cascade		Cascades all open windows.

Menu	Shortcut	Function
Tile Horizontally		Tiles all open windows vertically.
Tile Vertically		Tiles all open windows horizontally.
Arrange Icons		Arranges the icons for minimized windows.

10-4-14 Help Menu

Menu	Shortcut	Function
Topics	Alt + 0	Displays the NE Programmer help function.
Instruction Reference		Displays the NE1S instruction reference.
About		Displays information on the NE Programmer version.

10-5 Shortcut Keys

The following tables list the NE Programmer’s shortcut key operations.

10-5-1 Window/View Operations

Shortcut keys	Menu name	Function
Ctrl + PageUp/ PageDown	---	Switches the Tab Page.
Shift + F10	---	Displays the popup menu at the cursor position (same as right-clicking).
Ctrl + F4	---	Closes the program window.
F6	Window - Toggle Split Window	Switches between the Variable Editor and Program Editor.
Ctrl + Tab	Window - *(open window name)*	Switches to an open program window.
Alt + 0	Window - Next Docked Window	Switches to the next window as the active window.
Alt + 1	View - Window - Workspace	Displays/hides the Workspace Window.
Alt + 2	View - Window - Output	Displays/hides the Output Window.
Alt + 3	View - Window - Watch	Displays/hides the Watch Window.
Alt + 4	View - Window - Cross Reference	Displays/hides the Cross Reference Window.
Alt + 5	View - Window - Library	Displays/hides the Library Window.
Alt + 6	View - Window - Property	Displays/hides the Property Window.
Alt + 7	View - Window - Outline	Displays/hides the Outline Window.

10-5-2 Window/View Operations

Shortcut keys	Menu name	Function
Ctrl + N	File - New - Project	Creates a new project.
Ctrl + O	File -Open	Opens an existing project.
Ctrl + S	File - Save	Saves the project being edited (overwrites current project data).
Ctrl + Shift + S	File - Save Changes to project	Updates the project for the program or function block that is being edited.
Alt + G	File - Configuration - Edit Global Variables	Edits global variables.
Ctrl + Shift + R	File - Configuration -Replace physical addresses in programs	Replaces physical addresses in the program with global variables specified by that physical address.
Ctrl + P	File - Print	Prints the specified items.

10-5-3 Edit Operations

Shortcut keys	Menu name	Function
Ctrl + Z	Edit - Undo	Undoes the previously performed operation.
Ctrl + Y	Edit - Redo	Redoes the operation that was undone with *Undo*.
Ctrl + X	Edit - Cut	Cuts the specified range of data.
Ctrl + C	Edit - Copy	Copies the specified range of data.
Ctrl + V	Edit - Paste	Pastes the contents of the clipboard.
Del	Edit - Delete	Deletes the specified range of data.
F2	Edit - Rename	Used to change the name of a project, function block, program, or configuration.
Ctrl + A	Edit - Select All	Selects all data in the selected window.
Ctrl + F	Edit - Find	Searches for a text string.
F3	Edit - Find Next	Finds the next instance of a text string.
Shift + F3	Edit - Find Prev	Finds the previous instance of a text string.
Ctrl + H	Edit - Replace	Replaces a text string.
Ctrl + B	Edit - Bookmark	Edits a bookmark.
Ctrl + Shift + F	Edit - Find in Programs	Searches for the specified item.
Ctrl + G	Edit - Jump - Jump Edit - Jump - Go To Link Edit - Jump - Jump	• In Ladder Editor, jumps to a specified step number or rung number. • In ST Editor, adds the selected text as a variable. • In Mnemonic Editor, inverts the specified instruction.
Tab	Edit - Jump - Bit Address Reference	In Ladder Editor, searches for corresponding contacts or coils associated with the selected operand. If a contact is specified, a search is made backward for a coil with the same bit address. If a coil is specified, a search is made foreword for a contact with the same bit address.
Shift + Alt + N	Edit - Jump - Next Operand Reference	In Ladder Editor, jumps to the next operand with the same address/variable.
Shift + Alt + I	Edit - Jump - Next Input	In Ladder Editor, jumps to the next input with the same address/variable.
Shift + Alt + O	Edit - Jump - Next Output	In Ladder Editor, jumps to the next output with the same address/variable.
Shift + Alt + B	Edit - Jump - Previous Jump Point	In Ladder Editor, returns to the position before the jump.
F8	Edit - Edit Variable	Used to edit the selected instruction operand's variable.
Ctrl + Shift + C	Edit - Copy Operand	Used to copy the operand only.

10-5-4 Offline/Programming Operations

Shortcut keys	Menu name	Function
Ctrl + R	Ladder - Rung - Select ST - Add Variable Mnemonic - Invert	• In Ladder Editor, selects a rung. • In ST Editor, adds the selected text as a variable. • In Mnemonic Editor, inverts the specified instruction.
Ctrl + I	Ladder - Rung - Insert Row Above	Inserts an open row above the cursor position.
Ctrl + Shift + I	Ladder - Rung - Insert Row Below	Inserts an open row below the cursor position.
Ctrl + D	Ladder - Rung - Delete Row	Deletes the selected row.
Ctrl + L	Ladder - Draw Line	Switches the editing mode to Draw Line Mode.
Ctrl + Shift + L	Ladder - Erase Line	Switches the editing mode to Erase Line Mode.
Ctrl + →/ ←	---	In a ladder program, draws a line with the cursor position as a starting point.
Ctrl + Shift + →/ ←	---	In a ladder program, deletes a line with the cursor position as a starting point.
/	Ladder - Invert (NOT)	Inverts the specified instruction.

Shortcut keys	Menu name	Function
Ctrl + E	Ladder - Online Edit - Begin	Starts online editing.
Ctrl + U	Ladder - Online Edit - Cancel	Cancels online editing.
Ctrl + Shift + E	Ladder - Online Edit - Finish	Ends online editing.

10-5-5 Variable Operations

Shortcut keys	Menu name	Function
Ctrl + G	Variable - Group Input/Output Variables - Group	In Variable Editor or Global Variable Editor, groups the selected I/O variables.
Ctrl + Shift + G	Variable - Group Input/Output Variables - Release Group	In Variable Editor or Global Variable Editor, ungroups all of the I/O variables in the group.
Ctrl + Shift + M	Variable - Group Input/Output Variables - Release Member	In Variable Editor or Global Variable Editor, removes the selected member from the group.
Ctrl + Up	Variable - Up	Shifts the function block's selected input variable or output variable up one position in the Variable Editor.
Ctrl + Down	Variable - Down	Shifts the function block's selected input variable or output variable down one position in the Variable Editor.

10-5-6 Build Operations

Shortcut keys	Menu name	Function
Ctrl + F7	Build - Compile	Compiles the active program or function block.
F7	Build - Build	Builds the project.

10-5-7 Online/Controller Operations

Shortcut keys	Menu name	Function
Ctrl + W	Controller - Connect	Connects to the Controller.
Ctrl + Shift + W	Controller - Disconnect	Disconnects from the Controller.
Ctrl + Shift + A	Controller - Auto upload from Controller	Automatically uploads from the connected Controller.
Ctrl + Shift + T	Controller - Upload from Controller	Uploads from a Controller.
Ctrl + T	Controller - Download to Controller	Downloads to a Controller.
Ctrl + 1	Controller - Operating Mode - Program	Switches the Controller to PROGRAM mode.
Ctrl + 3	Controller - Operating Mode - Monitor	Switches the Controller to MONITOR mode.
Ctrl + 4	Controller - Operating Mode - Run	Switches the Controller to RUN mode.
F4	Controller - Clear Error	Clears errors.
Ctrl + J	Controller - Set - Force On	Force-sets (forces ON) a bit.
Ctrl + K	Controller - Set - Force Off	Force-resets (forces OFF) a bit.

10-6 Option Settings

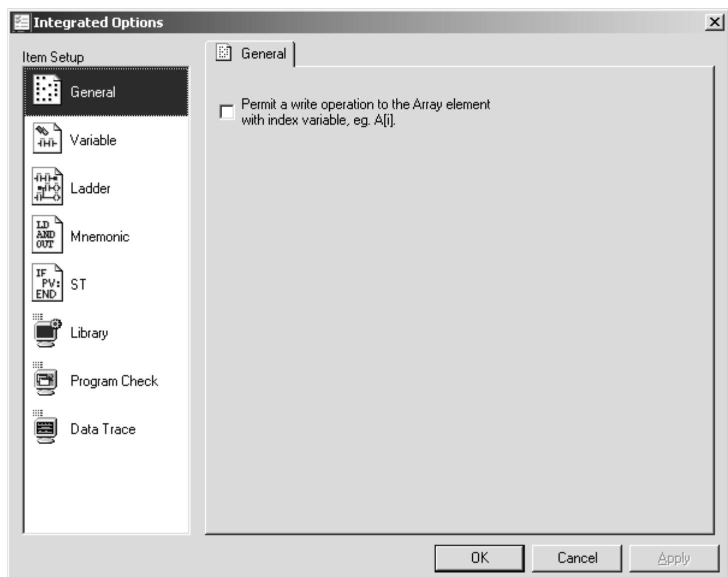
This menu contains various settings for NE Programmer displays and operations.

Select ***Tool - Option***. The following Integrated Options Dialog Box will be displayed.

Item	Description
General	Specifies whether to permit array variables to be written with an index variable, such as A[i].
Variable	Specifies which variable properties will be displayed.

Item	Description
Ladder	Specifies whether variable names, physical addresses, and comments will be displayed/hidden in the Ladder Editor, and how many lines will be displayed. Displays/hides grid lines, sets fonts, cell widths, and colors for items in the Ladder Editor. Specifies whether instance names will be generated automatically.
Mnemonic	Specifies whether values, comments, addresses, and data types will be displayed/hidden in the Mnemonic Editor.
ST	Sets the font properties and colors of items in the ST Editor.
Library	Specifies whether to edit local variables when rung elements are inserted. Also specifies whether prefixes are added to global variables when using the library's POUs.
Program Check	Sets the program check level.
Data Trace	Sets the colors of items in the data trace display.

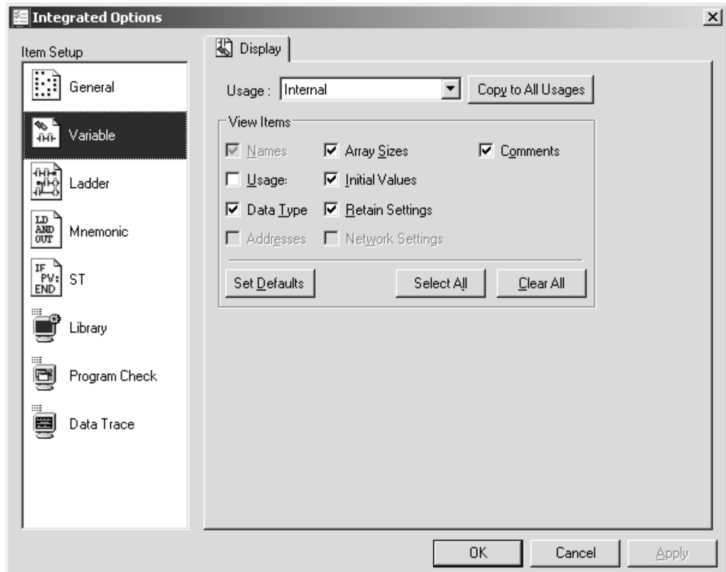
10-6-1 General Window



- Permit a write operation to the Array element with index variable, e.g., A[i]. Select this option to allow write operations (PV change, force-set/reset, and set/reset operations) on array variables specified with an index, such as A[i].

Note An array variable's index value is based on the index value the last time that the value was monitored, so a write operation may operate on a different array element if the variable's index value was changed since the last write operation.

10-6-2 Variable Window



Usage

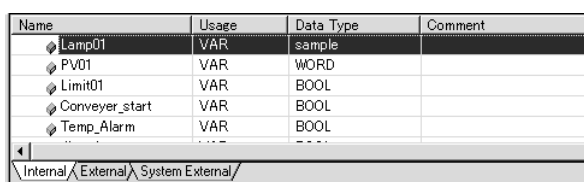
Select the variable type: *Internal*, *Global*, *System*, *Input*, *Output*, *External*, or *System External*.

View Items

Select the variable properties to be displayed in the Variable Editor Window. (Different properties can be set for each variable type.)

> *Usages*, *Data Types*, *Addresses*, *Array Sizes*, *Initial Values*, *Retain Settings*, *Network Settings*, and *Comments*

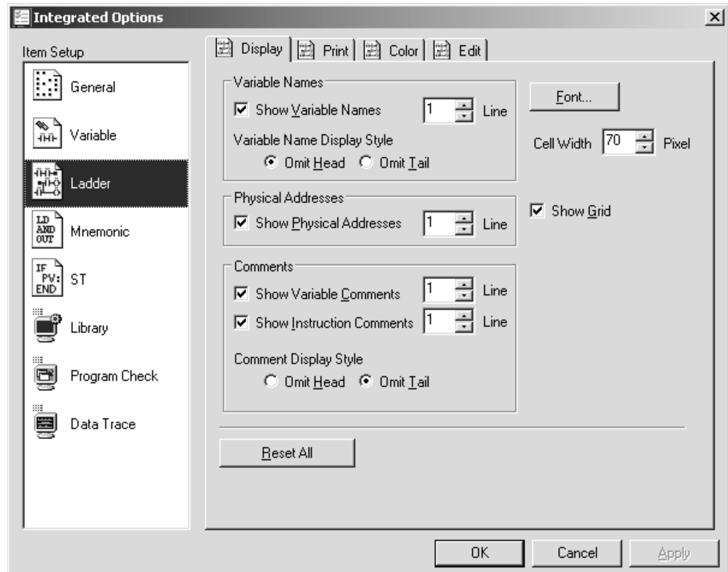
In the following example, the *Usages*, *Data Types* and *Comments* Options have been selected.



Name	Usage	Data Type	Comment
Lamp01	VAR	sample	
PV01	VAR	WORD	
Limit01	VAR	BOOL	
Conveyer_start	VAR	BOOL	
Temp_Alarm	VAR	BOOL	

10-6-3 Ladder Window

Display Tab Page



Variables

Show Variable Name:
Select this option to display the variable names. The style of the variable name display can be selected.

Variable Name Display Style:

- Omit Head
  If all of the variable name cannot be displayed, select this option to display only the last part of the variable name.
- Omit Tail
  If all of the variable name cannot be displayed, select this option to display only the first part of the variable name.

Physical Addresses

Show Physical Addresses:
Select this option to display the physical addresses. The number of lines of physical addresses can be set.

Comments

Show Variable Comments:
Select this option to display the variable comments. The number of lines of variable comments can be set.

Show Instruction Comments:
Select this option to display the instruction comments. The number of lines of instruction comments can be set.

Comment Display Style:

- Omit Head
  If all of the comment cannot be displayed, select this option to display only the last part of the comment.
- Omit Tail
  If all of the comment cannot be displayed, select this option to display only the first part of the comment.

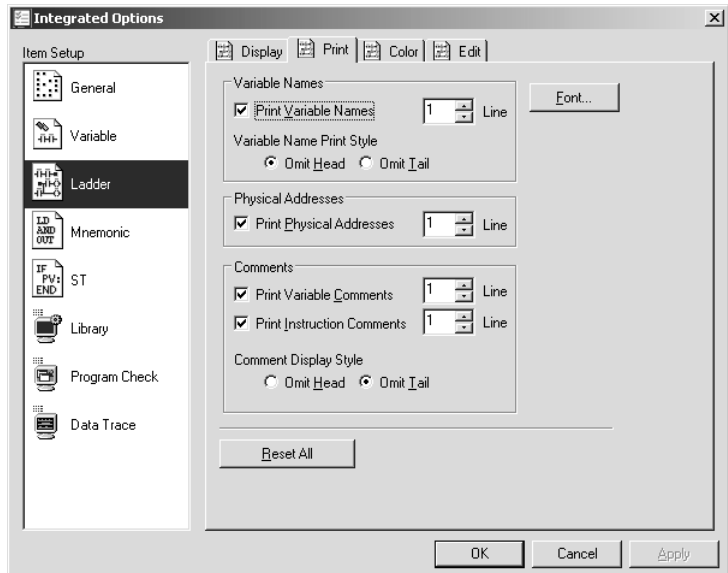
Font

Click the **Font** Button to set the font used in the Ladder Editor.

Cell Width: Sets the width of cells in the Ladder Editor in pixels.

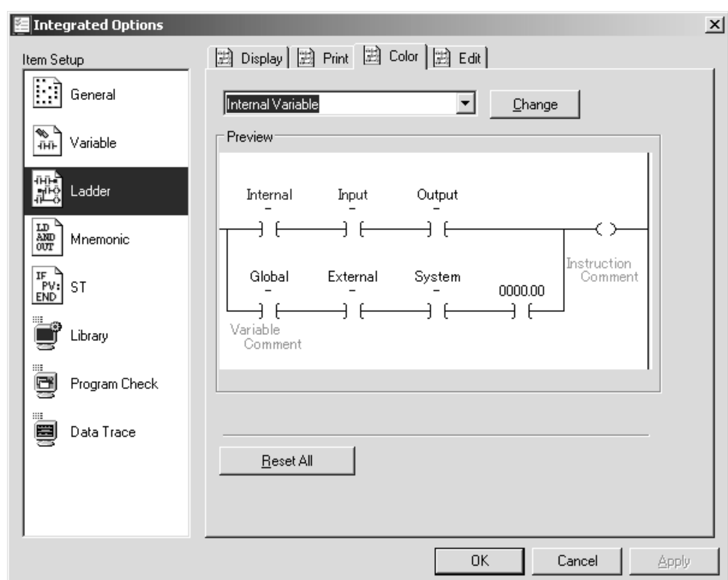
Show Grid: Remove the check in this option to hide the Ladder Editor's grid.

Print Tab Page



Contains the same settings as the Display Tab Page.

Color Tab Page



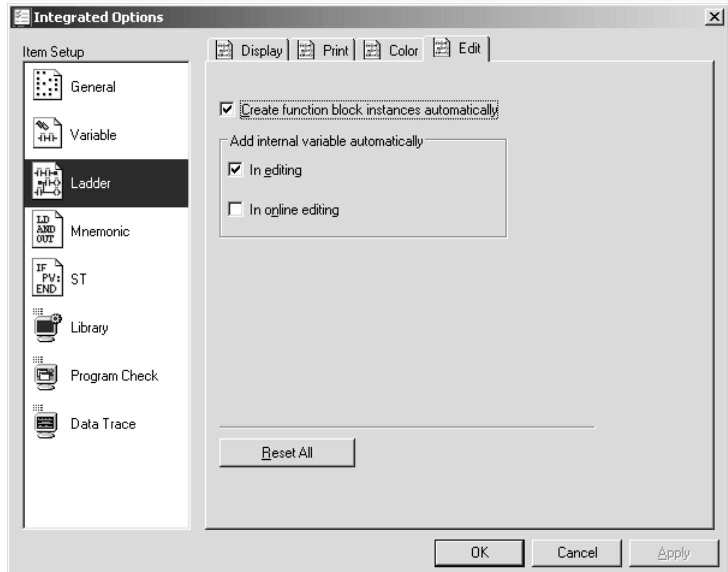
The colors can be selected for the following Ladder Editor items. The *Preview* Area displays an example of the selected colors.

> *Internal Variable*, *Input Variable*, *Output Variable*, *Global Variable*, *External Variable*, *System Variable*, *Variable Comment*, etc.

Change

Click the **Change** Button to set the color. The Set Color Dialog Box will be displayed. Set the colors for the selected ST Editor items.

Edit Tab Page



Create function block instances automatically

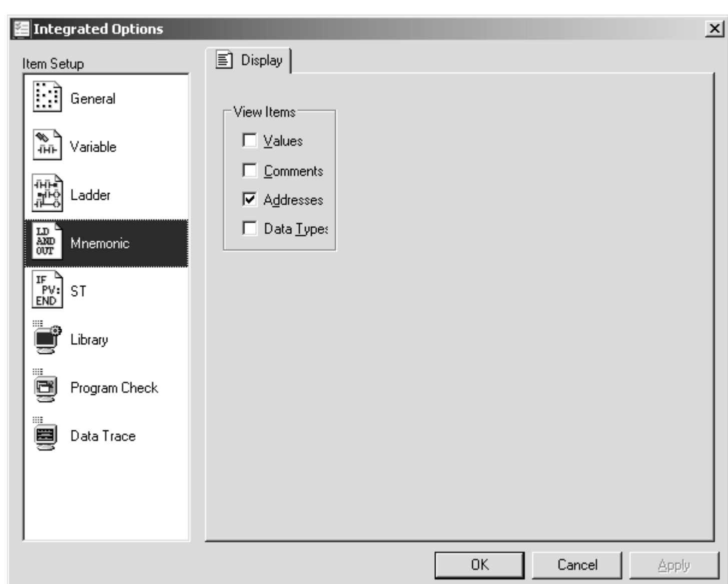
Select this option to automatically generate the instance name when a function block is pasted in the program. (The file name is the function block name with a consecutive number attached.)

Add internal variable automatically

In editing:
If this option is selected, variables in function block instances are automatically added as internal variables when editing.

In online editing:
If this option is selected, variables in function block instances are automatically added as internal variables when online editing.

10-6-4 Mnemonic Window



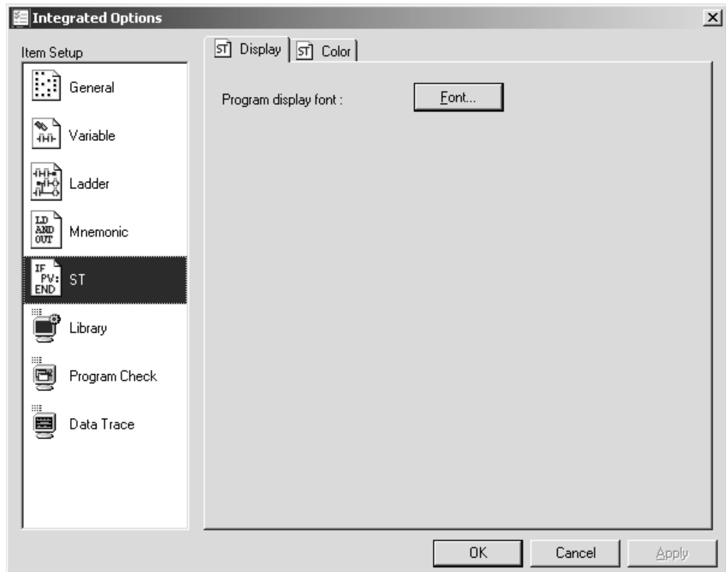
Display Tab Page

Select the following items to be displayed in the Mnemonic Editor.

Values, *Comments*, *Addresses*, and *Data Type*

10-6-5 ST Window

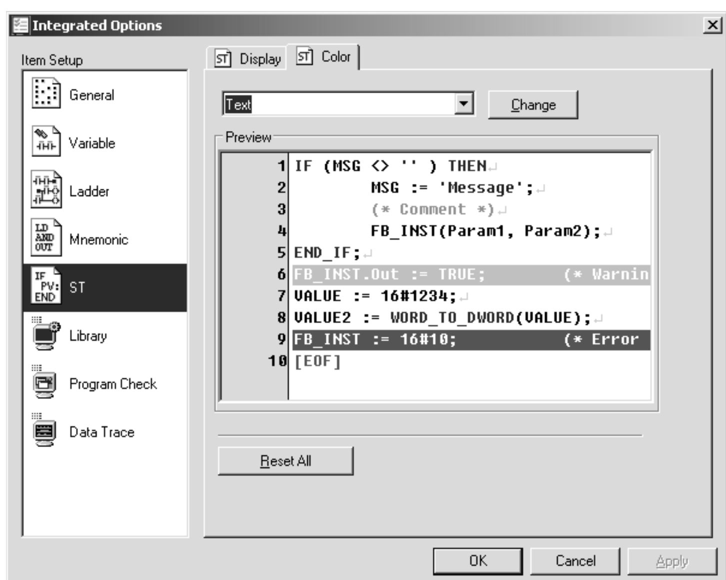
Display Tab Page



Font

Click the **Font** Button to set the font used in the ST Editor.

Color Tab Page



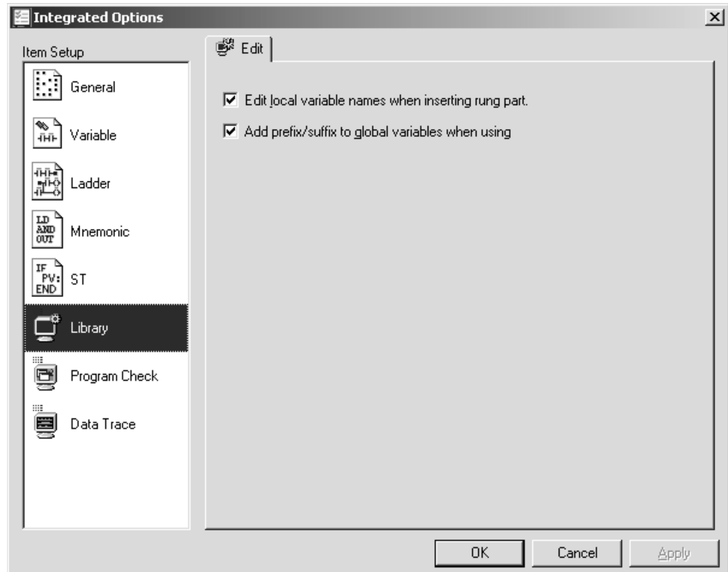
The colors can be selected for the following ST Editor items. The *Preview* Area displays an example of the selected colors.

Text, *Background*, *Error Line*, *Warning Line*, *Warning Line Text*, *Comment*, *Keyword*, *Fixed String*, *Line Feed*, *End of File*

Change

Click the **Change** Button to set the color. The Set Color Dialog Box will be displayed. Set the colors for the selected ST Editor items.

10-6-6 Library Window



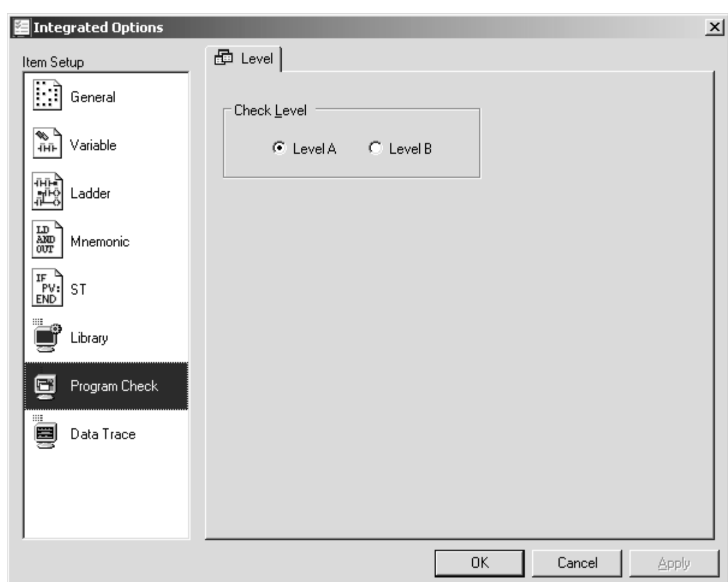
Edit local variable names when inserting rung part

When this option is selected, the Edit Local Variables Dialog Box will be displayed when one of the library's rung elements is dragged and dropped into the program.

Add prefix/suffix to global variables when using

When this option is selected, a prefix will be added to all global variables in a POU when one of the library's POUs (program or function block) is added to a project.

10-6-7 Program Check Window



Check Level

The program check level can be set to *Level A* (default setting) or *Level B*. If level A is selected, a stricter program check will be performed.

Level A

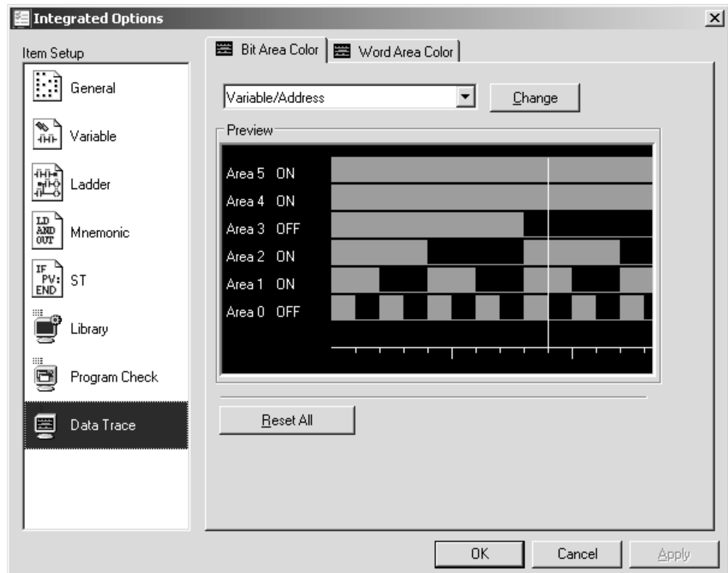
Check item	Normal POUs		FB POUs	
	Compiling	Building	Compiling	Building
Duplicated outputs	Yes	Yes	Yes	Yes
Duplicated timer numbers	Yes	Yes	Yes	Yes
Duplicated counter numbers	Yes	Yes	Yes	Yes
Duplicated FAL numbers	Yes	Yes	Yes	Yes
Duplicated variables	Yes	Yes	Yes	Yes
Check within specific areas	Yes	Yes	Yes	Yes
Check outside of specific areas	Yes	Yes	Yes	Yes
Array indices	Yes	Yes	Yes	Yes
Automatic allocation area duplication	No	Yes	No	Yes
END instruction	Yes	Yes	Yes	Yes
FOR-NEXT check	Yes	Yes	Yes	Yes
IL-ILC check	Yes	Yes	Yes	Yes
Jump 0 check	Yes	Yes	Yes	Yes
Jump check	Yes	Yes	Yes	Yes
Intertask check	No	Yes	No	No

Level B

Check item	Normal POUs		FB POUs	
	Compiling	Building	Compiling	Building
Duplicated outputs	No	No	No	No
Duplicated timer numbers	No	No	No	No
Duplicated counter numbers	No	No	No	No
Duplicated FAL numbers	No	No	No	No
Duplicated variables	No	No	No	No
Check within specific areas	Yes	Yes	Yes	Yes
Check outside of specific areas	Yes	Yes	Yes	Yes
Array indices	Yes	Yes	Yes	Yes
Automatic allocation area duplication	No	Yes	No	Yes
END instruction	Yes	Yes	Yes	Yes
FOR-NEXT check	Yes	Yes	Yes	Yes
IL-ILC check	Yes	Yes	Yes	Yes
Jump 0 check	Yes	Yes	Yes	Yes
Jump check	Yes	Yes	Yes	Yes
Intertask check	No	Yes	No	No

10-6-8 Data Trace Window

Bit Area Colors Tab Page



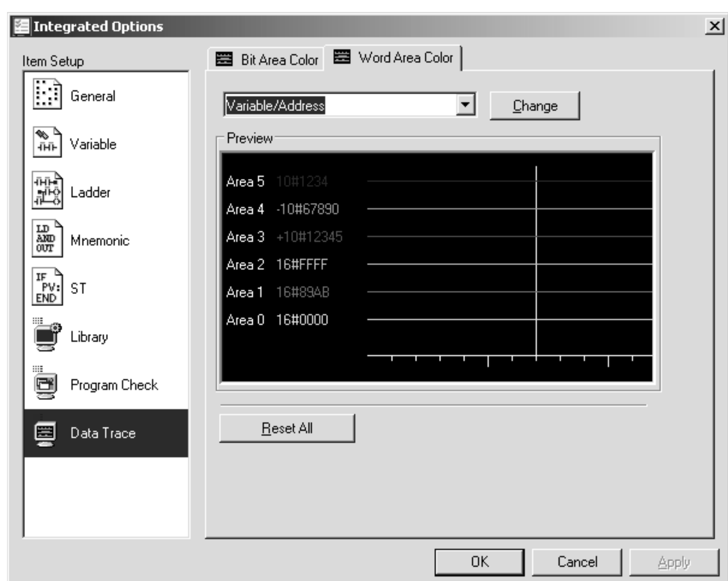
The colors can be selected for the following items in the data trace bit area. The *Preview* Area displays an example of the selected colors.

Variable/Address, *Value*, *Background*, *Grid Line*, *Graph*

Change

Click the **Change** Button to set the color. The Set Color Dialog Box will be displayed. Set the colors for the selected data trace bit area items.

Word Area Colors Tab Page



The colors can be selected for the following items in the data trace word area. The *Preview* Area displays an example of the selected colors.

Variable/Address, *Area 1*, *Area 2*, *Area 3*, *Area 4*, *Area 5*, *Area 6*, *Background*, *Grid Line*

Change

Click the **Change** Button to set the color. The Set Color Dialog Box will be displayed. Set the colors for the selected data trace word area items.

SECTION 11
Programming

This section provides details on programming.

11-1 Overview

11-1-1 Basic Flow of Programming

Step	Operation	Reference
Create a project.	File - New	*11-2-1*
Create logical POUs (programs or function blocks).	File - New - POU	*11-2-2*
Programming in ladder diagrams or standard text and registering/editing variables with local variable editor.		*11-3*
Saving edited data in project.	File - Save Changes to project	*11-2-3*
Creating function blocks and pasting them into programs.	Creating function blocks and then dragging and dropping them into programs.	*11-4*
Creating a configuration.	File - New - Configuration	*11-5-1*
Editing global variables.		*11-5-2*
Pasting programs into tasks.	Dragging and dropping programs into task folders.	*11-5-3*
Saving the project.	File - Save As	*11-2-4*

See note.

Note The configuration can be created and the global variables can be edited before the logical POUs (programs and function blocks) are created.

11-2 Creating Projects and Logical POUs

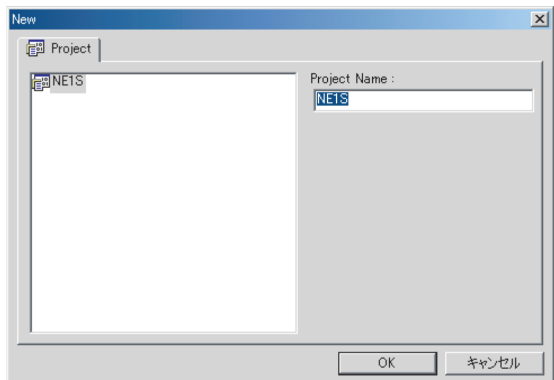
Before actual programming, a project and logical POUs must be created.

11-2-1 Creating Projects

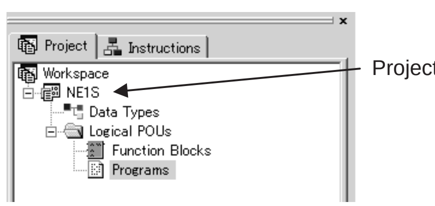
Use the following procedure to create a project.

1,2,3...

1. 1 Right-click the Workspace in the Project Window and select ***Create Project***. Alternately, press the **Ctrl+N** Keys or select ***File - New - Project***. The following New Dialog Box will be displayed.



2. If the name *NE1S* is acceptable, click the **OK** Button. To change the project, input the project name and then click the **OK** Button.
   The following will be displayed in the Project Window.



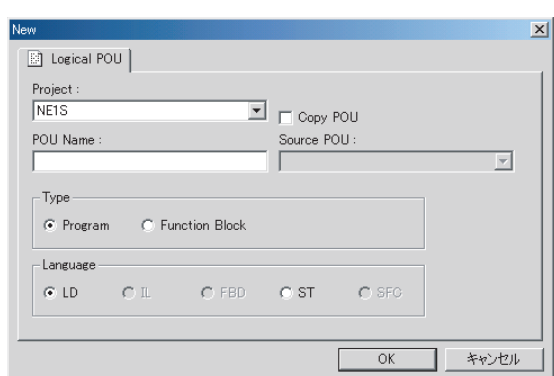
This completes creating a project.

11-2-2 Creating Logical POUs (Programs or Function Blocks)

Use the following procedure to create logical POUs (programs or function blocks).

1,2,3...

1. Right-click the ***Logical POUs*** in the Project Window and select ***Create POU***. Alternately, select ***File - New - POU***.
   The following New Dialog Box will be displayed.

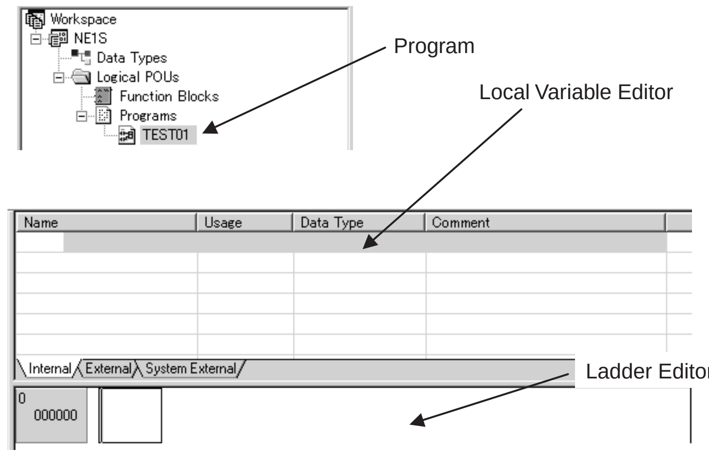


2. Input the program name in the *POU Name* field.

3. For *Type*, select ***Program***.

   Note If *Function Block* is selected a new function block will be created.

4. For *Language*, select the desired language. Select ***LD*** to program a ladder diagram or ***ST*** to program in standard text.
5. Click the **OK** Button.
   The logical POUs that are created will be displayed under *Logical POUs* in the Project Window. Also, a Ladder Editor and Local Variable Editor will be displayed for the program that was created.



   This completes creating a program.

6. 6 Finally, code the program.
   Refer to *11-3* for information on programming methods.

Note (1) By default, the Local Variable Editor will be display connected to the top of the Ladder Editor as shown above.

(2) To hide the Local Variable Editor, select ***View - Variable Editor - Visible***.

(3) To display the Local Variable Editor, select ***View - Variable Editor - Visible*** again.

11-2-3 Saving Edited Data in the Project

After coding programs or function blocks, be sure to save any data that was edited in the program.

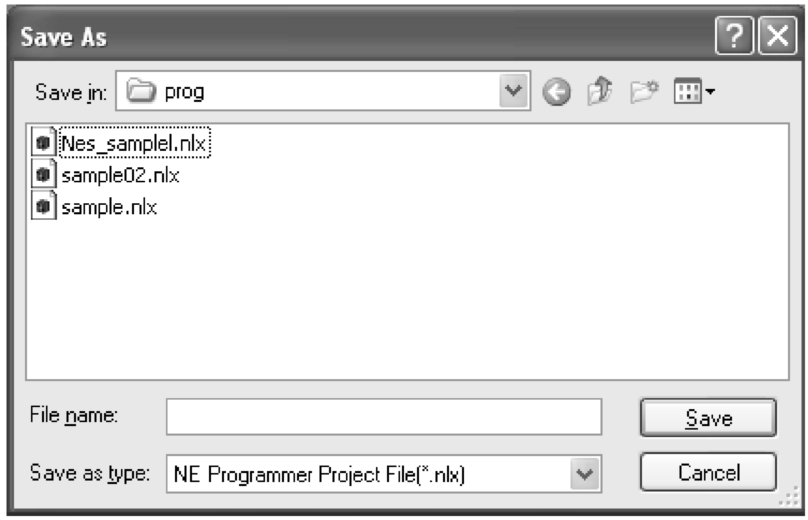
To save the data, select ***File - Save Changes to project***. Alternately, press the **Ctrl+Shift+S** Keys.

An alarm will be displayed if there is an error in the program. Correct the error and then select ***File - Save Changes to project*** again.

11-2-4 Saving the Project

Use the following procedure to save the project.

1,2,3... 1. Select ***File - Save*** or ***File - Save As***.
The Save As Dialog Box will be displayed.



2. Specify the file location and file name to save in and then click the **Save** Button.
The project will be saved with an .nlx filename extension.

11-3 Programming Methods

This section describes how to program using ladder diagrams or standard text, and how to edit local variables.

11-3-1 Ladder Diagrams

This section describes basic ladder programming methods. Refer to *SECTION 2* for details on variables and methods for inputting constants and operand numbers.

Note (1) The default input mode is the insert mode. Press the **Insert** Key to change to overwrite mode. The mode will switch between overwrite and insert mode every time the Insert Key is pressed. Either INS or OVR will be displayed in the status bar to indicate the input mode.

(2) After programming, always select ***File - Save Changes to project***. Alternately, press the **Ctrl+Shift+S** Keys.
An alarm will be displayed if there is an error in the program. Correct the error and then select ***File - Save Change to project*** again. Alternately, press the **Ctrl+Shift+S** Keys.

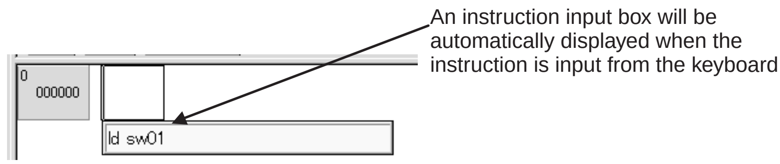
(3) Always place an END instruction at the end of each program.

Inputting NO Conditions

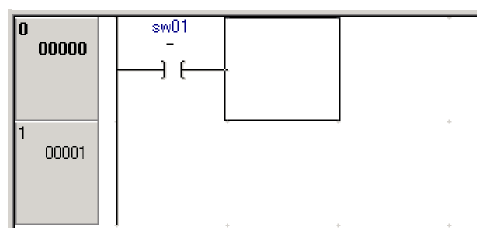
On the Ladder Editor, move the cursor to the cell where the NC condition is to be input.

Input ***ld****(space)(variable_name)* from the keyboard and then press the **Enter** Key. Either ***ld*** or ***LD*** may be input. Inputs for all other instructions are not case-sensitive.
Variable names are also not case-sensitive, even though they are registered with the case that is input. Regardless of the case that is displayed, variables are not case-sensitive. Example: The following all specify the same variable: abc, Abc, and ABC.

Example: ld sw01



When the Enter Key is pressed, the NO condition and variable will be displayed as shown below and the cursor will move to the next cell to the right.



If the variable that was specified is not registered in the Variable Editor, it will automatically be registered as an internal variable in the Local Variable Editor. This is true of all the instructions described below.

Name	Data Type	Array Size	Initial Value	Retain/Nonretain
sw01	BOOL			Nonretain

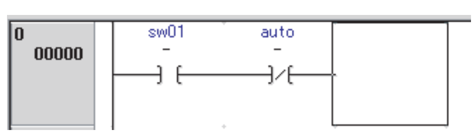
Inputting NC Conditions

On the Ladder Editor, move the cursor to the cell where the NC condition is to be input.

Input ***ld****(space)****not****(space)(variable_name)* from the keyboard and then press the **Enter** Key.

The NC condition and variable will be displayed as shown below and the cursor will move to the next cell to the right.

Example: ld not auto



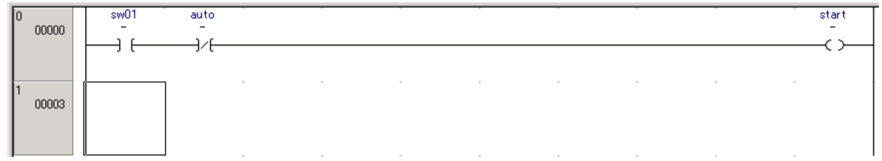
Inputting Outputs (Coils)

Move the cursor to the cell to the right on the input conditions of the line in which to input the output, as indicated by the position of the cursor in the above figure.

Input ***out****(space)(variable_name*) from the keyboard and then press the **Enter** Key. To specify a negated output condition, input as follows:
out*(space)****not****(space)(variable_name)*.

The output and variable will be displayed in the rightmost cell as shown below and the cursor will move to the beginning of the next line.

Example: out start



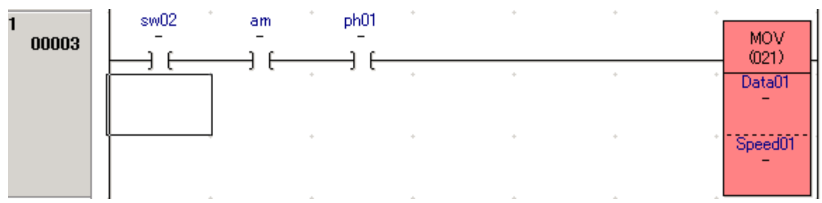
Inputting Special Instructions

Move the cursor to the cell to the right on the input conditions of the line in which to input the instruction.

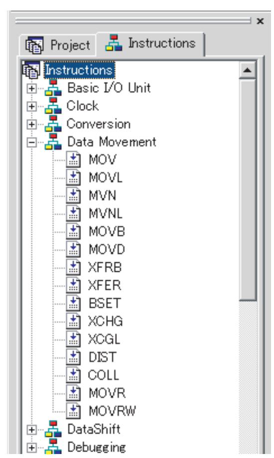
From the keyboard, input *(instruction)(space)(variable_name_of_operand)*..., repeated "*(space)(variable_name_of_operand)*" for each operand required by the instruction, and then press the **Enter** Key.

The instructions and operand variables will be displayed in the rightmost cell as shown below and the cursor will move to the beginning of the next line.

Example: mov DataNo01 Speed01

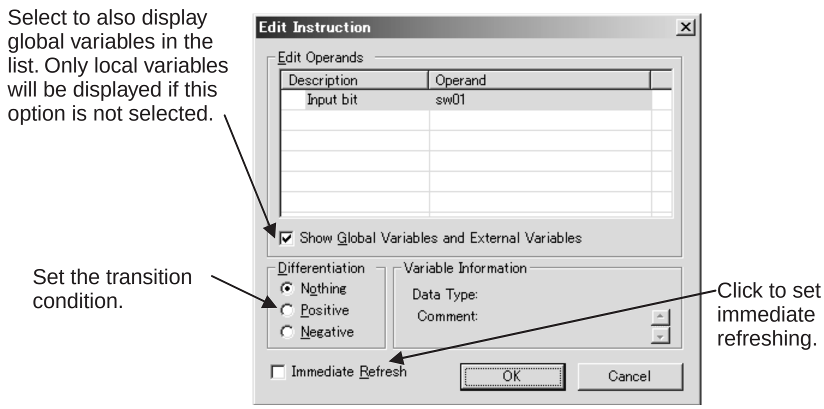


Special instructions can also be added to the program by dragging and dropping them from the Function Window onto the Ladder Editor.

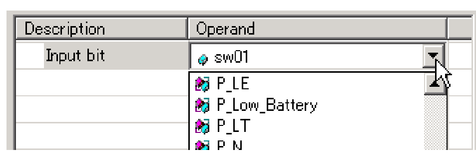


Editing Instructions

1,2,3... 1. Double-click the instruction to be edited. Alternately, right-click the instruction and select ***Edit Instruction***. Alternately, move the cursor to the instruction and press the **Enter** Key.
The Edit Instruction Dialog Box will be displayed.



2. Double-click an operand to edit it. (Alternately, select the operand and press the **Enter** Key.
A list of selections for the operand will be displayed as shown below.



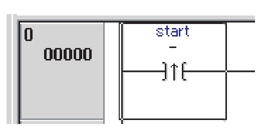
3. Complete editing the instruction and then click the **OK** Button.

Inputting Differentiated Conditions

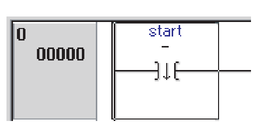
1,2,3... 1. Input the input condition.

2. Right-click the input condition, select ***Transition-Sensing - Positive*** or ***Transition-Sensing - Negative***. Alternately, select ***Positive*** or ***Negative*** as the *Transition-Sensing* Condition in the above dialog box.

- Example of Positive Transition Sensing



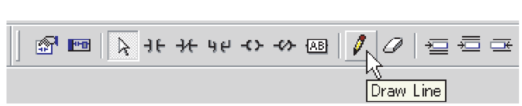
- Example of Negative Transition Sensing



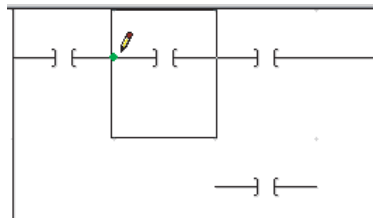
Inputting Vertical and Horizontal Lines: Line Connection Mode

Inputting Lines with the Mouse

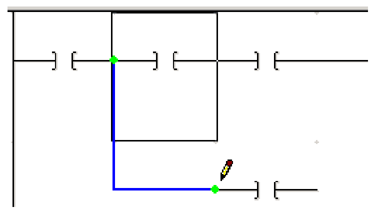
1,2,3... 1. Click the **Draw Line** Icon in the toolbar to enter line connection mode.



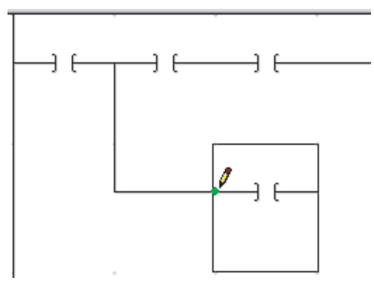
2. Move the cursor to the starting point of the line (indicated as a light green dot) and click the left mouse button to set the starting point.



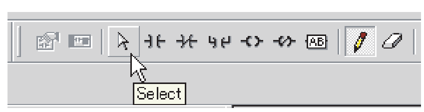
3. Move the cursor to the end point of the line. The line to be created will be indicated in blue.



4. Click the end point of the line.
   The line will be displayed.



To leave line connection mode, click the **Select** Icon.



Inputting Lines from the Keyboard

1,2,3...

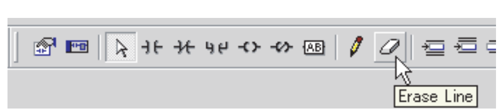
1. Move the cursor to the starting point for the line and then press the **Ctrl+L** Keys.
2. Move the cursor to the end point of the line using the cursor keys. The line to be created will be indicated in blue.
3. Press the **Ctrl+L** Keys at the end point for the line.
   The line will be displayed.

Deleting Vertical and Horizontal Lines: Line Deletion Mode

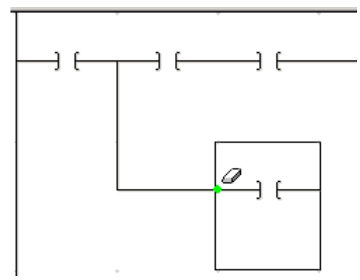
Deleting Lines with the Mouse

1,2,3...

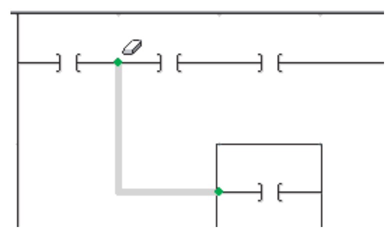
1. Click the **Erase Line** Icon in the toolbar to enter line deletion mode.



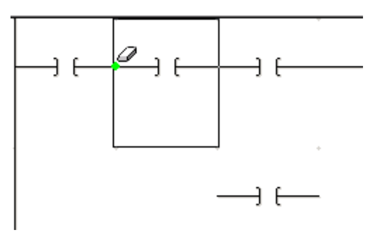
2. Move the cursor to the starting point of the line to be deleted (indicated as a light green dot) and click the left mouse button to set the starting point.



3. Move the cursor to the end point of the line to be deleted. The line to be deleted will be indicated in gray.



4. Click the end point of the line to be deleted.
   The line will be deleted.



   To leave line deletion mode, click the **Select** Icon.

Deleting Lines from the Keyboard

1,2,3...

1. Move the cursor to the starting point for the line and then press the **Ctrl+Shift+L** Keys.
2. Move the cursor to the end point of the line using the cursor keys. The line to be deleted will be indicated in gray.
3. Press the **Ctrl+Shift+L** Keys at the end point for the line.
   The line will be deleted.

Deleting Instructions and Lines

Instructions and lines can be deleted by pressing the **Backspace** or **Delete** Key in the same way as for standard text editors or word processors.

Lines can also be deleted in line deletion mode. Refer to *Deleting Vertical and Horizontal Lines: Line Deletion Mode* for the procedure.

Registering and Editing Global Variables

Registering Variables

When a logical POU (program or function block) is created, a Local Variable Editor will be displayed for each one. Variables can be registered or edited in the Local Variable Editor using either of the following methods.

1,2,3...

1. Inputting Instructions First: As shown above under *Inputting NO Conditions*, if a new variable is specified as an operand when inputting an instruction, the variable will be registered as an internal variable in the Local Variable Editor. If necessary, the parameters of these variables can later be edited in the Local Variable Editor. (Refer to *Editing Variables* later in this section.)
2. Entering Variables in Variable Editor First: The Edit Variables Dialog Box can be displayed by right-clicking in the Local Variable Editor and selecting *Insert* (or alternately by double-clicking the table) to enable registering variables in the Local Variable Editor. Refer to *Editing Variables* later in this section for information on the Edit Variable Dialog Box.
   The variables that were registered can then be input for operands when inputting instructions.

For local variables, physical addresses cannot be input directly in the *Address* column of the Edit Variables Dialog Box.

Editing Variables

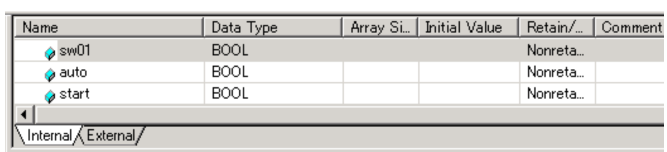
This section describes methods for editing variables and variable parameters.

■ Displaying the Local Variable Editor

If the Local Variable Editor is not displayed, double-click the program name or function block name in the Project Window.

The Local Variable Editor and Ladder Editor will be displayed.

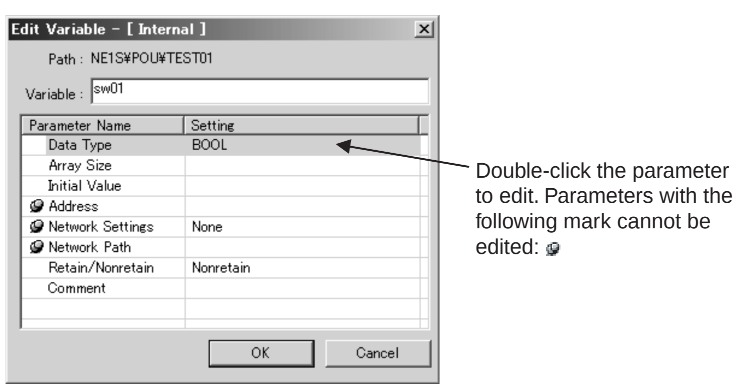
If the Local Variable Editor is still not displayed, select ***View - Variable Editor***.



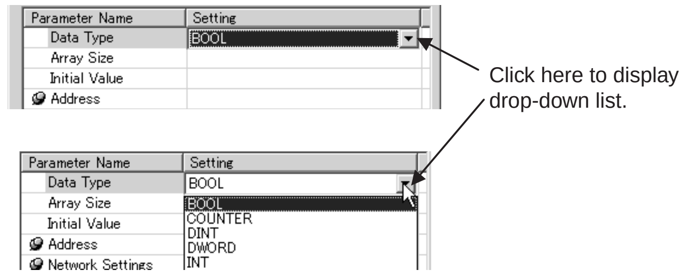
■ Editing Variable Parameters

1,2,3...

1. Double-click the variable to be edited. Alternately, right-click the instruction and select ***Edit***. The Edit Variables Dialog Box will be displayed.



2. To change the name of a variable, input the name in the condition of the window shown above.
   To edit other parameters, double-click the parameter to be edited.
   Editing the parameter will be enabled.
   Example: The following display will appear when *Data Type* is double-clicked:



3. Edit the parameter.
4. When all editing has been completed, click the **OK** Button.
   Parameters are displayed as described in the following table. Refer to *2-2 Variables* for more information on variables.

Parameter	Description	Values
Variable name	Displays the name of the variable being edited. The name of the variable can also be changed.	---
Data Type	Set the data type of the variable.	BOOL, INT, UINT, DINT, UDINT, WORD, DWORD, TIMER, COUNTER, STRING, or REAL For a function block, the name of the logical POU of the function block will be displayed.
Array Size	Specify whether an array is to be used and the number of elements in the array.	Not an array variable: Blank Array variable: Set the number of elements from 1 to 255.
Initial Value	For programs, set the initial value of the variable when program execution is started. For function blocks, set the value of the variable when an instance of the function block is executed.	Set the initial value of the variable according to the data type. • BOOL, WORD, or DWORD: Unsigned hexadecimal Input the number after “16#”. • INT or DINT: Signed decimal Input the number after “+10#” or “–10#”. • UINT or UDINT: Unsigned decimal Input the number after “10#”. • REAL: Real number (e.g., +1.0, –0.23, +9.8E-3)
Address	Not supported for local variables.	---
Network Settings	Not supported.	---
Network Path	Not supported for local variables.	---
Retain/Nonretain	Specify whether to maintain the value of the variable when power is turned OFF and ON, and when operation is started.	Retain or Nonretain
Comment	Input a comment for the variable.	256 ASCII characters max.

Note
(1) When the Cross Reference Window has been displayed (by selecting ***View - Window - Cross Reference***), a variable’s cross reference information (program address, instruction name, program name, etc.) can be displayed just by selecting that variable in the Variable Editor.
(2) Refer to *2-2 Variables* for detailed variable specifications.

■ Inserting Variables

1,2,3...
1. Right-click the line at which to insert a variable and select ***Insert*** from the pop-up menu. Alternately, double-click an empty row.
   The Edit Variables Dialog Box will be displayed.
2. Set the parameters of the Variable and then click the **OK** Button.
   The variable will be inserted.

■ Deleting Variables

1,2,3...
1. Select the variable to be deleted and press the **Delete** Key.
2. A dialog box will appear to confirm the deletion. Click the **Yes** Button.
   The variable will be deleted.

■ Keyboard Operations

The Enter Key can be pressed on the Variable Editor in insert mode to register a new variable. (This is the same as right-clicking and selecting *Add* from the popup menu.)

The Enter Key can be pressed when a variable is selected in overwrite mode to edit the variable. (This is the same as double-clicking.)

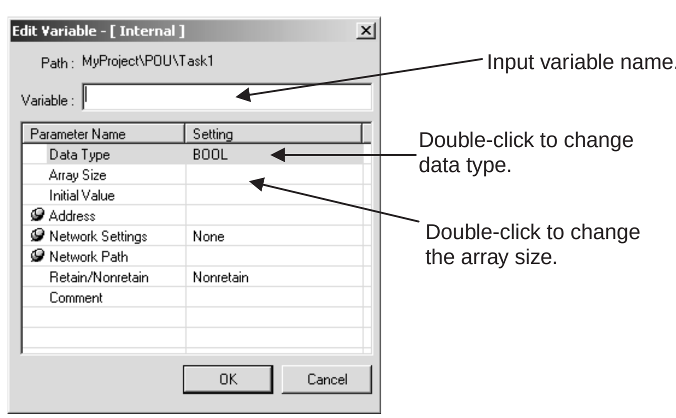
Specifying Arrays

Arrays can be used to handle an array as a group of data elements with the same properties. To create an array variables, set the *Array Size* in the Edit Variables Dialog Box to a value between 1 and 255.

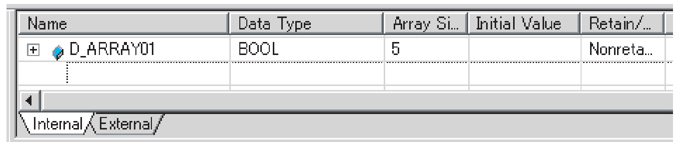
- Arrays can be specified for internal variable (VAR), input variables (VAR_INPUT), output variables (VAR_OUTPUT), or external variables (VAR_EXTERNAL).
- Only one-dimensional arrays can be created.
- When specifying the name of an array variable in a program, the index must be placed in brackets after the variable name (example: ARRAY[0]). Refer to *2-2-5 Array Elements (Array Specification)* for details on using indices.

Use the following procedure to specify an array variable.

1,2,3...
1. Add a variable in the Variable Editor. Right-click the line at which to insert a variable and select ***Insert*** from the pop-up menu. Alternately, double-click an empty row. To edit an existing variable, double-click the variable to be edited.
   The Edit Variables Dialog Box will be displayed.

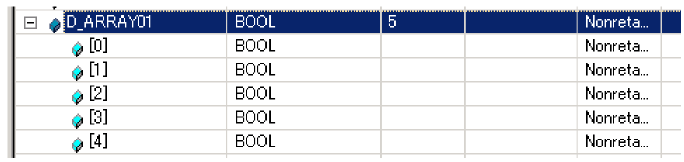


2. Set the *Variable Size* of the Variable to the required number of elements and then click the **OK** Button.
   The variable will be defined as an array and displayed as shown below. In this example, the array variable contains 5 elements.



Name	Data Type	Array Si...	Initial Value	Retain/...
D_ARRAY01	BOOL	5		Nonreta...

   If the + mark is clicked, the array will be expanded on the display as shown below.



D_ARRAY01	BOOL	5		Nonreta...
[0]	BOOL			Nonreta...
[1]	BOOL			Nonreta...
[2]	BOOL			Nonreta...
[3]	BOOL			Nonreta...
[4]	BOOL			Nonreta...

Creating Data Structures

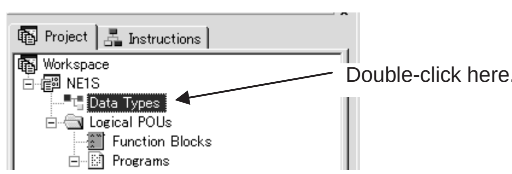
A data structure is a variable consisting of elements with different data types that are treated as a single variable. The user can define data structures as required. The variable name and element names can be specified for specific elements.

Use the following procedure to create a data structure.

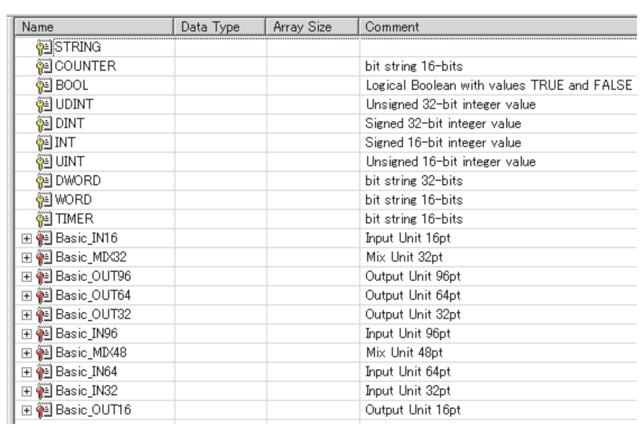
Inserting a Data Structure

1,2,3...

1. Double-click ***Data Types*** in the Workspace.

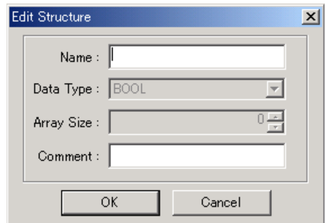


   The Data Structure Table will be displayed. All data structures that are currently registered will be displayed in the Data Structure Table.



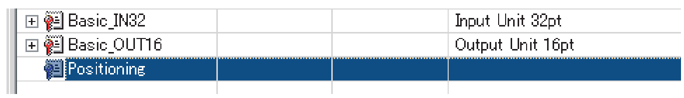
Name	Data Type	Array Size	Comment
STRING			
COUNTER			bit string 16-bits
BOOL			Logical Boolean with values TRUE and FALSE
UDINT			Unsigned 32-bit integer value
DINT			Signed 32-bit integer value
INT			Signed 16-bit integer value
UINT			Unsigned 16-bit integer value
DWORD			bit string 32-bits
WORD			bit string 16-bits
TIMER			bit string 16-bits
Basic_IN16			Input Unit 16pt
Basic_MIX32			Mix Unit 32pt
Basic_OUT96			Output Unit 96pt
Basic_OUT64			Output Unit 64pt
Basic_OUT32			Output Unit 32pt
Basic_IN96			Input Unit 96pt
Basic_MIX48			Mix Unit 48pt
Basic_IN64			Input Unit 64pt
Basic_IN32			Input Unit 32pt
Basic_OUT16			Output Unit 16pt

2. Right-click the last line (where nothing is registered) and select ***Insert - Struct***.
   The Edit Structure Dialog Box will be displayed.



3. Input the name of the structure and any comment that is required and then click the **OK** Button.

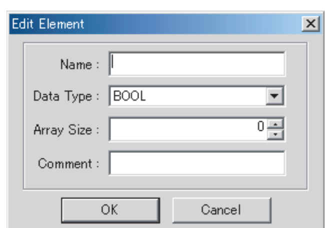
   The data structure will be inserted as a data type and displayed as shown below.



Inserting Elements

1,2,3...

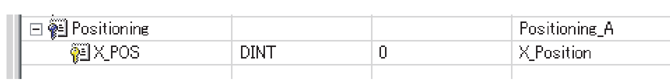
1. Right-click the data structure that was added (*Positioning* in the above example) and select ***Insert - Element***.
   The Edit Element Dialog Box will be displayed.



2. Input the name of the element, the data type, the array size (if required), and any comment that is required and then click the **OK** Button.

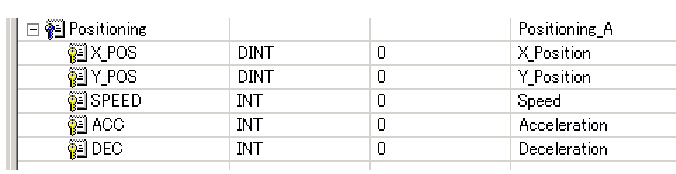
   Note If an array size is specified, the element will be defined as an array inside the data structure.

   The element will be inserted and displayed as shown below.



3. Repeat the above steps to add other elements.

   Setting Example



Grouping Variables

Input variables (VAR_INPUT) or output variables (VAR_OUTPUT) can be grouped for function block to display a group name for the inputs or outputs. Refer to page 309 in *11-4-2* for details.

11-3-2 Toolbar Icons for Ladder Programming

The icons that are displayed on the toolbar for the Ladder Editor are listed below.

Icon	Name	Function
	Edit Instruction	Displays Edit Variables Dialog Box.
	Select Rung	Selects the rung at the cursor.
	Select	Changes to selection mode.
	Open Contact	Inserts a NO condition at the specified position.
	Closed Contact	Inserts a NC condition at the specified position.
	Contact OR	Inserts an OR for a NO condition at the specified position.
	Coil	Inserts an output (coil) at the specified position.
	Negated Coil	Inserts an inverse output (coil) at the specified position.
	Function Block Mode	Used to input function blocks.
	Draw Line	Used to input horizontal and vertical lines.
	Delete Line	Used to delete horizontal and vertical lines.
	Insert Row Above	Inserts a line above the cursor.
	Insert Row Below	Inserts a line below the cursor.
	Delete Row	Deletes a line with nothing in it.
	Insert Column	Inserts a column at the cursor.
	Delete Column	Deletes a column with nothing in it.
	Begin Online Edit	Starts online editing.
	Cancel Online Edit	Cancels editing performed online.
	Finish Online Edit	Ends online editing.

11-3-3 List of Inputs for Instructions

Use the following keys to input execution conditions.

Key	Execution condition
@	Upward differentiation
%	Downward differentiation
!	Immediate refreshing

Use the following inputs to specify instructions.

Spaces are indicated by □.

Instruction	Input
LD	LD□ *variable_name*
OR	OR□ *variable_name*
AND	AND□ *variable_name*
LDNOT	LD□ NOT□ *variable_name*
ORNOT	OR□ NOT□ *variable_name*
ANDNOT	AND□ NOT□ *variable_name*
OUT	OUT□ *variable_name*
OUTNOT	OUT□ NOT□ *variable_name*
!LD	! LD□ *variable_name*
!AND	! AND□ *variable_name*
!OR	! OR□ *variable_name*
!LDNOT	! LD□ NOT□ *variable_name*
!ANDNOT	! AND□ NOT□ *variable_name*
%LD	% LD□ *variable_name*
%AND	% AND□ *variable_name*
%OR	% OR□ *variable_name*
!@LD	! @ LD□ *variable_name*
!@AND	! @ AND□ *variable_name*
!@OR	! @ OR□ *variable_name*
!%LD	! % LD□ *variable_name*
!%AND	! % AND□ *variable_name*
!%OR	! % OR□ *variable_name*
!OUT	! OUT□ *variable_name*
!OUTNOT	! OUT□ NOT□ *variable_name*
TIMER (TIM)	TIM□*timer_number*□10# *set_value* or 16# *set_value*
COUNTER (CNT)	CNT□*counter_number*□10# *set_value* or 16# *set_value*
Special Instructions executed each scan	*instruction*□*operand*□*operand*□*operand*
Upwardly differentiated instructions	@ *instruction*□*operand*□*operand*□*operand*
Downwardly differentiation instructions	% *instruction*□*operand*□*operand*□*operand*
Immediate refresh instructions	! *instruction*□*operand*□*operand*□*operand*

Refer to *2-4* for the methods for inputting constants and operand numbers.

11-3-4 Inputting Bit and Word Addresses

The inputs used to specify bit and word addresses are listed in the following table.

Refer to *2-4* for the methods for inputting constants and operand numbers.

Area	Bit address	Word address
CIO Area	*bit-address*	*word-address*
Work Area (word bits)	W*bit-address*	**W***word-address*
Holding Area	**H***bit-address*	**H***word-address*
Auxiliary Area	**A***bit-address*	**A***word-address*

Area	Bit address	Word address
Timer Area	**T***timer_number*	**T***timer-number*
Counter Area	**C***counter-number*	**C***counter-number*
Task Area	**TK***task-number*	---
DM Area	---	**D***word-address*
EM Area	---	**E***word-address*
Indirect address in DM Area	---	@**D***word-address*
Indirect address in EM Area	---	***E0**_*word-address* @**E0**_*word-address,*
Constants	---	10# +*number*16# +number
Data Register Area	---	**DR***address*
Index Register Area	---	**IR***address*
Indirect address in Index Register Area	---	,IR* ,IR*+ ,IR*++ ,–IR* ,—IR*DR*, IR* XXXX, IR*

11-3-5 Converting Specified Physical Addresses to Variables

If a ladder program was created with physical addresses instead of variable names and global variables for the physical addresses were added later, the physical addresses in the program can all be converted to global variables at once. (In this case, the global variables are automatically added to the external variables.)

Example Application:
This function is useful when a program is created first with physical addresses and then corresponding variable names are later assigned altogether.

Note
(1) If there is a physical address duplication error (build error), the conversion will not be performed because the replacement global variable cannot be determined.
(2) A global variable will not be replaced if there is another kind of variable (local variable in the ladder program or external variable) with the same name as the global variable.

Procedure

1,2,3...

1. Select **File - Configuration - Replace physical addresses in programs** or right-click a *Global Variables* in the Project Window and select **Replace physical addresses in programs** from the popup menu.

   A dialog box will be displayed to confirm that the physical addresses in the program will be replaced by the specified variables. Click the **OK** Button to continue.
2. The Find Tab of the Output Window will show the progress of the conversion (replacing or completed).

   If there were any physical addresses in the program that could not be replaced, they will be listed. To jump to one of those physical address locations, either double-click the address in the list or select the address and press the **Enter** Button.

11-3-6 Programming in Standard Text Language

Character Set

The character set conforms to IEC 61131-3.

- Characters are not case-sensitive.
- The character set conforms to the *Basic Code Table* in ISO 646.
- Keywords can combine uppercase and lowercase letters (e.g., iF or if).

Identifiers

Overview

Identifiers are text strings used to express the following language elements.

- Naming program control units
- Naming I/O and variables
- Naming functions and function blocks

Text Allowed for Identifiers

The same characters and the same number of characters as used for variables in ladder diagram programming can be used for identifiers in ST programming.

Note Refer to *2-2 Variables* for detailed variable specifications.

Restrictions

The following restrictions apply to identifiers.

- The first character must not be a number.
- Two underscores (_) must not be used consecutively.
- Spaces cannot be used.
- In addition to the restrictions on characters and number of characters for variables in ladder programming, reserved words for ST language programming cannot be used. Refer to *ST Language Reserved Words* on page 485 for details on reserved words for ST language programming.

Data Types

Basic Data Types

The basic data types and their sizes are listed in the following table.

Variables are edited with the Variable Editor. Variables can be registered with the ST Editor, but they cannot be changed.

Data type	Meaning	Size	Words allocated	Description
INT	Integer	16 bits	1 word	(–32768 to +32767)
DINT	Double-word integer	32 bits	2 words	(–2147483648 to 2147483647) Word 0: Lower 16 bits Word 1: Upper 16 bits
UINT	Unsigned integer	16 bits	1 word	(0 to 65535)
UDINT	Unsigned double-word integer	32 bits	2 words	(0 to 4294967295) Word 0: Lower 16 bits Word 1: Upper 16 bits
BOOL	Bit string of 1 bit	1 bit	1 bit	1 or 0
WORD	Bit string of 16 bits	16 bits	1 word	
DWORD	Bit string of 32 bits	32 bits	2 words	Word 0: Lower 16 bits Word 1: Upper 16 bits
REAL	Real number	32 bits	2 words	Conforms to IEEE754.
STRING	Text string	64 words	64 words	

Basic Data Types That Are Not Supported

The following basic data types are not supported.

Data type	Meaning
SINT	8-bit integer
USINT	Unsigned 8-bit integer
LINT	64-bit integer
ULINT	Unsigned 64-bit integer
LREAL	Double-word real number
BYTE	Bit string of 8 bits
LWORD	Bit string of 64 bits

Declaring Variables and Data Structures

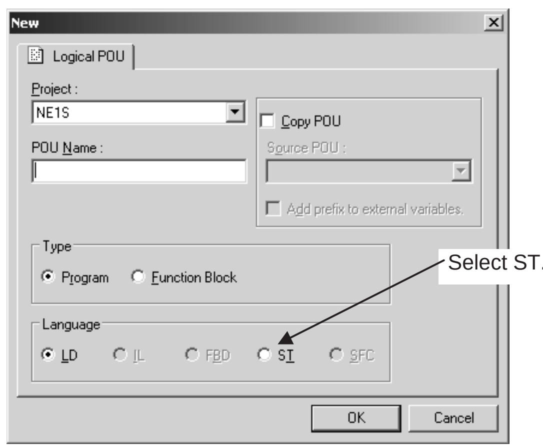
Variables and data structures are edited with the Variable Editor and Structure Editor. Direct address (AT) specifications and variable parameters (e.g., retain/nonretain) are set with the Variable Editor.

Variables and data structures cannot be declared (VAR - END_VAR) in the ST Editor.

Creating an ST Program

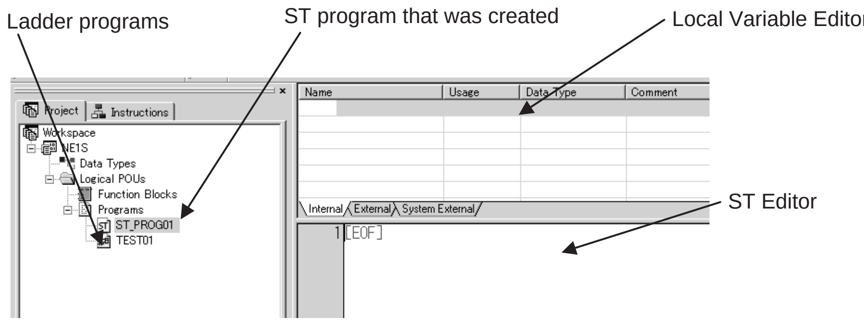
1,2,3...

1. Right-click the ***Logical POUs*** in the Project Window and select ***Create POU***. Alternately, select ***File - New - POU***.
   The following New Dialog Box will be displayed.



2. Input the program name or function name in the *POU Name* field. In this example, *ST_PROG01* is used.
3. For *Type*, select ***Program*** or ***Function Block***. In this example *Program* is selected.
4. For *Language*, select ***ST***.

5. Click the **OK** Button.
   The logical POUs that are created will be displayed under *Logical POUs* in the Project Window. Also, an ST Editor will be displayed for the program or function block that was created.



   This completes creating a program.

6. Finally, code the program.
   Refer to *Appendix B* for operators and conditional statements.
   Refer to page 296 in *11-3-1* for procedures to use the Local Variable Editor.

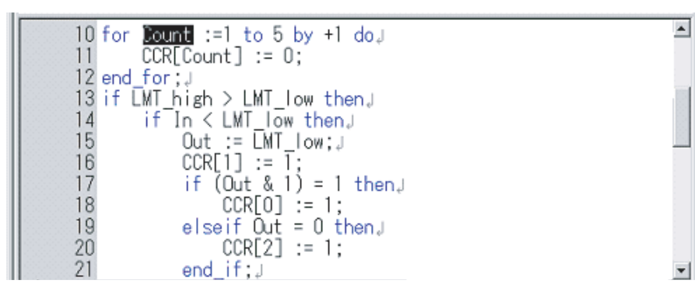
Note After coding a program or function block, always select ***File - Save Changes to project*** to save the changes to the project. Alternately, press the **Ctrl+Shift+S** Keys.

Registering Variables from the ST Editor

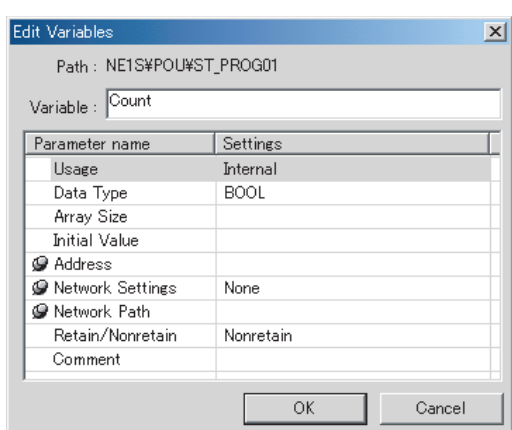
Variables and data structures are edited with the Variable Editor and Structure Editor, but variables can be registered from the ST Editor. Use the following procedure.

1,2,3...

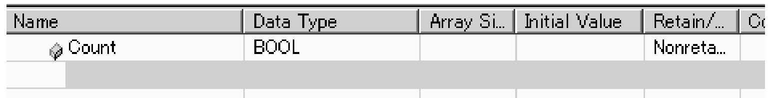
1. Select the variable in the ST Editor as shown below by clicking and dragging.



2. Right-click and select ***Add Variable***.
   The following Edit Variables Dialog Box will be displayed.



3. Set the parameters and then click the **OK** Button.
   The variable will be registered in the variable editor.



   Refer to page 296 in *11-3-1* for procedures to use the Edit Variables Dialog Box and the Variable Editor.

11-4 Creating Function Blocks and Pasting Them into Programs

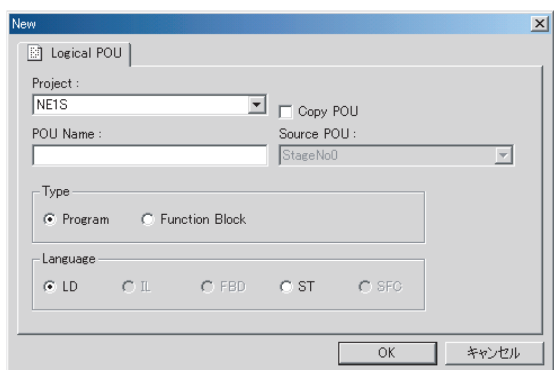
This section describes basic methods for creating function blocks.

Refer to *2-3* for information on the function of function blocks.

11-4-1 Creating Function Blocks

1,2,3...

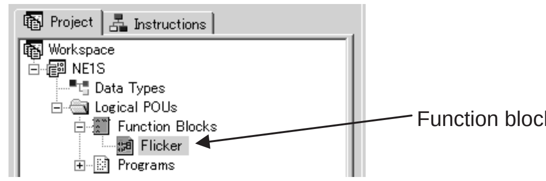
1. Right-click the ***Logical POUs*** in the Project Window and select ***Create POU***. Alternately, select ***File - New - POU***.
   The following New Dialog Box will be displayed.

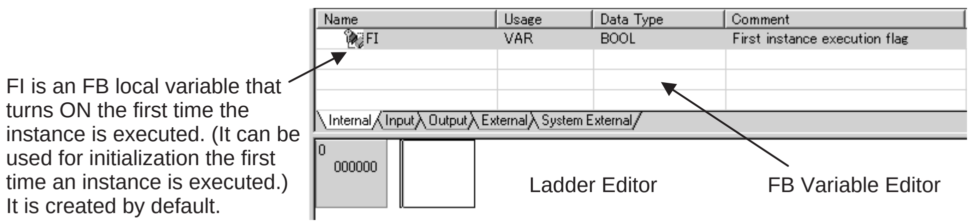


2. Input the function name in the *POU Name* field. In this example, *Flicker* is used.
3. For *Type*, select ***Function Block***.

   Note If *Program* is selected a new program will be created.

4. Click the **OK** Button.
   The function block will be created and displayed in the Workspace as shown below. Also, a Ladder Editor and FB Variable Editor will be displayed for the function block that was created.





This completes creating a function block.

11-4-2 Programming a Function Block

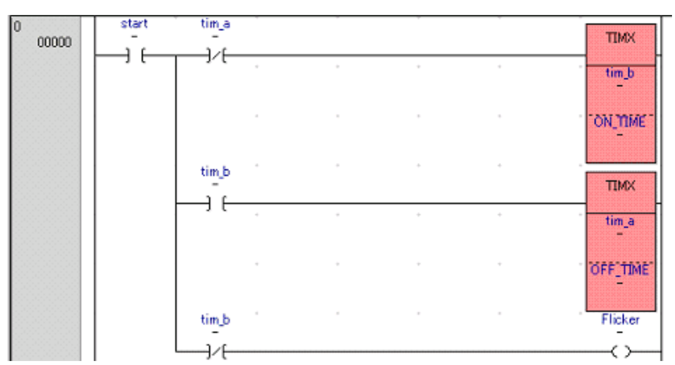
The procedures for programming the contents of a function block are the same as those described in *11-3 Programming Methods*.

Note Refer to *2-3 Function Blocks* for information on the function of function blocks.

1,2,3...

1. As an example, ladder diagram program will be used to create a function block for the following flicker rung.

Variable name	Type	Data type
tim_a	VAR	TIMER
tim_b	VAR	TIMER
ON_TIME	VAR_INPUT	UNIT
OFF_TIME	VAR_INPUT	UNIT
start	VAR_INPUT	BOOL
Flicker	VAR_OUTPUT	BOOL



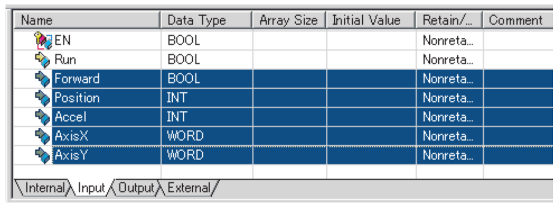
2. Select ***File - Save Change to project***. Alternately, press the **Ctrl+Shift+S** Keys.
   The function block will be saved in the project.
   A program cannot be assigned to a function block unless the function block has been saved in the project.

Reference Information: Grouping Input Variables or Output Variables for Function Blocks

When a function block is pasted into a program with the procedure described later in this section, the program can be very difficult to read if there are too many input variables in the operand input box or too many output variables in the output operand box. The following procedure can be used to simplify the display and use one input variable or one output variable to represent all of the input or output variables.

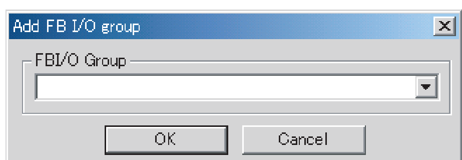
1,2,3...

1. On the Input Tab Page or Output Tab Page of the Local Variable Editor, select the variables to be grouped.
   - The following example shows a selection to group input variables.

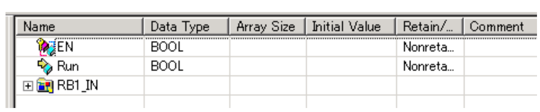


Name	Data Type	Array Size	Initial Value	Retain/...	Comment
EN	BOOL			Nonreta...	
Run	BOOL			Nonreta...	
Forward	BOOL			Nonreta...	
Position	INT			Nonreta...	
Accel	INT			Nonreta...	
AxisX	WORD			Nonreta...	
AxisY	WORD			Nonreta...	

2. Right-click the variables that were selected and select ***Group Input/Output Variables - Group***.
   The Add FB I/O Group Dialog Box will be displayed.

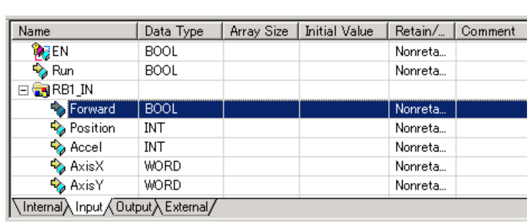


3. Input the name of the I/O group and then click the **OK** Button.
   As shown below, the variables have been grouped under the specified I/O group name.
   - The following example shows a group called RB1_IN.



Name	Data Type	Array Size	Initial Value	Retain/...	Comment
EN	BOOL			Nonreta...	
Run	BOOL			Nonreta...	
RB1_IN					

   - To ungroup the variables, right-click the I/O group name and select ***Group Input/Output Variables - Release Group***.
   - To delete elements from the group, open the group folder, right-click the element, and then select ***Release Member*** from the ***Group Input/Output Variables*** Menu. To change the display order of the group elements, right-click the element, and then select ***Up***, or ***Down***.



Name	Data Type	Array Size	Initial Value	Retain/...	Comment
EN	BOOL			Nonreta...	
Run	BOOL			Nonreta...	
RB1_IN					
Forward	BOOL			Nonreta...	
Position	INT			Nonreta...	
Accel	INT			Nonreta...	
AxisX	WORD			Nonreta...	
AxisY	WORD			Nonreta...	

Reference Information: Changing Variable Types

Internal variables can be changed to input, output, or external variables, and output, input, or external variables can be changed to internal variables. This is performed on the Ladder Editor.

The following example shows how to change an internal variable to an input variable.

Right-click the internal variable to be changed on the Ladder Editor and select ***Change Variable Usage - Input***.

The variable type will be changed from an internal variable to an input variable. When the variable type is change, the variable will be removed from the Internal Tab Page of the Variable Editor and placed on the Input Tab Page.

Commands on the menu for variable types that cannot be used will be grayed output. (For example, *Change Variable Usage - Output* will be grayed out for a variable used for an input condition.)

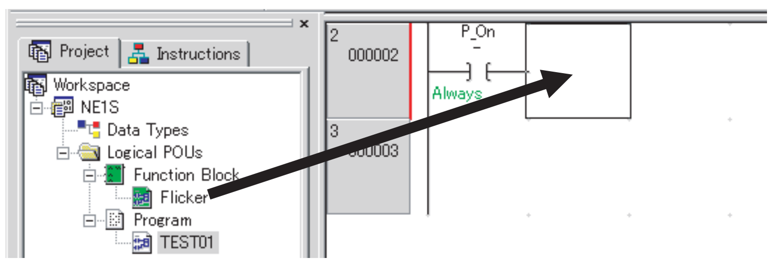
11-4-3 Pasting Function Blocks into Programs

Use the following procedure to paste a function block into a program.

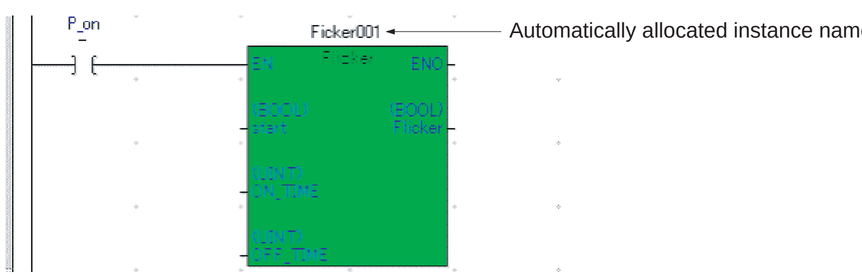
1,2,3...

1. Drag the function block from the Workspace and drop it at the insertion point in the ladder diagram.

   If the insertion point is selected and clicked in Function Block Mode (entered by selecting ***Ladder - Mode - Function Block***), the Edit Function Block Dialog Box will be displayed. Select the function block, input the instance name, and click the **OK** Button.

   The function block will be inserted into the ladder diagram and displayed as shown below.

   Automatically Generated Instance Variables

   If the *Create Function Block Instances Automatically* setting is enabled (default) in the Edit Tab of the Ladder Window under *Tool - Option*, instance variables will be automatically generated and an instance name will be automatically displayed, as shown in the above diagram. The instance name consists of the FB body name + _XXX (where XXX indicates a serial number starting from 001). When the *Create Function Block Instances Automatically* setting is enabled (default), the procedure described in step 3 is not required.

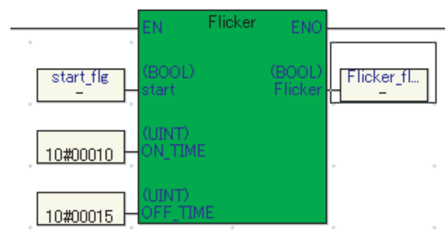
2. Input the function block operands.

   • Method 1:
   Double-click the operand input position (or select it and press the **Enter** Key) to display the Edit Function Block Argument Dialog Box. Input the function block operands in this dialog box.

- Method 2:
  It is also possible to select the input position and input the value directly from the keyboard, or drag and drop the variable from the Variable Editor to the operand.

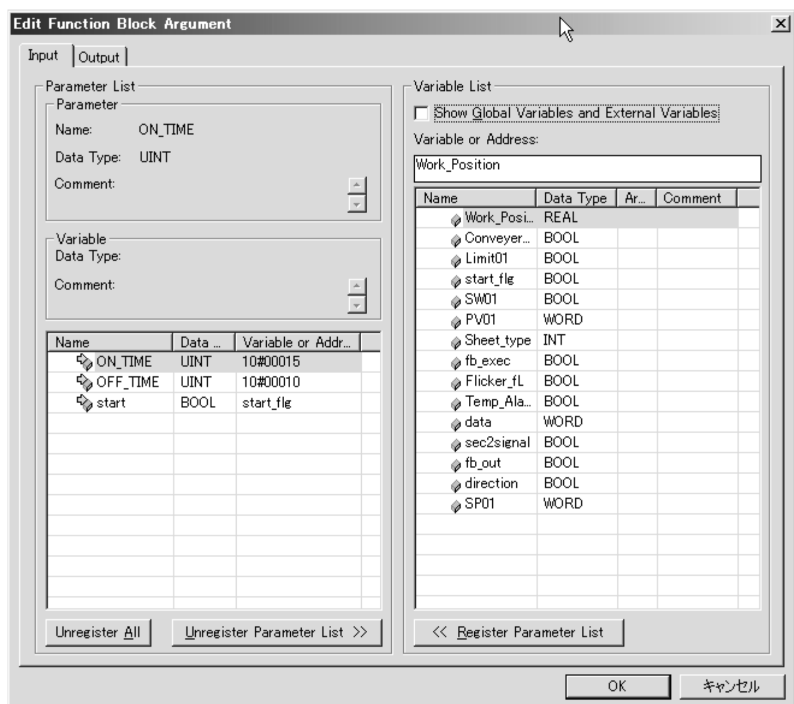
Note
(1) Addresses can also be specified for operands, but data types and sizes will not be checked. Always use the proper data type so that data is not corrupted.
(2) Refer to *2-4* for the methods for inputting constants and operand numbers.

- Input Example



Operand Input Method 1

a. Double-click the function block operand input position to display the Edit Function Block Argument Dialog Box. (The Input Tab Page or Output Tab Page will be moved to the front automatically, based on the operand position.)



b. In the *Parameter List* area, select the desired variable in the function block. The selected variable will be highlighted in gray. (In this example, select an input variable such as *start* in the Input Tab Page. Select the output variable *Flicker* in the Output Tab Page.)

c. In the *Variable List* at the upper-right, select the variable that you want to be the input source and click the **Register Parameter List** Button. (In this example, *start_flg* is registered to input variable *start*.)

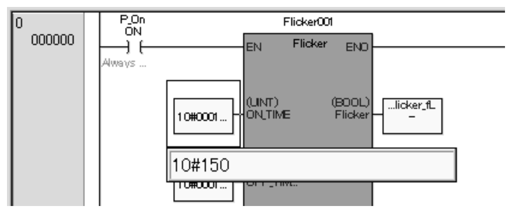
If you want to input a constant, input the constant directly in the *Variable or Address* Field. (In this example, *10#00010* is input directly for input variable *ON_TIME* and *10#00015* is input directly for input variable *OFF_TIME*.

In the *Variable List* at the upper-right, select the variable that you want to be the output destination and click the **Register Parameter List** Button. (In this example, *Flicker_fL* is registered to output variable *Flicker*.)

Operand Input Method 2

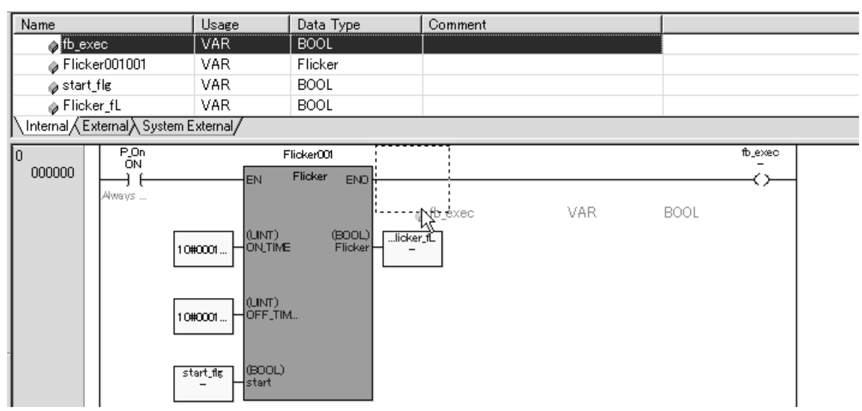
Direct Input

Select the position where you want to input the function block operand (without double-clicking) and directly input the value from the keyboard.

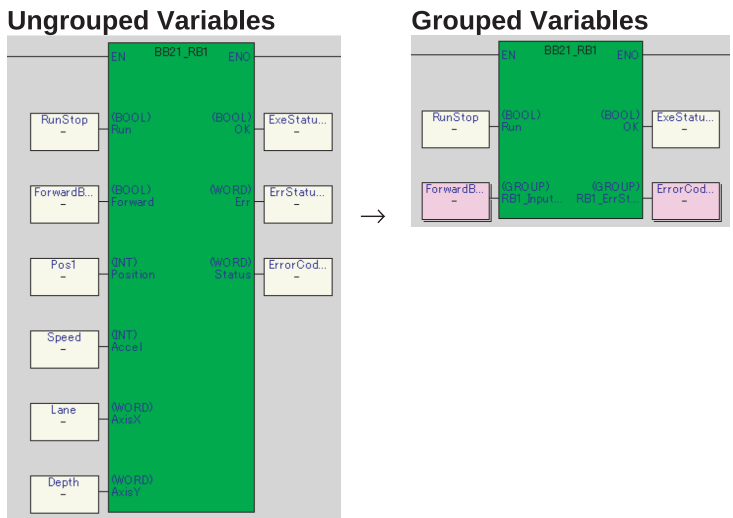


Dragging and Dropping a Variable

Drag and drop the variable from the Variable Editor to the desired input position. The data types of the source and destination variables must be the same.



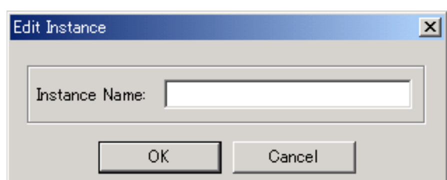
Note If I/O groups have been created for input variables or output variables, the group names will be displayed as shown below when the function block is pasted into a program.



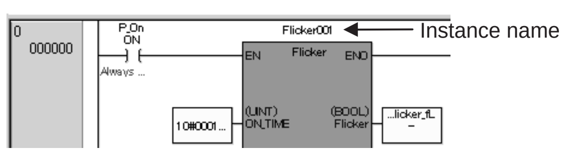
3. Assign an instance name to the function block.

   Note The procedure in step 3 is not required if the *Create Function Block Instances Automatically* setting is enabled (default) in the Edit Tab of the Ladder Window under *Tool - Option*. In this case, the instance names will be generated automatically. Proceed to step 4.

   • Double-click the function block in the ladder diagram. The following dialog box will be displayed.



   • Input the instance name (*Flicker01* in this example) and then click the **OK** Button. The instance name will be displayed on the function block as shown below.



4. Code the rest of the program through the END instruction.
5. Select ***File - Save Change to project***. Alternately, press the **Ctrl+Shift+S** Keys.
   The program will be saved in the project.
   A program cannot be assigned to a task unless the program has been saved in the project.

This completes pasting a function block.

Note (1) After programming, always select ***File - Save Changes to project***. Alternately, press the **Ctrl+Shift+S** Keys.
An alarm will be displayed if there is an error in the program. Correct the error and then select ***File - Save Change to project*** again.

(2) Read-protection can be set for a function block. Refer to *11-4-5 Read-protecting Function Blocks* for the procedure.

11-4-4 Editing the Function Block Body after Pasting

Changes to a function block definition itself (editing of input variables or output variables) can be reflected in an instance even if the function body was already pasted as an instance. Function block instances can be updated by selecting ***Edit - Update Function Block Instance***.

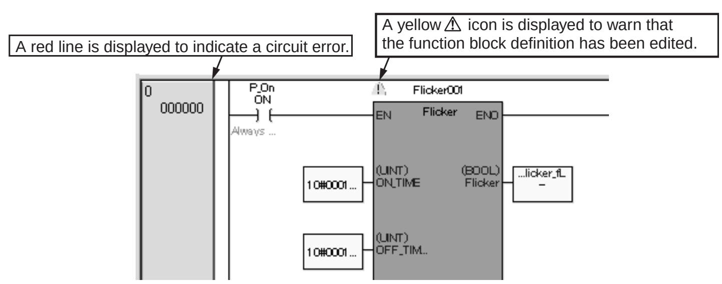
With this menu command, it is no longer necessary to delete and repaste instances in the program after editing the function block definition itself. It is also unnecessary to input the argument/operand values again.

Procedure

Use the following procedure to update function block instances.

1,2,3...

1. After pasting the function body as an instance, edit the function block body's input variables or output variables (change, add, or delete) and save the project.
   - A confirmation dialog box will be displayed, indicating the following: *Function block was modified. Please check the output window.* At the same time, the Find Dialog Box will show the program addresses of that function block's instances.
   - A yellow ⚠: icon will be displayed at the upper-left corner of instances in the program, warning that there has been a change, and a circuit error will occur.



2. Either select ***Edit - Update Function Block Instance*** or select the instance, right-click, and select ***Update Function Block Instance*** from the popup menu. The changes to the function block definition will be reflected in the program's instances and the yellow warning icons and circuit errors will be cleared.

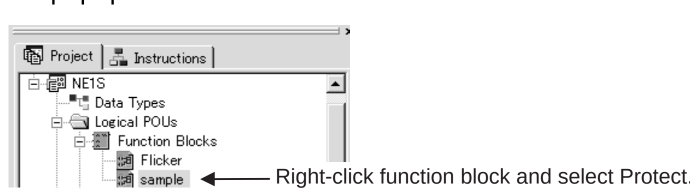
11-4-5 Read-protecting Function Blocks

Passwords can be set to set read protection for individual function blocks.

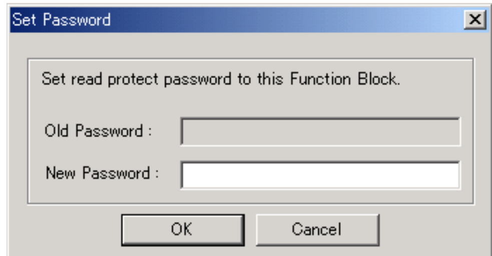
If read protection is set, a function block cannot be displayed or edited without inputting the password.

1,2,3...

1. Right-click the function block to be read-protected and select ***Protect*** from the popup menu.



The following Set Password Dialog Box will be displayed.



2. Input the password and then click the **OK** Button.
   A password will be set for the function block specified in step 1. After the password is set, the function block cannot be displayed or edited without inputting the password.

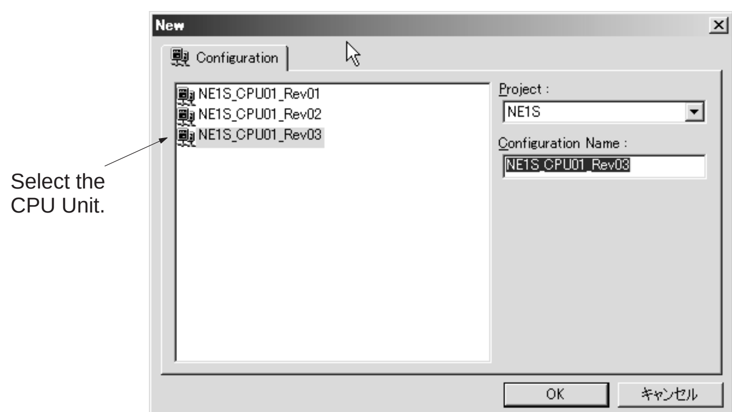
11-5 Creating Configurations and Assigning Programs to Tasks

11-5-1 Creating a Configuration

Use the following procedure to create a new configuration.

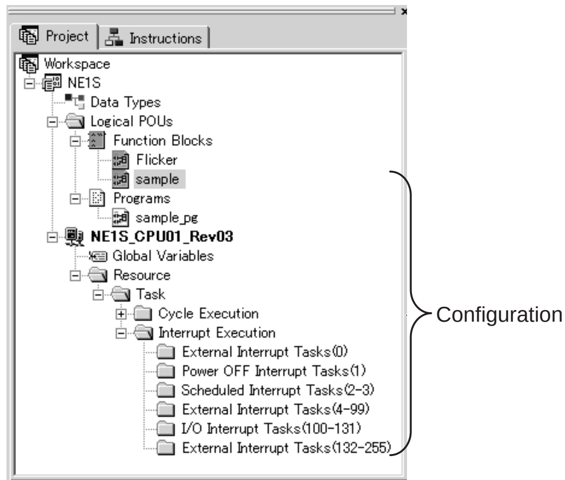
1,2,3...

1. 1 Right-click the project name in the Workspace and select ***Create Configuration***. Alternately, select ***File - New - Configuration***.
   The following New Dialog Box will be displayed.



2. Select the CPU Unit.

3. If the displayed name is acceptable, click the **OK** Button. To change the configuration name, input the configuration name and then click the **OK** Button.
   The configuration will be created and displayed in the Workspace.



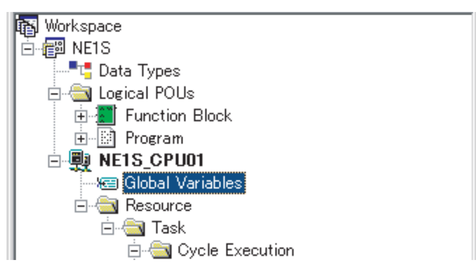
This completes creating a configuration.

11-5-2 Creating and Editing Global Variables

Creating Global Variables

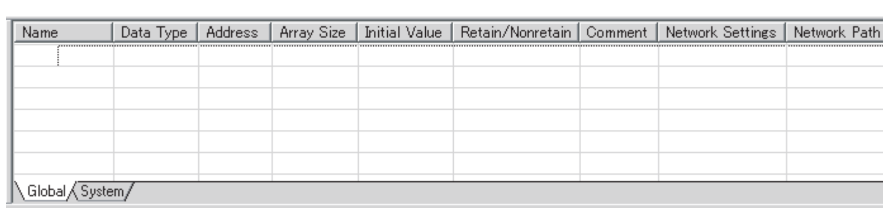
1,2,3...

1. Create a new configuration (*File - New - Configuration*).
   The global variables will be displayed in the Workspace.

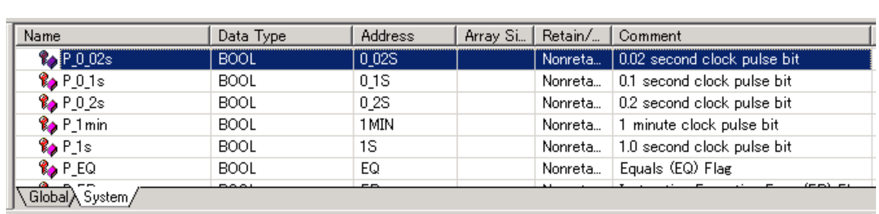


2. Double-click ***Global Variables*** in the Workspace.
   The Global Variable Editor will be displayed.

■ Global Sheet



■ System Sheet

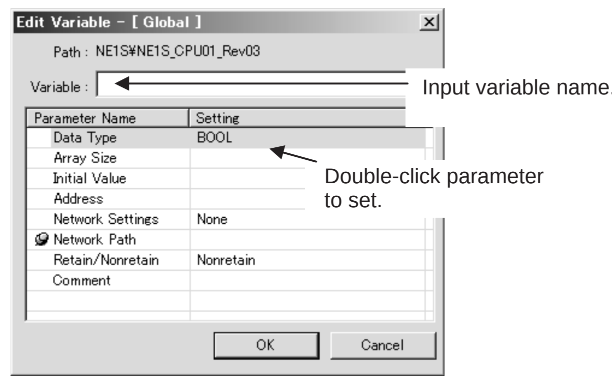


System global variables are registered in advance in the System Sheet. System global variables cannot be changed.

Adding Variables

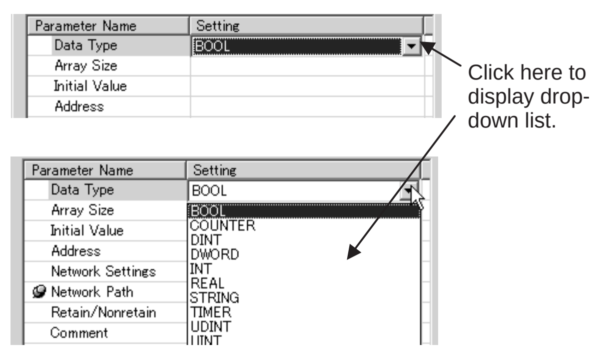
1,2,3...

1. Right-click on the Global Variable Sheet and select ***Add*** from the pop-up menu. Alternately, double-click an empty row.
   The Edit Variables Dialog Box will be displayed.



2. Input the variable name.
3. Set the variable parameters. Double-click the parameter to be set. Editing the parameter will be enabled.

   Example: The following display will appear when *Data Type* is double-clicked:



4. Set the parameter.
5. When all settings have been completed, click the **OK** Button.
   Parameters are displayed as described in the following table.

Note Refer to *2-2* for detailed variable specifications.

Parameter	Meaning	Values
Path	Displays the valid scope of the variable.	Global variables: Project_name/ configuration_name/variable_name
Data type	Set the data type of the variable.	BOOL, INT, UINT, DINT, UDINT, WORD, DWORD, TIMER, COUNTER, STRING, or REAL
Array Size	Specify whether an array is to be used and the number of elements in the array.	Not an array variable: 0 Array variable: Set the number of elements from 1 to 255.

Parameter	Meaning	Values
Initial Value	For programs, set the initial value of the variable when program execution is started. For function blocks, set the value of the variable when an instance of the function block is executed.	Set the initial value of the variable according to the data type. • BOOL, WORD, or DWORD: Unsigned hexadecimal Input the number after “16#”. • INT or DINT: Signed decimal Input the number after “+10#” or “–10#”. • UINT or UDINT: Unsigned decimal Input the number after “10#”. • REAL: Real number (e.g., +1.0, –0.23, +9.8E-3)
Address	Set when a specific address is manually set for the variable (AT designation). This setting is supported only for global variables.	
Network settings	Select ***Input*** to disclose the variable as an input from the network for cyclic communications, ***Output*** to disclose the variable as an output to the network, and ***None*** to not disclose the variable. Any variables set as inputs or outputs here can be imported into the Network Configurator after saving the project. This allows connection settings for cyclic communications with ControlNet to be performed with the variable names used in the program.	None, Input, or Output
Network Path	Reserved	
Retain/Nonretain	Specify whether to maintain the value of the variable when power is turned OFF and ON, and when operation is started.	Retain or Nonretain
Comment	Input a comment for the variable.	256 ASCII characters max.

Note

(1) Refer to *2-4-4* for details on network paths.

(2) When the Cross Reference Window has been displayed (by selecting ***View - Window - Cross Reference***), a variable’s cross reference information (program address, instruction name, program name, etc.) can be displayed just by selecting that variable in the Variable Editor.

Editing Global Variables

1,2,3...

1. Double-click the variable to be edited on the Global Sheet. Alternately, right-click the variable and select ***Edit***.
   The Edit Variables Dialog Box will be displayed.
2. Edit the parameters of the Variable and then click the **OK** Button.

Deleting Variables

1,2,3...

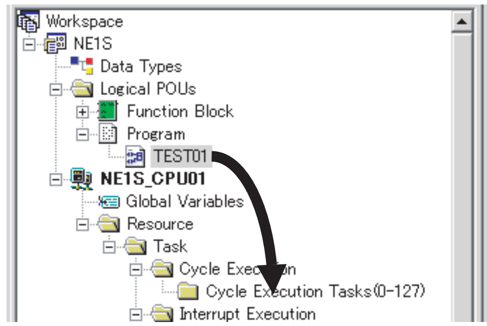
1. Select the variable to be deleted and press the **Delete** Key.
2. A dialog box will appear to confirm the deletion. Click the **Yes** Button.
   The variable will be deleted.

11-5-3 Pasting Programs into Tasks

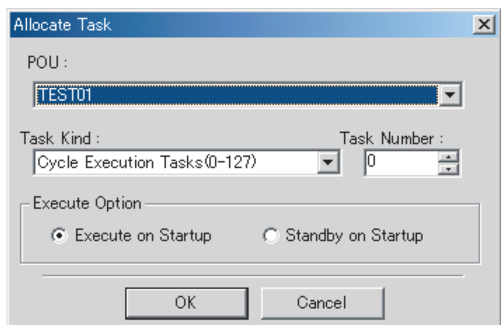
Use the following procedure to paste a program into a task.

1,2,3...

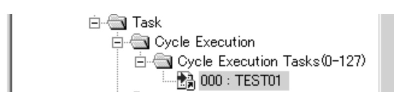
1. Drag the program from the Workspace and drop it on the task folder. In this example, the program TEST01 is assigned to *Cycle Execution Task (0-127)*.



   The following Allocate Task Dialog Box will be displayed.



2. Set the *Execute Option* and *Task Number* (these are automatically set if the program is dragged and dropped), and then click the **OK** Button.
   The program will be assigned to the task and displayed as shown below.



Program assigned to a task.

Assigning a program to a task makes it possible to execute the task.

11-5-4 Checking External Variables for Consistency

The NE Programmer can check whether there are any inconsistencies between the configuration's global variables and the external variables of the logical POUs assigned to that configuration as tasks. Inconsistencies can occur for either of the following two reasons.

1. Variables with the same name exist in both programs, but the data types or sizes do not match.
2. External variables exist, but global variables do not exist.

If there is an inconsistency, a build error will occur.

The External Variable Consistency Check function checks for the two inconsistencies described above and helps correct the inconsistencies if any are detected.

1,2,3...

1. Either select ***Variable - Check Consistent with Extern*** from the Menu Bar or select ***Check Consistent with Extern*** from the popup menu in the Variable Window or Program Window.

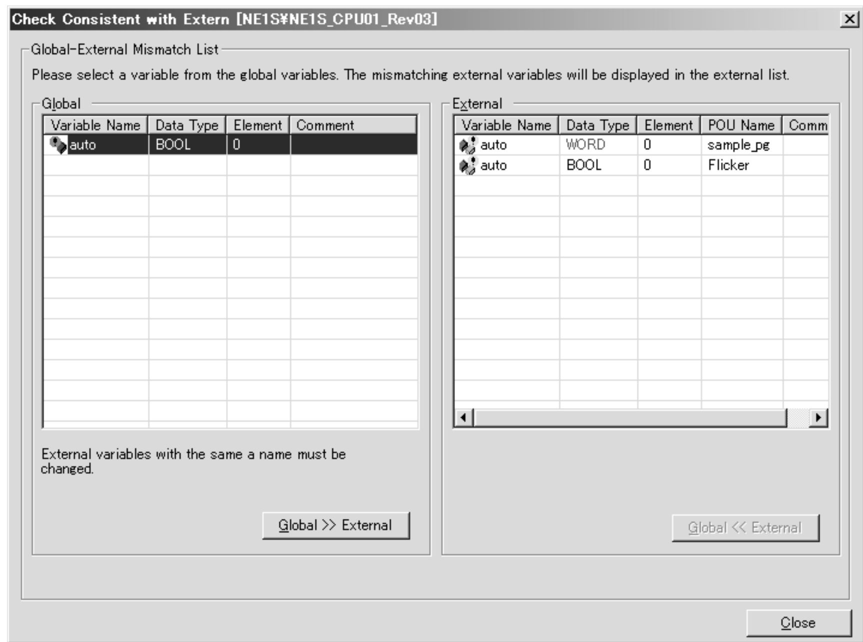
If no inconsistencies are found, the message *A problem is not in adjustment* will be displayed. In this case, click the **OK** Button to continue.

2. If any inconsistencies are found, the following Check Consistent with Extern Dialog Box will be displayed.

   a. Unifying to the Global Variable's Data Definition

   Select the variable in the *Global* variable mismatch list and click the **Global >> External** Button. Click the **Yes** Button in the confirmation dialog box that is displayed.

Note For consistency, the external variable's comment is overwritten by the selected global variable's comment.



   b. Unifying to the External Variable's Data Definition

   Select the variable in the *External* variable mismatch list and click the **Global << External** Button. Click the **Yes** Button in the confirmation dialog box that is displayed.

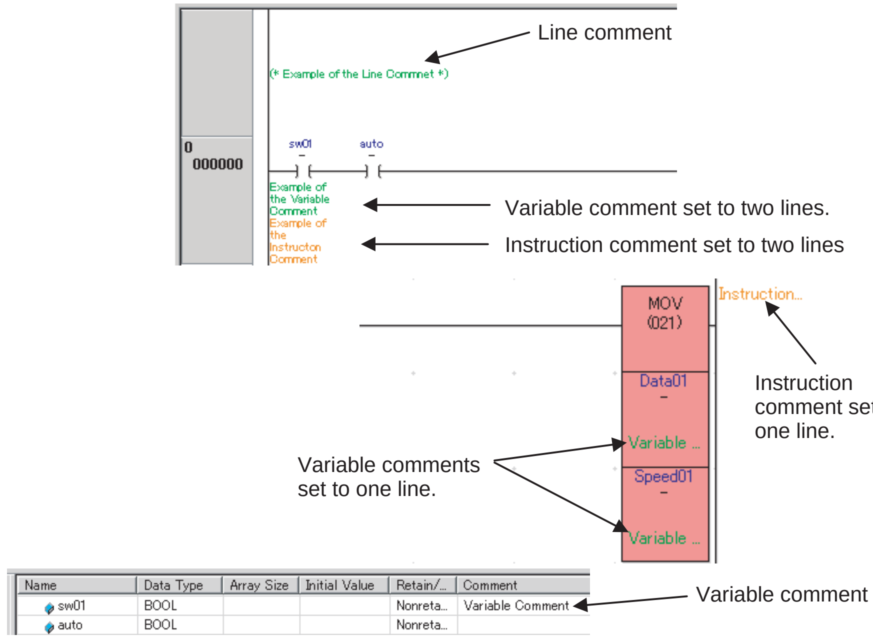
Note For consistency, the comments of the external variable/global variable are overwritten by the selected external variable's comment. In addition, the global variable's AT specification, initial value, and network variable settings will be deleted.

11-6 Editing Comments

11-6-1 Overview

The following comments can be input and edited.

Comment type	Description	Display location
Variable comments	Comments for variables	Below instructions
Instruction comments	Comments for instructions	Below variable comments and in the comment column in the Variable Editor.
Line comments	Comments that are input on comment lines inserted above or below rungs	Above or below rungs



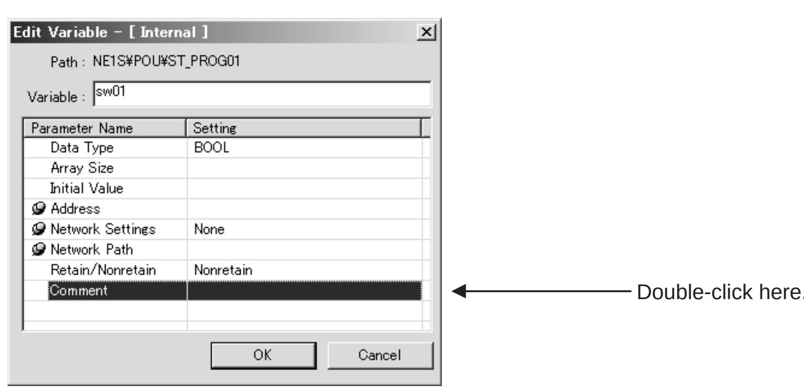
Note (1) Variable comments and instruction comments can be displayed or hidden and the number of display lines can be displayed on the Display Setting Tab Page of the View Options Dialog Box displayed when ***View - Option*** is selected.

(2) The default settings are shown above, i.e., 1 line displayed for both variable and instruction comments.

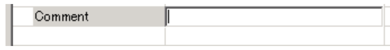
(3) To set the display colors for comments, use the Color Tab Page of the View Options Dialog Box.

11-6-2 Inputting Variable Comments

1,2,3... 1. Double-click the variable to which a comment is to be added on the Variable Editor.
The Edit Variables Dialog Box will be displayed.



2. Double-click the comment field.
   Editing the comment will be enabled as shown below.



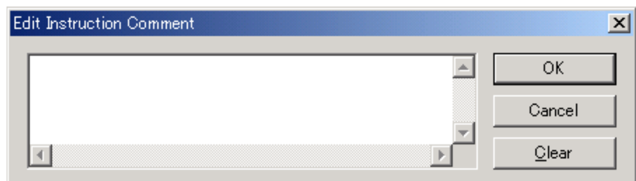
3. Input the comment and then click the **OK** Button.

Refer to *11-6-1 Overview* for display examples.

11-6-3 Inputting Instruction Comments

1,2,3...

1. Right-click the instruction to which a comment is to be added on the Ladder Editor and then select ***Edit Comment***.
   The Edit Instruction Comment Dialog Box will be displayed.



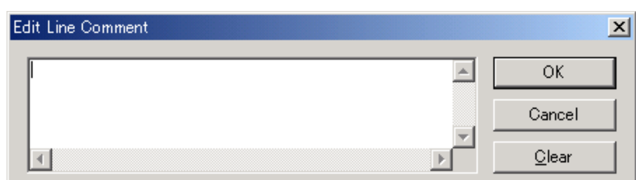
2. Input the comment and then click the **OK** Button.

Refer to *11-6-1 Overview* for display examples.

11-6-4 Inputting Line Comments

1,2,3...

1. Right-click the line where a comment is to be added and then select ***Insert Line Comment***.
   The Edit Line Comment Dialog Box will be displayed.



2. Input the comment and then click the **OK** Button.

Refer to *11-6-1 Overview* for display examples.

11-7 Search/Replace Function

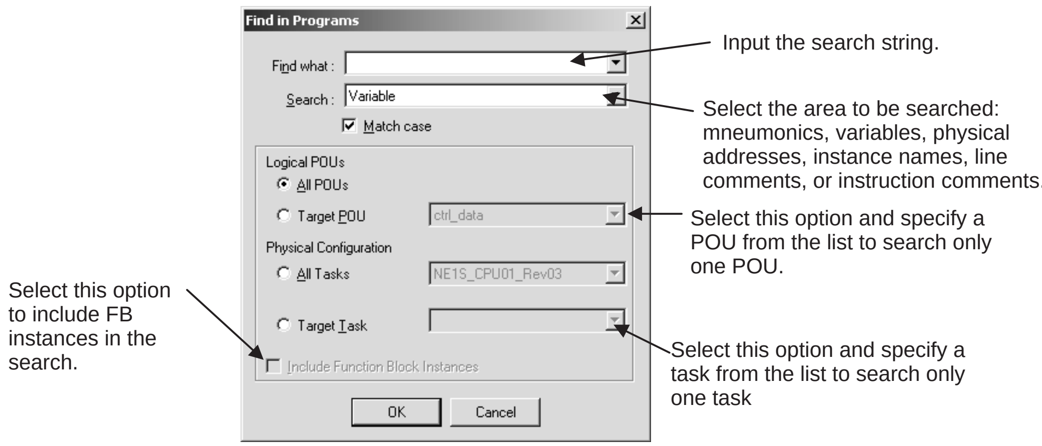
11-7-1 Overview

The following search and replace operations can be performed.

Type	Function
Searching programs	Mnemonics, variables, physical addresses, instance names, line comments, and instruction comments can be searched in all or specific POUs. ST programs are not searched. The search results are output to the Output Window. The locations that were found can be jumped to by double-clicking in the Output Window.
Search/replace/jump operations in the Ladder Editor	Search/replace operations can be performed in the active Ladder Editor. Jumping is possible to specified step numbers (program addresses) or rung numbers.
Search/replace/jump operations in the Variable Editor	Search/replace operations can be performed in the active Variable Editor.

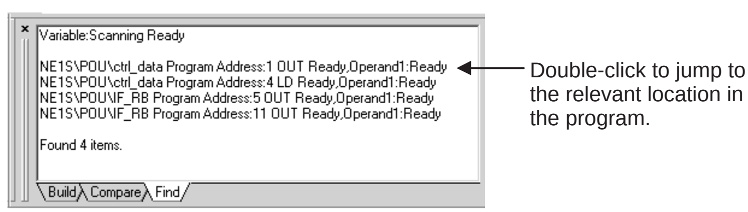
11-7-2 Searching Programs

1,2,3... 1. Press the **Ctrl+Shift+F** Keys. Alternately select ***Edit - Find in Programs***. The Find in Programs Dialog Box will be displayed.



Note ST programs are not searched.

2. Set the range to be searched, the text string to search for, and the search conditions, and then click the **OK** Button.
   The search results will be displayed in the Output Window.

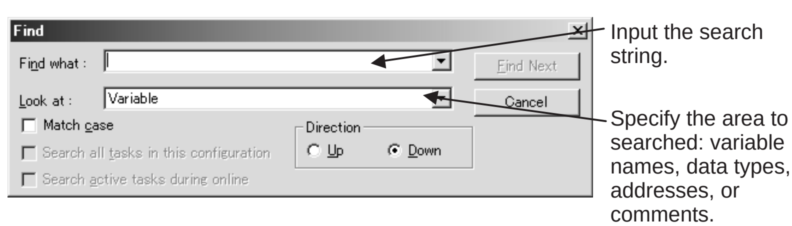


11-7-3 Search/Replace/Jump Operations in the Ladder Editor

Searching in the Ladder Editor

1,2,3... 1. Open the Ladder Editor to be searched.

2. Press the **Ctrl+F** Keys. Alternately select ***Edit - Find***.
   The Find Dialog Box will be displayed.



Note When the NE1S is connected online, a search operation was executed in the Ladder Editor, and you want to search through all tasks, select the *Search all tasks in this configuration* Option. If you want to search only the active tasks, select the *Search active tasks during online* Option.

3. Set the range to be searched, the text string to search for, and the search conditions, and then click the **Find Next** Button.
   - If the text string is found, the line in the Variable Editor containing it will be highlighted (selected).
   - Press the **F3** Key to find the next occurrence.
   - Press the **Shift+F3** Keys to return to the previously found location.

Jumping to a Variable Declaration from the Ladder Editor

When a variable is selected in the Ladder Editor, the NE Programmer can jump directly to that variable's declaration position (in the Variable Editor).

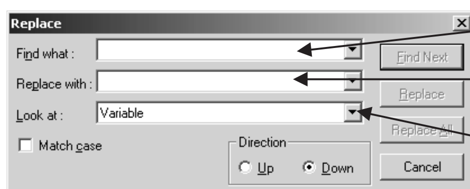
1,2,3...

1. Select the variable in the Ladder Editor.
2. Select ***Edit - Jump - Jump Variable Define***. The NE Programmer will jump to the same variable name in the Variable Editor.

Replacing in the Ladder Editor

1,2,3...

1. Open the Ladder Editor in which to perform the replace operation.
2. Press the **Ctrl+H** Keys. Alternately select ***Edit - Replace***.
   The Replace Dialog Box will be displayed.



Input the search string.

Input the replacement string.

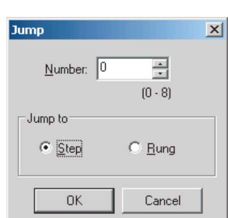
Specify the area to be searched: variables or physical addresses

3. Set the range to be searched, the text string to search for, the replacement text string, and the search conditions, and then click the **Find Next** Button. If the text string is found, the instruction in the Ladder Editor containing it will be highlighted (selected).
4. Click the **Replace** Button.
   The text string will be replaced.

Jumping in the Ladder Editor

1,2,3...

1. Make the Ladder Editor active and press the **Ctrl+G** Keys. Alternately select ***Edit - Jump***.
   The Step/Rung No. Jump Dialog Box will be displayed.



2. Set the step number (program address) or rung number and then click the **OK** Button.
   The cursor will jump to the specified step or rung and the instruction or rung will be highlighted (selected).

Searching for Address References in the Ladder Editor (Tab Key)

Jumps can be made from an input instruction at the cursor to an output instruction using the same variable for from an output instruction to an input instruction using the same variable.

1,2,3...

1. In the Ladder Editor, select the input or output instruction for which to search for address references.
2. Press the **Tab** Key. Alternately select ***Edit - Jump - Bit Address Reference***.
   The cursor will jump to an output or input instruction using the same variable. The cursor will jump to the next output or input instruction each time the **Tab** Key is pressed.

Other Search Operation in the Ladder Editor

Next Address (Shift+Alt+N Keys)

A jump can be made from the instruction at the cursor to an instruction with the same address.

1,2,3...

1. In the Ladder Editor, select the instruction with the address for which to search.
2. Press the **Shift+Alt+N** Key. Alternately select ***Edit - Jump - Next Address Reference***.
   The cursor will jump to an instruction with the same address.

Next Input (Shift+Alt+I Keys)

A jump can be made from the instruction at the cursor to an input instruction with the same variable.

1,2,3...

1. In the Ladder Editor, select the instruction with the variable for which to search.
2. Press the **Shift+Alt+I** Key. Alternately select ***Edit - Jump - Next Input***.
   The cursor will jump to an input instruction with the same variable.

Next Output (Shift+Alt+O Keys)

A jump can be made from the instruction at the cursor to an output instruction with the same variable.

1,2,3...

1. In the Ladder Editor, select the instruction with the variable for which to search.
2. Press the **Shift+Alt+O** Key. Alternately select ***Edit - Jump - Next Output***.
   The cursor will jump to an output instruction with the same variable.

Back (Shift+Alt+B Keys)

The cursor can be returned to the previous instruction from which a search was made.

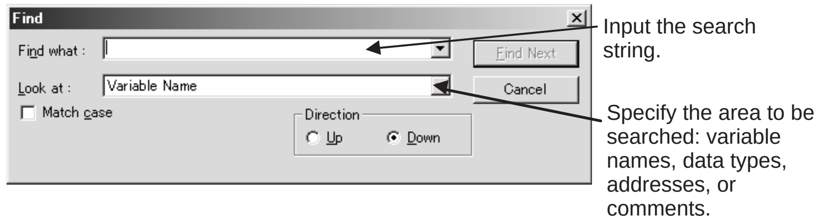
Press the **Shift+Alt+B** Key. Alternately select ***Edit - Jump - Previous Jump Point***.
The cursor will return to the previous instruction.

11-7-4 Search/Replace Operations in the Variable Editor

Search Operations in the Variable Editor

1,2,3...

1. Place the cursor in the Variable Editor to be searched.
2. Press the **Ctrl+F** Keys. Alternately select ***Edit - Find***.
   The Find Dialog Box will be displayed.

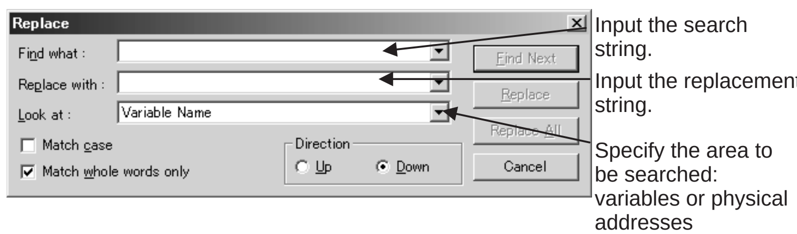


3. Set the range to be searched, the text string to search for, and the search conditions, and then click the **Find Next** Button.
   - If the text string is found, the line in the Variable Editor containing it will be highlighted.
   - Press the **F3** Key to find the next occurrence.
   - Press the **Shift+F3** Keys to return to the previously found location.

Replace Operations in the Variable Editor

1,2,3...

1. Place the cursor in the Variable Editor in which to perform a replace operation.
2. Press the **Ctrl+H** Keys. Alternately select ***Edit - Replace***.
   The Replace Dialog Box will be displayed.



3. Set the range to be searched, the text string to search for, the replacement text string, and the search conditions, and then click the **Find Next** Button. If the text string is found, the line in the Variable Editor containing it will be highlighted.
4. Click the **Replace** Button.
   The text string will be replaced.

11-8 Cross Reference Function

11-8-1 Overview

When the cursor is over a variable or address in the Ladder Editor or Variable Table, the cross reference function can display a list (in the Cross Reference Window) showing the corresponding instructions in which that variable or address is used. Jumps can be made to the instructions by double-clicking items displayed in the list.

A log is also kept of the instructions that are found to enable jumping to them by selected instructions registered in the log.

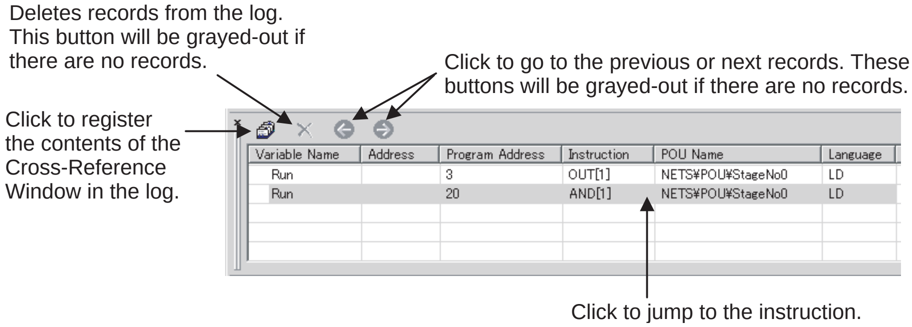
11-8-2 Cross Reference Window Displays and Operations

1,2,3...

1. If the Cross Reference Window is not currently displayed, select ***View - Window - Cross Reference***.

2. Move the cursor to the variable or address for which to search.
   A list will be displayed to show the instructions in which the variable or address at the cursor is used, as shown below.

   In this example, the cursor is located over the variable *RbStat2*.

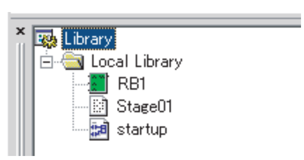


Note When some of the function block's I/O variables have been grouped, the Cross Reference Window will display all of the cross reference information for the group members. Also, when an array variable is displayed, the Cross Reference Window will display all of the cross reference information for the other array variables with the same variable name.

11-9 Using the Library

11-9-1 Overview

Logical POUs (programs or function blocks) and groups of rungs can be saved as library files so that they can be reused.



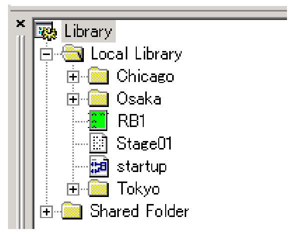
As shown above, programs, function blocks, and rung groups (one or more rungs) can be stored and reused with one file for each program, one file for each function block, and one file for each rung group.

Each library contains information such as the local variables (internal, input, output, and external variables), function blocks, data type definitions, and global variables.

By default, the library files are stored under a *Local Library* folder in the installation directory, as shown above.

As shown below, groups of library files can be registered and managed in a directory hierarchy.

You can also right-click the *Library* Directory and select ***Map Folder*** to register another folder in the first level of the library. System development by multiple engineers is possible by specifying a shared folder on a server in the first level of the library directory.



- Program or function block library files that have been registered can be added to projects by right-clicking and selecting ***Register to Library***.
- To reuse rung groups, they can be dragged and dropped on the ladder diagram program, or alternately the rung groups can be added by right-clicking and selecting ***Insert to Program***. When a rung group is added to a program, the variables can be changed or deleted.

Note Library parts cannot be edited after they have been registered in the library. They must be re-registered in order to change them.

11-9-2 Displaying the Library Window

If the Library Window is not currently displayed, select ***View - Window - Library***.

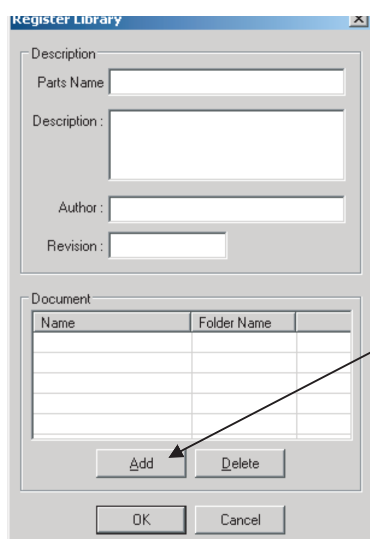
The Library Window will be displayed.

The Library Window will be automatically displayed when a library file is registered.

11-9-3 Registering Logical POUs in the Library

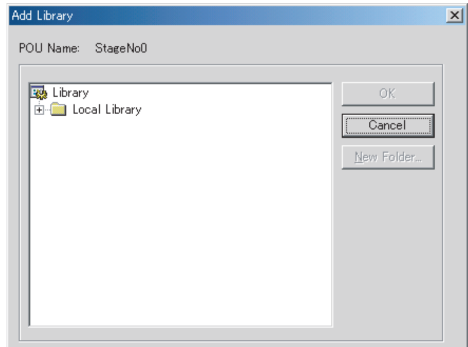
1,2,3...

1. Right-click the logical POU (program or function block) in the Workspace and select ***Register to Library*** from the popup menu.
   The Register Library Dialog Box will be displayed.



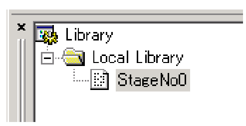
- When the logical POU used to register a library has a text file, the library can be linked to the relevant text file.
- Click the **Add** Button and specify the text file.
- The added text file can be opened from the Properties Window for the registered library (right-click on the library and select ***Properties***).

2. Input the name of the part and any other required items and then click the **OK** Button.
   The following dialog box will be displayed.



• To create a new folder in the library tree, select the insertion location and click the **New Folder** Button.

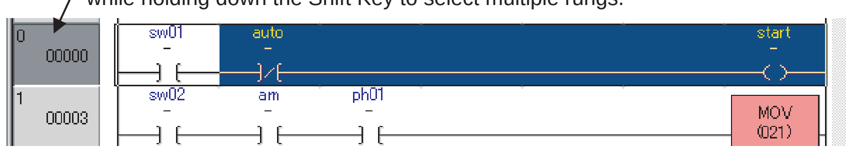
3. Select the folder in which to register the library part (Local Library, the default folder, in this example) and then click the **OK** Button.
   The part will be registered in the library and displayed in the Library Window as shown below.



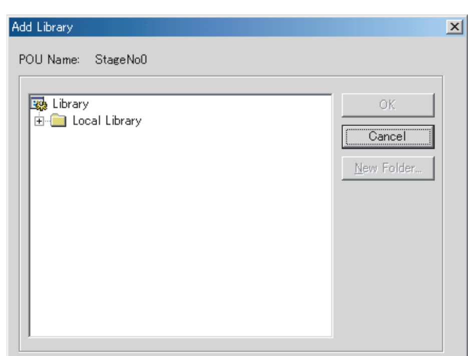
11-9-4 Registering Rung Groups in the Library

1,2,3... 1. Select one or more rungs in the Ladder Editor and select ***Library - Rung - Register to Library***.

Click the rung header area to select the rung. Click the rung header area while holding down the Shift Key to select multiple rungs.

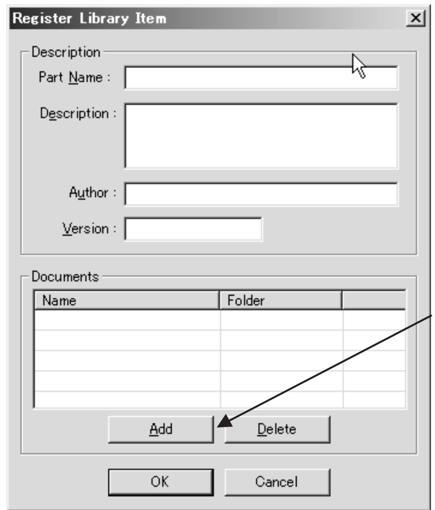


The following dialog box will be displayed.



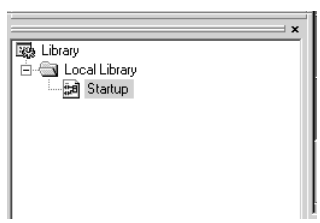
• To create a new folder in the library tree, select the insertion location and click the **New Folder** Button.

2. Select the folder in which to register the library part (Local Library, the default folder, in this example) and then click the **OK** Button.
   - The Register Library Dialog Box will be displayed.



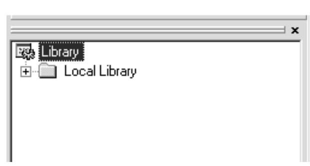
- When the rung group used to register a library has a text file, the rung group can be linked to the relevant text file.
- Click the **Add** Button and specify the text file.
- The added text file can be opened from the Properties Window for the registered rung group (right-click on the rung group and select ***Properties***).

3. Input the name of the part and any other required items and then click the **OK** Button.
   The part will be registered in the library and displayed in the Library Window as shown below.



11-9-5 Registering Folders in the Library

1,2,3... 1. Right-click the Library and select ***Map Folder***.



The following Browse Folder Dialog Box will be displayed.

2. Select the folder to be added and then click the **OK** Button.
   The specified folder will be added to the library tree.

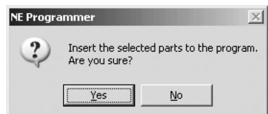
11-9-6 Using the Library

POUs (Programs and Function Blocks)

Program sections and function blocks can be inserted in projects.

1,2,3...

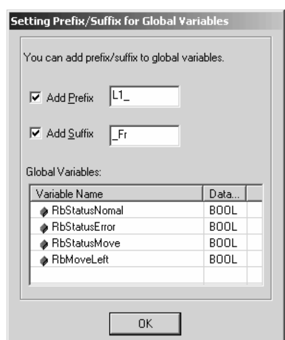
1. Right-click the program or function block and select ***Add to Project***. The following confirmation dialog box will be displayed.



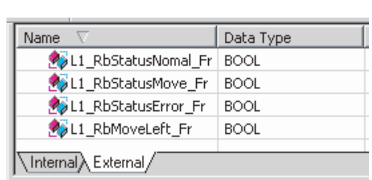
2. Click the **Yes** Button.
   - If the POU contains a global variable, the following Setting Prefix/Suffix for Global Variables Dialog Box will be displayed.
   - Use this dialog box to attach prefixes and suffixes to all global variables contained in POUs at once.

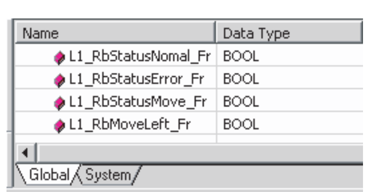
   Note This function enables the variable names for global variables to be changed altogether, making reuse of global variables easy.

   Example: Adding the prefix L1 and the suffix _Fr.



3. After setting, click the **OK** Button. When prefix/suffix settings are not required, click the **OK** Button without making any settings.
   - The program or function block will be added to the project. The POU name will be displayed in the project (do not use a library name).
   - The prefix/suffix settings in the above example will be displayed as follows:



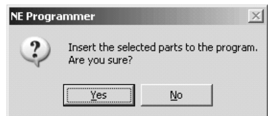


Rung Groups

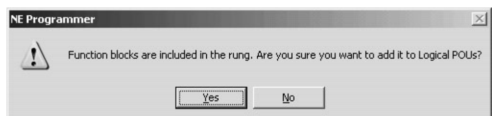
Rung groups can be simply dragged and dropped into ladder programs. Delete variables and change variable names as required.

1,2,3...

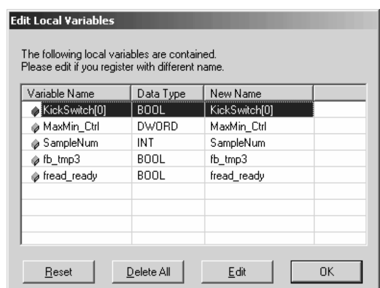
1. Drag and drop the rung group into the ladder program (or right-click on the rung group and select ***Insert to Program***). The following confirmation dialog box will be displayed.



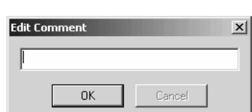
2. Click the **Yes** Button.
   - If the rung contains function blocks, the following confirmation dialog box will be displayed.



   - Click the **Yes** Button.
   - The following Edit Local Variables Dialog Box will be displayed.

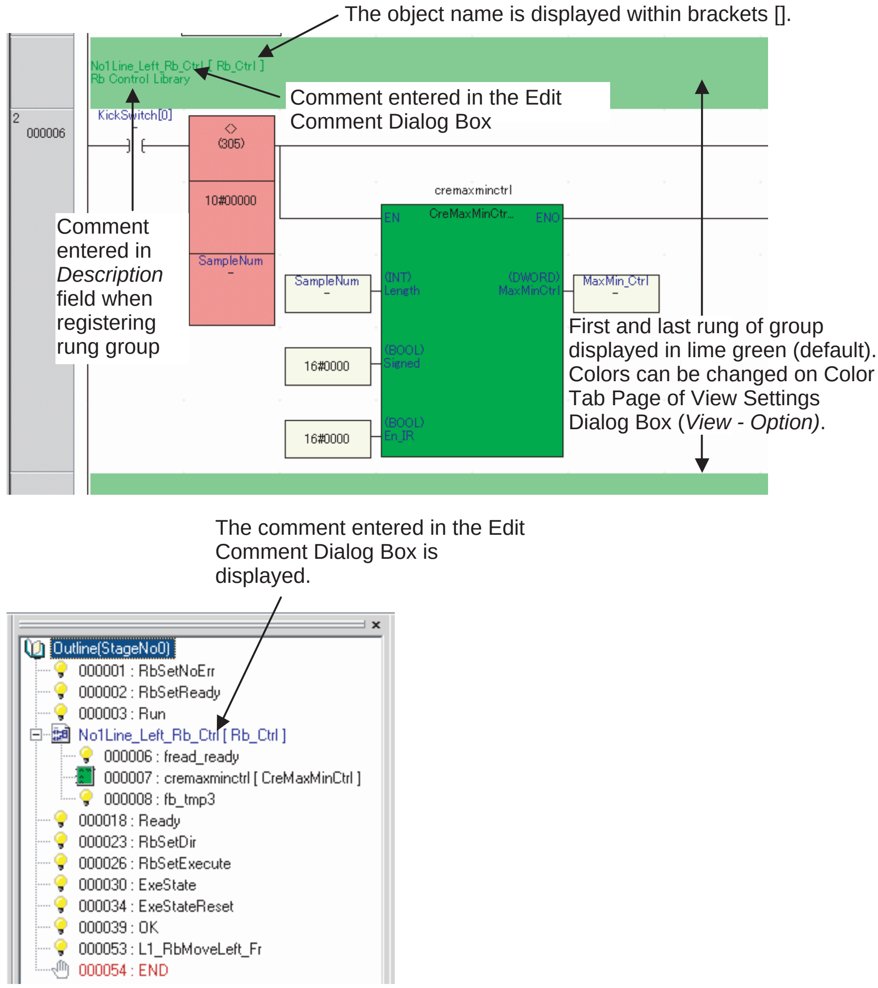


3. To use the variables without any changes, click the **OK** Button.
   - To change the variable name, double-click the variable name and change the name.
   - To delete all variables, click the **Delete All** Button.
   - To return to the original settings after editing the variables, click the **Reset** Button.
   Click the **OK** Button to display the Edit Comment Dialog Box.



4. Enter a comment and click the **OK** Button. The rung will be inserted into the ladder program.

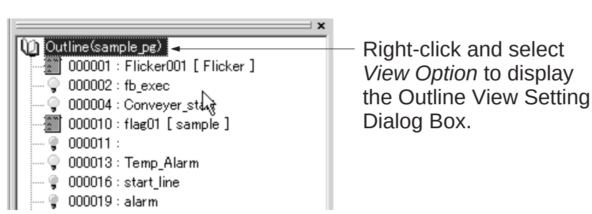
   Example



11-10 Outline Window

11-10-1 Outline Window

The Outline Window displays an outline of the logical POU (program or function block) that is being edited. Jumping to any instruction in the Ladder Editor or ST Editor is possible by clicking it in the Outline Window.

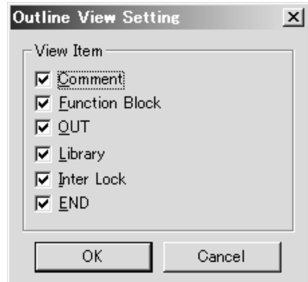


As shown above, line comments, outputs, function blocks, rung groups, and the END instruction are displayed in the Outline Window.

Note For structured text, only the line comments and function blocks are displayed in the Outline Window.

View Option

The displayed items in the Outline Window can be changed. To change the viewed items, right-click ***Outline*** displayed at the top of the Outline Window, select ***View Option***, and then select the required items in the Outline View Setting Dialog Box.



11-10-2 Displaying the Outline Window

1,2,3...

1. If the Outline Window is not currently displayed, select ***View - Window - Outline***.
   The Outline Window will be displayed. If a logical POU is being edited at the time, an outline will be displayed in the Outline Window.
2. To display an outline of a specific logical POU, double-click it in the Workspace.
   When the logical POU is opened, an outline will be displayed in the Outline Window at the same time.

11-11 Building and Compiling Programs

11-11-1 Building and Compiling

A project is build, all of the programs in the project are checked and an executable file for the CPU Unit is generated. Always build the project before downloading it to the CPU Unit.

Compiling can be used to perform program checks on individual logical POUs (programs or function blocks). Only one active logical POU (i.e., the one being edited) can be compiled at the same time.

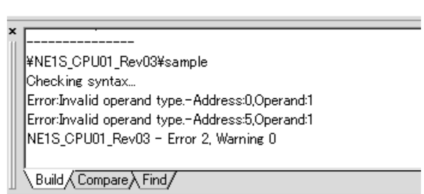
The program check level can be set.

11-11-2 Building

Press the **F7** Key. Alternately select ***Build - Build***.

The project will be built. Any errors or warning generated during building will be displayed in the Output Window.

Display Example

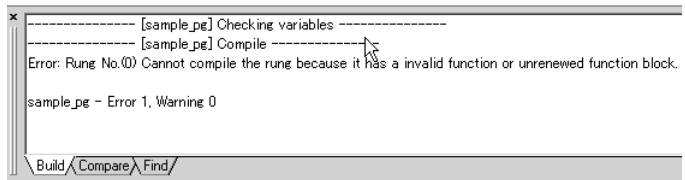


11-11-3 Compiling

1,2,3...

1. Make the program or function block to be checked active (i.e., open it).
2. Press the **Ctrl+F7** Keys. Alternately select ***Edit - Compile***.
   The program will be checked and any errors or warning generated during compilation will be displayed in the Output Window.

Display Example

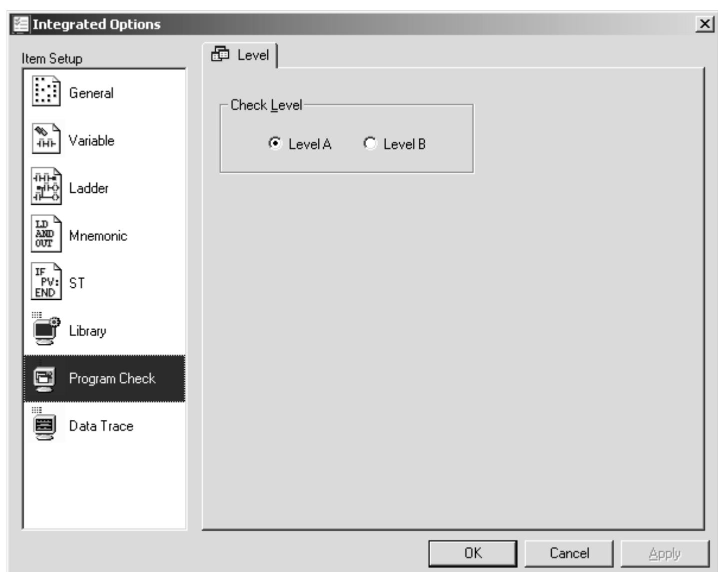


11-11-4 Level Settings for Program Check

Use the following procedure to set the level for program checking. If level A is set, a stricter program check will be performed. (The default is level A.)

1,2,3...

1. Select ***Tool - Option*** to display the Integrated Options Window. Select the ***Program Check*** Icon in the *Item Setup* List.



2. Select *Level A* or *Level B* and then click the **OK** Button.

■ **Level A**

Check item	Normal POUs		FB POUs	
	Compiling	**Building**	**Compiling**	**Building**
Duplicated outputs	Yes	Yes	Yes	Yes
Duplicated timer numbers	Yes	Yes	Yes	Yes
Duplicated counter numbers	Yes	Yes	Yes	Yes
Duplicated FAL numbers	Yes	Yes	Yes	Yes
Duplicated variables	Yes	Yes	Yes	Yes
Check within specific areas	Yes	Yes	Yes	Yes
Check outside of specific areas	Yes	Yes	Yes	Yes
Array indices	Yes	Yes	Yes	Yes

Check item	Normal POUs		FB POUs	
	Compiling	Building	Compiling	Building
Automatic allocation area duplication	No	Yes	No	Yes
END instruction	Yes	Yes	Yes	Yes
FOR-NEXT check	Yes	Yes	Yes	Yes
IL-ILC check	Yes	Yes	Yes	Yes
Jump 0 check	Yes	Yes	Yes	Yes
Jump check	Yes	Yes	Yes	Yes
Intertask check	No	Yes	No	No

■ **Level B**

Check item	Normal POUs		FB POUs	
	Compiling	Building	Compiling	Building
Duplicated outputs	No	No	No	No
Duplicated timer numbers	No	No	No	No
Duplicated counter numbers	No	No	No	No
Duplicated FAL numbers	No	No	No	No
Duplicated variables	No	No	No	No
Specific regions	Yes	Yes	Yes	Yes
Check outside of specific areas	Yes	Yes	Yes	Yes
Array indices	Yes	Yes	Yes	Yes
Automatic allocation area duplication	No	Yes	No	Yes
END instruction	Yes	Yes	Yes	Yes
FOR-NEXT check	Yes	Yes	Yes	Yes
IL-ILC check	Yes	Yes	Yes	Yes
Jump 0 check	Yes	Yes	Yes	Yes
Jump check	Yes	Yes	Yes	Yes
Intertask check	No	Yes	No	No

11-12 Importing and Exporting

11-12-1 Overview

The following logical POUs can be imported and variables can be imported/ exported from the Variable Editor.

Element	Import/Export files
Mnemonics	Import: Text (.txt) files in special formats
	Export: Text (.txt) files in special formats
Variables	Import: CSV files in special formats
	Export: • CSV files in special formats • CSV files for NE OPC Server • Text files for CX-Designer • Text files for SPU-Console

11-12-2 Importing Mnemonics

Import

Reads mnemonic data stored in text files in proprietary formats.

1,2,3... 1. Select ***Mnemonic - Import*** in the Mnemonic Editor, specify the desired file name, and select ***Open***.

Export

Saves edited mnemonic data in a text file in proprietary formats.

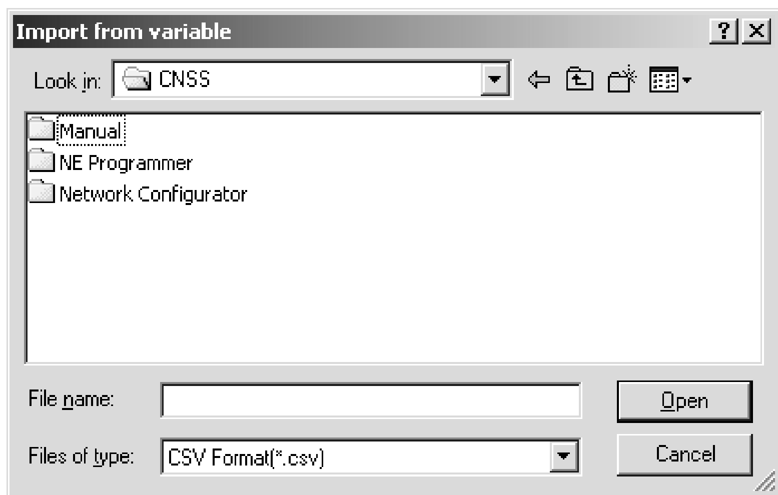
1,2,3... 1. Select ***Mnemonic - Export*** in the Mnemonic Editor, specify the desired file name, and select ***Save***.

11-12-3 Importing and Exporting Variables

Importing

Use to following procedure to import CSV files created on external Programming Devices, Excel, etc.

1,2,3... 1. Either select ***Variable - Import - CSV Format***. from the Menu Bar, or right-click the Local Variable Editor or Global Variable Editor and select ***Import - CSV Format*** from the popup menu.
The following dialog box will be displayed.

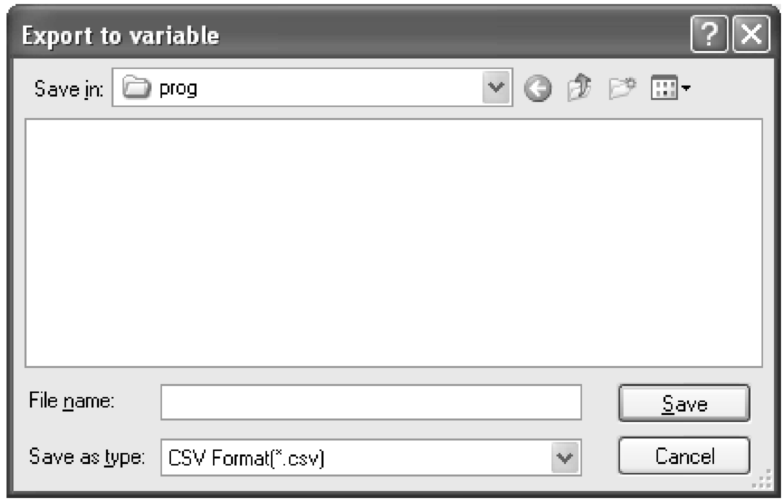


2. Specify the location and name of the file and click the **Open** Button.
The variables will be imported.

Exporting CSV Files

Variable data can be saved in CSV flies for use on external Programming Devices, Excel, etc.

1,2,3... 1. Either select ***Variable - Export - CSV Format***. from the Menu Bar, or right-click the Local Variable Editor or Global Variable Editor and select ***Export - CSV Format*** from the popup menu.
The following dialog box will be displayed.



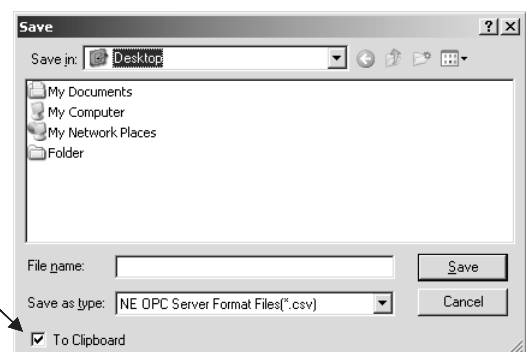
2. Specify the location and name of the file to save and click the **Save** Button. The variables will be saved in a CSV file.

Exporting in NE OPC Server, CX-Designer, or SPU-Console Format

Use the following procedure to save variable data to a CSV or text file for use with NE OPC Server, CX-Designer, or SPU-Console.

1,2,3...

1. Select ***File - Export Variables***. The Save Dialog Box will be displayed.



This option can be selected to output the variable file to the clipboard, so that the file can be exported by pasting.

2. Select the file format in the *Save as type* Field. Files can be saved as *NE OPC Server Format Files (*.csv)*, *CX Designer Format Files (*.txt)*, or *SPU Console Format Files (*.txt)*.
3. Specify the location and name of the file to save and click the **Save** Button. The variables will be saved in the specified application's format.

11-13 Printing

All kinds of project data can be printed.

11-13-1 Page Setup

1,2,3...

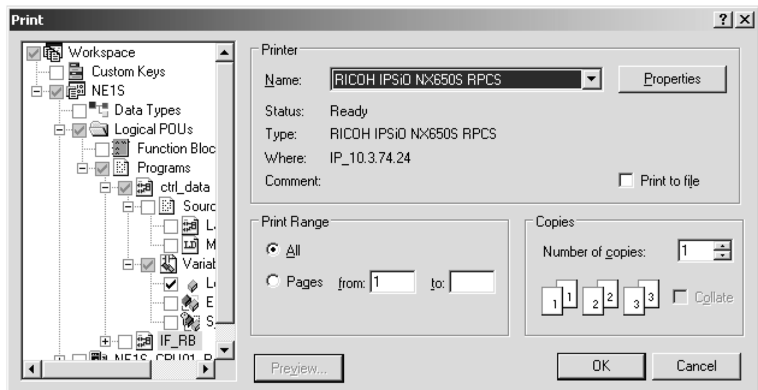
1. Select ***File - Page Setup***.

Tab page	Contents
Margin	Use to set the page margins in the *Top*, *Bottom*, *Left*, *Right*, *Header*, *and Footer* Fields.
Title Setting	Use to input the title text and set the number of lines, position, and font.

Tab page	Contents
Header	Header text can be input in the *Left*, *Center*, or *Right* Fields.
Footer	Footer text can be input in the *Left*, *Center*, or *Right* Fields.

11-13-2 Printing

1,2,3... 1. Select ***File - Print***.



2. Select the item to be printed in the Project Workspace at the left side of the Window. The contents of logical POUs (programs or function blocks), variables, or Custom Keys can be printed.

Note Sub-items can be added or removed from the print job by right-clicking the item and selecting or removing the sub-items (such as ladder diagrams or variable types) in the popup menu.

11-13-3 Print Preview

1,2,3... Click the **Preview** Button at the bottom of the Print Window or select ***File - Print Preview*** from the Menu Bar.

SECTION 12
PLC System Configuration

This section describes the configuration of the PLC system.

12-1 Overview

This section provides an overview of the settings and describes how to display the setting windows.

12-1-1 Settings

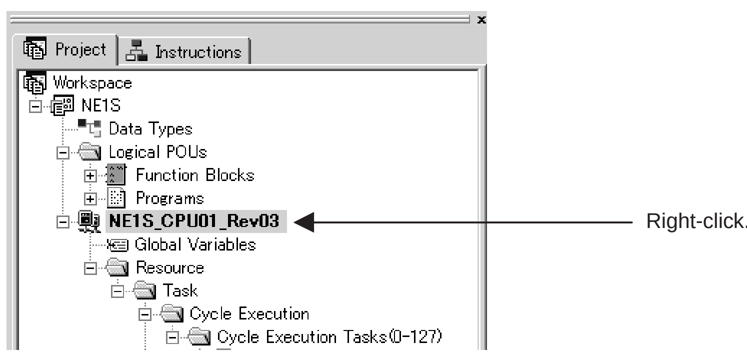
The following PLC system configuration settings can be made.

Setting tab name	Setting
General	Includes settings such as startup settings, CPU Unit settings, timer/interrupt settings, Special I/O Unit refresh settings, and communications settings.
Ethernet	Includes the built-in Ethernet Setup.
Build	Includes settings such as detailed timer/counter settings and IR/DR sharing settings for tasks.
I/O Table	Used to create I/O tables online or edit I/O tables offline.

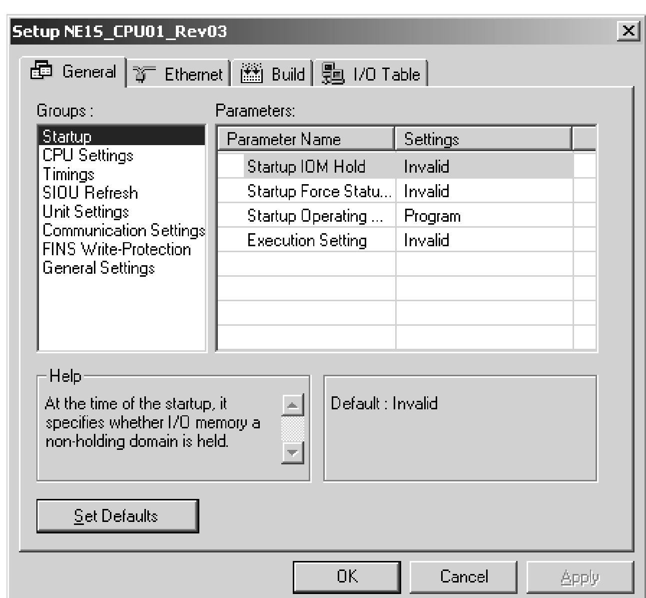
12-1-2 Displaying the Configuration Setting Window

Use the following procedure to display the System Configuration Window.

1,2,3... 1. Right-click the configuration name (i.e., the PLC name) in the Project Workspace.



2. Select ***System Configuration*** from the popup menu.
The following System Configuration Setting Window will be displayed.



The setting on each tab page are described in the following sections.

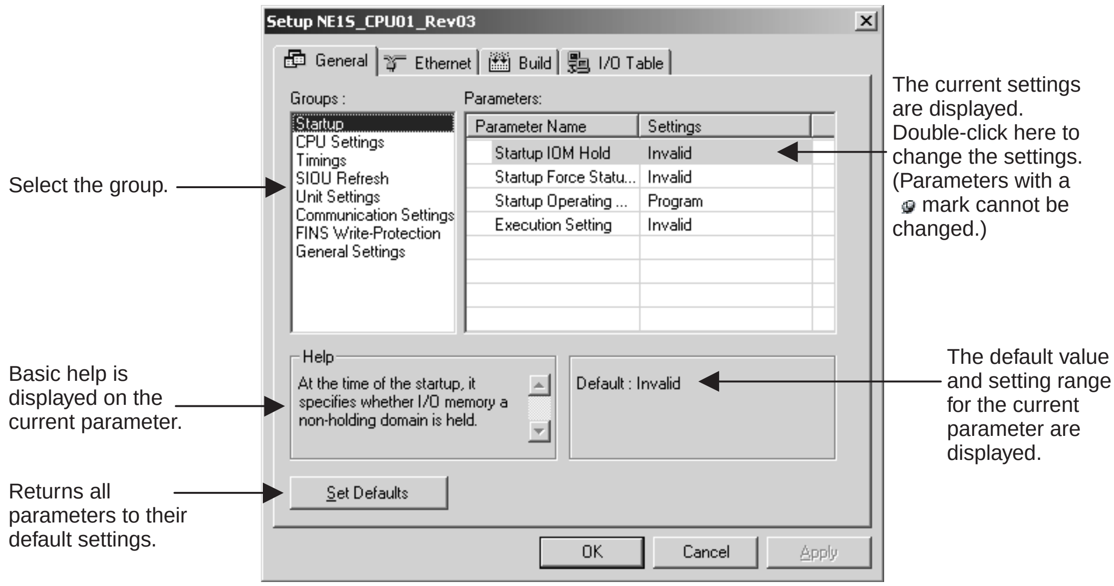
12-2 PLC Setup: General Tab Page

This section describes the setting procedure for the PLC Setup.

Refer to *4-1 General Tab Page* for details on the individual settings.

1,2,3...

1. Click the **General** Tab from the Configuration Setting Window. The General Tab Page, shown below, will be displayed.



2. Make all of the required settings and then click the **OK** Button.

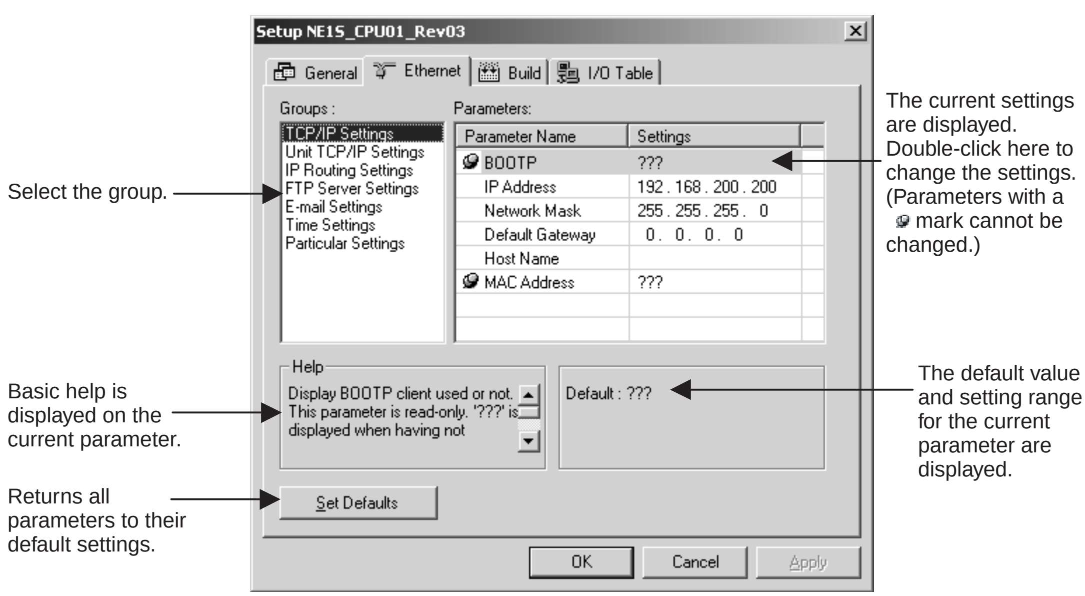
12-3 Ethernet Setup (Ethernet Tab page)

This section describes the setting procedure for the Ethernet Setup.

Refer to *4-2 Ethernet Settings* for details on the individual settings.

1,2,3...

1. Click the **Ethernet** Tab from the Configuration Setting Window. The Ethernet Tab Page, shown below, will be displayed.



2. Make all of the required settings and then click the **OK** Button.

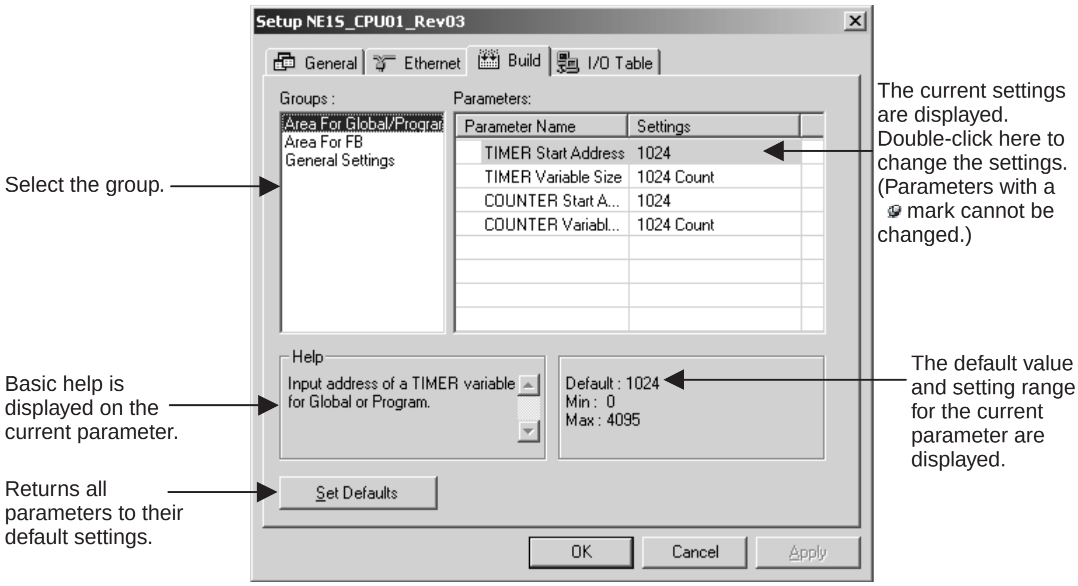
12-4 Build Settings (Build Tab Page)

This section describes the setting procedure for the Build Settings.

Refer to *4-3 Build Settings* for details on the individual settings.

1,2,3...

1. Click the **Build** Tab from the Configuration Setting Window.
   The Build Tab Page, shown below, will be displayed.



2. Make all of the required settings and then click the **OK** Button.

12-5 I/O Table Settings (I/O Table Tab Page)

This section describes the setting procedure for the I/O table settings.

Refer to *SECTION 5 Memory Areas* for details on I/O allocations.

I/O table settings can be perform online (mostly to create the I/O tables) or they can be performed offline (e.g., editing the I/O tables).

The I/O setting procedures for both online operation and offline operation are described in this section.

12-5-1 Online Operations: Creating, Deleting, Comparing

Creating I/O Tables

Use the following procedure to create I/O tables.

1,2,3...

1. Select ***Controller - Connect*** to connect online to the PLC.
2. Select ***Controller - I/O Table - Create***.
   The real I/O tables will be created.

Deleting I/O Tables

Use the following procedure to delete I/O tables.

1,2,3...

1. Select ***Controller - Connect*** to connect online to the PLC.
2. Select ***Controller - I/O Table - Delete***.
   The real I/O tables will be deleted.

Comparing I/O Tables

Use the following procedure to compare the real I/O tables and the registered I/O tables.

1,2,3...

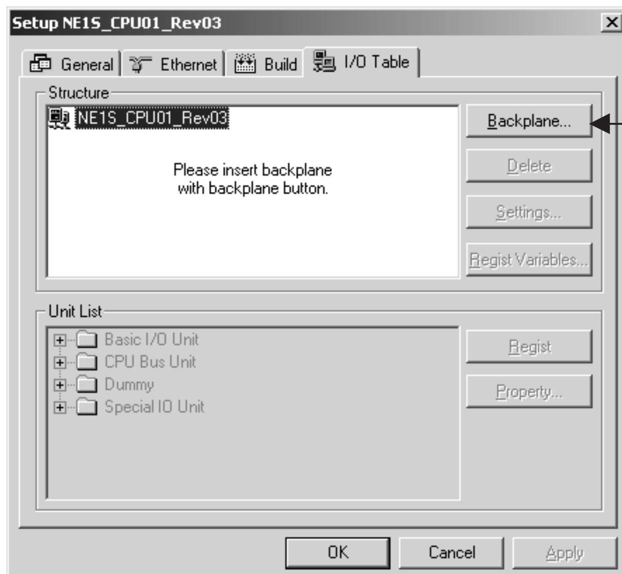
1. Select ***Controller - Connect*** to connect online to the PLC.
2. Select ***Controller - I/O Table - Compare***.
   The real I/O tables and the registered I/O tables will be compared.

12-5-2 Offline Operations: Editing I/O Tables

Use the following procedure to edit I/O tables offline.

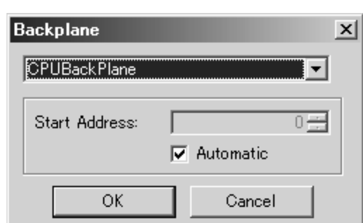
1,2,3...

1. Click the **I/O Table** Tab from the Configuration Setting Window.
   The following window will be displayed.



First, click here to add a CPU Backplane. Once the CPU Rack has been completed, click here again to add any Expansion Backplanes needed by the system.

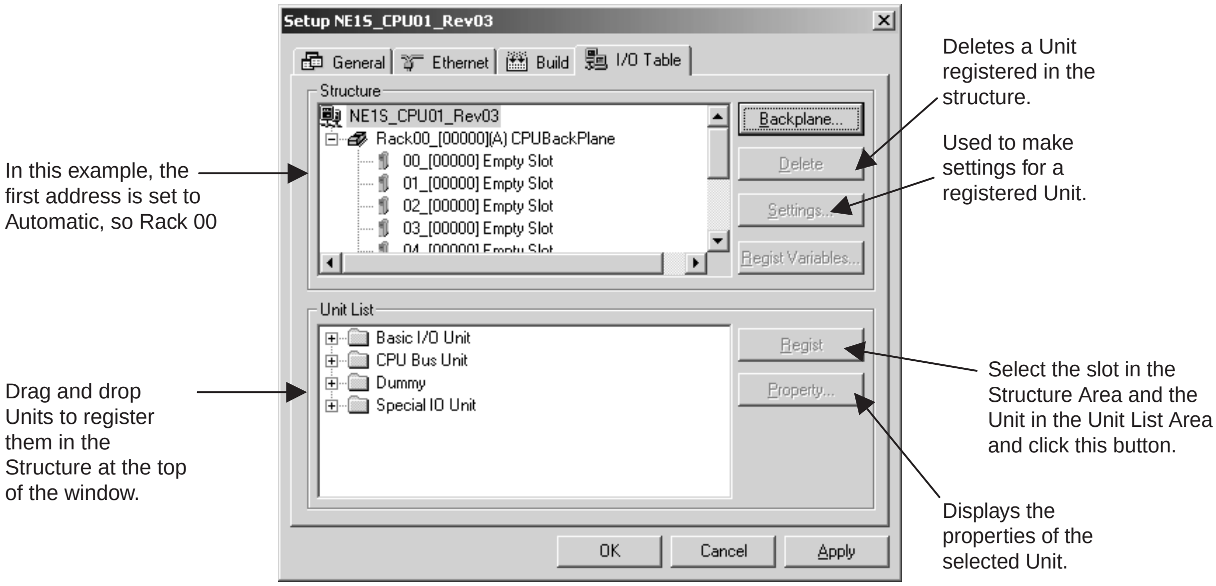
2. Click the **Backplane** Button.
   The Backplane Setting Dialog Box will be displayed.



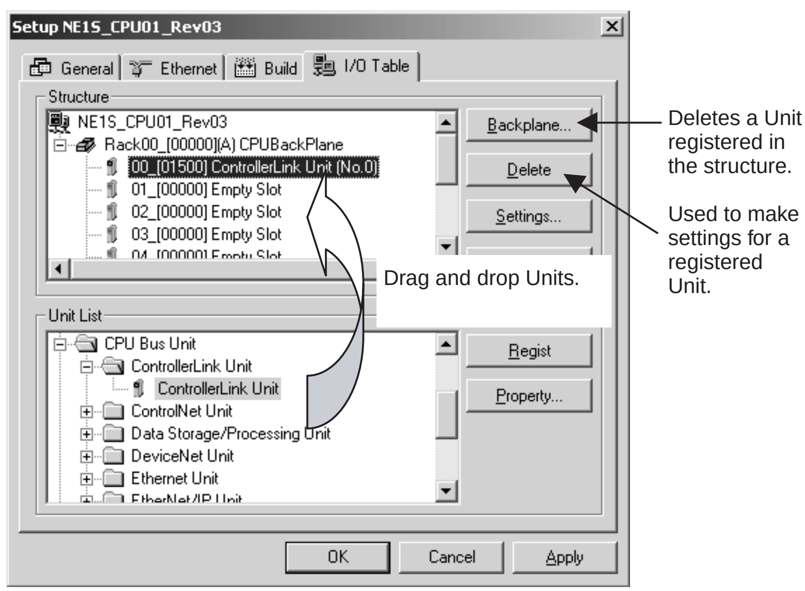
The CPU Backplane is added first. The display will automatically change to Expansion Backplanes after a CPU Backplane is added.

To set a specific start address, clear the check mark from *Automatic* and input the start address for the Rack.

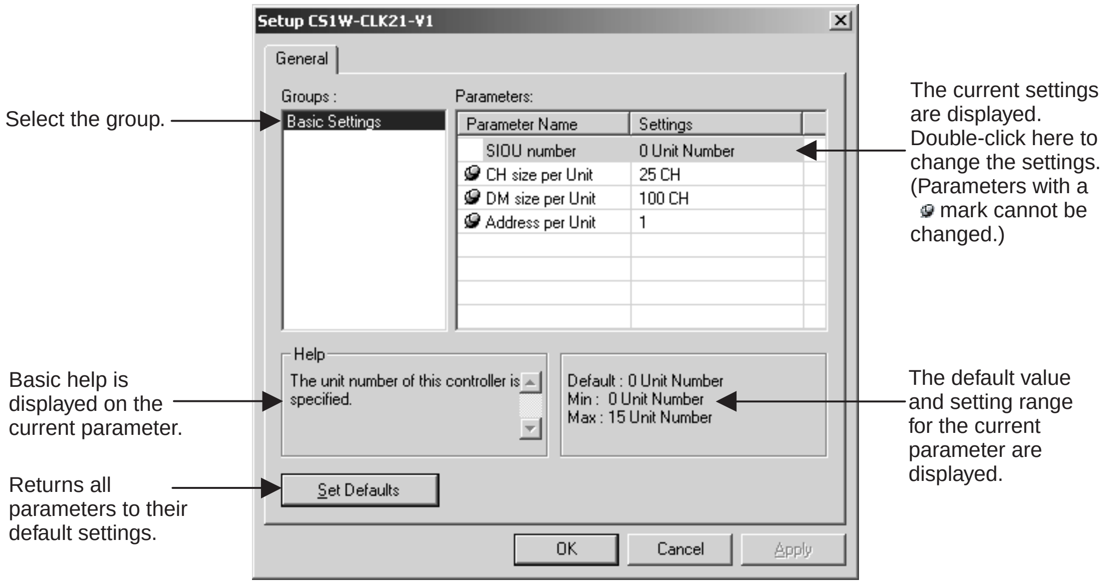
3. Click the **OK** Button.
   The following CPU Backplane Window will be displayed.



4. Drag and drop any of the Units from the *Unit List Area* to an *Empty Unit* in the *Structure* Area.
   The Unit will be registered in the slot.



5. Settings for the Unit must be made next. Click the **Setting** Button. The Unit Setting Dialog Box will be displayed.



6. Make all of the required settings and then click the **OK** Button. The I/O Table Tab Page will return.
7. Make all of the required settings and then click the **OK** Button.

SECTION 13
Online Operation

This section provides the procedures for online operation.

13-1 Connecting via Serial Communications (USB/RS-232C)

This section describes how to connect the NE Programmer online to the CPU Unit using serial communications (USB/RS-232C).

The first time USB is used to connect to the CPU Unit, the USB driver will need to be installed. To connect using RS-232C communications, go to *13-1-2*.

13-1-1 Installing the USB Driver

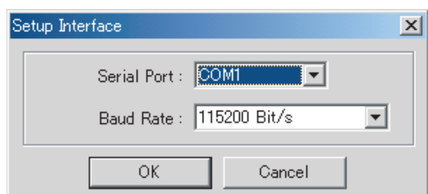
Refer to *8-3 Installing the USB Driver* for details on installing the USB driver.

13-1-2 Connecting Online via USB or RS-232C

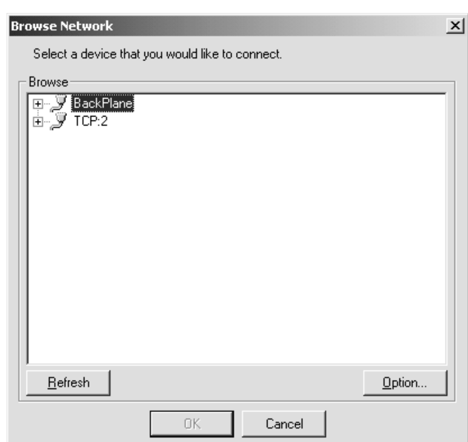
Use the following procedure to connect the NE Programmer online to the CPU Unit using USB or RS-232C communications.

1,2,3...

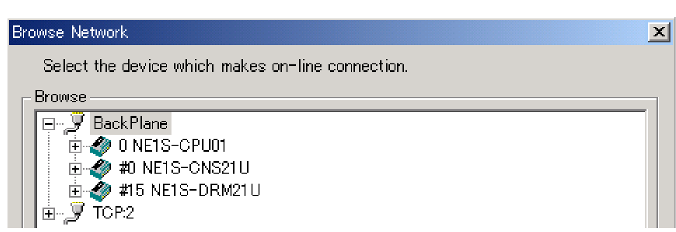
1. Select ***Option - Select Interface - NE1S Serial PORT***.
2. Select ***Controller - Connect***. Alternately, press the **Ctrl+W** Keys.
   The following dialog box will be displayed.



3. Select the USB or RS-232C port number for the *Serial Port* and then click the **OK** Button.
   The following dialog box will be displayed.

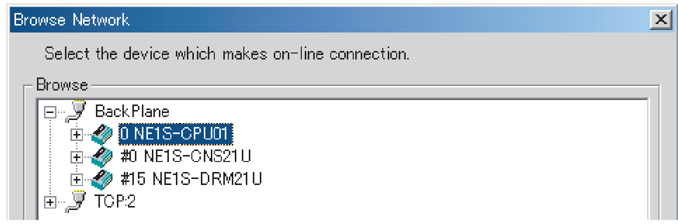


4. Click the **Refresh** Button.
   The CPU Unit and Communications Unit mounted to the CPU Rack will be displayed as shown below.

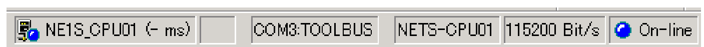


5. Select the ***NE1S-CPU01*** Icon as shown below and then click the **OK** Button.

   Note This connection path can be saved. (See the note at the end of this procedure.)



   If an online connection is made normally, *On-line* will be displayed in the status bar at the bottom of the window, as shown in the following figure.



6. To go offline again, select ***Controller - Disconnect***. Alternately, press the **Ctrl+Shift+W** Keys.

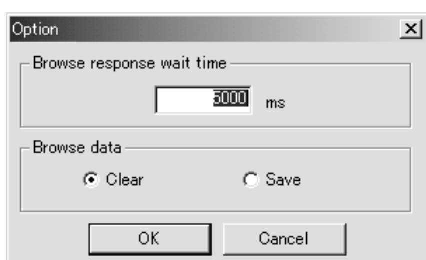
Note When connected online, there is a setting in the Browse Network Dialog Box to select whether or not to save the connection path.

- Clear (default)
  Each time the NE Programmer connects online, it is necessary to browse to a network path.
- Save
  The connection path is displayed from the last time that the NE Programmer was connected. In this case, the window in step 5 is displayed when step 3 is completed, so you can connect online to the last network path just by clicking the **OK** Button.

Use the following procedure to set this option.

1,2,3...

1. Click the **Option** Button in the lower-right corner of the Browse Network Dialog Box. The following window will be displayed.



2. Select *Clear* or *Save* and click the **OK** Button.

13-2 Connecting Online via Ethernet

This section describes how to connect the NE Programmer online to the CPU Unit using Ethernet.

Caution If the destination node address (IP address) is not set correctly, the NE Programmer may connect unexpectedly to another PLC, and invalid device parameters may be set. Always verify that the correct PLC is connected before downloading data.

13-2-1 Connecting Online via Ethernet

There are several methods that can be used when connecting online to the CPU Unit via Ethernet for the first time. Here, the IP address of the computer will be changed to make the online connection.

To change the IP address of the CPU Unit (i.e., to match an existing network address) after once having connected online to the CPU Unit via Ethernet, refer to *13-2-3 Changing the IP Address of the CPU Unit* of the CPU Unit.

1,2,3...

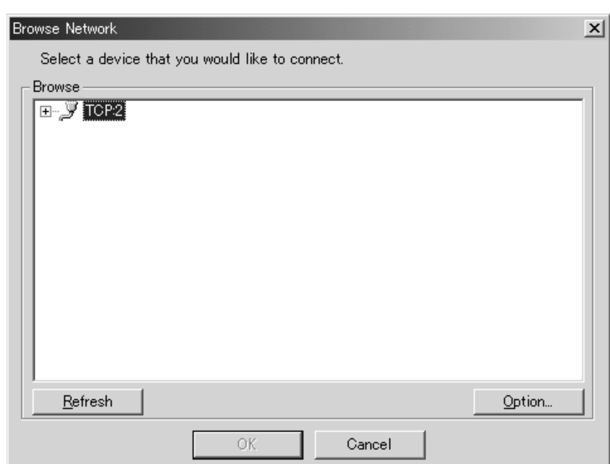
1. The default IP address of an NE1S CPU Unit is 192.168.200.200. In this example, we will assume the IP address of the computer will be changed to 192.168.200.10.
   Refer to the user's manual for your operating system for the method of changing the IP address of your computer.
2. Connect the CPU Unit and the computer to Ethernet.
3. Turn ON the power supply to the CPU Unit.

13-2-2 Connecting Online via Ethernet

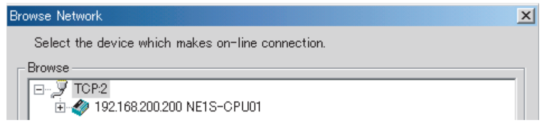
Use the following procedures to connect the NE Programmer online to the CPU Unit using Ethernet.

1,2,3...

1. Select ***Option - Select Interface - Ethernet I/F***.
2. Select ***Controller - Connect***. Alternately, press the **Ctrl+W** Keys.
   The following dialog box will be displayed.



3. Click the **Refresh** Button.
   The network will be browsed and the results of browsing will be displayed as shown in the following figure.



Note The default IP address of an NE1S CPU Unit is 192.168.200.200.

4. Select the ***192.168.200.200 NE1S-CPU01*** Icon as shown below and then click the **OK** Button.

Note (a) This connection path can be saved. (See the note at the end of this procedure.)

(b) NE1S CPU Units connected on different Ethernet segments from the segment to which the NE Programmer is connected will not be displayed in the Browse Network Dialog Box even if the **Refresh** Button is clicked. To connect online to an NE1S CPU Unit on a different segment, use the information provided below and input the IP address directly.

If an online connection is made normally, *On-line* will be displayed in the status bar at the bottom of the window, as shown in the following figure.

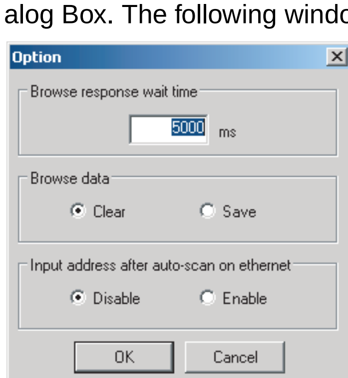
5. To go offline again, select ***Controller - Disconnect***. Alternately, press the **Ctrl+Shift+W** Keys.

Note When connected online, there is a setting in the Browse Network Dialog Box to select whether or not to save the connection path.

- Clear (default)
  Each time the NE Programmer connects online, it is necessary to browse to a network path.
- Save
  The connection path is displayed from the last time that the NE Programmer was connected. In this case, the window in step 5 is displayed when step 3 is completed, so you can connect online to the last network path just by clicking the **OK** Button.

Use the following procedure to set this option.

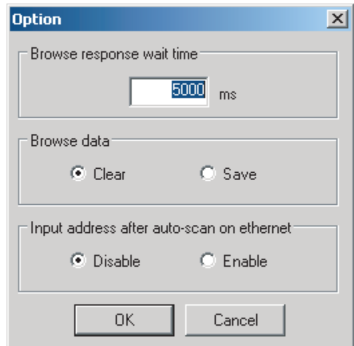
1,2,3... 1. Click the **Option** Button in the lower-right corner of the Browse Network Dialog Box. The following window will be displayed.



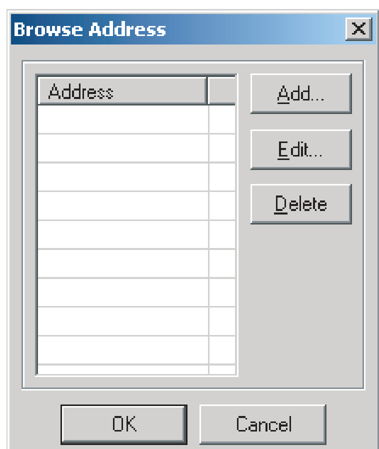
2. Select *Clear or Save in the Browse data* Area and click the **OK** Button.

Note To connect online to an NE1S CPU Unit on a different segment, use the following procedure to input the IP address directly.

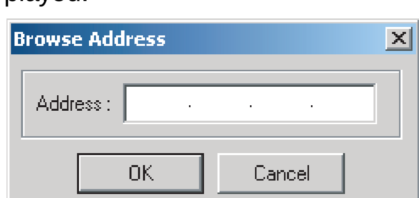
1. Click the **Option** Button in the Browse Network Dialog Box. The following dialog box will be displayed.



2. Select *Disable* in the *Input address after auto-scan on ethernet* Area and then click the **OK** Button.
3. Click the **Refresh** Button in the Browse Network Dialog Box.
4. The following Browse Address Dialog Box will be displayed along with the IP addresses found in the Browse Network Dialog Box.



5. Click the **Add** Button. The following Browse Address Dialog Box will be displayed.



6. Directly input the IP address in the *Address* Field.

13-2-3 Changing the IP Address of the CPU Unit

This section describes the procedure for changing the IP address of the CPU Unit. The IP address of the CPU Unit can be changed both on the rotary switches on the CPU Unit or in the Ethernet Setup in the NE Programmer.

If these settings do not agree, the setting on the rotary switches will take priority over the settings in the Ethernet Setup made from the NE Programmer.

Note IP Address Settings

- As a rule, the IP address is first set on the rotary switches on the CPU Unit first and then the setting on the Ethernet Tab Page under Controller - System Configuration, described later, is set to match the setting on the rotary switches.
- The network portion of the IP address, however, must be changed using the Ethernet Settings on the NE Programmer.
- If the rotary switches are set to 00, the IP address set with the NE Programmer will be used. Set the IP address switches on all CPU Units to 00 when you want to set and manage IP addresses using the Ethernet Settings on the NE Programmer.
- If the rotary switches are set to FF, the CPU Unit will use BOOTP.

CPU Unit IP Address Switch Settings

Setting	Meaning
00	• The IP address saved in the flash memory of the CPU Unit will be used at startup. • The defaults stored in flash memory are an IP address of 192.168.200.200, a network mask of 255.255.255.0, and a default gateway of 0.0.0.0 (no gateway). • If the IP address is changed on the Ethernet Tab Page under *Controller - System Configuration* of the NE Programmer and then downloaded to the CPU Unit, the changed settings will be stored in flash memory. The IP address saved in the flash memory of the CPU Unit will be used at startup. **Note** Set the IP address switches on all CPU Units to 00 when you want to set and manage IP addresses from the NE Programmer.
01 to FE	• Only the rightmost 8 bits of the IP address is set on the IP address switches on the CPU Unit. • The default IP address will be from 192.168.200.01 to 192.168.200.254, a network mask of 255.255.255.0, and a default gateway of 0.0.0.0 (no gateway). **Note** 1. The rightmost 8 bits of the IP address set on the Ethernet Tab Page under *Controller - System Configuration* of the NE Programmer will be ignored. 2. As a rule, set the rightmost 8 bits of the IP address on the rotary switches and then set the remaining portion of the IP address from the NE Programmer. The enables reading IP addresses directly off the rotary switches on the CPU Units onsite. • The network portion of the IP address, however, must be changed using the Ethernet Settings on the NE Programmer.
FF	The CPU Unit will use BOOTP. Refer to *4-2-3* for details on BOOTP. **Note** The settings in the Ethernet Setup made from the Ethernet Settings on the NE Programmer will be ignored if FF is set.

Note Any changes to the settings on the rotary switches or the Ethernet Setup from NE Programmer are effective the next time the CPU Unit is started.

Ethernet Setup

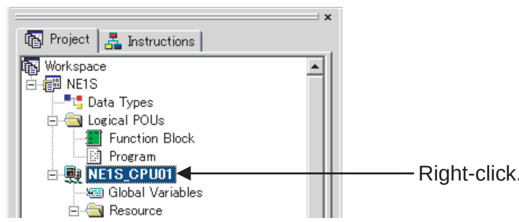
To change the network or host portion of the IP address, first change them in the Ethernet Setup (IP address settings) from the NE Programmer and then download the Ethernet Setup to the CPU Unit.

Change the rotary switches on the CPU Unit to 00 and then restart the CPU Unit. (If desired, the rightmost 8 bits of the IP address can be set on the rotary switches.)

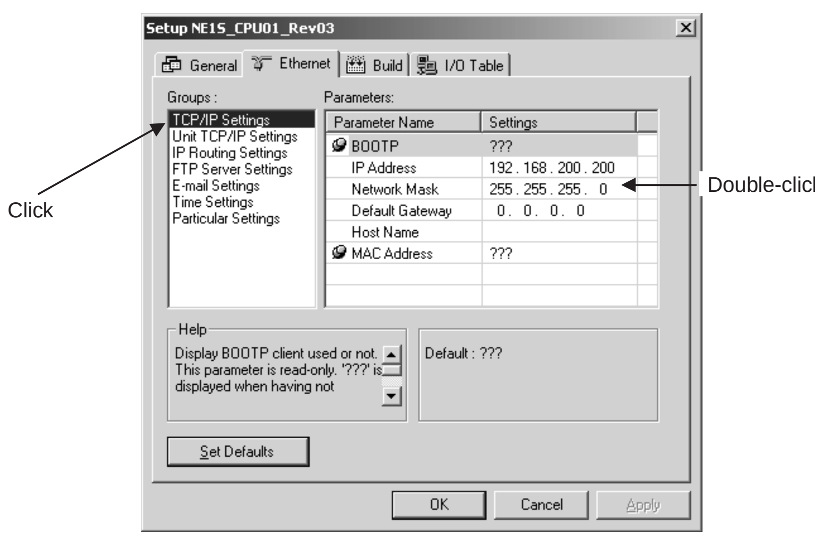
The setting on the rotary switch will take priority over the settings in the Ethernet Setup. Refer to *CPU Unit IP Address Switch Settings* for details on the rotary switches on the CPU Unit.

1,2,3...

1. Right-click the configuration name (i.e., the PLC name) in the Project Workspace. (Alternately, select ***Controller - System Configuration***.)



2. Select ***System Configuration*** from the popup menu.
   The System Configuration Setting Window shown in step 4 will be displayed.
3. Click the **Ethernet** Button.
4. Select ***TCP/IP Settings*** in the *Group* List.
   The following dialog box will be displayed to set TCP/IP.



5. Double-click *IP Address* and enter the IP address. If required, set the network mask.
6. Click the **OK** Button.

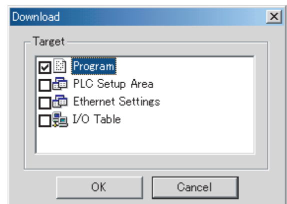
This completes setting the IP address. Refer to the following procedure *Downloading Ethernet Settings* and transfer the settings that have been made to the CPU Unit.

Downloading Ethernet Settings

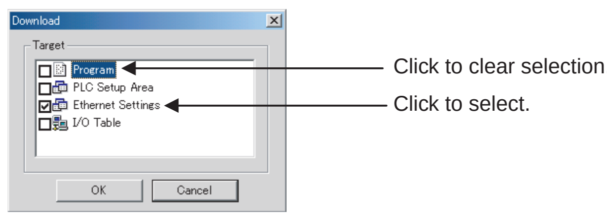
Use the following procedure to download the Ethernet settings online to the CPU Unit.

1,2,3...

1. Connect online to the CPU Unit. (A serial connection can be used if desired.)
2. Select ***Controller - Download***. Alternately, press the **Ctrl+T** Keys.
   The following dialog box will be displayed.



3. Select only the *Ethernet Settings* option (do not select any other options), and click the **OK** Button.



   The Ethernet settings will be downloaded. If the download ends normally, the following dialog box will be displayed.

Note The Ethernet settings are automatically backed up in flash memory inside the CPU Unit.
Do not turn OFF the power supply to the CPU Unit while the settings are being backed up.
The BKUP indicator on the front of the CPU Unit will light during the automatic backup operation.

4. Set the rotary switches on the front of the CPU Unit to 00 (factory setting) or to the rightmost 8 bits of the IP address. The value is set in hexadecimal. The setting on the rotary switch will take priority over the settings in the Ethernet Setup. Refer to *CPU Unit IP Address Switch Settings* for details on the rotary switches on the CPU Unit.
5. Restart the CPU Unit.

This completes changing the IP address of the CPU Unit.

13-3 Automatic Upload Function

13-3-1 Overview

If the personal computer and CPU Unit are directly connected via USB or RS-232C communications, the NE Programmer can be automatically connected online to the CPU Unit. After going online, you can automatically upload and monitor the program as well.

Refer to information starting in *13-6 Uploading, Downloading, and Comparing Programs and Other Data* and *13-8 Monitoring* for details on transferring and monitoring the program.

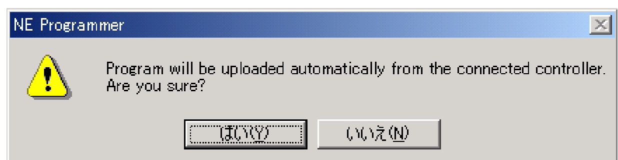
Note
(1) The automatic upload function cannot be used if another application is using the serial port.
(2) The automatic upload function can be used only with a direct serial connection.

13-3-2 Executing the Automatic Upload Function

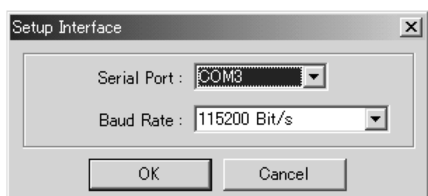
Use the following procedure to execute the automatic upload function.

Note The automatic upload function cannot be used if another application is using the serial port on the personal computer.

1,2,3...
1. Select ***Option - Select Interface - NE1S Serial Port***.
2. Select ***Controller - Auto Upload***.
   The following dialog box will be displayed.

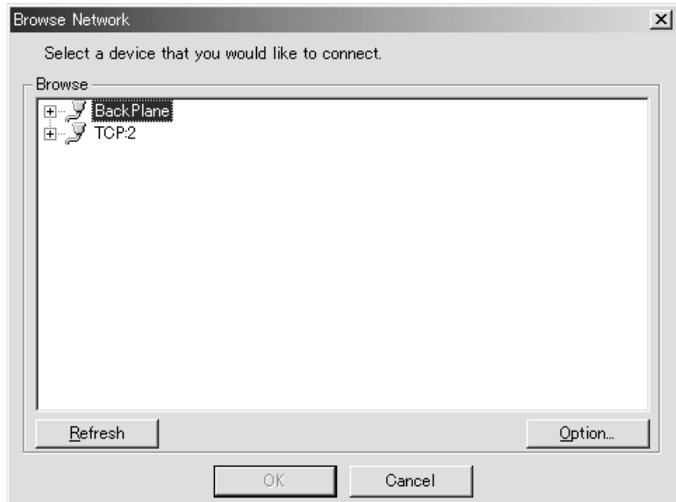


3. Click the **Yes** Button. The Setup Interface Dialog Box will be displayed.

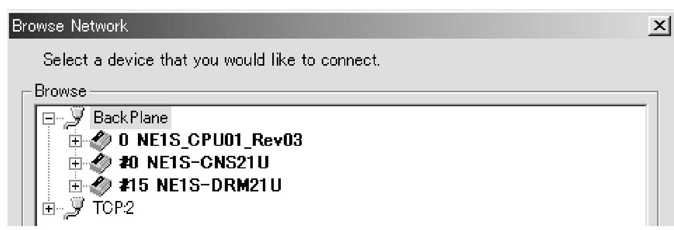


Note In some cases, this dialog box will not be displayed. If it is not displayed, proceed to step 4.
This software retains the interface settings from the previous upload (automatic upload or regular upload). When the **Yes** Button is clicked in step 3, the NE Programmer checks for a connection at this port. If there is a connection, the Setup Interface Dialog Box will not be displayed; the Browse Network Dialog Box (step 4) will be displayed instead.

4. Set the USB port's communications port and baud rate and click the **OK** Button. The Browse Network Dialog Box will be displayed.

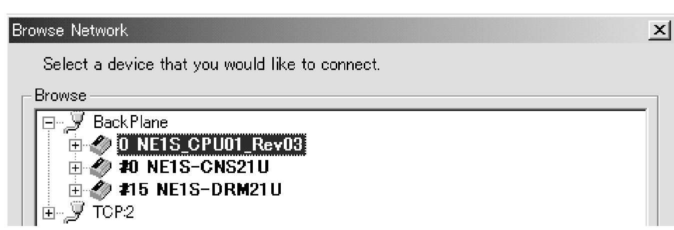


5. Click the **Refresh** Button.
   The CPU Unit and Communications Units mounted in the CPU Rack will be displayed, as shown in the following diagram.



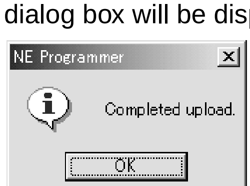
6. Select the ***NE1S-CPU01*** Icon as shown below and then click the **OK** Button.

   Note This connection path can be saved. (See the note at the end of this procedure.)



- If there is unsaved data, a dialog box will be displayed that prompts you to save the data. Follow the directions and save the data.

  After connecting online to the CPU Unit, the program, PLC Setup, Ethernet settings, and I/O table will be uploaded from the CPU Unit. The following dialog box will be displayed when the data has been uploaded.



7. Click the **OK** Button.

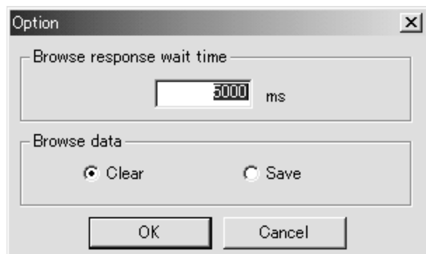
Note When connected online, there is a setting in the Browse Network Dialog Box to select whether or not to save the connection path.

- Clear (default)
  Each time the NE Programmer connects online, it is necessary to browse to a network path.
- Save
  The connection path is displayed from the last time that the NE Programmer was connected. In this case, the window in step 5 is displayed when step 3 is completed, so you can connect online to the last network path just by clicking the **OK** Button.

Use the following procedure to set this option.

1,2,3...

1. Click the **Option** Button in the lower-right corner of the Browse Network Dialog Box. The following window will be displayed.



2. Select *Clear* or *Save* and click the **OK** Button.

13-4 Changing the CPU Unit That Is Connected

13-4-1 Overview

Use this function to connect to a different CPU Unit when the NE Programmer is already connected online to a CPU Unit. When the CPU Unit connection is changed, the automatic upload function will be executed after connecting to the new CPU Unit.

This function eliminates the need to for the user to disconnect from one CPU Unit and then upload data after connecting to a different CPU Unit.

There are two ways to change the CPU Unit:

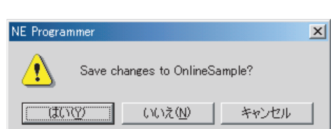
- Changing to a CPU Unit on the same network. For example, changing to a CPU Unit connected to the same Ethernet network as the CPU Unit that is currently connected. Refer to *13-4-2*.
- Changing to a CPU Unit on a different network. For example, changing to a CPU Unit connected to an Ethernet network when a CPU Unit is currently connected via USB. Refer to *13-4-3*.

13-4-2 Changing to a CPU Unit on the Same Ethernet Network

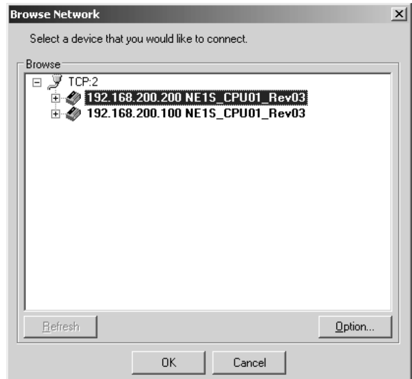
Use the following procedure to change the connection to a CPU Unit connected to the same Ethernet network as the CPU Unit that is currently connected.

1,2,3...

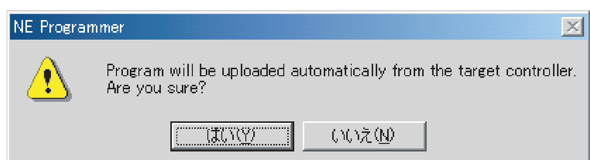
1. Select ***Controller - Change Controller***.
   The following dialog box will appear if there is any unsaved data.



2. Click the **Yes** Button and save the data on the NE Programmer if the data is required.
   Click the **No** Button is the data is not required.
   - The Browse Network Dialog Box will be displayed.



   - If the IP address of the CPU Unit to be connected is not displayed, click the **Refresh** Button.
3. Select the IP address of the CPU Unit to be connected.
4. Click the **OK** Button.
   The following dialog box will be displayed.



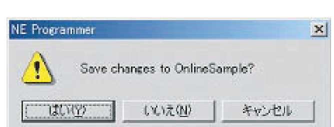
5. Click the **Yes** Button.
   The automatic upload function will be performed after connecting to the specified CPU Unit. The following dialog box will be displayed if the automatic upload function is executed normally.

13-4-3 Changing to a CPU Unit on a Different Network

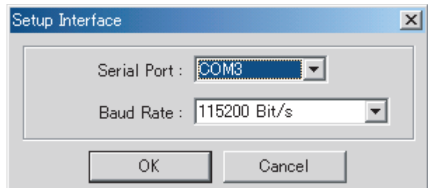
Use the following procedure to change the connection to a CPU Unit connected to an Ethernet network when a CPU Unit is currently connected via USB.

1,2,3...

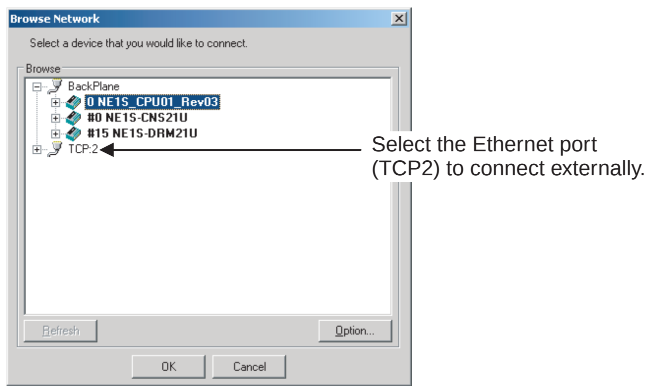
1. Select ***Controller - Change Controller***.
   The following dialog box will be displayed if there is any unsaved data.



2. Click the **Yes** Button and save the data on the NE Programmer if the data is required.
   Click the **No** Button is the data is not required.
   - The Serial Interface Dialog Box will be displayed.



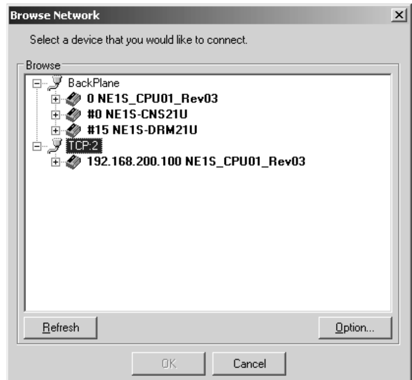
3. The serial port and baud rate being used for the current connection will be displayed. Check the settings and then click the **OK** Button.
   The Browse Network Dialog Box will be displayed.



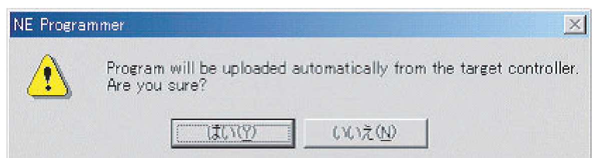
4. Select ***TCP:2*** (Ethernet port) and then click the **Refresh** Button.
   The Browse Address Dialog Box will be displayed.

5. Click the **Add** Button and input the IP address in the dialog box that will be displayed. The Browse Address Dialog Box will return.
   The IP address that was input will be displayed as shown below.

6. Click the **OK** Button.
   The IP address that was input will be displayed in the Browse Network Dialog Box.



7. Select the IP address of the CPU Unit to be connected.
8. Click the **OK** Button.
   The following dialog box will be displayed.



9. Click the **Yes** Button.
   The automatic upload function will be performed after connecting to the specified CPU Unit. The following dialog box will be displayed if the automatic upload function is executed normally.

13-5 Online Operations for I/O Tables

This section describes creating, deleting, and comparing I/O tables online.

13-5-1 I/O Tables

I/O tables are tables that list the models and locations of the Units mounted in the PLC.

The I/O tables registered in the CPU Unit are used to allocate I/O memory in the CPU Unit to real I/O (i.e., Basic I/O Units), Special I/O Units, and CPU Bus Units.

Always create I/O tables and register them in the CPU Unit after adding any new Unit to the PLC or after removing any Unit from the PLC.

13-5-2 Real I/O Tables and Registered I/O Tables

Real I/O Tables

Real I/O tables are tables that list the models and locations of the Units actually mounted in the PLC.

When the I/O table creation operation is performed, the real I/O tables are registered in the CPU Unit as the registered I/O tables. The real I/O tables cannot be changing from the NE Programmer.

Registered I/O Tables

The I/O tables registered in the CPU Unit are used by the CPU Unit when allocating I/O. Use either of the following procedures to register I/O tables in the CPU Unit.

- Execute the I/O table creation operation online to register the real I/O tables in the CPU Unit as the registered I/O tables.
- Create and edit I/O tables offline and then transfer them to the CPU Unit.

13-5-3 Creating I/O Tables

Use the following procedure to create the real I/O tables online.

1,2,3...

1. Connect online to the CPU Unit.
2. Select Controller - I/O Table - Create.
   The following dialog box will be displayed.

3. Click the **Yes** Button.
   The real I/O tables will be created.

13-5-4 Deleting I/O Tables

Use the following procedure to delete the registered I/O tables online.

1,2,3...

1. Connect online to the CPU Unit.
2. Select ***Controller - I/O Table - Delete***.
   The following dialog box will be displayed.

3. Click the **Yes** Button.
   The real I/O tables will be created.

13-6 Uploading, Downloading, and Comparing Programs and Other Data

13-6-1 Overview

The following data can be uploaded to the NE Programmer, downloaded to the CPU Unit, or compared between the CPU Unit and NE Programmer: I/O tables, PLC Setup, and Ethernet Setup.

The CPU Unit operating modes during which uploading, downloading, and comparing are possible are shown in the following table.

Menu command	RUN	MONITOR	PROGRAM
Controller - Upload from Controller	Yes	Yes	Yes
Controller - Download to Controller	No	No	Yes
Controller - Compare with Controller	Yes	Yes	Yes

⚠ Caution Confirm safety at the destination node before editing or transferring a program, PLC Setup, I/O table, I/O memory data, or parameter data to another node. Doing either of these without confirming safety may result in unexpected operation and injury.

⚠ Caution Before actual operation, check the parameter settings and user program (such as the ladder program) for proper execution in trial operation. Always check the program before transferring it.

Note (1) Confirm that the controlled system will not be adversely affected before changing the operating mode of the CPU Unit.

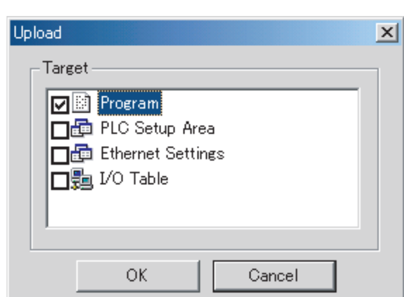
(2) Valuable programs may be lost if the direction of program transfer is not correct. Double-check the transfer direction before transferring data.

Perform all of the following operations online. Before starting, connect the NE Programmer online to the CPU Unit.

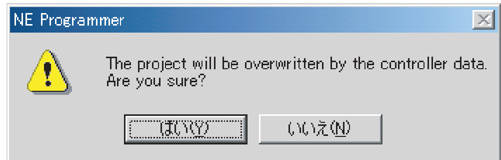
13-6-2 Uploading

Use the following procedure to upload any of the following from the CPU Unit to the NE Programmer: Program, PLC Setup, Ethernet settings, and I/O tables.

1,2,3... 1. Select ***Controller - Upload from Controller***.
The following dialog box will be displayed.



2. Select the data to be uploaded and then click the **OK** Button.
   The following dialog box will be displayed.



3. Check to be sure that is it okay to overwrite the selected data (in step 2) in the NE Programmer and then click the **Yes** Button.
   The data selected in step 2 will be uploaded from the CPU Unit to the NE Programmer.
   The following dialog box will be displayed if the data is uploaded normally.

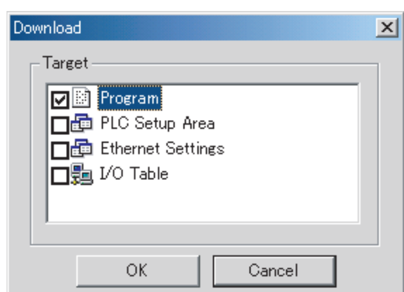
4. Click the **OK** Button.

13-6-3 Downloading

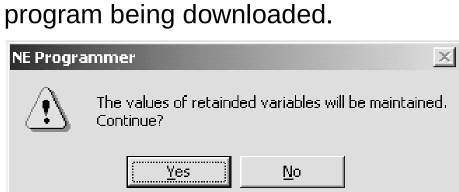
Use the following procedure to download any of the following from the NE Programmer to the CPU Unit: Program, PLC Setup, Ethernet settings, and I/O tables.

1,2,3...

1. Select ***Controller - Download to Controller***.
   The following dialog box will be displayed.



2. Select the data to be downloaded and then click the **OK** Button.
   The following dialog box will be displayed if there are already retained variables in the CPU Unit with the same data type as retained variables in a program being downloaded.



   Refer to *Preserving the Values of Retained Variables* below for details on preserving the PVs of retained variables.

3. Select the data to be downloaded and then click the **OK** Button.
   The selected data selected will be downloaded from the NE Programmer to the CPU Unit. The following dialog box will be displayed if the data is downloaded normally.

4. Click the **OK** Button.

Preserving the Values of Retained Variables

If there are already retained variables in the CPU Unit with the same data type as retained variables in a program being downloaded, the retained variables in the CPU Unit can be preserved.

Note Observe the following precautions.

- The force-set/force-reset status of automatically allocated variables will not be preserved.
- The force-set/force-reset status of automatically allocated variables is cleared when the CPU Unit is switched from PROGRAM mode to MONITOR or RUN mode.
- If the program is downloaded when automatically allocated variables have been force-set/force-reset, the variables may have unexpected force-set/force-reset status after the download is completed. We recommend clearing the forced status of all automatically allocated variables before downloading the program.

This list shows various requirements for the PVs of retained variables to be preserved after downloading. There is a table following the list with examples for cases *1 to *10.

- Variables in the downloaded program are set to be retained and automatically allocated. Variables with address specifications or non-retained variables are not preserved. (*1)
- The variable names are being used in the program in the Controller before the download. (*2, *3, *4)
- The variables and data types in the Controller before the download match the ones in the program being downloaded. (*5)
  - For structures, the data types match for each member. Also, members are valid even if the member's declaration location changed. (*6)
  - For array variables, it depends on whether the number of elements increased or decreased. If the number of elements decreased, all variables will be valid. If the number of elements increased, only the number of elements before the download will be valid. (*7)
- There are no changes to task allocations or nesting of function block instances. (*8, *9, *10)
- The variables are not written from another source while the download is in progress.

When a variable has been overwritten from another application, the variable may be returned to its previous value.

	Status	Information in program (downloaded information)	Information in Controller (information before download)	PV status after download
1	Retain/Non-retain and Address specification	BOOL VarA (with address specification)	BOOL VarA (automatic allocation/retain/non-retain)	Not retained
		BOOL VarA (with address specification)	BOOL VarA (with address specification)	Not retained
		BOOL VarA (automatic allocation/retain)	BOOL VarA (with address specification)	Retained
		BOOL VarA (automatic allocation/retain)	BOOL VarA (automatic allocation/non-retain)	Retained
		BOOL VarA (automatic allocation/non-retain)	BOOL VarA (automatic allocation/retain)	Not retained
2	Variable exists in both project and Controller.	VarA	VarA	Retained
3	Variable does not exist in Controller.	VarA	None	Not retained
4	Variable does not exist in project.	None	VarA	Not retained
5	Data types match.	BOOL VarA	BOOL VarA	Retained
6	Data types do not match.	BOOL VarA	WORD VarA	Not retained
		DWORD VarA	WORD VarA	Not retained
		TYPE stA: STRUCT Member1: BOOL; Member2: WORD; Member3: UINT; Member4: DWORD; END_STRUCT; END_TYPE VAR VarA: stA; END_VAR;	TYPE stA: STRUCT Member2: BOOL; Member1: WORD; Member3: UDINT; END_STRUCT; END_TYPE VAR VarA: stA; END_VAR;	 Retained Retained Not retained Not retained
		TYPE stA: STRUCT Member2: BOOL; Member1: WORD; Member3: UDINT; END_STRUCT; END_TYPE VAR VarA: stA; END_VAR;	TYPE stA: STRUCT Member1: BOOL; Member2: WORD; Member3: UINT; Member4: DWORD; END_STRUCT; END_TYPE VAR VarA: stA; END_VAR;	 Retained Retained Not retained Not retained
7	Number of array elements is different.	BOOL VarA[3] VarA[0] VarA[1] VarA[2]	BOOL VarA[3] VarA[0] VarA[1]	 Retained Retained Not retained
		BOOL VarA[3] VarA[0] VarA[1]	BOOL VarA[3] VarA[0] VarA[1] VarA[2]	 Retained Retained Not retained

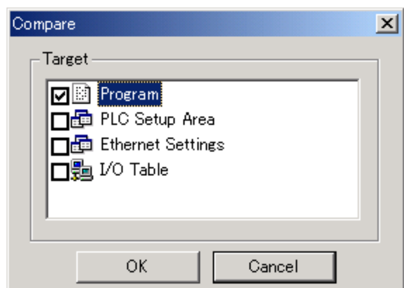
	Status	Information in program (downloaded information)	Information in Controller (information before download)	PV status after download
8	Task allocation was changed.	\Cyclic task (0 to 127) 0 \VarA	\Cyclic task (0 to 127) 1 \VarA	Not retained
9	FB instance name was changed.	\Cyclic task (0 to 127) 0 \InstA\VarA	\Cyclic task (0 to 127) 0 \InstB\VarA	Not retained
10	FB instance nesting was changed.	\Cyclic task (0 to 127) 0 \InstA\VarA	\Cyclic task (0 to 127) 0 \InstA\InstB\VarA	Not retained

13-6-4 Comparing

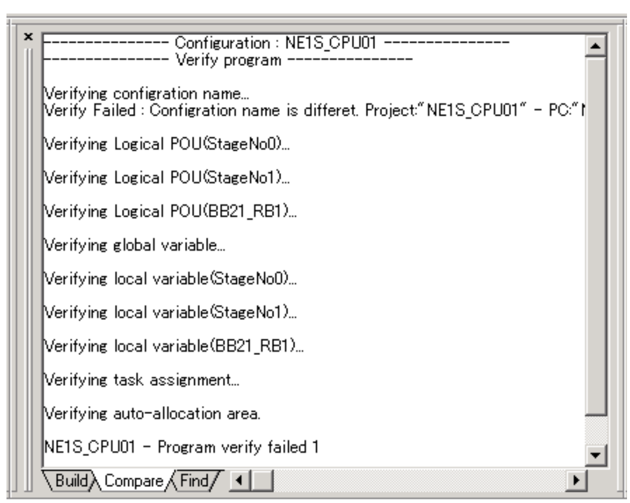
Use the following procedure to copy any of the following between the NE Programmer and the CPU Unit: Program, PLC Setup, Ethernet settings, and I/O tables.

1,2,3...

1. Select ***Controller - Compare with Controller***.
   The following dialog box will be displayed.



2. Select the data to be compared and then click the **OK** Button.
   The selected data selected will be compared between the NE Programmer and the CPU Unit.
   The results of comparison will be displayed on the Compare Tab Page of the Output Window as shown below.



13-7 Changing the Operating Mode

Use the following menu commands to change the operating mode of the CPU Unit from the NE Programmer.

Note Confirm that the controlled system will not be adversely affected before changing the operating mode of the CPU Unit.

Operating mode	Menu command
PROGRAM	Controller - Operating Mode - Program
MONITOR	Controller - Operating Mode - Monitor
RUN	Controller - Operating Mode - Run

13-8 Monitoring

13-8-1 Overview

The monitoring function enables ladder program execution to be monitored in a window.

The NE Programmer supports the following two types of monitoring.

- Monitoring status and present values on the Ladder Editor
- Monitoring present values of specified variables (or addresses) in the I/O memory of the CPU Unit in the Watch Window.

The following operations are used in the above monitoring windows.

- Status/PV Monitor
- Force ON/Force OFF
- Differential Monitor (detecting ON to OFF and OFF to ON transitions in bit status)
- PV Change
- Program Change while Monitoring: Refer to *13-14 Online Editing*.
- Timer/Counter SV Change: Refer to *13-14 Online Editing*.

⚠ **Caution** Confirm safety sufficiently before starting to monitor status or PVs on the Ladder Editor or in the Watch Window.
Operating errors, e.g., of shortcut keys that result in forcing ON or OFF bits or turning ON or OFF bits may result in operating errors in the controlled system connected to Output Units regardless of the operating mode of the CPU Unit.

Relationship between Monitoring Types and Operations

Operation	Type of monitoring		Operating mode of CPU Unit
	Ladder Editor	Watch Window	
Status/PV Monitor	Yes	Yes	All modes
Force ON/Force OFF	Yes	Yes	All modes except RUN
Differential monitor	Yes	Yes	All modes
PV Change for timers, counters, I/O areas, DM Area, or EM Area	Yes	Yes	All modes except RUN
Online editing (Including Timer/Counter SV Change)	Yes	No	All modes except RUN

13-8-2 Starting Monitoring Functions

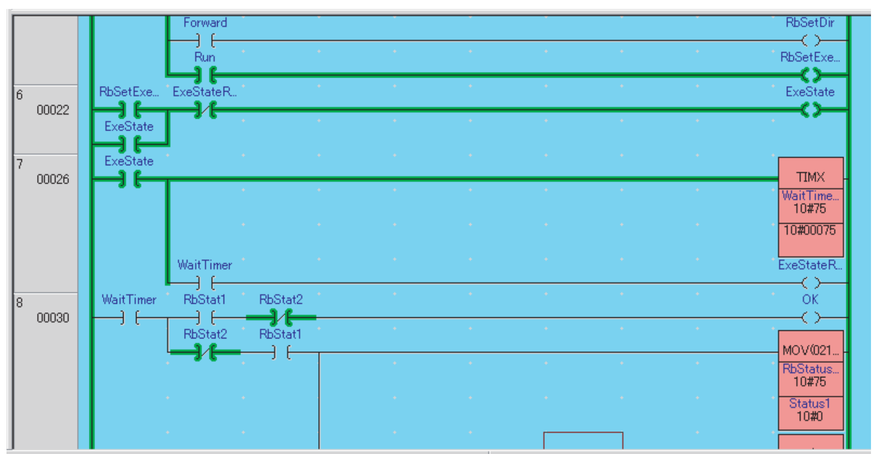
When the NE Programmer is connected online to a CPU Unit, monitoring will automatically be started in the Ladder Editor and Watch Window.

Note (1) The program in the NE Programmer and the program in the CPU Unit must be the same to enable monitoring. If they are not the same, transfer the program. For details on transferring programs, refer to *13-6 Uploading, Downloading, and Comparing Programs and Other Data*.

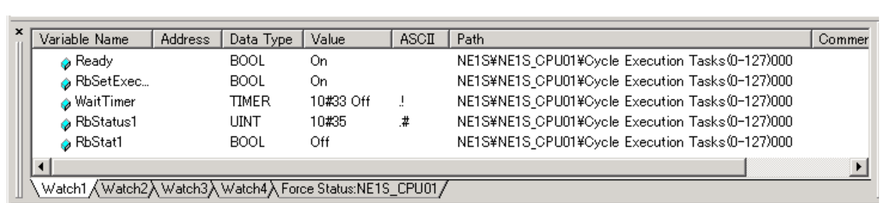
(2) The automatic upload function can be used to automatically upload the program from the CPU Unit and start monitoring. (The automatic upload function can be used only when the NE Programmer and CPU Unit are connected directly via USB or RS-232C communications. Refer to *13-3 Automatic Upload Function* for details on the automatic upload function.

1,2,3...

1. Connect the NE Programmer online to the CPU Unit.
2. Double-click the task to be monitored in the Project Window to display the Ladder Editor.
   Monitoring will automatically be started in the Ladder Editor and Watch Window.



- ON execution status (i.e., the power flow) in the ladder diagram will be indicated in light green. (Light green is the default color.)
- If the CPU Unit is in MONITOR or RUN mode and the current task is active, the background will be light blue. (Light blue is the default color.)



Variable Name	Address	Data Type	Value	ASCII	Path	Commer
Ready		BOOL	On		NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	
RbSetExec...		BOOL	On		NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	
WaitTimer		TIMER	10#33 Off	!	NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	
RbStatus1		UINT	10#35	#	NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	
RbStat1		BOOL	Off		NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	

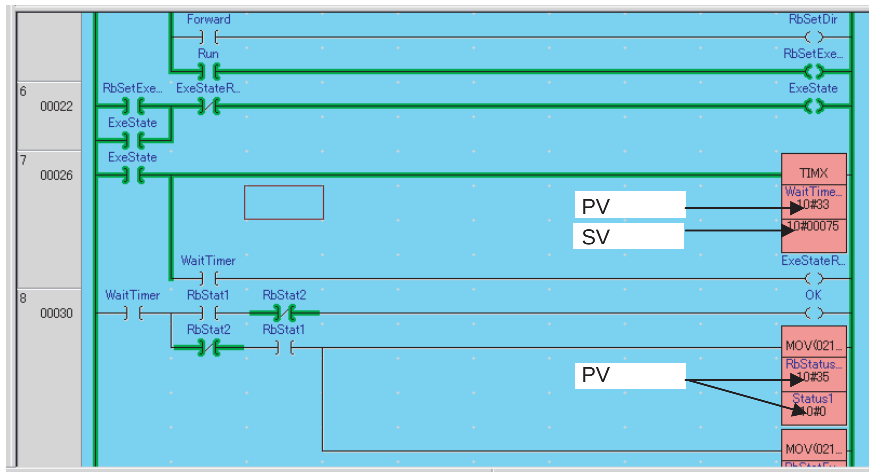
- If the Watch Window is not currently displayed, select ***View - Window - Watch***.

Operations that can be performed in the Ladder Editor or Watch Window during monitoring are described in the following sections.

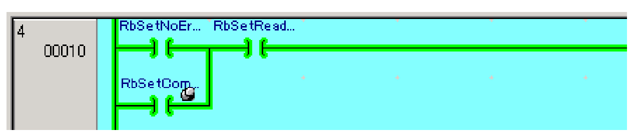
13-8-3 Monitoring in the Ladder Editor

In online status, double-click the task to be monitored in the Project Window to display the Ladder Editor.

The following Ladder Editor Status Monitor Window will be displayed.



- PV: Present value, SV: Set value
- Depending on the operating status of the CPU Unit, ON execution status (i.e., the power flow) in the ladder diagram will be indicated in light green. (Light green is the default color.)
- If the CPU Unit is in MONITOR or RUN mode, the background will be light blue. (Light blue is the default color.)
- To change the display colors, select ***Tool - Option*** to display the Integrated Options Window, click the **Ladder** Icon, and click the **Color** Tab.
- The icon shown below will be displayed to indicated forced status if a bit is forced ON or forced OFF.

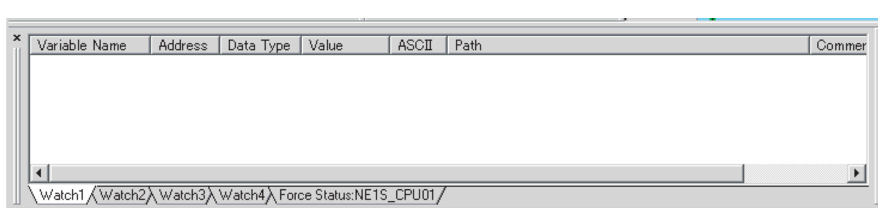


13-8-4 Monitoring in the Watch Window

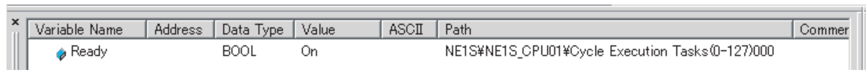
Monitoring by Designating Variables

1,2,3...

1. In online status, double-click the task to be monitored in the Project Window to display the Ladder Editor.
2. If the Watch Window is not currently displayed, select ***View - Window - Watch***.
   The Watch Window will be displayed.

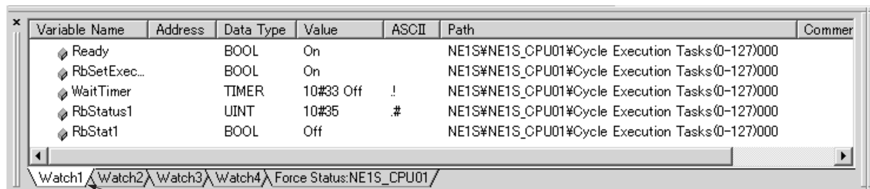


3. In the Ladder Editor, right-click the variable (bit or word) to be monitored and select **Add to Watch** from the popup menu.
   The specified bit or word will be added to the Watch Window and it's present value will be automatically monitored as shown below.



Variable Name	Address	Data Type	Value	ASCII	Path	Commer
Ready		BOOL	On		NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	

4. Repeat the previous step to monitor other bits or words at the same time.
   Display Example



Variable Name	Address	Data Type	Value	ASCII	Path	Commer
Ready		BOOL	On		NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	
RbSetExec...		BOOL	On		NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	
WaitTimer		TIMER	10#33 Off	!	NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	
RbStatus1		UINT	10#35	#	NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	
RbStat1		BOOL	Off		NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	

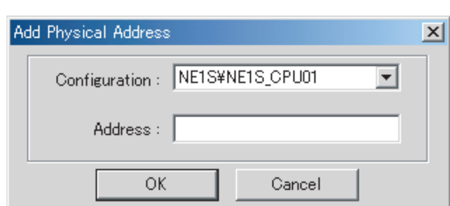
Right-click the tab to select *Create Tab*, *Delete Tab*, *Move*, or *Rename.*

- *"S On"* will be displayed for bits that are forced ON and *R Off* will be displayed for bits that are forced ON.
- If the project is saved, information on the variables that have been added to the Watch Window will be saved as well.
- The tab pages from *Watch 1* to *Watch 4* can be used to group the variables that are being monitored.
- Any variables that have been forced ON or OFF will be displayed on the *Force Status* Tab Page.
  Execute ***Update Force Status*** the first time the Force Status Tab Page is selected. (A dialog box will be displayed automatically when the tab is clicked.)
  Once a bit has been forced ON or OFF, it will remain on the Force Status Tab Page even if its forced status is cleared. To remove a bit from the Force Status Tab Page, right-click the Watch Window and select ***Update Status***.

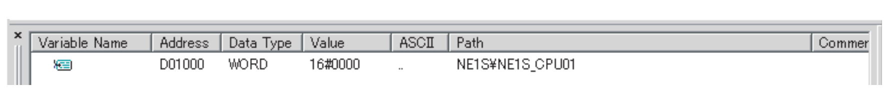
Monitoring by Designating Addresses

1,2,3...

1. Right-click the Watch Window and select ***Add Address*** from the popup menu.
   The ***Add Physical Address*** Dialog Box will be displayed.



2. Input the address to be monitored and then click the **OK** Button.
   The specified address will be added to the Watch Window and the present value will be monitored.
   Example when *D100* Is Designated



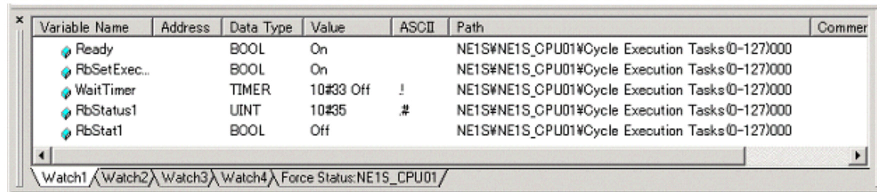
Variable Name	Address	Data Type	Value	ASCII	Path	Commer
	D01000	WORD	16#0000	..	NE1S¥NE1S_CPU01	

3. Repeat the previous step to monitor other addresses at the same time.

Changing the Display Format

To change the display format, select the items in the Watch Window (multiple items can be selected), right-click, and select ***Display Format*** and then ***Hex***, ***Dec***, ***Bin***, or ***Data Type*** from on the popup menu.

Example when *Data Type* Is Selected



13-9 Saving and Restoring Variable PVs

13-9-1 Function

This function reads all of the variable PVs saved in the CPU Unit and saves the data in a CSV file; it also restores the saved variable to the CPU Unit.

All of the variables are saved and restored together, including global, local, retained, and non-retained variables.

Note Observe the following precautions.

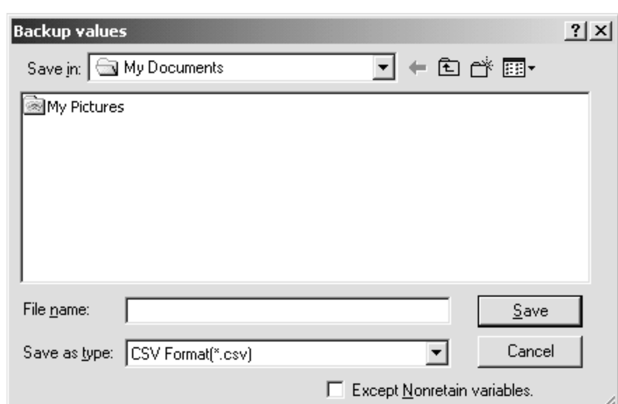
- The force-set/force-reset status of automatically allocated variables will not be preserved.
- The force-set/force-reset status of automatically allocated variables is cleared when the CPU Unit is switched from PROGRAM mode to MONITOR or RUN mode.

13-9-2 Procedure

Saving PVs

1,2,3...

1. Select ***Controller - Backup value of variables***. The Backup values Dialog Box will be displayed.

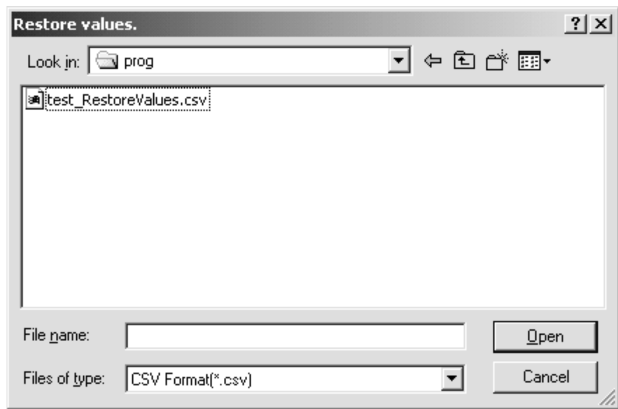


2. Specify the file location and file name and click the **Save** Button. The following dialog box will be displayed if the save operation was completed normally.

Restoring PVs

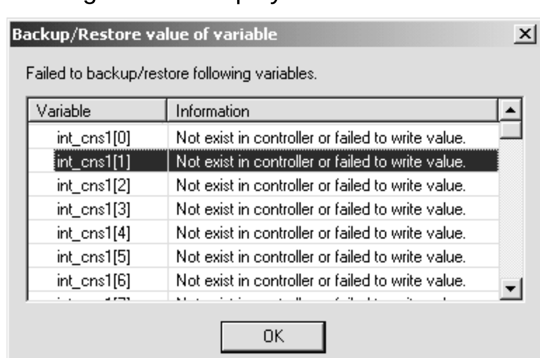
1,2,3...

1. Select ***Controller - Restore value of variables***. The Restore values Dialog Box will be displayed.



2. Specify the source file and click the **Open** Button. The following dialog box will be displayed if the restore operation was completed normally.

If none of the saved variables exist in the CPU Unit, the following error message will be displayed.



13-10 Forcing Bits ON and OFF (Force-set and Force-reset)

13-10-1 Overview

Force ON/Force OFF

When the CPU Unit is online in MONITOR or PROGRAM mode, input bits (contacts), output bits (coils), and timer/counter Completion Flags can be forced ON or OFF from the Ladder Editor or Watch Window.

The forced status is maintained until it is cleared or until it is forced ON or OFF again. Forced status will not change regardless of external input status or the results of program execution.

Force ON/Force OFF

When the CPU Unit is online in MONITOR or PROGRAM mode, input bits (contacts), output bits (coils), and timer/counter Completion Flags can be turned ON or OFF from the Ladder Editor or Watch Window.

The set status, however, is not forced and will change if the external input status changes or if the status changes as a result of ladder program execution. This is the difference between turning bits ON or OFF normally and forcing bits ON or OFF.

Note

(1) Confirm that the controlled system will not be adversely affected before changing the status of any bit in memory, including turning bits ON and OFF, forcing bits ON and OFF, and resetting forced status.

(2) The status of bits forced ON and OFF while online will be maintained even after the NE Programmer is taken offline.

(3) Before going offline, check the forced status on the Force Status Tab Page and clear the forced status.

(4) Do not unintentionally go offline while there is forced status remaining in the CPU Unit.

(5) The program in the NE Programmer and the program in the CPU Unit must be the same to enable forcing bits ON or OFF. If they are not the same, upload the program.

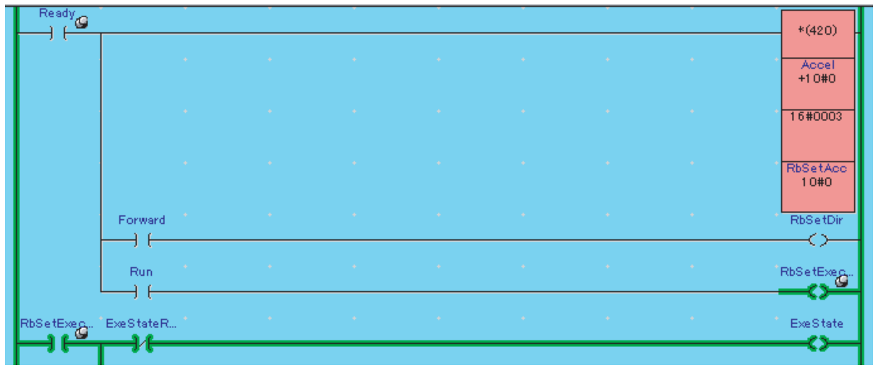
An option in the Integrated Options Window's General Tab Page must be selected in order to allow write operations (PV change, force-set/reset, and set/reset operations) on array variables specified with an index, such as A[i]. To enable these write operations, select ***Tool - Option*** to display the Integrated Options Window, click the **General** Icon, and select the *Permit a write operation to the Array element with index variable, e.g., A[i].* Option.

Note An array variable's index value is based on the index value the last time that the value was monitored, so a write operation may operate on a different array element if the variable's index value was changed since the last write operation.

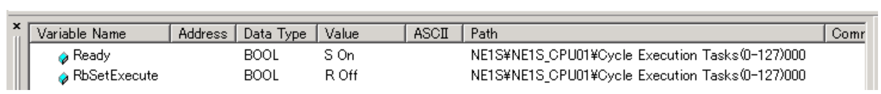
13-10-2 Turning Bits ON/OFF, Forcing Bits ON/OFF, and Clearing Forced Status

To change the status of a bit, select it in the Ladder Editor or Watch Window and select one of the following from the ***Set*** Menu: ***On, Off, Force On, Force Off, Release Force Status***.

Example for Force ON/Force OFF



- The icon shown below will be displayed on the Ladder Editor to indicate forced status if a bit is forced ON or forced OFF.



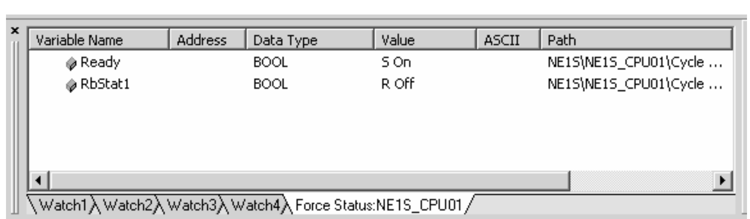
Variable Name	Address	Data Type	Value	ASCII	Path	Comm
Ready		BOOL	S On		NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	
RbSetExecute		BOOL	R Off		NE1S¥NE1S_CPU01¥Cycle Execution Tasks(0-127)000	

- In the Watch Window, *S On* will be displayed for bits that are forced ON and *R Off* will be displayed for bits that are forced ON.

Note When bits are turned ON or OFF, the icon, *S On*, and *R Off* will not be displayed in the Ladder Editor or Watch Window and only the normal display for ON and OFF status will be displayed.

13-10-3 Forced Status Display

Click the **Force Status** Tab in the Watch Window to display all the forced statuses for connected CPU Units. The following screen shows an example of the forced status display.



Variable Name	Address	Data Type	Value	ASCII	Path
Ready		BOOL	S On		NE1S\NE1S_CPU01\Cycle ...
RbStat1		BOOL	R Off		NE1S\NE1S_CPU01\Cycle ...

The forced status can be read from the CPU Unit using the following two methods:

- Click the **Force Status** Tab the first time the NE Programmer is started.
- Select ***Scan Force Status*** after right-clicking the Force Status Tab.

When the forced status for the connected CPU Unit is changed by the NE Programmer, the change is immediately shown on the Force Status Tab Page. When the forced status is changed from another NE Programmer installation, the change will not be shown until the forced status is refreshed with the most recent data.

13-11 Changing the PVs of Variables

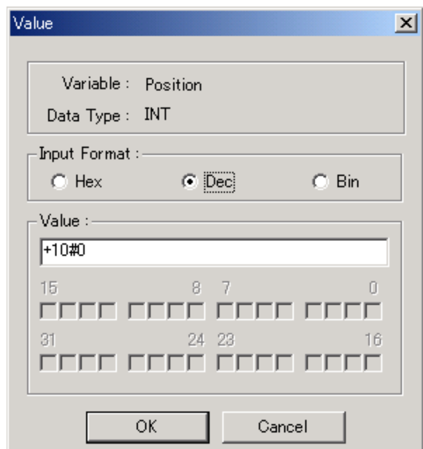
When the CPU Unit is online in MONITOR or PROGRAM mode, the present values of words can be changed from the Watch Window or the Ladder Editor.

Any present values that are changed, however, are not forced and will change if the external input status changes or if the status changes as a result of ladder program execution.

Note Confirm that the controlled system will not be adversely affected before changing a present value.

1,2,3...

1. Right-click the PV to be changed in the Watch Window and select ***Set - Value*** from the popup menu.
   The Value Dialog Box will be displayed.



2. Input the desired value and then click the **OK** Button.
   The specified value will be displayed in the Watch Window and Ladder Editor.

An option in the Integrated Options Window's General Tab Page must be selected in order to allow write operations (PV change, force-set/reset, and set/reset operations) on array variables specified with an index, such as A[i]. To enable these write operations, select ***Tool - Option*** to display the Integrated Options Window, click the **General** Icon, and select the *Permit a write operation to the Array element with index variable, e.g., A[i].* Option.

Note An array variable's index value is based on the index value the last time that the value was monitored, so a write operation may operate on a different array element if the variable's index value was changed since the last write operation.

13-12 Changing Timer/Counter Set Values

When the CPU Unit is online in MONITOR or PROGRAM mode, timer and counter set values can be changed with an online editing operation. Timer and counter set values can also be changed from constants to external settings such as a CIO word, Work word, or DM word.

Note Before changing a timer or counter set value, verify that the change will not adversely effect the system or equipment.

Supported Instructions

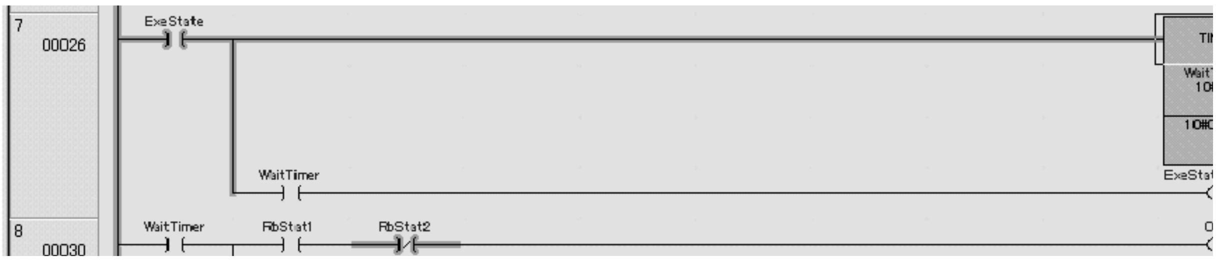
Set values can be changed in the following instructions.

Timer/Counter Operation Mode: Binary Mode

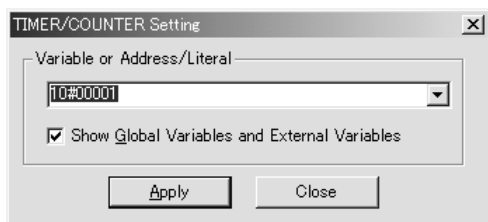
Timer instructions: TIMX, TIMHX(551), TIMLX(553), TMHHX(552), and TTIMX(555)
Counter instructions: CNTX and CNTRX(548)

The following example procedure shows how to change a timer's set value.

1,2,3... 1. Monitor the rung containing the timer that will be edited.



2. Select the TIMX instruction's set value, right-click, and select ***Set - Timer/ Counter Setting Value*** from the popup menu. The following dialog box will be displayed.

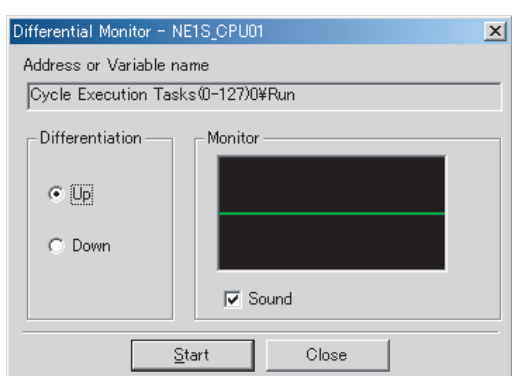


3. Change the variable or address/value and click the **Apply** Button. The set value will be changed. (The timer set value can be changed repeatedly until the **Close** Button is clicked.)

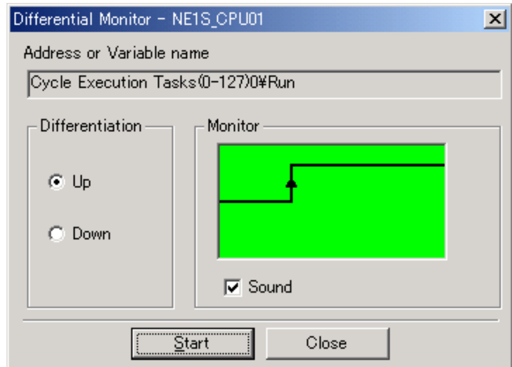
13-13 Differential Monitor

ON to OFF and OFF to ON transitions in bits can be detected online from the Ladder Editor and Watch Window.

1,2,3... 1. Right-click the variable for differential monitoring in the Watch Window or Ladder Editor and select ***Differential Monitor*** from the popup menu. The Differential Monitor Dialog Box will be displayed.



2. Select the *Differentiation Up/Down* and then click the **Start** Button. When the differentiation condition is met, the display will change as shown below for an Up condition.



13-14 Online Editing

13-14-1 Online Editing

When the CPU Unit is online in MONITOR or PROGRAM mode, multiple ladder rungs can be edited simultaneously from the Ladder Editor while monitoring status. With NE1S-CPU01 Rev. 3.1 or later revisions, multiple ladder rungs can be edited even if there are rung comments included with the rungs.

Broadly speaking, the following two types of online editing operations are supported.

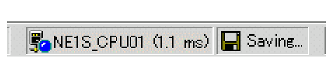
- Editing or deleting the selected rung
- Inserting rungs before or after the selected rung

The following editing functions are supported for online editing.

- Inserting, deleting, and changing basic instructions
- Inserting, deleting, and changing special instructions
- Adding external or internal variables
- Changing differentiated instructions (setting/releasing upward or downward differentiation or adding/deleting/changing differentiated instructions)
- Adding global variables

⚠ **Caution** Confirm that the controlled system will not be adversely affected even if the cycle time is increased before performing online editing. An increased cycle time may prevent input signals from being read.

Note (1) When online editing is performed, changes will be written first to the normal RAM in the CPU Unit and then they will be backed up in the flash memory in the CPU Unit. Do not turn OFF the power supply to the CPU Unit while the settings are being backed up to flash memory (i.e., while the BKUP indicator is lit). The following display will appear in the status bar while data is being written to flash memory.



(2) If the CPU Unit and NE Programmer do not contain the same program, and error message will be displayed and online editing will not be possible.

(3) To may major corrections to the program or to move rungs, edit the program offline and then download it to the CPU Unit.

(4) Do not edit the program in a way that would cause the maximum cycle time set in the PLC Setup to be exceeded.
If the maximum cycle time is exceeded, a cycle time exceeded error will occur and operation will stop.

If a cycle time exceeded error occurs, operation will stop.

- To start operation again, switch to PROGRAM mode and then return to RUN or MONITOR mode.

If a cycle time exceeded error persists, do the following.

- Change the program or increase the maximum cycle time setting.
- Perform the operation to clear error displays.

The CPU Unit will stop for a short period of time when any of the following instructions is inserted or deleted.
JMP, JME, or END

(5) Refer to *13-15* for information on clearing error displays.

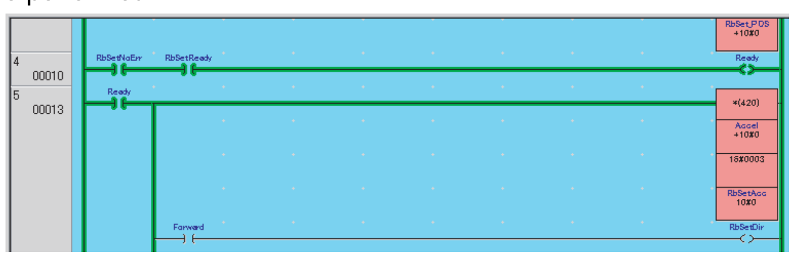
(6) There must be at least one END instruction in a program. Leave at least one END instruction in the program when performing online editing.

13-14-2 Online Editing Procedures

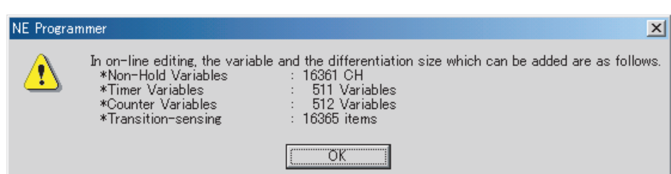
The following example shows how to input an OR using online editing.

1,2,3...

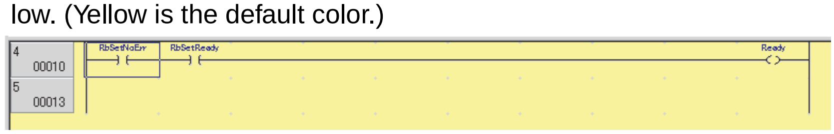
1. While monitoring the program, display the rung where online editing is to be performed.



2. Right-click the rung to be edited and select ***Online Edit - Begin*** from the popup menu. It does not matter what portion of the rung is selected. In this example, rung 4 will be edited.
The following dialog box will be displayed.

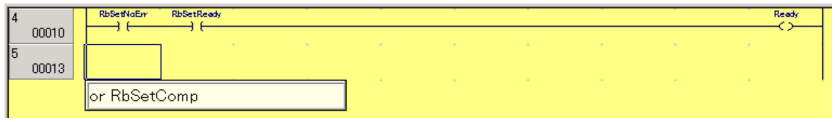


3. Click the **OK** Button.
The selected run will be displayed with a yellow background as shown below. (Yellow is the default color.)



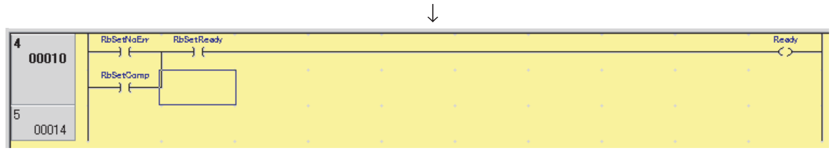
4. Edit the run. Editing methods are the same as for offline operations.
   In this example, an OR will be added for the NC contact at the beginning of the above rung.
   Move the cursor down one line, input ***or*** and the variable name, and then press the **Enter** Key.

   Input Example

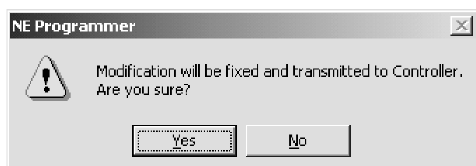


Note You can input only ***or***, double-click the ***or***, and then select or input the variable from the Edit Instruction Dialog Box that will be displayed.

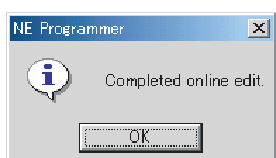
↓



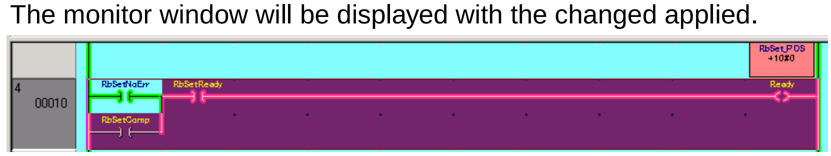
5. Right-click the Edit Window and select ***Online Edit - Finish*** from the pop-up menu.
   A confirmation dialog box will be displayed.



6. Click the **Yes** Button.
   The changes will be downloaded and the following dialog box will be displayed when the download has been finished.



7. Click the **OK** Button.
   The monitor window will be displayed with the changed applied.



This completes the online editing.

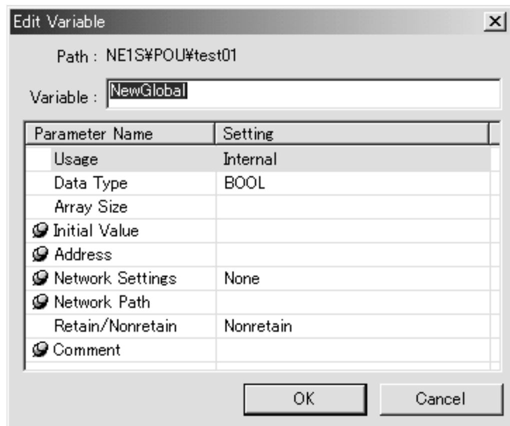
13-14-3 Adding Global Variables

Global variables can be added in online editing, although the following restrictions apply.

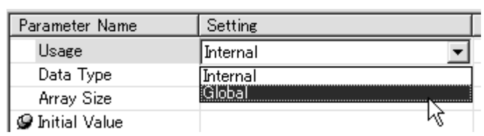
Procedure

1,2,3...

1. When a new variable is added during online editing, the following Edit Variable Dialog Box will be displayed.



2. Change the *Usage* setting to *Global*.



   Addresses and network variables can be set for global variables.

3. Click the **OK** Button. The new variable will be registered as a global variable. The new variable will also be registered as an external variable.

Precautions

The added global variable cannot be edited or deleted during online editing.

- The addition of a new variable to the global and external variables can be cancelled during online editing by ending global editing.
- When online editing has been completed (changes transferred), it is necessary to close the connection, edit/delete the global variable, build the program, and download the program.

13-15 Clearing Errors

Use the following procedure to clear error displays.

1,2,3...

1. If an error is displayed, first remove the cause of the error.
2. Select ***Controller - Clear Error***. Alternately, press the **F4** Key.
   Errors, however, cannot be cleared when another Programming Devices has the access right or the CPU Unit is in RUN mode.

The information will be displayed in the status bar when an error occurs.

Information will also be display on the Error Tab Page of the Error Log Window.

Refer to *13-18* for information on errors and the Error Log Window.

Controller - Clear Error is the same function as that executed for the **Clear** Button on the Error Tab Page in the Error Log Window.

13-16 Clearing Memory

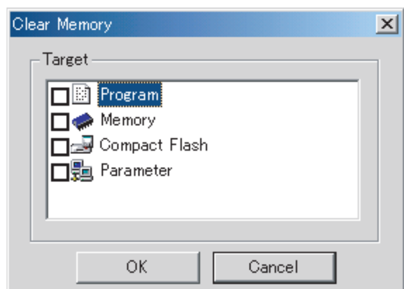
Memory in the CPU Unit can be cleared in the following units.

- All user programming (multiple programs)
- I/O Memory Area
- Parameter Area

Perform all of the following operations online. Before starting, connect the NE Programmer online to the CPU Unit.

1,2,3...

1. Select ***Controller - Clear Memory***.
   The following dialog box will be displayed.



2. Select the data to be cleared and then click the **OK** Button.
   The following dialog box will be displayed.

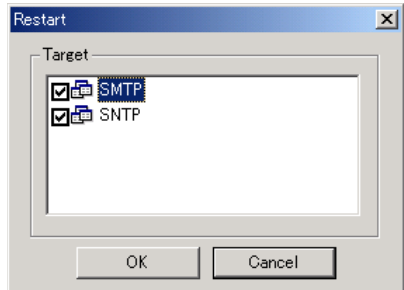
3. Click the **Yes** Button.
   The data specified in step 2 will be cleared.

13-17 Restarting Services

Use the following procedures to restart the SMTP and SNTP services.

Note This function can be used to make changes to the email settings (SMTP) and time settings (SNTP) in the Ethernet Setting valid, eliminating the need to restart the CPU Unit.
Refer to *4-2-7* and *4-2-8* for details on SMTP and SNTP.

1,2,3... 1. Select ***Controller - Restart Service***.
The Restart Dialog Box will be displayed.



2. Select the service or services to be restarted and then click the **OK** Button.
The specified service or services will be restarted.

13-18 Displaying Errors and the Error Log

This section describes the procedures to display current errors and delete error displays.

Errors and the error log are displayed in the Error Log Window. Messages generated by the MSG instruction are also displayed.

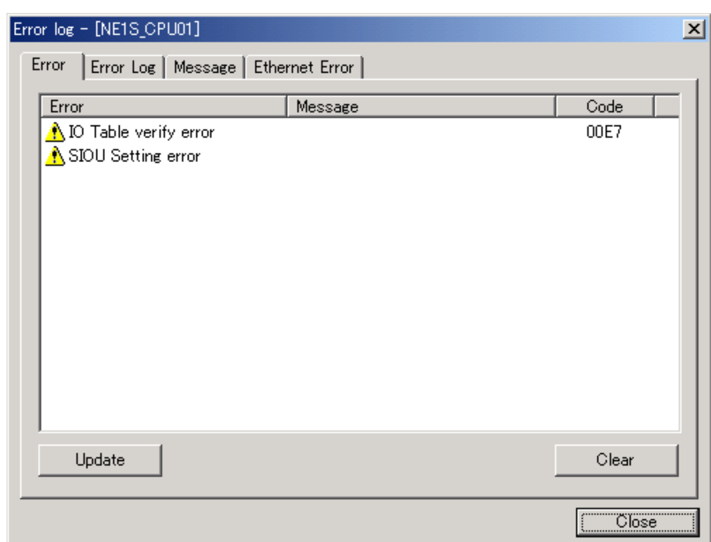
When an error that is stored in the error log occurs, an error message will flash in red in the status bar.

Perform all of the following operations online. Before starting, connect the NE Programmer online to the CPU Unit.

13-18-1 Displaying Current Errors

Use the following procedure to display current errors.

1,2,3... 1. Select ***Controller - Error Log***.
The Error Tab Page of the Error Log Window will be displayed as shown below.



- Current errors will also be display on the Error Tab Page of the Error Log Window.
- Errors that occur after this window has been displayed will automatically be added to the display.

- The error level of each error (fatal or nonfatal) will also be indicated.
- The errors that may be displayed as follows:

Fatal Errors (in Order of Priority)	Nonfatal Errors (in Order of Priority)
Memory error	System error (FAL)
I/O bus error	Interrupt task error
Unit/Rack number duplication error	PLC Setup error
Too many I/O error	I/O verification error
I/O setting error	CPU Bus Unit error
Program error	Special I/O Unit error
Cycle time exceeded error	Battery error
System error (FALS)	CPU Bus Unit setting error
	Special I/O Unit setting error
	Communications interface error

2. If an error is displayed, first remove the cause of the error.
3. Click the **Clear** Button. All error displays will be cleared. (This is the same function as that executed for ***Controller - Clear Error***.

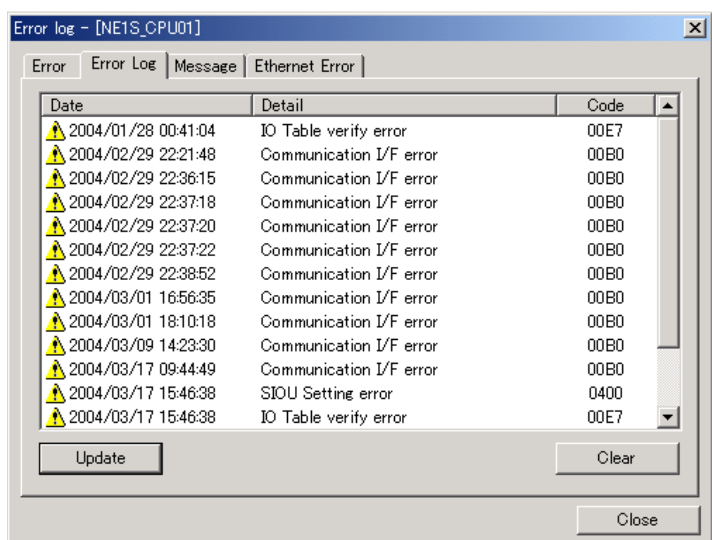
Errors, however, cannot be cleared when another Programming Devices has the access right or the CPU Unit is in RUN mode.

13-18-2 Displaying the Error log

Use the following procedure to display the error log.

Click the **Error Log** Tab in the Error Log Window.

An error log like the following one will be displayed.



- Error errors displayed on the Error Log Tab Page are the same as those display on the Error Tab Page. Refer to the previous section for details.
- The error log contains up to 20 records. If more than 21 error occur, the oldest records will be deleted.
- The error log will be cleared if the **Clear** Button is pressed.

Note A CPU waiting error (fatal) is not displayed on the Error or Error Log Tab Page. A CPU waiting error will have occurred when the POWER indicator on the

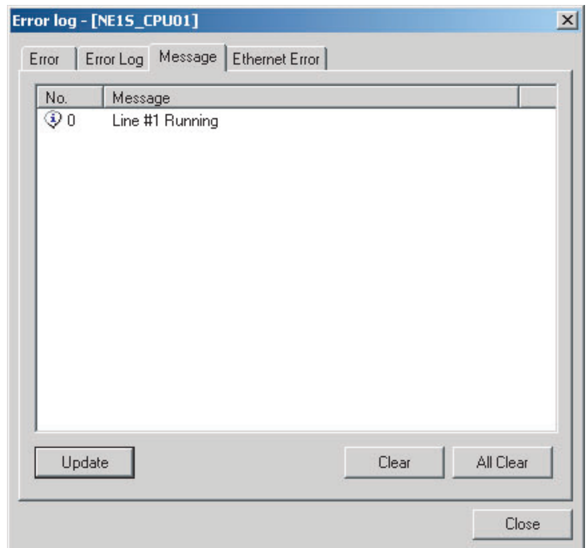
Power Supply Unit is lit, but the RUN and ERR/ALM indicators on the front of the CPU Unit are both not lit.

13-18-3 Displaying Messages

Use the following procedure to display messages generated by the MSG instruction.

Click the **Message** Tab in the Error Log Window.

A message list like the following one will be displayed.



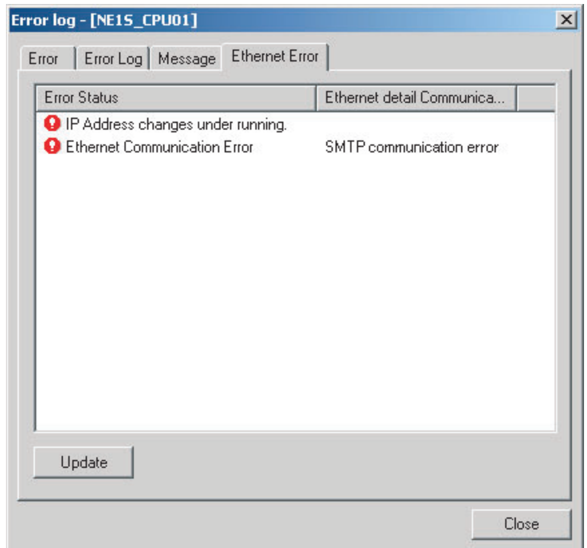
- The message numbers will be displayed along with the text of the message.
- Up to 8 messages will be displayed.
- The currently selected message will be cleared if the **Clear** Button is pressed.
- All messages will be cleared if the **All Clear** Button is pressed.
- Double-byte codes can be displayed.
- Messages generated by the FAL and FALS instructions will not be displayed.

13-18-4 Displaying Ethernet Errors

Use the following procedure to display Ethernet errors.

Click the **Ethernet Error Log** Tab in the Error Log Window.

Ethernet errors, like those shown below, will be displayed.



Error Status

The error status is indicated as follows:

IP Address Settings Error
Unit IP Address Settings Error
FTP Server Error
SMTP Settings Error
SNTP Settings Error
Unit IP Address Settings Over
IP Routing Settings Error
IP Address Duplicate Error
IP Address Changes Under Running
Ethernet Communication Error

Ethernet Communications Errors

The following error status is displayed for the Ethernet communications error.

SMTP Communications Error
SNTP Communications Error
FINS/TCP Connection Error
FINS/TCP Send Error
FINS/TCP Receive Error
FINS/UCP Send Error
FINS/UCP Receive Error

13-19 Change Log

13-19-1 Overview

The log data created when downloading, editing online, creating/clearing I/O tables, or clearing memory can be recorded in the CPU Unit.

Item	Details
Save timing	Downloading, executing online editing, creating/clearing I/O tables, clearing memory, and changing Timer/Counter SVs
Number of saved data records	100 records
Saved data	Time, operation (download, online edit, I/O table create/clear, memory clear), version, author, comments (120 bytes)
Location of saved data	Recorded data = Flash ROM in CPU Unit
Simple backup	Supports simple backup operations

- The settings can be made to send the data by email when the change log is recorded.
- Recorded change log entries can be deleted by the user.
- Recorded log data can also be overwritten during online editing.
- The change log can be enabled/disabled by the user (default: disabled).

13-19-2 Enabling/Disabling the Change Log

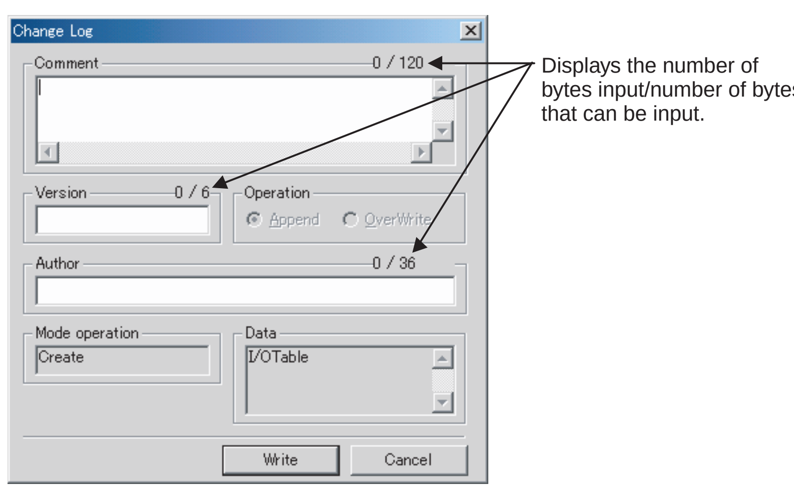
Set whether to enable or disable the change log. The default setting is disabled.

Select ***Controller - Change Log*** and then either ***Enable Mode*** or ***Disable Mode***. When *Enable Mode* is selected for change log, the Change Log Dialog Box will be displayed when downloading, editing online, creating/clearing I/O tables, or clearing memory.

13-19-3 Change Log Input Examples

Creating I/O Tables

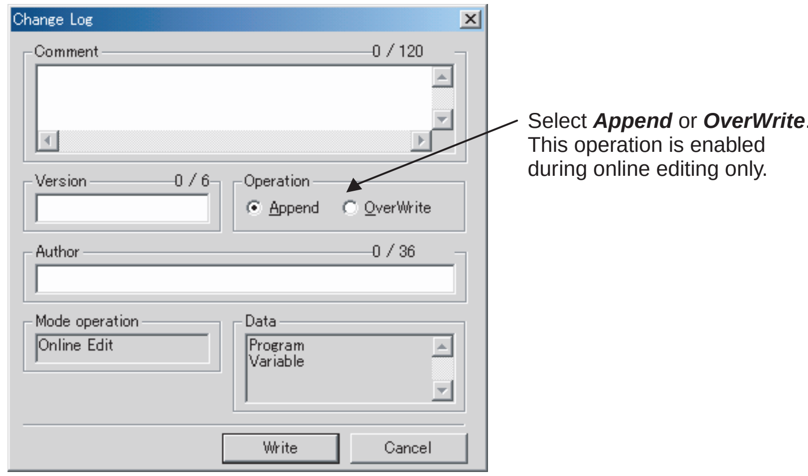
- When creating I/O tables, the following Change Log Dialog Box will be displayed.



- Input the comment and other required data, and then click the **Write** Button.

Editing Online

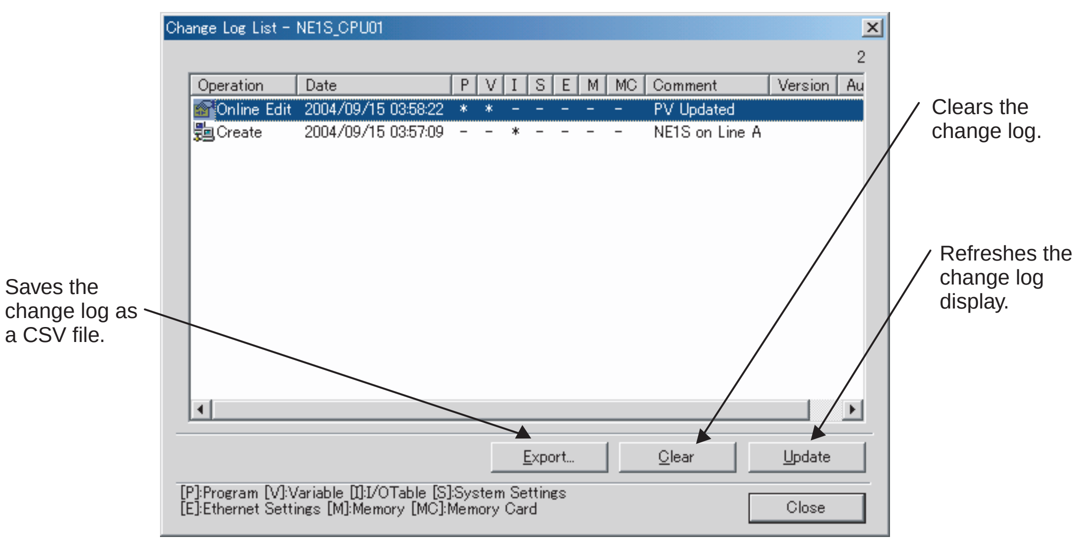
- When online editing is completed, the following Change Log Dialog Box will be displayed.



- Input the comment and other required data, select ***Append*** or ***OverWrite***, and then click the **Write** Button.
  To record several online editing operations as a single change log, select ***Overwrite***. To record each operation as a separate change log, select ***Append***.

13-19-4 Change Log Display

Select ***Controller - Change Log - Change Log List*** to display the Change Log List Window.

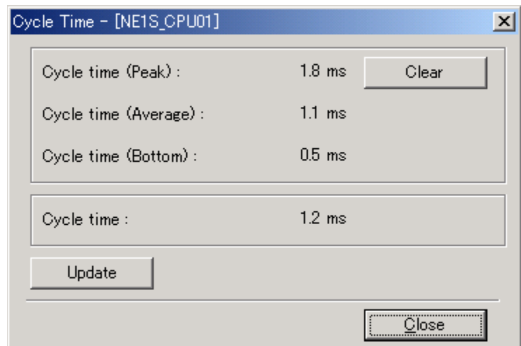


13-20 Displaying the Cycle Time

The following procedure can be used to measure and display the cycle time of the program that is being executed. The CPU Unit must be in RUN or MONITOR mode to use this function.

Select ***Controller - Cycle Time***.

The following Cycle Time Dialog Box will be displayed.



- The peak, average, bottom, and current cycle times will be cleared if the **Clear** Button is pressed.
- The peak, average, bottom, and current cycle times will be remeasured and displayed again if the **Update** Button is pressed.
- The Cycle Time Dialog Box will be closed if the **Close** Button is pressed.

Note Even if the above procedure is not performed, the average cycle time will be displayed in the status bar when the CPU Unit is in RUN or MONITOR mode.

13-21 Data Tracing

13-21-1 Data Tracing

Data tracing displays the operating status of the CPU Unit on the NE Programmer. Operating status includes the status of bits and the present values (PVs) of words.

When the specified trigger condition is met after starting data tracing, the CPU Unit will store the status of specified bits and the PVs of specified words in trace memory in the CPU Unit according to sampling conditions. The NE Programmer can read the data stored in the CPU Unit and display it on time charts.

Note The Data Trace Start Flag will turn ON when the data trace operation is executed.

- Any of three trigger conditions can be selected.

  Bit OFF to ON transition, bit ON to OFF transition, or specified word contents
- Either of two sampling conditions can be selected.

  Each cycle or when the TRSM instruction is executed.
- Data tracing will stop when the trace memory become full and then the trace data will be transferred to the NE Programmer and displayed.

- Data tracing is performed by the CPU Unit itself, which enables high-speed sampling in comparison to time chart monitoring.

■ **Number of traceable bits and words:**

Bits: 31, Words: 6

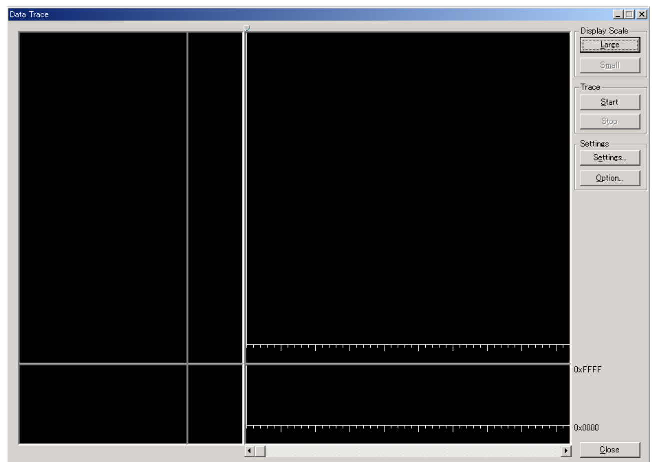
13-21-2 Opening and Closing the Data Trace Window

Monitoring data traces is performed in the Data Trace Window.

Opening

Select ***Controller - Data Trace***.

The following Data Trace Window will be displayed.



Closing

Click the **Close** Button in the Data Trace Window.

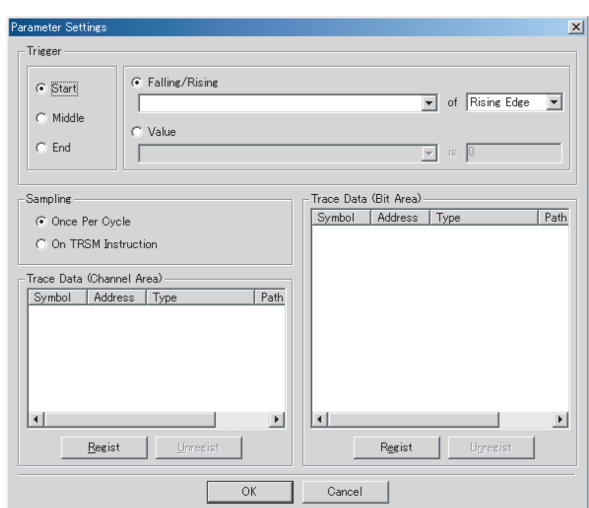
The Data Trace Window will be closed.

13-21-3 Setting Data Trace Parameters

This section described how to set the data trace parameters.

Displaying the Parameter Settings Dialog Box

1,2,3... 1. Click the **Settings** Button in the Data Trace Window. The following Parameter Settings Dialog Box will be displayed.



2. Make all of the required settings and then click the **OK** Button. The settings are described individually below.

Trigger

The *Trigger* fields are used to set the type of trigger and the condition for the bit or word trigger.

Select *Start*, *Middle*, or *End* as the type of trigger.

Start: Used to display status after the trigger.
Middle: Used to display status before and after the trigger.
End: Used to display status before the trigger.

■ Using a Bit Trigger

Select ***Falling/Rising***, specify the variable or address, and select ***Falling*** or ***Rising***.

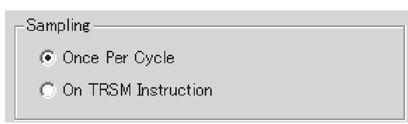
■ Using a Word Trigger

Select ***Value***, specify the variable or address, and input a hexadecimal value.

If *TIM/CNT* is input, the timer/counter Completion Flag will be used.

Sampling

Set the sampling condition in the *Sampling* Area.

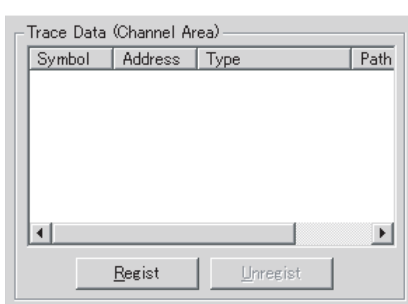


Select one of the two sampling methods.

Sampling method	Function	Contents of sampled data
Once Per Cycle	Sampling will be performed once per cycle.	I/O data after the END instruction is executed
On TRSM Instruction	Sampling will be performed when the TRSM instruction is executed.	I/O data when the END instruction is executed

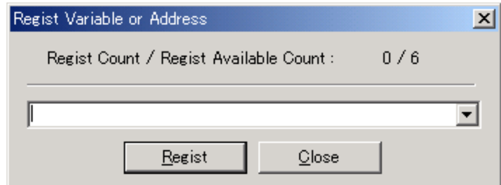
Trace Data (Channel Area)

Use the following procedure to specify the word variables or addresses to be sampled in the *Trace Data (Channel Area)*.



1,2,3...

1. Click the **Regist** Button.
   The following Regist Variables or Address Dialog Box will be displayed.

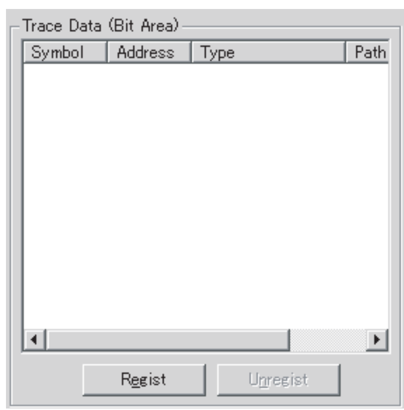


2. Input the word variable or address to be sampled and then click the **Regist** Button.
3. Repeat step 2 to monitor more than one word. Up to 6 word variables and addresses can be registered.
4. Set all the required variables or addresses and then click the **Close** Button.

Note To delete a variable or address that has been set, select the variable or address and click the **Unregist** Button.

Trace Data (Bit Area)

Use the following procedure to specify the bit variables or addresses to be sampled in the *Trace Data (Bit Area)*.



1,2,3...

1. Click the **Regist** Button.
   The following Regist Variables or Address Dialog Box will be displayed.

2. Input the bit variable or address to be sampled and then click the **Regist** Button.
3. Repeat step 2 to monitor more than one bit. Up to 31 bit variables and addresses can be registered.
4. Set all the required variables or addresses and then click the **Close** Button.

Note To delete a variable or address that has been set, select the variable or address and click the **Unregist** Button.

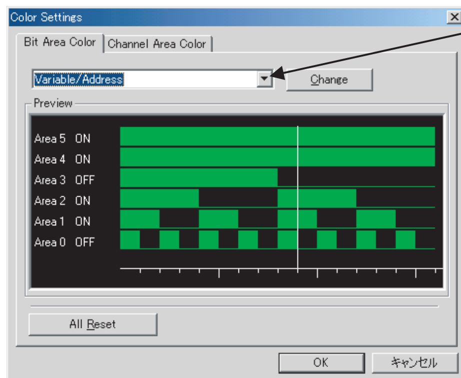
13-21-4 Setting Data Trace Display Colors

This section described how to set the colors displayed on the Data Trace Window.

Bit Area Colors

1,2,3...

1. Click the **Option** Button in the Data Trace Window.
   The following Bit Area Color Tab Page of the Color Settings Dialog Box will be displayed.



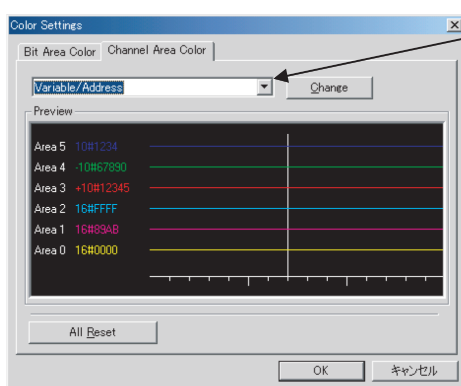
Select the item to change:
Variable/Address
Bit Status
Background
Grid Lines
Graph

2. Select the item for which to change the color and then click the **Change** Button.
3. The standard Windows color setting dialog box will be displayed. Set the desired color.

Word Area Colors

1,2,3...

1. Click the **Channel Area Color** Tab in the Color Settings Dialog Box.



Select the item to change:
Variable/Address
Value 1 to 6
Background
Grid Line

2. Select the item for which to change the color and then click the **Change** Button.
3. The standard Windows color setting dialog box will be displayed. Set the desired color.

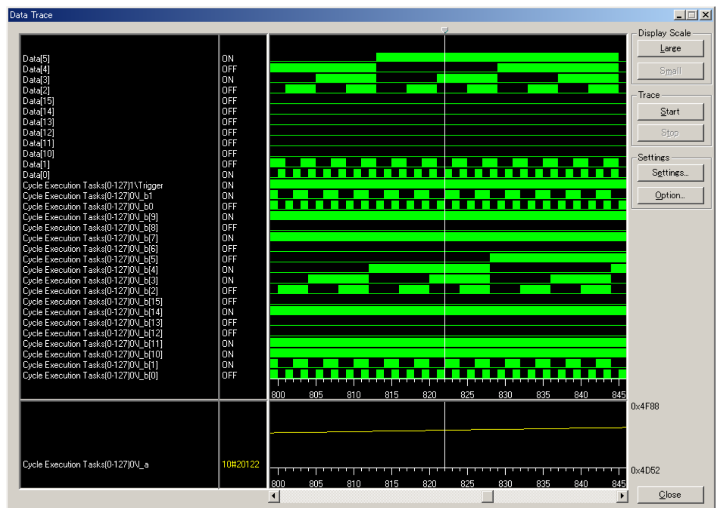
13-21-5 Executing the Data Trace Monitor Function

Use the following procedure to execute the data trace monitor function.

Be sure to set the data trace parameters before executing this procedure.

Click the **Start** Button in the Data Trace Window.

- The data trace will be started. Sampling will be started when the trigger condition is met, and traced data will be stored in the trace memory in the CPU Unit.
- When the trace memory becomes full, sampling will stop automatically. Traced data will be read from the CPU Unit and the trace data will be displayed in the Data Trace Window as shown below.



- To stop the data trace before trace memory becomes full, click the **Stop** Button. The Data Trace Window will be displayed when the Stop Button is clicked.

Changing the Display Scale

The trace data display scale can be enlarged or reduced by clicking the **Large** or **Small** Button while the data trace data is being displayed.

13-22 Variable Reference List

The Variable Reference List Window displays the Variable Usage Report and Cross Reference Report. This window can be accessed both offline and online.

Variable Usage Report

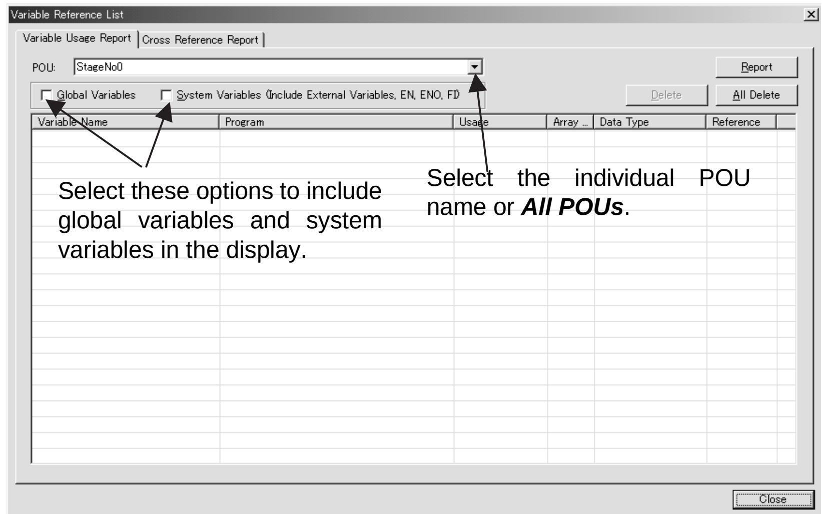
The variables registered in the project are listed in this report together with whether the registered variables are being used in the program. The variables not being used can be deleted.

Cross Reference Report

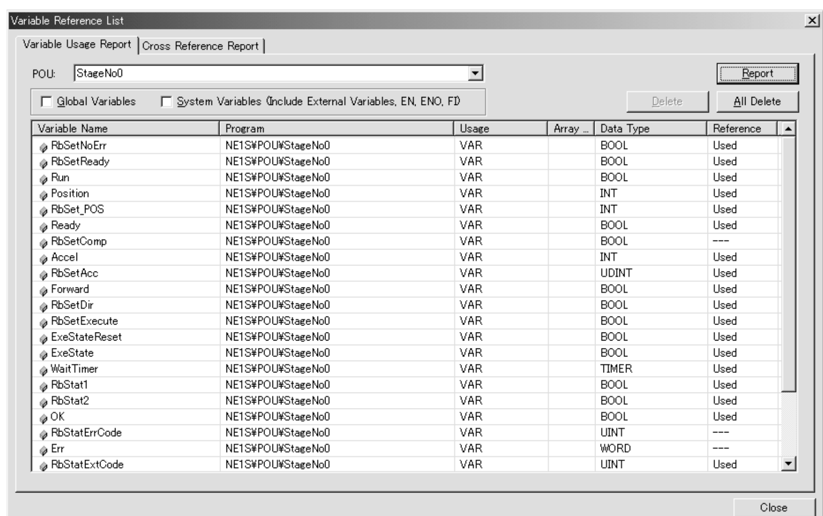
This report lists in which instructions, location, and program the variables are being used. The usage status of variables can be listed for individual POUs or for all POUs.

13-22-1 Variable Usage Report

1,2,3... 1. Select ***Controller - Variable Reference***. The Variable Usage Report Tab will be displayed in the Variable Reference List Window, as shown here.



2. Select a POU name from the *POU* field, and then click the **Report** Button. The following Variable Usage Report will be displayed.

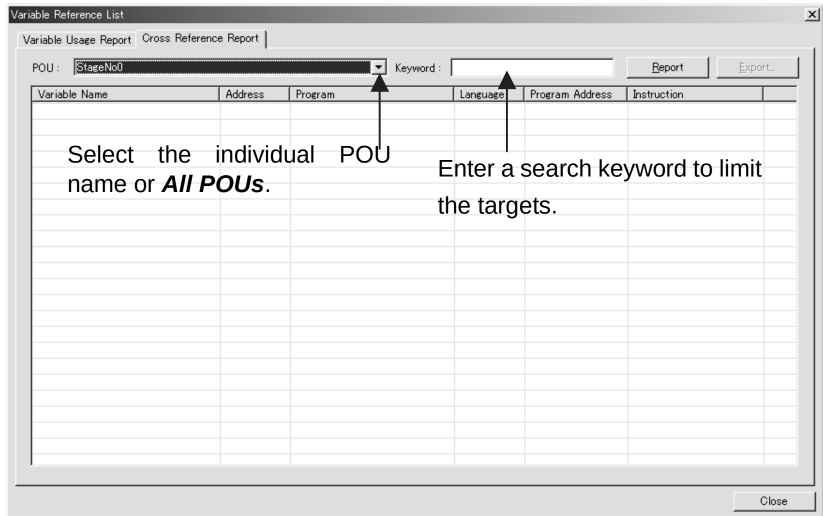


- Click the **All Delete** Button to delete all unused variables.
- To delete unused variables individually, select the variable to be deleted (multiple variables can also be selected), and click the **Delete** Button.

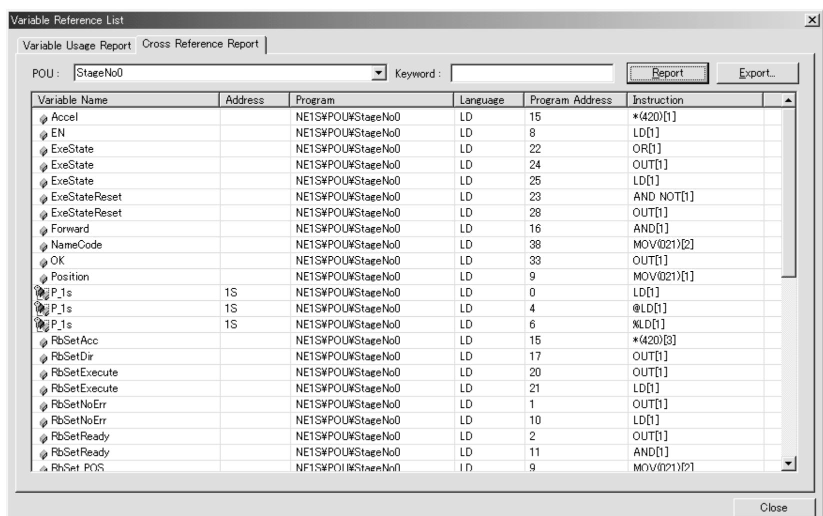
13-22-2 Cross Reference Report

1,2,3...

1. Click the Cross Reference Report Tab in the Variable Reference List Window to display the following window.



2. Select a POU name from the *POU* field, and then click the **Report** Button. The following Cross Reference Report will be displayed.



Variable Name	Address	Program	Language	Program Address	Instruction
Accel		NE1S¥POU¥StageNo0	LD	15	*(420)[1]
EN		NE1S¥POU¥StageNo0	LD	8	LD[1]
ExeState		NE1S¥POU¥StageNo0	LD	22	OR[1]
ExeState		NE1S¥POU¥StageNo0	LD	24	OUT[1]
ExeState		NE1S¥POU¥StageNo0	LD	25	LD[1]
ExeStateReset		NE1S¥POU¥StageNo0	LD	23	AND NOT[1]
ExeStateReset		NE1S¥POU¥StageNo0	LD	28	OUT[1]
Forward		NE1S¥POU¥StageNo0	LD	16	AND[1]
NameCode		NE1S¥POU¥StageNo0	LD	38	MOV(021)[2]
OK		NE1S¥POU¥StageNo0	LD	33	OUT[1]
Position		NE1S¥POU¥StageNo0	LD	9	MOV(021)[1]
P_1s	1S	NE1S¥POU¥StageNo0	LD	0	LD[1]
P_1s	1S	NE1S¥POU¥StageNo0	LD	4	@LD[1]
P_1s	1S	NE1S¥POU¥StageNo0	LD	6	%LD[1]
RbSetAcc		NE1S¥POU¥StageNo0	LD	15	*(420)[3]
RbSetDir		NE1S¥POU¥StageNo0	LD	17	OUT[1]
RbSetExecute		NE1S¥POU¥StageNo0	LD	20	OUT[1]
RbSetExecute		NE1S¥POU¥StageNo0	LD	21	LD[1]
RbSetNoErr		NE1S¥POU¥StageNo0	LD	1	OUT[1]
RbSetNoErr		NE1S¥POU¥StageNo0	LD	10	LD[1]
RbSetReady		NE1S¥POU¥StageNo0	LD	2	OUT[1]
RbSetReady		NE1S¥POU¥StageNo0	LD	11	AND[1]
RbSet_POS		NE1S¥POU¥StageNo0	LD	9	MOV(021)[2]

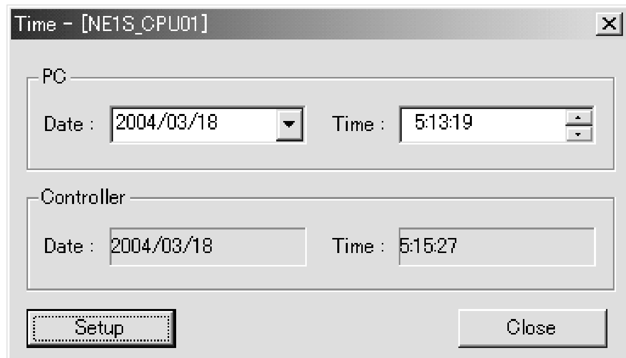
- Click the **Export** Button to save the contents of the report as a CSV file.

13-23 Setting the CPU Unit Clock

Use the following procedure to synchronize the CPU Unit's clock with the personal computer's clock.

1,2,3...

1. Select ***Controller - Set Clock***. The following Time Dialog Box will be displayed.



2. Click the **Setup** Button. The CPU Unit's clock will be synchronized with the personal computer's clock.

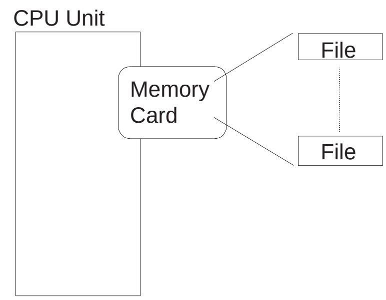
   The CPU Unit's clock can also be set by setting the *Date* and *Time* fields in this dialog box, and then clicking the Setup Button.

SECTION 14
File Memory Functions

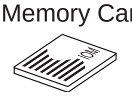
This section describes the functions used to manipulate file memory.

14-1 File Memory

The NE1S Series support file memory, which is used to store files. The Memory Card can be used as memory for storing files. File memory can be used to store the entire user program, I/O memory, parameter areas, and variable data as files.

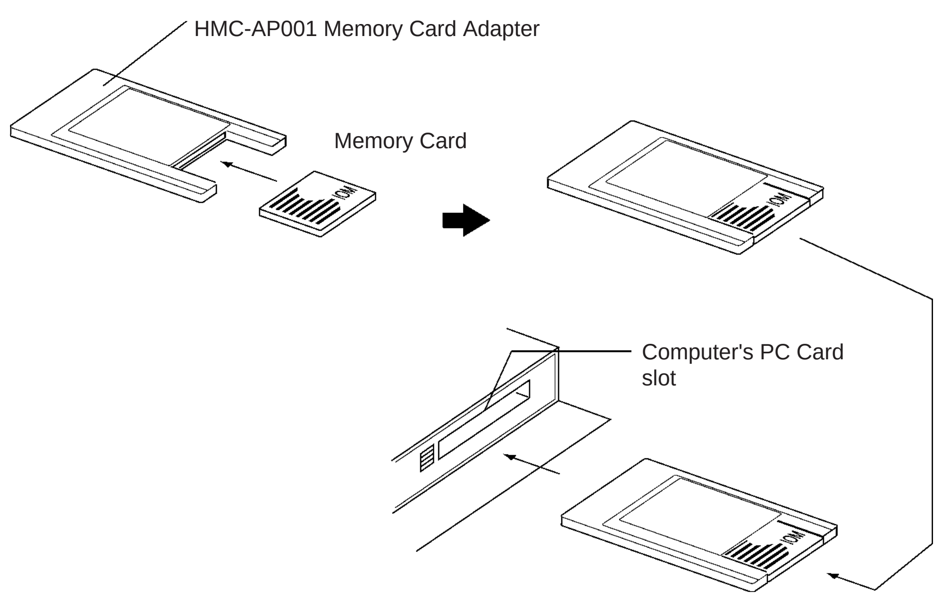


14-1-1 Types of File Memory

Category	Type	Capacity	Model	File data recognized by the CPU Unit
Memory Cards 	Flash memory	15 Mbytes	HMC-EF172	1) Entire user program
		30 Mbytes	HMC-EF372	2) Specified range in I/O memory
		64 Mbytes	HMC-EF672	3) Parameter area data (PLC Setup and other settings)
				4) Variable data
				5) ST (structured text) language source code, comment, program change logs (These are system files, however.)

Note

1. The Memory Cards are already formatted and can be used in the CPU Unit without any further formatting. Do not format the Memory Cards on a personal computer or any other device; doing so will make it impossible to use them on the CPU Unit.
2. Refer to *14-3 Formatting the Memory Card* for details on formatting the Memory Card from the CPU Unit using NE Programmer.
3. The HMC-AP001 Memory Card Adapter can be used to mount a Memory Card in the PLC card slot of a personal computer to use the Memory Card as a storage device. This can be done to manipulate files using other programs, such as Windows Explorer.



Memory Card Precautions

Confirm the following items before using a Memory Card.

Format

Memory Cards are formatted before shipping. There is no need to format them after purchase. To format them once they have been used, always do so in the CPU Unit using the NE-Programmer. (Refer to *14-3 Formatting the Memory Card*.)

If a Memory Card is formatted directly in a notebook computer or other computer, the CPU Unit may not recognize the Memory Card. If this occurs, you will not be able to use the Memory Card even if it is reformatted in the CPU Unit.

Number of Files in Root Directory

There is a limit to the number of files that can be placed in the root directory of a Memory Card (just as there is a limit for a hard disk). Although the limit depends on the type and format of the Memory Card, it will be between 128 and 512 files. When using applications that write log files or other files at a specific interval, write the files to a subdirectory rather than to the root directory.

Subdirectories can be created in advance on a computer.

Number of Writes

Generally speaking, there is no limit to the number of write operations that can be performed for a flash memory. For the Memory Cards, however, a limit of 100,000 write operations has been set for warranty purposes. For example, if the Memory Card is written to every 10 minutes, over 100,000 write operations will be performed within 2 years.

Minimum File Size

If many small files, such as ones containing only a few words of DM Area data, are stored on the Memory Card, it will not be possible to use the complete capacity of the Memory Card. For example, if a Memory Card with an allocation unit size of 4,096 bytes is used, at least 4,096 bytes of memory will be used for each file regardless of how small the file is. If you save 10 words of DM Area data to the Memory Card, 4,096 bytes of memory will be used even though the actual file size is only 68 bytes. Using files of such a small size greatly reduces the utility rate of the Memory Card. If the allocation unit size is increase, the access speed will be increased but the utility rate will be reduced.

The allocation unit size of the Memory Card can be checked from a DOS prompt using CHKDSK. The specific procedure is omitted here. Refer to general computer references for more information on allocation unit sizes.

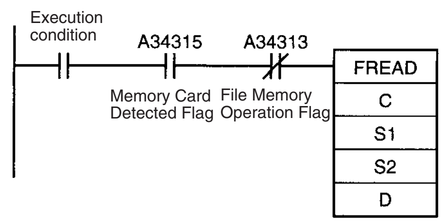
Memory Card Access Precautions

When the PLC is accessing the Memory Card, the BUSY indicator will light on the CPU Unit. Observe the following precautions.

1,2,3...

1. Never turn OFF the power supply to the CPU Unit when the BUSY indicator is lit. The Memory Card may become unusable if this is done.
2. Never remove the Memory Card from the CPU Unit when the BUSY indicator is lit. Press the Memory Card power OFF button and wait for the BUSY indicator to go out before removing the Memory Card. The Memory Card may become unusable if this is not done.

3. Insert the Memory Card with the label facing to the right. Do not attempt to insert it in any other orientation. The Memory Card or CPU Unit may be damaged.
4. A few seconds will be required for the CPU Unit to recognize the Memory Card after it is inserted. When accessing a Memory Card immediately after turning ON the power supply or inserting the Memory Card, program an NC condition for the Memory Card Recognized Flag (A34315) as an input condition, as shown below.

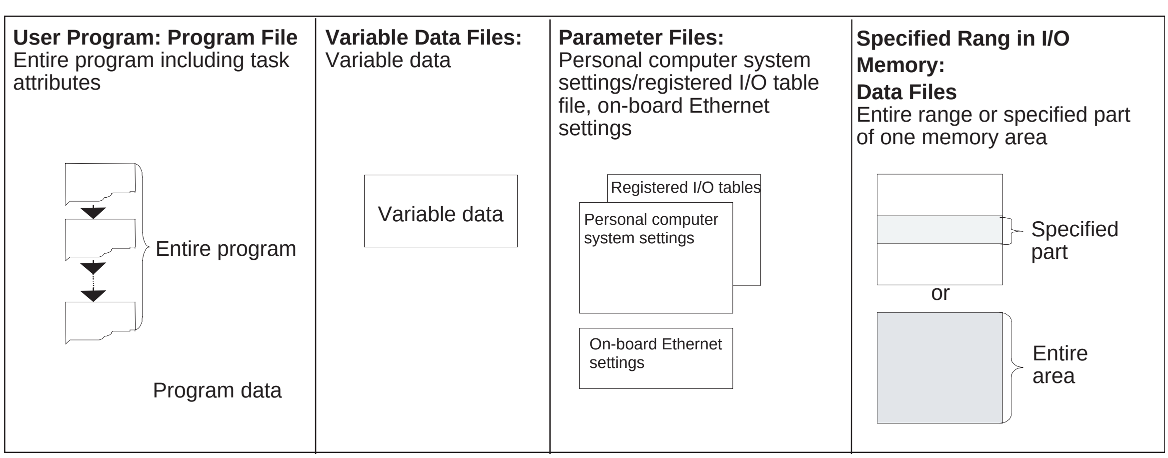


Note Power Interruptions while Accessing a Memory Card
Do not turn OFF the power supply to the PLC while a Memory Card is being accessed.
If a power interruption occurs (i.e., if the power supply goes OFF) while the Memory Card is being accessed, the Memory Card data cannot be guaranteed, the file being rewritten may not be overwritten correctly, and, in some cases, the Memory Card itself may be damaged.
When this occurs, the File Deletion Notification Flag (A39507) will be automatically set to 1 (ON). (The File Deletion Notification Flag will be set to 0 (OFF) the next time the power supply is turned OFF.

14-1-2 File Data

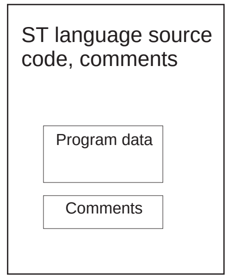
The following files can be written. These files can be transferred by FTP over an Ethernet network, or accessed (read or written) by instructions.

- Program Files (all user programs and program information)
- Variable Data Files
- Parameter Files (PLC Setup, registered I/O tables, built-in Ethernet Setup)
- Data Files



Depending on the system, the following files are automatically saved on the Memory Card: ST (structured text) language source code, comments (variable, instruction, rung, or structural member comments), change log (See note).

Note This function will be supported soon.



14-1-3 Files

Files are formatted in DOS, and therefore can be used as regular files on a Windows computer.

Files are identified by file names and extensions, as shown in the following table. A file name is written using the following characters: Letters A to Z, numbers 0 to 9, !, &, $, #, `, {, }, –, ^, (,), and _

The following characters cannot be used in file names: ,, ., /, ¥, ?, *, “, :, :, <, >, =, +, space, and 2-byte characters.

The filename extensions depend upon the type of file being stored. The location of a file in memory can be specified in the directory, and a directory can be up to 5 subdirectories deep (counting the root directory).

File Types, Names, and Extensions

There are 4 types of files that can be managed (read and written) by the CPU Unit.

- **General-purpose Files**
  These files can be transferred by FTP over an Ethernet network using CIP commands, or accessed (read or written) by instructions. The file names can be defined freely by the user.
- **Backup Files**
  These files are transferred between the Memory Card and CPU Unit by the backup function. The filenames are fixed as BACKUP□□.
- **System Files**
  These are files that the system automatically saves onto the Memory Card.

General-purpose Files The following table shows file names and extensions of general-purpose files.

Type	Name[1]	Extension	Description	Explanation	
Data file	********	.IOM	Specified range in I/O memory	• Data from start to end word in word units (16 bits) located in one area. • The area can be the CIO, HR, WR, AR, DM, or EM Area.	Binary format
		.TXT			TXT format (non-delimited or tab-delimited)
		.CSV			CSV format (comma-delimited)

Note File names, represented by “********” above, consist of up to 8 ASCII characters.

Backup Files

The following files are all created on the Memory Card during write processing for the simple backup operation.

Type	Name	Extension	Description	Explanation
Program file	BACKUP	.OBJ	Entire user program	• Contains all cyclic and interrupt task programs as well as task data for one CPU Unit. • This file must exist on the Memory Card when reading data from the Memory Card during backup.
		.PFL	Program data	• Contains the program source file for one CPU Unit. • This file must exist on the Memory Card when reading data from the Memory Card during backup.
Variable data file		.USV .TNS .VTP	Variable data	• Variable data for one CPU Unit. Four files maximum. (See note.) • This settings data is automatically stored in a predetermined location of the CPU Unit when the power supply is turned ON. • This file must exist on the Memory Card when reading data from the Memory Card during backup. Note: The number of files varies depending on variable usage.
Parameter file		.STD	PLC Setup, registered I/O table	• Contains all initial settings for one CPU Unit. • The user does not have to distinguish parameter data in the file by type. • This file must exist on the Memory Card when reading data from the Memory Card during backup.
		.ETP	Built-in Ethernet Setup	• Contains the settings for the Ethernet function built into the CPU Unit. • This settings data is automatically stored in a predetermined location of the CPU Unit when the power supply is turned ON. • This file must exist on the Memory Card when reading data from the Memory Card during backup.

Type	Name	Extension	Description	Explanation
Data file	BACKUP	.IOM	DM Area words allocated to Special I/O Units and CPU Bus Units	• Contains DM data from D20000 to D32767. • This file must exist on the Memory Card when reading data from the Memory Card during backup.
	BACKUPIO	.IOR	I/O memory data areas	• Contains all of the data in the CIO, WR, HR, and AR data areas as well as timer/counter Completion Flags and PVs. (See note.) • This file must exist on the Memory Card when reading data from the Memory Card during backup.
	BACKUPDM	.IOM	General-purpose DM Area	• Contains DM data from D00000 to D19999. • This file must exist on the Memory Card when reading data from the Memory Card during backup.
	BACKUPE□	.IOM	General-purpose EM Area	Contains all of the EM data for EM bank □ with addresses ranging from E□_00000 to E□_32767. (The maximum bank number depends upon the model of CPU Unit being used.) This file must exist on the Memory Card when reading data from the Memory Card during backup. • When data is backed up to the Memory Card, all of the data in each EM bank is automatically written to a separate file.
	BACKUPDM	.FSR	DM area force-set/force-reset data	• Contains all force-set/force-reset data from D00000 to D32767. • This file must exist on the Memory Card when reading data from the Memory Card during backup.
	BACKUPE□	.FSR	EM area force-set/force-reset data	• Contains all force-set/force-reset data from E□_00000 to E□_32767. • This file must exist on the Memory Card when reading data from the Memory Card during backup.
Change log file	BACKUP	.TRK	Program change log	• Log of program and variable revisions. • This file must exist on the Memory Card when reading data from the Memory Card during backup.
Unit backup files	BACKUP□□ (where □□ is the unit address of the Unit being backed up)	.PRM	Data for specific Unit	• Control backup data from one Unit. Refer to *14-2-1 Simple Backup Function* for details.
Parameter files	BACKUP	.PRT	CIP write-protection settings file for writing over the network	• Contains CIP write-protection settings for writing over the network. • These settings are made with another Programming Device.
	BACKUP	.LST	Required file list	• This file indicates which files are required and which files are optional.

System Files

<table>
<tr><th>Type</th><th>Name</th><th>Extension</th><th>Description</th><th>Explanation</th></tr>
<tr><td rowspan="2">Comments</td><td colspan="2">Automatically generated by the system.</td><td rowspan="2">Comments on variables, instructions, rungs, and structural members.</td><td rowspan="2">This file is automatically saved on the Memory Card when data is downloaded from the NE Programmer or executed by an online editor.</td></tr>
<tr><td rowspan="2">******** (8 ASCII characters)</td><td rowspan="2">.PFL</td></tr>
<tr><td>Program data</td><td>Structured text (ST) language source code.</td><td>This file is automatically saved on the Memory Card when data is downloaded from a Programming Device or executed by an online editor.</td></tr>
</table>

Directories

It is possible to access files in subdirectories with NE1S-series PLCs. The maximum length of a directory path is 65 characters. Be sure not to exceed the maximum number of characters when creating subdirectories in the Memory Card with a program such as Windows.

File Sizes

The size of files in bytes can be calculated with the equations in the following table.

File type	File size
Data files (.IOM)	(Number of words × 2) + 48 bytes Example: Entire DM Area (D00000 to D32767) (32,768 words × 2) + 48 = 65,584 bytes
Data files (.TXT or .CSV)	The file size depends upon the number of delimiters and carriage returns being used. The delimiter code is one byte and the carriage return code is two bytes. Example 1: Non-delimited words, no carriage return 123456789ABCDEF012345678 occupies 24 bytes. Example 2: Delimited words, carriage return every 2 fields 1234,5678↵ 9ABC,DEF0↵ 1234,5678↵ occupies 33 bytes. Example 3: Delimited double words, carriage return every 2 fields 56781234,DEF01234↵ 56781234↵ occupies 29 bytes.
Program files (.OBJ)	(Number of steps used × 4) + 48 bytes (See note.)
Parameter files (.STD)	16,048 bytes

Note Five directories (IOGROUP, PRGCHT, SRC, TYPCHT, and VARCHT) are created on the Memory Card for system files (comments and program data). The file names saved in the directories are in the format ********.PFL. The "********" part is automatically generated.

General-purpose Files

1,2,3... 1. General-purpose data files have filename extensions IOM, TXT, or CSV.

Extension	Data format	Contents	Words/field
.IOM	Binary	NE1S-series data format	---

Extension	Data format	Contents		Words/field
.TXT (See notes.)	Non-delimited words	ASCII format	This format is created by converting one-word fields of I/O memory (4-digit hexadecimal) to ASCII and packing the fields without delimiters. Records can be delimited with carriage returns.	1 word
	Non-delimited double words		This format is created by converting two-word fields of I/O memory (8-digit hexadecimal) to ASCII and packing the fields without delimiters. Records can be delimited with carriage returns.	2 words
	Tab-delimited words		This format is created by converting one-word fields of I/O memory (4-digit hexadecimal) to ASCII and delimiting the fields with tabs. Records can be delimited with carriage returns.	1 word
	Tab-delimited double words		This format is created by converting two-word fields of I/O memory (8-digit hexadecimal) to ASCII and delimiting the fields with tabs. Records can be delimited with carriage returns.	2 words
.CSV (See notes.)	Comma-delimited words		This format is created by converting one-word fields of I/O memory (4-digit hexadecimal) to ASCII and delimiting the fields with commas. Records can be delimited with carriage returns.	1 word
	Comma-delimited double words		This format is created by converting two-word fields of I/O memory (8-digit hexadecimal) to ASCII and delimiting the fields with commas. Records can be delimited with carriage returns.	2 words

Note a) Reading and Writing TXT and CSV Data Files:
TXT and CSV data files can be read and written with FREAD(700) and FWRIT(701) only.

b) Precautions on Characters:
Data cannot be written to I/O memory properly if the TXT or CSV file contains characters other than hexadecimal characters (0 to 9, A to F, or a to f.)

c) Precautions on Field Size:
When words are being used, data cannot be written to I/O memory properly if the TXT or CSV file contains fields that are not 4-digit hexadecimal. Likewise, when double words are being used, data cannot be written properly if the file contains fields that are not 8-digit hexadecimal.

d) Storage Order:
When words are being used, I/O memory data is converted to ASCII and stored in one-word fields in order from the lowest to the highest I/O memory address.
When double words are being used, I/O memory data is converted to ASCII and stored in two-word fields in order from the lowest to the highest I/O memory address. (Within the two-word fields, the higher-address word is stored first and the lower-address word is stored second.)

e) Delimiters:
When there are no delimiters, the fields are packed consecutively and then stored. When delimited by commas, commas are inserted between fields before they are stored. When delimited by tabs, tab codes are inserted between fields before they are stored.
When delimiters (commas or tabs) are specified in FREAD(700), the data is read as delimited data with one-word delimiters (commas or tabs).

f) Carriage Returns:
Data is packed consecutively when carriage returns are not used. When carriage returns are used, a carriage return code is inserted after the specified number of fields. An offset from the beginning of the file (starting read word or starting write word) cannot be specified in the FREAD(700)/FWRIT(701) instructions if carriage returns are used in the file.

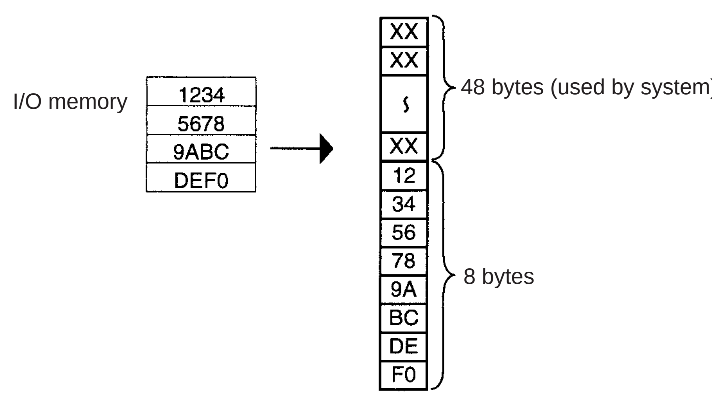
g) Number of Fields:
The overall amount of data in the file depends upon the number of fields (number of write items) specified in the FWRIT(701) instruction and the number of words/field. There is one word/field when words are used and two words/field when double words are used.

2. Data files do not contain information indicating what data is stored, i.e., what memory area is stored. Be sure to give file names that indicate the contents, as shown in the examples below, to aid in file management.
   Examples: D00100.IOM, CIO0020.IOM

Note Data files with the TXT and CSV format contain hexadecimal (0 to 9, A to F) data that allows the I/O memory numerical data to be exchanged with spreadsheet programs.

Binary (.IOM)

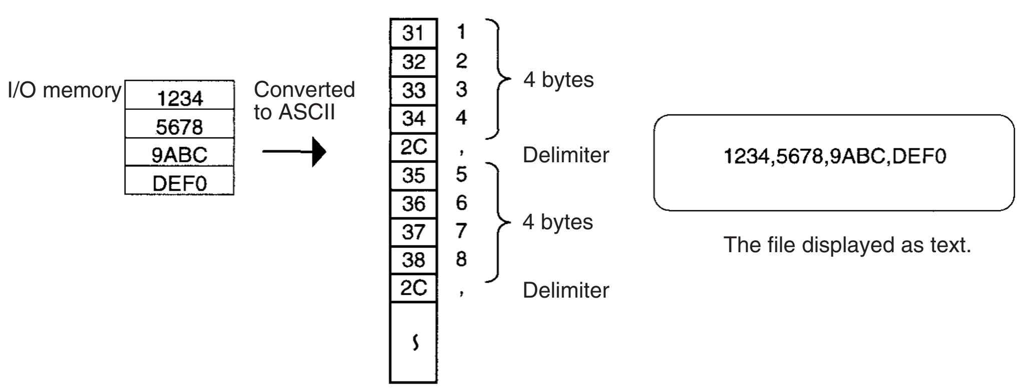
The following illustration shows and example of the binary data structure of a data file (ABC.IOM) containing four words from I/O memory: 1234 Hex, 5678 Hex, 9ABC Hex, and DEF0 Hex. The user, however, does not have to consider the data format in normal operations.



Contents of ABC.IOM

CSV/TXT Data File Structure (Single Word)

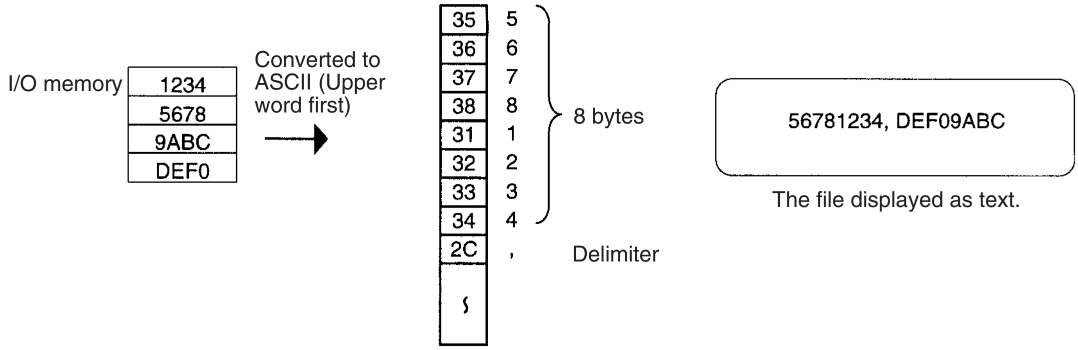
The following illustration shows an example of the data structure of a CSV data file (ABC.CSV) with single-word fields containing four words from I/O memory: 1234 Hex, 5678 Hex, 9ABC Hex, and DEF0 Hex. The structure of the TXT file with single-word fields is the same.



Contents of ABC.CSV

CSV/TXT Data File Structure (Double Word)

The following illustration shows an example of the data structure of a CSV data file (ABC.CSV) with double-word fields containing four words from I/O memory: 1234 Hex, 5678 Hex, 9ABC Hex, and DEF0 Hex. The structure of the TXT file with double-word fields is the same.



Contents of ABC.CSV

Creating Data Files with Spreadsheet Software

Use the following procedure to create TXT and CSV data files with spreadsheet software such as Microsoft Excel.

- Set the cell contents to numeric or characters.
- Input 4 characters in each cell if single-word fields are being used or 8 characters if double-word fields are being used. For example, if single-word fields are being used input 000A, not just A.
- Be sure to input only hexadecimal characters (0 to 9, A to F, or a to f) in the cells. Other characters and codes cannot be used.

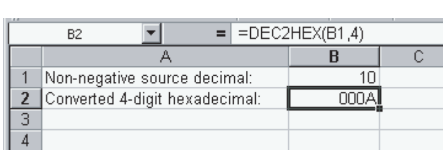
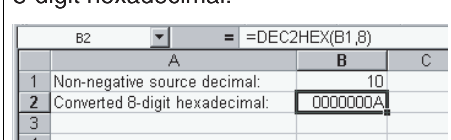
When you want to store hexadecimal digits in I/O memory, it is helpful to convert the spreadsheet's decimal inputs to hexadecimal. Use the following procedure to convert to hexadecimal.

1,2,3...

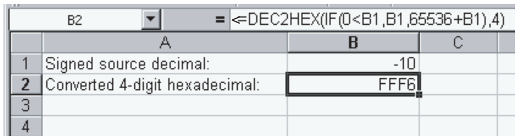
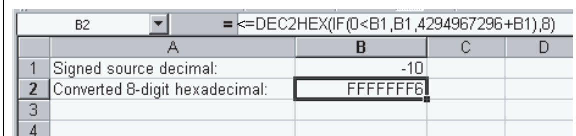
1. Select ***Add-Ins...*** from the Tools Menu.
2. Select **Analysis ToolPak** in the Add-Ins Menu.
3. Select ***Function*** from the Insert Menu at the cell where the function will be used.
4. Select ***DEC2HEX (number, digits)*** from Engineering in the Category Field.
5. When converting to 4-digit hexadecimal, input the following at the number variable: IF(0<=cell location,cell location,65535+cell location)

   When converting to 8-digit hexadecimal, input the following at the number variable: IF(0<=cell location,cell location,4294967296+cell location)

- **Example 1:** Inputting non-negative decimal values.

Item	Converting unsigned decimal to 4-digit hexadecimal	Converting unsigned decimal to 8-digit hexadecimal
Function used	DEC2HEX(*cell_location*,4)	DEC2HEX(*cell_location*,8)
Example	Input 10 in decimal and convert to 000A in 4-digit hexadecimal.  	Input 10 in decimal and convert to 0000000A in 8-digit hexadecimal. 

- **Example 2:** Inputting signed decimal values.

Item	Converting signed decimal to 4-digit hexadecimal	Converting signed decimal to 8-digit hexadecimal
Function used	DEC2HEX(IF(0<=*cell_location*,*cell_location*,65536+ *cell_location*),4)	DEC2HEX(IF(0<=*cell_location*,*cell_location*, 4294967296+*cell_location*),8)
Example	Input -10 in decimal and convert to FFF6 in 4-digit hexadecimal. 	Input -10 in decimal and convert to FFFFFFF6 in 8-digit hexadecimal. 

Backup Data Files

The backup function creates 6 kinds of data files as described below.

To backup data, turn pin 7 ON and turn pin 8 OFF on the CPU Unit's DIP switch, insert the Memory Card, and press and hold the Memory Card Power Supply Switch for three seconds. The four backup files (BACKUP.IOM, BACKUPIO.IOR, BACKUPDM.IOM, BACKUPE□.IOM, BACKUPDM.FSR, and BACKUPE0.FSR) will be created automatically and written to the Memory Card.

14-1-4 Description of File Operating Procedures

The following table summarizes the 4 methods that can be used to read and write files.

Read: Transfers files from file memory to the CPU Unit.
Write: Transfers files from the CPU Unit to file memory.

Operating procedure	Medium	File type/name	Description	Entire program	Variable data file	Parameter Area data	Data Area data (See note.)	Change log file (to be supported soon)	Comment, program data
Simple backup operation	Memory Card	Simple backup files BACKUP□□	Read	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	No
			Write	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	No
FREAD(700) and FWRIT(701) (READ/WRITE DATA FILE) instructions		General-purpose files (********)	Read data from one file.	No	No	No	**Yes**	No	No
Transfer by FTP over an Ethernet network		General-purpose files (********)	Read	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	No
			Write	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	No
			Write data to one file.	No	No	No	**Yes**	No	No
Generated by the system (automatic)		Files automatically generated by system	Write	No	No	No	No	**Yes**	**Yes**

Note Data files with the TXT or CSV formats can be read and written only with the FREAD(700) and FWRIT(701) instructions. They cannot be read and written with the NE Programmer.

14-2 Manipulating Files

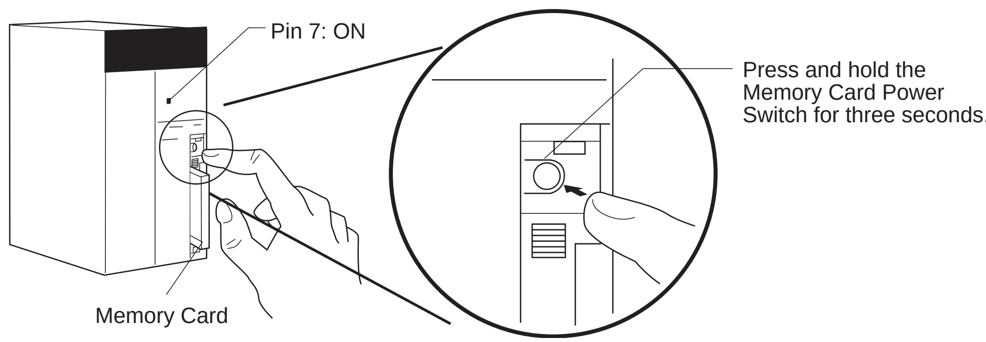
The following procedures are used to read, write and otherwise work with files using the following methods.

- Simple backup function
- FREAD(700) and FWRIT(701) instructions in the user program
- Transfer by FTP over an Ethernet network
- System generation

14-2-1 Simple Backup Function

Backing Up Data from the CPU Unit to the Memory Card

To backup data, turn ON pin 7 on the CPU Unit’s DIP switch, press and hold the Memory Card Power Supply Switch for three seconds. The simple backup function will automatically create backup files and write them to the Memory Card. The backup files contain the program, parameter area data, and I/O memory data. This function can be executed in any operating mode.



Restoring Data from the Memory Card to the CPU Unit

To restore the backup files to the CPU Unit, check that pin 7 is ON and turn the PLC’s power OFF and then ON again. The backup files containing the program, parameter area data, and I/O memory data will be read from the Memory Card to the CPU Unit.

Note

1. The simple backup function will override the automatic transfer at startup function, so the backup files will be read to the CPU Unit when the PLC is turned ON even if pin 2 of the DIP switch is ON.
2. Data will not be read from the Memory Card to the CPU Unit if pin 1 of the DIP switch is ON (write-protecting program memory).
3. When a simple backup operation first starts, it may not be possible to read/write variables.
4. When the backup files are read from the Memory Card by the simple back-up function, the status of I/O memory and force-set/force-reset bits will be cleared unless the necessary settings are made in the Auxiliary Area and PLC Setup.

   If the IOM Hold Bit (A50012) is ON and the PLC Setup is set to maintain the IOM Hold Bit Status at Startup when the backup files are written, the status of I/O memory data will be maintained when data is read from the Memory Card.

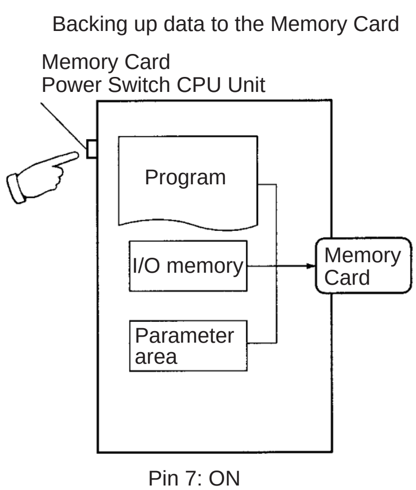
   If the Forced Status Hold Bit (A50013) is ON and the PLC Setup is set to maintain the Forced Status Hold Bit Status at Startup when the backup files are written, the status of force-set and force-reset bits will be maintained when data is read from the Memory Card.

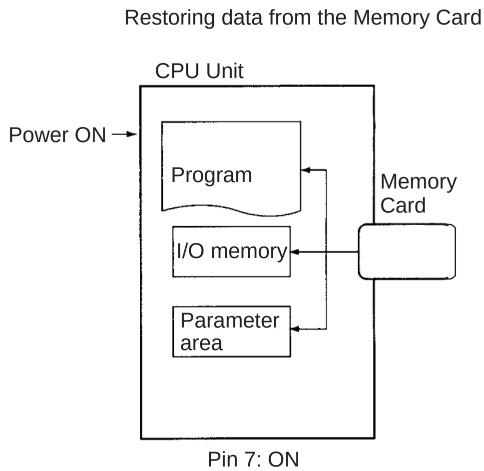
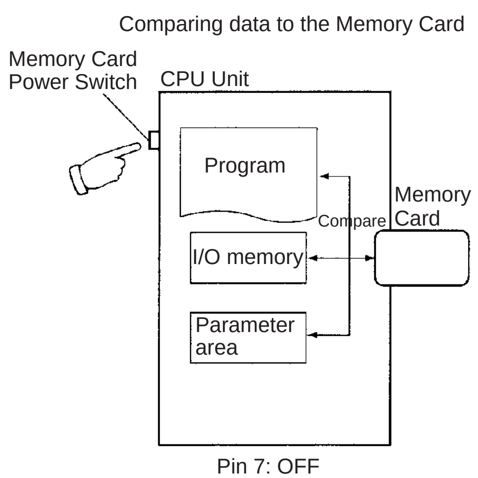
5. A CS1-H, CJ1-H, CJ1M, or CS1D CPU Unit will remain in PROGRAM mode after the simple backup operation has been performed and cannot be changed to MONITOR or RUN mode until the power supply has been cycled. After completing the simple backup operation, turn OFF the power supply to the CPU Unit, changes the settings of pin 7, and then turn the power supply back ON.
6. File backups may take from several seconds to several minutes.

Comparing Data in the Memory Card and CPU Unit

To compare the backup files in the Memory Card with the data in the CPU Unit, turn OFF pin 7 on the CPU Unit's DIP switch, and press and hold the Memory Card Power Supply Switch for three seconds. The backup function will compare the program, parameter area data, and I/O memory data in the Memory Card with the corresponding data in the CPU Unit. This function can be executed in any operating mode.

The following table provides a summary of the simple backup operations.

Backup operation	Pin status	Procedure
	Pin 7	
Backing up data from the CPU Unit to the Memory Card  	ON	Press and hold the Memory Card Power Switch for three seconds.

Backup operation	Pin status Pin 7	Procedure
Restoring data from the Memory Card to the CPU Unit  	ON	Turn the PLC OFF and ON again. This operation is given priority over automatic transfer at startup.
Comparing data between the CPU Unit and the Memory Card  	OFF	Press and hold the Memory Card Power Switch for three seconds.

Note

1. Refer to *Verifying Simple Backup Operations with Indicators* on page 419 for details on the results of read, write, and compare operations.
2. Refer to *14-2-1 Simple Backup Function* for guidelines on the time required for Memory Card backup operations.

Backup Files

Data Files

File name and extension	Data area and range of addresses stored		Backup from I/O memory to Memory Card (creating files)	Restore from Memory Card to I/O memory	Comparing Memory Card to I/O memory		Files required when restoring data
CPU Unit	**NE1S Series**				**CS1/ CJ1**	**CS1-H**	
BACKUP.IOM	DM	D20000 to D32767	Yes	Yes	Yes	---	Required in Memory Card
BACKUPIO.IOR	CIO	0000 to 6143 (Including forced bit status.)	Yes	--- (See note 2.)	Yes	---	Required in Memory Card
	WR	W000 to W511 (Including forced bit status.)	Yes	--- (See note 2.)	Yes	---	
	HR	H000 to H511	Yes	Yes	Yes	---	
	AR	A000 to A447	Yes	---	---	---	
		A448 to A959	Yes	Yes	Yes	---	
	Timer (Completion Flags and PVs)	T0000 to T4095	Yes	Yes (See note 2.)	Yes	---	
	Counter (Completion Flags and PVs)	C0000 to C4095	Yes	Yes	Yes	---	
BACKUPDM.IOM	DM	D00000 to D19999	Yes	Yes	Yes	---	Required in Memory Card
BACKUPE□.IOM	EM (See note 1.)	E□_00000 to E□_32767 (□: Bank No. from 0 to C, but max. bank No. depends on CPU Unit.)	Yes	Yes	Yes	---	Required in Memory Card (must match CPU Unit)
BACKUPDM.FSR	DM forced status information	Forced status information for D000000 to D32767	Yes	Yes	Yes	---	Required in Memory Card
BACKUPE□.FSR	EM forced status information (See note 1.)	Forced status information for E□_00000 to E□_32767	Yes	Yes	Yes	---	Required in Memory Card

Note 1. The □ represents the bank number and the number of banks depends upon the CPU Unit being used.

When the BACKUPE□.IOM files in the Memory Card are restored to the CPU Unit, the files are read in order beginning with bank 0 and ending with the maximum bank number in the CPU Unit. Excess BACKUPE□.IOM files will not be read if the number of banks backed up exceeds the number of banks in the CPU Unit. Conversely, any remaining EM banks in the CPU Unit will be left unchanged if the number of banks backed up is less than the number of banks in the CPU Unit.

If a BACKUPE□.IOM file is missing (e.g., 0, 2), only the consecutive files will be read. In this case, data would be read to bank 0 only.

2. Normally, the contents of the CIO Area, WR Area, Timer Completion Flags, Timer PVs, and the status of force-set/force-reset bits will be cleared when the PLC is turned ON and BACKUPIO.IOR is read from the Memory Card.

   If the IOM Hold Bit (A50012) is ON and the PLC Setup is set to maintain the IOM Hold Bit Status at Startup when the backup files are written, the status of I/O memory data will be maintained when data is read from the Memory Card.

   If the Forced Status Hold Bit (A50013) is ON and the PLC Setup is set to maintain the Forced Status Hold Bit Status at Startup when the backup files are written, the status of force-set and force-reset bits will be maintained when data is read from the Memory Card.

Program Files

File name and extension	Contents	Backup from I/O memory to Memory Card (creating files)	Restore from Memory Card to I/O memory	Comparing Memory Card to I/O memory	Files required when restoring data
CPU Unit	**NE1S**				
BACKUP.OBJ	Entire user program	Yes	Yes	Yes	Required in Memory Card
BACKUP.PFL	Program data	Yes	Yes	Yes	

Variable Data Files

File name and extension	Contents	Backup from I/O memory to Memory Card (creating files)	Restore from Memory Card to I/O memory	Comparing Memory Card to I/O memory	Files required when restoring data
BACKUP.USV BACKUP.TNS BACKUP.VTP	Variable data	Yes	Yes	Yes	Required in Memory Card

Parameter Files

File name and extension	Contents	Backup from I/O memory to Memory Card (creating files)	Restore from Memory Card to I/O memory	Comparing Memory Card to I/O memory	Files required when restoring data
CPU Unit	**NE1S**				
BACKUP.STD	PLC Setup Registered I/O tables	Yes	Yes	Yes	Required in Memory Card
BACKUP.ETP	Built-in Ethernet Setup	Yes	Yes	Yes	
BACKUP.PRT	CIP Service Protection Settings (CIP write-protection settings for writing over the network) (Rev. 3.1 or later only)	Yes	Yes	Yes	

Change Log Files

File name and extension	Contents	Backup from I/O memory to Memory Card (creating files)	Restore from Memory Card to I/O memory	Comparing Memory Card to I/O memory	Files required when restoring data
BACKUP.TRK	Change log	Yes	Yes	Yes	Required in Memory Card
BACKUP.LST	Data file list (Indicates which files must be present in the Memory Card and which files are optional.)	Yes	Yes	Yes	

Unit Backup Files

File name and extension	Contents	Backup from I/O memory to Memory Card (creating files)	Restore from Memory Card to I/O memory	Comparing Memory Card to I/O memory	Files required when restoring data
BACKUP□□.PRM (where □□ is the unit address of the Unit being backed up)	Backup data from the Unit with the specified unit address (Specific contents depends on the Unit.)	Yes	Yes	Yes	Required in Memory Card (See note 2.)

Note
1. Unit addresses are as follows:
   CPU Bus Units: Unit number + 10 Hex
   Special I/O Units: Unit number + 20 Hex
2. An error will not occur in the CPU Unit even if this file is missing when data is transferred from the Memory Card to I/O memory, but an error will occur in the Unit if the data is not restored. Refer to the operation manual for the specific Unit for details on Unit errors.

Simple Backup Function

There are 3 backup operations: backing up data to the Memory Card, restoring data from the Memory Card, and comparing data with the Memory Card.

Backing Up Data from the CPU Unit to the Memory Card

1,2,3...
1. Insert a Memory Card into the CPU Unit. (Already initialized.)
2. Turn ON pin 7 and turn OFF pin 8 on the CPU Unit's DIP switch.
3. Press and hold the Memory Card Power Supply Switch for three seconds.
4. Verify that the MCPWR Indicator flashes once and then goes OFF. (Other changes indicate that an error occurred while backing up the data.)

Restoring Data from the Memory Card to the CPU Unit

1,2,3...
1. Insert the Memory Card containing the backup files into the CPU Unit.
2. Turn ON pin 7 and turn OFF pin 8 on the CPU Unit's DIP switch.
3. The backup files will be restored when the PLC is turned ON.
4. Verify that the MCPWR Indicator flashes once and then goes OFF. (Other changes indicate that an error occurred while restoring the data.)

Comparing Data in the Memory Card and CPU Unit

1,2,3...
1. Insert the Memory Card containing the backup files into the CPU Unit.
2. Turn OFF pins 7 and 8 on the CPU Unit's DIP switch.
3. Press and hold the Memory Card Power Supply Switch for three seconds.
4. The data matches if the MCPWR Indicator flashes once and then goes OFF.

Note The MCPWR Indicator will flash if an error occurs while writing or comparing data. This flashing will stop and the MCPWR Indicator will be lit when the Memory Card Power Supply Switch is pressed.

The following table shows the time required for backup operations with a 20-Kstep Program and 10-ms Cycle Time in RUN mode:

Mode	Backing up	Restoring	Comparing
PROGRAM	Approx. 50 s	Approx. 30 s	Approx. 7 s
RUN	Approx. 5 min	Approx. 2 min	Approx. 7 s

The following table shows the time required for backup operations with a 30-Kstep Program and 10-ms Cycle Time in RUN mode:

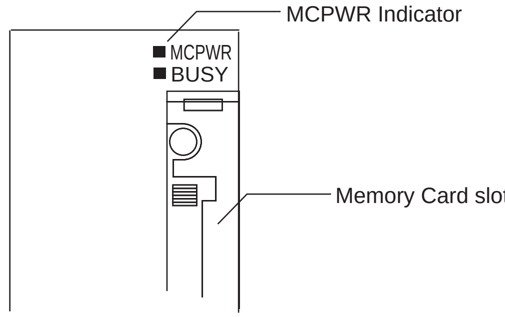
Mode	Backing up	Restoring	Comparing
PROGRAM	Approx. 50 s	Approx. 30 s	Approx. 7 s
RUN	Approx. 5 min 30 s	Approx. 2 min 40 s	Approx. 7 s

The following table shows the time required for backup operations with a 250-Kstep Program and 12-ms Cycle Time in RUN mode:

Mode	Backing up	Restoring	Comparing
PROGRAM	Approx. 1 min 30 s	Approx. 1 min 30 s	Approx. 20 s
RUN	Approx. 13 min	Approx. 7 min 30 s	Approx. 20 s

Verifying Simple Backup Operations with Indicators

The status of the Memory Card Power (MCPWR) indicator shows whether a simple backup operation has been completed normally or not.



Backup operation	Normal completion (See note 1.)	Error occurred	
	MCPWR status	**MCPWR status**	**Error**
Backing up data from the CPU Unit to the Memory Card	Lit → Remains lit while the Memory Card Power Switch is pressed. → Flashes once. → Lit while writing. → OFF after data is written.	Lit → Remains lit while the Memory Card Power Switch is pressed. → Remains flashing. → Lights when the Memory Card Power Switch is pressed.	No files will be created with the following errors: Insufficient Memory Card capacity (See note 2.) Memory error in CPU Unit I/O bus error (when writing data to a Unit)
Restoring data from the Memory Card to the CPU Unit	Lit when power is turned ON. → Flashes once. → Lit while reading. → OFF after data is read.	Lit when power is turned ON. → Flashes five times. → Goes OFF.	Data won't be read with the following errors: Program in Memory Card exceeds CPU Unit capacity Required backup files do not exist in Memory Card. Program can't be written because it is write-protected (Pin 1 of the DIP switch is ON.)
		Lit when power is turned ON. → Flashes once. → Lit while reading. → Flashes three times. → OFF after data is read.	Caution: Data will be read with the following error. EM files and CPU Unit EM banks do not match (non-consecutive bank numbers or max. bank number mismatch).

Backup operation	Normal completion (See note 1.)	Error occurred	
	MCPWR status	MCPWR status	Error
Comparing data between the CPU Unit and the Memory Card	Lit → Remains lit while the Memory Card Power Switch is pressed. → Flashes once. → Lit while comparing. → OFF after data is compared.	Lit → Remains lit while the Memory Card Power Switch is pressed. → Remains flashing. → Lights when the Memory Card Power Switch is pressed.	The following comparison errors can occur (See note 2.): Memory Card and CPU Unit data do not match. Required backup files do not exist in Memory Card. EM files and CPU Unit EM banks do not match (non-consecutive bank numbers or max. bank number mis-match). Memory error in CPU Unit I/O bus error (when comparing data to a Unit)
Common to all three backup operations.	---	Reading: Flashes five times. → Goes OFF. → Lights when the Memory Card Power Switch is pressed. Writing or comparing: Remains flashing.→ Lights when the Memory Card Power Switch is pressed.	Memory Card access error (format error or read/write error)

Note 1. When the backup operation is completed normally, power to the Memory Card will go OFF when the MCPWR indicator goes OFF. If the Memory Card will be used again, press the Memory Card Power Switch to supply power and execute the desired operation.

2. The backup files for Units are also compared.

Related Auxiliary Bits/Words

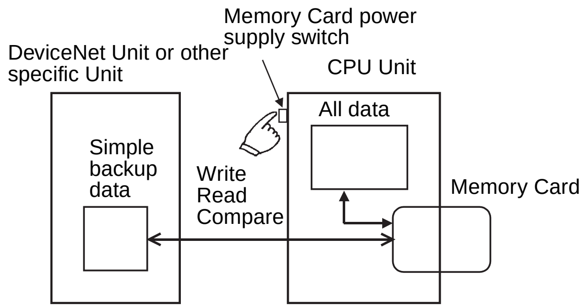
Name	Address	Description
File Memory Operation Flag	A34313	ON when any of the following are being performed. OFF when execution has been completed. • Memory Card detection • CMND instruction executed for local CPU Unit • FREAD/FWRIT instructions • Program replacement via special control bits • Simple backup operation Wiring data to or verifying the contents of the Memory Card is not possible while this flag is ON.
Communications Port Enabled Flags	A20200 to A20207	• Turns OFF when writing or comparing Memory Card data begins. • Turn ON when writing or comparing Memory Card data has been completed. Unit data cannot be written or compared if all of the Communications Port Enabled Flags are OFF when Memory Card write or compare operations are started and an error will occur if this is attempted.
Communications Port Completion Codes	A203 to A210	Provide the results of communications with the Unit when Memory Card write or compare operations are performed.
Communications Port Error Flags	A21900 to A21907	• Turns ON is an error occurs in communications with the Unit when Memory Card write or compare operations are performed. • Remains OFF (or turns OFF) is no error occurs in communications with the Unit when Memory Card write or compare operations are performed.

Note These bits and words in the Auxiliary Area are related because the CPU Unit will automatically use an available communications port when writing or comparing data for a Memory Card using the simple backup operation.

Backing Up Specific Unit Data

Introduction

When the specific Unit is mounted to the CPU Unit, the data in the specific Unit will also be backed up by the simple backup operation. The data within each Unit is also backed up separately for each individual Unit.



Application

This function can be used to back up data for the entire PLC, including the CPU Unit, DeviceNet Units, Serial Communications Units, Motion Control Units, etc. It can also be used for Unit replacement.

Unit Backup Files

The data from each Unit is stored in the Memory Card using the following file names: BACKUP□□.PRM. Here, "□□" is the unit address of the Unit in hexadecimal.

Note Unit addresses are as follows:
CPU Bus Units: Unit number + 10 hex
Special I/O Units: Unit number + 20 hex

These files are also used when reading from the Memory Card or comparing Memory Card data.

Applicable Units

For Unit data to be backed up, the Unit must also support the backup function. Refer to the operation manual for the Unit for details on support.

Unit	Model numbers	Data backed up for simple backup when used with CPU Unit	Data capacity used in Memory Card for simple backup
ControlNet Units	NE1S-CNS21U	Device parameters	171 Kbytes
DeviceNet Units	NE1S-DRM21U	Device parameters (all data in EEPROM in the Unit) (Although this is the same data as is backed up from the Memory Card backup function supported by the Unit or the DeviceNet Configuration (Ver. 2.□), there is no file compatibility.)	7 Kbytes

Unit	Model numbers	Data backed up for simple backup when used with CPU Unit	Data capacity used in Memory Card for simple backup
Serial Communications Units	CS1W-SCU21-V1 CJ1W-SCU41	Protocol macro data (Including both standard system protocols and user-defined protocols from the flash memory in the Unit)	129 Kbytes
Motion Control Units	CS1W-MCH71	• Positioning data • System parameters • G language programs	8,192 Kbytes
	CS1W-MC221 CS1W-MC421-R1		142 Kbytes
Position Control Units (NC Units)	CS1W-NC113/133/213/233/413/433 Unit Ver. 2.0 or later	• Axis parameters • Sequence data • Speed data • Acceleration/deceleration time data • Dwell timer data • Zone data	7 Kbytes

Note
1. Data from the Units listed above will be automatically backed up for the simple backup operation. There is no setting available to include or exclude them.
2. When the CS1W-SCU21-V1 is being used, the time required for a simple backup operation will be longer than when the CS1W-SCU21-V1 is not being used by the times given in the following tables.

Additional Time when the CPU Bus Unit Settings File (BACKUP□□.PRM) on the Memory Card Is 60 Kbytes

Operating mode	Additional time when writing to a Memory Card	Additional time when verifying a Memory Card	Additional time when reading from a Memory Card
PROGRAM	Approx. 25 s	Approx. 10 s	Approx. 4 s
RUN	Approx. 1 min 30 s	Approx. 30 s	Approx. 4 s

Additional Time when the CPU Bus Unit Settings File (BACKUP□□.PRM) on the Memory Card Is 128 Kbytes

Operating mode	Additional time when writing to a Memory Card	Additional time when verifying a Memory Card	Additional time when reading from a Memory Card
PROGRAM	Approx. 40 s	Approx. 14 s	Approx. 8 s
RUN	Approx. 2 min 30 s	Approx. 1 min	Approx. 8 s

Note
1. Confirm that the Units are running properly before attempting the above operations. The write, read, and compare operations will not be performed unless the Units are running properly.
2. Before performing a simple backup operation for specific Units, make sure either that the CPU Unit is in PROGRAM mode or that performing the simple backup operation will not adversely affect instructions that use communications port numbers. When data is backed up from specific Units, a communications port will be searched for beginning from port 0 and the first available port will be used. If the port number is the same as one used by a network communications instruction, the network communications instruction will not be executed until the simple backup operation has been completed.

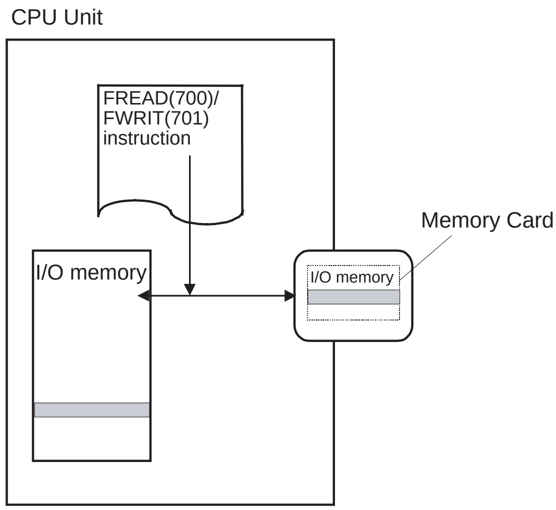
14-2-2 FREAD(700) and FWRIT(701)

The FWRIT(701) (WRITE DATA FILE) instruction can be used to create a data file containing the specified I/O memory data in a Memory Card. It can also add to or overwrite from any point in existing files.

The FREAD(700) (READ DATA FILE) instruction will read I/O memory data from a specified location from a data file in a Memory Card and write it to the specified portion of I/O memory. It can read from any point in the specified file.

Note These instructions do not transfer the specified file, but rather the specified amount of data beginning at the specified start position in the file.

FREAD(700)/FWRIT(701): Transfers between I/O memory and file memory



FREAD(700)/FWRIT(701) Instructions

Instructions

Name	Mnemonic	Description
READ DATA FILE	FREAD(700)	Reads specified data file data or data elements to specified I/O memory.
WRITE DATA FILE	FWRIT(701)	Uses specified I/O memory area data to create a specified data file.

The data file format is specified as binary data (.IOM files) non-delimited or tab-delimited text (.TXT files), or comma-delimited text (.CSV files) according to the operand settings of FREAD(700) and FWRIT(701), as shown in following table.

Bits in C	Settings	Programming Device limitations
12 to 15	Data type 0: Binary (.IOM) 1: Non-delimited words (.TXT) 2: Non-delimited double-words (.TXT) 3: Comma-delimited words (.CSV) 4: Comma-delimited double-words (.CSV) 5: Tab-delimited words (.TXT) 6: Tab-delimited double-words (.TXT)	0 to 6 hex
08 to 11	Carriage returns 0: No returns 8: Return every 10 fields 9: Return every 1 field A: Return every 2 fields B: Return every 4 fields C: Return every 5 fields D: Return every 16 fields	0 or 8 to D hex

Note The time from the CPU Unit's internal clock is used to date files created in file memory with FWRIT(701).

The FREAD(700) and FWRIT(701) instructions cannot be executed when another FREAD(700) or FWRIT(701) instruction is being executed or the simple backup operation is being performed. Therefore, use the File Memory Operation Flag (A34313) for exclusive control of file memory instructions in the program.

During FREAD(700) execution, the File Read Error Flag (A34310) will turn ON (the Error Flag will remain OFF) if the specified file contains the wrong data type or the file data is corrupted (either non-hexadecimal data, or data other than 4-digit/8-digit data). Data will be read up to the point where an illegal character is detected.

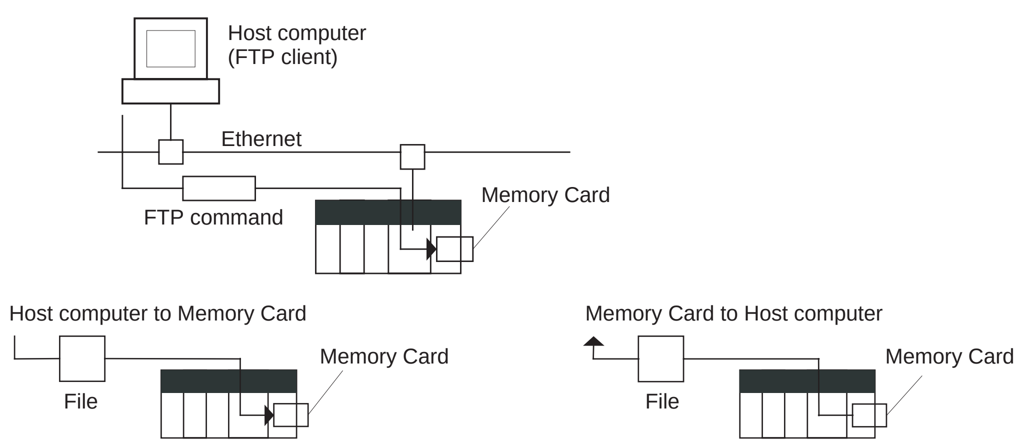
Related Auxiliary Bits/Words

Name	Address	Operation
Memory Card Type	A34300 to A34302	Indicates the type of Memory Card, if any, that is installed.
Memory Card Format Error Flag	A34307	ON when the Memory Card is not formatted or a formatting error has occurred.
File Write Error Flag	A34308	ON when an error occurred when writing to the file.
File Write Impossible Flag	A34309	ON when the data couldn't be written because the file was write-protected or there was insufficient free memory.
File Read Error Flag	A34310	ON when a file could not be read because its data was corrupted or if it contains the wrong data type.

Name	Address	Operation
No File Flag	A34311	ON when data could not be read because the specified file doesn't exist.
File Memory Operation Flag	A34313	ON for any of the following: The CPU Unit is processing a FINS command sent to itself using CMND(490). FREAD(700) or FWRIT(701) is being executed. The program is being overwritten using an Auxiliary Area control bit. A simple backup operation is being performed.
Accessing File Flag	A34314	ON when file data is actually being accessed.
Memory Card Detected Flag	A34315	ON when a Memory Card has been detected. OFF when a Memory Card cannot be detected.
Number of Items to Transfer	A346 to A347	These words indicate the number of words or fields remaining to be transferred (32 bits). When a binary (.IOM) file is being transferred, this number is decremented each time a word is read. When a text or CSV file is being transferred, this number is decremented each time a field is transferred.

14-2-3 Transferring Files by FTP over an Ethernet Network

The built-in Ethernet port can be used to write or read files to or from the file memory (Memory Card) in the CPU Unit from a personal computer on the Ethernet network. Using the host computer as an FTP client and the NE1S-series Programmable Controller as server, data can be exchanged in file units.



14-2-4 Operations Executed Automatically by the System

Comments, structured text (ST) language source code, and a program change log are automatically saved as system files on the Memory Card when data is downloaded from a Programming Device or executed by an online editor.

14-3 Formatting the Memory Card

OMRON Memory Cards (refer to *14-1-1 Types of File Memory*) are formatted before shipping. They can be used in the NE1S CPU Unit as is. To format them once they have been used, always do so in the CPU Unit using the NE Programmer.

Note If a Memory Card is formatted in a personal computer or other computer, it will not be usable in the NE1S CPU Unit. Always format Memory Cards in the CPU Unit using the NE Programmer.

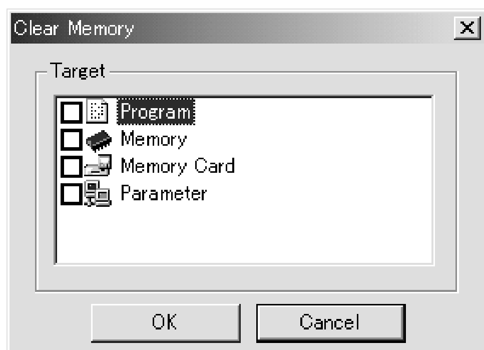
14-3-1 Formatting Memory Cards with the NE Programmer

Use the following procedure to format Memory Cards in the CPU Unit using the NE Programmer.

This operation is done online. Connect the NE Programmer to the CPU Unit before starting.

1,2,3...

1. Select ***Controller - Clear Memory***. The following dialog box will be displayed.



2. Select the ***Memory Card*** option, then click the **OK** Button. The following dialog box will be displayed.

3. Click the **Yes** Button. Formatting will then begin.

SECTION 15
CPU Unit Operation and the Cycle Time

This section describes the internal operation of the CPU Unit and the cycle used to perform internal processing.

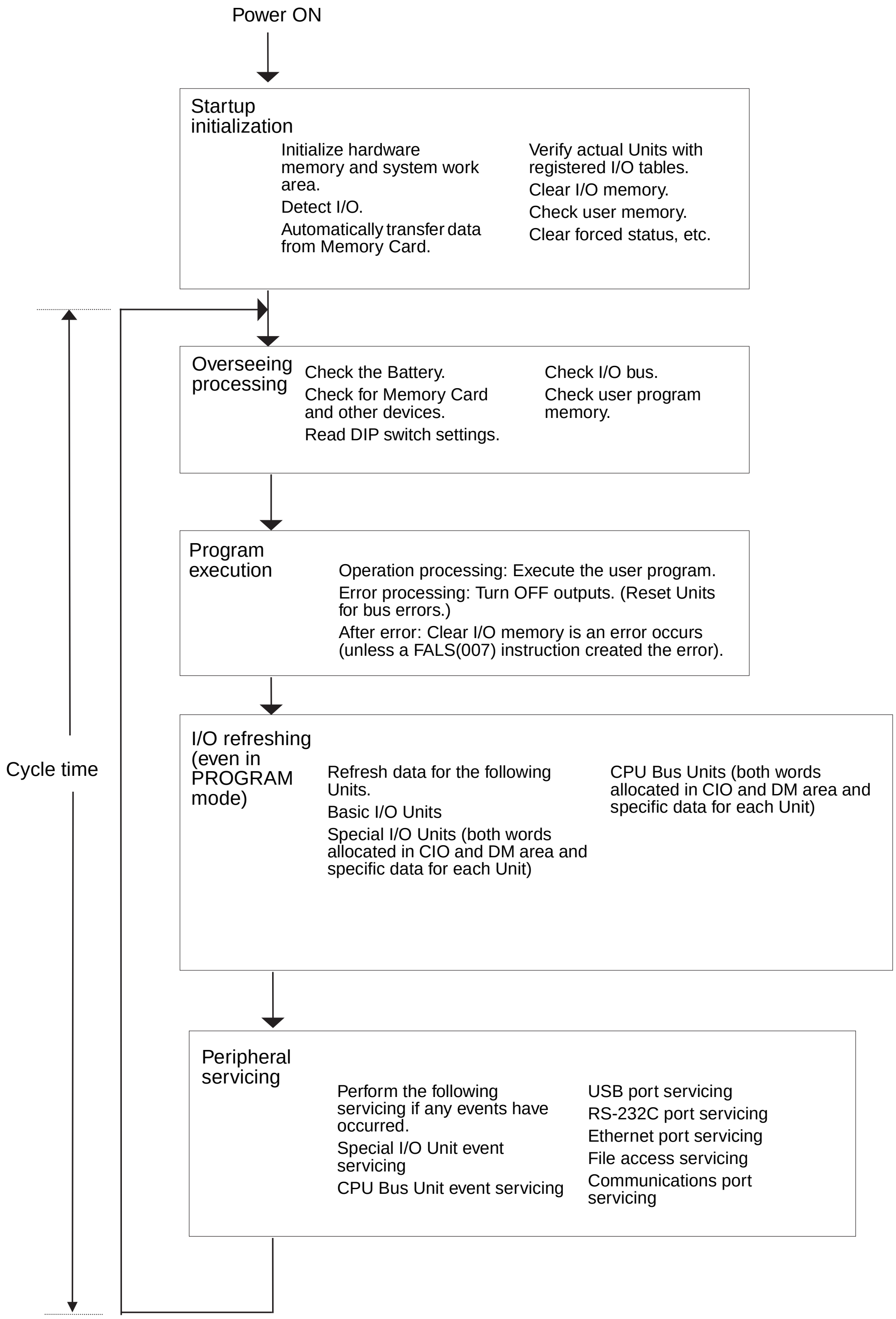
15-1 CPU Unit Operation

15-1-1 General Flow

The following flowchart shows the overall operation of the CPU Unit.

Normal Mode

In the normal mode, the program is executed before I/O is refreshed and peripherals are serviced. This cycle is executed repeatedly.



15-1-2 I/O Refreshing and Peripheral Servicing

I/O Refreshing

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

- Refreshing between Basic I/O Units and I/O words in the CIO Area
- Refreshing between Special I/O Units, CPU Bus Units, and Inner Boards, and the words allocated to these in the CIO Area (and for CPU Bus Units, words allocated in the DM Area)
- Refreshing Unit-specific data for Special I/O Units, CPU Bus Units, and Inner Boards.

All I/O refreshing is performed in the same cycle (i.e., time slicing is not used). I/O refreshing is always performed after program execution.

Units			Max. data exchange	Data exchange area
Basic I/O Units			Depends on the Unit.	I/O Bit Area
Special I/O Units	Words allocated in CIO Area		10 words/Unit (Depends on the Unit.)	Special I/O Unit Area
NE1S-series CPU Bus Units	Words allocated in CIO Area		25 words/Unit	NE1S-series CPU Bus Unit Area
	Words allocated in DM Area		100 words/Unit	NE1S-series CPU Bus Unit Area
	Unit-specific data	ControlNet Unit	Depends on the Unit.	Words set for data links
		NE1S-series DeviceNet Unit	Depends on the Unit.	Words set for remote I/O communications (for either fixed or user-set allocations)
		Serial Communi-cations Unit	Depends on the protocol macros.	Communications data set for protocol macros

Peripheral Servicing

Peripheral servicing involves servicing non-scheduled events for external devices. This includes both events from external devices and service requests to external devices.

Most peripheral servicing for NE1S-series PLCs involved CIP commands. The specific amount of time set in the system is allocated to each type of servicing and executed every cycle. If all servicing cannot be completed within the allocated time, the remaining servicing is performed the next cycle.

Units	Servicing
Event servicing for Special I/O Units	Non-scheduled servicing for FINS commands from Special I/O Units, NE1S-series CPU Bus Units, and Inner Boards (e.g., requests to start external interrupt tasks) Non-scheduled servicing for FINS commands from the CPU Unit to the above Units.
Event servicing for NE1S-series CPU Bus Units	
USB port servicing	Non-scheduled servicing for CIP or Host Link commands received via the USB port or RS-232C ports from Programming Devices, PTs, or host computers (e.g., requests to transfer programming, monitoring, forced-set/reset operations, or online editing
RS-232C port servicing	
Communications port servicing	Servicing to execute network communications, serial communications, or file memory access for the CSND or PMCR instructions using communications ports 0 to 7 (internal logical ports) Servicing to execute network communications specifying information system communications ports 0 and 1 (internal logical ports) using the MLSND instruction. Servicing to execute background execution using communications ports 0 to 7 (internal logical ports)
File access servicing	File read/write operations for Memory Cards.

Note Special I/O Units, NE1S-series CPU Bus Units, RS-232C communications ports, USB ports, and file servicing is allocated 8% of the cycle time by default (the default can be changed). If servicing is separated over many cycles, delaying completion of the servicing, increase the percentage of the service time allocated to peripheral servicing.

15-1-3 Initialization at Startup

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

- Detect mounted Units.
- Compare the registered I/O tables and the actual Units.
- Clear the non-holding areas of I/O memory according to the status of the IOM Hold Bit. (See note 1.)
- Clear forced status according to the status of the Forced Status Hold Bit. (See note 2.)
- Perform self-diagnosis (user memory check).

Note

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

Auxiliary bit / **PLC Setup setting**		**IOM Hold Bit (A50012)**	
		Clear (OFF)	**Hold (ON)**
IOM Hold Bit Status at Startup	**Clear (OFF)**	At power ON: Clear At mode change: Clear	At power ON: Clear At mode change: Hold
	Hold (ON)		At power ON: Hold At mode change: Hold

Mode Change: Between PROGRAMMING mode and RUN or MONITOR mode

2. The forced status held or cleared according to the status of the Forced Status Hold Bit and the setting for Forced Status Hold Bit Status at Startup in the PLC Setup.

Auxiliary bit / **PLC Setup setting**		**Forced Status Hold Bit (A50013)**	
		Clear (OFF)	**Hold (ON)**
Forced Status Hold Bit Status at Startup	**Clear (OFF)**	At power ON: Clear At mode change: Clear	At power ON: Clear At mode change: Hold
	Hold (ON)		At power ON: Hold At mode change: Hold

Mode Change: Between PROGRAMMING mode and RUN or MONITOR mode

15-2 CPU Unit Operating Modes

15-2-1 Operating Modes

The CPU Unit has three operating modes that control the entire user program and are common to all tasks.

PROGRAM: Programs are not executed and preparations, such as creating I/O tables, initializing the PLC Setup and other settings, transferring programs, checking programs, force-setting and force-resetting can be executed prior to program execution.

MONITOR: Programs are executed, but some operations, such as online editing, forced-set/reset, and changes to present values in I/O memory, are enabled for trial operation and other adjustments.

RUN: Programs are executed and some operations are disabled.

15-2-2 Status and Operations in Each Operating Mode

PROGRAM, RUN, and MONITOR are the three operating modes available in the CPU Unit. The following lists status and operations for each mode.

Overall Operation

Mode	Program (See note)	I/O refresh	External outputs	I/O Memory	
				Non-holding areas	Holding areas
PROGRAM	Stopped	Executed	OFF	Clear	Hold
RUN	Executed	Executed	Controlled by program	Controlled by program	
MONITOR	Executed	Executed	Controlled by program	Controlled by program	

Programming Console Operations

Mode	Monitor I/O Memory	Monitor Program	Transfer Program		Check Program	Create I/O Table
			PLC to Programming Device	Programming Device to PLC		
PROGRAM	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**
MONITOR	**OK**	**OK**	**OK**	X	X	X
RUN	**OK**	**OK**	**OK**	X	X	X

Mode	PLC Setup	Modify Program	Force-set/reset	Changing Timer/Counter SV	Changing Timer/Counter PV	Changing I/O Memory PV
PROGRAM	**OK**	**OK**	**OK**	**OK**	**OK**	**OK**
RUN	X	X	X	X	X	X
MONITOR	X	**OK**	**OK**	**OK**	**OK**	**OK**

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

<table>
<tr><th>Mode</th><th>Cyclic task status</th><th>Interrupt task status</th></tr>
<tr><td>PROGRAM</td><td>Disabled status (INI)</td><td>Stopped</td></tr>
<tr><td>RUN</td><td rowspan="2">• Any task that has not yet been executed, will be in disabled status (INI).
• A task will go to READY status if the task is set to go to READY status at startup or the TASK ON (TKON) instruction has been executed for it.
• A task in READY status will be executed (RUN status) when it obtains the right to execute.
• A status will go to Standby status if a READY task is put into Standby status by a TASK OFF (TKOF) instruction.</td><td rowspan="2">Executed if interrupt condition is met.</td></tr>
<tr><td>MONITOR</td></tr>
</table>

Operating Mode Changes and I/O Memory

Mode Changes	Non-holding areas	Holding Areas
	• I/O bits • Data Link bits • CPU Bus Unit bits • Special I/O Unit bits • Inner Board bits • SYSMAC BUS bits • I/O Terminal bits • DeviceNet bits • Work bits • Timer PV/Completion Flags • Index Registers • Data Registers • Task Flags • Automatically allocated variables, not held (Auxiliary Area bits/words are holding or non-holding depending on the address.)	• HR Area • DM Area • EM Area • Counter PV and Completion Flags (Auxiliary Area bits/words are holding or non-holding depending on the address.) • Automatically allocated variables, held
RUN or MONITOR to PROGRAM	Cleared (See note 1.)	Held
PROGRAM to RUN or MONITOR	Cleared (See note 1.)	Held
RUN to MONITOR or MONITOR to RUN	Held (See note 2.)	Held

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

I/O Memory Hold Bit status (A50012)	I/O Memory			Output bits allocated to Output Units		
	Mode changed between PROGRAM and RUN/ MONITOR	Operation stopped		Mode changed between PROGRAM and RUN/ MONITOR	Operation stopped	
		Fatal error other than FALS	FALS executed		Fatal error other than FALS	FALS executed
OFF	Cleared	Cleared	Held	OFF	OFF	OFF
ON	Held	Held	Held	Held	OFF	OFF

Note Refer to *5-2 I/O Memory Areas* for more details on I/O Memory.

15-3 Power OFF Operation

15-3-1 Overview

The following processing is performed if CPU Unit power is turned OFF. Power OFF processing will be performed if the power supply falls below 85% of the rated voltage while the CPU Unit is in RUN or MONITOR mode.

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

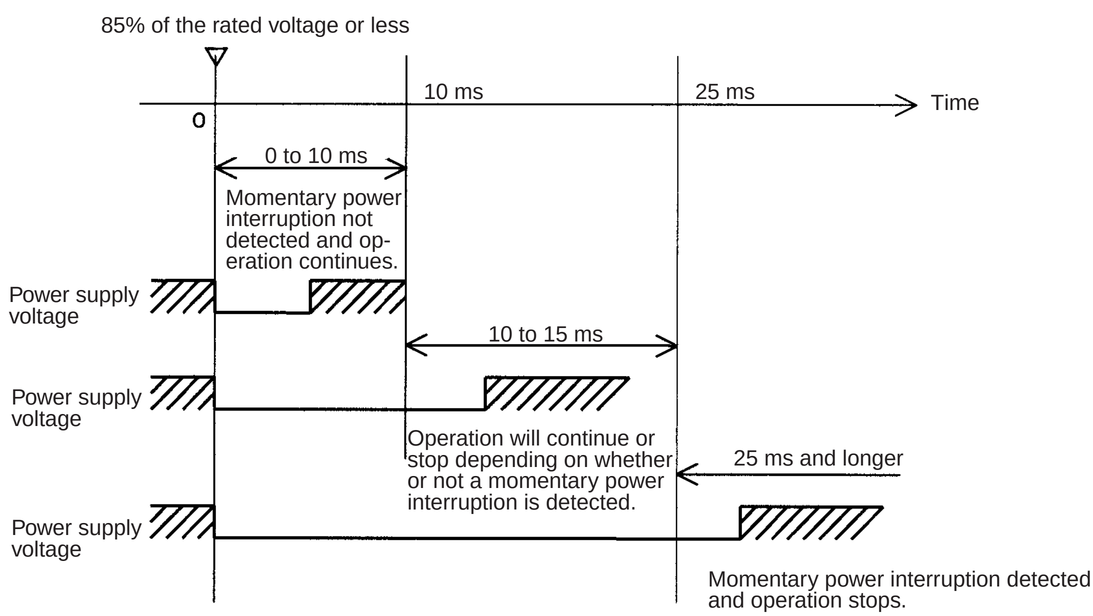
Note All output will turn OFF despite an I/O Memory Hold Bit or I/O Memory Hold Bit at power ON settings in the PLC Setup.

85% of the rated voltage:
AC power: 85 V for a 100 V AC system and 170 V for a 200 V AC system
DC power: 19.2 V DC

The following processing will be performed if power drops only momentarily (momentary power interruption).

1,2,3...
1. The system will continue to run unconditionally if the momentary power interruption lasts less than 10 ms, i.e., the time it takes the rated voltage at 85% or less to return to 85% or higher is less than 10 ms.
2. A momentary power interruption that lasts more than 10 ms but less than 25 ms is difficult to determine and a power interruption may or may not be detected.
3. The system will stop unconditionally if the momentary power interruption lasts more than 25 ms.

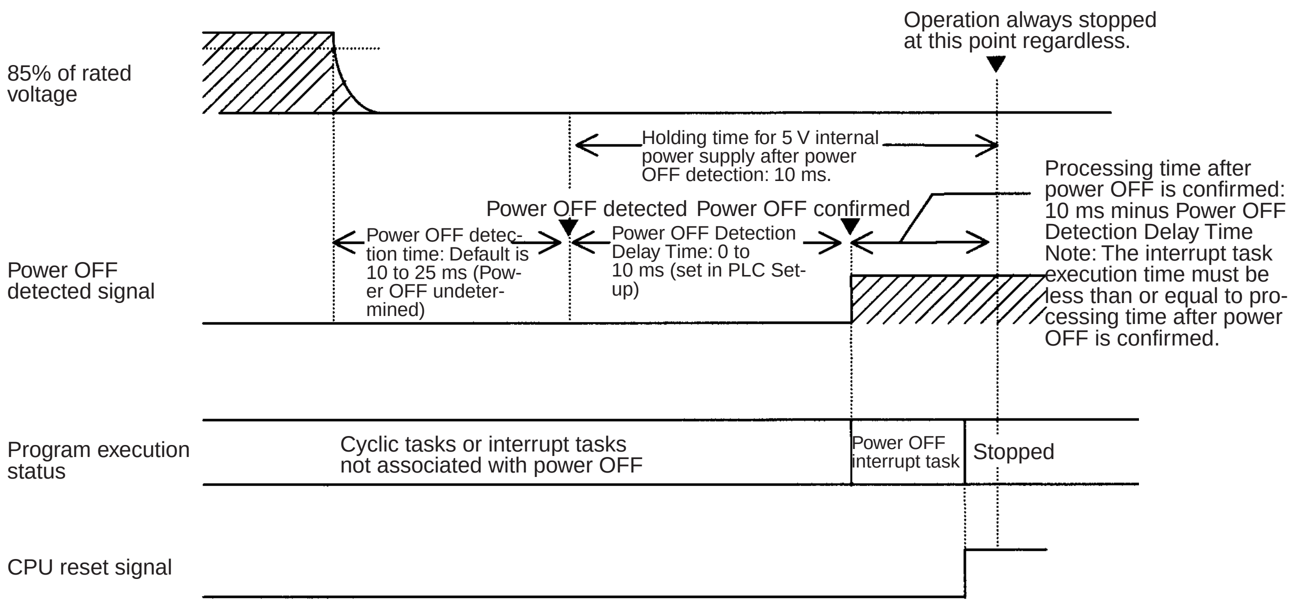
If operation stops under the conditions given in items 2 and 3 above, the timing used to stop operation (or the timing used to start execution of the Power OFF Interrupt Task) can be delayed by setting the Power OFF Detection Delay Time (0 to 10 ms) in the PLC Setup. Operation, however, will always be stopped 10 ms after detecting a momentary power interruption regardless of the setting in the PLC Setup.



Note The above timing chart shows an example when the power OFF detection time is set to 0 ms (the default value).

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

Power OFF Timing Chart



Power OFF Detection Time

The time it takes to detect power OFF after the power supply falls below 85% of the rated voltage.

Power OFF Detection Delay Time

The delay time after power OFF is detected until it is confirmed. This can be set in the PLC Setup within a range from 0 to 10 ms. (The default is 0 ms.)

Power Holding Time

The amount of time (fixed at 10 ms) that 5 V will be held internally after power shuts OFF. The time that it takes for the power OFF interrupt task to execute must not exceed 10 ms minus the Power OFF Detection Delay Time (processing time after power OFF is confirmed). The power OFF interrupt task will be ended even if it has not been completely executed the moment this time expires.

Description of Operation

1,2,3...

1. Power OFF will be detected if the 100 to 120 V AC, 200 to 240 V AC or 24 V DC power supply falls below 85% of the rated voltage for the power OFF detection time (somewhere between 10 to 25 ms).
2. If the Power OFF Detection Delay Time is set (0 to 10 ms) in the PLC Setup, then the following operations will be performed when the set time expires.
    a) If the power OFF interrupt task is disabled (default PLC Setup setting) The CPU reset signal will turn ON and the CPU will be reset immediately.
    b) If the power OFF interrupt task is enabled (in the PLC Setup), the CPU reset signal will turn ON and the CPU will be reset after the power OFF interrupt task has been executed. Make sure that the power OFF interrupt task will finish executing within 10 ms minus the Power OFF Detection Delay Time = processing time after power OFF. The 5-V internal power supply will be maintained only for 10 ms after power OFF is detected.

15-3-2 Instruction Execution for Power Interruptions

If power is interrupted and the interruption is confirmed when the CPU Unit is operating in RUN or MONITOR mode, the instruction currently being executed will be completed (see note) and the following power interruption processing will be performed.

- If the power OFF interrupt task has not been enabled, the CPU Unit will be reset immediately.
- If the power OFF interrupt task has been enabled, the task will be executed and then the CPU Unit will be reset immediately.

The power OFF interrupt task is enable and disabled in the PLC Setup.

Note The current instruction can be completed only when the time required to complete execution is less than or equal to the processing time after power interruption detection (10 ms – power interruption detection delay time). If the instruction is not completed within this time, it will be interrupted and the above processing will be performed.

Disabling Power Interruption Processing in the Program

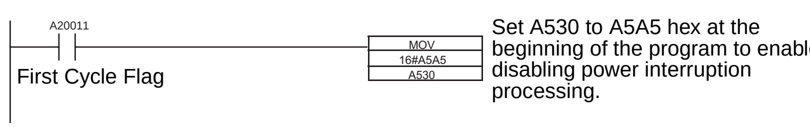
If the power OFF interrupt task is disabled, areas of the program can be protected from power interruptions so that the instructions will be executed before the CPU Unit performs power OFF processing even if the power supply is interrupted. This is achieved by using the DISABLE INTERRUPTS (DI(693)) and ENABLE INTERRUPTS (EI(694)) instructions.

The following procedure is used.

1,2,3...

1. Insert DI(693) before the program section to be protected to disable interrupts and then place EI(694) after the section to enable interrupts.
2. Set the Disable Setting for Power OFF Interrupts in A530 to A5A5 hex to enable disabling power interruption processing.

   Note A530 is normally cleared when power is turned OFF. To prevent this, the IOM Hold Bit (A50012) must be turned ON and the PLC Setup must be set to maintain the setting of the IOM Hold Bit at Startup, or the following type of instruction must be included at the beginning of the program to set A530 to A5A5 hex.



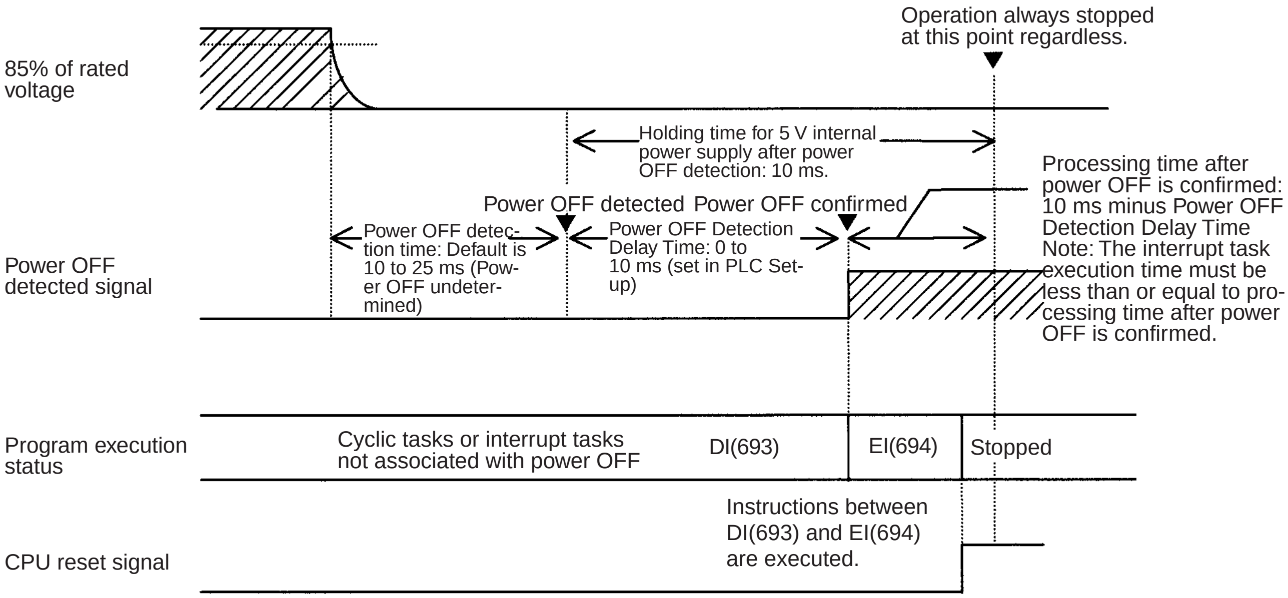
3. Disable the Power OFF Interrupt Task in the PLC Setup.

With the above procedure, all instructions between DI(693) and EI(694) (or END) will be completed (see note 1) before the Power OFF Interrupt is executed even if the power interruption occurs while executing the instructions between DI(693) and EI(694).

Note

1. The protected instructions can be completed only when the time required to complete execution is less than or equal to the processing time after power interruption detection (10 ms – power interruption detection delay time). If the instructions is not completed within this time, they will be interrupted and the above processing will be performed.
2. If the Power OFF Interrupt Task is not disabled in the PLC Setup, the Power OFF Interrupt Task will be executed, and the CPU Unit will be reset without executing the protected instructions as soon as the power interruption is detected.

3. If a power interrupt is detected while DI(693) is being executed, the CPU Unit will be reset without executing the protected instructions.



Interrupt processing is performed according to the contents of A530 and the PLC Setup as shown below.

A530		**A5A5 hex (disabling power interrupt processing)**	**Other**
Power OFF Interrupt Task (PLC Setup)	Disabled	All instructions between DI(693) and EI(694) are executed and the CPU Unit is reset.	Execution of the current instruction is completed and the CPU Unit is reset.
	Enabled	Execution of the current instruction is completed, the Power OFF Interrupt Task is executed, and the CPU Unit is reset.	

15-4 Instruction Execution Times and Number of Steps

The following table lists the execution times for all instructions that are available for NE1S-series PLCs.

The total execution time of instructions within one whole user program is the process time for program execution when calculating the cycle time (See note.).

Note User programs are allocated tasks that can be executed within cyclic tasks and interrupt tasks that satisfy interrupt conditions.

Execution times for most instructions differ depending on the conditions when the instruction is executed. The top line for each instruction in the following table shows the minimum time required to process the instruction and the necessary execution conditions, and the bottom line shows the maximum time and execution conditions required to process the instruction.

The execution time can also vary when the execution condition is OFF.

The following table also lists the length of each instruction in the *Length (steps)* column. The number of steps required in the user program area for each of the NE1S-series instructions varies from 1 to 7 steps, depending upon the instruction and the operands used with it. The number of steps in a program is not the same as the number of instructions.

Note 1. Program capacity for NE1S-series PLCs is measured in steps. Basically speaking, 1 step is equivalent to 1 word.

Most instructions are supported in differentiated form (indicated with ↑, ↓, @, and %). Specifying differentiation will increase the execution times by the following amounts.

Symbol	NE1S CPU Units
	NE1S-CPU01
↑ or ↓	+0.24 µs
@ or %	+0.24 µs

2. Use the following times as guidelines when instructions are not executed.

NE1S CPU Units
NE1S-CPU01
Approx. 0.1 µs

15-4-1 Sequence Input Instructions

Instruction	Mnemonic	Code	Length (steps)	Execution time (µs)	Conditions
				CPU01	
LOAD	LD	---	1	0.019	---
	!LD	---	2	+12.1	Increase for CS Series
LOAD NOT	LD NOT	---	1	0.019	---
	!LD NOT	---	2	+12.1	Increase for CS Series
AND	AND	---	1	0.019	---
	!AND	---	2	+12.1	Increase for CS Series
AND NOT	AND NOT	---	1	0.019	---
	!AND NOT	---	2	+12.1	Increase for CS Series
OR	OR	---	1	0.019	---
	!OR	---	2	+12.1	Increase for CS Series
OR NOT	OR NOT	---	1	0.019	---
	!OR NOT	---	2	+12.1	Increase for CS Series
AND LOAD	AND LD	---	1	0.019	---
OR LOAD	OR LD	---	1	0.019	---
NOT	NOT	520	1	0.019	---
CONDITION ON	UP	521	3	0.3	---
CONDITION OFF	DOWN	522	4	0.3	---
LOAD BIT TEST	LD TST	350	4	0.14	---
LOAD BIT TEST NOT	LD TSTN	351	4	0.14	---
AND BIT TEST NOT	AND TSTN	351	4	0.14	---
OR BIT TEST	OR TST	350	4	0.14	---
OR BIT TEST NOT	OR TSTN	351	4	0.14	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-2 Sequence Output Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
OUTPUT	OUT	---	1	0.019	---
	!OUT	---	2	+17.3	Increase for CS Series
OUTPUT NOT	OUT NOT	---	1	0.019	---
	!OUT NOT	---	2	+17.3	Increase for CS Series
KEEP	KEEP	011	1	0.06	---
DIFFERENTIATE UP	DIFU	013	2	0.24	---
DIFFERENTIATE DOWN	DIFD	014	2	0.24	---
SET	SET	---	1	0.019	---
	!SET	---	2	+17.3	Increase for CS Series
RESET	RSET	---	1	0.019	Word specified
	!RSET	---	2	+17.3	Increase for CS Series
MULTIPLE BIT SET	SETA	530	4	4.1	With 1-bit set
				21.3	With 1,000-bit set
MULTIPLE BIT RESET	RSTA	531	4	4.8	With 1-bit reset
				21.6	With 1,000-bit reset
SINGLE BIT SET	SETB	532	2	0.24	---
	!SETB		3	+17.3	---
SINGLE BIT RESET	RSTB	534	2	0.24	---
	!RSTB		3	+17.3	---
SINGLE BIT OUTPUT	OUTB	534	2	0.22	---
	!OUTB		3	+17.3	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-3 Sequence Control Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
END	END	001	1	7.8	---
NO OPERATION	NOP	000	1	0.019	---
INTERLOCK	IL	002	1	0.06	---
INTERLOCK CLEAR	ILC	003	1	0.06	---
JUMP	JMP	004	2	0.38	---
JUMP END	JME	005	2	---	---
CONDITIONAL JUMP	CJP	510	2	0.38	When JMP condition is satisfied
CONDITIONAL JUMP NOT	CJPN	511	2	0.38	When JMP condition is satisfied
MULTIPLE JUMP	JMP0	515	1	0.06	---
MULTIPLE JUMP END	JME0	516	1	0.06	---
FOR LOOP	FOR	512	2	0.52	Designating a constant
BREAK LOOP	BREAK	514	1	0.06	---
NEXT LOOP	NEXT	513	1	0.18	When loop is continued
				0.22	When loop is ended

Note
1. When a double-length operand is used, add 1 to the value shown in the length column in the following table.
2. Supported only by CPU Units Ver. 2.0 or later.

15-4-4 Timer and Counter Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
TIMER	TIMX	550	3	0.56	---
COUNTER	CNTX	546	3	0.56	---
HIGH-SPEED TIMER	TIMHX	551	3	0.88	---
ONE-MS TIMER	TMHHX	552	3	0.86	---
ACCUMULATIVE TIMER	TTIMX	555	3	16.2	---
				12.6	When resetting
				7.3	When interlocking
LONG TIMER	TIMLX	553	4	6.6	---
				5.2	When interlocking
MULTI-OUTPUT TIMER	MTIMX	554	4	6.1	---
				4.7	When resetting
REVERSIBLE COUNTER	CNTRX	548	3	17.2	---
RESET TIMER/COUNTER	CNRX	547	3	14.9	When resetting 1 word
				8.29 ms	When resetting 1,000 words

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-5 Comparison Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
Input Comparison Instructions (unsigned)	LD, AND, OR +=	300	4	0.10	---
	LD, AND, OR + <>	305			
	LD, AND, OR + <	310			
	LD, AND, OR +<=	315			
	LD, AND, OR +>	320			
	LD, AND, OR +>=	325			
Input Comparison Instructions (double, unsigned)	LD, AND, OR +=+L	301	4	0.10	---
	LD, AND, OR +<>+L	306			
	LD, AND, OR +<+L	311			
	LD, AND, OR +<=+L	316			
	LD, AND, OR +>+L	321			
	LD, AND, OR +>=+L	326			

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
Input Comparison Instructions (signed)	LD, AND, OR +=+S	302	4	0.10	---
	LD, AND, OR +<>+S	307			
	LD, AND, OR +<+S	312			
	LD, AND, OR +<=	317			
	LD, AND, OR +>+S	322			
	LD, AND, OR +>=+S	327			
Input Comparison Instructions (double, signed)	LD, AND, OR +=+SL	303	4	0.10	---
	LD, AND, OR +<>+SL	308			
	LD, AND, OR +<+SL	313			
	LD, AND, OR +<=+SL	318			
	LD, AND, OR +>+SL	323			
	LD, AND, OR +>=+SL	328			
COMPARE	CMP	020	3	0.04	---
	!CMP	020	7	+42.1	Increase for CS Series
DOUBLE COMPARE	CMPL	060	3	0.08	---
SIGNED BINARY COMPARE	CPS	114	3	0.08	---
	!CPS	114	7	+35.9	Increase for CS Series
DOUBLE SIGNED BINARY COMPARE	CPSL	115	3	0.08	---
TABLE COMPARE	TCMP	085	4	9.8	---
MULTIPLE COMPARE	MCMP	019	4	15.2	---
UNSIGNED BLOCK COMPARE	BCMP	068	4	15.9	---
AREA RANGE COMPARE	ZCP	088	3	4.4	---
DOUBLE AREA RANGE COMPARE	ZCPL	116	3	4.0	---

Note

1. When a double-length operand is used, add 1 to the value shown in the length column in the following table.
2. Supported only by CPU Units Ver. 2.0 or later.

15-4-6 Data Movement Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
MOVE	MOV	021	3	0.18	---
	!MOV	021	7	+21.38	Increase for CS Series
DOUBLE MOVE	MOVL	498	3	0.32	---
MOVE NOT	MVN	022	3	0.18	---
DOUBLE MOVE NOT	MVNL	499	3	0.32	---
MOVE BIT	MOVB	082	4	0.24	---
MOVE DIGIT	MOVD	083	4	0.24	---

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
MULTIPLE BIT TRANSFER	XFRB	062	4	6.4	Transferring 1 bit
				94.7	Transferring 255 bits
BLOCK TRANSFER	XFER	070	4	0.36	Transferring 1 word
				300.1	Transferring 1,000 words
BLOCK SET	BSET	071	4	0.26	Setting 1 word
				200.1	Setting 1,000 words
DATA EXCHANGE	XCHG	073	3	0.40	---
DOUBLE DATA EXCHANGE	XCGL	562	3	0.76	---
SINGLE WORD DISTRIBUTE	DIST	080	4	4.4	---
DATA COLLECT	COLL	081	4	4.1	---
MOVE TO REGISTER	MOVR	560	3	0.08	---
MOVE TIMER/COUNTER PV TO REGISTER	MOVRW	561	3	0.42	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-7 Data Shift Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
REVERSIBLE SHIFT REGISTER	SFTR	084	4	5.8	Shifting 1 word
				577.9	Shifting 1,000 words
ASYNCHRONOUS SHIFT REGISTER	ASFT	017	4	5.6	Shifting 1 word
				966	Shifting 1,000 words
WORD SHIFT	WSFT	016	4	5.0	Shifting 1 word
				187.1	Shifting 1,000 words
ARITHMETIC SHIFT LEFT	ASL	025	2	0.22	---
DOUBLE SHIFT LEFT	ASLL	570	2	0.40	---
ARITHMETIC SHIFT RIGHT	ASR	026	2	0.22	---
DOUBLE SHIFT RIGHT	ASRL	571	2	0.40	---
ROTATE LEFT	ROL	027	2	0.22	---
DOUBLE ROTATE LEFT	ROLL	572	2	0.40	---
ROTATE LEFT WITHOUT CARRY	RLNC	574	2	0.22	---
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL	576	2	0.40	---
ROTATE RIGHT	ROR	028	2	0.22	---
DOUBLE ROTATE RIGHT	RORL	573	2	0.40	---
ROTATE RIGHT WITHOUT CARRY	RRNC	575	2	0.22	---
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL	577	2	0.40	---
ONE DIGIT SHIFT LEFT	SLD	074	3	6.4	Shifting 1 word
				467.0	Shifting 1,000 words
ONE DIGIT SHIFT RIGHT	SRD	075	3	5.8	Shifting 1 word
				682.2	Shifting 1,000 words

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
SHIFT N-BIT DATA LEFT	NSFL	578	4	3.5	Shifting 1 bit
				33.0	Shifting 1,000 bits
SHIFT N-BIT DATA RIGHT	NSFR	579	4	3.3	Shifting 1 bit
				41.0	Shifting 1,000 bits
SHIFT N-BITS LEFT	NASL	580	3	0.22	---
DOUBLE SHIFT N-BITS LEFT	NSLL	582	3	0.40	---
SHIFT N-BITS RIGHT	NASR	581	3	0.22	---
DOUBLE SHIFT N-BITS RIGHT	NSRL	583	3	0.40	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-8 Increment/Decrement Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
INCREMENT BINARY	++	590	2	0.22	---
DOUBLE INCREMENT BINARY	++L	591	2	0.40	---
DECREMENT BINARY	– –	592	2	0.22	---
DOUBLE DECREMENT BINARY	– –L	593	2	0.40	---
INCREMENT BCD	++B	594	2	8.8	---
DOUBLE INCREMENT BCD	++BL	595	2	3.9	---
DECREMENT BCD	– –B	596	2	9.8	---
DOUBLE DECREMENT BCD	– –BL	597	2	3.4	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-9 Symbol Math Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
SIGNED BINARY ADD WITHOUT CARRY	+	400	4	0.18	---
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L	401	4	0.32	---
SIGNED BINARY ADD WITH CARRY	+C	402	4	0.18	---
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL	403	4	0.32	---
BCD ADD WITHOUT CARRY	+B	404	4	6.7	---
DOUBLE BCD ADD WITHOUT CARRY	+BL	405	4	10.5	---
BCD ADD WITH CARRY	+BC	406	4	7.4	---
DOUBLE BCD ADD WITH CARRY	+BCL	407	4	10.4	---

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
SIGNED BINARY SUBTRACT WITHOUT CARRY	–	410	4	0.18	---
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	–L	411	4	0.32	---
SIGNED BINARY SUBTRACT WITH CARRY	–C	412	4	0.18	---
DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	–CL	413	4	0.32	---
BCD SUBTRACT WITHOUT CARRY	–B	414	4	6.8	---
DOUBLE BCD SUBTRACT WITHOUT CARRY	–BL	415	4	10.3	---
BCD SUBTRACT WITH CARRY	–BC	416	4	8.4	---
DOUBLE BCD SUBTRACT WITH CARRY	–BCL	417	4	10.3	---
SIGNED BINARY MULTIPLY	*	420	4	0.38	---
DOUBLE SIGNED BINARY MULTIPLY	*L	421	4	6.0	---
UNSIGNED BINARY MULTIPLY	*U	422	4	0.38	---
DOUBLE UNSIGNED BINARY MULTIPLY	*UL	423	4	5.3	---
BCD MULTIPLY	*B	424	4	7.6	---
DOUBLE BCD MULTIPLY	*BL	425	4	16.3	---
SIGNED BINARY DIVIDE	/	430	4	0.40	---
DOUBLE SIGNED BINARY DIVIDE	/L	431	4	4.7	---
UNSIGNED BINARY DIVIDE	/U	432	4	0.40	---
DOUBLE UNSIGNED BINARY DIVIDE	/UL	433	4	5.1	---
BCD DIVIDE	/B	434	4	7.0	---
DOUBLE BCD DIVIDE	/BL	435	4	14.2	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-10 Conversion Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
BCD-TO-BINARY	BIN	023	3	0.22	---
DOUBLE BCD-TO-DOUBLE BINARY	BINL	058	3	5.4	---
BINARY-TO-BCD	BCD	024	3	0.24	---
DOUBLE BINARY-TO-DOUBLE BCD	BCDL	059	3	6.4	---
2'S COMPLEMENT	NEG	160	3	0.18	---
DOUBLE 2'S COMPLEMENT	NEGL	161	3	0.32	---

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
16-BIT TO 32-BIT SIGNED BINARY	SIGN	600	3	0.32	---
DATA DECODER	MLPX	076	4	0.32	Decoding 1 digit (4 to 16)
				0.98	Decoding 4 digits (4 to 16)
				3.30	Decoding 1 digit 8 to 256
				6.50	Decoding 2 digits (8 to 256)
DATA ENCODER	DMPX	077	4	5.7	Encoding 1 digit (16 to 4)
				5.7	Encoding 4 digits (16 to 4)
				22.1	Encoding 1 digit (256 to 8)
				22.3	Encoding 2 digits (256 to 8)
ASCII CONVERT	ASC	086	4	6.2	Converting 1 digit into ASCII
				26.5	Converting 4 digits into ASCII
ASCII TO HEX	HEX	162	4	6.6	Converting 1 digit
COLUMN TO LINE	LINE	063	4	10.9	---
LINE TO COLUMN	COLM	064	4	14.9	---
SIGNED BCD-TO-BINARY	BINS	470	4	8.1	Data format setting No. 0
				8.2	Data format setting No. 1
				6.5	Data format setting No. 2
				7.8	Data format setting No. 3
DOUBLE SIGNED BCD-TO-BINARY	BISL	472	4	8.1	Data format setting No. 0
				8.3	Data format setting No. 1
				8.6	Data format setting No. 2
				8.8	Data format setting No. 3
SIGNED BINARY-TO-BCD	BCDS	471	4	67	Data format setting No. 0
				12.6	Data format setting No. 1
				7.6	Data format setting No. 2
				7.8	Data format setting No. 3
DOUBLE SIGNED BINARY-TO-BCD	BDSL	473	4	7.4	Data format setting No. 0
				7.7	Data format setting No. 1
				8.0	Data format setting No. 2
				8.4	Data format setting No. 3
GRAY CODE CONVERSION (See note 2.)	GRY	474	4	7.86	8-bit binary

Note
1. When a double-length operand is used, add 1 to the value shown in the length column in the following table.
2. Supported only by Rev. 2.1 or later.

15-4-11 Logic Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
LOGICAL AND	ANDW	034	4	0.18	---
DOUBLE LOGICAL AND	ANDL	610	4	0.32	---
LOGICAL OR	ORW	035	4	0.22	---
DOUBLE LOGICAL OR	ORWL	611	4	0.32	---
EXCLUSIVE OR	XORW	036	4	0.22	---
DOUBLE EXCLUSIVE OR	XORL	612	4	0.32	---
EXCLUSIVE NOR	XNRW	037	4	0.22	---

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
DOUBLE EXCLUSIVE NOR	XNRL	613	4	0.32	---
COMPLEMENT	COM	029	2	0.22	---
DOUBLE COMPLEMENT	COML	614	2	0.40	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-12 Special Math Instruction

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
BIT COUNTER (See note 2.)	BCNT	067	4	0.3	Counting one word

Note
1. When a double-length operand is used, add 1 to the value shown in the length column in the following table.
2. Supported only by Rev. 2.1 or later.

15-4-13 Floating Point Math Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
FLOATING TO 16-BIT	FIX	450	3	7.3	
FLOATING TO 32-BIT	FIXL	451	3	7.1	
16-BIT TO FLOATING	FLT	452	3	5.0	
32-BIT TO FLOATING	FLTL	453	3	5.0	
FLOATING-POINT ADD	+F	454	4	9.0	
FLOATING-POINT SUBTRACT	–F	455	4	9.0	
FLOATING- POINT DIVIDE	/F	457	4	8.6	
FLOATING- POINT MULTIPLY	*F	456	4	7.8	
DEGREES TO RADIANS	RAD	458	3	11.3	
RADIANS TO DEGREES	DEG	459	3	11.5	
SINE	SIN	460	3	19.6	
COSINE	COS	461	3	20.2	
TANGENT	TAN	462	3	22.6	
ARC SINE	ASIN	463	3	10.5	
ARC COSINE	ACOS	464	3	12.0	
ARC TANGENT	ATAN	465	3	9.0	
SQUARE ROOT	SQRT	466	3	8.8	
EXPONENT	EXP	467	3	27.4	
LOGARITHM	LOG	468	3	9.9	
EXPONENTIAL POWER	PWR	840	4	81.3	

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
Floating Symbol Comparison	LD, AND, OR+=F	329		5.3	
	LD, AND, OR+<>F	330			
	LD, AND, OR+<F	331			
	LD, AND, OR+<=F	332			
	LD, AND, OR+>F	333			
	LD, AND, OR+>=F	334			
FLOATING- POINT TO ASCII	FSTR	448	4	25.3	
ASCII TO FLOATING-POINT	FVAL	449	3	12.3	
FIND MAXIMUM FLOATING-POINT	MAXF	176	4	9.6	Searching for 1 word
				830.9	Searching for 1,000 words
FIND MINIMUM FLOATING-POINT	MINF	177	4	9.6	Searching for 1 word
				831.9	Searching for 1,000 words

Note 1. When a double-length operand is used, add 1 to the value shown in the length column in the following table.

2. All floating-point math instructions are supported only by Rev. 2.1 or later.

15-4-14 Table Data Processing Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
SET STACK	SSET	630	3	17.4	Designating 5 words in stack area
				222.5	Designating 1,000 words in stack area
PUSH ONTO STACK	PUSH	632	3	6.2	---
FIRST IN FIRST OUT	FIFO	633	3	6.0	Designating 5 words in stack area
				321.7	Designating 1,000 words in stack area
LAST IN FIRST OUT	LIFO	634	3	7.1	---
DIMENSION RECORD TABLE	DIM	631	5	10.3	---
SET RECORD LOCATION	SETR	635	4	4.2	---
GET RECORD NUMBER	GETR	636	4	3.3	---
DATA SEARCH	SRCH	181	4	4.7	Searching for 1 word
				486.9	Searching for 1,000 words
SWAP BYTES	SWAP	637	3	3.8	Swapping 1 word
				732.5	Swapping 1,000 words
FIND MAXIMUM	MAX	182	4	6.4	Searching for 1 word
				491.0	Searching for 1,000 words
FIND DOUBLE-LENGTH MAXIMUM (See note 2.)	MAXL	174	4	6.3	Searching for 1 word
				500.0	Searching for 1,000 words
FIND MINIMUM	MIN	183	4	6.3	Searching for 1 word
				501.4	Searching for 1,000 words

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
FIND DOUBLE-LENGTH MINIMUM (See note 2.)	MINL	175	4	5.5	Searching for 1 word
				496.6	Searching for 1,000 words
SUM	SUM	184	4	8.4	Adding 1 word
				316.1	Adding 1,000 words
FRAME CHECKSUM	FCS	180	4	5.2	For 1-word table length
				481.7	For 1,000-word table length
STACK SIZE READ	SNUM	638	3	5.2	---
STACK DATA READ	SREAD	639	4	6.8	---
STACK DATA OVER-WRITE	SWRIT	640	4	6.3	---
STACK DATA INSERT	SINS	641	4	6.7	---
				324.4	For 1,000-word table
STACK DATA DELETE	SDEL	642	4	7.6	---
				335.5	For 1,000-word table

Note 1. When a double-length operand is used, add 1 to the value shown in the length column in the following table.
2. Supported only by Rev. 2.1 or later.

15-4-15 Data Control Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
PID CONTROL (See note 2.)	PID	190	4	83.9	When executing the first time
				59.7	When sampling
				30.7	When not sampling
PID CONTROL WITH AUTOTUNING (See note 2.)	PIDAT	191	4	87.4	During PID control, when executing the first time
				85.5	During PID control, when sampling
				32.7	During PID control, when not sampling

Note 1. When a double-length operand is used, add 1 to the value shown in the length column in the following table.
2. Supported only by Rev. 2.1 or later.

15-4-16 Interrupt Control Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
SET INTERRUPT MASK	MSKS	690	3	9.3	---
READ INTERRUPT MASK	MSKR	692	3	4.1	---
CLEAR INTERRUPT	CLI	691	3	8.3	---
DISABLE INTERRUPTS	DI	693	1	9.3	---
ENABLE INTERRUPTS	EI	694	1	9.1	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-17 Basic I/O Unit Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
I/O REFRESH	IORF	097	3	15.3	1-word refresh (IN) for CS-series Basic I/O Units
				21.3	1-word refresh (OUT) for CS-series Basic I/O Units
				319.9	60-word refresh (IN) for CS-series Basic I/O Units
				358.00	60-word refresh (OUT) for CS-series Basic I/O Units
7-SEGMENT DECODER	SDEC	078	4	15.5	---
INTELLIGENT I/O READ	IORD	222	4	Read/write times depend on the Special I/O Unit for which the instruction is being executed.	---
INTELLIGENT I/O WRITE	IOWR	223	4		---
CPU BUS I/O REFRESH	DLNK	226	4	260.5	Allocated 1 word

Note
1. When a double-length operand is used, add 1 to the value shown in the length column in the following table.
2. The ON execution time for IORD and IOWR instructions depends on the Special I/O Unit being used to read them.

15-4-18 Serial Communications Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
PROTOCOL MACRO	PMCR	260	5	37.3	Sending 0 words, receiving 0 words
				50.2	Sending 249 words, receiving 249 words

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-19 Network Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
SEND CIP COMMAND	CSND	489	4	40.3	
SEND MAIL	MLSND	795	5	16.3	Text string of 0 characters, one destination mail address with 0 characters
				16.9	Text string of up to 128 characters, two destination mail addresses of up to 64 characters

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-20 File Memory Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
READ DATA FILE	FREAD	700	5	66.9	2-character directory + file name in binary
				81.6	73-character directory + file name in binary
WRITE DATA FILE	FWRIT	701	5	66.7	2-character directory + file name in binary
				109.0	73-character directory + file name in binary

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-21 Display Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
DISPLAY MESSAGE	MSG	046	3	3.2	Displaying message
				2.9	Deleting displayed message

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-22 Clock Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
CALENDAR ADD	CADD	730	4	24.1	---
CALENDAR SUBTRACT	CSUB	731	4	25.2	---
HOURS TO SECONDS	SEC	065	3	8.3	---
SECONDS TO HOURS	HMS	066	3	11.5	---
CLOCK ADJUSTMENT	DATE	735	2	12.2	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-23 Debugging Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs)	Conditions
				CPU01	
Trace Memory Sampling	TRSM	045	1	35.9	Sampling 1 bit and 0 words

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-24 Failure Diagnosis Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
FAILURE ALARM	FAL	006	3	25.8	Recording errors
				42.5	Deleting errors (in order of priority)
				106.5	Deleting errors (all errors)
				41.1	Deleting errors (individually)
SEVERE FAILURE ALARM	FALS	007	3	---	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-25 Other Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
SET CARRY	STC	040	1	0.06	---
CLEAR CARRY	CLC	041	1	0.06	---
EXTEND MAXIMUM CYCLE TIME	WDT	094	2	380.6	---
SAVE Condition Flags	CCS	282	1	12.8	---
LOAD Condition Flags	CCL	283	1	7.8	---

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-26 Text String Processing Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (µs) CPU01	Conditions
MOV STRING	MOV$	664	3	6.3	Transferring 1 character
CONCATENATE STRING	+$	656	4	23.5	1 character + 1 character
GET STRING LEFT	LEFT$	652	4	20.3	Retrieving 1 character from 2 characters
GET STRING RIGHT	RGHT$	653	4	14.0	Retrieving 1 character from 2 characters
GET STRING MIDDLE	MID$	654	5	16.0	Retrieving 1 character from 3 characters
FIND IN STRING	FIND$	660	4	12.2	Searching for 1 character from 2 characters
STRING LENGTH	LEN$	650	3	14.5	Detecting 1 character
REPLACE IN STRING	RPLC$	661	6	41.1	Replacing the first of 2 characters with 1 character
DELETE STRING	DEL$	658	5	19.5	Deleting the leading character of 2 characters
EXCHANGE STRING	XCHG$	665	3	16.9	Exchanging 1 character with 1 character
CLEAR STRING	CLR$	666	2	6.8	Clearing 1 character
INSERT INTO STRING	INS$	657	5	512.5	Inserting 1 character after the first of 2 characters

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (μs)	Conditions
				CPU01	
String Comparison	LD, AND, OR+=$	670	4	9.2	Comparing 1 character with 1 character
	LD, AND, OR+<>$	671			
	LD, AND, OR+<$	672		358.1	Comparing 2,047 characters with 2,047 characters
	LD, AND, OR+>$	674			
	LD, AND, OR+>=$	675			

Note When a double-length operand is used, add 1 to the value shown in the length column in the following table.

15-4-27 Task Control Instructions

Instruction	Mnemonic	Code	Length (steps) (See note.)	Execution time (μs)	Conditions
				CPU01	
TASK ON	TKON	820	2	4.1	---
TASK OFF	TKOF	821	2	5.2	---

SECTION 16
Troubleshooting

This section provides information on hardware and software errors that occur during PLC operation.

16-1 Error Log

Each time that an error occurs, the CPU Unit stores error information in the Error Log Area. The error information includes the error code (stored in A400), error contents, and time that the error occurred. Up to 20 records can be stored in the Error Log.

Errors Generated by FAL(006)/FALS(007)

In addition to system-generated errors, the PLC records user-defined FAL(006) and FALS(007) errors, making it easier to track the operating status of the system.

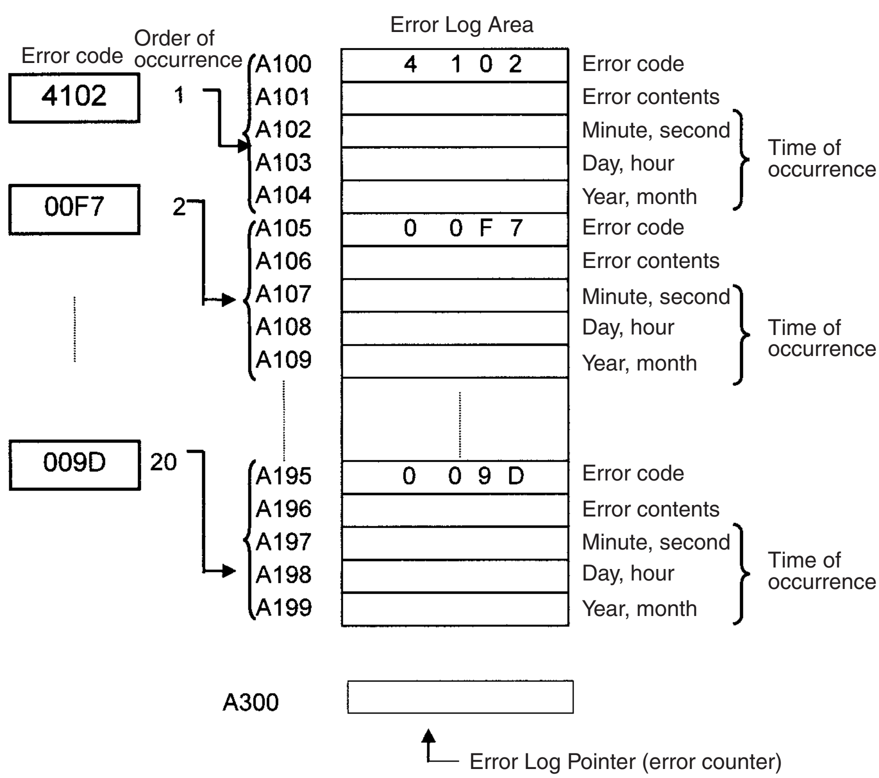
A user-defined error is generated when FAL(006) or FALS(007) is executed in the program. The execution conditions of these instructions constitute the user-defined error conditions. FAL(006) generates a non-fatal error and FALS(007) generates a fatal error that stops program execution.

The following table shows the error codes for FAL(006) and FALS(007).

Instruction	FAL numbers	Error codes
FAL(006)	#0001 to #01FF (1 to 511 decimal)	4101 to 42FF
FALS(007)	#0001 to #01FF (1 to 511 decimal)	C101 to C2FF

Error Log Structure

When more than 20 errors occur, the oldest error data (in A100 to A104) is deleted, the newest record is stored in A195 to A199, and the other errors are shifted by one.



The number of records is stored in binary in the Error Log Pointer (A300). The data in the Error Log can be cleared from a Programming Device.

Note The Error Log Pointer can be reset by turning ON the Error Log Pointer Reset Bit (A50014), but this operation does not clear the data in the Error Log (A100 to A199) itself.

16-2 Error Processing

16-2-1 Error Categories

Errors in NE1S-series PLCs can be broadly divided into the following three categories.

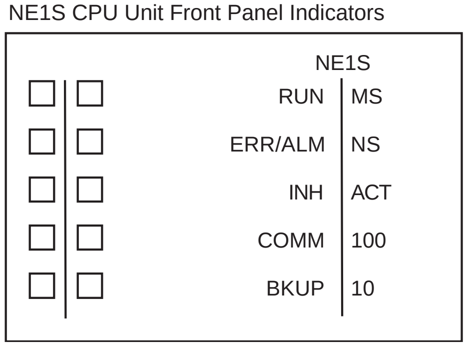
Category	Result	Indicators			Comments
		RUN	**ERR/ALM**	**MS**	
CPU Standby	The CPU Unit will not start operation in RUN or MONITOR mode.	OFF	OFF	ON (Green)	---
Non-fatal Errors (including FAL(006))	The CPU Unit will continue operating in RUN or MONITOR mode.	ON (Green)	Flashing (Red)	Flashing (Red)	Other indicators will also operate when a communications error has occurred or the Output OFF Bit is ON.
Fatal Errors (including FALS(007))	The CPU Unit will stop operating in RUN or MONITOR mode.	OFF	ON (Red)	Flashing (Red)	The indicators will all be OFF when there is a power interruption.

16-2-2 Error Information

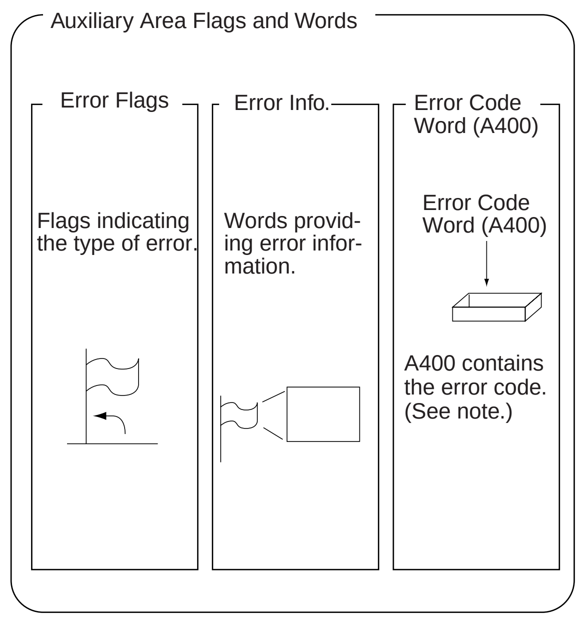
There are basically four sources of information on errors that have occurred:

1,2,3...
1. The CPU Unit's indicators
2. The Auxiliary Area Error Flags
3. The Auxiliary Area Error Information Words
4. The Auxiliary Area Error Code Word

NE1S CPU Unit Front Panel Indicators



The meaning of the indicators is described in the following table.



Indicator	Color	Status	Meaning
RUN	Green	ON	CPU Unit is executing the program in MONITOR or RUN mode.
		Flashing	DIP switch setting error
		OFF	PLC has stopped operating while in PROGRAM mode, or has stopped operating due to a fatal error.

Indicator	Color	Status	Meaning
ERR/ALM	Red	ON	A fatal error has occurred (including FALS instruction execution), or a hardware error (watchdog timer error) has occurred. The CPU Unit stops operating, and the outputs from all Output Units will be turned OFF.
		Flashing	A non-fatal error has occurred (including FAL instruction execution) The CPU Unit will continue operating.
		OFF	CPU Unit is operating normally.
INH	Yellow	ON	Output OFF Bit (A50015) is ON. The outputs from all Output Units will be OFF.
		OFF	Output OFF Bit (A50015) is OFF.
COMM	Yellow	Flashing	CPU Unit is communicating (sending or receiving) via the RS-232C port.
		OFF	CPU Unit is not communicating via the RS-232C port.
BKUP	Yellow	ON	User program, program information, variable information, and parameter area data is being backed up to built-in flash memory in the CPU Unit or being restored from flash memory (as specified when turning ON the PLC power supply). **Note** Do not turn OFF the power supply to the PLC while this indicator is lit.
		OFF	Data is not being written to flash memory.
MS (CPU Unit error indicator)	Green	ON	CPU Unit is operating normally.
	Red	ON	A serious error has occurred in the CPU Unit. (The ERR/ALM indicator will light if the error is a WDT error.)
		Flashing	A minor error has occurred in the CPU Unit. (The ERR/ALM indicator will flash.)
NS (Network status indicator)	Green	ON	A CIP connection established.
		Flashing	No CIP connection established.
	Red	ON	IP address duplication
		Flashing	CIP I/O connection error (not supported)
		OFF	Illegal IP address
ACT	Yellow	ON	Ethernet communications are in progress.
		OFF	Ethernet communications are not in progress.
100	Orange	ON	100Base-TX Ethernet link established (100 Mbps).
		OFF	No 100Base-TX Ethernet link established.
10	Orange	ON	10Base-TX Ethernet link established (10 Mbps).
		OFF	No 10Base-TX Ethernet link established.
MCPWR (Memory Card power indicator)	Green	Flashing	Power is being supplied to the Memory Card.
		OFF	Power is not being supplied to the Memory Card.
BUSY	Yellow	Flashing	Memory Card is being accessed.
		OFF	Memory Card is not being accessed.

Note When two or more errors occur at the same time, the highest (most serious) error code will be stored in A400.

Indicator Status and Error Conditions

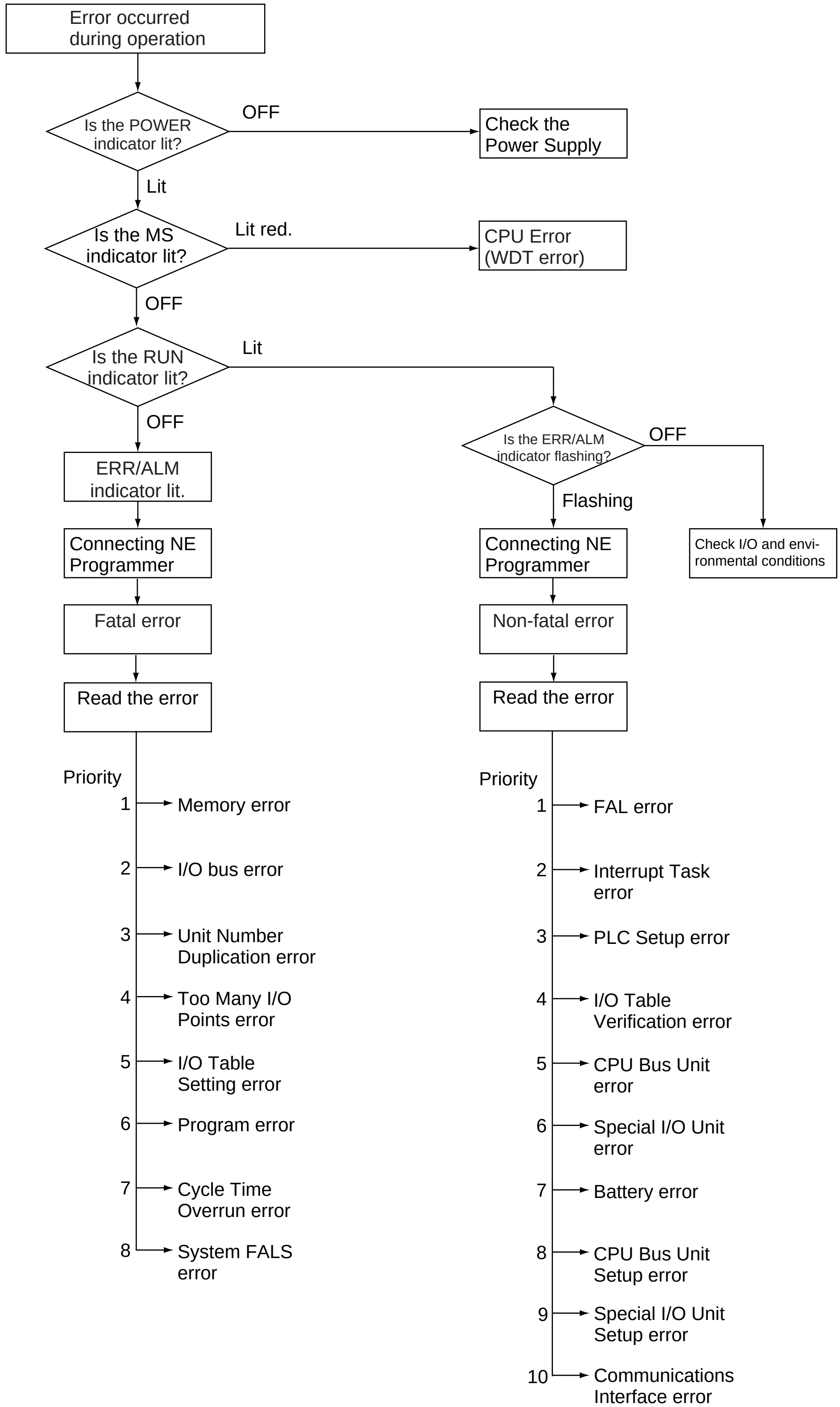
The following table shows the status of the CPU Unit's indicators for errors that have occurred in RUN or MONITOR mode.

Indicator*	CPU error	CPU reset	CPU standby	Fatal error	Non-fatal error	Special Communications error		Output OFF Bit ON
						USB	RS-232C	
RUN	OFF	OFF	OFF	OFF	ON	ON	ON	---
ERR/ALM	ON	OFF	OFF	ON	Flashing	---	---	---
INH	OFF	OFF	---	---	---	---	---	ON
COMM	---	---	---	---	---	---	OFF	---
NS	OFF	OFF	---	---	---	---	---	---
MS	ON (Green)	OFF	ON (Red)	Flashing (Red)	Flashing (Red)	---	---	---

16-2-3 Error Codes and Error Flags

Classification	Error code	Error name	Page
Fatal system errors	80F1	Memory error	460
	80C0 to 80C7, 80CF	I/O bus error	461
	80CB	I/O bus error B	461
	80E9	Duplicated number error	461
	80E1	Too many I/O points	462
	80E0	I/O setting error	462
	80F0	Program error	462
	809F	Cycle time too long	463
	80EA	Expansion Rack number duplicated	462
Non-fatal system errors	008B	Interrupt task error	464
	009A	Basic I/O error	464
	009B	PLC Setup setting error	464
	00B0	Communications interface error	466
	00E7	I/O verification error	464
	0200 to 020F	NE1S-series CPU Bus Unit error	465
	0300 to 035F	Special I/O Unit error	465
	00F7	Battery error	465
	0400 to 040F	NE1S-series CPU Bus Unit setting error	465
	0500 to 055F	Special I/O Unit setting error	465
User-defined fatal errors	4101 to 42FF	FALS(007) error (C101 to C2FF are stored for FALS numbers 001 to 511)	464
User-defined non-fatal errors	C101 to C2FF	FAL(006) error (4101 to 42FF are stored for FAL numbers 001 to 511)	463

16-2-4 Error Processing Flowchart



16-2-5 Error Messages

The following tables show error messages for errors which can occur in NE1S-series PLCs and indicate the likely cause of the errors.

⚠ **Caution** Always check the safety of the controlled system before turning the power supply OFF or ON.

CPU Errors

A CPU error has occurred if the indicators have the following conditions in RUN or MONITOR mode. A Programming Device, such as a Programming Console, cannot be connected when the CPU Unit is in this state.

<table>
<tr><th>Power Supply Unit Indicator</th><th colspan="9">CPU Unit Indicators</th></tr>
<tr><th>POWER</th><th>RUN</th><th>ERR/ALM</th><th>INH</th><th>COMM</th><th>MS</th><th>NS</th><th>ACT</th><th>10</th><th>100</th></tr>
<tr><td>ON</td><td>---</td><td>ON</td><td>---</td><td>---</td><td>ON (Red)</td><td colspan="4">---</td></tr>
</table>

Status	Error	Error flags in Auxiliary Area	Error code (in A400)	Flags and word data	Probable cause	Possible remedy
Stopped	CPU error	None	None	None	Watchdog timer has exceeded maximum setting.	Turn the power OFF and restart. The Unit may be faulty. Contact your OMRON representative if the problem persists.

CPU Reset

The following indicators status indicates an Expansion Rack power interruption, not a CPU error. A Programming Device, such as a Programming Console, cannot be connected when the CPU Unit is in this state.

<table>
<tr><th>Power Supply Unit Indicator</th><th colspan="9">CPU Unit Indicators</th></tr>
<tr><th>POWER</th><th>RUN</th><th>ERR/ALM</th><th>INH</th><th>COMM</th><th>MS</th><th>NS</th><th>ACT</th><th>10</th><th>100</th></tr>
<tr><td>ON</td><td>OFF</td><td>OFF</td><td>OFF</td><td>---</td><td colspan="3">OFF</td><td colspan="2">---</td></tr>
</table>

<table>
<tr><th>Status</th><th>Error</th><th>Error flags in Auxiliary Area</th><th>Error code (in A400)</th><th>Flags</th><th>Probable cause</th><th>Possible remedy</th></tr>
<tr><td rowspan="2">Stopped</td><td rowspan="2">CPU reset</td><td rowspan="2">None</td><td rowspan="2">None</td><td rowspan="2">None</td><td>Power is not being supplied to an Expansion Rack.</td><td>Supply power to the Expansion Rack.</td></tr>
<tr><td>An I/O Connecting Cable is not connected properly.
For example, the IN and OUT connections between the CPU Rack and an Expansion Rack are reversed.</td><td>Turn OFF the power supply, check the connections all I/O Connecting Cables, and turn the power supply back ON.</td></tr>
</table>

Note When power supply is interrupted to an Expansion Rack, the CPU Unit will stop program execution and the same operations as are performed when the power supply to the CPU Unit is interrupted will be performed. For example, if the power OFF interrupt task is enabled, it will be executed. If power is then restored to the Expansion Rack, the CPU Unit will perform startup processing, i.e., the same operational status as existed before the power interrupt will not necessarily be continued.

CPU Standby Errors

A CPU standby error has occurred if the indicators have the following conditions in RUN or MONITOR mode.

Power Supply Unit Indicator	CPU Unit Indicators								
POWER	**RUN**	**ERR/ALM**	**INH**	**COMM**	**MS**	**NS**	**ACT**	**100**	**10**
ON	OFF	OFF	---	---	---				

Status	Error	Error flags in Auxiliary Area	Error code (in A400)	Flags	Probable cause	Possible remedy
Stopped	CPU standby error	None	None	None	Recognition of one or more of the following Unit has not been completed: CPU Bus Units, Special I/O Units, or Interrupt Input Units.	Turn OFF the power supply, check the mounting and settings of all the Unit, and turn the power supply back ON.

Fatal Errors

A fatal error has occurred if the indicators have the following conditions in RUN or MONITOR mode.

Power Supply Unit Indicator	CPU Unit Indicators								
POWER	**RUN**	**ERR/ALM**	**INH**	**COMM**	**MS**	**NS**	**ACT**	**100**	**10**
ON	OFF	ON	---	---	---				

Connect the NE Programmer to display the error message (in the PLC Error Log Window on the NE Programmer). The cause of the error can be determined from the error message and related Auxiliary Area flags and words.

Errors are listed in order of importance. When two or more errors occur at the same time, the more serious error's error code will be recorded in A400.

If the IOM Hold Bit hasn't been turned ON to protect I/O memory, all non-retained areas of I/O memory will be cleared when a fatal error other than FALS(007) occurs. If the IOM Hold Bit is ON, the contents of I/O memory will be retained but all outputs will be turned OFF.

If the IOM Hold Bit hasn't been turned ON to protect I/O memory, all non-retained areas of I/O memory will be cleared when a fatal error other than FALS(007) occurs. When the IOM Hold Bit is ON, the contents of I/O memory will be retained but all outputs will be turned OFF.

Error	Error code (in A400)	Flag and word data	Probable cause	Possible remedy
Memory error	80F1	A40115: Memory Error Flag A403: Memory Error Location	An error has occurred in memory. A bit in A403 will turn ON to show the location of the error as listed below.	See below.
			A40300 ON: A checksum error has occurred in the user program memory. An illegal instruction was detected.	Check the program and correct the error.
			A40304 ON: A checksum error has occurred in the PLC Setup.	Clear the entire PLC Setup to 0000 and reenter the settings.
			A40305 ON: A checksum error has occurred in the registered I/O table.	Initialize the registered I/O table and generate a new I/O table.

Error	Error code (in A400)	Flag and word data	Probable cause	Possible remedy
Memory error	80F1	A40115: Memory Error Flag A403: Memory Error Location	A40307 ON: A checksum error has occurred in the routing tables.	Initialize the routing tables and reenter the tables.
			A40308 ON: A checksum error has occurred in the NE1S-series CPU Bus Unit setup.	Initialize the NE1S-series CPU Bus Unit setup and reenter the settings.
			A40309 ON: An error occurred during automatic transfer from the Memory Card at startup.	Make sure that the Memory Card is installed properly and that the correct file is on the Card.
			A40310 ON: Flash memory has failed.	Retransfer the user program and parameter data.
			A40311 ON: An error occurred in the variable information.	Retransfer the user program.
			A40312 ON: An error occurred in the Ethernet settings.	Retransfer the Ethernet settings.
I/O Bus error	80C0 to 80C7 or 80CF	A40114: I/O Bus Error Flag A404: I/O Bus Error Slot and Rack Numbers	Error has occurred in the bus line between the CPU and I/O Units. A40400 to A40407 contain the error slot number (00 to 09) in binary. 0F indicates that the slot cannot be determined. A40408 to A40415 contain the error rack number (00 to 07) in binary. 0F indicates that the rack cannot be determined.	Try turning the power OFF and ON again. If the error isn't corrected, turn the power OFF and check cable connections between the I/O Units and Racks. Check for damage to the cable or Units. Correct the cause of the error and then turn the Rack's power supply OFF and then ON again
	80CF	A40114: I/O Bus Error Flag A404: 0F0F	A Backplane that is not supported has been used.	Replace the Backplane with a CJ1W-BC□□2 Backplane.
Unit/Rack Number Duplication error	80E9	A40113: Duplication Error Flag A410: CPU Bus Unit Duplicate Number Flags	The same number has been allocated to more than one NE1S-series CPU Bus Unit. Bits A41000 to A41015 correspond to unit numbers 0 to F.	Check the unit numbers, eliminate the duplications, and turn the Rack's power supply OFF and then ON again.
		A40113: Duplication Error Flag A411 to A416: Special I/O Unit Duplicate Number Flags	The same number has been allocated to more than one Special I/O Unit. Bits A41100 to A41615 correspond to unit numbers 0 to 95.	Check the unit numbers, eliminate the duplications, and turn the Rack's power supply OFF and then ON again.

Error	Error code (in A400)	Flag and word data	Probable cause	Possible remedy
Unit/Rack Number Duplication error	80EA	A409: Expansion Rack Duplicate Rack Number	The same I/O word has been allocated to more than one Basic I/O Unit.	Check allocations to Units on the rack number whose bit in ON in A40900 to A40907. Correct the allocations so that no words are allocated more than once, including to Units on other Racks, and turn the Rack's power supply OFF and then ON again.
			An Expansion I/O Rack's starting word address exceeds CIO 0901. The corresponding bit in A40900 to A40907 (Racks 0 to 7) will be turned ON.	Check the first word setting for the Rack indicated in A40900 to A40907 and change the setting to a valid word address below CIO 0901 with a Programming Device.
Too Many I/O Points error	80E1	A40111: Too Many I/O Points Flag A407: Too Many I/O Points, Details	The probable causes are listed below. The 3-digit binary value (000 to 101) in A40713 to A40715 indicates the cause of the error. The value of these 3 bits is also output to A40700 to A40712. 1) The total number of I/O points set in the I/O Table exceeds the maximum allowed for the CPU Unit (bits: 000). 2) There are more than 32 interrupt inputs (bits: 001). 3) The unit number of an I/O Interface is duplicated (bits: 011).	Correct the problem indicated by the content of A407 and turn the power OFF and ON again.
I/O Table Setting error	80E0	A40110: I/O Setting Error Flag	Input and output word allocations do no agree with input/output words required by Units actually mounted.	Check the I/O table with I/O Table Verification operation. When the system has been corrected, register the I/O table again.
Program error	80F0	A40109: Program Error Flag A294 to A299: Program error information	The program is incorrect. See the following rows of this table for details. The address at which the program stopped will be output to A298 and A299.	Check A295 to determine the type of error that occurred and check A298/A299 to find the program address where the error occurred. Correct the program and then clear the error.
			A29511: No END error	Be sure that there is an END(001) instruction at the end of the task specified in A294 (program stop task number).
			A29512: Task error A task error has occurred. The following conditions will generate a task error. 1) There isn't an executable cyclic task. 2) There isn't a program allocated to the task. Check A294 for the number of the task missing a program. 3) The task specified in a TKON(820), TKOF(821), or MSKS(690) instruction doesn't exist.	Check the startup cyclic task attributes. Check the execution status of each task as controlled by TKON(820) and TKOF(821). Make sure that all of the task numbers specified in TKON(820), TKOF(821), and MSKS(690) instructions have corresponding tasks. Use MSKS(690) to mask any I/O or scheduled interrupt tasks that are not being used and that do not have programs set for them.

Error	Error code (in A400)	Flag and word data	Probable cause	Possible remedy
Program error	80F0	A40109: Program Error Flag A294 to A299: Program error information	A29510: Illegal access error An illegal access error has occurred and the PLC Setup has been set to stop operation for an instruction error. The following are illegal access errors: 1. Reading/writing a parameter area. 2. Writing memory that is not installed. 3. Writing to a read-only area. 4. Indirect DM/EM address that is not in BCD when BCD mode is specified.	Find the program address where the error occurred (A298/A299) and correct the instruction.
			A29509: Indirect DM/EM BCD error An indirect DM/EM BCD error has occurred and the PLC Setup has been set to stop operation for an instruction error.	Find the program address where the error occurred (A298/A299) and correct the indirect addressing or change to binary mode.
Program error (contd.)	80F0	A40109: Program Error Flag A294 to A299: Program Error Information	A29508: Instruction error An instruction processing error has occurred and the PLC Setup has been set to stop operation for an instruction error.	Find the program address where the error occurred (A298/A299) and correct the instruction. Alternatively, set the PLC Setup to continue operation for an instruction error.
			A29514: Illegal instruction error The program contains an instruction that cannot be executed.	Retransfer the program to the CPU Unit.
			A29515: UM overflow error The last address in UM (user program memory) has been exceeded.	Use a Programming Device to transfer the program again.
Cycle Time Overrun error	809F	A40108: Cycle Time Too Long Flag	The cycle time has exceeded the maximum cycle time (watch cycle time) set in the PLC Setup.	Change the program to reduce the cycle time or change the maximum cycle time setting. Check the Maximum Interrupt Task Processing Time in A440 and see if the Cycle Time Watch Time can be changed. The cycle time can be reduced by dividing unused parts of the program into tasks, jumping unused instructions in tasks, and disabling cyclic refreshing of Special I/O Units that don't require frequent refreshing.
System FALS error	C101 to C2FF	A40106: FALS Error Flag	FALS(007) has been executed in the program. The error code in A400 will indicate the FAL number. The leftmost digit of the code will be C and the rightmost 3 digits of the code will be from 100 to 2FF hex and will correspond to FAL numbers 001 to 511.	Correct according to cause indicated by the FAL number (set by user).

Non-fatal Errors

A non-fatal error has occurred if the indicators have the following conditions in RUN or MONITOR mode.

Power Supply Unit Indicator	CPU Unit Indicators								
POWER	RUN	ERR/ALM	INH	COMM	MS	NS	ACT	100	10
ON	ON	Flashing	---	---	Flashing (Red)	---			

Connect the NE Programmer to display the error message (in the PLC Error Log Window on the NE Programmer). The cause of the error can be determined from the error message and related Auxiliary Area flags and words.

Errors are listed in order of importance. When two or more errors occur at the same time, the more serious error's error code will be recorded in A400.

Error	Error code (in A400)	Flag and word data	Probable cause	Possible remedy
System FAL error	4101 to 42FF	A40215: FAL Error Flag A360 to A391: Executed FAL Number Flags	FAL(006) has been executed in program. Executed FAL Number Flags A36001 to A39115 correspond to FAL numbers 001 to 511. The error code in A400 will indicate the FAL number. The leftmost digit of the code will be 4 and the rightmost 3 digits of the code will be from 100 to 2FF hex and will correspond to FAL numbers 001 to 511.	Correct according to cause indicated by FAL number (set by user).
Interrupt Task error	008B	A40213: Interrupt Task Error Flag A426: Interrupt Task Error, Task Number	PLC Setup Set to Detect Interrupt Task Errors: Attempted to refresh a Special I/O Unit's I/O from an interrupt task with IORF(097) while the Unit's I/O was being refreshed by cyclic I/O refreshing (duplicate refreshing).	Check the program. Either disable detection of interrupt task errors in the PLC Setup (address 128, bit 14) or correct the problem in the program.
Basic I/O error	009A	A40212: Basic I/O Unit Error Flag A408: Basic I/O Unit Error, Slot Number	An error has occurred in a Basic I/O Unit. A408 contains the errant rack/slot number.	Check the errant Unit for blown fuse, etc.
PLC Setup error	009B	A40210: PLC Setup Error Flag A406: PLC Setup Error Location	There is a setting error in the PLC Setup. The location (binary offset) of the error is written to A406.	Change the indicated setting to a valid setting.
I/O Table Verification error	00E7	A40209: I/O Verification Error Flag	A Unit has been added or removed, so the registered I/O tables don't agree with the actual Units in the PLC. The I/O Verification Error Flag goes OFF when the situation is corrected.	Execute the I/O Table Verify operation to find the problem location. Create new I/O tables or replace the Unit to match the registered I/O tables.

Error	Error code (in A400)	Flag and word data	Probable cause	Possible remedy
NE1S-series CPU Bus Unit error	0200 to 020F	A40207: NE1S-series CPU Bus Unit Error Flag A417: NE1S-series CPU Bus Unit Error, Unit Number Flags	An error occurred in a data exchange between the CPU Unit and a NE1S-series CPU Bus Unit. The corresponding flag in A417 is turned ON to indicate the problem Unit. Bits A41700 to A41715 correspond to unit numbers 0 to F.	Check the Unit indicated in A417. Refer to the Unit's operation manual to find and correct the cause of the error. Restart the Unit by toggling its Restart Bit or turn the power OFF and ON again. Replace the Unit if it won't restart.
Special I/O Unit error	0300 to 035F, or 03FF	A40206: Special I/O Unit Error Flag A418 to A423: Special I/O Unit Error, Unit Number Flags	An error occurred in a data exchange between the CPU Unit and a Special I/O Unit. The corresponding flag in A418 to A423 is turned ON to indicate the problem Unit. Bits A41800 to A42315 correspond to unit numbers 0 to 95.	Check the Unit indicated in A418 to A423. Refer to the Unit's operation manual to find and correct the cause of the error. Restart the Unit by toggling its Restart Bit or turn the power OFF and ON again. Replace the Unit if it won't restart.
Battery error	00F7	A40204: Battery Error Flag	This error occurs when the PLC Setup has been set to detect battery errors and the CPU Unit's backup battery is missing or its voltage has dropped.	Check battery and replace if necessary. Change the PLC Setup setting if battery-free operation is being used.
NE1S-series CPU Bus Unit Setup error	0400 to 040F	A40203: NE1S-series CPU Bus Unit Setting Error Flag A427: NE1S-series CPU Bus Unit Setting Error, Unit Number Flags	An installed NE1S-series CPU Bus Unit does not match the NE1S-series CPU Bus Unit registered in the I/O table. The corresponding flag in A427 will be ON. Bits 00 to 15 correspond to unit numbers 0 to F.	Change the registered I/O tables.
Special I/O Unit Setup error	0500 to 055F	A40202: Special I/O Unit Setting Error Flag A428 to A433: Special I/O Unit Setting Error, Unit Number Flags	An installed Special I/O Unit does not match the Special I/O Unit registered in the I/O table. The corresponding flag in A428 to A433 will be ON. Bits A42800 to A43315 correspond to unit numbers 0 to 95.	

Error	Error code (in A400)	Flag and word data	Probable cause	Possible remedy
Communications interface error	00B0	A40201	An error occurred in communications for the built-in Ethernet port. The cause of the error is indicated by the following bits of A438.	See below.
			A43800 = ON: IP address setting is incorrect.	Retransfer the Ethernet settings and then cycle the power supply. **Note** For these errors, Controller servicing can be restarted from the NE Programmer instead of cycling the power supply.
			A43802 = ON: Unit IP address setting is incorrect.	
			A43803 = ON: FTP server settings are incorrect.	
			A43805 = ON: SMTP settings are incorrect. (See note.)	
			A43806 = ON: SNTP settings are incorrect.(See note.)	
			A43808 = ON: The IP routing settings are incorrect.	
			A43812 = ON: The same IP address is set twice.	Correct the IP address settings and then cycle the power supply.
			A43813 = ON: A rotary switch for the IP address was changed during operation.	

Other Errors

RS-232C or USB Port Communications Error

A communications error has occurred in communications with the device connected to the RS-232C or USB port if the indicators have the following conditions.

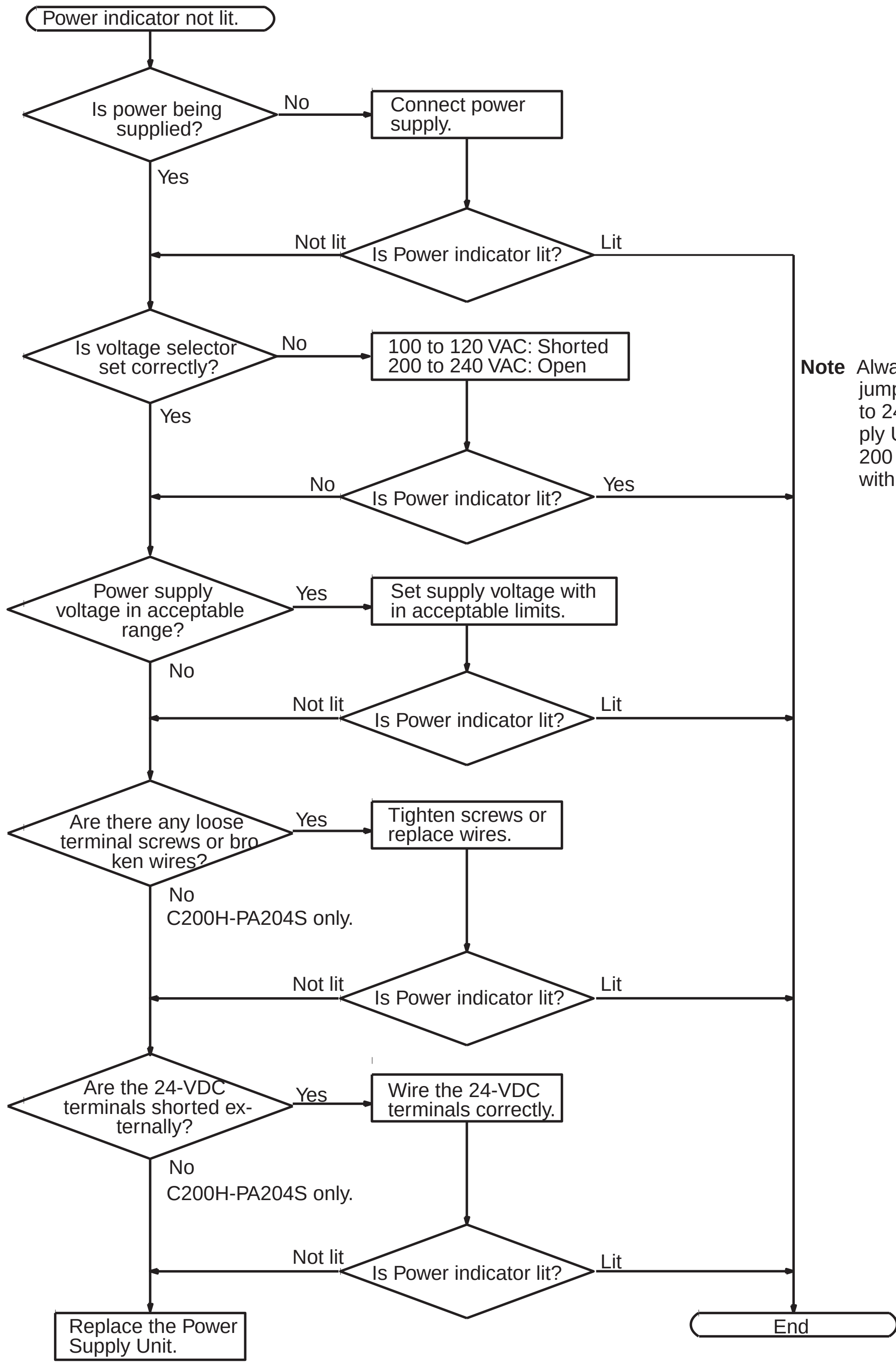
Power Supply Unit Indicator	CPU Unit Indicators			
POWER	**RUN**	**ERR/ALM**	**INH**	**COMM**
ON	ON	---	---	OFF

Check the setting of pin 5 on the DIP switch and the RS-232C port settings in the PLC Setup. Also check the cable connections. If a host computer is connected, check the communications settings of the serial port on the host computer and the communications program in the host computer.

16-2-6 Power Supply Check

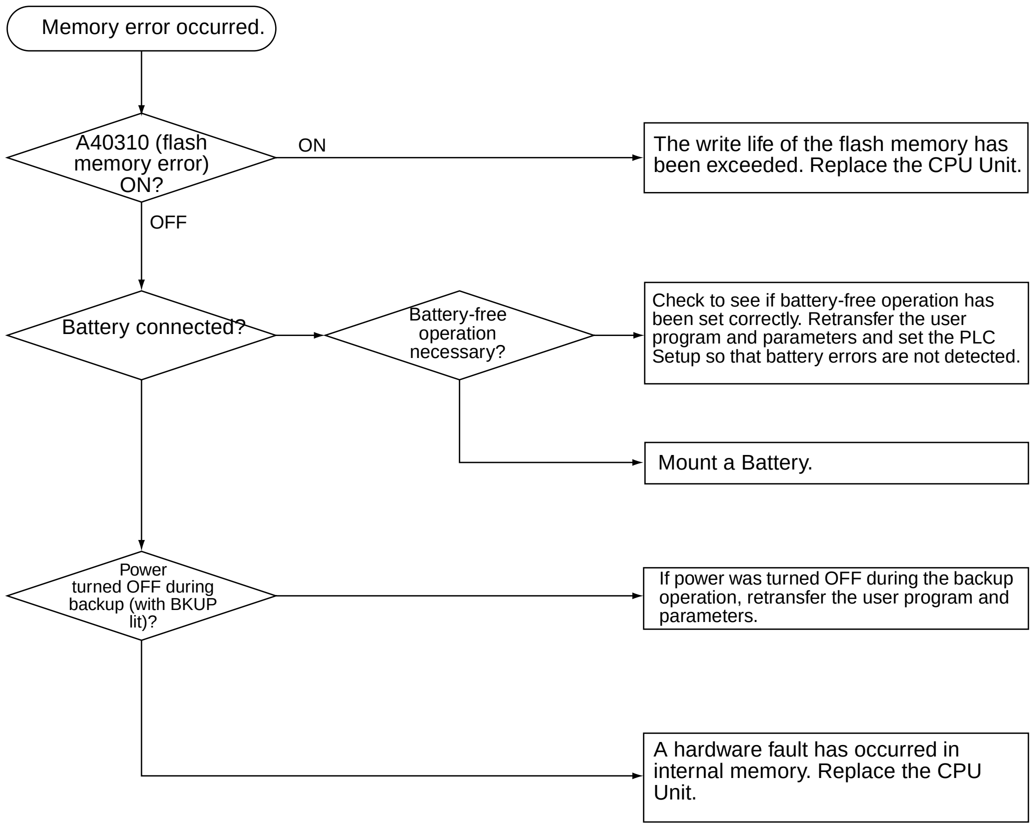
The allowable voltage ranges are shown in the following table.

Power Supply Unit	Power supply voltage	Allowable voltage range
C200HW-PA204, C200HW-PA204S, C200HW-PA204R, or C200HW-P209R	100 to 120 V AC	85 to 132 V AC
	200 to 240 V AC	170 to 264 V AC

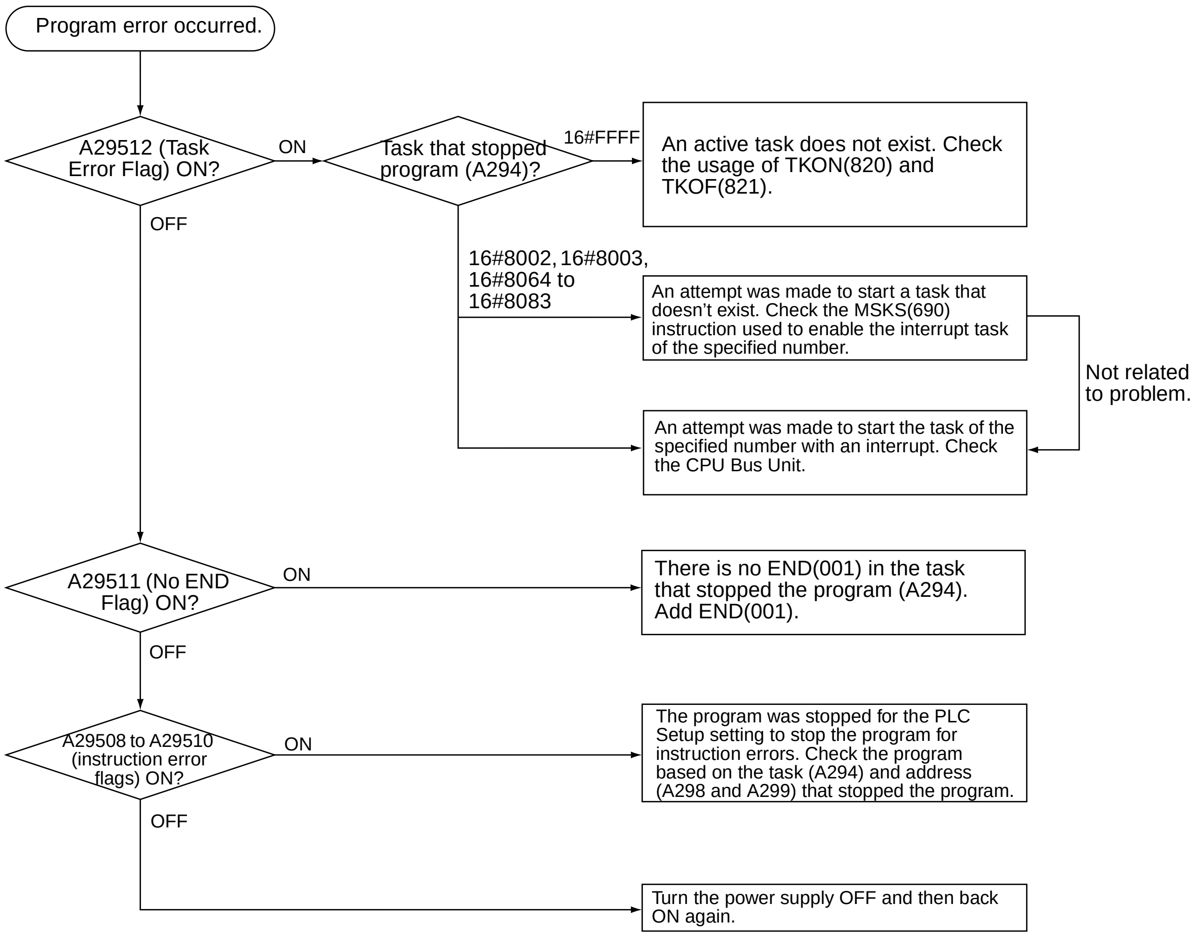


Note Always remove the short jumper before supplying 200 to 240 VAC. The Power Supply Unit will be destroyed if 200 to 240 VAC is supply with the jumper connected.

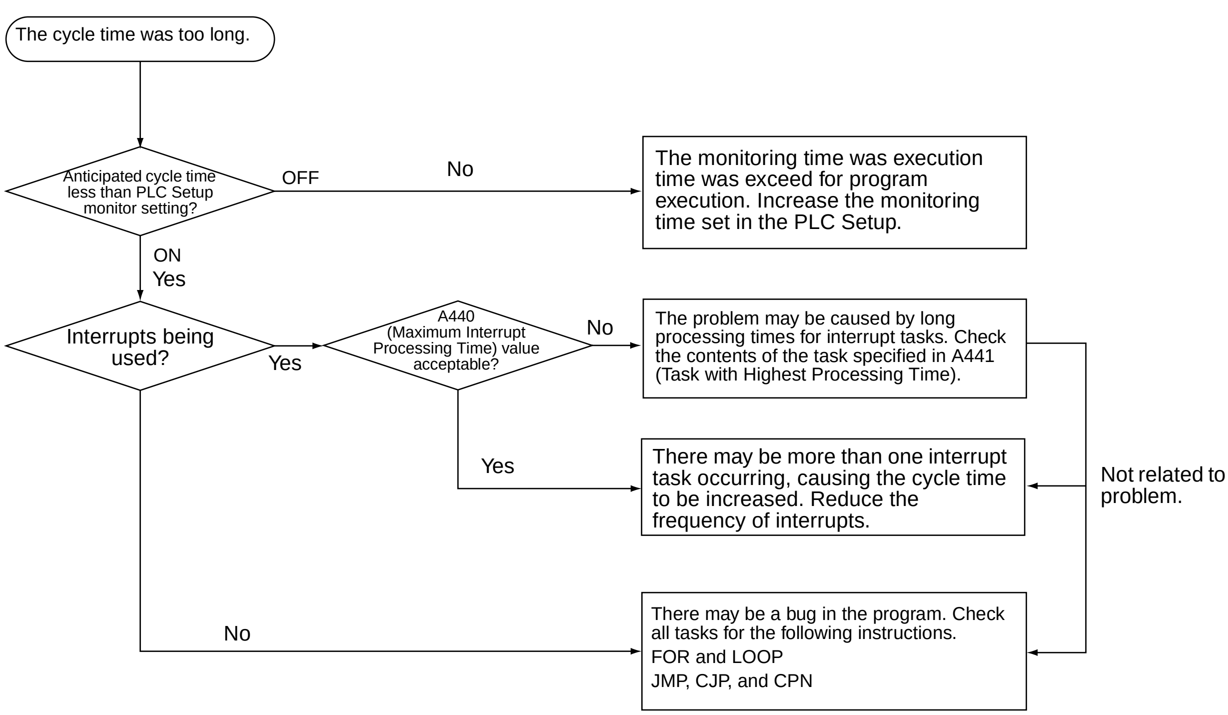
16-2-7 Memory Error Check



16-2-8 Program Error Check

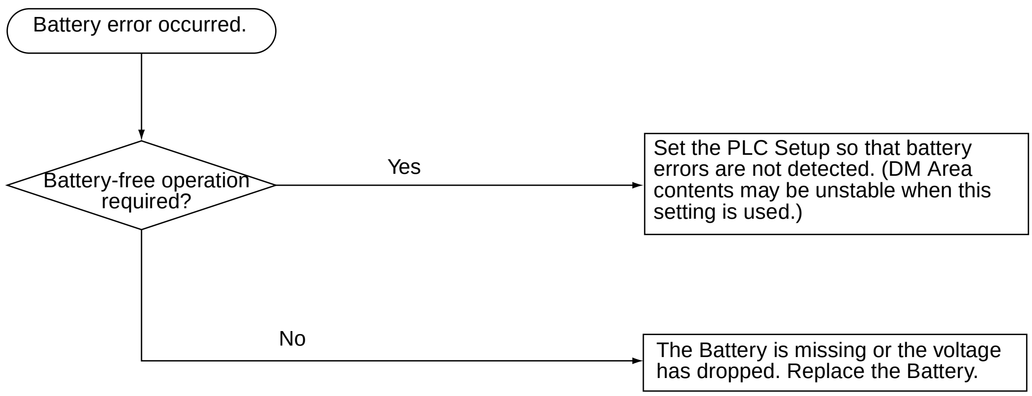


16-2-9 Cycle Time Too Long Error Check

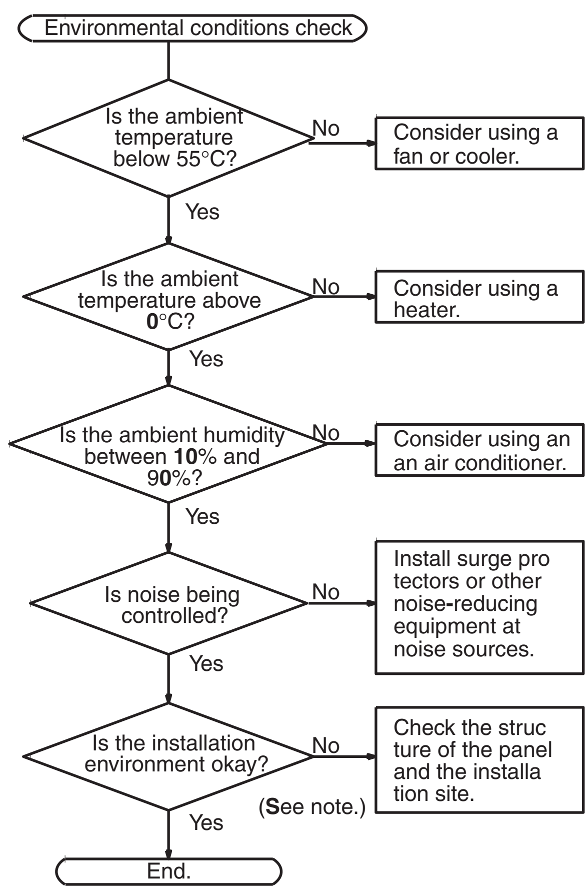


16-2-10 PLC Setup Setting Error Check

16-2-11 Battery Error Check



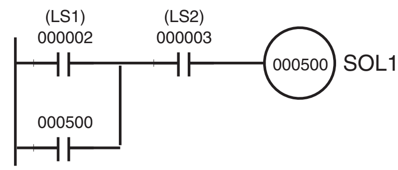
16-2-12 Environmental Conditions Check

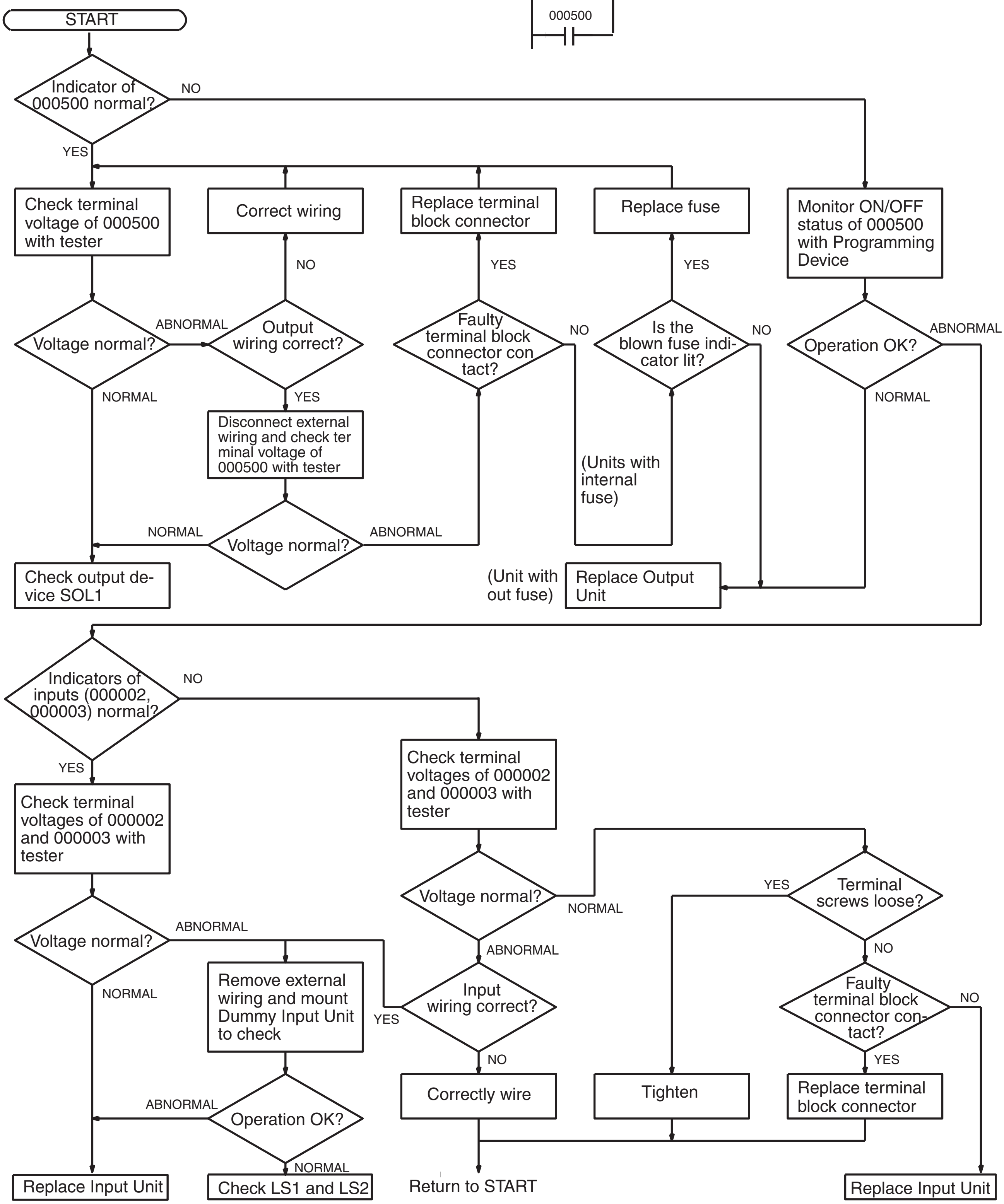


Note Check for corrosive gases, flammable gases, dust, dirt, salts, metal dust, direct light, water, oils, and chemicals.

16-2-13 I/O Check

The I/O check flowchart is based on the following ladder diagram section assuming that SOL1 does not turn ON.





16-3 Troubleshooting Racks and Units

CPU Racks and Standard Expansion Racks

Symptom	Cause	Remedy
POWER indicator is not lit.	PCB short-circuited or damaged.	Replace Power Supply Unit or Backplane.
	(1) Error in program.	Correct program
	(2) Power line is faulty.	Replace Power Supply Unit.
RUN output* does not turn ON. RUN indicator lit. (*C200HW-PS204R/209R)	Internal circuitry of Power Supply Unit is faulty.	Replace Power Supply Unit.
Serial Communications Unit or NE1S-series CPU Bus Unit does not operate or malfunctions.	(1) The I/O Connecting Cable is faulty. (2) The I/O bus is faulty.	Replace the I/O Connecting Cable Replace the Backplane.
Bits do not operate past a certain point.		
Error occurs in units of 8 points.		
I/O bit turns ON		
All bits in one Unit do not turn ON.		

Special I/O Units

Refer to the *Operation Manual* for the Special I/O Unit to troubleshoot any other errors.

Symptom	Cause	Remedy
The ERH and RUN indicators on the Special I/O Unit are lit.	I/O refreshing is not being performed for the Unit from the CPU Unit (CPU Unit monitoring error). It's possible that cyclic refreshing has been disabled for the Special I/O Unit in the Cyclic Refresh Disable Setting in the PLC Setup (i.e., the bit corresponding to the unit number has been set to 1).	Change the bit corresponding to the unit number to 0 to enable cyclic refreshing, or make sure that the Unit is refreshed from the program using IORF at least once every 11 s.

Input Units

Symptom	Cause	Remedy
Not all inputs turn ON or indicators are not lit.	(1) Power is not supplied to Input Unit.	Supply power
	(2) Supply voltage is low.	Adjust supply voltage to within rated range.
	(3) Terminal block mounting screws are loose.	Tighten screws.
	(4) Faulty contact of terminal block connector.	Replace terminal block connector.
Not all inputs turn ON (indicator lit).	Input circuit is faulty. (There is a short at the load or something else that caused an overcurrent to flow.)	Replace Unit.
Not all inputs turn OFF.	Input circuit is faulty.	Replace Unit.
Specific bit does not turn ON.	(1) Input device is faulty.	Replace input devices.
	(2) Input wiring disconnected.	Check input wiring
	(3) Terminal block screws are loose.	Tighten screws
	(4) Faulty terminal block connector contact.	Replace terminal block connector.
	(5) Too short ON time of external input.	Adjust input device
	(6) Faulty input circuit	Replace Unit.
	(7) Input bit number is used for output instruction.	Correct program.
Specific bit does not turn OFF.	(1) Input circuit is faulty.	Replace Unit.
	(2) Input bit number is used for output instruction.	Correct program.

Symptom	Cause	Remedy
Input irregularly turns ON/OFF.	(1) External input voltage is low or unstable.	Adjust external input voltage to within rated range.
	(2) Malfunction due to noise.	Take protective measures against noise, such as: 1. Increase input response time (PLC Setup 2. Install surge suppressor. 3. Install insulation transformer. 4. Install shielded cables between the Input Unit and the loads.
	(3) Terminal block screws are loose.	Tighten screws
	(4) Faulty terminal block connector contact.	Replace terminal block connector.
Error occurs in units of 8 points or 16 points, i.e., for the same common.	(1) Common terminal screws are loose.	Tighten screws
	(2) Faulty terminal block connector contact.	Replace terminal block connector.
	(3) Faulty data bus	Replace Unit.
	(4) Faulty CPU	Replace CPU.
Input indicator is not lit in normal operation.	Faulty indicator or indicator circuit.	Replace Unit.

Output Units

Symptom	Cause	Remedy
Not all outputs turn ON	(1) Load is not supplied with power.	Supply power
	(2) Load voltage is low.	Adjust voltage to within rated range.
	(3) Terminal block screws are loose.	Tighten screws
	(4) Faulty terminal block connector contact.	Replace terminal block connector.
	(5) An overcurrent (possibly caused by a short at the load) resulted in a blown fuse in the Output Unit. (Some Output Units provide an indicator for blown fuses.)	Replace fuse or Unit.
	(6) Faulty I/O bus connector contact.	Replace Unit.
	(7) Output circuit is faulty.	Replace Unit.
	(8) If the INH indicator is lit, the Output OFF Bit (A50015) is ON.	Turn A50015 OFF.
Not all outputs turn OFF	Output circuit is faulty.	Replace Unit.
Output of a specific bit number does not turn ON or indicator is not lit	(1) Output ON time too short because of a mistake in programming.	Correct program to increase the time that the output is ON.
	(2) Bit status controlled by multiple instructions.	Correct program so that each output bit is controlled by only one instruction.
	(3) Faulty output circuit.	Replace Unit.
Output of a specific bit number does not turn ON (indicator lit).	(1) Faulty output device.	Replace output device.
	(2) Break in output wiring.	Check output wiring.
	(3) Loose terminal block screws.	Tighten screws.
	(4) Faulty terminal block connector faulty.	Replace terminal block connector.
	(5) Faulty output bit.	Replace relay or Unit.
	(6) Faulty output circuit.	Replace Unit.
Output of a specific bit number does not turn OFF (indicator is not lit).	(1) Faulty output bit.	Replace relay or Unit.
	(2) Bit does not turn OFF due to leakage current or residual voltage.	Replace external load or add dummy resistor.

Symptom	Cause	Remedy
Output of a specific bit number does not turn OFF (indicator lit).	(1) Bit status controlled by multiple instructions.	Correct program.
	(2) Faulty output circuit.	Replace Unit.
Output irregularly turns ON/OFF.	(1) Low or unstable load voltage.	Adjust load voltage to within rated range
	(2) Bit status controlled by multiple instructions.	Correct program so that each output bit is controlled by only one instruction.
	(3) Malfunction due to noise.	Protective measures against noise: 1. Install surge suppressor. 2. Install insulation transformer. 3. Use shielded cables between the Output Unit and the loads.
	(4) Terminal block screws are loose.	Tighten screws.
	(5) Faulty terminal block connector contact.	Replace terminal block connector.
Error occurs in units of 8 points or 16 points, i.e., for the same common.	(1) Loose common terminal screw.	Tighten screws.
	(2) Faulty terminal block connector contact.	Replace terminal block connector.
	(3) An overcurrent (possibly caused by a short at the load) resulted in a blown fuse in the Output Unit.	Replace fuse or Unit.
	(4) Faulty data bus.	Replace Unit.
	(5) Faulty CPU.	Replace CPU.
Output indicator is not lit (operation is normal).	Faulty indicator.	Replace Unit.

SECTION 17
Inspection and Maintenance

This section provides inspection and maintenance information.

17-1 Inspections

Daily or periodic inspections are required in order to maintain the PLC's functions in peak operating condition.

17-1-1 Inspection Points

Although the major electronic components in NE1S-series PLCs have an extremely long life time, they can deteriorate under improper environmental conditions. Periodic inspections are thus required to ensure that the required conditions are being kept.

Inspection is recommended at least once every six months to a year, but more frequent inspections will be necessary in adverse environments.

Take immediate steps to correct the situation if any of the conditions in the following table are not met.

Inspection Points

No.	Item	Inspection	Criteria	Action
1	Source Power Supply	Check for voltage fluctuations at the power supply terminals.	The voltage must be within the allowable voltage fluctuation range. (See note.)	Use a voltage tester to check the power supply at the terminals. Take necessary steps to bring voltage fluctuations within limits.
2	I/O Power Supply	Check for voltage fluctuations at the I/O terminals.	Voltages must be within specifications for each Unit.	Use a voltage tester to check the power supply at the terminals. Take necessary steps to bring voltage fluctuations within limits.
3	Ambient environment	Check the ambient temperature. (Inside the control panel if the PLC is in a control panel.)	0 to 55°C	Use a thermometer to check the temperature and ensure that the ambient temperature remains within the allowed range of 0 to 55°C.
		Check the ambient humidity. (Inside the control panel if the PLC is in a control panel.)	Relative humidity must be 10% to 90% with no condensation.	Use a hygrometer to check the humidity and ensure that the ambient humidity remains within the allowed range.
		Check that the PLC is not in direct sunlight.	Not in direct sunlight	Protect the PLC if necessary.
		Check for accumulation of dirt, dust, salt, metal filings, etc.	No accumulation	Clean and protect the PLC if necessary.
		Check for water, oil, or chemical sprays hitting the PLC.	No spray on the PLC	Clean and protect the PLC if necessary.
		Check for corrosive or flammable gases in the area of the PLC.	No corrosive or flammable gases	Check by smell or use a sensor.
		Check the level of vibration or shock.	Vibration and shock must be within specifications.	Install cushioning or shock absorbing equipment if necessary.
		Check for noise sources near the PLC.	No significant noise sources	Either separate the PLC and noise source or protect the PLC.
4	Installation and wiring	Check that each Unit is installed securely.	No looseness	Tighten loose screws with a Phillips-head screwdriver.
		Check that cable connectors are fully inserted and locked.	No looseness	Correct any improperly installed connectors.
		Check for loose screws in external wiring.	No looseness	Tighten loose screws with a Phillips-head screwdriver.
		Check crimp connectors in external wiring.	Adequate spacing between connectors	Check visually and adjust if necessary.
		Check for damaged external wiring cables.	No damage	Check visually and replace cables if necessary.

No.	Item	Inspection	Criteria	Action
5	User-serviceable parts	Check whether internal relays in Relay Output Units have reached their service life.	No open relay contacts, improper operation, or faulty contacts Refer to *Appendix D Specifications of Basic I/O Units* for information on relay service life.	Replace the Unit.
		Check whether the CJ1W-BAT01 Battery has reached its service life.	Life expectancy is 5 years at 25°C, less at higher temperatures. (From 1.7 to 5 years depending on model, power supply rate, and ambient temperature.)	Replace the battery when its service life has passed even if a battery error has not occurred. (Battery life depends upon the model, the percentage of time in service, and ambient conditions.)

Note The following table shows the allowable voltage fluctuation ranges for source power supplies.

Supply voltage	Allowable voltage range
100 to 120 V AC	85 to 132 V AC
200 to 240 V AC	170 to 264 V AC
24 V DC	19.2 to 28.8 V DC

Tools Required for Inspections

Required Tools

- Slotted and Phillips-head screwdrivers
- Voltage tester or digital voltmeter
- Industrial alcohol and clean cotton cloth

Tools Required Occasionally

- Synchroscope
- Oscilloscope with pen plotter
- Thermometer and hygrometer (humidity meter)

17-1-2 Unit Replacement Precautions

Check the following after replacing any faulty Unit.

- Do not replace a Unit until the power is turned OFF.
- Check the new Unit to make sure that there are no errors.
- If a faulty Unit is being returned for repair, describe the problem in as much detail as possible, enclose this description with the Unit, and return the Unit to your OMRON representative.
- For poor contact, take a clean cotton cloth, soak the cloth in industrial alcohol, and carefully wipe the contacts clean. Be sure to remove any lint prior to remounting the Unit.

Note

1. When replacing a CPU Unit, be sure that not only the user program but also all other data required for operation is transferred to or set in the new CPU Unit before starting operation, including DM Area and HR Area settings. If data area and other data are not correct for the user program, unexpected accidents may occur. Be sure to include CPU Bus Unit data, which are stored as parameters in the CPU Unit. Refer to the CPU Bus Unit and Special I/O Unit operation manuals for details on the data required by each Unit.
2. The simple backup operation can be used to store the user program and all parameters for the NE1S CPU Unit, ControlNet Units, DeviceNet Units, Serial Communications Units, and other specific Units in a Memory Card

as backup files. A Memory Card and the simple backup operation can be used to easily restore data after replacing any of these Units.

17-2 Replacing User-serviceable Parts

The following parts should be replaced periodically as preventative maintenance. The procedures for replacing these parts are described later in this section.

- Battery (the CPU Unit's RAM-backup battery)

17-2-1 Battery Replacement

Battery Functions

The battery retains the following data of the CPU Unit's RAM when the main power supply is OFF.

- The user program

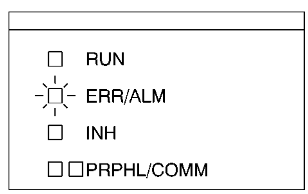
If the battery is not installed or battery voltage drops too low, the data in RAM will be lost when the main power supply goes OFF.

Battery Service Life and Replacement Period

At 25°C, the maximum service life for batteries is 5 years whether or not power is supplied to the CPU Unit while the battery is installed. The battery's lifetime will be shorter when it is used at higher temperatures and when power is not supplied to the CPU Unit for long periods. In the worst case conditions, the battery will last for only 1.7 years.

Low Battery Indicators

If the PLC Setup has been set to detect a low-battery error, the ERR/ALM indicator on the front of the CPU Unit will flash when the CPU Unit detects that the battery is nearly discharged.



When the ERR/ALM indicator flashes, connect the NE Programmer to the serial or Ethernet port and read the error message. If the Battery Error Flag (A40204) is ON*, first check whether the battery is properly connected to the CPU Unit. If the battery is properly connected, replace the battery as soon as possible.

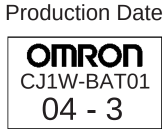
Once a low-battery error has been detected, it will take 5 days before the battery fails. Battery failure can be delayed by ensuring that the CPU Unit power is not turned OFF until the battery has been replaced.

Note *The PLC Setup must be set to detect a low-battery error (Detect Low Battery). If this setting has not been made, the Battery Error Flag (A40204) will not go ON when the battery fails.

Replacement Battery

The following diagram shows the CJ1W-BAT01 Battery Set. Be sure to install a replacement battery within 2 years of the production date shown on the battery's label.

Production Date



Manufactured in March 2004.

⚠ Caution The battery can be replaced with the power ON, but be sure to touch a grounded metal object to discharge any static electricity before replacing the battery. After replacing the battery, connect a Programming Device and clear the battery error.

CPU Unit Battery Replacement Procedure

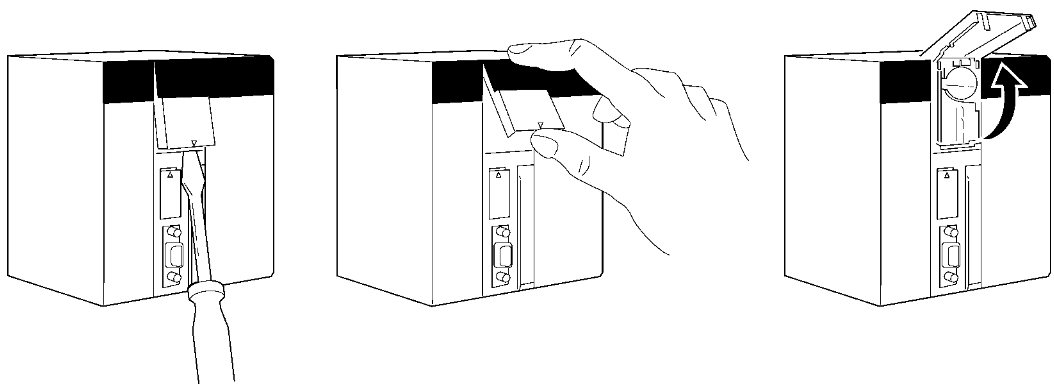
Use the following procedure to replace a battery that has been completely discharged.

1,2,3...

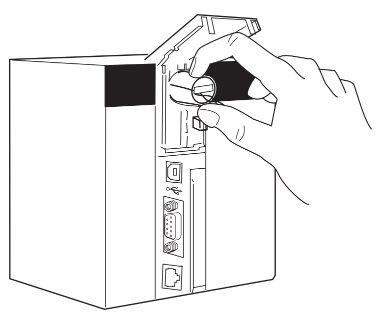
1. Turn OFF the power to the PLC. (If the power was already OFF, turn the power ON for at least one minute before turning the power OFF again.)

Note There is a capacitor in the CPU Unit that will back up memory while the battery is being replaced. If this capacitor is not completely charged by turning ON the power supply for one minute, data may be lost during battery replacement.

2. Insert a small flat-blade screwdriver into the notch at the bottom of the battery compartment cover and lift open the cover.



3. Remove the old battery from the compartment and replace it with a new one.



Note Complete the battery replacement procedure within five minutes of turning OFF the power supply. If more than three minutes elapse, data may be lost during battery replacement.

Battery Replacement Precautions

Note

1. With the NE1S CPU Units, even if this procedure is not used and the old battery is disconnected with the power OFF (power OFF and no battery connected), RAM data will be backed up for a short time by an internal capacitor. In this case, be sure to connect the new battery quickly, before the internal capacitor discharges.

2. With the NE1S CPU Units, if this procedure is not used and the old battery is disconnected with the power ON (power ON and no battery connected), RAM data will still be backed up. However, be sure to touch a grounded metal object to discharge any static electricity before replacing the battery.

⚠ Caution Do not short the battery terminals or charge, disassemble, heat, or incinerate the battery. Do not subject the battery to strong shocks. Doing any of these may result in leakage, rupture, heat generation, or ignition of the battery. Dispose of any battery that has been dropped on the floor or otherwise subjected to excessive shock. Batteries that have been subjected to shock may leak if they are used. Also, UL standards required that batteries be replaced only by experienced technicians. Do not allow unqualified persons to replace batteries.

Appendix A
Variable Applications Guidelines

This section provides guidelines for using function blocks with the NE Programmer.

Using Variable Data Types

Integer Data Types (1, 2, or 4-word Data)

Use the following data types when handling single numbers in 1, 2, or 4-word units.

- INT and UINT
- DINT and DINT

Note Use signed integers if the numbers being used will fit in the range.

Word Data Types (1, 2, or 4-word Data)

Use the following data types when handling groups of data (non-numeric data) in 1, 2, or 4-word units.

- WORD
- DWORD

Array Settings

Array Variables
Use for First or End Addresses of Word Ranges

When specifying an instruction operand that is the first address or end address of a range of words (see note), the required values cannot be passed to variables through input parameters or output parameters.

Note Refer to *Appendix G Instruction Support and Operand/Variable Restrictions* to determine which instruction operands must have array variables because they specify the first/end address of a range of words.

In this case, prepare an array variable with the required number of array elements, set the data in each array element in the function block, and specify the beginning (or end) array variable in the operand. Using an array variable allows you to specify the first address or end address of a range of words.

Handling a Single String of Data in Multiple Words

In this example, an array contains the directory and filename (operand S2) for an FREAD instruction.

- Variable
  Internal variable (VAR), data type = WORD, array setting with 10 elements, variable names = filename[0] to filename[9]
- Ladder Programming

MOV 16#5C31 file_name[0]
MOV 16#3233 file_name[1] ← Set data in each array element.
MOV 16#0000 file_name[2])
FREAD (*omitted*) (*omitted*) file_name[0] (*omitted*) ← Specify the first element of the array in the instruction operand.

Handling Control Data in Multiple Words

In this example, an array contains the number of words and first source word (operand S1) for an FREAD instruction.

- Variable
  Internal variable (VAR), data type = DINT, array setting with 3 elements, variable names = read_num[0] to read_num[9]

- Ladder Programming

MOVL 10#100 read_num[0] (*No._of_words*)
MOVL 10#0 read_num[1] (*1st_source_word*) ← Set data in each array element.

FREAD (*omitted*) (*omitted*) file_name[0] (*omitted*) ← Specify the first element of the array in the instruction operand.

Handling a Block of Read Data in Multiple Words

The allowed amount of read data must be determined in advance and an array must be prepared that can handle the maximum amount of data. In this example, an array receives the FREAD instruction's read data (operand D).

- Variable
  Internal variable, data type = WORD, array setting with 100 elements, variable names = read_data[0] to read_data[99]
- Ladder Programming

FREAD (*omitted*) (*omitted*) (*omitted*) read_data[0]

Division Using Integer Array Variables (Ladder Programming Only)

A two element array can be used to store the result from a ladder program's SIGNED BINARY DIVIDE (/) instruction. The result from the instruction is D (quotient) and D+1 (remainder). This method can be used to obtain the remainder from a division operation in ladder programming.

Note When ST language is used, it isn't necessary to use an array to receive the result of a division operation. Also, the remainder can't be calculated directly in ST language. The remainder must be calculated as follows:
Remainder = Dividend – (Divisor × Quotient)

Specifying External Variables or Physical Addresses for Function Blocks

Specify either the external variable or the physical address to enable reading or writing of Auxiliary Area bits within the algorithm (i.e., within the execution cycle) in the function block. (Auxiliary Area bits can also be used to receive and pass I/O variables.)

Example: Using Communications Port Enabled Flags (A20200 to A20207) in Function Blocks

The function block input variables are executed by referencing the items copied to the instances when the function block is called. The Communications Port Enabled Flags turn ON/OFF asynchronously with the program execution. When these bits are referenced as input variables in a function block, the Communications Port Enabled Flags are copied to the Instance Area when the function block is called. After copying, even if the value is changed prior to referencing the Communications Port Enabled Flags, detection is not possible in the function block program. Therefore, either define the Communications Port Enabled Flags as local variables and reference them as external variables from the function block, or specify the physical addresses (e.g., A20200).

Appendix B
Structured Text Keywords

Operators

Operation	Symbol	Data types supported by operator	Example	Value (example)	Priority 1: Lowest 11: Highest
Parentheses and brackets	(*expression*), *array*[*index*]	---	(2 + 3) * (4 + 5)	45	1
Function evaluation	*identifier* (*operand_list*)	---	---	---	2
Exponential	**	Not supported	---	---	3
Complement	–	INT, DINT	–10		4
Boolean complement	NOT	BOOL, WORD, DWORD	NOT TRUE	FALSE	4
Multiplication	*	INT, DINT, UINT, UDINT	10 * 3	30	5
Division	/	INT, DINT, UINT, UDINT	6/2	3	5
Remainder calculation	MOD	INT, DINT, UINT, UDINT	17 MOD 10	7	5
Addition	+	INT, DINT, UINT, UDINT, STRING	2 + 3	5	6
Subtraction	–	INT, DINT, UINT, UDINT	4 – 2	2	6
Comparisons	<, >, <=, >=	BOOL, WORD, DWORD, INT, DINT, UINT, UDINT, STRING	4 > 12	FALSE	7
Equality	=	BOOL, WORD, DWORD, INT, DINT, UINT, UDINT, STRING	10#16 = 16#10	TRUE	8
Non-equality	<>	BOOL, WORD, DWORD, INT, DINT, UINT, UDINT	8 <> 16	TRUE	8
Boolean AND	&, AND	BOOL, WORD, DWORD	TRUE & FALSE	FALSE	9
Boolean exclusive OR	XOR	BOOL, WORD, DWORD	TRUE XOR FALSE	TRUE	10
Boolean OR	OR	BOOL, WORD, DWORD	TRUE OR FALSE	TRUE	11

Restrictions

- Ladder programming special instructions cannot be used.
- Writing “100” means “10#100.”

Conditional Statements

Keyword	Example	Function
RETURN	RETURN;	Return. Leaves the called function block and returns to the calling POU. With the NE Programmer, the RETURN statement can be used in the function block only.
IF	IF a < b THEN c: = 1; ELSIF a = b THEN C: = 2; ELSE c: = 3; END_IF;	Selection. Evaluates group expression when the condition (a<b) is true. When the condition is false, the expression is not evaluated and the group following ELSE is evaluated.
CASE	CASE f OF 1: a: = 3; 2..5: a: = 4; ELSE a: = 0; END_CASE;	Selection. Evaluates one group according to the value in the expression following the keyword CASE. The variable or expression *f* must be an INT data type.

Keyword	Example	Function

FOR	FOR a: = 1 TO 10 BY 3 DO
	f[a]: = b;
	END_FOR;
		Repetition.
		Evaluates the expressions in the range of DO to END_FOR, starting with 1 for variable *a* and sequentially adding 3 each execution, repeating until *a* reaches 10. It starts with the value of variable *a*, and ends with the value following TO, increasing with the value of BY. All values must be ANY_INT types.
		Note If BY is omitted, the default is 1 and all data must be INT type.

WHILE	WHILE b > 1 DO b: = b/2 END_WHILE;	Repetition. Evaluates the expressions of one group, repeating until the condition (b>1) is false. The condition for this expression is evaluated at the beginning of the loop, and when it is not true, the loop is not evaluated.

REPEAT	REPEAT
	a: = a*b;
	UNTIL a < 10000
	END_REPEAT;
		Repetition.
		Evaluates the expressions of one group, repeating until the condition (a<10000) is true. The condition for this expression is evaluated at the end of the loop, i.e., even if it is not true, the loop is evaluated at least once.

EXIT	FOR a: = 1 TO 2 DO IF flag THEN EXIT; END_IF SUM: = SUM + a END_FOR	End. This statement can be used to exit an evaluation for a repetition statement.

Functions

Type	Name	Description
Math	ADD	Adds
	MUL	Multiplies
	SUB	Subtracts
	MOD	Finds remainder
	DIV	Divides
	MOVE	Assigns
Bit manipulation	SHL	Shifts 1 bit left
	SHR	Shifts 1 bit right
	ROR	Rotates 1 bit right without carry bit
	ROL	Rotates 1 bit left without carry bit
Logic operation	AND	Logical AND
	OR	Logical OR
	XOR	Logical exclusive OR
	NOT	Logical NOT
Character string manipulation	LEFT	Gets character string from the left
	RIGHT	Gets character string from the right
	MID	Gets character string from any position
	DELETE	Deletes character string
	CONCAT	Concatenates character strings
	INSERT	Inserts character string
	REPLACE	Replaces character string
	LEN	Gets character string length
	FIND	Finds character string j
Selection	MAX	Gets maximum value
	MIN	Gets minimum value

Type	Name	Description
Data type conversion	INT_TO_UINT	Converts INT to UINT
	INT_TO_DINT	Converts INT to DINT
	INT_TO_UDINT	Converts INT to UDINT
	INT_TO_WORD	Converts INT to WORD
	UINT_TO_INT	Converts UINT to INT
	UINT_TO_DINT	Converts UINT to DINT
	UINT_TO_UDINT	Converts UINT to UDINT
	UINT_TO_WORD	Converts UINT to WORD
	DINT_TO_INT	Converts DINT to INT
	DINT_TO_UINT	Converts DINT to UINT
	DINT_TO_UDINT	Converts DINT to UDINT
	DINT_TO_DWORD	Converts DINT to DWORD
	UDINT_TO_INT	Converts UDINT to INT
	UDINT_TO_UINT	Converts UDINT to UINT
	UDINT_TO_DINT	Converts UDINT to DINT
	UDINT_TO_DWORD	Converts UDINT to DWORD
	WORD_TO_INT	Converts WORD to INT
	WORD_TO_UINT	Converts WORD to UINT
	WORD_TO_DWORD	Converts WORD to DWORD
	DWORD_TO_INT	Converts DWORD to INT
	DWORD_TO_UINT	Converts DWORD to UINT
	DWORD_TO_WORD	Converts DWORD to WORD

ST Language Reserved Words

The following ST language reserved words cannot be used as identifiers.

N	END_RESOURCE	DATE	F_TRIG	~
R	RETAIN	TIME_OF_DAY	CTU	*
L	RETURN	TIME	CTD	/
D	STEP	AND	CTUD	MOD
P	END_STEP	OR	TP	+
SD	STRUCT	NOT	TON	-
DS	END_STRUCT	SHL	TOF	<
SL	TASK	SHR	RTC	>
ACTION	TRANSITION	ROR	LD	<=
END_ACTION	FROM	ROL	LDN	>=
ARRAY	TO	SUB	ST	=
AT	END_TRANSITION	MUL	STN	<>
CASE	TRUE	MOD	S	&
OF	TYPE	EXPT	R	VAR_SYSTEM
ELSE	END_TYPE	ABS	ANDN	CHANNEL
END_CASE	VAR	SQRT	AND(	FI
CONFIGURATION	END_VAR	LN	ANDN(	
END_CONFIGURATION	VAR_INPUT	LOG	ORN	
CONSTANT	VAR_IN_OUT	EXP	OR(	
EN	VAR_OUTPUT	SIN	ORN(	
ENO	VAR_EXTERNAL	COS	XOR	
EXIT	VAR_ACCESS	TAN	XORN	
FALSE	VAR_GLOBAL	ASIN	XOR(	

F_EDGE	VAR_TEMP	ACOS	XORN(	
FOR	WHILE	ATAN	ADD	
BY	END_WHILE	USINT_TO_DINT	ADD(	
DO	WITH	BOOL_TO_BYTE	SUB(	
END_FOR	ANY	SEL	MUL(	
FUNCTION	ANY_NUM	MIN	DIV	
END_FUNCTION	ANY_REAL	MAX	DIV(	
FUNCTION_BLOCK	LREAL	LIMIT	GT	
END_FUNCTION_BLOCK	REAL	MUX	GT(	
IF	ANY_INT	T	GE(	
THEN	SINT	GE	EQ(	
ELSEIF	INT	EQ	NE(	
ELSE	DINT	LT	LE	
END_IF	LINT	NE	LE(	
INITIAL_STEP	USINT	LEN	JMP	
END_STEP	UINT	LEFT	JMPNC	
PROGRAM	ULINT	RIGHT	JMPC	
WITH	UDINT	MID	CAL	
END_PROGRAM	ANY_BIT	CONCAT	CALNC	
R_EDGE	BOOL	INSERT	CALC	
READ_ONLY	BYTE	DELETE	RET	
READ_WRITE	WORD	REPLACE	RETNC	
REPEAT	DWORD	FIND	RETC	
UNTIL	LWORD	SR	(	
END_REPEAT	STRING	RS	)	
RESOURCE	ANY_DATE	SEMA	Function	
ON	DATE_AND_TIME	R_TRIG	**	

Appendix C
External Variables

Classification	Name	External variable in NE Programmer	Data type	Address
Conditions Flags	Greater Than or Equals (GE) Flag	P_GE	BOOL	CF00
	Not Equals (NE) Flag	P_NE	BOOL	CF001
	Less Than or Equals (LE) Flag	P_LE	BOOL	CF002
	Instruction Execution Error (ER) Flag	P_ER	BOOL	CF003
	Carry (CY) Flag	P_CY	BOOL	CF004
	Greater Than (GT) Flag	P_GT	BOOL	CF005
	Equals (EQ) Flag	P_EQ	BOOL	CF006
	Less Than (LT) Flag	P_LT	BOOL	CF007
	Negative (N) Flag	P_N	BOOL	CF008
	Overflow (OF) Flag	P_OF	BOOL	CF009
	Underflow (UF) Flag	P_UF	BOOL	CF010
	Access Error Flag	P_AER	BOOL	CF011
	Always OFF Flag	P_Off	BOOL	CF114
	Always ON Flag	P_On	BOOL	CF113
Clock Pulses	0.02 second clock pulse bit	P_0_02s	BOOL	CF103
	0.1 second clock pulse bit	P_0_1s	BOOL	CF100
	0.2 second clock pulse bit	P_0_2s	BOOL	CF101
	1 minute clock pulse bit	P_1mim	BOOL	CF104
	1.0 second clock pulse bit	P_1s	BOOL	CF102
Auxiliary Area Flags/ Bits	First Cycle Flag	P_First_Cycle	BOOL	A200.11
	First Task Execution Flag	P_First_Cycle_Task	BOOL	A200.15
	Maximum Cycle Time	P_Max_Cycle_Time	DWORD	A262
	Present Scan Time	P_Cycle_Time_Value	DWORD	A264
	Cycle Time Error Flag	P_Cycle_Time_Error	BOOL	A401.08
	Low Battery Flag	P_Low_Battery	BOOL	A402.04
	Output OFF Bit	P_Output_Off_Bit	BOOL	A500.15

Appendix D
Specifications of Basic I/O Units

Input Units

Category	Name	Specifications	Model	Page
Basic Input Units with Terminal Blocks	DC Input Units	24 V DC, 16 inputs	CS1W-ID211	493
	Interrupt Input Units	24 V DC, 16 inputs	CS1W-INT01	495
	High-speed Input Unit	24 V DC, 16 inputs	CS1W-IDP01	496
Basic Input Units with Connectors	DC Input Units	24 V DC, 32 inputs	CS1W-ID231	498
		24 V DC, 64 inputs	CS1W-ID261	500
		24 V DC, 96 inputs	CS1W-ID291	502
		Simultaneously ON 24-V DC inputs for CS1W-ID291/MD291/MD292		504

Output Units

Category	Name	Specifications	Model	Page
Basic Outputs Units with Terminal Blocks	Relay Output Units	250 V AC/24 V DC, 2 A; 120 V DC, 0.1 A; independent contacts, 8 outputs	CS1W-OC201 (See note.)	506
		250 V AC/24 V DC, 2 A; 120 V DC, 0.1 A; 16 outputs	CS1W-OC211	505
		Relay contact outputs		507
	Transistor Output Units, sinking outputs	12 to 24 V DC, 0.5 A, 16 outputs	CS1W-OD211	511
	Transistor Output Units, sinking	12 to 24 V DC, 0.5 A, 32 outputs	CS1W-OD231	512
		12 to 24 V DC, 0.3 A, 64 outputs	CS1W-OD261	514
		12 to 24 V DC, 0.1 A, with fuse burnout detection circuit, 96 outputs	CS1W-OD291	516
	Transistor Output Units, sourcing outputs	24 V DC, 0.5 A, load short-circuit protection, 16 outputs	CS1W-OD212	518
		Load short-circuit protection for CS1W-OD212/OD232/OD262/MD262		519
		24 V DC, 0.3 A, load short-circuit protection, 64 outputs	CS1W-OD262	523
		24 V DC, with fuse burnout detection circuit, 0.1 A, 96 outputs	CS1W-OD292	525

Note This Unit uses only 8 bits for external I/O even though 16 bits (1 word) are allocated. This Unit is also treated as a 16-point Unit in the I/O tables.

Mixed I/O Units

Category	Name	Specifications	Model	Page
Basic I/O Units with Connectors	DC Input/Transistor Output Units	24 V DC inputs; 12 to 24 V DC, 0.3-A, sinking outputs; 32 inputs, 32 outputs	CS1W-MD261	528
		24 V DC inputs; 12 to 24 V DC, 0.1A, sinking outputs with fuse burnout detection circuit; 48 inputs, 48 outputs	CS1W-MD291	530
		24 V DC inputs 24 V DC, 0.3 A, sourcing outputs with load short-circuit protection; 32 inputs, 32 outputs	CS1W-MD262	533
		24 V DC inputs 24 V DC, 0.1 A, sourcing outputs with fuse burnout detection circuit; 48 inputs, 48 outputs	CS1W-MD292	535
	TTL I/O Unit	5 V DC, 3.5 mA inputs; 5 V DC, 35 mA, outputs; 32 inputs, 32 outputs	CS1W-MD561	538

Reading Terminal Connection Diagrams

- I/O terminals in terminal connection diagrams are shown as viewed from the front panel of the Unit.
- Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on all Units.
- A0 to A20 and B0 to B20 are printed on the Units.

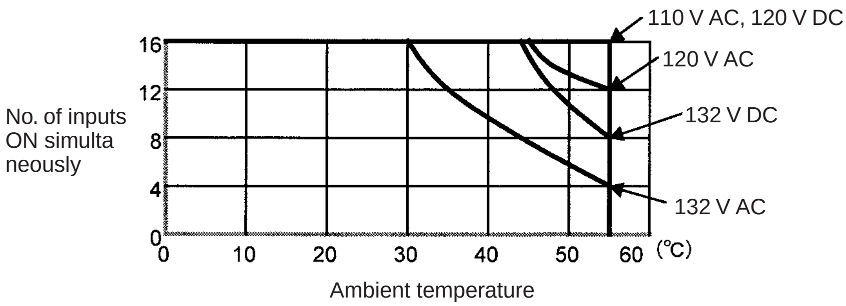
Basic I/O Units

CS1W-IA111 100-V AC Input Unit (16 Points)

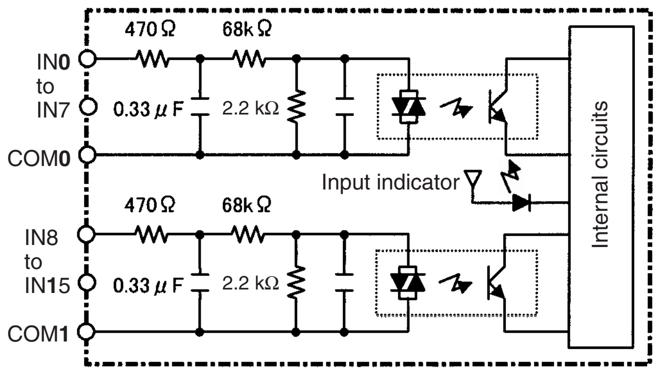
Rated Input Voltage	100 to 120 V AC (50/60 Hz), 100 to 120 V DC
Operating Input Voltage	85 to 132 V AC (50/60 Hz), 85 to 132 V DC
Input Impedance	10 kΩ (50 Hz), 8 kΩ (60 Hz), 69 kΩ (DC)
Input Current	10 mA typical (at 100 V AC), 1.5 mA typical (at 100 V DC)
ON Voltage	65 V AC min., 75 V DC min.
OFF Voltage	20 V AC max., 25 V DC max.
ON Response Time	18 ms max. when PLC Setup on default setting (8 ms) (See note.)
OFF Response Time	63 ms max. when PLC Setup on default setting (8 ms) (See note.)
Insulation Resistance	20 MΩ between external terminals and the GR terminal (500 V DC)
Dielectric Strength	2,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
No. of Circuits	2 (each 8 points/common)
Number of Inputs ON Simultaneously	100% simultaneously ON (for 110 V AC, 120 V DC) Refer to the diagram below.
Internal Current Consumption	110 mA 5 V DC max.
Weight	260 g max.

Note The Input ON and OFF response times for Basic I/O Units can be set to 0 ms, 0.5 ms, 1 ms, 2 ms, 4 ms, 8 ms, 16 ms, or 32 ms in the PLC Setup. When the response times have been set to 0 ms, the ON response time will be 10 ms maximum and the OFF response time will be 40 ms maximum due to internal element delays.

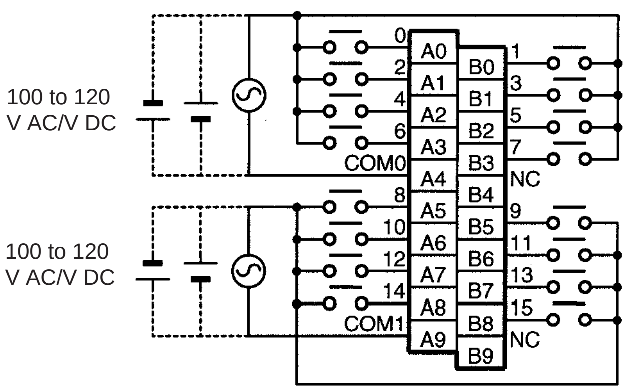
Number of Inputs ON Simultaneously



Circuit Layout



Terminal Connections



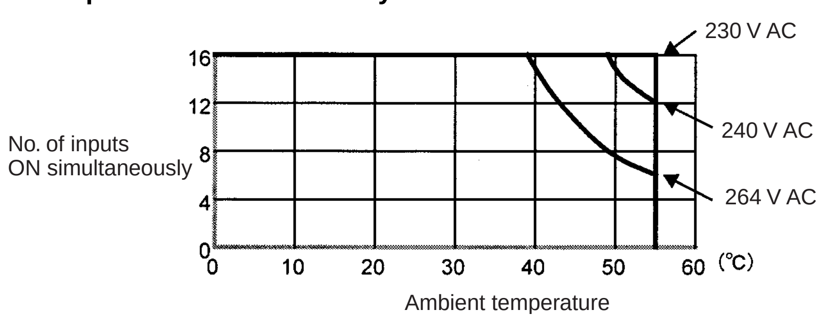
Note Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on the Unit.

CS1W-IA211 200-V AC Input Unit (16 Points)

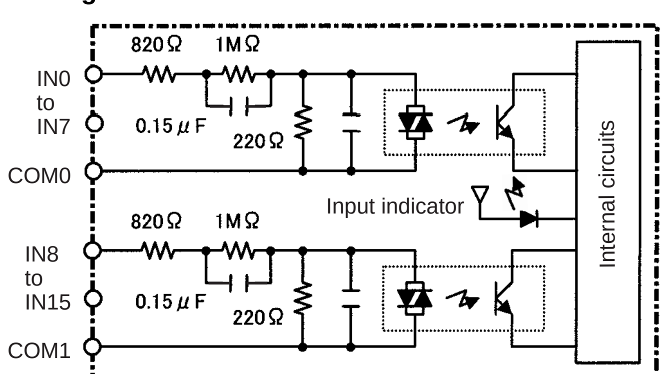
Rated Input Voltage	200 to 240 V AC (50/60 Hz)
Operating Input Voltage	170 to 264 V AC (50/60 Hz)
Input Impedance	21 kΩ (50 Hz), 18 kΩ (60 Hz)
Input Current	10 mA typical (at 200 V AC)
ON Voltage/ON current	120 V AC min.
OFF Voltage/OFF current	40 V AC max.
ON Response Time	18 ms max. when PLC Setup on default setting (8 ms) (See note.)
OFF Response Time	48 ms max. when PLC Setup on default setting (8 ms) (See note.)
No. of Circuits	2 (each 8 points/common)
Number of Inputs ON Simultaneously	100% simultaneously ON (for 230 V AC). Refer to the diagram below.
Insulation Resistance	20 MΩ between external terminals and the GR terminal (500 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	110 mA 5 V DC max.
Weight	260 g max.

Note The Input ON and OFF response times for Basic I/O Units can be set to 0 ms, 0.5 ms, 1 ms, 2 ms, 4 ms, 8 ms, 16 ms, or 32 ms in the PLC Setup. When the response times have been set to 0 ms, the ON response time will be 10 ms maximum and the OFF response time will be 40 ms maximum due to internal element delays.

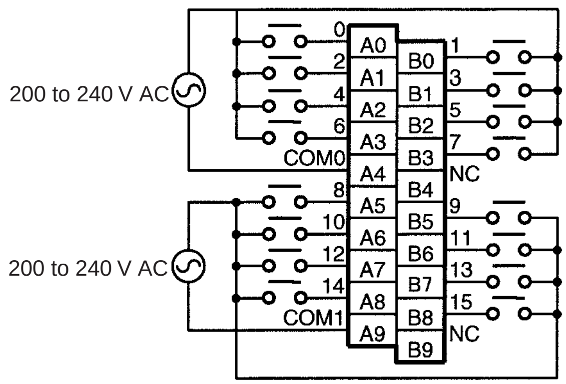
Number of Inputs ON Simultaneously



Circuit Configuration



Terminal Connections

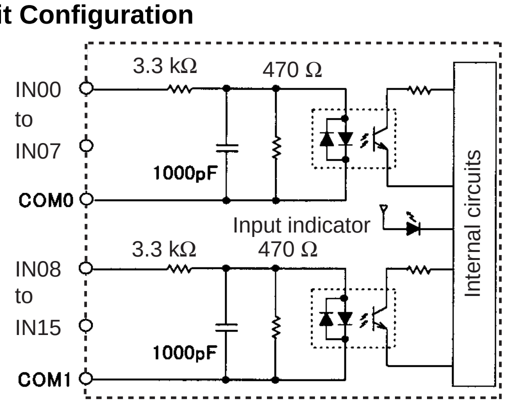


Note Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on the Unit.

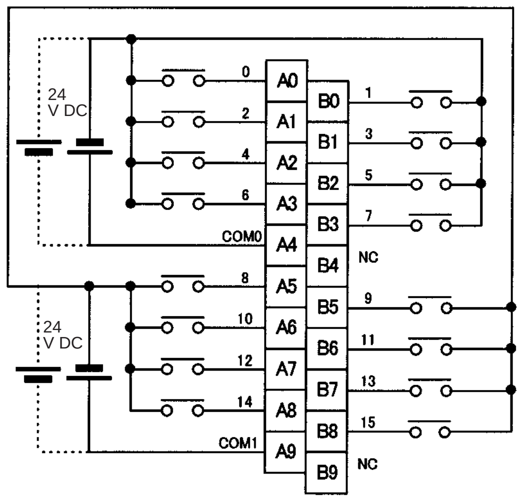
CS1W-ID211 DC Input Unit (16 Points)

Rated Input Voltage	24 V DC
Operating Input Voltage	20.4 to 26.4 V DC
Input Impedance	3.3 kΩ
Input Current	7 mA typical (at 24 V DC)
ON Voltage/ON Current	14.4 V DC min./3 mA min.
OFF Voltage/OFF Current	5 V DC max./1 mA max.
ON Response Time	8.0 ms max. (Possible to set to between 0 and 32 ms in the PLC Setup.)
OFF Response Time	8.0 ms max. (Possible to set to between 0 and 32 ms using PLC)
No. of Circuits	16 (8 points/common, 2 circuits)
Number of Simultaneously ON Points	100% simultaneously ON
Insulation Resistance	20 MΩ between external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	100 mA max.
Weight	270 g max.

Circuit Configuration



Terminal Connections



Polarity of the input power supply can connected in either direction.

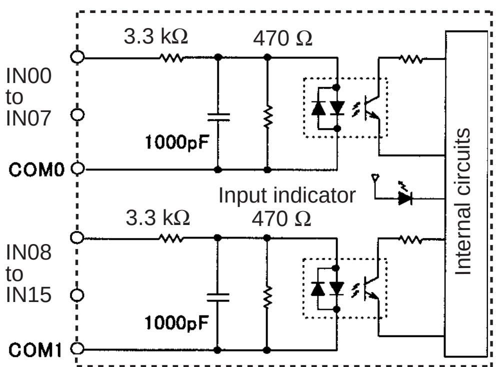
Note (1) The ON response time will be 20 µs maximum and OFF response time will be 300 µs maximum even if the response times are set to 0 ms due to internal element delays.

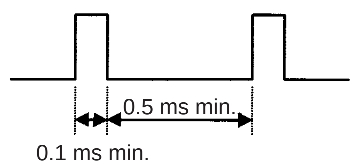
(2) Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on the Unit.

CS1W-INT01 Interrupt Input Unit (16 Points)

Rated Input Voltage	24 V DC
Operating Input Voltage	20.4 to 26.4 V DC
Input Impedance	3.3 kΩ
Input Current	7 mA typical (at 24 V DC)
ON Voltage/ON Current	14.4 V DC min./3 mA min.
OFF Voltage/OFF Current	5 V DC max./1 mA max.
ON Response Time	0.1 ms max.
OFF Response Time	0.5 ms max.
No. of Circuits	16 (8 points/common, 2 circuits)
Number of Simultaneously ON Points	100% simultaneously ON
Insulation Resistance	20 MΩ between external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	100 mA max.
Weight	270 g max.

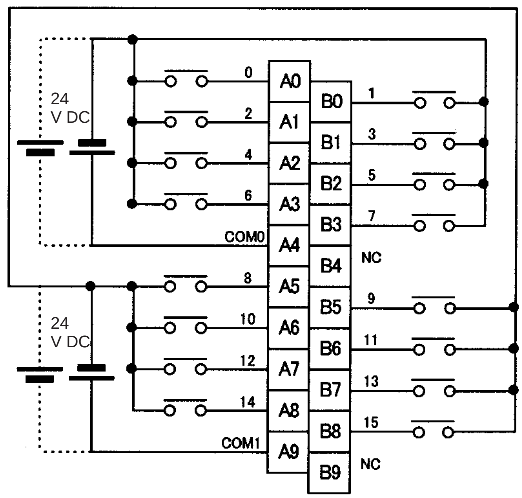
Circuit Configuration





- Up to two Interrupt Input Units can be mounted to the CPU Rack.
- Interrupts cannot be used when an Interrupt Input Unit is mounted to an Expansion I/O Rack, i.e., it will be treated as a 16-point Input Unit.
- Set the pulse width of signals input to the Interrupt Input Unit so they satisfy the above conditions.

Terminal Connections



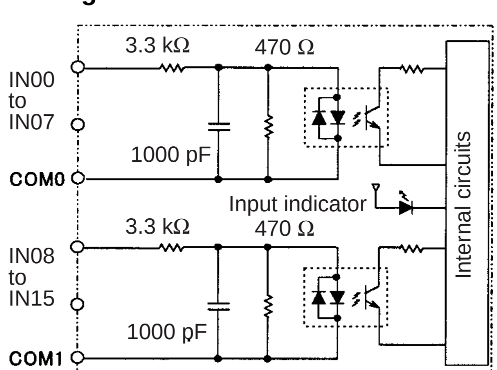
Polarity of the input power supply can connected in either direction.

Note Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on the Unit.

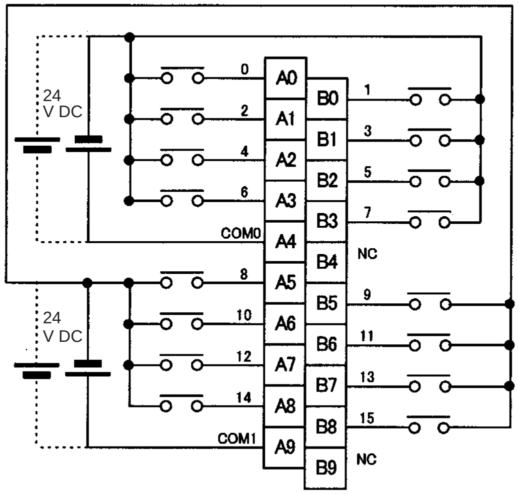
CS1W-IDP01 High-speed Input Unit (16 Points)

Rated Input Voltage	24 V DC
Operating Input Voltage	20.4 to 26.4 V DC
Input Impedance	3.3 kΩ
Input Current	7 mA typical (at 24 V DC)
ON Voltage/ON Current	14.4 V DC min./3 mA min.
OFF Voltage/OFF Current	5 V DC max./1 mA max.
ON Response Time	0.1 ms max.
OFF Response Time	0.5 ms max.
No. of Circuits	16 (8 points/common, 2 circuits)
Number of Simultaneously ON Points	100% simultaneously ON
Insulation Resistance	20 MΩ between external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	100 mA max.
Weight	270 g max.

Circuit Configuration



Terminal Connections



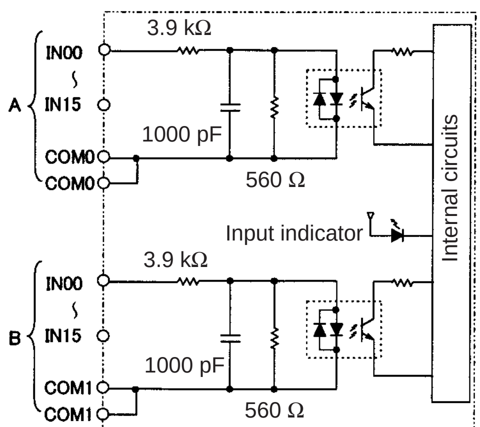
*1 Polarity of the input power supply can be connected in either direction.
*2 Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed the Unit.

- With a High-speed Input Unit, pulse inputs shorter than the cycle time of the CPU Unit can be read.
- The minimum pulse width (ON time) that can be read by the High-speed Input Unit is 0.1 ms.
- Input data in the internal circuits is cleared during the input refresh period.

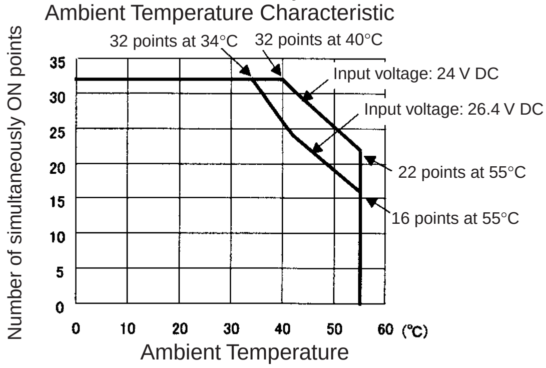
CS1W-ID231 DC Input Unit (32 Points)

Rated Input Voltage	24 V DC
Operating Input Voltage	20.4 to 26.4 V DC
Input Impedance	3.9 kΩ
Input Current	6 mA typical (at 24 V DC)
ON Voltage/ON Current	15.4 V DC min./3 mA min.
OFF Voltage/OFF Current	5 V DC max./1 mA max.
ON Response Time	8.0 ms max. (Can be set to between 0 and 32 in the PLC Setup.)
OFF Response Time	8.0 ms max. (Can be set to between 0 and 32 in the PLC Setup)
No. of Circuits	32 (16 points/common, 2 circuits)
Number of Simultaneously ON Points	70% (11 points/common) (at 24 V DC) (Refer to the following illustrations.)
Insulation Resistance	20 MΩ between external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	150 mA max.
Weight	200 g max.
Accessories	One connector for external wiring (soldered)

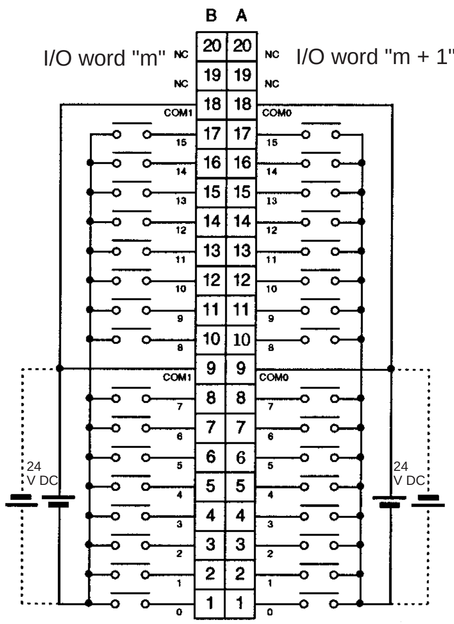
Circuit Configuration



Number of Simultaneously ON Points vs. Ambient Temperature Characteristic



Terminal Connections



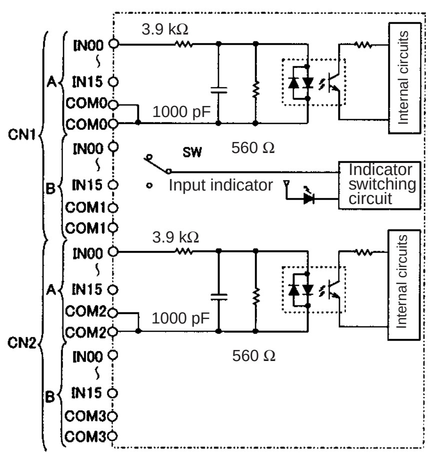
- The input power polarity can be connected in either direction provided that the same polarity is set for rows A and B.
- Both COM0 and COM1 have two pins each. Although they are internally connected, wire all points completely.

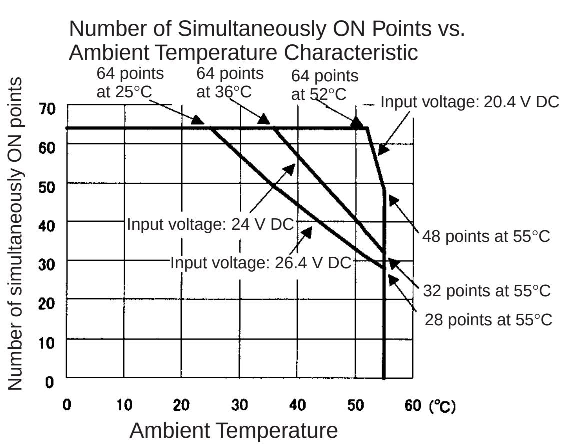
Note The ON response time will be 20 µs maximum and OFF response time will be 300 µs maximum even if the response times are set to 0 ms due to internal element delays.

CS1W-ID261 DC Input Unit (64 Points)

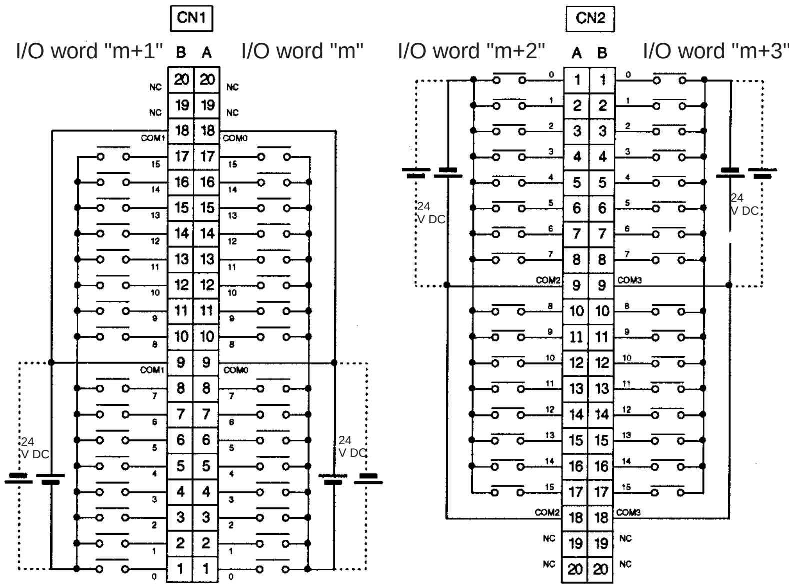
Rated Input Voltage	24 V DC
Operating Input Voltage	20.4 to 26.4 V DC
Input Impedance	3.9 kΩ
Input Current	6 mA typical (at 24 V DC)
ON Voltage/ON Current	15.4 V DC min./3 mA min.
OFF Voltage/OFF Current	5 V DC max./1 mA max.
ON Response Time	8.0 ms max. (Can be set to between 0 and 32 in the PLC Setup.)
OFF Response Time	8.0 ms max. (Can be set to between 0 and 32 in the PLC Setup.)
No. of Circuits	64 (16 points/common, 4 circuits)
Number of Simultaneously ON Points	50% (8 points/common) (at 24 V DC) (Refer to the following illustrations.)
Insulation Resistance	20 MΩ between external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	150 mA max.
Weight	260 g max.
Accessories	Two connectors for external wiring (soldered)

Circuit Configuration





Terminal Connections



- The input power polarity can be connected in either direction provided that the same polarity be set for rows A and B.
- COM0, COM1, COM2, and COM3 have two pins each. Although they are internally connected, wire all points completely.

Note The ON response time will be 20 µs maximum and OFF response time will be 300 µs maximum even if the response times are set to 0 ms due to internal element delays.

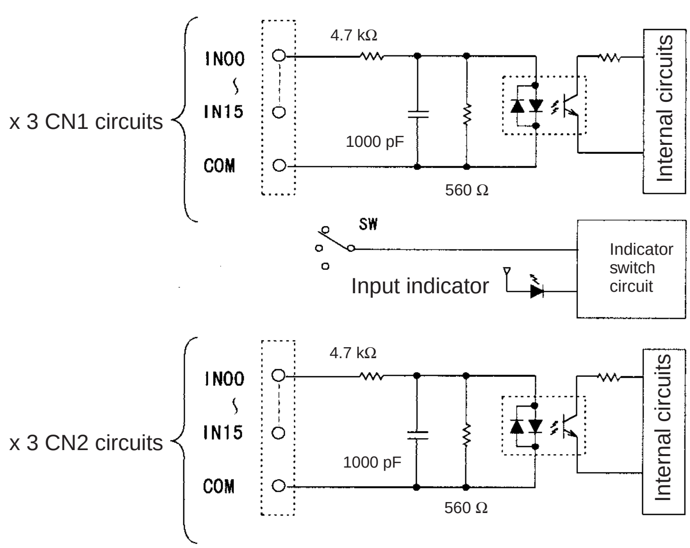
CS-series High-density I/O Units

CS1W-ID291 DC Input Unit (96 Points)

Rated Input Voltage	24 V DC
Operating Input Voltage	20.4 to 26.4 V DC
Input Impedance	4.7 kΩ
Input Current	Approx. 5 mA (at 24 V DC)
ON Voltage/ON Current	17 V DC min./3 mA min.
OFF Voltage/OFF Current	5 V DC max./1 mA max.
ON Response Time	8.0 ms max.(Possible to select one out of eight times from 0 to 32 ms in the PLC Setup.) (See note.)
OFF Response Time	8.0 ms max. (Possible to select one out of eight times from 0 to 32 ms in the PLC Setup.) (See note.)
No. of Circuits	6 (16 points/common)
Number of Inputs ON Simulta-neously	50% (8 points/common) (at 24 V DC) (Depends on ambient temperature)
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	200 mA max.
Weight	320 g max.
Accessories	Two connectors for external wiring (soldered)

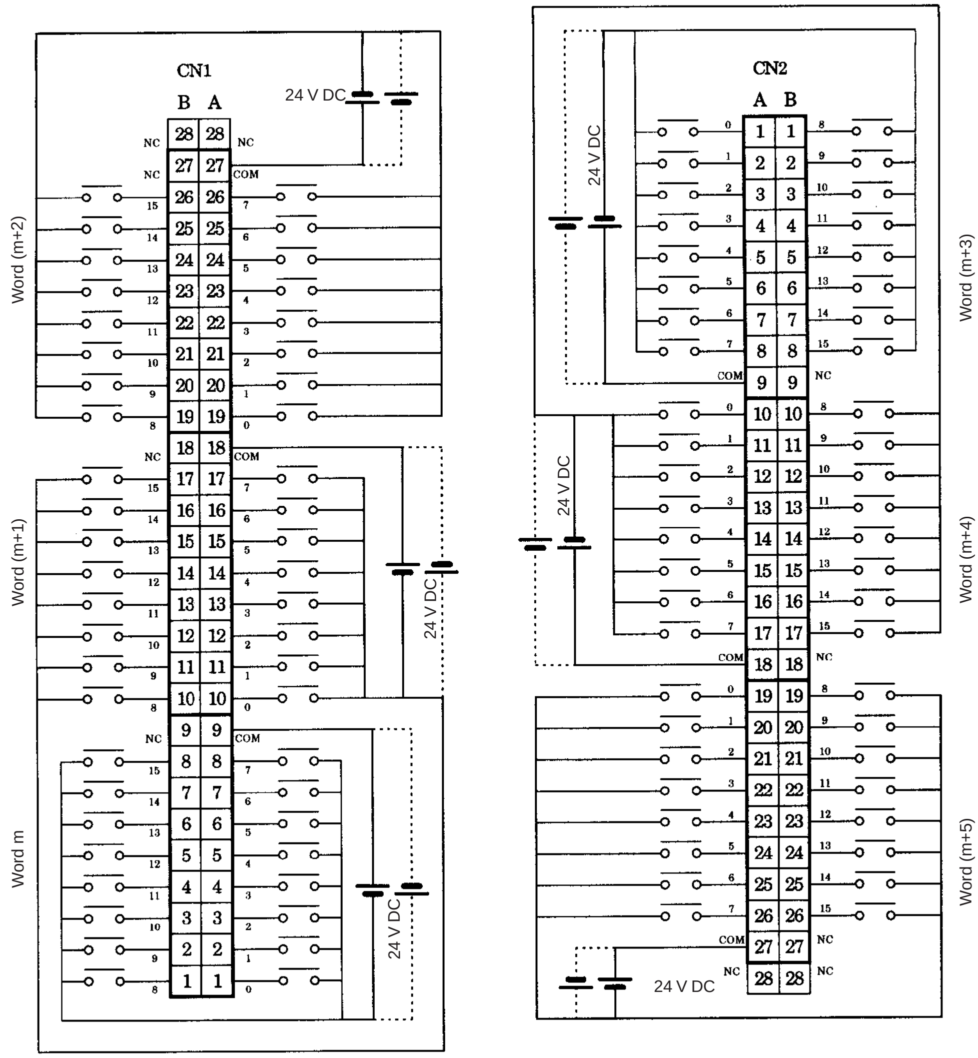
Circuit Configuration

The ON response time will be 20 µs maximum and OFF response time will be 300 µs maximum even if the response times are set to 0 ms due to internal element delays (See note below.)



Note The Input ON and OFF response times for Basic I/O Units can be set to 0 ms, 0.5 ms, 1 ms, 2 ms, 4 ms, 8 ms, 16 ms, or 32 ms in the PLC Setup.

Terminal Connections: CS1W-ID291 24-V DC 96-point Input Unit

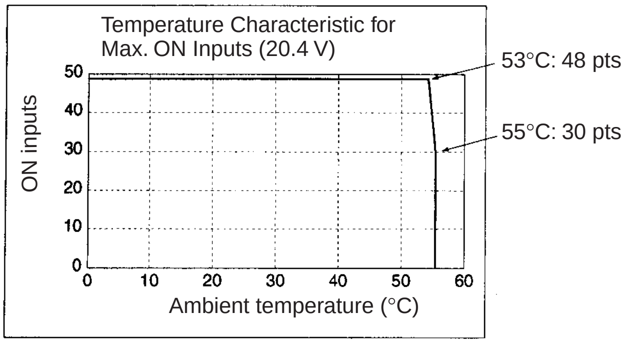


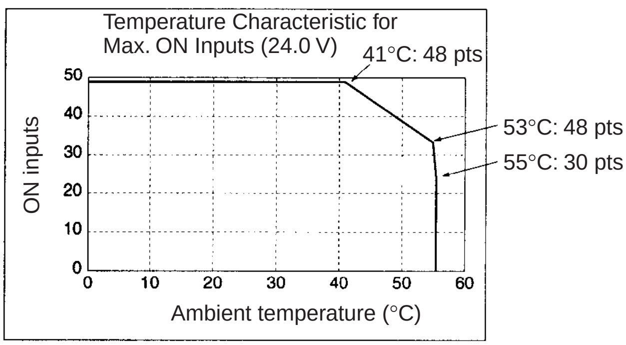
• The polarity of the input power supply can be in either direction, as indicated by the dotted lines.

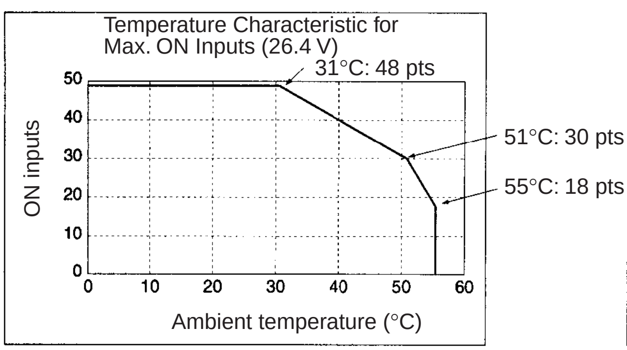
Maximum Number of ON Inputs

The maximum number of 24-V DC inputs that can be ON simultaneously for the CS1W-ID291/MD291/MD292 depends on the ambient temperature, as shown in the following diagrams.

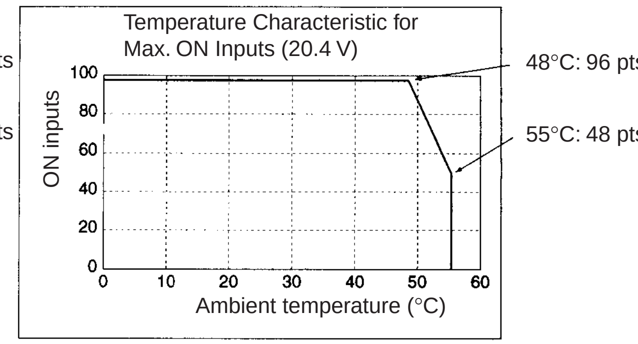
CS1W-MD291/MD292 Inputs

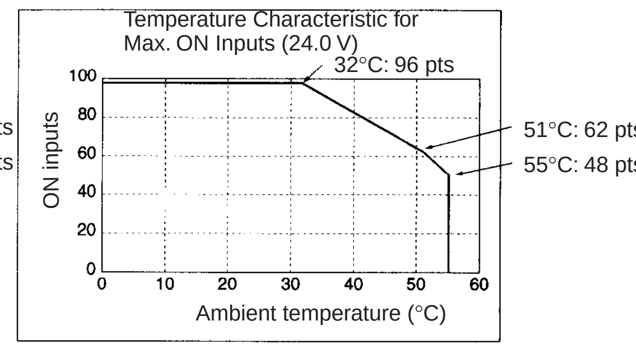


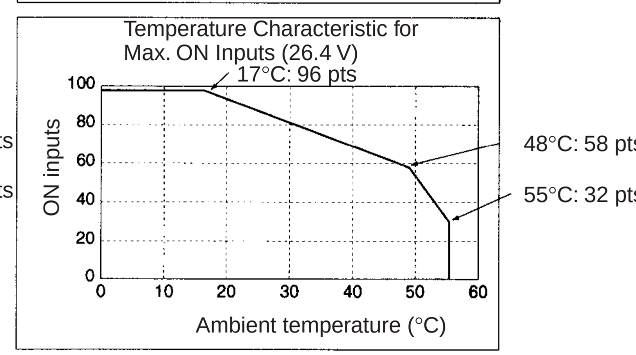




CS1W-ID291 Inputs





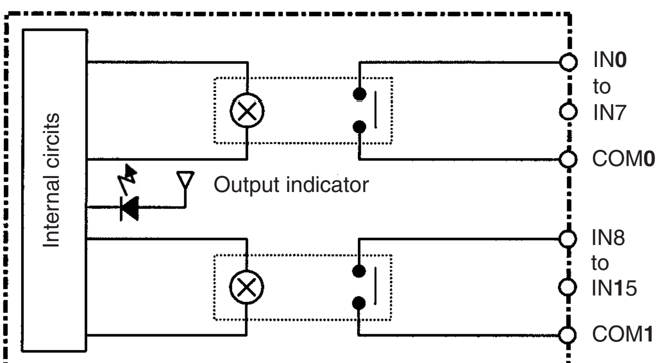


If the maximum number of ON points is exceeded for the CS1W-ID291/MD291/MD292, heat generated by electronic elements will increase the temperature of the electronic elements and the interior of the Unit. This will reduce the reliability and life of the electronic elements and cause Unit malfunctions. There will be a delay in the temperature increase, however, and there will be no problems if all inputs are ON for 10 minutes or less at the start of operations or any other time that all inputs have been off for at least 2 hours.

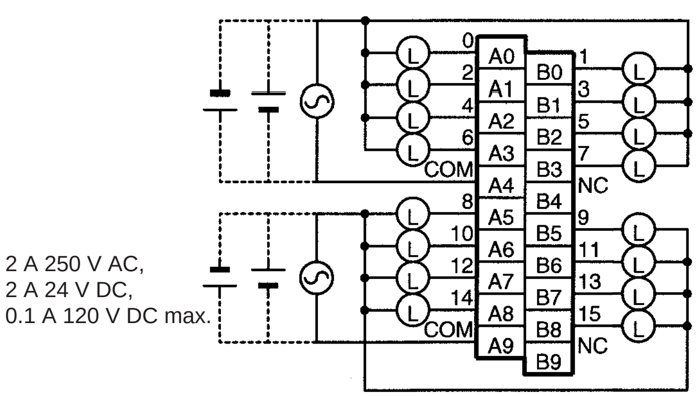
CS1W-OC211 Contact Output Unit (16 points)

Max. Switching Capacity	2 A 250 V AC (cosφ = 1), 2 A 24 V DC (8 A/common, 16 A/Unit), 0.1 A 120 V DC
Min. Switching Capacity	1 mA 5 V DC
Service Life of Relay	Electrical: 150,000 operations (resistive load)/ 100,000 operations (inductive load) Mechanical: 20,000,000 operations Service life will vary depending on the connected load. Refer to page 507 for information on service life according to the load.
Relay replacement	NY-24W-K-IE (Fujitsu Takamizawa Component Ltd.) Relays cannot be replaced by users.
ON Response Time	15 ms max.
OFF Response Time	15 ms max.
No. of Circuits	2 (each 8 points/common)
Number of Inputs ON Simultaneously	16
Surge Protector	None
Fuses	None
Insulation Resistance	20 MΩ between external terminals and the GR terminal (500 V DC)
Dielectric Strength	2,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	130 mA 5 V DC max. 96 mA 26 V DC (6 mA × No. points ON)
Weight	290 g max.

Circuit Configuration



Terminal Connections

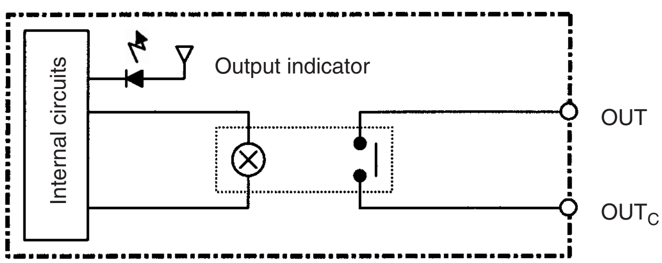


Note Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on the Unit.

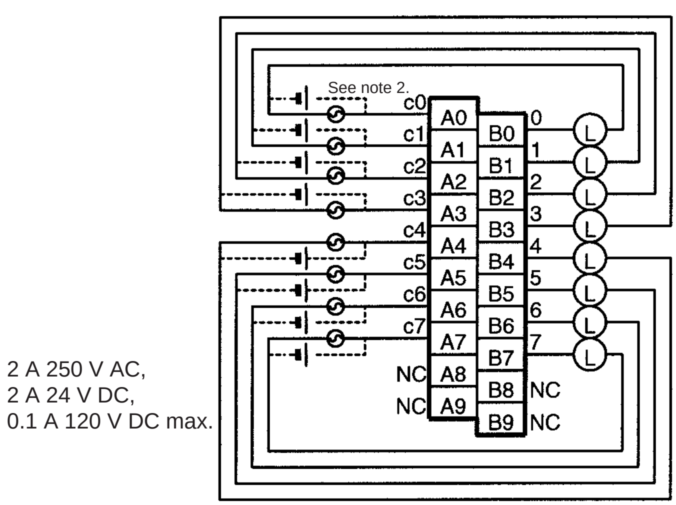
CS1W-OC201 Contact Output Unit (8 points) (See note 1.)

Max. Switching Capacity	2 A 250 V AC (cosϕ = 1), 2 A 24 V DC (16 A/Unit), 0.1 A 120 V DC
Min. Switching Capacity	1 mA 5 V DC
Service Life of Relay	Electrical: 150,000 operations (resistive load)/ 100,000 operations (inductive load) Mechanical: 20,000,000 operations Service life will vary depending on the connected load. Refer to page 507 for information on service life according to the load.
Relay replacement	NY-24W-K-IE (Fujitsu Takamizawa Component Ltd.) Relays cannot be replaced by users.
ON Response Time	15 ms max.
OFF Response Time	15 ms max.
No. of Circuits	8 independent contacts
Number of Inputs ON Simultaneously	8
Surge Protector	None
Fuses	None
Insulation Resistance	20 MΩ between external terminals and the GR terminal (500 V DC)
Dielectric Strength	2,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	100 mA 5 V DC max. 48 mA 26 V DC (6 mA × No. points ON)
Weight	270 g max.

Circuit Configuration



Terminal Connections



Note: There are no restrictions in the polarity when connecting a DC power supply.

Note (1) This Unit uses only 8 bits for external I/O even though 16 bits (1 word) are allocated. This Unit is also treated as a 16-point Unit in the I/O tables.

(2) Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on the Unit.

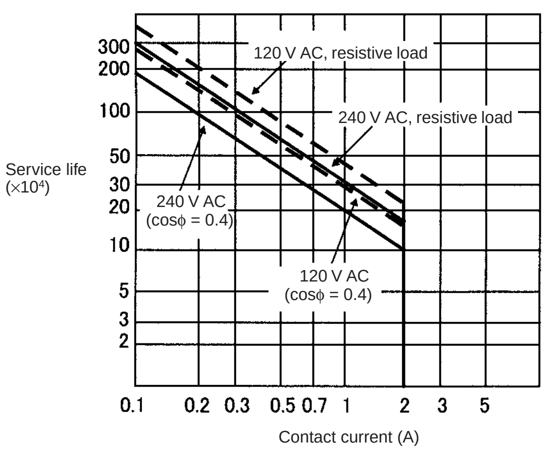
About Contact Output Units

Life Expectancy of CS1W-OC201/211 Relays

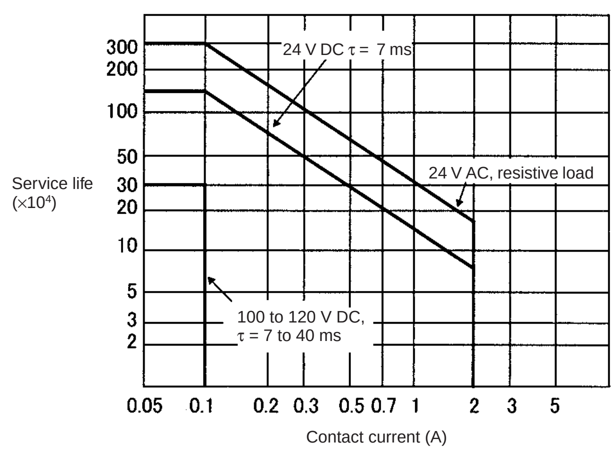
The life expectancy of the CS1W-OC201/211 Contact Output Unit is shown in the following diagrams. Use the diagrams to calculate the relay service life based on the operating conditions, and replace the relay before the end of its service life.

Note The diagrams show the life expectancy of the relay itself. Do not use a contact current, therefore, that exceeds the maximum switching capacity specified in the specifications for each Contact Output Unit. If a switching capacity exceeding the specifications is used, the reliability and life expectancy of other parts will be reduced and the Unit may malfunction.

Service Life with AC Load

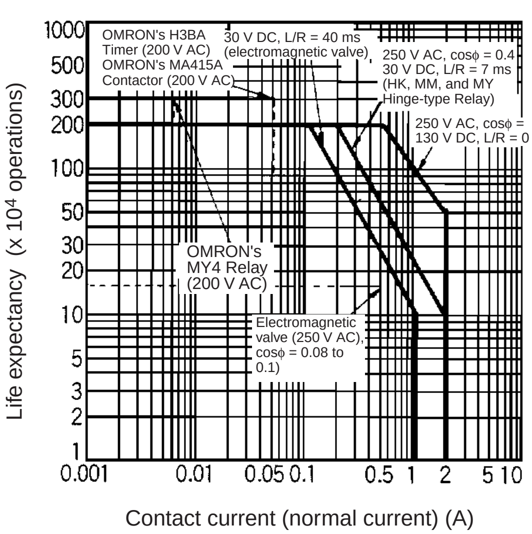


Service Life with DC Load

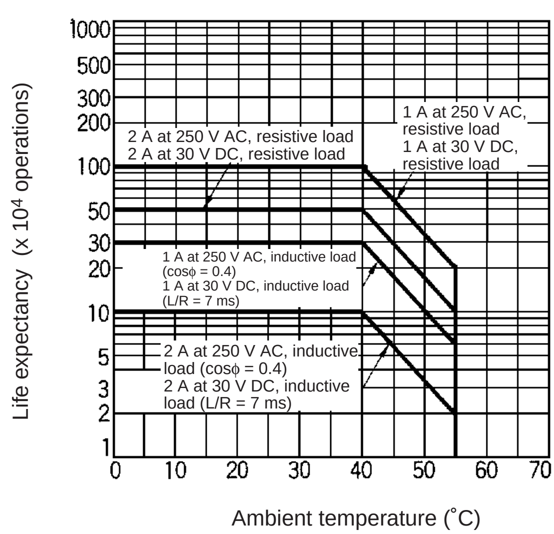


Max. switching frequency: 1,800 times/h

Contact Current vs. Life Expectancy
Conditions
Switching frequency: 1,800 times/hour max.
Ambient temperature: 23°C



Ambient Temperature vs. Life Expectancy
Conditions
Switching frequency: 1,800 times/hour max.



Note (1) If the Contact Output Unit is panel-mounted, the temperature inside the panel represents the ambient temperature.

(2) The life of the Relay at an ambient temperature of 55°C is one-fifth the life of the Relay at room temperature (0° to 40°C).

Inductive Load

The life of the Relay varies with the load inductance. If any inductive load is connected to the Contact Output Unit, use an arc killer with the Contact Output Unit using an inductive load.

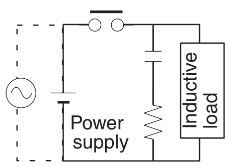
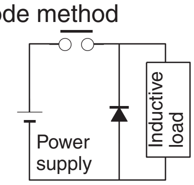
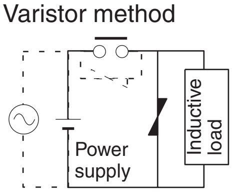
Be sure to connect a diode in parallel with every DC inductive load that is connected to the Contact Output Unit.

Contact Protection Circuit

Arc killers are used with the Contact Output Unit in order to prolong the life of each Relay mounted to the Contact Output Unit, prevent noise, and reduce the generation of carbide and nitrate deposits. Arc killers can, however, reduce relay life if not use correctly.

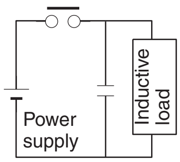
Note Arc killers used with the Contact Output Unit can delay the resetting time required by each Relay mounted to the Contact Output Unit.

Arc killer circuit examples are listed in the following table.

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

Note Do not connect a capacitor as an arc killer in parallel with an inductive load as shown in the following diagram. This arc killer is very effective for preventing spark discharge at the moment when the circuit is opened. However when the contacts are closed, the contacts may be welded due to the current charged in the capacitor.

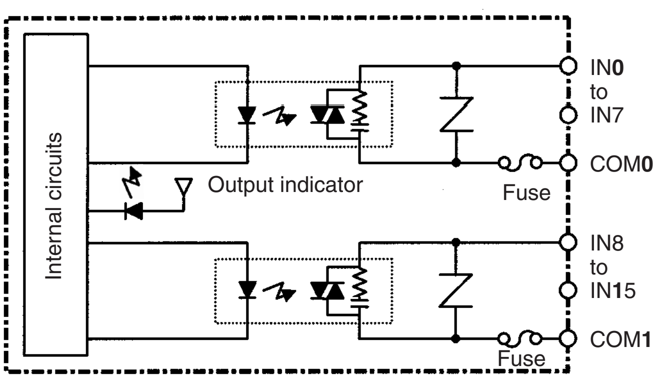
DC inductive loads can be more difficult to switch than resistive loads. If appropriate arc killers are used, however, DC inductive loads will be as easy to switch as resistive loads.



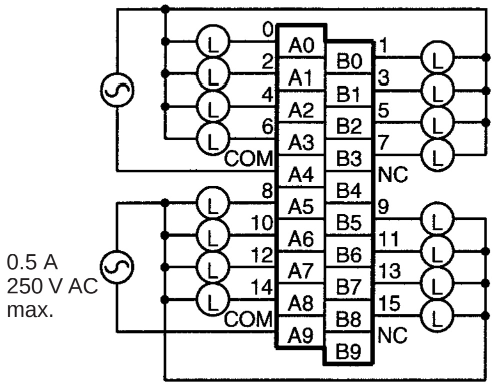
CS1W-OA211 Triac Output Unit (16 Points)

Max. Switching Capacity	0.5 A 250 V AC, 50/60 Hz (2 A/common, 4 A/Unit)
Max. Inrush Current	15 A (pulse width: 10 ms)
Min. Switching Capacity	50 mA 75 V AC
Leakage Current	1.5 mA (200 V AC) max.
Residual Voltage	1.6 V AC max.
ON Response Time	1 ms max.
OFF Response Time	1/2 of load frequency+1 ms or less.
No. of Circuits	2 (8 points/common)
Surge Protector	C.R Absorber + Surge Absorber
Fuses	2 × 4 A (1 per common) The fuse cannot be replaced by the user.
Blown Fuse Detection Circuit	None
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (500 V DC)
Dielectric Strength	2,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	406 mA 5 V DC max. (70 mA + 21 mA × No. of ON points)
Weight	300 g max.

Circuit Configuration



Terminal Connections

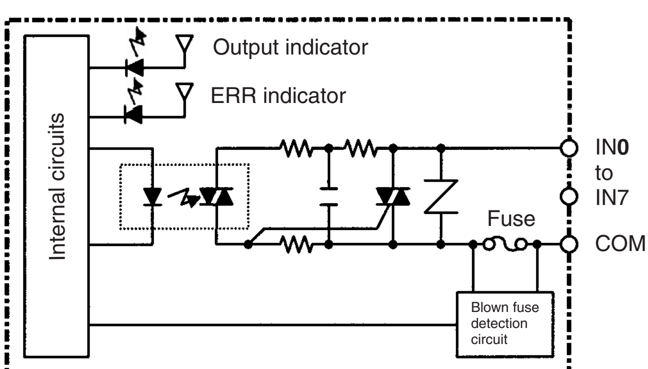


Note Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on the Unit.

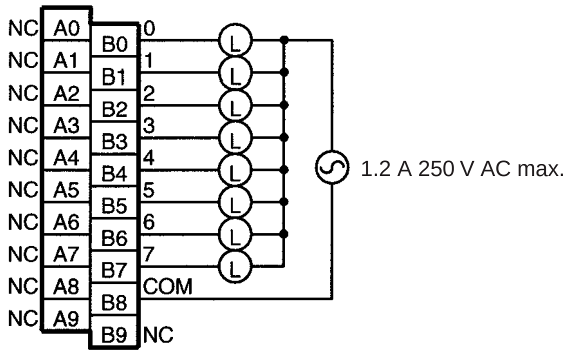
CS1W-OA201 Triac Output Unit (8 Points) (See note 1.)

Max. Switching Capacity	1.2 A 250 V AC, 50/60 Hz (4.8 A/Unit)
Max. Inrush Current	10 A (pulse width: 100 ms), 20 A (pulse width: 10 ms)
Min. Switching Capacity	100 mA 10 V AC, 50 mA 24 V AC, 10 mA 100 V AC min.
Leakage Current	1.5 mA (120 V AC) max., 3.0 mA (240 V AC) max.
Residual Voltage	1.5 V AC max. (50 to 500 mA), 5.0 V AC max. (10 to 50 mA)
ON Response Time	1 ms max.
OFF Response Time	1/2 of load frequency+1 ms or less.
No. of Circuits	1 (8 points/common)
Surge Protector	C.R Absorber + Surge Absorber
Fuses	8A The fuse cannot be replaced by the user.
Blown Fuse Detection Circuit	ERR indicator lit when fuse blown. Also, the corresponding Flag in the Basic I/O Unit Information Area (A050 to A089) will turn ON.
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (500 V DC)
Dielectric Strength	2,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	230 mA 5 V DC max. (70 mA + 20 mA × No. of ON points)
Weight	300 g max.

Circuit Configuration



Terminal Connections



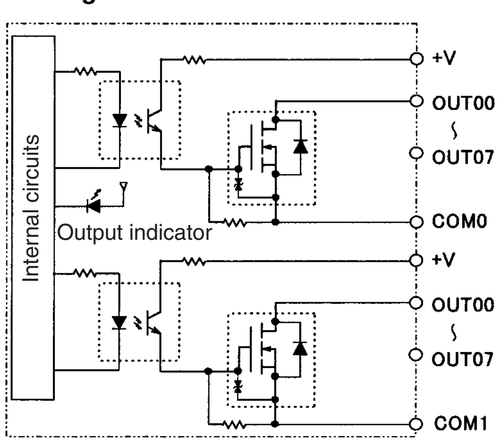
Note (1) This Unit uses only 8 bits for external I/O even though 16 bits (1 word) are allocated. This Unit is also treated as a 16-point Unit in the I/O tables.

(2) Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on the Unit.

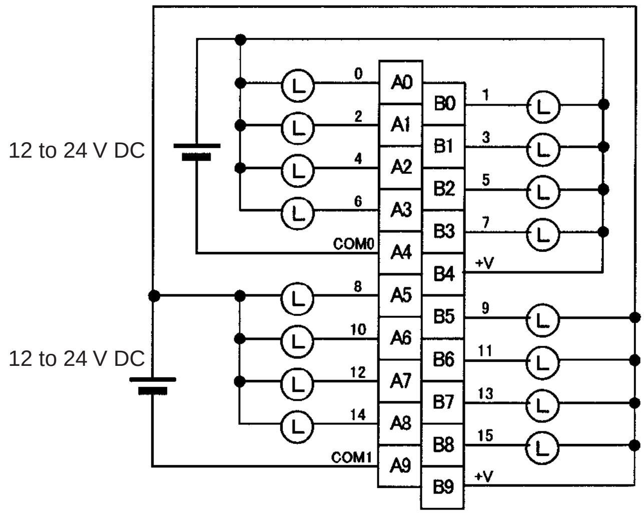
CS1W-OD211 Transistor Output Unit (16 Points, Sinking)

Rated Voltage	12 to 24 V DC
Operating Load Voltage Range	10.2 to 26.4 V DC
Maximum Load Current	0.5 A/point, 4.0 A/common, 8.0 A/Unit
Maximum Inrush Current	4.0 A/point, 10 ms max.
Leakage Current	0.1 mA max.
Residual Voltage	1.5 V max.
ON Response Time	0.5 ms max.
OFF Response Time	1.0 ms max.
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
No. of Circuits	16 (8 points/common, 2 circuits)
Internal Current Consumption	5 V DC 170 mA max.
Fuse	None
External Power Supply	10.2 to 26.4 V DC, 20 mA min.
Weight	270 g max.

Circuit Configuration



Terminal Connections



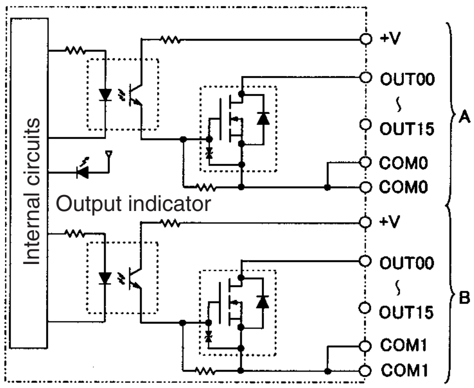
Note Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on the Unit. When wiring, pay careful attention to the polarity. The load may operate incorrectly if the polarity is reversed.

CS1W-OD231 Transistor Output Unit (32 Points, Sinking)

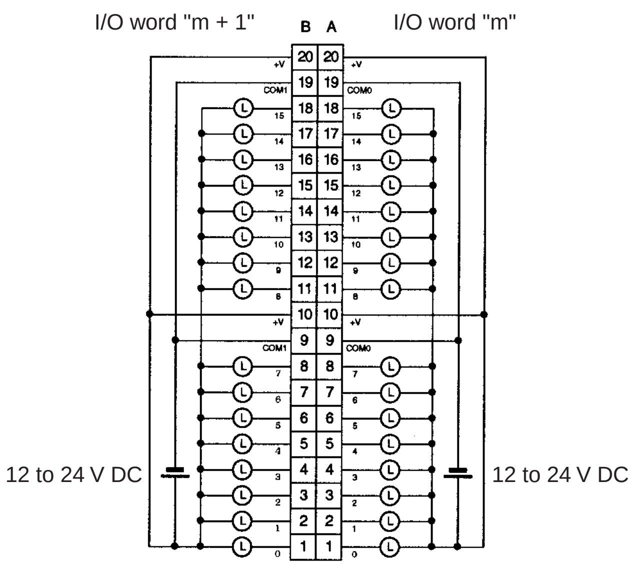
Rated Voltage	12 to 24 V DC
Operating Load Voltage Range	10.2 to 26.4 V DC
Maximum Load Current	0.5 A/point, 2.5 A/common, 5.0 A/Unit (See note.)
Maximum Inrush Current	4.0 A/point, 10 ms max.
Leakage Current	0.1 mA max.
Residual Voltage	1.5 V max.
ON Response Time	0.5 ms max.
OFF Response Time	1.0 ms max.
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
No. of Circuits	32 (16 points/common, 2 circuits)
Internal Current Consumption	5 V DC 270 mA max.
Fuse	None
External Power Supply	10.2 to 26.4 V DC, 30 mA min.
Weight	200 g max.
Accessories	One connector for external wiring (soldered)

Note The maximum load currents will be 2.0 A/common and 4.0 A/Unit if a pressure-welded connector is used.

Circuit Configuration



Terminal Connections

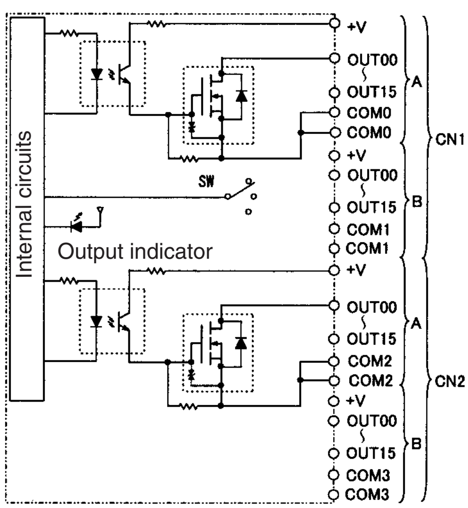


- When wiring, pay careful attention to the polarity. The load may operate if the polarity is reversed.
- Although the +V and COM terminals of rows A and B are internally connected, wire all points completely.

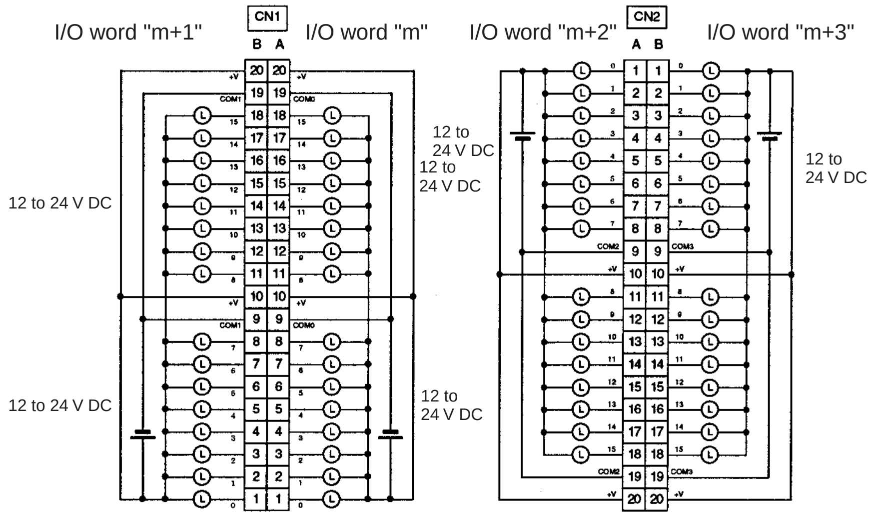
CS1W-OD261 Transistor Output Unit (64 Points, Sinking)

Rated Voltage	12 to 24 V DC
Operating Load Voltage Range	10.2 to 26.4 V DC
Maximum Load Current	0.3 A/point, 1.6 A/common, 6.4 A/Unit
Maximum Inrush Current	3.0 A/point, 10 ms max.
Leakage Current	0.1 mA max.
Residual Voltage	1.5 V max.
ON Response Time	0.5 ms max.
OFF Response Time	1.0 ms max.
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
No. of Circuits	64 (16 points/common, 4 circuits)
Internal Current Consumption	5 V DC 390 mA max.
Fuse	None
External Power Supply	10.2 to 26.4 V DC, 50 mA min.
Weight	260 g max.
Accessories	Two connectors for external wiring (soldered)

Circuit Configuration



Terminal Connections

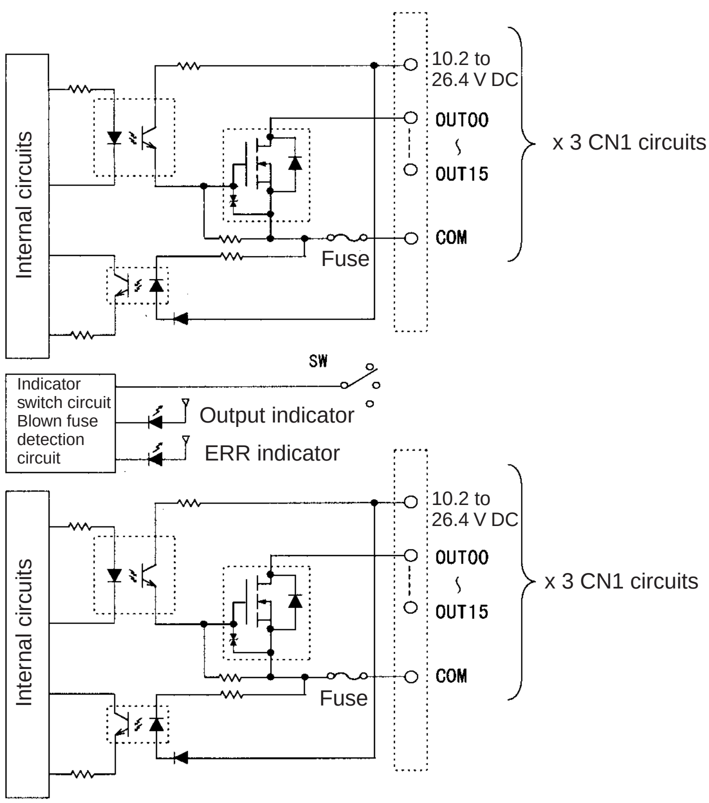


- When wiring, pay careful attention to the polarity. The load may operate if the polarity is reversed.
- Although the +V and COM terminals of rows A and B of CN1 and CN2 are internally connected, wire all points completely.

CS1W-OD291 Transistor Output Unit (96 Points, Sinking)

Rated Voltage	12 to 24 V DC
Operating Load Voltage	10.2 to 26.4 V DC
Maximum Load Current	0.1 A/point, 1.2 A/common, 7.2 A/Unit (See note 2.)
Maximum Inrush Current	1.0 A/point, 10 ms max. 8.0 A/common, 10 ms max.
Leakage Current	0.1 mA max.
Residual Voltage	1.5 V max.
ON Response Time	0.5 ms max.
OFF Response Time	1.0 ms max.
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
No. of Circuits	6 (16 points/common)
Internal Current Consumption	480 mA max. at 5 V DC
Fuse	3 A (1 per common) The fuse cannot be replaced by the user.
External Power Supply	10.2 to 26.4 V DC, 100 mA min.
Weight	320 g max.
Accessories	Two connectors for external wiring (soldered)

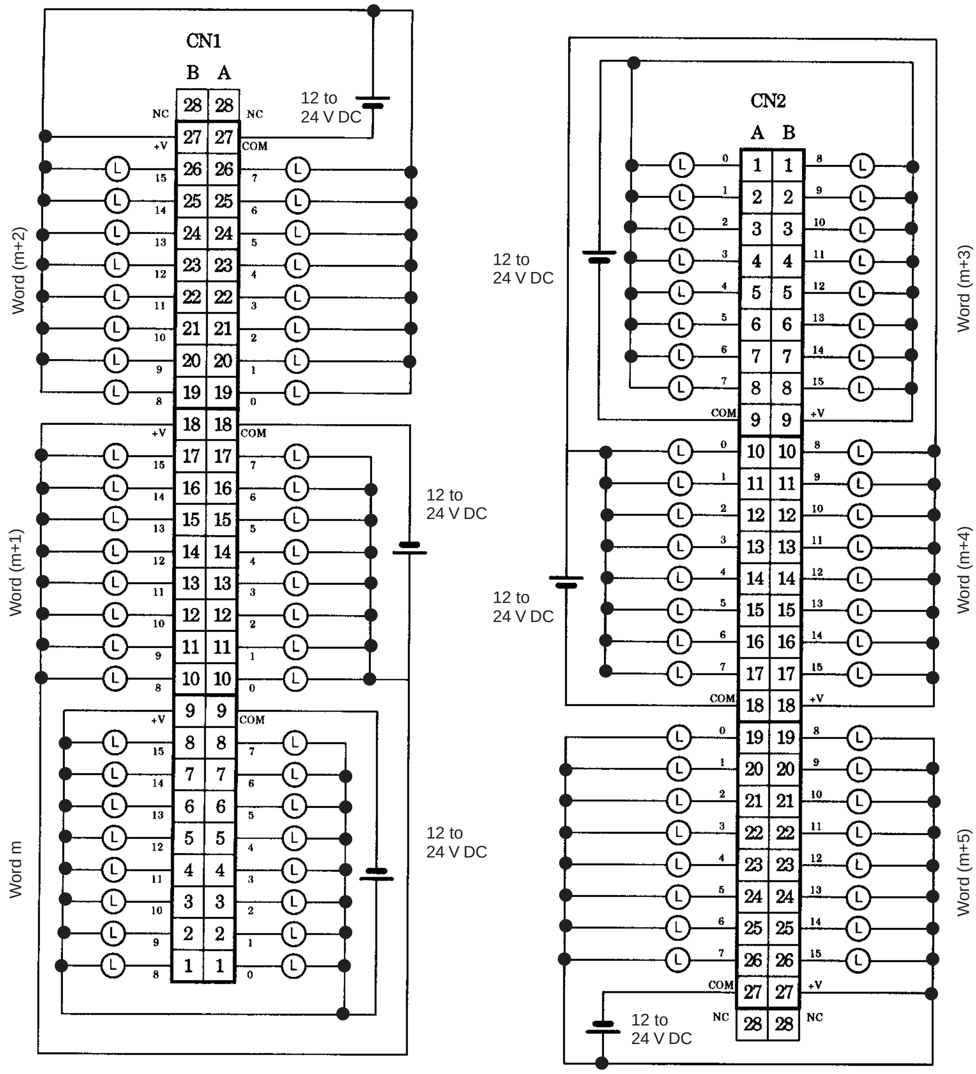
Circuit Configuration



Note (1) The ERR indicator will light if a fuse blows or if the external power supply is turned OFF, and the corresponding Flag in the Basic I/O Unit Information Area (A050 to A089) will turn ON.

(2) The maximum load currents will be 1.0 A/common and 6.0 A/Unit if a pressure-welded connector is used.

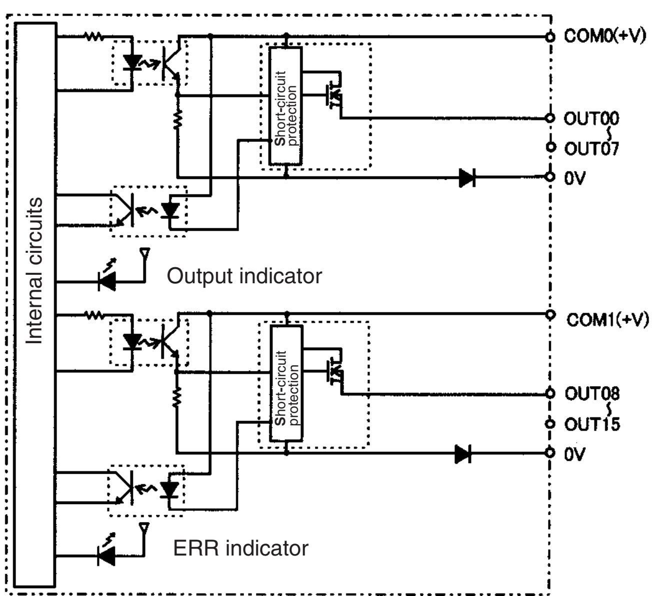
Terminal Connections: CS1W-OD291 24-V DC 96-point Transistor Output Unit (Sinking Outputs)



CS1W-OD212 Transistor Output Unit (16 Points, Sourcing)

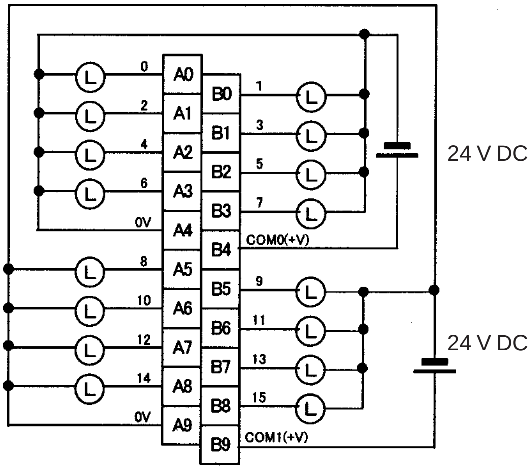
Rated Voltage	24 V DC
Operating Load Voltage Range	20.4 to 26.4 V DC
Maximum Load Current	0.5 A/point, 2.5 A/common, 5.0 A/Unit
Maximum Inrush Current	0.1 mA max.
Leakage Current	1.5 V max.
ON Response Time	0.5 ms max.
OFF Response Time	1.0 ms max.
Load Short-circuit Prevention	Detection current: 0.7 to 2.5 A Automatic restart after error clearance. (Refer to the following pages.)
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
No. of Circuits	16 (8 points/common, 2 circuits)
Internal Current Consumption	5 V DC 170 mA max.
External Power Supply	20.4 to 26.4 V DC, 40 mA min.
Weight	270 g max.

Circuit Configuration



When overcurrent is detected, the ERR indicator will light, and the corresponding flag in the Basic I/O Unit Information Area (A050 to A089) will turn ON.

Terminal Connections



Note Terminal numbers A0 to A9 and B0 to B9 are used in this manual, but they are not printed on the Unit.

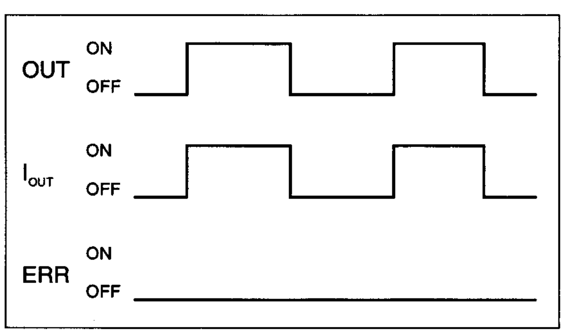
When wiring, pay careful attention to the polarity of the external power supply. The load may operate if the polarity is reversed.

Load Short-circuit Protection

This section describes the load short-circuit protection of the CS1W-OD212/OD232/OD262/MD262 Output Units.

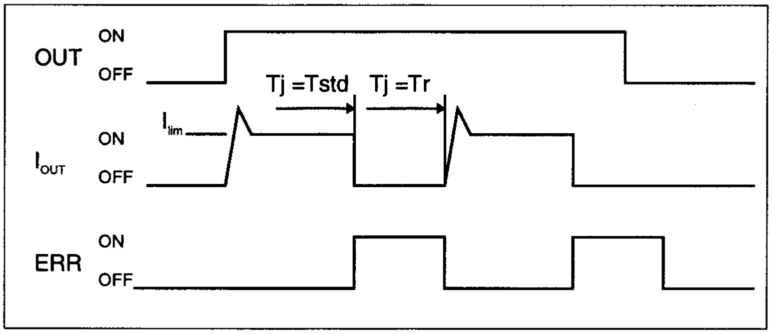
As shown below, normally when the output bit turns ON (OUT), the transistor will turn ON and then output current (Iout) will flow. If the output (Iout) is overloaded or short-circuited exceeding the detection current (Ilim), the output current (Iout) will be limited as shown in *Figure 2* below. When the junction temperature (Tj) of the output transistor reaches the thermal shutdown temperature (Tstd), the output will turn OFF to protect the transistor from being damaged, and the alarm output bit will turn ON to light the ERR indicator. When the junction temperature (Tj) of the transistor drops down to the reset temperature (Tr), the ERR indicator will be automatically reset and the output current will start flowing.

Figure 1: Normal Condition



OUT: OUTPUT instruction
I_{OUT}: Output current
ERR: Alarm output, ERR indicator
I_{lim}: Detection current
Tj: Junction temperature of transistor
Tstd: Thermal shutdown temperature
Tr: Reset temperature

Figure 2: Overload or Short-circuit



Operating Restrictions

Although the CS1W-OD212/OD232/OD262/MD262 are provided with short-circuit protection, these are for protecting internal circuits against momentary short-circuiting in the load. As shown in *Figure 2* below, the short-circuit protection is automatically released when the Tj equals to Tr. Therefore, unless the cause of short-circuit is removed, ON/OFF operations will be repeated in the output. Leaving short-circuits for any length of time will cause internal temperature rise, deterioration of elements, discoloration of the case or PCBs, etc. Therefore, observe the following restrictions.

Restrictions

If a short-circuit occurs in an external load, immediately turn OFF the corresponding output and remove the cause. The CS1W-OD212/OD232/OD262/MD262 turn ON an alarm output bit that corresponds to the external load output number. There is an alarm output bit for every common.

When an alarm output bit turns ON, use a self-holding bit for the alarm in the user program and turn OFF the corresponding output.

The alarm output bit is allocated in the Basic I/O Unit Information Area (A050 to A089) for every Unit mounting slot.

The following table shows the correspondence between output bits and bits in the Basic I/O Unit Information Area.

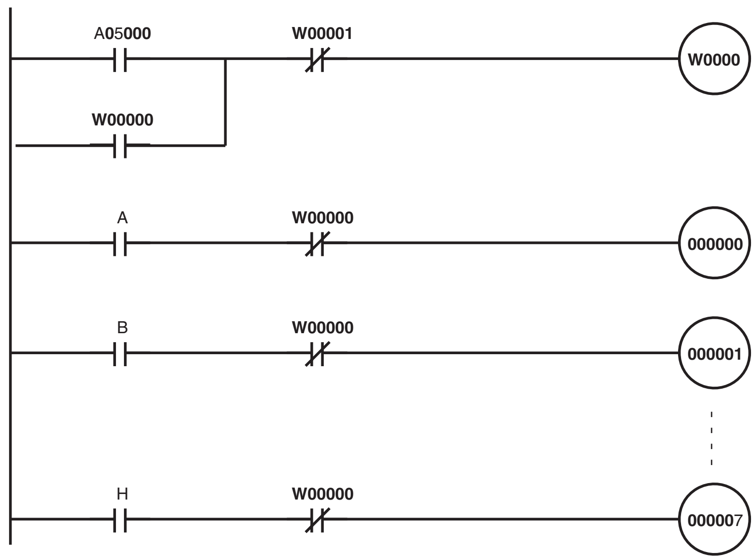
Output bit		m		m+1	m+2	m+3
		0 to 7	**8 to 15**	**0 to 15**	**0 to 15**	**0 to 15**
CS1W-OD212	Mounted in even slot	0	1	---	---	---
	Mounted in odd slot	8	9	---	---	---
CS1W-OD232	Mounted in even slot	0		1	---	---
	Mounted in odd slot	8		9	---	---
CS1W-OD262	Mounted in even slot	0		1	2	3
	Mounted in odd slot	8		9	10	11
CS1W-MD262	Mounted in even slot	0		1	---	---
	Mounted in odd slot	8		9	---	---

For example, when the CS1W-OD212 is mounted in slot 0 on Rack 0, A05001 will turn ON if the output 8 is short-circuited. When the CS1W-OD262 is mounted in slot 1 of Rack 0, A05011 will turn ON if the output m+3 is short-circuited

Programming Example

In this example, CS1W-OD212 is mounted in slot 0 of the Rack 0.

This example shows how to turn OFF output bits CIO 000000 to CIO 000007 immediately if the alarm output bit A05000 turns ON and how to keep the output bits OFF until the cause is removed and the bit is reset using work bit W000001.

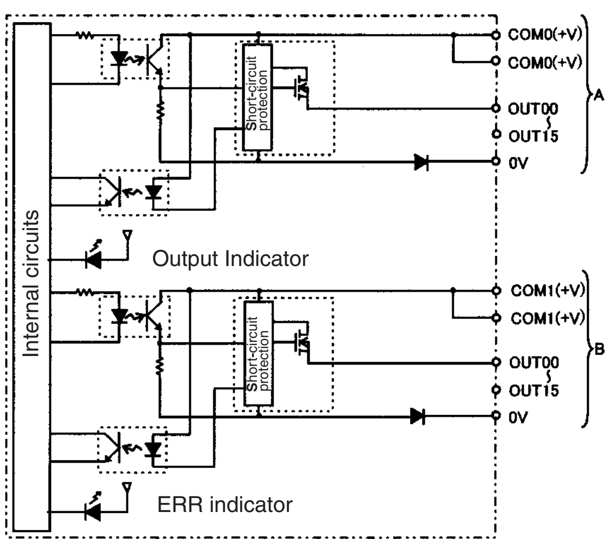


CS1W-OD232 (32 Points, Sourcing)

Rated Voltage	24 V DC
Operating Load Voltage Range	20.4 to 26.4 V DC
Maximum Load Current	0.5 A/point, 2.5 A/common, 5.0 A/Unit (See note.)
Leakage Current	0.1 mA max.
Residual Voltage	1.5 V max.
ON Response Time	0.5 ms max.
OFF Response Time	1.0 ms max.
Load Short-circuit Prevention	Detection current: 0.7 to 2.5 A Automatic restart after error clearance. (Refer to the above Load Short-circuit Protection.)
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
No. of Circuits	32 (16 points/common, 2 circuits)
Internal Current Consumption	5 V DC 270 mA max.
External Power Supply	20.4 to 26.4 V DC, 70 mA min.
Weight	210 g max.
Accessories	One connector for external wiring (soldered)

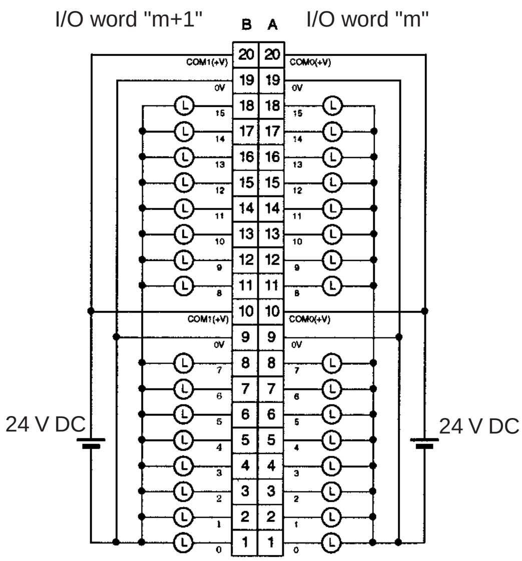
Note The maximum load currents will be 2.0 A/common and 4.0 A/Unit if a pressure-welded connector is used.

Circuit Configuration



When the output current of any output exceeds the detection current, the output for that point will turn OFF. At the same time, the ERR indicator will light and the corresponding flag (one for each common) in the Basic I/O Unit Information Area (A050 to A089) will turn ON.

Terminal Connections

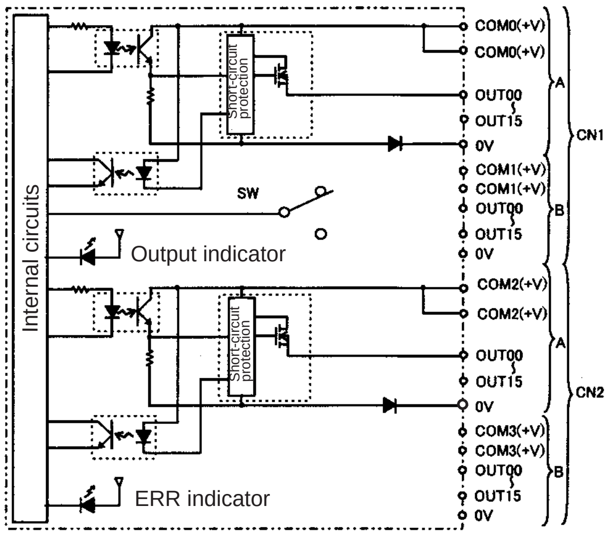


- When wiring, pay careful attention to the polarity of the external power supply. The load may operate if the polarity is reversed.
- Although the COM(+V) and 0V of rows A and B are internally connected, wire all points completely.

CS1W-OD262 (64 Points, Sourcing)

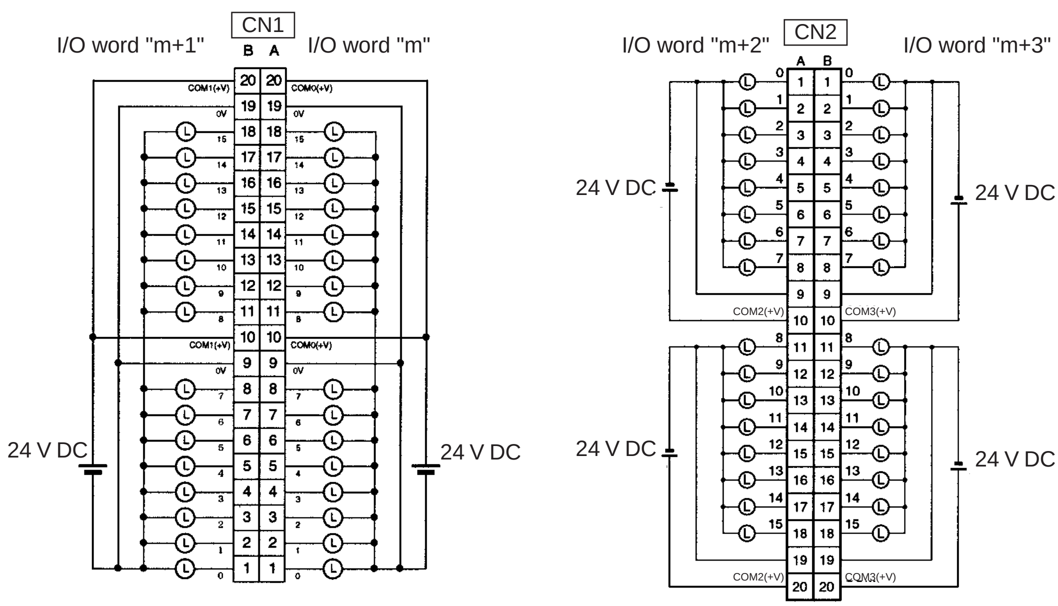
Rated Voltage	24 V DC
Operating Load Voltage Range	20.4 to 26.4 V DC
Maximum Load Current	0.3 A/point, 1.6 A/common, 6.4 A/Unit
Leakage Current	0.1 mA max.
Residual Voltage	1.5 V max.
ON Response Time	0.5 ms max.
OFF Response Time	1.0 ms max.
Load Short-circuit Prevention	Detection current: 0.7 to 2.5 A Automatic restart after error clearance. (Refer to the above Load Short-circuit Protection.)
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
No. of Circuits	64 (16 points/common, 4 circuits)
Internal Current Consumption	5 V DC 390 mA max.
External Power Supply	20.4 to 26.4 V DC, 130 mA min.
Weight	270 g max.
Accessories	Two connectors for external wiring (soldered)

Circuit Configuration



When overcurrent is detected, the ERR indicator will light. At the same time, the corresponding flag (one for each common) in the Basic I/O Unit Information Area (A050 to A089) will turn ON.

Terminal Connections



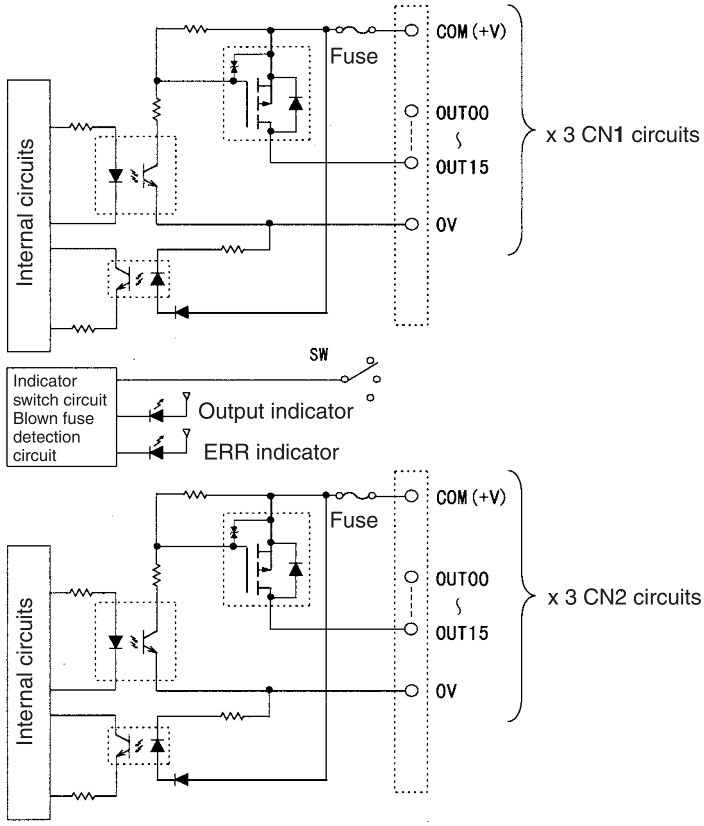
- When wiring, pay careful attention to the polarity of the external power supply. The load may operate if the polarity is reversed.
- Although the COM(+V) and 0V of rows A and B of CN1 and CN2 are internally connected, wire all points completely.

CS1W-OD292 Transistor Output Unit (96 Points, Sourcing)

Rated Voltage	12 to 24 V DC
Operating Load Voltage Range	10.2 to 26.4 V DC
Maximum Load Current	0.1 A/point, 1.2 A/common, 7.2 A/Unit (See note.)
Maximum Inrush Current	1.0 A/point, 10 ms max. 8.0 A/common, 10 ms max.
Leakage Current	0.1 mA max.
Residual Voltage	1.5 V max.
ON Response Time	0.5 ms max.
OFF Response Time	1.0 ms max.
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
No. of Circuits	6 (16 points/common)
Internal Current Consumption	480 mA max. at 5 V DC
Fuse	3 A (1 per common) The fuse cannot be replaced by the user.
External Power Supply	10.2 to 26.4 V DC, 100 mA min.
Weight	320 g max.
Accessories	Two connectors for external wiring (soldered)

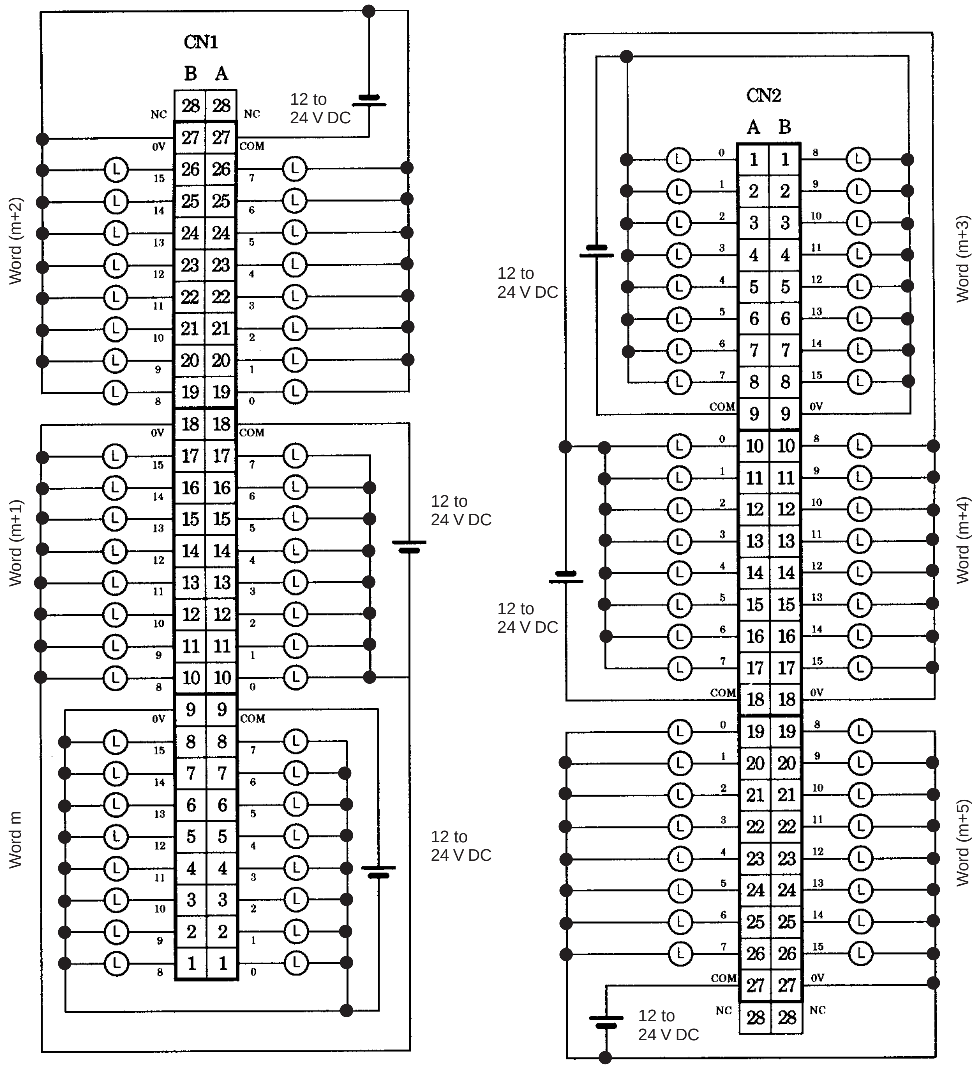
Note The maximum load currents will be 1.0 A/command and 6.0 A/Unit if a pressure-welded connector is used.

Circuit Configuration



Note The ERR indicator will light if a fuse blows or if the external power supply is turned OFF, and the corresponding Flag in the Basic I/O Unit Information Area (A050 to A089) will turn ON.

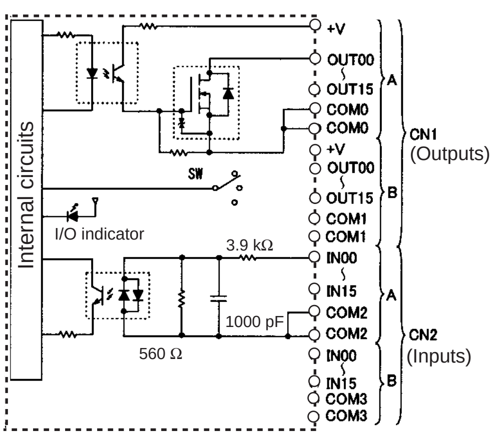
Terminal Connections: CS1W-OD292 24-V DC 96-point Transistor Output Unit (Sourcing Outputs)

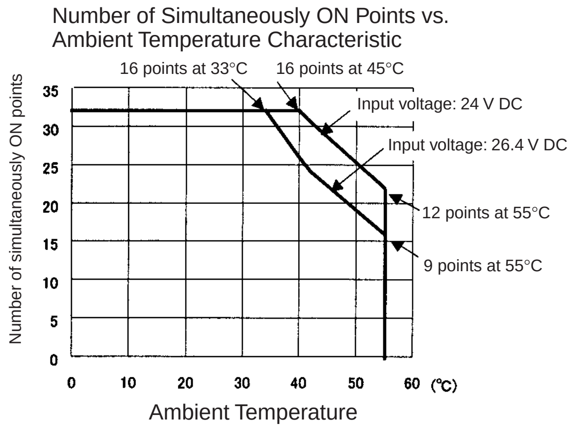


CS1W-MD261 DC Input/Transistor Output Unit (32/32 Points, Sinking)

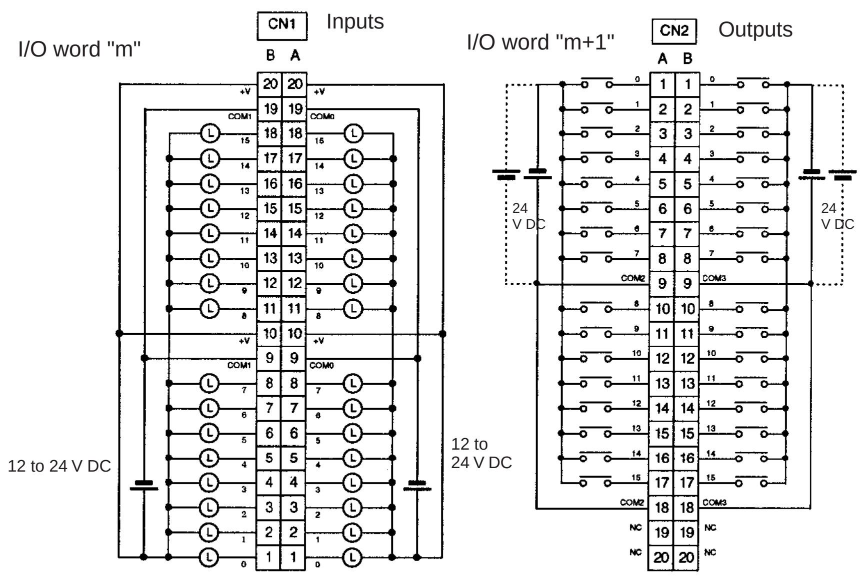
Output section (CN1)		Input section (CN2)	
Rated Voltage	12 to 24 V DC	Rated Input Voltage	24 V DC
		Operating Input Voltage	20.4 to 26.4 V DC
Operating Load Voltage Range	10.2 to 26.4 V DC	Input Impedance	3.9 kΩ
Maximum Load Current	0.3 A/point, 1.6 A/common, 3.2 A/Unit	Input Current	6 mA typical (at 24 V DC)
Maximum Inrush Current	3.0/point, 10 ms max.	ON Voltage/ON Current	15.4 V DC min./3 mA min.
Leakage Current	0.1 mA max.	OFF Voltage/OFF Current	5 V DC min./1 mA min.
Residual Voltage	1.5 V max.	ON Response Time	8.0 ms max. (Can be set to between 0 and 32 in the PLC Setup.) (See notes 1 and 2.)
ON Response Time	0.5 ms max.		
OFF Response Time	1.0 ms max.	OFF Response Time	8.0 ms max. (Can be set to between 0 and 32 in the PLC Setup.) (See notes 1 and 2.)
No. of Circuits	32 (16 points/common, 2 circuits)		
Fuse	None	No. of Circuits	32 (16 points/common, 2 circuits)
External Power Supply	10.2 to 26.4 V DC, 30 mA min.	Number of Simultaneously ON Points	70% (11 points/common) (at 24 V DC)
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)		
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.		
Internal Current Consumption	5 V DC 270 mA max.		
Weight	260 g max.		
Accessories	Two connectors for external wiring (soldered)		

Circuit Configuration





Terminal Connections



- When wiring, pay careful attention to the polarity. The load may operate is polarity is reversed.
- Although the +V and COM terminals of rows A and B of CN1 and CN2 are internally connected, wire all points completely.

Note (1) The ON response time will be 20 µs maximum and OFF response time will be 300 µs maximum even if the response times are set to 0 ms due to internal element delays.

(2) The input ON and OFF response times for Basic I/O Units can be set to 0, 0.5, 1, 2, 4, 8, 16, or 32 ms in the PLC Setup.

CS1W-MD291 DC Input/Transistor Output Unit (48/48 Points, Sinking)

Outputs (CN1)

Rated Voltage	12 to 24 V DC
Operating Load Voltage Range	10.2 to 26.4 V DC
Maximum Load Current	0.1 A/point, 1.2 A/common, 3.6 A/Unit (See note.)
Maximum Inrush Current	1.0 A/point, 10 ms max. 8.0 A/common, 10 ms max.
Leakage Current	0.1 mA max.
Residual Voltage	1.5 V max.
ON Response Time	0.5 ms max.
OFF Response Time	1.0 ms max.
No. of Circuits	48 points (16 points/common, 3 commons)
Fuse	3 A (1 per common) The fuse cannot be replaced by the user.
External Power Supply	10.2 to 26.4 V DC, 50 mA min.
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	350 mA max. at 5 V DC
Weight	320 g max.
Accessories	Two connectors for external wiring (soldered)

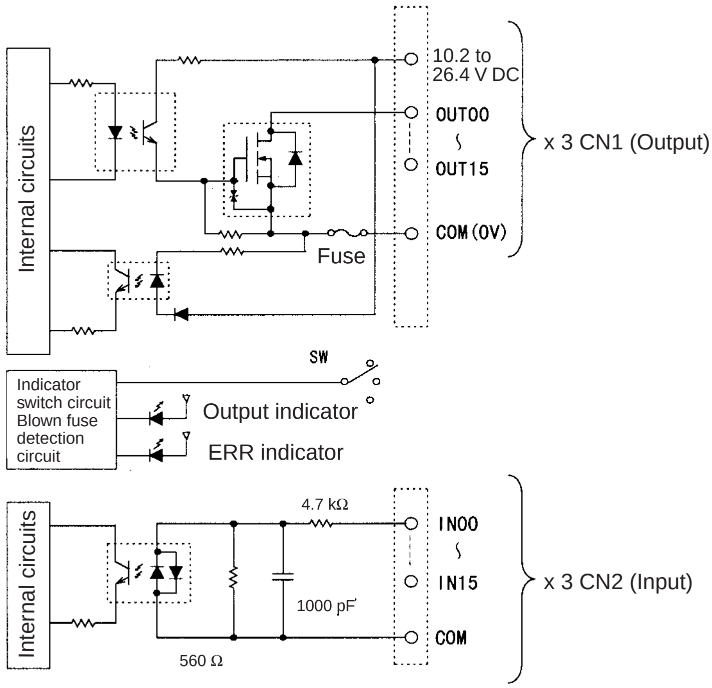
Note The maximum load currents will be 1.0 A/common and 3.0 A/Unit if a pressure-welded connector is used.

Inputs (CN2)

Rated Input Voltage	24 V DC
Operating Input Voltage	20.4 to 26.4 V DC
Input Impedance	4.7 kΩ
Input Current	Approx. 5 mA (at 24 V DC)
ON Voltage/ON Current	17 V DC min./3 mA min.
OFF Voltage/OFF Current	5.0 V DC max./1 mA max.
ON Response Time	8.0 ms max.(Possible to select one out of eight times from 0 to 32 ms in the PLC Setup.) (See note below.)
OFF Response Time	8.0 ms max. (Possible to select one out of eight times from 0 to 32 ms in the PLC Setup.) (See note below.)
No. of Circuits	48 points (16 points/common, 3 commons)
Number of Inputs Simultaneous ON	50% (8 points/common) (at 24 V DC) (Depends on ambient temperature.)
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	350 mA max. at 5 V DC
Weight	320 g max.
Accessories	Two connectors for external wiring (soldered)

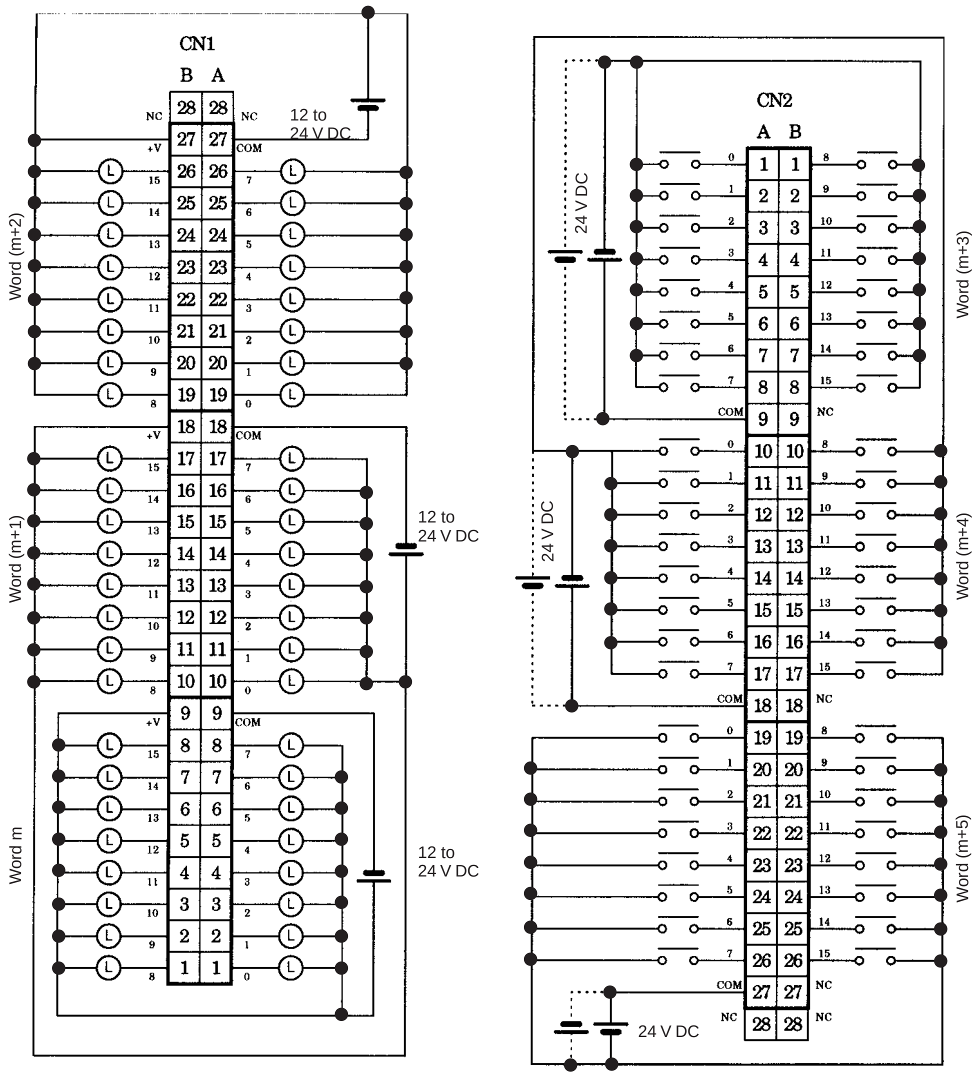
Note The input ON and OFF response times for Basic I/O Units can be set to 0 ms, 0.5 ms, 1 ms, 2 ms, 4 ms, 8 ms, 16 ms, or 32 ms in the PLC Setup.

Circuit Configuration



Note The ERR indicator will light if a fuse blows or if the external power supply is turned OFF, and the corresponding Flag in the Basic I/O Unit Information Area (A050 to A089) will turn ON.

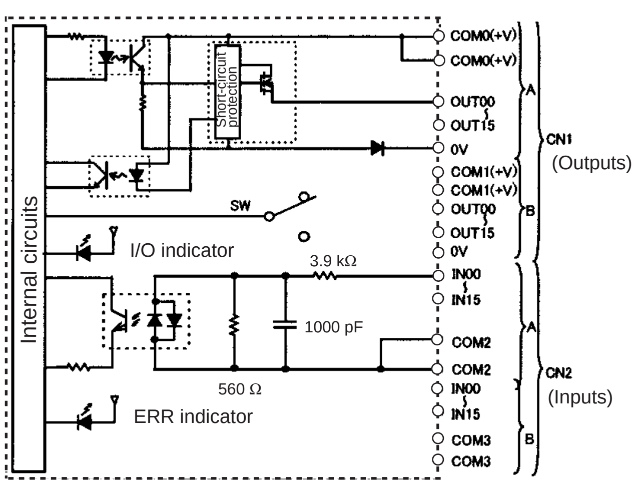
Terminal Connections: CS1W-MD291 24-V DC 48-point Input/48-point Output Unit (Sinking Outputs)



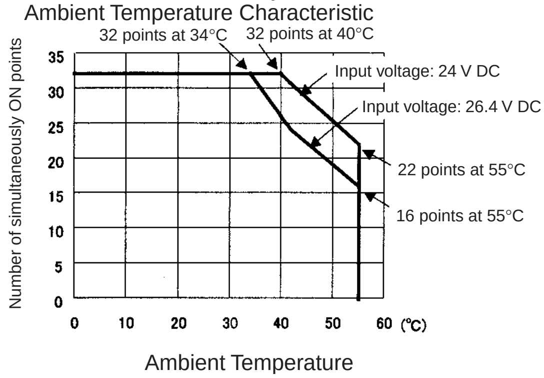
CS1W-MD262 DC Input/Transistor Output Unit (32/32 Points, Sourcing)

Output section (CN1)		Input section (CN2)	
Rated Voltage	24 V DC	Rated Input Voltage	24 V DC
		Operating Input Voltage	20.4 to 26.4 V DC
Operating Load Voltage Range	20.4 to 26.4 V DC	Input Impedance	3.9 kΩ
Maximum Load Current	0.3 A/point, 1.6 A/common, 3.2 A/Unit	Input Current	6 mA typical (at 24 V DC)
Leakage Current	0.1 mA max.	ON Voltage/ON Current	15.4 V DC min./3 mA min.
Residual Voltage	1.5 V max.	OFF Voltage/OFF Current	5 V DC min./1 mA min.
ON Response Time	0.5 ms max.	ON Response Time	8.0 ms max. (Can be set to between 0 and 32 in the PLC Setup.) (Refer to the above Load Short-circuit Protection.)
OFF Response Time	1.0 ms max.		
Load Short-circuit Prevention	Detection current: 0.7 to 2.5 A Automatic restart after error clearance. (Refer to the above *Load Short-circuit Protection*.)	OFF Response Time	8.0 ms max. (Can be set to between 0 and 32 in the PLC Setup.) (Refer to the above *Load Short-circuit Protection*.)
No. of Circuits	32 (16 points/common, 2 circuits)		
External Power Supply	20.4 to 26.4 V DC, 70 mA min.	Number of Simultaneously ON Points	70% (11 points/common) (at 24 V DC)
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)		
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.		
Internal Current Consumption	5 V DC 270 mA max.		
Weight	270 g max.		
Accessories	Two connectors for external wiring (soldered)		

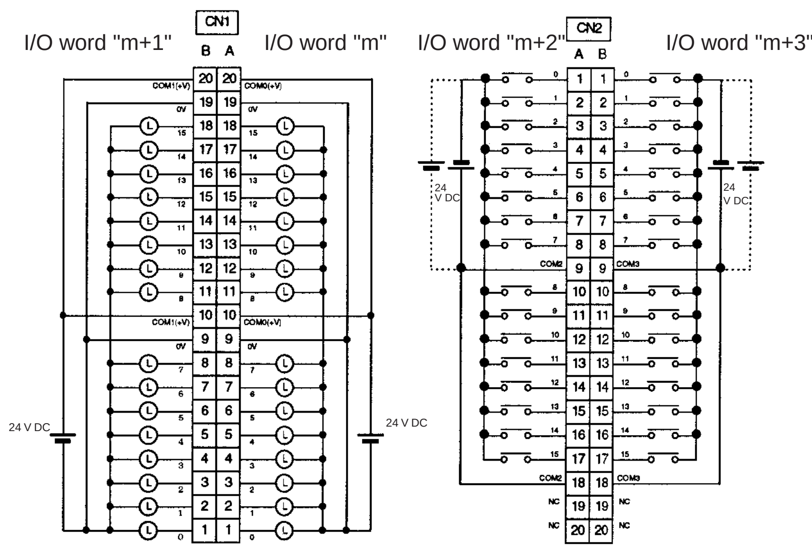
Circuit Configuration



Number of Simultaneously ON Points vs.
Ambient Temperature Characteristic



Terminal Connections



- When wiring, pay careful attention to the polarity. The load may operate if the polarity is reversed.
- Although the +V and COM terminals of rows A and B of CN1 and CN2 are internally connected, wire all points completely.

Note The ON response time will be 20 µs maximum and OFF response time will be 300 µs maximum even if the response times are set to 0 ms due to internal element delays.

CS1W-MD292 DC Input/Transistor Output Unit (48/48 Points, Sourcing)

Outputs (CN1)

Rated Voltage	12 to 24 V DC
Operating Load Voltage Range	10.2 to 26.4 V DC
Maximum Load Current	0.1 A/point, 1.2 A/common, 3.6 A/Unit (See note.)
Maximum Inrush Current	1.0 A/point, 10 ms max. 8.0 A/common, 10 ms max.
Leakage Current	0.1 mA max.
Residual Voltage	1.5 V max.
ON Response Time	0.5 ms max.
OFF Response Time	1.0 ms max.
No. of Circuits	48 points (16 points/common, 3 commons)
Fuse	3 A (1 per common) The fuse cannot be replaced by the user.
External Power Supply	10.2 to 26.4 V DC, 50 mA min.
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	350 mA max. at 5 V DC
Weight	320 g max.
Accessories	Two connectors for external wiring (soldered)

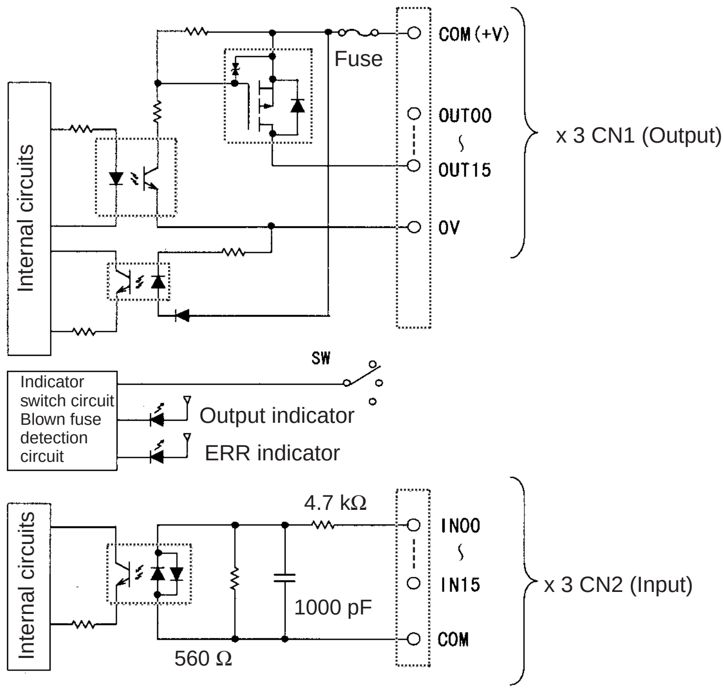
Note The maximum load currents will be 1.0 A/common and 3.0 A/Unit if a pressure-welded connector is used.

Inputs (CN2)

Rated Input Voltage	24 V DC
Operating Input Voltage	20.4 to 26.4 V DC
Input Impedance	4.7 kΩ
Input Current	Approx. 5 mA (at 24 V DC)
ON Voltage/ON Current	17 V DC min./3 mA min.
OFF Voltage/OFF Current	5.0 V DC max./1 mA max.
ON Response Time	8.0 ms max. (Possible to select one out of eight times from 0 to 32 ms in the PLC Setup.) (See note below.)
OFF Response Time	8.0 ms max. (Possible to select one out of eight times from 0 to 32 ms in the PLC Setup.) (See note below.)
No. of Circuits	48 points (16 points/common, 3 commons)
Number of Simultaneously ON Points	50% (8 points/common) (at 24 V DC) (Depends on ambient temperature.)
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	350 mA max. at 5 V DC
Weight	320 g max.
Accessories	Two connectors for external wiring (soldered)

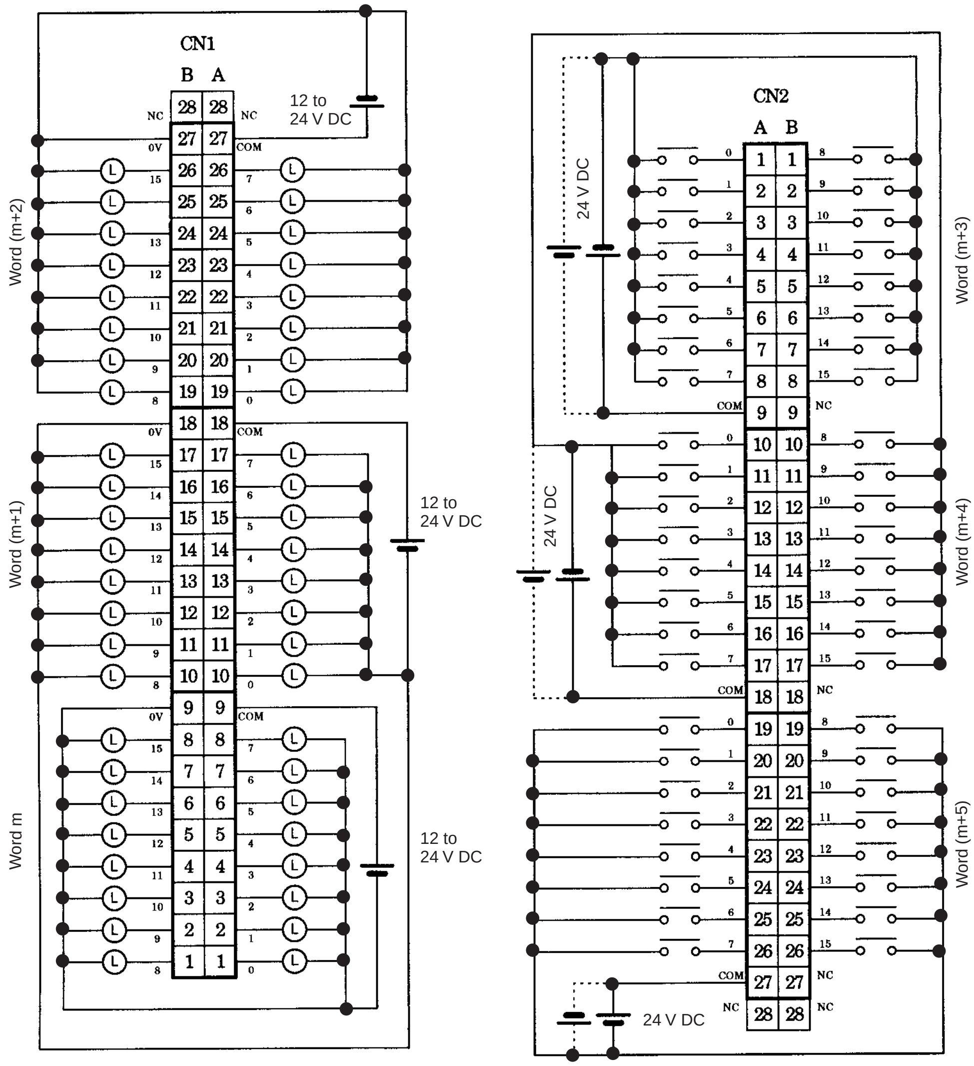
Note The input ON and OFF response times for Basic I/O Units can be set to 0 ms, 0.5 ms, 1 ms, 2 ms, 4 ms, 8 ms, 16 ms, or 32 ms in the PLC Setup.

Circuit Configuration



Note The ERR indicator will light if a fuse blows or if the external power supply is turned OFF, and the corresponding Flag in the Basic I/O Unit Information Area (A050 to A089) will turn ON.

Terminal Connections: CS1W-MD292 24-V DC 48-point Input/48-point Transistor Output Unit (Sourcing Outputs)



CS1W-MD561 TTL I/O Unit (32/32 Points)

Outputs (CN1)

Rated Voltage	5 V DC±10%
Operating Load Voltage Range	4.5 to 5.5 V DC
Maximum Load Current	35 mA/point, 560 mA/common, 1.12 A/Unit
Leakage Current	0.1 mA max.
Residual Voltage	0.4 V max.
ON Response Time	0.2 ms max.
OFF Response Time	0.3 ms max.
No. of Circuits	32 points (16 points/common, 2 commons)
Fuse	None
External Power Supply	5 V DC±10%, 40 mA min. (1.2 mA × No. of ON points)
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	270 mA max. at 5 V DC
Weight	260 g max.
Accessories	Two connectors for external wiring (soldered)

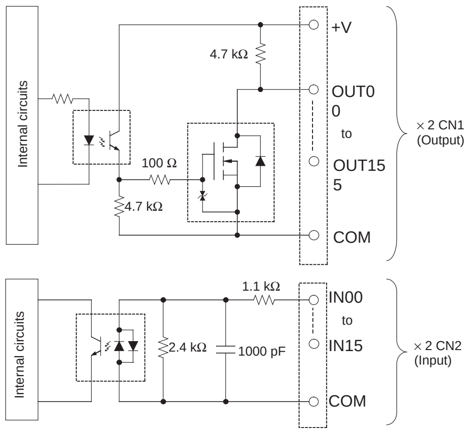
Inputs (CN2)

Rated Input Voltage	5 V DC±10%
Input Impedance	1.1 kΩ
Input Current	Approx. 3.5 mA (at 5 V DC)
ON Voltage	3.0 V DC min.
OFF Voltage	1.0 V DC max.
ON Response Time	8.0 ms max. (Possible to select one out of eight times from 0 to 32 ms in the PLC Setup.) (See notes 1 and 2.)
OFF Response Time	8.0 ms max. (Possible to select one out of eight times from 0 to 32 ms in the PLC Setup.) (See notes 1 and 2.)
No. of Circuits	32 points (16 points/common, 2 commons)
Number of Simultaneously ON Points	No restrictions
Insulation Resistance	20 MΩ between the external terminals and the GR terminal (100 V DC)
Dielectric Strength	1,000 V AC between the external terminals and the GR terminal for 1 minute at a leakage current of 10 mA max.
Internal Current Consumption	270 mA max. at 5 V DC
Weight	260 g max.
Accessories	Two connectors for external wiring (soldered)

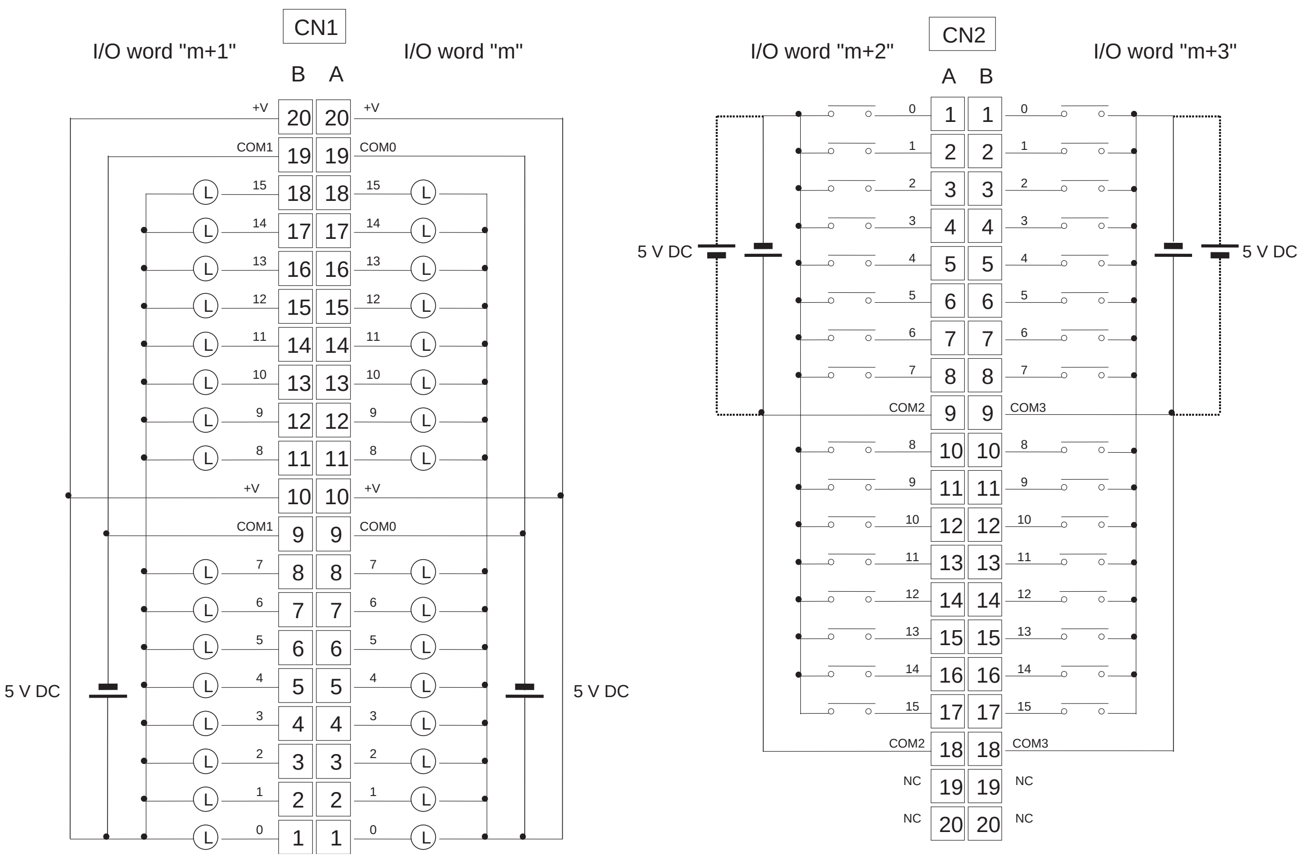
Note (1) The ON response time will be 20 µs maximum and OFF response time will be 300 µs maximum even if the response times are set to 0 ms due to internal element delays.

(2) The input ON and OFF response times for Basic I/O Units can be set to 0 ms, 0.5 ms, 1 ms, 2 ms, 4 ms, 8 ms, 16 ms, or 32 ms in the PLC Setup.

Circuit Configuration



Terminal Connections: CS1W-MD561 TTL 32-point Input/32-point Output Unit



- When wiring, pay careful attention to the polarity. The load may operate if the polarity is reversed.
- Although the +V and COM terminals of rows A and B of CN1 and CN2 are internally connected, wire all points completely.

Appendix E

Memory Map of PLC Memory Addresses

PLC Memory Addresses

PLC memory addresses are set in Index Registers (IR00 to IR15) to indirectly address I/O memory. Normally, use the MOVE TO REGISTER (MOVR(560)) and MOVE TIMER/COUNTER PV TO REGISTER (MOVRW(561)) instructions to set PLC memory addresses into the Index Registers.

Some instructions, such as DATA SEARCH (SRCH(181)), FIND MAXIMUM (MAX(182)), and FIND MINIMUM (MIN(183)), output the results of processing to an Index Register to indicate an PLC memory address.

There are also instructions for which Index Registers can be directly designated to use the PLC memory addresses stored in them by other instructions. These instructions include DOUBLE MOVE (MOVL(498)), some symbol comparison instructions (=L, <>L, <L, >L, <=L, and >=L), DOUBLE COMPARE (CMPL(060)), DOUBLE DATA EXCHANGE (XCGL(562)), DOUBLE INCREMENT BINARY (++L(591)), DOUBLE DECREMENT BINARY (—L(593)), DOUBLE SIGNED BINARY ADD WITHOUT CARRY (+L(401)), DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY (–L(411)), SET RECORD LOCATION (SETR(635)), and GET RECORD LOCATION (GETR(636)).

The PLC memory addresses all are continuous and the user must be aware of the order and boundaries of the memory areas. As reference, the PLC memory addresses are provided in a table at the end of this appendix.

Note Directly setting PLC memory addresses in the program should be avoided whenever possible. If PLC memory addresses are set in the program, the program will be less compatible with new CPU Unit models or CPU Units for which changed have been made to the layout of the memory.

Memory Configuration

There are two classifications of the RAM memory (with battery backup) in a NE1S-series CPU Unit.

Parameter Areas: These areas contain CPU Unit system setting data, such as the PLC Setup, CS-series 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

Do not access addresses reserved by the system.

Classification	PLC memory addresses (hex)	User addresses	Area
Parameter areas	00000 to 0B0FF	---	PLC Setup Area Registered I/O Table Area Routing Table Area CS-series CPU Bus Unit Setup Area Real I/O Table Area Unit Profile Area
I/O memory areas	0B100 to 0B1FF	---	Reserved for system.
	0B200 to 0B7FF	---	Reserved for system.
	0B800 to 0B801	TK00 to TK31	Task Flag Area
	0B802 to 0B83F	---	Reserved for system.
	0B840 to 0B9FF	A000 to A447	Read-only Auxiliary Area
	0BA00 to 0BBFF	A448 to A959	Read/Write Auxiliary Area
	0BC00 to 0BDFF	---	Reserved for system.
	0BE00 to 0BEFF	T0000 to T4095	Timer Completion Flags
	0BF00 to 0BFFF	C0000 to C4095	Counter Completion Flags
	0C000 to 0D7FF	CIO 0000 to CIO 6143	CIO Area
	0D800 to 0D9FF	H000 to H511	Holding Area
	0DA00 to 0DDFF	---	Reserved for system.
	0DE00 to 0DFFF	W000 to W511	Work Area
	0E000 to 0EFFF	T0000 to T4095	Timer PVs
	0F000 to 0FFFF	C0000 to C4095	Counter PVs
	10000 to 17FFF	D00000 to D32767	DM Area
	18000 to 1FFFF	E0_00000 to E0_32767	EM Area bank 0
	20000 to 27FFF	E1_00000 to E1_32767	EM Area bank 1 (See note.)
	28000 to 2FFFF	E2_00000 to E2_32767	EM Area bank 2 (See note.)
	30000 to 37FFF	E3_00000 to E3_32767	EM Area bank 3 (See note.)
	38000 to 3FFFF	E4_00000 to E4_32767	EM Area bank 4 (See note.)
	40000 to 47FFF	E5_00000 to E5_32767	EM Area bank 5 (See note.)
	48000 to 4FFFF	E6_00000 to E6_32767	EM Area bank 6 (See note.)
	50000 to 57FFF	E7_00000 to E7_32767	EM Area bank 7 (See note.)
	58000 to 5FFFF	E8_00000 to E8_32767	EM Area bank 8 (See note.)
	60000 to 67FFF	E9_00000 to E9_32767	EM Area bank 9 (See note.)
	68000 to 6FFFF	EA_00000 to EA_32767	EM Area bank A (See note.)
	70000 to 77FFF	EB_00000 to EB_32767	EM Area bank B (See note.)
	78000 to 7FFFF	EC_00000 to EC_32767	EM Area bank C (See note.)
	F8000 to FFFFF	E0000 to E32767	EM Area, current bank Reserved for system.

Note In the NE1S-CPU01, these words are reserved by the system.

Appendix F
Details on Auxiliary Area Operation

A100 to A199: Error Log Area

A100	Error code		
A101	Error flag contents		
A102	min	s	Error record
A103	day	hr	
A104	yr	mo	
⋮			⋮
A195	Error code		
A196	Error flag contents		
A197	min	s	Error record
A198	day	hr	
A199	yr	mo	

The following data would be generated in an error record if a memory error (error code 80F1) occurred on 1 April 1998 at 17:10:30 with the error located in the PLC Setup (04 hex).

80 F1	
00 04	
10	30
01	17
98	04

The following data would be generated in an error record if an FALS error with FALS number 001 occurred on 2 May 1997 at 8:30:15.

C1 01	
00 00	
30	15
02	08
97	05

Error Codes and Error Flags

Classification	Error code	Meaning	Error flags
System-defined fatal errors	80F1	Memory error	A403
	80C0 to 80C7, 80CF	I/O bus error	A404
	80E9	Duplicate number error	A410, A411 to 416 (See note 3.)
	80E1	Too many I/O error	A407
	80E0	I/O setting error	---
	80F0	Program error	A295 to 299 (See note 4.)
	809F	Cycle time too long error	---
	80EA	Duplicate Expansion Rack number error	A40900 to 40907
User-defined fatal errors	C101 to C2FF	FALS instruction executed (See note 1.)	---
User-defined non-fatal errors	4101 to 42FF	FAL instruction executed (See note 2.)	---
System-defined non-fatal errors	008B	Interrupt task error	A426
	009A	Basic I/O error	A408
	009B	PLC Setup setting error	A406
	00B0	Communications interface error	---
	0200 to 020F	CS-series CPU Bus Unit error	A417
	0300 to 035F	Special I/O Unit error	A418 to 423 (See note 5.)
	00A0 to 00A1	SYSMAC BUS error	A405
	0400 to 040F	CS-series CPU Bus Unit setup error	A427
	0500 to 055F	Special I/O Unit setup error	A428 to 433 (See note 5.)

Note
1. C101 to C2FF will be stored for FALS numbers 001 to 511.
2. 4101 to 42FF will be stored for FAL numbers 001 to 511.
3. The contents of the error flags for a duplicate number error are as follows:
   Bits 0 to 7: Unit number (binary), 00 to 5F hex for Special I/O Units, 00 to 0F hex for CS-series CPU Bus Units
   Bits 8 to 14: All zeros.
   Bit 15: Unit type, 0 for CS-series CPU Bus Units and 1 for Special I/O Units.
4. Only the contents of A295 is stored as the error flag contents for program errors.
5. 0000 hex will be stored as the error flag contents.

Appendix G

Instruction Support and Operand/Variable Restrictions

This appendix provides a list of supported instructions and the restrictions that apply to operands and variables used by operands, including the use of array variables and AT settings. For a list of instructions, refer to *Sequence Input Instructions*.

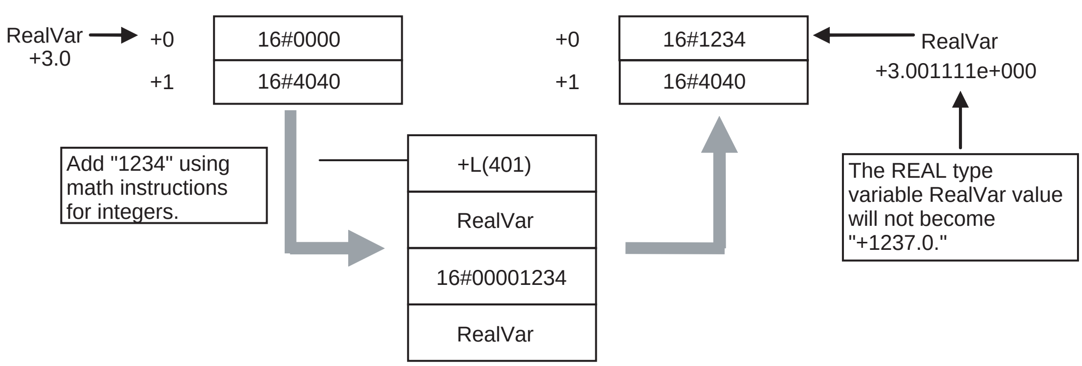
Precautions

- The variables used by operands are restricted and if these restrictions are ignored when inputting the variable, an error or warning may be output for the instruction during input or building. For details, refer to *Operand Restrictions*. Operands with variable restrictions are indicated by an asterisk (*) in the table.
- Instructions with restrictions depending on the method used to specify variables used by the operand are indicated in the table by "Note." For details on restrictions, refer to *Variable Specification Methods and Precautions*.
- Instructions that cannot be used in function block definitions are indicated in the table.

Operand Restrictions

When specifying the first or last word of a range of words for an instruction operand, I/O parameters cannot be used to pass the required values to or from I/O variables. In this case, prepare an array variable with the required number of array elements, set the data in each array element in the function block, and specify the beginning (or end) array variable in the operand.

Apart from floating-point math instructions, REAL type variables can also be used as operands, but processing is performed using the IEEE754 data type format.



Special restrictions also apply for some variables. These restrictions apply for the following instructions.

ASFT, IORF, SFTR, SLD, SRD, WSFT, or BSET

These seven instructions have the following restrictions.

- If automatically allocated variables and addresses (including global variables with allocated addresses) are used in the corresponding operand, an error will occur during building.
- If automatically allocated variables are combined in the corresponding operand, a warning will occur during building.

Instruction	Operand
ASFT	Second or third operand
IORF	First or second operand
SFTR	Second or third operand
SLD	First or second operand
SRD	First or second operand

Instruction	Operand
WSFT	Second or third operand
BSET	Second or third operand

CNR

The CNR instruction has the following restrictions.

- If the first and second operands each have different data types, an error will occur during input.
- If automatically allocated variables and addresses (including global variables with allocated addresses) are used in the instruction operand, an error will occur during building.
- If automatically allocated variables are combined in the instruction operand, a warning will occur during building.

COLL, DIST, PUSH, LIFO, FIFO, PMCR, SNUM, SREAD, SWRIT, SINS, SDEL, CSND, FREAD, FWRIT FAL, or FALS

These 16 instructions have the following restrictions.

- The variable data type used in the corresponding operand and the allocated size of the operand is not checked.
- If automatically allocated variables are used in the corresponding operand, a warning will occur during building.

Instruction	Operand
COLL	First operand
DIST	Second operand
PUSH	First operand
LIFO	First operand
FIFO	First operand
PMCR	Third or fourth operand
SNUM	First operand
SREAD	First operand
SWRIT	First operand
SINS	First operand
SDEL	First operand
CSND	First, second, or third operand
FREAD	Third operand
FWRIT	Third or fourth operand
FAL	Second operand
FALS	Second operand

MOVR

The MOVR instruction has the following restriction.

- The variable data type used in the corresponding operand and the allocated size of the operand is not checked.

Restrictions on Variable Types for Operands

The following symbols are used in the *Data type* column of the following tables to indicate restrictions on the variables types that can be used for operands.

Symbol	Meaning
❍	Can be specified.
∆	Can be specified, but only first or last address is used.
▲	Duplication with other operands may occur if specified.

Symbol	Meaning
Note	The conditions ❍, ∆, and × apply depending on the specification method (the conditions, however, are not checked when the operand is input).
AT	A global variable must be used, an address must be allocated, and the area must be checked (e.g., for size relationships with other operations).
*	Restrictions apply to the variables that can be used. If the restrictions are ignored, an error or warning will occur during input or building. For details, refer to *Operand Restrictions* on page 545.
×	Cannot be specified.

Note (1) When specifying the first or last word of multiple words for an instruction operand, I/O parameters cannot be used to pass data to or from I/O variables. Internal array variables must be used. For multiword operands, an array variable must be prepared in advance with the required number of elements and the data must be set for the array in the function block definition. The first or last element in the array variable is then specified for the operand to set the first or last word.

(2) The data types that can be used for constants, 16#, 10#, +10#, and –10#, are specified.

Sequence Input Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/ UINT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
LOAD	LD @LD %LD !LD !@LD !%LD		Supported		❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	❍	❍	×
LOAD NOT	LD NOT !LD NOT		Supported		❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	❍	❍	×
AND	AND @AND %AND !AND !@AND !%AND		Supported		❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	❍	❍	×
AND NOT	AND NOT !AND NOT		Supported		❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	❍	❍	×
OR	OR @OR %OR !OR !@OR !%OR		Supported		❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	❍	❍	×
OR NOT	OR NOT !OR NOT		Supported		❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	❍	❍	×
AND LOAD	AND LD		Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
OR LOAD	OR LD		Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
NOT	NOT	520	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
CONDITION ON	UP	521	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
CONDITION OFF	DOWN	522	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
BIT TEST	LD TST	350	Supported	S: Source word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	∆
				N: Bit number	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
BIT TEST	LD TSTN	351	Supported	S: Source word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	∆
				N: Bit number	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
BIT TEST	AND TST	350	Supported	S: Source word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	∆
				N: Bit number	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆

Instruction	Mnemonic	Func-tion code	Support	Operands	BOOL	WORD	DWORD	INT/ UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con-stant	TIMER	COUNTER	Literal text strings
BIT TEST	AND TSTN	351	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	Δ
				N: Bit number	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
BIT TEST	OR TST	350	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	Δ
				N: Bit number	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
BIT TEST	OR TSTN	351	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	Δ
				N: Bit number	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ

Sequence Output Instructions

Instruction	Mnemonic	Func-tion code	Support	Operands	BOOL	WORD	DWORD	INT/ UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con-stant	TIMER	COUNTER	Literal text strings
OUTPUT	OUT !OUT		Supported		❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	×	×	×
OUTPUT NOT	OUT NOT !OUT NOT		Supported		❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	×	×	×
KEEP	KEEP !KEEP	011	Supported	B: Bit	❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	×	×	×
DIFFERENTIATE UP	DIFU !DIFU	013	Supported	B: Bit	❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	×	×	×
DIFFERENTIATE DOWN	DIFD !DIFD	014	Supported	B: Bit	❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	×	×	×
SET	SET @SET %SET !SET !@SET !%SET		Supported	B: Bit	❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	×	×	×
RESET	RSET @RSET %RSET !RSET !@RSET !%RSET		Supported	B: Bit	❍	×	×	×	×	×	Array: No Element: Yes	Structure: No Members: Yes	×	×	×	×	×
MULTIPLE BIT SET	SETA @SETA	530	Supported	D: Beginning word	×	❍	Δ	❍	Δ	Δ	Note	Note	Δ	×	❍	❍	×
				N1: Beginning bit	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				N2: Number of bits	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
MULTIPLE BIT RESET	RSTA @RSTA	531	Supported	D: Beginning word	×	❍	Δ	❍	Δ	Δ	Note	Note	Δ	×	❍	❍	×
				N1: Beginning bit	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				N2: Number of bits	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
SINGLE BIT SET	SETB @SETB !SETB	532	Supported	D: Word address	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
				N: Bit number	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
SINGLE BIT RESET	RSTB @RSTB !RSTB	533	Supported	D: Word address	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
				N: Bit number	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/ UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
SINGLE BIT OUTPUT	OUTB @OUTB !OUTB	534	Supported	D: Word address	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
				N: Bit number	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ

Sequence Control Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/ UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
END	END	001	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
NO OPERATION	NOP	000	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
INTERLOCK	IL	002	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
INTERLOCK CLEAR	ILC	003	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
JUMP	JMP	004	Supported	N: Jump number	×	❍ (Not in FB)	Δ (Not in FB)	❍ (Not in FB)	Δ (Not in FB)	Δ (Not in FB)	Δ (Not in FB)	Δ (Not in FB)	Δ (Not in FB)	❍ 16# 10#	❍ (Not in FB)	❍ (Not in FB)	Δ (Not in FB)
JUMP END	JME	005	Supported	N: Jump number	×	×	×	×	×	×	×	×	×	❍ 16# 10#	×	×	×
CONDITIONAL JUMP	CJP	510	Supported	N: Jump number	×	❍ (Not in FB)	Δ (Not in FB)	❍ (Not in FB)	Δ (Not in FB)	Δ (Not in FB)	Δ (Not in FB)	Δ (Not in FB)	Δ (Not in FB)	❍ 16# 10#	❍ (Not in FB)	❍ (Not in FB)	Δ (Not in FB)
CONDITIONAL JUMP	CJPN	511	Not supported	N: Jump number	×	×	×	×	×	×	×	×	×	×	×	×	×
MULTIPLE JUMP	JMP0	515	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
MULTIPLE JUMP END	JME0	516	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
FOR-NEXT LOOPS	FOR	512	Supported	N: Number of loops	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
BREAK LOOP	BREAK	514	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
FOR-NEXT LOOPS	NEXT	513	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×

Timer and Counter Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/ UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
TIMER	TIMX (BIN)	550	Supported	N: Timer number	×	×	×	×	×	×	×	×	×	Not a constant. Specify normal number or timer area number.	❍	×	×
				S: Set value	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
HIGH-SPEED TIMER	TIMHX (BIN)	551	Supported	N: Timer number	×	×	×	×	×	×	×	×	×	Not a constant. Specify normal number or timer area number.	❍	×	×
				S: Set value	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
ONE-MS TIMER	TMHHX (BIN)	552	Supported	N: Timer number (Timer number variable cannot be specified)	×	×	×	×	×	×	×	×	×	Not a constant. Specify normal number or timer area number.	×	×	×
				S: Set value	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
ACCUMULATIVE TIMER	TTIMX (BIN)	555	Supported	N: Timer number	×	×	×	×	×	×	×	×	×	Not a constant. Specify normal number or timer area number.	❍	×	×
				S: Set value	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
LONG TIMER	TIMLX (BIN)	553	Supported	D1: Completion Flags	×	❍	∆	❍	∆	∆	∆	∆	∆	×	×	×	×
				D2: PV word	×	▲	❍	▲	❍	❍	▲	▲	❍	×	×	×	×
				S: SV word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10#	▲	▲	▲
MULTI-OUTPUT TIMER	MTIMX (BIN)	554	Supported	D1: Completion Flags	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	×
				D2: PV word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	×
				S: First SV word	×	▲	▲	▲	▲	▲	▲ Duplication may occur if less than 16 bytes.	▲ Duplication may occur if less than 16 bytes.	❍	×	∆	∆	∆
COUNTER	CNTX (BIN)	546	Supported	N: Counter number	×	×	×	×	×	×	×	×	×	Not a constant. Specify normal number or counter area number.	×	❍	×
				S: Set value	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
REVERSIBLE COUNTER	CNTRX (BIN)	548	Supported	N: Counter number	×	×	×	×	×	×	×	×	×	Not a constant. Specify normal number or counter area number.	×	❍	×
				S: Set value	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
RESET TIMER/ COUNTER	CNRX @CNRX (BIN)	547	Supported	N1: First number in range	×	×	×	×	×	×	×	×	×	×	*	*	×
				N2: Last number in range	×	×	×	×	×	×	×	×	×	×	*	*	×

Comparison Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
Symbol Comparison (Unsigned)	LD,AND, OR + =, <>, <, <=, >, >=	300 (=) 305 (<>) 310 (<) 315 (<=) 320 (>) 325 (>=)	Supported	S1: Comparison data 1	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				S2: Comparison data 2	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
Symbol Comparison (Double-word, unsigned)	LD,AND, OR + =, <>, <, <=, >, >= + L	301 (=) 306 (<>) 311 (<) 316 (<=) 321 (>) 326 (>=)	Supported	S1: Comparison data 1, lower word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10#	▲	▲	▲
				S2: Comparison data 2, upper word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10#	▲	▲	▲

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con-stant	TIMER	COUNTER	Literal text strings
Symbol Comparison (Signed)	LD,AND, OR + =, <>, <, <=, >, >= + S	302 (=) 307 (<>) 312 (<) 317 (<=) 322 (>) 327 (>=)	Supported	S1: Comparison data 1	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍	❍	❍	Δ
				S2: Comparison data 2	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
Symbol Comparison (Double-word, signed)	LD,AND, OR + =, <>, <, <=, >, >= + SL	303 (=) 308 (<>) 313 (<) 318 (<=) 323 (>) 328 (>=)	Supported	S1: Comparison data 1, lower word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10# +10# –10#	▲	▲	▲
				S2: Comparison data 2, upper word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10# +10# –10#	▲	▲	▲
UNSIGNED COMPARE	CMP !CMP	020	Supported	S1: Comparison data 1	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				S2: Comparison data 2	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
DOUBLE UNSIGNED COMPARE	CMPL	060	Supported	S1: Comparison data 1, lower word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10#	▲	▲	▲
				S2: Comparison data 2, upper word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10#	▲	▲	▲
SIGNED BINARY COMPARE	CPS !CPS	114	Supported	S1: Comparison data 1	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				S2: Comparison data 2	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
DOUBLE SIGNED BINARY COMPARE	CPSL	115	Supported	S1: Comparison data 1, lower word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10# +10# –10#	▲	▲	×
				S2: Comparison data 2, upper word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	▲	▲	▲
TABLE COMPARE	TCMP @TCMP	085	Supported	S: Source data	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				T: First word of table	×	▲	▲	▲	▲	▲	Note	Note	❍	×	▲	▲	▲
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
MULTIPLE COMPARE	MCMP @MCMP	019	Supported	S1: First word of set 1	×	▲	▲	▲	▲	▲	Note	Note	❍	×	▲	▲	▲
				S2: First word of set 2	×	▲	▲	▲	▲	▲	Note	Note	❍	×	▲	▲	▲
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
UNSIGNED BLOCK COMPARE	BCMP @BCMP	068	Supported	S: Source data	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				T: First word of table	×	▲	▲	▲	▲	▲	Note	Note	❍	×	▲	▲	▲
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
AREA RANGE COMPARE	ZCP	088	Supported	CD: Compare data (1 word)	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				LL: Lower limit of range	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				UL: Upper limit of range	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
DOUBLE AREA RANGE COMPARE	ZCPL	116	Supported	CD: Compare data (2 words)	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10#	▲	▲	▲
				LL: Lower limit of range, lower word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10#	▲	▲	▲
				UL: Upper limit of range, upper word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10#	▲	▲	▲

Data Movement Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
MOVE	MOV @MOV !MOV !@MOV	021	Supported	S: Source	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: Destination	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE MOVE	MOVL @MOVL	498	Supported	S: First source word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10#	▲	▲	▲
				D: First destination word	×	▲	❍	▲	❍	❍	▲	▲	❍	×	▲	▲	×
MOVE NOT	MVN @MVN	022	Supported	S: Source	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: Destination	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE MOVE NOT	MVNL @MVNL	499	Supported	S: First source word	×	▲	❍	▲	❍	❍	▲	▲	❍	❍ 16# 10#	▲	▲	▲
				D: First destination word	×	▲	❍	▲	❍	❍	▲	▲	❍	×	▲	▲	×
MOVE BIT	MOVB @MOVB	082	Supported	S: Source word or data	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
MOVE DIGIT	MOVD @MOVD	083	Supported	S: Source word or data	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
MULTIPLE BIT TRANSFER	XFRB @XFRB	062	Supported	C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				S: First source word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				D: First destination word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×
BLOCK TRANSFER	XFER @XFER	070	Supported	N: Number of words	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				S: First source word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				D: First destination word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
BLOCK SET	BSET @BSET	071	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				St: Starting word	×	*	*	*	*	*	*	*	*	×	*	*	×
				E: End word	×	*	*	*	*	*	*	*	*	×	*	*	×
DATA EXCHANGE	XCHG @XCHG	073	Supported	E1: First exchange word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
				E2: Second exchange word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE DATA EXCHANGE	XCGL @XCGL	562	Supported	E1: First exchange word	×	▲	❍	▲	❍	❍	▲	▲	❍	×	▲	▲	×
				E2: Second exchange word	×	▲	❍	▲	❍	❍	▲	▲	❍	×	▲	▲	×
SINGLE WORD DISTRIBUTE	DIST @DIST	080	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				Bs: Destination base address	×	*	*	*	*	*	*	*	*	×	*	*	×
				Of: Offset	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
DATA COLLECT	COLL @COLL	081	Supported	Bs: Source base address	×	*	*	*	*	*	*	*	*	×	*	*	*
				Of: Offset	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
MOVE TO REGISTER	MOVR @MOVR	560	Supported	S: Source (desired word orbit)	❍	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍ Completion Flag	❍ Completion Flag	×
				D: Destination (Index Register) (Variable specification not possible)	×	×	×	×	×	×	×	×	×	×	×	×	×
MOVE TIMER/ COUNTER PV TO REGISTER	MOVRW @MOVRW	561	Supported	S: Source (desired TC number)	×	×	×	×	×	×	×	×	×	×	❍ PV	❍ PV	×
				D: Destination (Index Register) (Variable specification not possible)	×	×	×	×	×	×	×	×	×	×	×	×	×

Data Shift Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
REVERSIBLE SHIFT REGISTER	SFTR @SFTR	084	Supported	C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	Δ
				St: Starting word	×	*	*	*	*	*	*	*	*	×	*	*	×
				E: End word	×	*	*	*	*	*	*	*	*	×	*	*	×
ASYNCHRONOUS SHIFT REGISTER	ASFT @ASFT	017	Supported	C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	Δ
				St: Starting word	×	*	*	*	*	*	*	*	*	×	*	*	×
				E: End word	×	*	*	*	*	*	*	*	*	×	*	*	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
WORD SHIFT	WSFT @WSFT	016	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	16# 10# +10# –10#	❍	❍	Δ
				St: Starting word	×	*	*	*	*	*	*	*	*	×	*	*	×
				E: End word	×	*	*	*	*	*	*	*	*	×	*	*	×
ARITHMETIC SHIFT LEFT	ASL @ASL	025	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE SHIFT LEFT	ASLL @ASLL	570	Supported	Wd: First word	×	▲	❍	▲	❍	❍	▲	▲	❍	×	▲	▲	×
ARITHMETIC SHIFT RIGHT	ASR @ASR	026	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE SHIFT RIGHT	ASRL @ASRL	571	Supported	Wd: First word	×	▲	❍	▲	❍	❍	▲	▲	❍	×	▲	▲	×
ROTATE LEFT	ROL @ROL	027	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE ROTATE LEFT	ROLL @ROLL	572	Supported	Wd: First word	×	▲	❍	▲	❍	❍	▲	▲	❍	×	▲	▲	×
ROTATE LEFT WITHOUT CARRY	RLNC @RLNC	574	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL @RLNL	576	Supported	Wd: First word	×	▲	❍	▲	❍	❍	×	×	❍	×	×	×	×
ROTATE RIGHT	ROR @ROR	028	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE ROTATE RIGHT	RORL @RORL	573	Supported	Wd: First word	×	▲	❍	▲	❍	❍	×	×	❍	×	×	×	×
ROTATE RIGHT WITHOUT CARRY	RRNC @RRNC	575	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL @RRNL	577	Supported	Wd: First word	×	▲	❍	▲	❍	❍	×	×	❍	×	×	×	×
ONE DIGIT SHIFT LEFT	SLD @SLD	074	Supported	St: Starting word	×	*	*	*	*	*	*	*	*	×	*	*	×
				E: End word	×	*	*	*	*	*	*	*	*	×	*	*	×
ONE DIGIT SHIFT RIGHT	SRD @SRD	075	Supported	St: Starting word	×	*	*	*	*	*	*	*	*	×	*	*	×
				E: End word	×	*	*	*	*	*	*	*	*	×	*	*	×
SHIFT N-BIT DATA LEFT	NSFL @NSFL	578	Supported	D: Beginning word for shift	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×
				C: Beginning bit	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				N: Shift data length	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
SHIFT N-BIT DATA RIGHT	NSFR @NSFR	579	Supported	D: Beginning word for shift	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×
				C: Beginning bit	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				N: Shift data length	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
SHIFT N-BITS LEFT	NASL @NASL	580	Supported	D: Shift word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
				C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
DOUBLE SHIFT N-BITS LEFT	NSLL @NSLL	582	Supported	D: First shift word	×	▲	❍	▲	❍	❍	▲	▲	❍	×	▲	▲	×
				C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
SHIFT N-BITS RIGHT	NASR @NASR	581	Supported	D: Shift word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
				C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
DOUBLE SHIFT N-BITS RIGHT	NSRL @NSRL	583	Supported	D: First shift word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
				C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ

Increment/Decrement Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
INCREMENT BINARY	++ @++	590	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE INCREMENT BINARY	++L @++L	591	Supported	Wd: First word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
DECREMENT BINARY	-- @--	592	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE DECREMENT BINARY	--L @--L	593	Supported	Wd: First word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
INCREMENT BCD	++B @++B	594	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE INCREMENT BCD	++BL @++BL	595	Supported	Wd: First word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
DECREMENT BCD	--B @--B	596	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE DECREMENT BCD	--BL @--BL	597	Supported	Wd: First word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×

Symbol Math Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
SIGNED BINARY ADD WITHOUT CARRY	+ @+	400	Supported	Au: Augend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				Ad: Addend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L @+L	401	Supported	Au: First augend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				Ad: First addend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
SIGNED BINARY ADD WITH CARRY	+C @+C	402	Supported	Au: Augend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				Ad: Addend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×

Instruction	Mnemonic	Func-tion code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con-stant	TIMER	COUNTER	Literal text strings
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL @+CL	403	Sup-ported	Au: First augend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				Ad: First addend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
BCD ADD WITHOUT CARRY	+B @+B	404	Sup-ported	Au: Augend word	×	×	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16#	❍	❍	Δ
				Ad: Addend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE BCD ADD WITHOUT CARRY	+BL @+BL	405	Sup-ported	Au: First augend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				Ad: First addend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
BCD ADD WITH CARRY	+BC @+BC	406	Sup-ported	Au: Augend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16#	❍	❍	Δ
				Ad: Addend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE BCD ADD WITH CARRY	+BCL @+BCL	407	Sup-ported	Au: First augend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				Ad: First addend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
SIGNED BINARY SUBTRACT WITHOUT CARRY	- @-	410	Sup-ported	Mi: Minu-end word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				Su: Sub-trahend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	-L @-L	411	Sup-ported	Mi: First minuend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				Su: First subtra-hend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
SIGNED BINARY SUBTRACT WITH CARRY	-C @-C	412	Sup-ported	Mi: Minu-end word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				Su: Sub-trahend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
DOUBLE SIGNED BINARY WITH CARRY	-CL @-CL	413	Supported	Mi: First minuend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				Su: First subtrahend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
BCD SUBTRACT WITHOUT CARRY	-B @-B	414	Supported	Mi: Minuend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16#	❍	❍	Δ
				Su: Subtrahend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE BCD SUBTRACT WITHOUT CARRY	-BL @-BL	415	Supported	Mi: First minuend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				Su: First subtrahend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	---	×	×	---
BCD SUBTRACT WITH CARRY	-BC @-BC	416	Supported	Mi: Minuend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16#	❍	❍	Δ
				Su: Subtrahend word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE BCD SUBTRACT WITH CARRY	-BCL @-BCL	417	Supported	Mi: First minuend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				Su: First subtrahend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
SIGNED BINARY MULTIPLY	* @*	420	Supported	Md: Multiplicand word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				Mr: Multiplier word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				R: First result word	×	×	❍	×	❍	❍	Δ	Δ	Δ	×	×	×	×
DOUBLE SIGNED BINARY MULTIPLY	*L @*L	421	Supported	Md: First multiplicand word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				Mr: First multiplier word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
UNSIGNED BINARY MULTIPLY	*U @*U	422	Supported	Md: Multiplicand word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				Mr: Multiplier word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				R: First result word	×	×	❍	×	❍	❍	Δ	Δ	Δ	×	×	×	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
DOUBLE UNSIGNED BINARY MULTIPLY	*UL @*UL	423	Supported	Md: First multiplicand word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
				Mr: First multiplier word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
BCD MULTIPLY	*B @*B	424	Supported	Md: Multiplicand word	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16#	❍	❍	∆
				Mr: Multiplier word	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16#	❍	❍	∆
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
DOUBLE BCD MULTIPLY	*BL @*BL	425	Supported	Md: First multiplicand word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				Mr: First multiplier word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				R: First result word	×	×	×	×	×	×	×	×	❍	×	×	×	×
SIGNED BINARY DIVIDE	/ @/	430	Supported	Dd: Dividend word	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10# +10# –10#	❍	❍	∆
				Dr: Divisor word	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10# +10# –10#	❍	❍	∆
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
DOUBLE SIGNED BINARY DIVIDE	/L @/L	431	Supported	Dd: First dividend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				Dr: First divisor word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
UNSIGNED BINARY DIVIDE	/U @/U	432	Supported	Dd: Dividend word	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				Dr: Divisor word	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				R: Result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
DOUBLE UNSIGNED BINARY DIVIDE	/UL @/UL	433	Supported	Dd: First dividend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
				Dr: First divisor word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
				R: First result word	×	×	×	×	×	×	×	×	❍	×	×	×	×
BCD DIVIDE	/B @/B	434	Supported	Dd: Dividend word	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16#	❍	❍	∆
				Dr: Divisor word	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16#	❍	❍	∆
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
DOUBLE BCD DIVIDE	/BL @/BL	435	Supported	Dd: First dividend word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				Dr: First divisor word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				R: First result word	×	×	×	×	×	×	×	×	❍	×	×	×	×

Conversion Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
BCD-TO-BINARY	BIN @BIN	023	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE BCD-TO-DOUBLE BINARY	BINL @BINL	058	Supported	S: First source word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
BINARY-TO-BCD	BCD @BCD	024	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE BINARY-TO-DOUBLE BCD	BCDL @BCDL	059	Supported	S: First source word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
2'S COMPLEMENT	NEG @NEG	160	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE 2'S COMPLEMENT	NEGL @NEGL	161	Supported	S: First source word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
16-BIT TO 32-BIT SIGNED BINARY	SIGN @SIGN	600	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
DATA DECODER	MLPX @MLPX	076	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	Δ
				C: Control word (digit designation)	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				R: First result word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	---
DATA ENCODER	DMPX @DMPX	077	Supported	S: First source word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
				C: Control word (digit designation)	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
ASCII CONVERT	ASC @ASC	086	Supported	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	Δ
				Di: Digit designator	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: First destination word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×
ASCII TO HEX	HEX @HEX	162	Supported	S: First source word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				Di: Digit designator	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con- stant	TIMER	COUNTER	Literal text strings
COLUMN TO LINE	LINE @LINE	063	Supported	S: First source word	×	×	×	×	×	×	Note	Note	❍	×	×	×	Note
				N: Bit number	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				D: Destination word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	×
LINE TO COLUMN	COLM @COLM	064	Supported	S: Source word	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10# +10# –10#	❍	❍	∆
				D: First destination word	×	×	×	×	×	×	Note	Note	❍	×	×	×	×
				N: Bit number	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
SIGNED BCD-TO-BINARY	BINS @BINS	470	Supported	C: Control word (data type)	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				S: Source word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	∆
				D: Destination word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	×
DOUBLE SIGNED BCD-TO-BINARY	BISL @BISL	472	Supported	C: Control word (data type)	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				S: First source word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
				D: First destination word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
SIGNED BINARY-TO-BCD	BCDS @BCDS	471	Supported	C: Control word (data type)	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				S: Source word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	∆
				D: Destination word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	×
DOUBLE SIGNED BINARY-TO-BCD	BDSL @BDSL	473	Supported	C: Control word (data type)	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				S: First source word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
				D: First destination word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
GRAY CODE CONVERSION	GRY @GRY	474	Supported (Rev. 2.1 or later)	C: Control word (data type)	×	×	×	×	×	×	Note	Note	∆	×	×	×	×
				S: First source word	×	❍	∆	❍	∆	∆	❍×	❍×	∆	❍ 16# 10#	❍	❍	×
				D: First destination word	×	×	❍	×	❍	❍	×	×	∆	×	×	×	×

Logic Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con- stant	TIMER	COUNTER	Literal text strings
LOGICAL AND	ANDW @ANDW	034	Supported	I1: Input 1	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10# +10# –10#	❍	❍	∆
				I2: Input 2	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10# +10# –10#	❍	❍	∆
				R: Result word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	×

Instruction	Mnemonic	Func-tion code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con-stant	TIMER	COUNTER	Literal text strings
DOUBLE LOGICAL AND	ANDL @ANDL	610	Supported	I1: Input 1 first word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				I2: Input 2 first word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
LOGICAL OR	ORW @ORW	035	Supported	I1: Input 1	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				I2: Input 2	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE LOGICAL OR	ORWL @ORWL	611	Supported	I1: Input 1 first word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				I2: Input 2 first word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
EXCLU-SIVE OR	XORW @XORW	036	Supported	I1: Input 1	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				I2: Input 2	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE EXCLU-SIVE OR	XORL @XORL	612	Supported	I1: Input 1 first word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				I2: Input 2 first word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
EXCLU-SIVE NOR	XNRW @XNRW	037	Supported	I1: Input 1	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				I2: Input 2	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE EXCLU-SIVE NOR	XNRL @XNRL	613	Supported	I1: Input 1 first word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				I2: Input 2 first word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10# +10# –10#	×	×	×
				R: First result word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
COMPLEMENT	COM @COM	029	Supported	Wd: Word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DOUBLE COMPLEMENT	COML @COML	614	Supported	Wd: First word	×	❍	Δ	❍	Δ	Δ	×	×	❍	×	×	×	×

Special Math Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
BIT COUNTER	BCNT @BCNT	067	Supported (Rev. 2.1 or later)	N: Number of words	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍	❍	❍	×
				S: First source word	×	×	×	×	×	×	×	×	Δ	×	❍	❍	×
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×

Floating-point Math Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
FLOATING TO 16-BIT	FIX @FIX	450	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	❍	❍	×
				R: Result word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	×	×	×
FLOATING TO 32-BIT	FIXL @FIXL	451	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				R: Result word	×	×	❍	×	❍	❍	Δ	Δ	Δ	×	×	×	×
16-BIT TO FLOATING	FLT @FLT	452	Supported (Rev. 2.1 or later)	S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍	❍	❍	×
				R: First result word	×	×	❍	×	❍	❍	Δ	Δ	Δ	×	×	×	×
32-BIT TO FLOATING	FLTL @FLTL	453	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	Δ	Δ	Δ	×	×	×	×
FLOATING-POINT ADD	+F @+F	454	Supported (Rev. 2.1 or later)	Au: First augend word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				Ad: First addend word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	Δ	Δ	Δ	×	×	×	×
FLOATING-POINT SUBTRACT	–F @–F	455	Supported (Rev. 2.1 or later)	Mi: First Minuend word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				Su: First Subtrahend word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	Δ	Δ	Δ	×	×	×	×
FLOATING-POINT MULTIPLY	*F @*F	456	Supported (Rev. 2.1 or later)	Md: First Multiplicand word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				Mr: First Multiplier word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	Δ	Δ	Δ	×	×	×	×
FLOATING-POINT DIVIDE	/F @/F	457	Supported (Rev. 2.1 or later)	Dd: First Dividend word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				Dr: First Divisor word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	Δ	Δ	Δ	×	×	×	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con-stant	TIMER	COUNTER	Literal text strings
DEGREES TO RADI-ANS	RAD @RAD	458	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
RADIANS TO DEGREES	DEG @DEG	459	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
SINE	SIN @SIN	460	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
COSINE	COS @COS	461	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
TANGENT	TAN @TAN	462	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
ARC SINE	ASIN @ASIN	463	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
ARC COSINE	ACOS @ACOS	464	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
ARC TAN-GENT	ATAN @ATAN	465	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
SQUARE ROOT	SQRT @SQRT	466	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
EXPONENT	EXP @EXP	467	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
LOGA-RITHM	LOG @LOG	468	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
EXPONEN-TIAL POWER	PWR @PWR	840	Supported (Rev. 2.1 or later)	B: First base word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				E: First exponent word	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				R: First result word	×	×	❍	×	❍	❍	∆	∆	∆	×	×	×	×
Floating Symbol Comparison	LD, AND, OR + =F, <>F, <F, <=F, >F, >=F	329 (=F) 330 (<>F) 331 (<F) 332 (<=F) 333 (>F) 334 (>=F)	Supported (Rev. 2.1 or later)	S1: Com-parison data 1	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×
				S2: Com-parison data 2	×	×	❍	×	❍	❍	∆	∆	∆	❍	×	×	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con- stant	TIMER	COUNTER	Literal text strings
FLOATING-POINT TO ASCII	FSTR @FSTR	448	Supported (Rev. 2.1 or later)	S: First source word	×	×	❍	×	❍	❍	Δ	Δ	Δ	❍	×	×	×
				C: Control word	×	×	×	×	×	×	Note	Note	Δ	×	×	×	×
				D: Destination word	×	×	×	×	×	×	Note	Note	❍	×	×	×	×
ASCII TO FLOATING-POINT	FVAL @FVAL	449	Supported (Rev. 2.1 or later)	S: Source word	×	Δ	Δ	Δ	Δ	Δ	Note	Note	❍	×	Δ	Δ	×
				D: First destination word	×	×	❍	×	❍	❍	Note	Note	Note	×	×	×	×
FIND MAXIMUM FLOATING-POINT	MAXF @MAXF	174	Supported (Rev. 2.1 or later)	C: Control word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				T: First word of table	×	Note	Note	Note	Note	❍	Note	Note	Note	×	Note	Note	Note
				D: Maximum destination word	×	❍	Δ	❍	Δ	❍	Δ	Δ	Δ	×	❍	❍	×
FIND MINIMUM FLOATING-POINT	MINF @MINF	175	Supported (Rev. 2.1 or later)	C: Control word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				T: First word of table	×	Note	Note	Note	Note	❍	Note	Note	Note	×	Note	Note	Note
				D: Minimum destination word	×	❍	Δ	❍	Δ	❍	Δ	Δ	Δ	×	❍	❍	×

Table Data Processing Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con- stant	TIMER	COUNTER	Literal text strings
SET STACK	SSET @SSET	630	Supported	TB: First stack address	×	×	×	×	×	×	Note	Note	Note	×	×	×	×
				N: Number of words	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
PUSH ONTO STACK	PUSH @PUSH	632	Supported	TB: First stack address	×	×	×	×	×	×	Note *	Note *	Note *	×	×	×	×
				S: Source word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
FIRST IN FIRST OUT	FIFO @FIFO	633	Supported	TB: First stack address	×	×	×	×	×	×	Note *	Note *	Note *	×	×	×	×
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	×
LAST IN FIRST OUT	LIFO @LIFO	634	Supported	TB: First stack address	×	×	×	×	×	×	Note *	Note *	Note *	×	×	×	×
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DIMENSION RECORD TABLE	DIM @DIM	631	Supported	N: Table number (Variable cannot be used)	×	×	×	×	×	×	×	×	×	❍ 10# or number only	×	×	×
				LR: Length of each record	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				NR: Number of records	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				TB: First table word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con- stant	TIMER	COUNTER	Literal text strings
SET RECORD LOCATION	SETR @SETR	635	Supported	N: Table number (Variable cannot be used)	×	×	×	×	×	×	×	×	×	❍ #10 or number	×	×	×
				R: Record number	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍	❍	❍	Δ
				D: Destination Index Register (Variable cannot be used)	×	×	×	×	×	×	×	×	×	×	×	×	×
GET RECORD NUMBER	GETR @GETR	636	Supported	N: Table number (Variable cannot be used)	×	×	×	×	×	×	×	×	×	❍ #10 or number	×	×	×
				IR: Index Register (Variable cannot be used)	×	×	×	×	×	×	×	×	×	×	×	×	×
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
DATA SEARCH	SRCH @SRCH	181	Supported	C: First control word (table length)	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
				R1: First word in range	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				Cd: Comparison data	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
SWAP BYTES	SWAP @SWAP	637	Supported	N: Number of words	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	16# 10#	❍	❍	Δ
				R1: First word in range	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×
FIND MAXIMUM	MAX @MAX	182	Supported	C: First control word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				R1: First word in range	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
FIND MINIMUM	MIN @MIN	183	Supported	C: First control word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				R1: First word in range	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
FIND DOUBLE-LENGTH MAXIMUM	MAXL @MAXL	176	Supported (Rev. 2.1 or later)	C: Control word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				T: First word of table	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				D: Maximum destination word	×	×	❍	×	❍	❍	Δ	Δ	Δ	×	❍	❍	×
FIND DOUBLE-LENGTH MINIMUM	MINL @MINL	177	Supported (Rev. 2.1 or later)	C: Control word	×	×	❍	×	❍	❍	×	×	❍	❍ 16#	×	×	×
				T: First word of table	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				D: Minimum destination word	×	❍	Δ	❍	Δ	❍	Δ	Δ	Δ	×	❍	❍	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
SUM	SUM @SUM	184	Supported	C: First control word	×	×	❍	×	❍	❍	×	×	❍	16# 10#	×	×	×
				R1: First word in range	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				D: First destination word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
FRAME CHECK SUM	FCS @FCS	180	Supported	C: First control word	×	×	❍	×	❍	❍	×	×	❍	16#	×	×	×
				R1: First word in range	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				D: First destination word (top row for byte designation, bottom row for word designation)	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
					×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
STACK SIZE READ	SNUM @SNUM	638	Supported	TB: First stack address	×	×	×	×	×	×	Note *	Note *	Note *	×	×	×	×
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
STACK DATA READ	SREAD @SREAD	639	Supported	TB: First stack address	×	×	×	×	×	×	Note *	Note *	Note *	×	×	×	×
				C: Offset value	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
STACK DATA OVERWRITE	SWRIT @SWRIT	640	Supported	TB: First stack address	×	×	×	×	×	×	Note *	Note *	Note *	×	×	×	×
				C: Offset value	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				S: Source data	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
STACK DATA INSERT	SINS @SINS	641	Supported	TB: First stack address	×	×	×	×	×	×	Note *	Note *	Note *	×	×	×	×
				C: Offset value	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				S: Source data	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10# +10# –10#	❍	❍	Δ
STACK DATA DELETE	SDEL @SDEL	642	Supported	TB: First stack address	×	×	×	×	×	×	Note *	Note *	Note *	×	×	×	×
				C: Offset value	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: Destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×

Data Control Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
PID CONTROL	PID	190	Supported (Rev. 2.1 or later)	S: Input word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
				C: First parameter word	×	×	×	×	×	×	Note	Note	Note	×	×	×	×
				D: Output word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
PID CONTROL WITH AUTO TUNING	PIDAT	191	Supported (Rev. 2.1 or later)	S: Input word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	×
				C: First parameter word	×	×	×	×	×	×	Note	Note	Note	×	×	×	×
				D: Output word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	×

Interrupt Control Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
SET INTERRUPT MASK	MSKS @MSKS	690	Supported	N: Interrupt identifier 1	×	×	×	×	×	×	×	×	×	❍ 10# or number	×	×	×
				S: Interrupt data 2	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
READ INTERRUPT MASK	MSKR @MSKR	692	Supported	N: Interrupt identifier	×	×	×	×	×	×	×	×	×	❍ 10# or number	×	×	×
				D: Destination word	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	×
CLEAR INTERRUPT	CLI @CLI	691	Supported	N: Interrupt identifier 1	×	×	×	×	×	×	×	×	×	❍ 10# or number	×	×	×
				S: Interrupt data 2	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
DISABLE INTERRUPTS	DI @DI	693	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
ENABLE INTERRUPTS	EI	694	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×

Basic I/O Unit Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UINT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
I/O REFRESH	IORF @IORF	097	Supported	St: Starting word	×	AT	AT∆	AT	AT∆	AT∆	AT∆	AT∆	AT∆	×	×	×	×
				E: End word	×	AT	AT∆	AT	AT∆	AT∆	AT∆	AT∆	AT∆	×	×	×	×
7-SEGMENT DECODER	SDEC @SDEC	078	Supported	S: Source word	×	❍	∆	❍	∆	∆	∆	∆	∆	×	❍	❍	∆
				Di: Digit designator	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				D: First destination word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×
INTELLIGENT I/O READ	IORD @IORD	222	Supported	C: Control data	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				S: Transfer source and number of words	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
				D: Transfer destination and number of words	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×
INTELLIGENT I/O WRITE	IOWR @IOWR	223	Supported	C: Control data	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				S: Transfer source and number of words	×	Note	Note	Note	Note	Note	Note	Note	Note	❍ 16# 10# +10# –10#	Note	Note	Note
				D: Transfer destination and number of words	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
CPU BUS UNIT I/O REFRESH	DLNK @DLNK	226	Supported	N: Unit number	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆

Serial Communications Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
PROTOCOL MACRO	PMCR @PMCR	260	Supported	C1:Control word 1 (come port No., serial port No., remote unit No.)	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				C2: Control word 2 (come sequence No.)	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	16# 10#	❍	❍	Δ
				S: First send word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	Note
				R: First receive word	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×

Network Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
SEND CIP COMMAND	CSND @CSND	489	Supported	S: First request word	×	× *	× *	× *	× *	× *	Note *	Note *	Note *	×	× *	× *	Note *
				D: First response word	×	× *	× *	× *	× *	× *	Note *	Note *	Note *	×	× *	× *	×
				C: First control word	×	× *	× *	× *	× *	× *	Note *	Note *	Note *	×	× *	× *	Note *
SEND MAIL	MLSND @MLSND	795	Supported	S: User-specified data	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				D: Destination address	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				C: Control data	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				O: Status output word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×

File Memory Instructions

Instruction	Mnemonic	Func-tion code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con-stant	TIMER	COUNTER	Literal text strings
READ DATA FILE	FREAD @FREAD	700	Supported	C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				S1: Number of read words and read start position	×	×	×	×	×	×	Note	Note	❍	×	×	×	Note
				S2: File-name	×	× *	× *	× *	× *	× *	Note *	Note *	❍	×	× *	× *	❍
				D: First destination word (Top row for reading data, bottom row for reading No. of data)	×	Note	Note	Note	Note	Note	Note	Note	Note	×	Note	Note	×
					×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
WRITE DATA FILE	FWRIT @FWRIT	701	Supported	C: Control word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D1: First destination word	×	×	×	×	×	×	Note	Note	❍	×	×	×	Note
				D2: File-name	×	× *	× *	× *	× *	× *	Note *	Note *	❍	×	× *	× *	❍
				S: First source word	×	Note *	Note *	Note *	Note *	Note *	Note *	Note *	Note *	×	Note *	Note *	Note *

Display Instructions

Instruction	Mnemonic	Func-tion code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con-stant	TIMER	COUNTER	Literal text strings
DISPLAY MESSAGE	MSG @MSG	046	Supported	N: Message number	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				M: First message word	×	×	×	×	×	×	×	×	❍	×	×	×	❍

Clock Instructions

Instruction	Mnemonic	Func-tion code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Con-stant	TIMER	COUNTER	Literal text strings
CALENDAR ADD	CADD @CADD	730	Supported	C: First calendar word	×	×	×	×	×	×	×	×	❍	×	×	×	×
				T: First time word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
				R: First result word	×	×	×	×	×	×	×	×	❍	×	×	×	×
CALENDAR SUBTRACT	CSUB @CSUB	731	Supported	C: First calendar word	×	×	×	×	×	×	×	×	❍	×	×	×	×
				T: First time word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
				R: First result word	×	×	×	×	×	×	×	×	❍	×	×	×	×
HOURS TO SECONDS	SEC @SEC	065	Supported	S: First source word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
				D: First destination word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
SECONDS TO HOURS	HMS @HMS	066	Supported	S: First source word	×	×	❍	×	❍	❍	×	×	❍	❍ 16# 10#	×	×	×
				D: First destination word	×	×	❍	×	❍	❍	×	×	❍	×	×	×	×
CLOCK ADJUSTMENT	DATE @DATE	735	Supported	S: First source word	×	×	×	×	×	×	×	×	❍	×	×	×	×

Debugging Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
TRACE MEMORY SAMPLING	TRSM	045	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×

Failure Diagnosis Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
FAILURE ALARM	FAL @FAL	006	Supported	N: FAL number	×	×	×	×	×	×	×	×	×	❍ 10#	×	×	×
				M: First message word or error code to generate(#0000 to #FFFF)	×	*	*	*	*	*	*	*	*	*	*	*	*
SEVERE FAILURE ALARM	FALS	007	Supported	N: FALS number (Variable cannot be used)	×	×	×	×	×	×	×	×	×	❍ 10#	×	×	×
				M: First message word or error code to generate(#0000 to #FFFF)	×	*	*	*	*	*	*	*	*	*	*	*	*

Other Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
SET CARRY	STC @STC	040	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×
CLEAR CARRY	CLC @CLC	041	Supported		×	×	×	×	×	×	×	×	×	×	×	×	×

Text String Processing Instructions

Instruction	Mnemonic	Func tion code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
MOV STRING	MOV$ @MOV$	664	Supported	S: First source word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				D: First destination word	×	×	×	×	×	×	×	×	❍	×	×	×	×
CONCATENATE STRING	+$ @+$	656	Supported	S1: Text string 1	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S2: Text string 2	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				D: First destination word	×	×	×	×	×	×	×	×	❍	×	×	×	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/ UNIT	DINT/ UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
GET STRING LEFT	LEFT$ @LEFT$	652	Supported	S1: Text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S2: Number of characters	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: First destination word	×	×	×	×	×	×	×	×	❍	×	×	×	×
GET STRING RIGHT	RGHT$ @RGHT$	653	Supported	S1: Text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S2: Number of characters	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: First destination word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
GET STRING MIDDLE	MID$ @MID$	654	Supported	S1: Text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S2: Number of characters	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				S3: Beginning position	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: First destination word	×	×	×	×	×	×	×	×	❍	×	×	×	×
FIND IN STRING	FIND$ @FIND$	660	Supported	S1: Source text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S2: Found text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				D: First destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
STRING LENGTH	LEN$ @LEN$	650	Supported	S: Text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				D: First destination word	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	×	❍	❍	×
REPLACE IN STRING	RPLC$ @RPLC$	661	Supported	S1: Text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S2: Replacement text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S3: Number of characters	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				S4: Beginning position	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: First destination word	×	×	×	×	×	×	×	×	❍	×	×	×	×
DELETE STRING	DEL$ @DEL$	658	Supported	S1: Text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S2: Number of characters	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				S3: Beginning position	×	❍	Δ	❍	Δ	Δ	Δ	Δ	Δ	❍ 16# 10#	❍	❍	Δ
				D: First destination word	×	×	×	×	×	×	×	×	❍	×	×	×	×

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
EXCHANGE STRING	XCHG$ @XCHG$	665	Supported	Ex1: First exchange word 1	×	×	×	×	×	×	×	×	❍	×	×	×	×
				Ex2: First exchange word 2	×	×	×	×	×	×	×	×	❍	×	×	×	×
CLEAR STRING	CLR$ @CLR$	666	Supported	S: Text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	×
INSERT INTO STRING	INS$ @INS$	657	Supported	S1: Base text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S2: Inserted text string first word	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S3: Beginning position	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 16# 10#	❍	❍	∆
				D: First destination word	×	×	×	×	×	×	×	×	❍	×	×	×	×
String Comparison	LD,AND, OR + =$,<>$,<$, <=$,>$,>=$	670 (=$) 671 (<>$) 672 (<$) 673 (<=$) 674 (>$) 675 (>=$)	Supported	S1: Text string 1	×	×	×	×	×	×	×	×	❍	×	×	×	❍
				S2: Text string 2	×	×	×	×	×	×	×	×	❍	×	×	×	❍

Task Control Instructions

Instruction	Mnemonic	Function code	Support	Operands	BOOL	WORD	DWORD	INT/UNIT	DINT/UDINT	REAL	ARRAY	STRUCT	STRING	Constant	TIMER	COUNTER	Literal text strings
TASK ON	TKON @TKON	820	Supported	N: Task number	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 10#	❍	❍	∆
TASK OFF	TKOF @TKOF	821	Supported	N: Task number	×	❍	∆	❍	∆	∆	∆	∆	∆	❍ 10#	❍	❍	∆

Variable Specification Methods and Precautions

Restrictions apply to the following instructions depending on the method used to specify the variables for the operand (instructions indicated by “Note” in the previous tables). The restrictions for each instruction are explained here.

ASC(086): Third Operand (D)

Depending on the contents of the second operand Di, the third operand D will use up to 3 words, and the range for the variable specified by D may be exceeded.

ASFT(017): Second and Third Operands (St, E)

If automatically allocated variables and addresses (including global variables with allocated addresses) are used in the second or third operand, an error will occur during building. If automatically allocated variables are used in both the second and third operand, a warning will occur during building. Regardless of the variable size specified for St, data in the words from the address specified in the second operand St until the address in the third operand E will be shifted. As a result, the variable range specified in St may be exceeded.

BCMP(068): Second Operand (T)

The size of the area specified in the second operand T is always 32 words. Therefore, if the variable size specified by T is smaller than 32 words, the corresponding variable range may be exceeded. Conversely, if the variable size specified by T is larger than 32 words, only part of the corresponding variable values will be changed.

BSET(071): Second and Third Operands (St, E)

If automatically allocated variables and addresses (including global variables with allocated addresses) are used in the second or third operand, an error will occur during building. If automatically allocated variables are used in both the second and third operand, a warning will occur during building. Regardless of the variable size specified in St, the value for the first operand S will be written in the words from the address specified in the second operand St until the address of the third operand E.

COLL(081): First Operand (Bs)

The variable data type used in the first operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the first operand, a warning will occur during building. Regardless of the data type in Bs, the data source will be calculated as Bs + Of. If the value for Of is larger than the variable size specified in Bs, the range of the corresponding variable will be exceeded.

COLM(064): Second Operand (D)

The area indicated by the second operand D is always 16 words.Therefore, if the variable size specified by D is smaller than 16 words, the corresponding variable range will be exceeded. Conversely, if the variable size specified by D is larger than 16 words, only part of the corresponding variable values will be changed.

CSND(489): First, Second, and Third Operands (S, D, C)

The size of the area for the first, second, and third operands (S, D, and C) changes. Therefore, the variable range may be exceeded depending on whether the first variable word for S, D, or C is specified. The variable data type used in the operands and the allocated size of the operands is not checked. If an automatically allocated variable is used in the operands, a warning will occur during building.

DIM(631): Fourth Operand (TB)

The table area size is determined by the second and third operands. The variable range specified in TB may be exceeded.

DIST(080): Second Operand (Bs)

The variable data type used in the second operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the second operand, a warning will occur during building. Regardless of the data type in Bs, the data source will be calculated as Bs + Of. If the value for Of is larger than the variable size specified in Bs, the corresponding variable range will be exceeded.

DMPX(077): First Operand (S)

Starting from the address in the first operand S, 4 words are used for 16-to-4 bit encoding, and 32 words are used for 256-to-8 bit encoding. Therefore, the variable range specified in S may be exceeded.

FAL(006): Second Operand (M)

The second operand M can use up to 2,048 words. Therefore, depending on the variable size specified in the operand, only the first word of the specified variable will be used, or the variable range may be exceeded. The variable data type used in the second operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the second operand, a warning will occur during building.

FALS(007): Second Operand (M)

The second operand M can use up to 2,048 words. Therefore, depending on the variable size specified in the operand, only the first word of the specified variable will be used, or the variable range may be exceeded. The variable data type used in the second operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the second operand, a warning will occur during building.

FCS(180): Second Operand (R1)

The FCS processing area is determined by the first operand C. Therefore, the variable size specified in R1 may be exceeded.

FIFO(633): First Operand (TB)

The variable range specified by TB may be exceeded depending on the size of the stack region. The variable data type used in the first operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the first operand, a warning will occur during building.

FREAD(700): Second, Third, and Fourth Operands (S1, S2, D)

The second operand S1 uses 4 words. The third operand S2 uses up to 37 words. The size for the fourth operand D is determined by the read data. Therefore, depending on the size of the variable specified in the operand, only the first word of the specified variable will be used, or the variable range may be exceeded. The variable data type used in the third operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the third operand, a warning will occur during building.

FSTR(448): Second and Third Operands (C, D)

The second operand C is always 3 words. Therefore, the variable range specified in C may be exceeded. The third operand D uses between 2 and 13 words depending on the specification method used in C. Therefore, the range of the variable specified in D may be exceeded.

FVAL(449): First Operand (S)

The maximum data length for the first operand S is 13 words. The corresponding variable range may be exceeded due to the variable size specified in S.

FWRIT(701): Second, Third, and Fourth Operands (D1, D2, S)

The second operand D1 uses 4 words. The third operand D2 uses up to 37 words. The size for the fourth operand S is determined by the write data. Therefore, depending on the size of the variable specified in the operand, only the first word of the specified variable will be used, or the variable range may be exceeded. The variable data type used in the third and fourth operands and the allocated size of the operands is not checked. If an automatically allocated variable is used in the third and fourth operands, a warning will occur during building.

GRY(474): First and Second Operands (C, S)

The first operand C is always 3 words. Therefore, the corresponding variable area will only be partly used or the variable range may will be exceeded. The third operand D uses between 2 and 13 words depending on the specification method used in C. Therefore, the range of the variable specified in D may be exceeded. The second operand S is always 1 word. Therefore, the corresponding variable range will be exceeded depending on the size of the variable specified in S.

HEX(162): First Operand (S)

The first operand S may use up to 3 words, depending on the contents in the second operand Di. As a result, the variable range specified in S may be exceeded.

IORD(222): Third Operand (D)

The region specified by the third operand D is determined according to the second operand S. Depending on the size of the variable specified in D, only the first word will be used, or the variable range will be exceeded.

IOWR(223): Third Operand (D)

The region specified by the third operand D is determined according to the second operand S. Depending on the size of the variable specified in D, only the first word will be used, or the variable range will be exceeded.

LIFO(634): First Operand (TB)

The variable range specified by TB may be exceeded depending on the size of the stack region. The variable data type used in the first operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the first operand, a warning will occur during building.

LINE(064): First Operand (S)

The area indicated by the first operand S is always 16 words.Therefore, if the variable size specified by S is smaller than 16 words, the corresponding variable range will be exceeded. Conversely, if the variable size specified by S is larger than 16 words, only part of the corresponding variable values will be changed.

MAX(182): Second Operand (R1)

The variable size specified in S may be exceeded, depending on the specification in the first operand C.

MAXF(174): Second Operand (T)

The variable size specified in T may be exceeded, depending on the specification in the first operand C.

MAXL(176): Second Operand (T)
The variable size specified in T may be exceeded, depending on the specification in the first operand C.

MCMP(019): First and Second Operands (S1, S2)
The area indicated by the first and second operands (S1, S2) is always 16 words.Therefore, if the variable size specified by S1 and S2 is smaller than 16 words, the corresponding variable range will be exceeded. Conversely, if the variable size specified by S1 and S2 is larger than 16 words, only part of the corresponding variable values will be changed.

MIN(183): Second Operand (R1)
The variable size specified in R1 may be exceeded, depending on the specification in the first operand C.

MINL(177): Second Operand (T)
The variable size specified in T may be exceeded, depending on the specification in the first operand C.

MINF(175): Second Operand (T)
The variable size specified in T may be exceeded, depending on the specification in the first operand C.

MLPX(076): Third Operand (R)
Starting from the address in the third operand R, 4 words are used for 4-to-26 bit decoding, and 32 words are used for 8-to-256 bit decoding. Therefore, the variable range specified in R may be exceeded.

NSFL(578): First Operand (D)
The variable range specified in the first operand D may be exceeded depending on the combination of the second operand C and third operand N.

NSFR(579): First Operand (D)
The variable range specified in the first operand D may be exceeded depending on the combination of the second operand C and third operand N.

PID(190): Second Operand (C)
The second operand is always 39 words. Therefore, if the variable specified in C is larger than 39 words, only the first 39 words of the variable will be used. If the variable is less than 39 words, the variable range will be exceeded.

PIDAT(260): Second Operand (C)
The second operand is always 40 words. Therefore, if the variable specified in C is larger than 40 words, only the first 40 words of the variable will be used. If the variable is less than 40 words, the variable range will be exceeded.

PMCR(260): Third and Fourth Operands (S, R)
The third and fourth operands S and R store the area size specified in the first word. Therefore, depending on the area size, only the first word of the variable specified in S or R will be used, or the variable range will be exceeded. The variable data type used in the third and fourth operands and the allocated size of the operands is not checked. If an automatically allocated variable is used in the third and fourth operands, a warning will occur during building.

PUSH(632): First Operand (TB)
The variable range specified by TB may be exceeded depending on the size of the stack region. The variable data type used in the first operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the first operand, a warning will occur during building.

RSTA(531): First Operand (D)
When using a boolean type array, the start position of the array cannot be specified (Var[0] may not be bit 0), and, therefore, may be written to an area outside the variable range. The range of the corresponding variable may be exceeded for arrays of all data types depending on the value of the third operand N2.

SDEC(078): Third Operand (D)

Depending on the specification of the second operand Di, 3 words may be used for the region specified by the third operand D. Depending on the variable size specified by D, only the first 2 or 3 words will be used, or the variable range will be exceeded.

SDEL(642): First Operand (TB)

The variable range specified by TB may be exceeded depending on the size of the stack region. The variable data type used in the first operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the first operand, a warning will occur during building.

SETA(530): First Operand (D)

When using a boolean type array, the start position of the array cannot be specified (Var[0] may not be bit 0), and, therefore, may be written to an area outside the variable range. The range of the third operand N2 is 0 to 65,535 bits. Therefore, the corresponding variable range may be exceeded.

SFTR(084): Second and Third Operands (St, E)

If automatically allocated variables and addresses (including global variables with allocated addresses) are used in the second or third operand, an error will occur during building. If automatically allocated variables are used in both the second and third operand, a warning will occur during building. Regardless of the variable size specified in St, data in the words from the address specified in the second operand St until the address in the third operand E will be shifted. As a result, the variable range specified in St may be exceeded.

SINS(641): First Operand (TB)

The variable range specified by TB may be exceeded depending on the size of the stack region. The variable data type used in the first operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the first operand, a warning will occur during building.

SLD(074): First and Second Operands (St, E)

If automatically allocated variables and addresses (including global variables with allocated addresses) are used in the first or second operand, an error will occur during building. If automatically allocated variables are used in both the first and second operand, a warning will occur during building. Regardless of the variable size specified in St, data in the words from the address specified in the first operand St until the address in the second operand E will be shifted. As a result, the variable range specified in St may be exceeded.

SNUM(638): First Operand (TB)

The variable range specified by TB may be exceeded depending on the size of the stack region. The variable data type used in the first operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the first operand, a warning will occur during building.

SRCH(181): Second Operand (R1)

The search region in the second operand depends on the first operand C. Therefore, the variable range specified in R1 may be exceeded.

SRD(075): First and Second Operands (St, E)

If automatically allocated variables and addresses (including global variables with allocated addresses) are used in the first or second operand, an error will occur during building. If automatically allocated variables are used in both the first and second operand, a warning will occur during building. Regardless of the variable size specified in St, data in the words from the address specified in the first operand St until the address in the second operand E will be shifted. As a result, the variable range specified in St may be exceeded.

SREAD(639): First Operand (TB)

The variable range specified by TB may be exceeded depending on the size of the stack region. The variable data type used in the first operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the first operand, a warning will occur during building.

SSET(630): First Operand (TB)

The stack region size is the sum of 4 words and the words in the second operand N. Therefore, the variable range specified in TB may be exceeded.

SUM(184): Second Operand (R1)

The sum value processing region depends on the first operand C. Therefore, the variable size specified by R1 may be exceeded.

SWAP(637): Second Operand (R1)

The swap region depends on the first operand N. Therefore, the variable range specified by R1 may be exceeded.

SWRIT(640): First Operand (TB)

The variable range specified by TB may be exceeded depending on the size of the stack region. The variable data type used in the first operand and the allocated size of the operand is not checked. If an automatically allocated variable is used in the first operand, a warning will occur during building.

TCMP(085): Second Operand (T)

The area indicated by the second operand T is always 16 words.Therefore, if the variable size specified by T is smaller than 16 words, the corresponding variable range will be exceeded. Conversely, if the variable size specified by T is larger than 16 words, only part of the corresponding variable values will be changed.

WSFT(016): Second and Third Operands (St, E)

If automatically allocated variables and addresses (including global variables with allocated addresses) are used in the second or third operand, an error will occur during building. If automatically allocated variables are used in both the second and third operand, a warning will occur during building. Regardless of the variable size specified in St, data in the words from the address specified in the second operand St until the address in the third operand E will be shifted. As a result, the variable range specified in St may be exceeded.

XFER(070): Second and Third Operands (S, D)

The second and third operands can be specified in up to 65,535 words depending on the value for the number of words to be transferred in N. Therefore, if the variable size specified in S and D is smaller than the value specified in N, the corresponding variable area may be exceeded. Conversely, if the variable size specified for S and D is larger than the value specified in N, only part of the corresponding variable values will be changed.

XFRB(062): Second and Third Operands (S, D)

Bits in up to 16 words can be changed for the second and third operands depending on the control word C. Therefore, if the variable size specified in S and D is smaller than the value specified in C, the corresponding variable area may be exceeded. Conversely, if the variable size specified for S and D is larger than 16 words, only part of the corresponding variable values will be changed.

Appendix H

CIP Messages Sent Using CSND Instructions

This document describes CIP message transmission for NE1S Series by using CSND instructions.

CIP Object on page 579 through *Example of Use for NE1S Series* on page 585 describe the basic information required for CIP to use CSND instructions to help you understand CSND instruction specifications deeply.

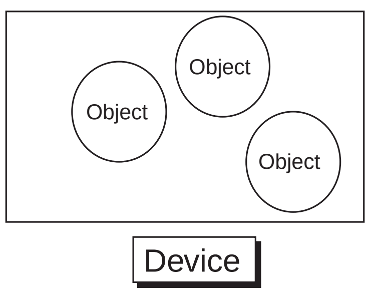
Refer to *Data Access for NE1S Series* on page 589 when the using CSND instructions.

Refer to *Data Type* on page 607 and *Response Code* on page 610 as required for a lists of data types and error codes supported by NE1S Series.

CIP Object

Object Model

Each device is modeled as a group of “Objects” in the conception of CIP. Object represents something that a particular element of a device is abstracted.



You should access to each Object when accessing from the outside.

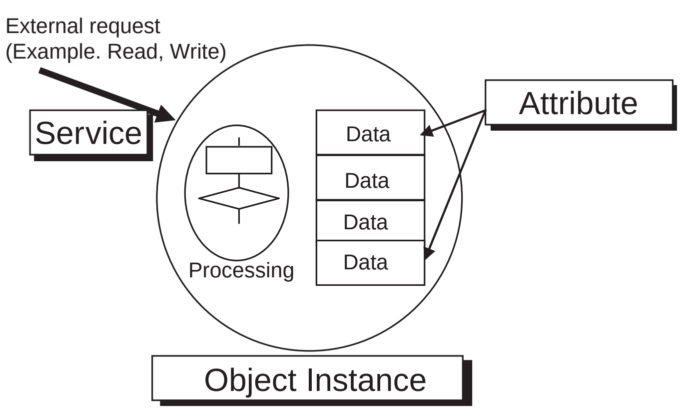
Object represents the processing and the data resulted from abstraction of a function in the device.

A request from the outside of Object, such as Read Data, is called “Service.”

Data belonging to Object is called “Attribute.”

The entity of Object is called “Instance” or “Object Instance.”

When Object is generalized, it is called “Class.” For example, “Japan” is one of Instances (Object Instances) of Class “Nation.”



Reference Information

In CIP Common Specifications, “Object,” “Class,” “Instance,” “Attribute” and “Service” are explained as follows:

(Extracts from CIP Common Specifications)

Object	An abstract representation of a particular component within a product.
Class	A set of objects that all represent the same kind of system component. A class is a generalization of an object. All objects in a class are identical in form and behavior, but may contain different attribute values.
Instance	A specific and real (physical) occurrence of an object. For example: NewZealand is an instance of the object class Country. The terms Object, Instance, and Object Instance all refer to a specific Instance.
Attribute	A description of an externally visible characteristic or feature of an object. Typically, attributes provide status information or govern the operation of an Object. For example: the ASCII name of an object; and the repetition rate of a cyclic object.
Service	A function supported by an object and/or object class. CIP defines a set of common services and provides for the definition of Object Class and/or Vendor Specific services.

Designation of Object Address

This is the concept to access to Object or Attribute.

Each Object Class has “Class ID”.

There are two types of “Class ID”; one is standardized by ODVA and the other is decided independently by each device vendor.

Each Object Instance also has ID. This is called “Instance ID.” Different Instance ID is assigned to each Object. As for Object Class standardized by ODVA, Instance ID is given to it according to the ODVA method. On the other hand, vendor's own Instance ID is decided independently by the vendor.

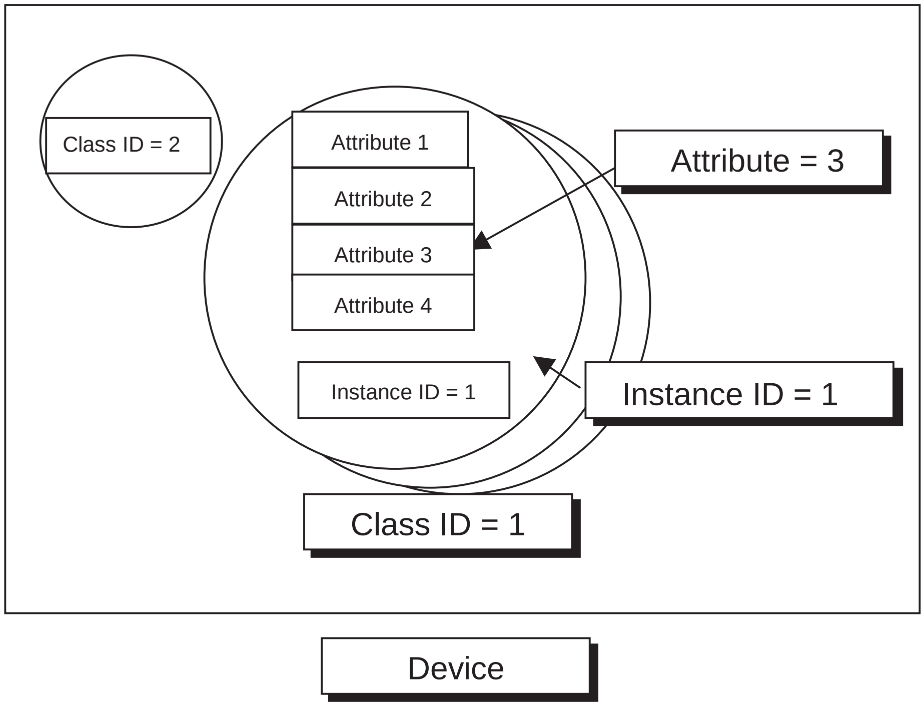
Each Attribute also has “Attribute ID.”

Each Object is accessed to by using “Class ID,” “Instance ID,” and “Attribute ID.”

In the device, you can designate Object by specifying these three IDs.

When requesting “Service,” you should specify “Class ID,” “Instance ID,” and “Attribute ID.” (Instance ID and Attribute ID may not be required, depending on the Service.)

These three IDs are called “IOI (Internal Object Identifier)” because they identify the location of Object in the device.



Link Path

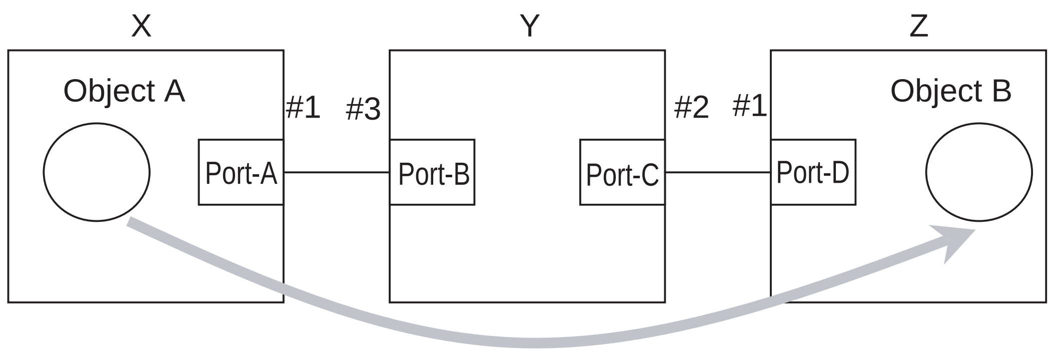
Link Path

For CIP, different from the internet protocol, the relay route from the transmission node to the reception node is all described in the transmission frame.

The described route is called “Link Path.” Link Path is described as “EPATH type.”

The conception of Link Path is as follows:

First of all, designate a network port of a transmission channel with the destination network, and designate node address on that network, which is called Link Address. For the relay channel, similarly, designate a network port with the destination network and node address on that network. Then, repeat the same procedure to the final destination.



When sending data from X to Z.

Link Path = Port A: #3, Port C: #1

Send data from the network port of X (Port-A) to #3 on that circuit, and the data reaches Y. Then, send it from the network port of Y (Port-C) to #1 on that circuit. Through this procedure, the destination node Z can be designated.

Description by EPATH Type

For CIP, EPATH type is employed for describing Link Path and IOI.

This is the method of dividing Link Path or IOI into segments and assigning a value to each of them.

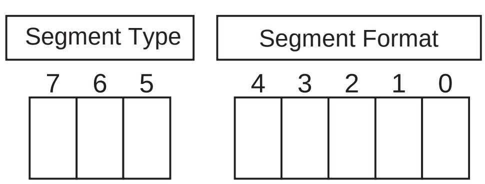
Therefore, Link Path description indicates the final destination by joining data called segments.

The segment includes the segment type information and the segment data.

Segment 1	Segment 2	Segment 3	Segment 4	· · · ·

Details of Segment Type

The interpretation method of a segment is included in the first 1 byte, which consists of tow parts; 3 bits of "Segment Type" and 5 bits of "Segment Format."



According to CIP Specifications, the Segment Type specifications are decided as follows:

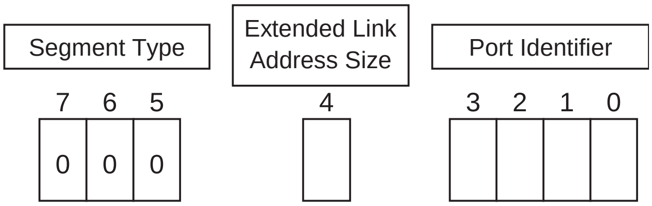
Segment Type			Description
7	6	5	
0	0	0	Port Segment
0	0	1	Logical Segment
0	1	0	Network Segment
0	1	1	Symbolic Segment
1	0	0	Data Segment
1	0	1	Data Type
1	1	0	Data Type
1	1	1	Reserved

The specifications of Segment Format are different for each Segment Type.

The following sections describe Port Segment, Logical Segment, and Data Segment which are to be required for using CSND instructions.

Port Segment

Port Segment is employed for describing the above-mentioned path.

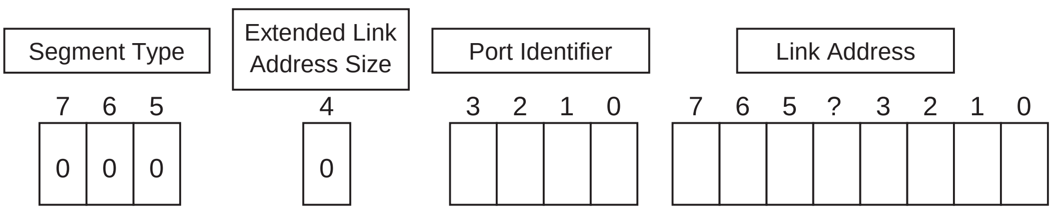


Set ID of that port in Port Identifier.

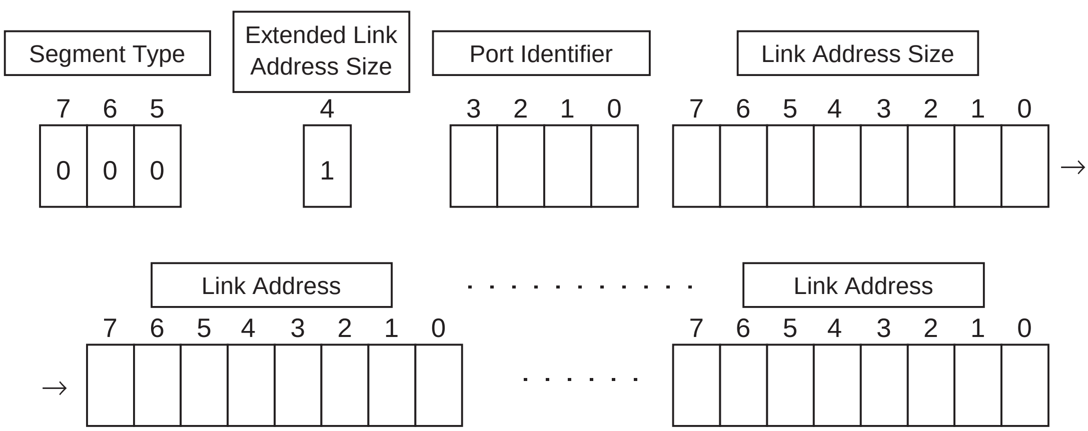
Port Identifier is 4-bit, so that it can take a value of 0 to 15. “0” is reserved and not available. “1” is to indicate the backplane port. “15” has a special meaning, indicating that the size of Port Identifier is larger than 1 byte. In this case, Port Identifier is followed by 2-byte Port Identifier. This case is not explained here because, for EN1S Series, Port Identifier will not exceed 1 byte.

Set “1” in Extended Link Address Size when Link Address of that port is larger than 1 byte.

Shown below is the description method of Port Segment when “0” is set in Extended Link Address Size.



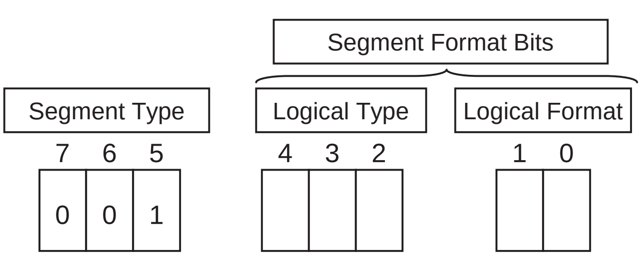
Designate the size of Link Address when “1” is set in Extended Link Address Size. Shown below is the description method of Port Segment.



Set even-number byte in Link Address without fail. If it is an odd number, surely change it to an even number by padding with “00.”

Logical Segment

Logical Segment is employed for describing IOI.



Logical Type			Description
4	**3**	**2**	
0	0	0	Class ID
0	0	1	Instance ID
0	1	0	Member ID
0	1	1	Connection Point
1	0	0	Attribute ID
1	0	1	Special (Do not use the logical addressing definition for the Logical Format.)
1	1	0	Service ID (Do not use the logical addressing definition for the Logical Format.)
1	1	1	Reserved

Logical Format		Description
1	**0**	
0	0	8-bit logical address
0	1	16-bit logical address
1	0	32-bit logical address
1	1	Reserved

The 32-bit logical address of Logical Format is reserved and not available.

The 8-bit and 16-bit logical addresses are available for Class ID and Instance ID which indicate IOI.

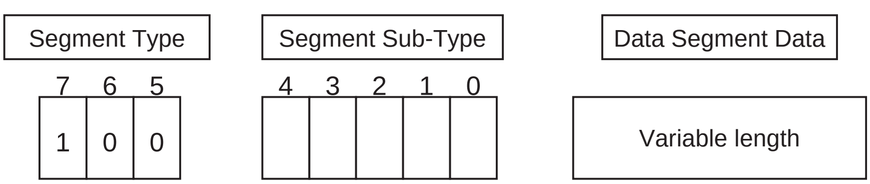
The 8-bit logical address is available for Attribute ID.

This is used for requesting Service to an optional Object of an optional device.

This can be also used for directly reading/writing IO memory of NE1S by designating the address.

Data Segment

Data Segment is employed for reading/writing variables of CPU Unit.

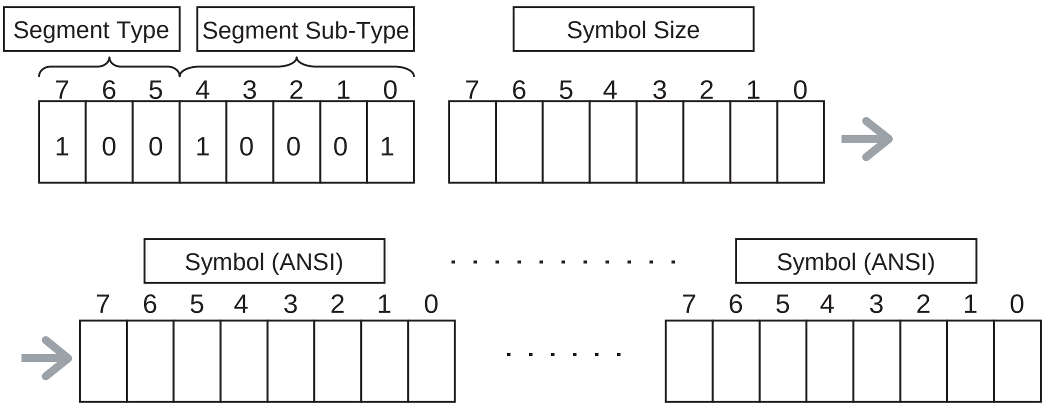


Segment Sub-Type					Description
4	**3**	**2**	**1**	**0**	
0	0	0	0	0	Simple Data Segment
1	0	0	0	1	ANSI Extended Symbol Segment
					All Segment Sub-Types are reserved except the above.

ANSI Extended Symbol Segment is mainly used for Data Segment.

Variable data is read/written by using the segment of this type.

ANSI Extended Symbol Segment



Variables and IOI

As mentioned above, Object Model is used in the concept of CIP. A device is recognized as a group of Objects. You have to identify the data you want to access to by using Class/Instance.

When accessing to a variable by using a variable name, the variable name must be converted to Class/ Instance. For NE1S Series, CPU Unit performs the conversion process, so that the user doesn't have to do it. All the user need to do is just to designate a variable in spite of Class/Instance.

Example of Use for NE1S Series

Setup of Link Path

Port Number

Described below is the network port for designating Link Path.

For NE1S Series, the base unit (backplane) is also recognized as part of the network when designating Link Address.

CPU Unit

CPU Unit has two ports.

One is a backplane port and the other is Ethernet port.

The backplane port is CS1 bus (base unit). Communication from CPU Unit via a CPU Bus Unit surely goes through the backplane.

Port	Port Number
Backplane	1
Ethernet	2

ControlNet Unit

ControlNet Unit has two ports.

One is a backplane port and the other is ControlNet port.

The backplane port is CS1 bus (base unit). Communication via CPU Unit or other CPU Bus Unit surely goes through the backplane.

Port	Port Number
Backplane	1
ControlNet	2

DeviceNet Unit

DeviceNet Unit has two ports.

One is a backplane port and the other is DeviceNet port.

The backplane port is CS1 bus (base unit). Communication via CPU Unit or other CPU Bus Unit surely goes through the backplane.

Port	Port Number
Backplane	1
DeviceNet	2

Link Address

Link Address is a node address on the network for designating Link Path.

The method to set up Link Address is different for each network.

For NE1S Series, the base unit (backplane) is also recognized as part of the network when designating Link Address.

Backplane (Base Unit)

- **CPU Bus Unit**
  For NE1S Series, the base unit is recognized as a backplane port.
  CPU Bus Units, such as ControlNet Unit or DeviceNet Unit, are also recognized as nodes on the backplane port.
  Link Address of a CPU Bus Unit, on the backplane, is “Unit No. + 10 hex.” For example, when Unit No. is 0, Link Address is 10 hex. When Unit No. is F, Link Address is 1F hex.
- **CPU Unit**
  Link Address of CPU Unit, on the backplane, will be surely 00 hex.
  IP address is employed to describe Link Address of Ethernet port of CPU Unit, which is the same as that of Ethernet port mentioned below.

Network

- **Ethernet**
  IP address is employed to describe Link Address of Ethernet port.
  All of IP address must be described with ASCII code.
  For example, IP address of 192.168.200.200 will be [31] [39] [32] [2E] [31] [36] [38] [2E] [32] [30] [30] [2E] [32] [30] [30].
- **ControlNet**
  The node address of ControlNet is described as Link Address of ControlNet port.
  The node address of ControlNet is “1” to “99” (01 hex to 63 hex). For ControlNet, Link Address doesn't have to be described with ASCII code.
- **DeviceNet**
  The node address of DeviceNet is described as Link Address of DeviceNet port.
  The node address of DeviceNet is “0” to “63” (00 hex to 3F hex). For DeviceNet, Link Address doesn't have to be described with ASCII code.

Example of Path

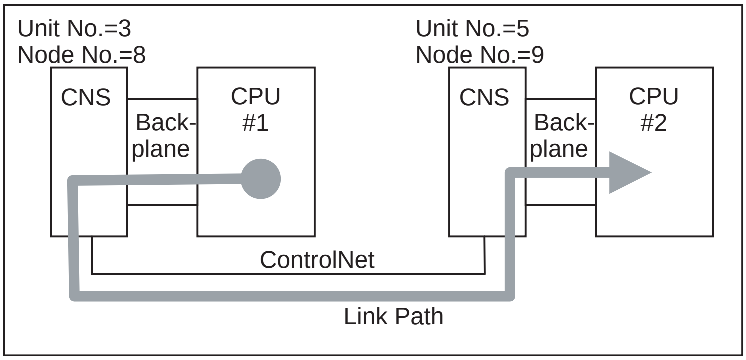
Example 1:

In case of accessing from CPU#1 to CPU#2 via ControlNet Unit

[Structure]

ControlNet Unit of CPU#1: Unit No. = 3, Node Address = 8

ControlNet Unit of CPU#2: Unit No. = 5, Node Address = 9



The route is as follows:

Backplane Port of CPU#1 → ControlNet Unit (Unit No. = 3) → ControlNet Port of ControlNet Unit → ControlNet Unit (Node Address =9) → Backplane Port of ControlNet Unit → CPU#2

The route above is explained below.

Port Segment is used for setting Link Path. Therefore, the top 3 bits (Segment Type) of the first byte will be “0” inevitably.

The backplane port of CPU#1 comes first. Port No. of the backplane port is “1.” Because it falls into 1 byte, Extended Link Address Size will be “0.”

Therefore, the first byte will be [01].

Unit No. of ControlNet Unit of CPU#1 is 03 hex, so that Link Address will be

10 hex + 03 hex=13 hex. Therefore, the second will be [13].

Port No. of ControlNet Port of ControlNet Unit is “2,” so that the third will be [02]. Node Address of the target ControlNet Unit (ControlNet Unit of CPU#2) is “9,” so that the fourth byte will be [09].

Link Address of CPU Unit, on the backplane, is “0.” Therefore, the fifth and sixth bytes will be [01] and [00], respectively.

Link Path will be as follows:

[01] [13] [02] [09] [01] [00]

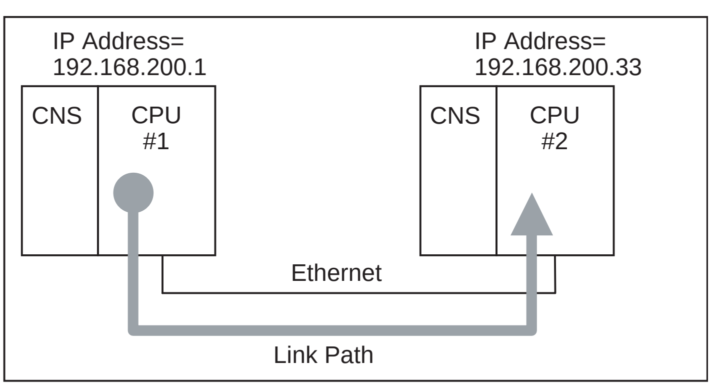
Example 2:

In case of accessing from CPU#1 to CPU#2 via Ethernet

[Structure]

IP Address of CPU#1 = 192.168.200.1

IP Address of CPU#2 = 192.168.200.33



The route is as follows:

Ethernet Port of CPU#1 → CPU#2 (IP Address = 192.168.200.33)

The route above is explained below.

Port Segment is used for setting Link Path. Therefore, the top 3 bits (Segment Type) of the first byte will be “0” inevitably.

In this example, data is transmitted via Ethernet Port. Port No. of Ethernet Port is “2.”

IP Address is used for describing Link Address on Ethernet. IP Address is larger than 1 byte, so that Extended Link Address Size will be “1.” Therefore, the first byte will be [12].

Link Address Size falls into the second byte. IP Address of the target CPU#2 is “192.168.200.33.” The number of the letters of this IP Address should be counted, including dot (“.”) because the whole IP Address must be described with ASCII code. In this case, there are 14 letters (= 0E hex).

Link Path is as follows:

[12] [0E] [31] [39] [32] [2E] [31] [36] [38] [2E] [32] [30] [30] [2E] [33] [33]

Example 3:

In case of accessing from CPU#1 to CPU#3 via CPU#2 and ControlNet Unit

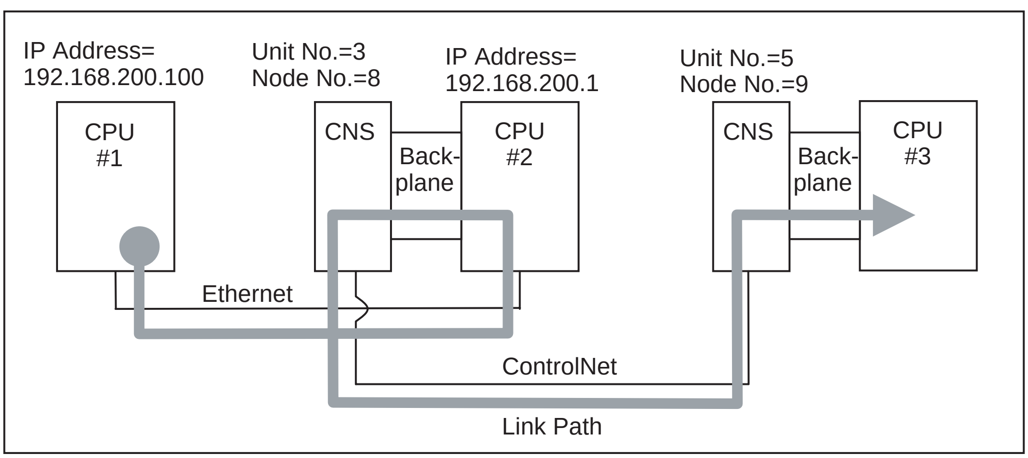
[Structure]

IP Address of CPU#1 = 192.168.200.100

IP Address of CPU #2 = 192.168.200.1

ControlNet Unit of CPU #2: Unit No. = 3, Node Address = 8

ControlNet Unit of CPU #3: Unit No. = 5, Node Address = 9



The route is as follows:

Ethernet Port of CPU#1 → CPU#2 (IP Address = 192.168.200.1) → ControlNet Unit (Unit No. = 3) → ControlNet Port of ControlNet Unit → ControlNet Unit (Node Address = 9) → Backplane Port of ControlNet Unit → CPU#3

The route above is explained below.

Port Segment is used for setting Link Path. Therefore, the top 3 bits (Segment Type) of the first byte will be “0” inevitably.

In this example, data is sent via Ethernet Port. Port No. of Ethernet Port is “2.”

IP Address is employed for Link Address on Ethernet. Because IP Address is larger than 1 byte, Extended Link Address Size will be “1.” Therefore, the first byte will be [12].

Link Address Size falls into the second byte. IP Address of the target CPU#2 is “192.168.200.1.” The number of the letters of this IP Address should be counted, including dot (“.”) because the whole IP Address must be described with ASCII code. In this case, there are 13 letters (= 0D hex). Thirteen (byte) is an odd number, so that [00] should be added at the end in order to make it an even number.

So far, Link Path is [12] [0D] [31] [39] [32] [2E] [31] [36] [38] [2E] [32] [30] [30] [2E] [31] [00].

Now data reaches CPU#2. Then, it is sent to CPU#3 via ControlNet Unit of CPU#2.

In order to go through ControlNet Unit of CPU#2, data must go through Backplane Port of CPU#2. Port No. of Backplane Port is inevitably "1." Because it falls into 1 byte, Extended Link Address Size will be "0." Therefore, the first byte will be [01].

Link Address is 10 hex + 03 hex=13 hex because Unit No. of ControlNet Unit of CPU#2 is 03 hex, so that the second byte will be [13].

Because Port No. of ControlNet Port of ControlNet Unit is "2," the third byte will be [02]. Node Address of the target ControlNet Unit (ControlNet Unit of CPU#3) is "9," so that the fourth byte will be [09].

Link Address of CPU Unit, on Backplane, is "0," so that the fifth and the sixth bytes will be [01] and [00], respectively.

Link Path of this part will be [01] [13] [02] [09] [01] [00].

The above-mentioned two Link Paths being joined, the whole Link Path is as follows:

[12] [0D] [31] [39] [32] [2E] [31] [36] [38] [2E] [32] [30] [30] [2E] [31] [00] [01] [13] [02] [09] [01] [00]

Designation of Variable Name

ANSI Extended Symbol Segment of Data Segment is employed for designating Variable Name.

Example 1: A variable of "ABCDE" is designated by the following description. The last part, [00], is the padding for making the number of bytes an even number.

[91] [05] [41] [42] [43] [44] [45] [00]

Example 2: The whole structure of an array variable is the same as in Example 1.

The whole of an array variable, ABC [10], is as follows:

[91] [03] [41] [42] [43] [00]

Example 3: An element of an array variable is treated as "a member of Instance." Member ID of Logical Segment is employed for describing Instance Member.

An element of an array variable, ABC [3], is described as follows:

[91] [03] [41] [42] [43] [00] [28] [03]

Example 4: For describing a member of a variable of the structured data type, ANSI Extended Symbol Segment is employed for every dot, ".", which divides members of a structure.

ABC.DE is described as follows:

[91] [03] [41] [42] [43] [00] [91] [02] [44] [45]

Data Access for NE1S Series

CSND Instruction

Overview of CSND Instruction

For NE1S Series, variable data can be accessed to by using CSND Instruction.

CSND Instruction is an instruction to send/receive a message of CIP.

CSND Instruction has three parameters; S data, D data, and C data.

Service data to be sent and transmission control information such as address are set up in S data and C data, respectively. When CSND Instruction is executed, received response data is stored in D data.

FUNC No.	Mnemonic	Instruction Name	Overview of Function
489	(@)CSND	CIP Transmission Instruction	Instruction to send CIP Explicit Message

Execution Condition/Immediate Refreshing Specification		
Execution Condition	Execute cyclically every time it goes to ON.	CSND
	Execute 1 cycle at rising up.	@CSND
	Execute 1 cycle at falling down.	Not supported.
Immediate Refreshing Specification		Not supported.

(@)CSND(489)	
S	Data to be sent
D	Data received.
C	Control data

S Data Details

Offset (word)	
S	Data Size to be sent
S+1	Service Code
S+2	Service Data
:	:

Data Size to be sent	Data Length of Service Code and the following data (unit: byte)
Service Code	CIP Explicit Message Service Code
Service Data	Service Data of Explicit Message Service of CIP Content is different for each Service Code. Service Data is stored in the order from lower byte to upper byte.

D Data Details

Offset (word)	
D	Size of Data received
D+1	Service Code
D+2	General Status
D+3	Additional Status
D+4	Response Data
:	:

Size of Data received	Data Length of Service Code and the following data (unit: byte)
Service Code	Transmitted Explicit Message Service Code of CIP The 8th bit is turned ON. For example, when Service Code is 4C hex, it will be CC hex.
General Status	Execution result of the transmitted Service 00 hex indicates a correct end. Other values indicate an error.
Additional Status	Additional information of General Status When Additional Status is larger than 1 Word, only the top 1 Word is stored. Content is different for each Object addressed and Service.
Response Data	Response Data received Received data is stored in the order from lower byte to upper byte.

C Data Details

Offset (word)				
C	Reception Buffer Size			
C+1	0	Comm. Port NO.	0	0
C+2	0	0	0	0
C+3	Service Execution Time			
C+4	8	0	IOI Size (N)	
C+5	9	1	Variable Name Size	
C+6	Variable Name_1		Variable Name _2	
:	:		:	
C+(4+N)	Variable Name _X		Variable Name _Y	
C+(4+N)+1	Link Path Size (M)			
C+(4+N)+2	LinkPath_1		LinkPath_1	
:	:		:	
C+(4+N)+1+M	LinkPath_X		LinkPath_Y	

Reception Buffer Size	Size of area for storing received data (area specified by D data). (Unit: WORD) When a response larger than the reception buffer size is received, excess part of the received data will be annulled.
Comm. Port NO.	Internal logic port number. There are 8 ports, 0 to 7.
Service Execution Time	Service execution time in the other node. Usually, 0000 hex should be specified.
Variable Name Size	Size of Variable Name to be accessed to. (Unit: Byte) When Variable Name is “ABC,” Variable Name Size will be “3.”
Variable Name_1 to Variable Name _Y	Variable Name to be accessed to. It should be designated with ASCII Code. When Variable Name is “ABC,” Variable Name_1 = 41 hex, Variable Name_2 = 42 hex, Variable Name _3 = 43 hex, and Variable Name _4 = 00 hex.
Link Path Size	Specifies length of Link Path. (Unit: WORD)
Link Path_1 to LinkPath_Y	Link Path

Communication Port Number

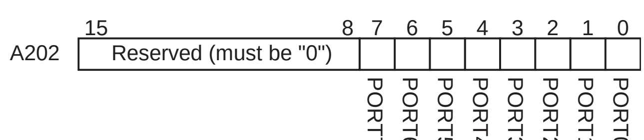
Eight logical communication ports are provided. Eight communication instructions can be executed simultaneously.

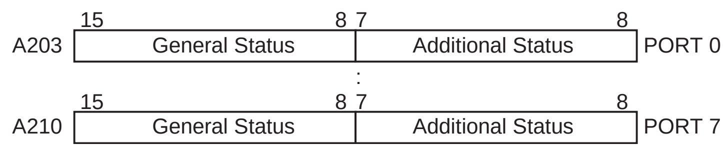
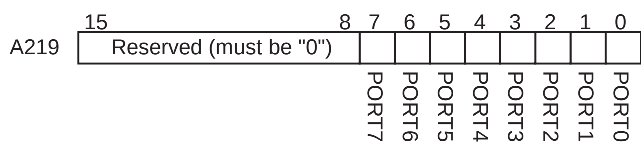
It is one instruction only that can be executed at one time in one communication port. When executing 9 or more communication instructions, you have to prepare exclusive control.

This communication port number is shared with Network Communication Instruction (CSND) and Protocol Macro Instruction (PMCR). Therefore, you must be careful not to designate the same number for these instructions.

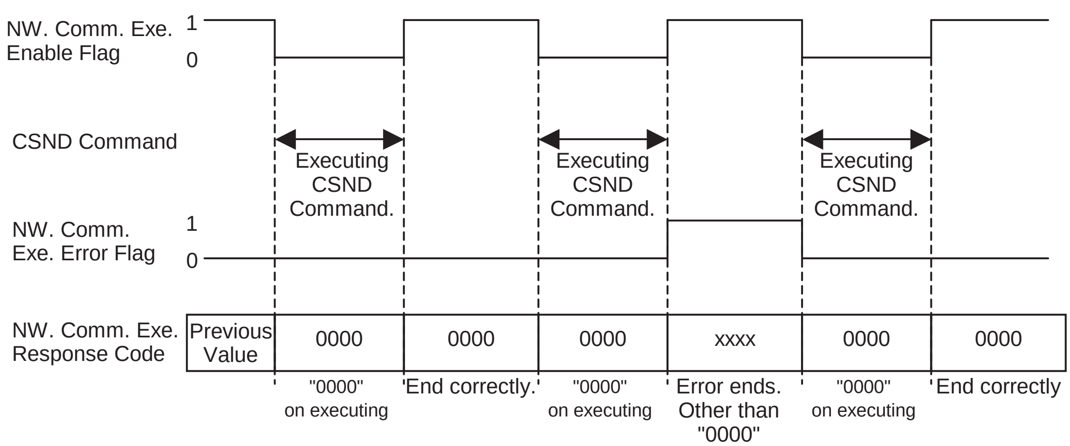
Flag and Status

The related flags are shown below.

Name	Address	Description
Error Flag	ER	Goes to ON when address range of S, D, and C areas is too large.Goes to ON when Network Instruction Execution Enable Flag is OFF for the communication port specified by C. Otherwise, goes to OFF.
Network Communication Instruction Execution Enable Flag	A20200 to A20207	Goes to 1 (ON) when network communication (CSND or PMCR instruction) is executable.Each bit indicates a communication port. Goes to 0 (OFF) during execution of network communication, and goes to 1 (ON) when execution ends in either case of correct or error. 

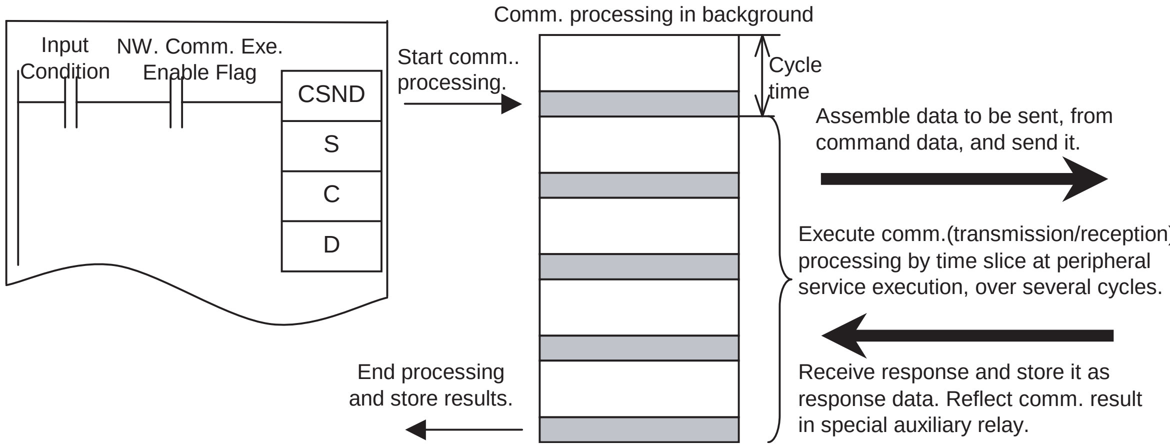
<table>
<tr><th>Name</th><th>Address</th><th>Description</th></tr>
<tr><td>Network Communication Response Code</td><td>A20300 to A21000</td><td>0 during execution of CIP instruction. When processing ends, the value is stored.
General Error Code and Additional Error Code are stored in 1 byte of High side and 1 byte of Low side, respectively.
When such an error occurs that can be detected in CPU Unit, such as time out or incorrect format, its response is stored only in this area. Note that it is not stored in the response area.

</td></tr>
<tr><td>Network Communication Execution Error Flag</td><td>A21900 to A21907</td><td>Goes to 1 (ON) when an error occurs during execution of network communication.
Each bit corresponds to each communication port. This status is held until next execution of network communication. Note that this bit goes to ON when an error occurs during communication (no data in response area) or when receiving an error response from Target (some data in response area).

</td></tr>
</table>

The figure below shows the relation between execution of CSND Instruction and each flag.



Execution Timing

For instructions for network communication, when input conditions are satisfied, the communication processing gets just started and it is in “Communication Port Service” of the peripheral services, in the background, that the actual processing is executed.



1. When the input conditions are satisfied, if Network Communication Instruction Execution Enable Flag (A20200 to A20207) is 1 (ON) at this moment, each instruction sets 0 (OFF) in Network Communication Instruction Execution Enable Flag (A20200 to A20207), 0 (OFF) in Network Communication Execution Error Flag (A21900 to A21907), and 0000 hex in Network Communication Response Code (A203 to A210), reads C, and starts the communication processing (CIP Instruction Issue/Response Reception).
2. In the peripheral service processing, data to be sent is created based on the operand (See note 1.), and CIP Instruction to the communication units are issued.
3. If the issue processing is not completed in one peripheral service, that processing will be executed, by time slice, in the next communication port service.
4. When a response is returned, the response data specified by the operand is updated in the peripheral service (See note 2.). At this moment, Network Communication Instruction Execution Enable Flag (A20200 to A20207) of the special auxiliary relay goes to 1 (ON), and Network Communication Execution Error Flag (A21900 to A21907) and Network Communication Response Code (A203 to A210) are updated.

Note (1) In case of CSND Instruction, it reads S and creates an optional CIP Instruction.
(2) In case of CSND Instruction, D is updated with CIP Response.

Read Service by Variables

Predefined Data Type

Shown below is the case that the data type of a variable is INT/WORD/UINT/UDINT/DWORD/BOOL/REAL.

Service Code=4C hex

Request Data

Word	
S+1	004C
S+2	0100

Response Data

Word		
D+4	Type	00
D+5	Data	
	:	

Type: Data Type Code of a variable which was read. Refer to the following “Data Type Code.”

Structure

The situation is a little different for structured variables.

In case of a structured variable, it has to be confirmed whether or not the specified structured variable is defined correctly. CRC Code (Cyclic Redundant Code) calculated from the structure definition is used for confirming that it is identified with the structure definition.

Although the format of Request Data is the same, that of Response Data is different.

“Type” will be A0 hex. CRC Code is stored in the channel next to Type Field.

Service Code=4C hex

Request Data

Word	
S+1	004C
S+2	0100

Response Data

Word		
D+4	A0	02
D+5	CRC	
D+6	Data	
	:	

Array Variable

In case of an array variable, access to the whole of an array variable is the same as in the case of Predefined Data Type.

Access to an array element is access to “a member of Instance.”

Service Code=4C hex

Request Data

Word	
S+1	004C
S+2	0100

Response Data

Word		
D+4	Type	00
D+5	Data	
:	:	
D+n		

Type: Data Type Code of a variable which was read. Refer to the following “Data Type Code.”

Example 1

In case that an array variable with ten INT-type elements, ArrayData[10], exists and that you access to the whole of this array variable:

Service Code=4C hex

Request Data

Word	
S+1	004C
S+2	0100

Response Data

Word		
D+4	C3	00
:	Data	
	:	
	:	
D+15	Data	

Control Data

Word			
C+5	91	09	
C+6	41	72	“Ar”
C+7	72	61	“ra”
C+8	79	44	“yD”
C+9	61	74	“at”
C+10	61	00	“a”

Example 2

In case that an array variable with ten INT-type elements, ArrayData[10], exists and that you access to the 3rd element of this array variable, ArryData:[2]:

Service Code=4C hex

Request Data

Word		
S+1	004C	
S+2	0100	

Response Data

Word		
D+4	C3	00
D+5	Data	

Control Data

Word			
C+5	91	0B	
C+6	41	72	“Ar”
C+7	72	61	“ra”
C+8	79	44	“yD”
C+9	61	74	“at”
C+10	61	28	“a”, 28 hex = Logical Segment: Member ID is specified.
C+11	02	00	02 hex = Member ID = “2.” The 3rd member is specified.

Write Service by Variables

Predefined Data Type

Shown below is the case that the data type of a variable is INT/WORD/UINT/UDINT/DWORD/BOOL/REAL.

Service Code=4D hex

Request Data

Word		
S+1	004D	
S+2	Type	00
S+3	0100	
S+4	Data to be written	
S+5	:	

Response Data

Word	
D+4	None

Type: Data Type Code of a variable which was read. Refer to the following “Data Type Code.”

Data to be written: Set the data to be written in order from the lower byte to the upper byte.

Structure

The situation is a little different for structured variables.

In case of a structured variable, it has to be confirmed whether or not the specified structured variable is defined correctly. CRC Code (Cyclic Redundant Code) calculated from the structure definition is used for confirming that it is identified with the structure definition.

Although the format of Request Data is the same, that of Response Data is different.

“Type” will be A0 hex. CRC Code is stored in the channel next to Type Field.

Service Code=4D hex

Request Data

Word		
S+1	004D	
S+2	A0	02
S+3	CRC	
S+4	0100	
S+5	Data to be written	
	:	

Response Data

Word	
D+4	None

Array Variable

In case of an array variable, access to the whole of an array variable is the same as in the case of Predefined Data Type.

Access to an array element is access to “a member of Instance.”

Service Code=4D hex

Request Data

Word		
S+1	004D	
S+2	Type	00
S+3	0100	
S+4	Data to be written	
:	:	
S+n	Data to be written:	

Response Data

Word	
D+4	None

Type: Data Type Code of a variable which was read. Refer to the following “Data Type Code.”

Data to be written: Set the data to be written in order from the lower byte to the upper byte.

Example 1

In case that an array variable with ten INT-type elements, ArrayData[10], exists and that you access to the whole of this array variable:

Service Code=4D hex

Request Data

Word		
S+1	004D	
S+2	C3	00
S+3	0100	
S+4	Data to be written	
:	:	
S+13	Data to be written:	

Response Data

Word	
D+4	None

Control Data

Word			
C+5	91	09	
C+6	41	72	“Ar”
C+7	72	61	“ra”
C+8	79	44	“yD”
C+9	61	74	“at”
C+10	61	00	“a”

Example 2

In case that an array variable with ten INT-type elements, ArrayData[10], exists and that you access to the 3rd element of this array variable, ArryData:[2]:

Service Code=4D hex

Request Data

Word		
S+1	004D	
S+2	C3	00
S+3	0100	
S+4	Data to be written	

Response Data

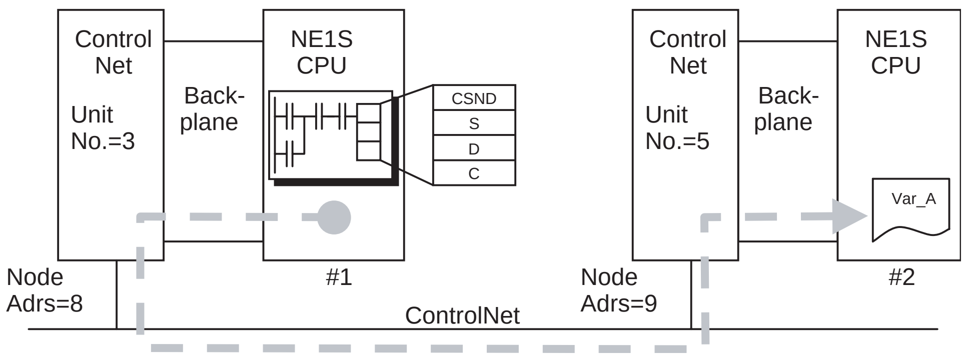
Word	
D+4	None

Control Data

Word			
C+5	91	09	
C+6	41	72	"Ar"
C+7	72	61	"ra"
C+8	79	44	"yD"
C+9	61	74	"at"
C+10	61	00	"a"
C+11	28	02	28 hex = Logical Segment: Member ID is specified. 02 hex = Member ID = "2". The 3rd member is specified.

Use Example of CSND Instruction

Read WORD-Type Variable via ControlNet



Read a variable of the right CPU Unit #2, "Var_A," from the left CPU Unit #1, and store it in a variable of #1, "Var_B."

The route is as follows:

Backplane Port of CPU#1 → ControlNet Unit (Unit No. = 3) → ControlNet Port of ControlNet Unit → ControlNet Unit (Node Address = 9) → Backplane Port of ControlNet Unit → CPU#2

Therefore, the path is described as follows:

[01] [13] [02] [09] [01] [00]

A variable to be accessed to is "Var_A" and it is described as follows:

[91] [05] [56] [61] [72] [5F] [41] [00]

Shown below is the parameter setting for CSND Instruction.

Service Code and Service Data are described in S data. Link Path and IOI (variable name) are described in C data. The received response is stored in D data.

S Data

S	0004	Request Data Size (unit: byte)
S+1	004C	Service Code
S+2	0100	Service Data

C Data

C	000A	Reception Buffer Size (unit: word) (when area of 10 words is specified)
C+1	0300	Set 3 in Communication Port No. of CSND Instruction.
C+2	0000	Reserved (0000 fixed)
C+3	0000	Service Execution Monitoring Timer (usually “0000”)
C+4	8004	Set “1” in the top bit. Describe the size of Variable Name in the unit of word.
C+5	9105	Describe Variable Name according to the method above.
C+6	5661	“Va”
C+7	725F	“r_”
C+8	4100	“A” (Variable Name ends here.)
C+9	0003	Describe Link Path Size in the unit of word.
C+10	0113	Describe Link Path according to the method above.
C+11	0209	
C+12	0100	Link Path ends here.

When S data and C data are set up and CSND Instruction is executed, received data is stored in D data. The execution result is stored in General Status area. For the details, refer to the following “General Status Code.” In some cases, not only General Status Code but also Additional Status Code may be added.

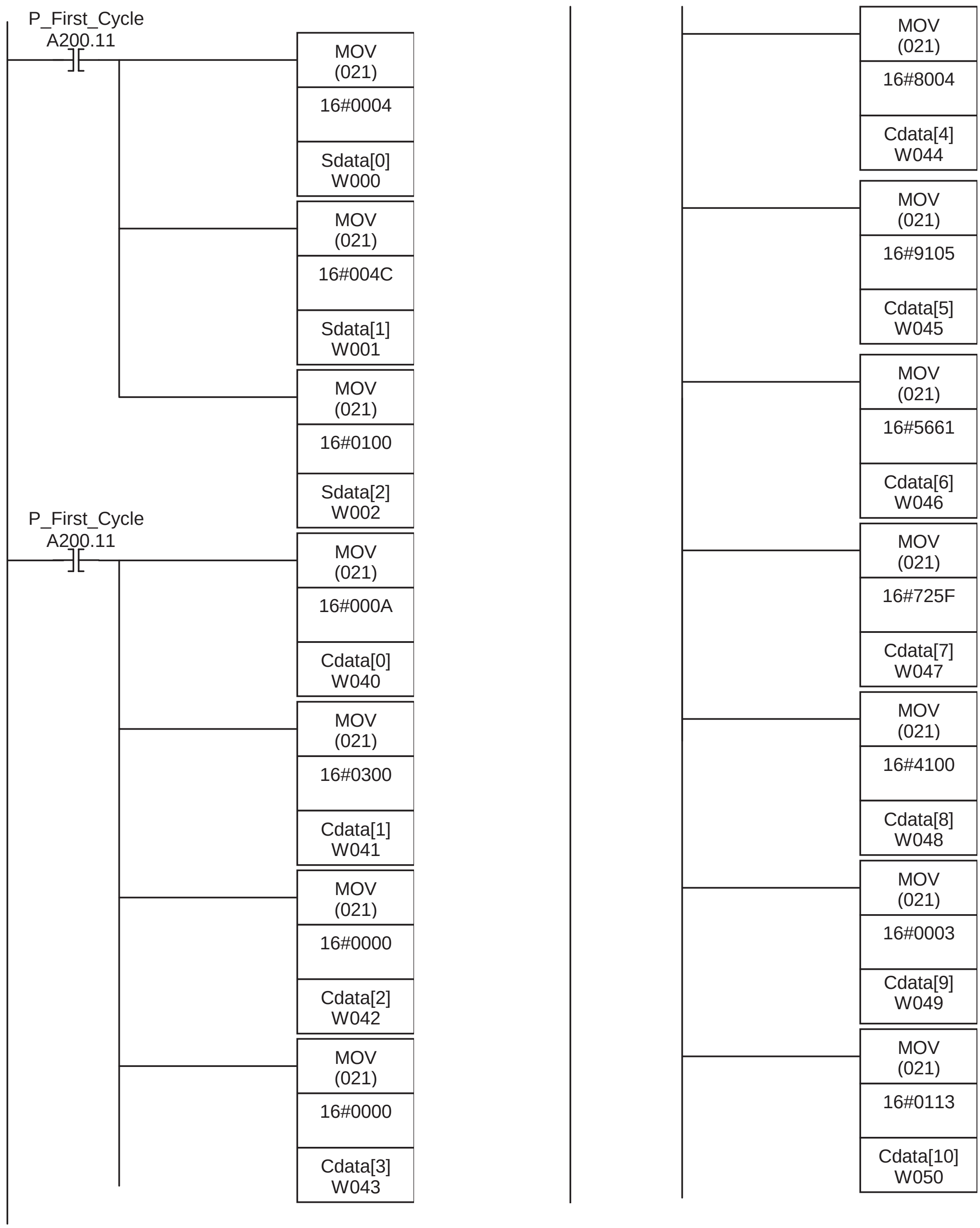
The data which was read is stored in Data of D+4. It is stored in the way of Little Endian (lining in order from the lower byte to the upper byte).

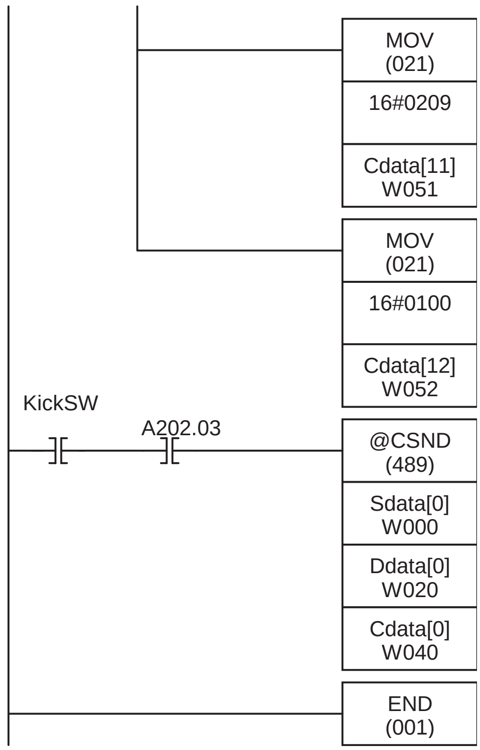
D Data

D	0008	Response Data Size (unit: byte)
D+1	00CC	Service Code (The 8th bit goes to On. 4C=>CC)
D+2	0000	General Status (0000 hex indicates correct end.)
D+3	0000	Additional Status (The upper 2 bytes will be stored if any additional information.)
D+4	D200	Data Type Code (WORD Type = D2)
D+5	Data	Data that was read.

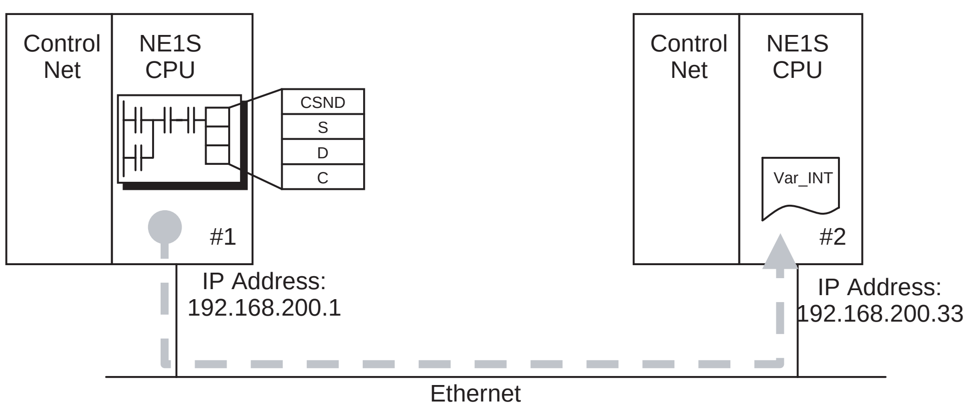
Example of Ladder Program

Name	Data Type	No. of Elements	Address	Comment
Sdata	WORD	16	W000	Operand S of CSND Instruction (data to be sent)
Ddata	WORD	16	W020	Operand S of CSND Instruction (data received)
Cdata	WORD	20	W040	Operand C of CSND Instruction (control)
KickSW	BOOL			Switch to start CSND Instruction
			A202.03	Communication Instruction Execution Enable Flag


P_First_Cycle
A200.11
MOV
(021)
16#0004
Sdata[0]
W000
MOV
(021)
16#004C
Sdata[1]
W001
MOV
(021)
16#0100
Sdata[2]
W002
P_First_Cycle
A200.11
MOV
(021)
16#000A
Cdata[0]
W040
MOV
(021)
16#0300
Cdata[1]
W041
MOV
(021)
16#0000
Cdata[2]
W042
MOV
(021)
16#0000
Cdata[3]
W043
MOV
(021)
16#8004
Cdata[4]
W044
MOV
(021)
16#9105
Cdata[5]
W045
MOV
(021)
16#5661
Cdata[6]
W046
MOV
(021)
16#725F
Cdata[7]
W047
MOV
(021)
16#4100
Cdata[8]
W048
MOV
(021)
16#0003
Cdata[9]
W049
MOV
(021)
16#0113
Cdata[10]
W050



Write INT-Type Variable via Ethernet



Write data into a variable of the right CPU Unit #2, “Var_INT,” from the left CPU Unit #1.

The route is as follows:

Ethernet Port of CPU#1 → CPU#2 (IP Address = 192.168.200.33)

The path is described as follows:

[12] [0E] [31] [39] [32] [2E] [31] [36] [38] [2E] [32] [30] [30] [2E] [33] [33]

A variable to be accessed, “Var_INT,” is described as follows:

[91] [07] [56] [61] [72] [5F] [49] [4E] [54] [00]

Shown below is the parameter setting for CSND Instruction.

Service Code and Service Data are described in S data. Link Path and IOI (variable name) are described in C data. The received response is stored in D data.

The data to be written is stored in the way of Little Endian (lining in order from the lower byte to the upper byte).

S Data

S	0008	Request Data Size (unit: byte)
S+1	004D	Service Code
S+2	C300	Data Type Code = C3
S+3	0100	No. of Elements = 1
S+4	Data	Data to be written

C Data

C	000A	Reception Buffer Size (unit: word) (when area of 10 words is specified.)
C+1	0000	When Communication Port No. of CSND Instruction is set to 0.
C+2	0000	Reserved (0000, fixed)
C+3	0000	Service Execution Monitoring Timer (usually, “0000”)
C+4	8005	Set the top bit to “1.” Describe the size of Variable Name in the unit of word.
C+5	9107	Describe Variable Name according to the method above.
C+6	5661	“Va”
C+7	725F	“r_”
C+8	494E	“IN”
C+9	5400	“T” (Variable Name ends here.)
C+10	0008	Describe Link Path Size in the unit of word.
C+11	120E	Describe Link Path according to the method above.
C+12	3139	“19”
C+13	322E	“2.”
C+14	3136	“16”
C+15	382E	“8.”
C+16	3230	“20”
C+17	302E	“0”
C+18	3333	“33” (Link Path ends here.)

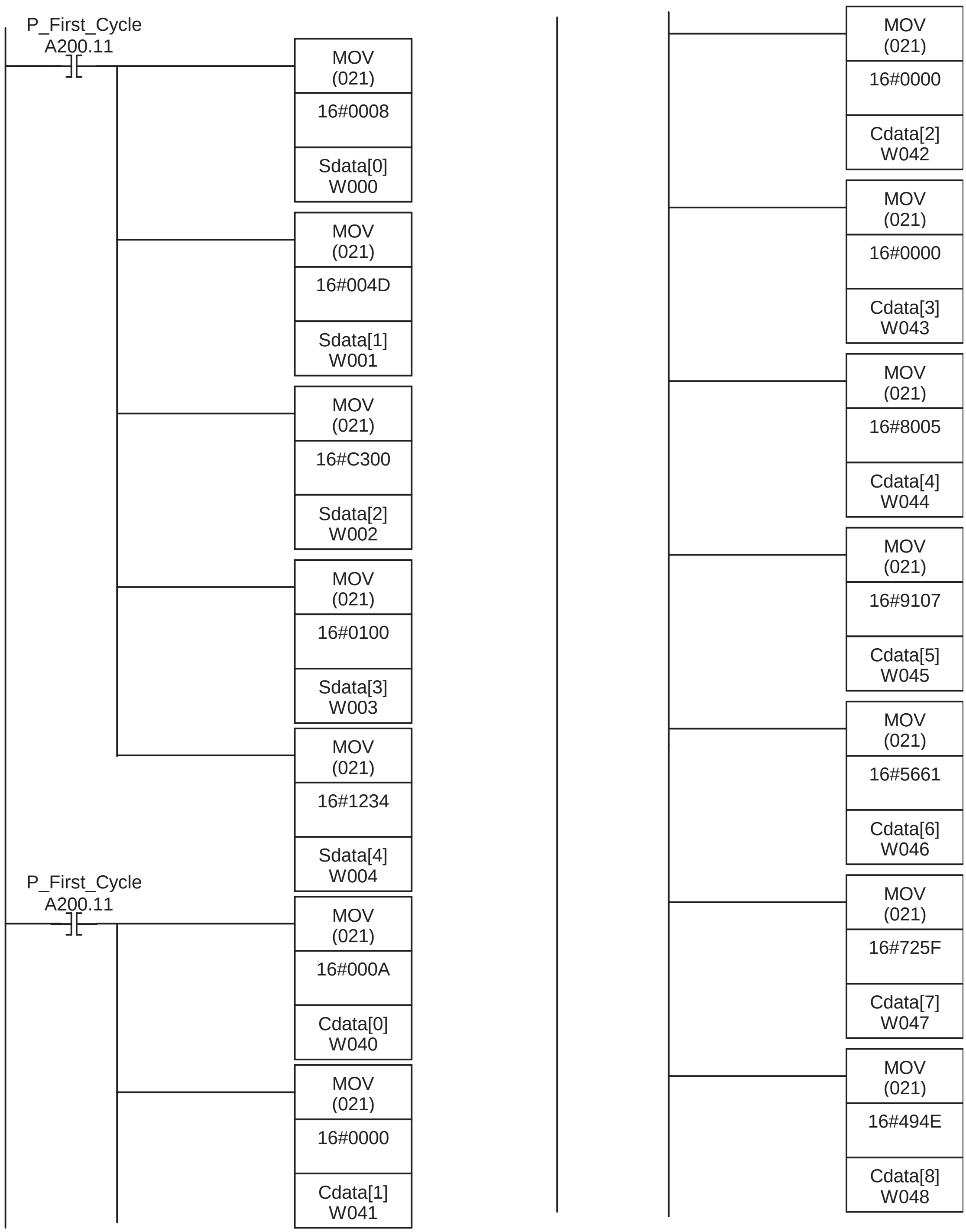
When S data and C data are set up and CSND Instruction is executed, received data is stored in D data. The execution result is stored in General Status area. For the details, refer to the following “General Status Code.” In some cases, not only General Status Code but also Additional Status Code may be added. If no Additional Code exists, “0000” is stored.

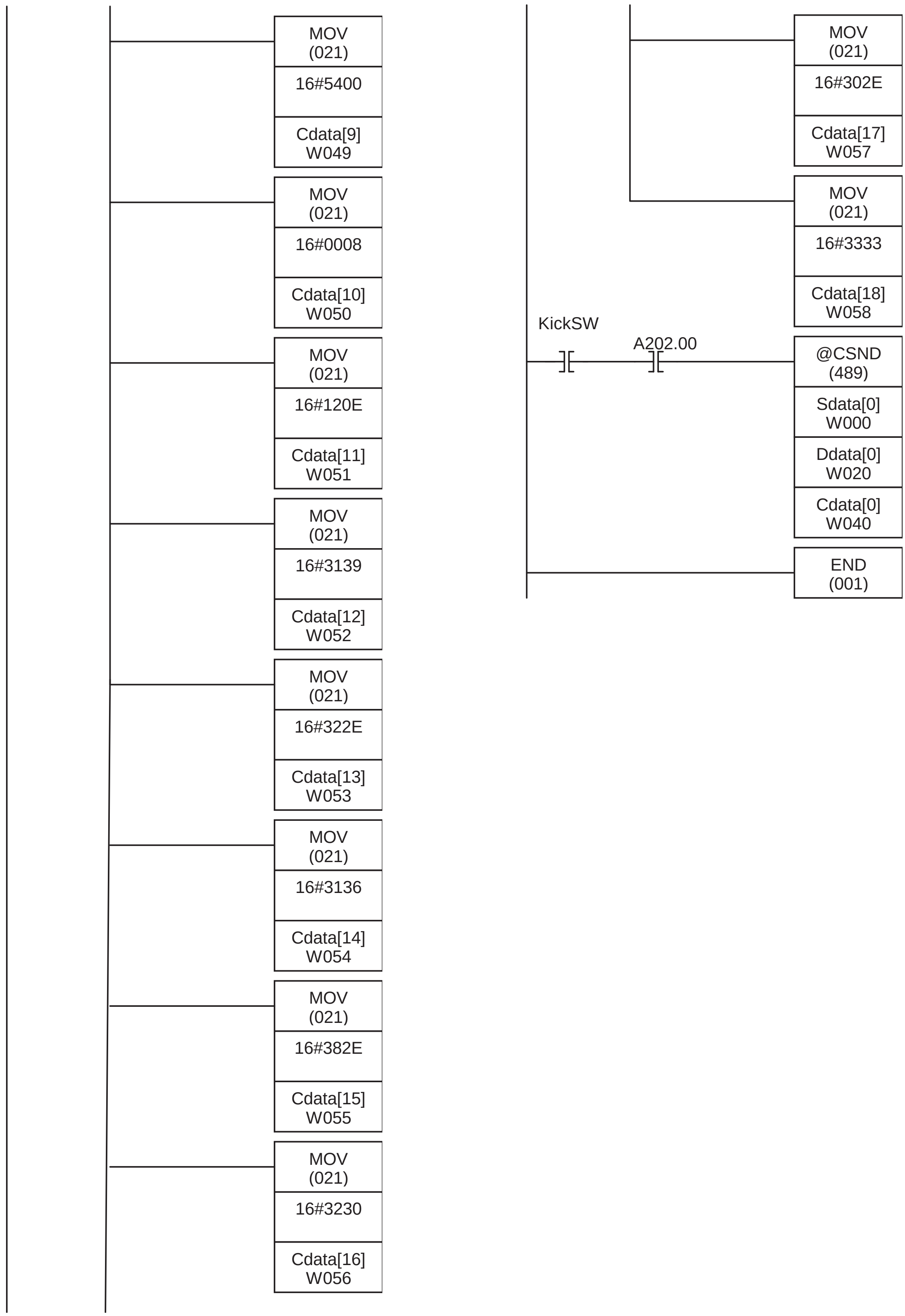
D Data

D	0006	Response Data Size (unit: byte)
D+1	00CD	Service Code (The 8th bit goes to On. 4D => CD)
D+2	0000	General Status (0000 hex indicates correct end.)
D+3	0000	Additional Status (The upper 2 bytes will be stored if any additional information.)

Example of Ladder Program

Name	Data Type	No. of Elements	Address	Comment
Sdata	WORD	16	W000	Operand S of CSND Instruction (data to be sent)
Ddata	WORD	16	W020	Operand S of CSND Instruction (data received)
Cdata	WORD	20	W040	Operand C of CSND Instruction (control)
KickSW	BOOL			Switch to start CSND Instruction
			A202.03	Communication Instruction Execution Enable Flag


P_First_Cycle
A200.11
MOV (021)
16#0008
Sdata[0] W000
MOV (021)
16#004D
Sdata[1] W001
MOV (021)
16#C300
Sdata[2] W002
MOV (021)
16#0100
Sdata[3] W003
MOV (021)
16#1234
Sdata[4] W004
P_First_Cycle
A200.11
MOV (021)
16#000A
Cdata[0] W040
MOV (021)
16#0000
Cdata[1] W041
MOV (021)
16#0000
Cdata[2] W042
MOV (021)
16#0000
Cdata[3] W043
MOV (021)
16#8005
Cdata[4] W044
MOV (021)
16#9107
Cdata[5] W045
MOV (021)
16#5661
Cdata[6] W046
MOV (021)
16#725F
Cdata[7] W047
MOV (021)
16#494E
Cdata[8] W048


MOV
(021)
16#5400
Cdata[9]
W049
MOV
(021)
16#0008
Cdata[10]
W050
MOV
(021)
16#120E
Cdata[11]
W051
MOV
(021)
16#3139
Cdata[12]
W052
MOV
(021)
16#322E
Cdata[13]
W053
MOV
(021)
16#3136
Cdata[14]
W054
MOV
(021)
16#382E
Cdata[15]
W055
MOV
(021)
16#3230
Cdata[16]
W056
MOV
(021)
16#302E
Cdata[17]
W057
MOV
(021)
16#3333
Cdata[18]
W058
KickSW
A202.00
@CSND
(489)
Sdata[0]
W000
Ddata[0]
W020
Cdata[0]
W040
END
(001)

Data Type

Data Type Code

No.	Data Type Name	Code [hex]	Description
1.	TIMER	01	OMRON Specific Data Type for Timer Instruction in which UP flag and Present Counter are involved.
2.	COUNTER	02	OMRON Specific Data Type for Counter Instruction in which UP flag and Present Counter are involved.
3.	BOOL	C1	Logical Boolean with values TRUE and FALSE TRUE = 01 hex, FALSE = 00 hex
4.	INT	C3	Signed 16-bit integer value
5.	DINT	C4	Signed 32-bit integer value
6.	UINT	C7	Unsigned 16-bit integer value
7.	UDINT	C8	Unsigned 32-bit integer value
8.	REAL	CA	32-bit floating point value
9.	WORD	D2	bit string - 16-bits
10.	DWORD	C3	bit string - 32-bits
11.	STRUCT	A0	Structured variable
12.	STRING	D0	String of letters

Data Placement

The data placement for each data type is described below. The data placement differs between the CPU Unit memory and the data sent and received using the CSND instruction. Be sure to reorder the data when sending or receiving data with the CSND instruction.

BOOL Data

CPU Unit Data Placement

BOOL data in the CPU Unit is at the specified bit location for fixed allocations using address specifications. For automatic allocations, the bit location is automatically allocated. Information on forced set/reset status cannot be read or written from the program.

CSND Instruction Data Placement

Bit

Wd	15 to 8	7 to 0
+0	Data	Information on Forced Set/Reset Status

Data: True = 01 hex, False = 00 hex

Information on Forced Set/Reset Status: Forced = 01 hex, Not Forced = 00 hex

Only data can be written, i.e., forced status information cannot be written.

BOOL Data (Whole Array)

CPU Unit Data Placement

Bit

Wd	15							8	7							0
+0	D15							D8	D7							D8
+1															D17	D16
:	:															

D0 to Dn: True = 01 hex, False = 00 hex

Allocations always start a bit 0 for arrays. It is not possible to start allocations from any other bit (e.g., starting from bit 4 is not possible).

CSND Instruction Data Placement

Bit

Wd	15							8	7							0
+0	D7							D0	D15							D8
+1							D17	D16								
:								:								

D0 to Dn: True = 01 hex, False = 00 hex

WORD, INT, or UINT Data

CPU Unit Data Placement

Bit

Wd	15 8	7 0
+0	Data (H)	Data (L)

CSND Instruction Data Placement

Bit

Wd	15 8	7 0
+0	Data (L)	Data (H)

DWORD, DINT, or UDINT Data

CPU Unit Data Placement

Bit

Wd	15 8	7 0
+0	Data (LH)	Data (LL)
+1	Data (HH)	Data (HL)

CSND Instruction Data Placement

Bit

Wd	15 8	7 0
+0	Data (LL)	Data (LH)
+1	Data (HL)	Data (HH)

TIMER Data

CPU Unit Data Placement

The data size and meaning of TIMER variables depend on the instruction that is used. When using instructions that require a bit operand (e.g., LD, AND, OR, or, OUT), the Completion Flag is accessed. When using instructions that require other operands, the PV is accessed.

Bit

Wd	15 8	7 0
+0	Current Value (H)	Current Value (L)

CSND Instruction Data Placement

Bit

Wd	15 8	7 0
+0	Current Value (L)	Current Value (H)
+1	Up Flag	Information of Forced Reset/Reset

Current Value: Current value of the timer

Up Flag: Time up = 01 hex, Others = 00 hex

Information of Forced Reset/Reset: Forced = 01 hex, Not Forced = 00 hex

Only current values can be written.

COUNTER Data

CPU Unit Data Placement

The data size and meaning of COUNTER variables depend on the instruction that is used. When using instructions that require a bit operand (e.g., LD, AND, OR, or, OUT), the Completion Flag is accessed. When using instructions that require other operands, the PV is accessed.

Wd	Bit 15 – 8	Bit 7 – 0
+0	Current Value (H)	Current Value (L)

CSND Instruction Data Placement

Wd	Bit 15 – 8	Bit 7 – 0
+0	Current Value (L)	Current Value (H)
+1	Up Flag	Information of Forced Reset/Reset

Current Value: Current value of the timer

Up Flag: Count up = 01 hex, Others = 00 hex

Information of Forced Reset/Reset: Forced = 01 hex, Not Forced = 00 hex

Only current values can be written.

STRING Data

CPU Unit Data Placement

Wd	Bit 15 – 8	Bit 7 – 0
+0	Data 1	Data 2
+1	:	:
:	Data m	Data n
+N	0 hex	0 hex

All data up to 00 hex is treated as string data. If there is an odd number of characters in the string, 00 hex is stored in the lower byte of the word.

CSND Instruction Data Placement

Wd	Bit 15 – 8	Bit 7 – 0
+0	Size	00
+1	Data 1	Data 2
:	:	:
+N	Data m	Data n

Size: Data Size (unit = byte)

Data: Data of letters. When Data Size is an odd number, pad the lower side of the last word with 00 hex.

REAL Data

READ data conforms to the definition of single-precision floating-point data in IEEE 754. Single-precision data uses 32 bits in the following format.

Actual value = (–1) s2^{e-127} (1.f)

s	e		f	
31	30	23	22	0

CPU Unit Data Placement

Wd	Bit 15	to 8	Bit 7 to 0
+0	f bits 08 to 15		f bits 00 to 07
+1	s	e	f bits 16 to 22

CSND Instruction Data Placement

Wd	Bit 15	to 8	Bit 7 to 0
+0	f bits 00 to 07		f bits 08 to 15
+1	e	f bits 16 to 22	e

Response Code

General Status Code

General Status Code is stored in the response reception area, D+2, after execution of CSND Instruction is completed.

This code is also reflected in the upper byte of A203 to A210. When Additional Code is added, only part for 1 byte is reflected in the lower byte of A203 to A230.

General Status Code (hex)	Status Name	Description of Status
00	Success	Service was successfully performed by the object specified.
01	Connection failure	A connection related service failed along the connection path.
02	Resource unavailable	Resources needed for the object to perform the requested service were unavailable.
03	Invalid parameter value	See Status Code 20 hex, which is the preferred value to use for this condition.
04	Path segment error	The path segment identifier or the segment syntax was not understood by the processing node. Path processing shall stop when a path segment error is encountered.
05	Path destination unknown	The path is referencing an object class, instance or structure element that is not known or is not contained in the processing node. Path processing shall stop when a path destination unknown error is encountered.
06	Partial transfer	Only part of the expected data was transferred.
07	Only part of the expected data was transferred.	The messaging connection was lost.
08	Service not supported	The requested service was not implemented or was not defined for this Object Class/Instance.
09	Invalid attribute value	Invalid attribute data detected.
0A	Attribute list error	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
0B	Already in requested mode/state	The object is already in the mode/state being requested by the service.
0C	Object state conflict	The object cannot perform the requested service in its current mode/state.

General Status Code (hex)	Status Name	Description of Status
0D	Object already exists	The requested instance of object to be created already exists.
0E	Attribute not settable	A request to modify a non-modifiable attribute was received.
0F	Privilege violation	A permission/privilege check failed.
10	Device state conflict	The device's current mode/state prohibits the execution of the requested service.
11	Reply data too large	The data to be transmitted in the response buffer is larger than the allocated response buffer
12	Fragmentation of a primitive value	The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type.
13	Not enough data	The service did not supply enough data to perform the specified operation.
14	Attribute not supported	The attribute specified in the request is not supported.
15	Too much data	The service supplied more data than was expected.
16	Object does not exist	The object specified does not exist in the device.
17	Service fragmentation sequence not in progress	The fragmentation sequence for this service is not currently active for this data.
18	No stored attribute data	The attribute data of this object was not saved prior to the requested service.
19	Store operation failure	The attribute data of this object was not saved due to a failure during the attempt.
1A	Routing failure (request packet too large)	The service request packet was too large for transmission on a network in the path to the destination. The routing device was forced to abort the service.
1B	Routing failure (response packet too large)	The service response packet was too large for transmission on a network in the path from the destination. The routing device was forced to abort the service.
1C	Missing attribute list entry data	The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior.
1D	Invalid attribute value list	The service is returning the list of attributes supplied with status information for those attributes that were invalid.
1E	Embedded service error	An embedded service resulted in an error.
1F	Vendor specific error	A vendor specific error has been encountered. The Additional Code Field of the Error Response defines the particular error encountered. Use of this General Error Code should only be performed when none of the Error Codes presented in this table or within an Object Class definition accurately reflect the error.
20	Invalid parameter	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.
21	Write-once value or medium already written	An attempt was made to write to a write-once medium (e.g. WORM drive, PROM) that has already been written, or to modify a value that cannot be changed once established.
22	Invalid Reply Received	An invalid reply is received (e.g. reply service code does not match the request service code, or reply message is shorter than the minimum expected reply size). This status code can serve for other causes of invalid replies.
23-24		Reserved by CIP for future extensions
25	Key Failure in path	The Key Segment that was included as the first segment in the path does not match the destination module. The object specific status shall indicate which part of the key check failed.
26	Path Size Invalid	The size of the path which was sent with the Service Request is either not large enough to allow the Request to be routed to an object or too much routing data was included.

General Status Code (hex)	Status Name	Description of Status
27	Unexpected attribute in list	An attempt was made to set an attribute that is not able to be set at this time.
28	Invalid Member ID	The Member ID specified in the request does not exist in the specified Class/Instance/Attribute.
29	Member not settable	A request to modify a non-modifiable member was received.
2A	Group 2 only server general failure	This error code may only be reported by DeviceNet group 2 only servers with 4K or less code space and only in place of Service not supported, Attribute not supported and Attribute not settable.
2B-CF		Reserved by CIP for future extensions
D0-FF	Reserved for Object Class and service errors	This range of error codes is to be used to indicate Object Class specific errors. Use of this range should only be performed when none of the Error Codes presented in this table accurately reflect the error that was encountered.

Example of Additional Status in case that General Status Is 01 Hex. (Status of Connection Manager Object)

General Status (hex)	Additional Status (hex)	Explanation
01	0100	Connection in Use or Duplicate Forward Open.
01	0103	Transport Class and Trigger combination not supported
01	0106	Ownership Conflict
01	0107	Connection not found at target application.
01	0108	Invalid Connection Type. Indicates a problem with either the Connection Type or Priority of the Connection.
01	0109	Invalid Connection Size
01	0110	Device not configured
01	0111	RPI not supported. May also indicate problem with connection time-out multiplier, or production inhibit time.
01	0113	Connection Manager cannot support any more connections
01	0114	Either the Vendor Id or the Product Code in the key segment did not match the device
01	0115	Product Type in the key segment did not match the device
01	0116	Major or Minor Revision information in the key segment did not match the device
01	0117	Invalid Connection Point
01	0118	Invalid Configuration Format
01	0119	Connection request fails since there is no controlling connection currently open.
01	011A	Target Application cannot support any more connections
01	011B	RPI is smaller than the Production Inhibit Time.
01	0203	Connection cannot be closed since the connection has timed out
01	0204	Unconnected Send timed out waiting for a response.
01	0205	Parameter Error in Unconnected Send Service
01	0206	Message too large for Unconnected message service
01	0207	Unconnected acknowledge without reply
01	0301	No buffer memory available
01	0302	Network Bandwidth not available for data
01	0303	No Tag filters available
01	0304	Not Configured to send real-time data
01	0311	Port specified in Port Segment Not Available
01	0312	Link Address specified in Port Segment Not Available
01	0315	Invalid Segment Type or Segment Value in Path
01	0316	Path and Connection not equal in close
01	0317	Either Segment not present or Encoded Value in Network Segment is invalid.

General Status (hex)	Additional Status (hex)	Explanation
01	0318	Link Address to Self Invalid
01	0319	Resources on Secondary Unavailable
01	031A	Connection already established
01	031B	Direct connection already established
01	031C	Miscellaneous
01	031D	Redundant connection mismatch
01	031F	No connection resources exist for target path
01	0320-07FF	Vendor specific

Error Code Unique to CSND Instruction

When CSND Instruction itself turns to be an error (incorrect parameter, etc.), this code is reflected in A203 to A210.

Error Code (hex)	Error Name	Cause	Remarks
2001	Inappropriate Request Data Length	The value for the Request Data Size in the first field of S is less than 4 (specified as less than 4 bytes), or that the total request data size exceeds the specified amount (512 bytes).	The message will not be sent and the Communications port Error Flag for the corresponding port will turn ON.
2002	Inappropriate Response Data Length	The value for the Response Data Size in the first field of D is less than 8 (specified as less than 8 bytes).	
2003	Inappropriate Control Data Length	The Link Address Size in the sixth field of C exceeds the specified amount (512 bytes).	
0201	Insufficient Response Area	The CIP response data length exceeds the Response Buffer Size in the first field of C.	The response will be discarded and the Communications Port Error Flag for the corresponding port will turn ON.
---	EPATH Error	CIP Segment Encoding Errors (EPATH Errors) are listed below. For details on CIP segments, refer to Appendix C: Data Management in CIP Common Specification Volume 1 Release 1.0.	The message will not be sent and the Communications port Error Flag for the corresponding port will turn ON.
2041		The value specified for the Path Type in the IOI Path area of operand C (C+4, bit 15) is not 1 or 0 (i.e., Class/Instance/ Attribute or EPATH is not specified).	
2042		The value for the IOI Size is 0 when the Path Type is specified as 1 in the IOI Path Area of operand C (C+4, bit 15).	
0401		Logical Segment/Data Segment is not specified when the Path Type is specified as 1 in the IOI Path Area of operand C (C+4, bit 15).	
2043		The Logical Type is not specified as Class/ Instance/Attribute.	
0411		The Logical Format is not specified as 8, 16, or 32bits.	
0421		The Segment Sub-Type is not specified as ANSI Extended Symbol Segment.	
2044		The IOI length is set to 0x100 or higher for the IOI Path in operand C.	
1F03	Maximum Send/Receive Length Exceeded	The send request exceeded the CIP maximum message size.	
1F02	Link Path Error	An undefined link path was specified. An undefined port such as port No. 0x05 or 0x10 was specified.	

Error Code (hex)	Error Name	Cause	Remarks
0281	Internal Error	An internal error occurred.	The message will not be sent and the Communications port Error Flag for the corresponding port will turn ON.
0282			
0283			
0284			
1F81			
1F01	Response Timeout	A timeout occurred in the processing at the target. The message was discarded during communications processing (the message frame length exceeded the maximum length, etc.).	The Communications Port Error Flag for the corresponding port will turn ON.

Index

Numerics

A

B

C

J-L

M

N

R

S

T

U

V–W

Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.

Cat. No. Z901-E1-05

Revision code

The following table outlines the changes made to the manual during each revision. Page numbers refer to the previous version.

Revision code	Date	Revised content
1	June 2004	Original production
2	September 2004	Revisions were made throughout the manual to provide additional explanations and information on the functionality accompanying the upgrade to Rev. 2.1. Corrections to the previous manual were also made.
03	January 2005	Corrections were made.
04	December 2005	Changes made accompanying upgrade to revision 3.1. (Refer to *Functionality Improved in Rev. 3.1* on page vi for an overview.)
05	December 2006	Additional explanations were added. Corrections were made.

OMRON Corporation

Integration Strategy &
Business Development Center
2-2-1 Nishikusatsu
Kusatsu-city, Shiga 525-0035
Japan

OMRON ELECTRONICS LLC
1 East Commerce Drive, Schaumburg, IL 60173
U.S.A.
Tel: (1)847-843-7900/Fax: (1)847-843-8568

Authorized Distributor:

Cat. No. Z901-E1-05 Note: Specifications subject to change without notice Printed in Japan

This manual is printed on 100% recycled paper.

Cat. No. Z901-E1-05 NE1S Series NE1S-CPU01 Programmable Controller OPERATION MANUAL

OMRON