B-1730(11)

FC6A SERIES

MICROSmart

Communication Manual

Automation Organizer

WindLDR

SAFETY PRECAUTIONS

- Read the "FC6A Series MICROSmart Communication Manual" to ensure correct operation before starting installation, wiring, operation, maintenance, and inspection of the FC6A Series MICROSmart.
- All FC6A Series MICROSmart modules are manufactured under IDEC's rigorous quality control system, but users must add a backup or failsafe provision to the control system when using the FC6A Series MICROSmart in applications where heavy damage or personal injury may be caused, in case the FC6A Series MICROSmart should fail.
- Care should be taken such that unauthorized access to the FC6A Series MICROSmart does not come from outside network connections. Please note that the Company shall not be liable for any loss, damage or other expenses incurred directly or indirectly by unauthorized access, etc.
- In this manual, safety precautions are categorized in order of importance:

⚠ **Warning**	**Warning notices are used to emphasize that improper operation may cause severe personal injury or death.**

- The FC6A Series MICROSmart is not designed for use in applications requiring a high degree of reliability and safety. The FC6A Series MICROSmart should not be used for such applications.
- When using the FC6A Series MICROSmart in applications (not described above) that require a high degree of reliability in terms of functionality and precision, appropriate measures such as failsafe mechanisms and redundant mechanisms must be taken for the system containing the FC6A Series MICROSmart. The following are specific examples.
  - Emergency stop and interlocking circuits must be configured outside the FC6A Series MICROSmart.
  - If relays or transistors in the FC6A Series MICROSmart output circuits should fail, outputs may remain at on or off state. For output signals which may cause serious accidents, configure monitor circuits outside the FC6A Series MICROSmart.
  - The FC6A Series MICROSmart self-diagnostic function may detect internal circuit or program errors, stop programs, and turn outputs off. Configure circuits so that the system containing the FC6A Series MICROSmart is not jeopardized when outputs turn off.
- Turn off power to the FC6A Series MICROSmart before installation, removal, wiring, maintenance, and inspection of the FC6A Series MICROSmart. Failure to turn power off may cause damage, electrical shocks or fire hazard.
- Special expertise is required to install, wire, program, and operate the FC6A Series MICROSmart. People without such expertise must not use the FC6A Series MICROSmart.
- Install the FC6A Series MICROSmart according to the instructions described in the "FC6A Series MICROSmart User's Manual". Improper installation will result in falling, failure, or malfunction of the FC6A Series MICROSmart.

⚠ **Caution**	**Caution notices are used where inattention might cause personal injury or damage to equipment.**

- The FC6A Series MICROSmart is designed for installation in a cabinet. Do not install the FC6A Series MICROSmart outside a cabinet.
- Install the FC6A Series MICROSmart in environments described in the "FC6A Series MICROSmart User's Manual". If the FC6A Series MICROSmart is used in places where the FC6A Series MICROSmart is subjected to high-temperature, high-humidity, condensation, corrosive gases, excessive vibrations, or excessive shocks, then electrical shocks, fire hazard, or malfunction will result.
- The environment for using the FC6A Series MICROSmart is "Pollution degree 2." Use the FC6A Series MICROSmart in environments of pollution degree 2 (according to IEC 60664-1).
- Prevent the FC6A Series MICROSmart from falling while moving or transporting the FC6A Series MICROSmart, otherwise damage or malfunction of the FC6A Series MICROSmart will result.
- Wiring must use lead sizes that are appropriate for the applied voltage and current. Terminal screws must be tightened with the prescribed tightening torque.
- Prevent metal fragments and pieces of wire from dropping inside the FC6A Series MICROSmart housing. Put a cover on the FC6A Series MICROSmart modules during installation and wiring. Ingress of such fragments and chips may cause fire hazard, damage, or malfunction.
- Use a power supply of the rated value. Use of a wrong power supply may cause fire hazard.
- Use an IEC 60127-approved fuse on the power line outside the FC6A Series MICROSmart. This is required when equipment containing the FC6A Series MICROSmart is destined for Europe.
- Use an IEC 60127-approved fuse on the output circuit. This is required when equipment containing the FC6A Series MICROSmart is destined for Europe.
- Use an EU-approved circuit breaker. This is required when equipment containing the FC6A Series MICROSmart is destined for Europe.
- Make sure of safety before starting and stopping the FC6A Series MICROSmart or when operating the FC6A Series MICROSmart to force outputs on or off. Incorrect operation of the FC6A Series MICROSmart may cause machine damage or accidents.
- Do not connect the ground wire directly to the FC6A Series MICROSmart. Connect a protective ground to the cabinet containing the FC6A Series MICROSmart using an M4 or larger screw. This is required when equipment containing the FC6A Series MICROSmart is destined for Europe.
- Do not disassemble, repair, or modify the FC6A Series MICROSmart modules.
- The FC6A Series MICROSmart contains electronic parts and batteries. When disposing of the FC6A Series MICROSmart, do so in accordance with national and local regulations.

About This Manual

Thank you for purchasing the FC6A Series MICROSmart manufactured by IDEC Corporation.
This document describes the FC6A Series MICROSmart system configuration, specifications, and installation methods, and it provides descriptions of the various functions.
Read this manual to ensure the correct understanding of the entire functions of the FC6A Series MICROSmart.
IDEC Corporation makes the latest product manual PDFs available on our website at no additional cost.
Please download the latest product manual PDFs from our website.
Product manual PDF download page (www.idec.com/FC6Amanuals)

This manual describes the powerful communications tools of the FC6A Series MICROSmart.

Chapter 1: General Information
General information about the FC6A Series MICROSmart with communication interfaces.

Chapter 2: Devices
Descriptions of the allocations of devices such as inputs, outputs, internal relays, registers, timers, and counters that are used in the basic and advanced instructions, as well as details about the allocations of special internal relays and special data registers for communication functions.

Chapter 3: Communication Settings
Functions for the FC6A Series MICROSmart communication, how to configure them, and examples of their use.

Chapter 4 through Chapter 8:
Various communication functions such as maintenance communication, user communication, Modbus communication, data link communication and J1939 communication.

Chapter 9: Bluetooth Communication
Description of the Bluetooth communication of the FC6A Series MICROSmart and how you can make the FC6A Series MICROSmart communicate with other devices with the Bluetooth communication.

Chapter 10: FTP Server/Client
Descriptions of the FTP server that enables you to get the files that are saved in the SD memory card inserted in the FC6A Series MICROSmart with an FTP client tool and the FTP client that enables the FC6A Series MICROSmart to transfer the files between the SD memory card and FTP servers.

Chapter 11: PING Instruction
Descriptions of the PING instruction that sends a ping packet to the specified remote host to check if communication is possible at the Internet Protocol (IP) layer.

Chapter 12: Send E-mail Function
Descriptions of the EMAIL instruction that sends preregistered e-mails.

Chapter 13: Web Server
Description of the Web server functions in the FC6A Series MICROSmart.

Chapter 14: Communication Monitor
Description of the communication monitor that enables you to monitor the communication between the FC6A Series MICROSmart and the connected devices.

Chapter 15: BACnet/IP
Description of the BACnet communication (BACnet/IP) functions in the FC6A Series MICROSmart.

Index
Alphabetical listing of key words.

Publication history

December 2015	First Edition
February 2016	Second Edition
April 2016	Third Edition
March 2017	Fourth Edition
August 2017	Fifth Edition
March 2018	Sixth Edition
June 2018	Seventh Edition
November 2018	Eighth Edition
May 2019	Ninth Edition
December 2019	Tenth Edition
April 2020	Eleventh Edition
July 2020	Twelfth Edition

Regarding laws and compatible standards

This product adheres to the laws and compatible standards of all countries involved, as shown below.

European laws and standards

This product complies with the following EU directives.

- Low Voltage Directive
- RoHS Directive
- RE Directive (FC6A-PC4 only)

To comply with these directives, this product has been designed and evaluated on the basis of the following international and European standard.

- IEC/EN 61131-2: 2007
- EN50581:2012
- EN301 489-1 V2.1.1& EN301 489-17 V2.1.1 (FC6A-PC4 only)

For details on the compatible standards and EU Directives, contact the distributor from which you purchased this product or visit our web site.

North America laws and standards

This product complies with the following standards.

- UL508*1
- UL61010-1*1
- UL61010-2-201*1
- CSA C22.2 No.142*1
- CSA C22.2 No.61010-1*1
- CSA C22.2 No.61010-2-201*1
- ANSI/ISA 12.12.01
- CAN/CSA C22.2 No.213

*1 Certain FC6A Series MICROSmart models are not compatible. For details about applicable standards, please contact IDEC Corporation.

Chinese laws and standards

The FC6A-PC4 complies with the following certification.

- SRRC

Marine standards

This product has been certified by the following classification societies.
(Applications have been submitted for certain models.)

- ABS (American Bureau of Shipping)
- DNV GL (Det Norske Veritas·Germanischer Lloyd)
- LR (Lloyd's Register)
- NK (Nippon Kaiji Kyokai)

* This product has not been certified for use on the bridge or deck.

For details on applicable standards and EU directives, please contact the dealer where purchased or check the IDEC website.

IMPORTANT INFORMATION

Under no circumstances shall IDEC Corporation be held liable or responsible for indirect or consequential damages resulting from the use of or the application of IDEC PLC components, individually or in combination with other equipment.

All persons using these components must be willing to accept responsibility for choosing the correct component to suit their application and for choosing an application appropriate for the component, individually or in combination with other equipment.

All diagrams and examples in this manual are for illustrative purposes only. In no way does including these diagrams and examples in this manual constitute a guarantee as to their suitability for any specific application. To test and approve all programs, prior to installation, is the responsibility of the end user.

About the Warranty of the Products

1. **Warranty Period**
   The Products are warranted for 3 years from the date of purchase, or from the date of delivery completion.
   * Consumable/maintenance parts such as batteries and relays if the operation exceeds 100,000 times are excluded from the 3-year warranty.

2. **Extent of Warranty**
   IDEC CORPORATION is responsible for failures or defects of the Products during the above warranty period, either a replacement part will be provided or the defective parts of the Products will be repaired free of charge. If such failure or defects should occur, please offer them to the distributor, dealer or IDEC CORPORATION with the materials in which the date of purchase is specified.
   * The expenses for installation and construction at the time of repair will not be borne.

3. **Indemnification**
   IDEC CORPORATION will not be liable under this Warranty and be indemnified and held harmless from any and all demands, suits, expenses, claims, damages and liabilities in the following event that:
   1) The Products are used or operated beyond the conditions or environment range as described in catalog, specifications or instruction; or
   2) The failure or defects of the Products arise from the cause other than the Products; or
   3) The Products are improved, modified or altered by the party other than IDEC; or
   4) The failure or defects and damages of the Products arise from the usage of the Product in the way that is not intended; or
   5) The failure or defects and damages of the Products arise from the cause beyond IDEC's control including, but not limited to, fire, earthquake, flood, lightning, other natural disasters, and acts of God; or
   6) The failure or defects and damages of the Products arise from the relocation, transportation or drop after you purchase the Products; or
   7) The failure or defects and damages of the Products arise from improper installation; or
   8) Maintenance and inspection are not carried out in accordance with instruction.

   * Customers assume their own risk in programming products, Company will not be held liable for damages as a result of improper programming.

   IDEC CORPORATION DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR USE OR PURPOSE, AS WELL AS LIABILITY FOR INCIDENTAL, SPECIAL, INDIRECT, CONSEQUENTIAL OR OTHER DAMAGES RELATING TO THE PRODUCTS

4. **Extent of Service**
   The price of the Products will not include the fee for any service such as sending technicians and engineers, IDEC CORPORATION will charge you the fee for the following:
   1) Instruction for installment and visiting for test operation, including, but not limited to creating application software and operation tests; and
   2) Maintenance and inspection, arrangement and repair; and
   3) Technical assistance and technical education; and
   4) Product test and inspection based on you request.

RELATED MANUALS

The following manuals related to the FC6A Series MICROSmart are available. Refer to them in conjunction with this manual.

Type No.	Manual Name	Description
FC9Y-B1722	FC6A Series MICROSmart User's Manual	Describes product specifications, installation and wiring instructions, instructions for basic programming operations and special functions, device and instruction lists, and troubleshooting procedures for the FC6A Series MICROSmart.
FC9Y-B1726	FC6A Series MICROSmart Ladder Programming Manual	Describes basic operations for programming with ladders on the FC6A Series MICROSmart, monitoring methods, device and instruction lists, and details of each instruction.
FC9Y-B1730	FC6A Series MICROSmart Communication Manual (this manual)	Describes specifications related to FC6A Series MICROSmart communication, descriptions of functions, configuration methods, and usage examples.
FC9Y-B1734	FC6A Series MICROSmart PID Module User's Manual	Describes PID module specifications and functions.
WindLDR Help		Describes usage instructions for WindLDR, programming software for the FC6A Series MICROSmart.

Names and Abbreviations Used in this Manual

Model Names

Name Used in This Manual			Type Number, Part Code, or Official Name
FC6A Series MICROSmart			FC6A Series MICROSmart
CPU module	All-in-One CPU module		FC6A-C16R1AE, FC6A-C16R1CE, FC6A-C16K1CE, FC6A-C16P1CE, FC6A-C16R1DE, FC6A-C16K1DE, FC6A-C16P1DE, FC6A-C24R1AE, FC6A-C24R1CE, FC6A-C24K1CE, FC6A-C24P1CE, FC6A-C40R1AE, FC6A-C40R1CE, FC6A-C40K1CE, FC6A-C40P1CE, FC6A-C40R1DE, FC6A-C40K1DE, FC6A-C40P1DE
	CAN J1939 All-in-One CPU module		FC6A-C40R1AEJ, FC6A-C40R1CEJ, FC6A-C40K1CEJ, FC6A-C40P1CEJ, FC6A-C40R1DEJ, FC6A-C40K1DEJ, FC6A-C40P1DEJ
	Plus CPU module		FC6A-D16R1CEE, FC6A-D16K1CEE, FC6A-D16P1CEE, FC6A-D32K3CEE, FC6A-D32P3CEE, FC6A-D16R4CEE, FC6A-D16K4CEE, FC6A-D16P4CEE, FC6A-D32K4CEE, FC6A-D32P4CEE
	16-I/O type		FC6A-C16R1AE, FC6A-C16R1CE, FC6A-C16K1CE, FC6A-C16P1CE, FC6A-C16R1DE, FC6A-C16K1DE, FC6A-C16P1DE
	24-I/O type		FC6A-C24R1AE, FC6A-C24R1CE, FC6A-C24K1CE, FC6A-C24P1CE
	40-I/O type		FC6A-C40R1AE, FC6A-C40R1CE, FC6A-C40K1CE, FC6A-C40P1CE, FC6A-C40R1DE, FC6A-C40R1DE, FC6A-C40K1DE, FC6A-C40P1DE, FC6A-C40R1AEJ, FC6A-C40R1CEJ, FC6A-C40K1CEJ, FC6A-C40P1CEJ, FC6A-C40R1DEJ, FC6A-C40K1DEJ, FC6A-C40P1DEJ
	Plus 16-I/O type		FC6A-D16R1CEE, FC6A-D16R4CEE, FC6A-D16K1CEE, FC6A-D16K4CEE, FC6A-D16P1CEE, FC6A-D16P4CEE
	Plus 32-I/O type		FC6A-D32K3CEE, FC6A-D32P3CEE, FC6A-D32K4CEE, FC6A-D32P4CEE
	AC power type		FC6A-C16R1AE, FC6A-C24R1AE, FC6A-C40R1AE, FC6A-C40R1AEJ
	DC power type	24V DC power type	FC6A-C16R1CE, FC6A-C24R1CE, FC6A-C40R1CE, FC6A-C16K1CE, FC6A-C24K1CE, FC6A-C40K1CE, FC6A-C16P1CE, FC6A-C24P1CE, FC6A-C40P1CE, FC6A-C40R1CEJ, FC6A-C40K1CEJ, FC6A-C40P1CEJ, FC6A-D16R1CEE, FC6A-D16K1CEE, FC6A-D16P1CEE, FC6A-D32K3CEE, FC6A-D32P3CEE, FC6A-D16R4CEE, FC6A-D16K4CEE, FC6A-D16P4CEE, FC6A-D32K4CEE, FC6A-D32P4CEE
		12V DC power type	FC6A-C16R1DE, FC6A-C16K1DE, FC6A-C16P1DE, FC6A-C40R1DE, FC6A-C40K1DE, FC6A-C40P1DE, FC6A-C40R1DEJ, FC6A-C40K1DEJ, FC6A-C40P1DEJ
	Relay output type		FC6A-C16R1AE, FC6A-C16R1CE, FC6A-C16R1DE, FC6A-C24R1AE, FC6A-C24R1CE, FC6A-C40R1AE, FC6A-C40R1CE, FC6A-C40R1DE, FC6A-C40R1AEJ, FC6A-C40R1CEJ, FC6A-C40R1DEJ, FC6A-D16R1CEE, FC6A-D16R4CEE
	Transistor output type	Transistor sink output type	FC6A-C16K1CE, FC6A-C16K1DE, FC6A-C24K1CE, FC6A-C40K1CE, FC6A-C40K1DE, FC6A-C40K1CEJ, FC6A-C40K1DEJ, FC6A-D16K1CEE, FC6A-D16K4CEE, FC6A-D32K3CEE, FC6A-D32K4CEE
		Transistor protection source output type	FC6A-C16P1CE, FC6A-C16P1DE, FC6A-C24P1CE, FC6A-C40P1CE, FC6A-C40P1DE, FC6A-C40P1CEJ, FC6A-C40P1DEJ, FC6A-D16P1CEE, FC6A-D16P4CEE, FC6A-D32P3CEE, FC6A-D32P4CEE
Expansion module	I/O module	Digital I/O module	Digital input module, digital output module, digital mixed I/O module
		Analog I/O module	Analog input module, analog output module, analog mixed I/O module
	Communication module		Serial communication module
			PID module
Expansion interface module			Expander, remote master, remote slave
Cartridge	I/O cartridge	Digital I/O cartridge	Digital input cartridge, digital output cartridge
		Analog I/O cartridge	Analog input cartridge, analog output cartridge
	Communication cartridge		RS232C communication cartridge, RS485 communication cartridge, Bluetooth communication cartridge
WindLDR			WindLDR application software
USB cable			USB maintenance cable (HG9Z-XCM42), USB Mini-B extension cable (HG9Z-XCE21)

Name Used in this Manual	WindLDR Operating Procedure
Function area settings	**Configuration** tab > **Function Area Settings** group
Monitors	Select **Online** > **Monitor** > **Start Monitor**.
PLC status	Select **Online** > **PLC** > **Status**.
Communication settings	Select **Online** > **Communication** > **Set Up**.
Modbus master request table	On the **Configuration** tab, in **Function Area Settings**, click **Communication Ports**, and in the displayed **Function Area Settings** dialog box, for **Communication Mode** under **Communication Ports**, select **Modbus RTU Master** or **Modbus TCP Client**
Application button	The button displayed on the left side of the menu bar. Click to display the menu with **New**, **Save**, and **Save As**, recent projects, **WindLDR Options**, and **Exit WindLDR**.

TABLE OF CONTENTS

1: GENERAL INFORMATION

Introduction

This chapter describes an overview of the FC6A Series MICROSmart, which is equipped with a communication interface.

Description

The FC6A Series MICROSmart can perform communication using various communication interfaces.

All-in-One CPU Module/J1939 All-in-One CPU Module

Yes: Can be used as a communication port.
No: Cannot be used as a communication port.
—: Not equipped with the CPU module.

<table>
<tr><th colspan="2" rowspan="2">Communication</th><th colspan="6">Port</th><th colspan="3">Cartridge Slot</th><th>Expansion module</th></tr>
<tr><th>Serial Port 1</th><th>USB Port</th><th>Ethernet Port 1</th><th>Ethernet Port 2</th><th>HMI-Ehternet Port</th><th>CAN Port</th><th>1</th><th>2</th><th>3</th><th>Communication module</th></tr>
<tr><td colspan="2">Maintenance Communication</td><td rowspan="4">Yes*1</td><td>Yes</td><td>Yes</td><td rowspan="11">—</td><td>Yes*2</td><td rowspan="8">No</td><td rowspan="5">Yes*4</td><td rowspan="5">Yes*4</td><td rowspan="11">—</td><td rowspan="4">Yes</td></tr>
<tr><td rowspan="3">RS232C Communication/ RS485 Communication</td><td>User Communication</td><td rowspan="10">No</td><td rowspan="4">No</td><td rowspan="4">No</td></tr>
<tr><td>Modbus RTU Communication (master/slave)</td></tr>
<tr><td>Data Link Communication (master station/slave station)</td></tr>
<tr><td>Bluetooth Communication</td><td>User Communication</td><td rowspan="7">No</td><td rowspan="7">No</td></tr>
<tr><td rowspan="3">Ethernet Communication</td><td>User Communication (server/client)</td><td rowspan="3">Yes</td><td rowspan="3">Yes*2</td><td rowspan="6">No</td><td rowspan="6">No</td></tr>
<tr><td>User Communication (UDP)</td></tr>
<tr><td>Modbus TCP Communication (server/client)</td></tr>
<tr><td colspan="2">J1939 Communication</td><td rowspan="3">No</td><td rowspan="3">No</td><td>Yes*3</td></tr>
<tr><td colspan="2">BACnet Communication</td><td rowspan="2">No</td></tr>
<tr><td colspan="2">EtherNet/IP Communication</td></tr>
</table>

*1 Only the All-in-One CPU module can perform various types of communication using serial port 1.
*2 The HMI-Ethernet port can be expanded and used for communication by connecting the HMI module to the CPU module.
*3 Only the CAN J1939 All-in-One CPU module can perform J1939 communication using the CAN port.
*4 Communication can be performed by connecting a communication cartridge to the cartridge slot.

Plus CPU Module

Yes: Can be used as a communication port.

No: Cannot be used as a communication port.

—: Not equipped with the CPU module.

<table>
<tr><th colspan="2" rowspan="2">Communication</th><th colspan="6">Port</th><th colspan="3">Cartridge Slot</th><th>Expansion module</th></tr>
<tr><th>Serial Port 1</th><th>USB Port</th><th>Ethernet Port 1</th><th>Ethernet Port 2</th><th>HMI-Ehternet Port</th><th>CAN Port</th><th>1</th><th>2</th><th>3</th><th>Communication module</th></tr>
<tr><td colspan="2">Maintenance Communication</td><td rowspan="11">—</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes*1</td><td rowspan="11">—</td><td rowspan="5">Yes*2</td><td rowspan="5">Yes*2</td><td rowspan="5">Yes*3</td><td rowspan="4">Yes</td></tr>
<tr><td rowspan="3">RS232C Communication/ RS485 Communication</td><td>User Communication</td><td rowspan="10">No</td><td rowspan="4">No</td><td rowspan="4">No</td><td rowspan="4">No</td></tr>
<tr><td>Modbus RTU Communication (master/slave)</td></tr>
<tr><td>Data Link Communication (master station/slave station)</td></tr>
<tr><td>Bluetooth Communication</td><td>User Communication</td><td rowspan="7">No</td></tr>
<tr><td rowspan="3">Ethernet Communication</td><td>User Communication (server/client)</td><td rowspan="3">Yes</td><td rowspan="3">Yes</td><td rowspan="3">Yes*1</td><td rowspan="6">No</td><td rowspan="6">No</td><td rowspan="6">No</td></tr>
<tr><td>User Communication (UDP)</td></tr>
<tr><td>Modbus TCP Communication (server/client)</td></tr>
<tr><td colspan="2">J1939 Communication</td><td>No</td><td rowspan="2">No</td><td rowspan="3">No</td></tr>
<tr><td colspan="2">BACnet Communication</td><td>Yes</td></tr>
<tr><td colspan="2">EtherNet/IP Communication</td><td>No</td><td>Yes</td></tr>
</table>

*1 The HMI-Ethernet port can be expanded and used for communication by connecting the HMI module to the CPU module.

*2 Communication can be performed by connecting the cartridge base module to the CPU module and then connecting a communication cartridge to the cartridge slot.

*3 Communication can be performed by connecting the HMI module to the CPU module and then connecting a communication cartridge to the cartridge slot.

Communication Interfaces

An overview and the specifications of the communication interfaces are shown below.

USB Port

Maintenance communication can be performed by using this port to connect to a computer.

Communication Type	USB2.0 Full speed, CDC class
Communication Functions	Capable of maintenance communication with a PC, program downloads via USB power
Connector	USB mini-B
Isolation between Internal Circuit	Not isolated

Serial Port 1

This port can be used to communicate with RS232C/RS485 communication-compatible external devices such as computers, operator interfaces, and printers.

Maintenance communication, user communication, Modbus RTU communication (master/slave), and data link communication (master station/slave station) are possible. Only the All-in-One CPU module is equipped with serial port 1.

Communication Type	EIA RS-232C or RS-485 software selectable
Maximum Communication Speed	115,200 bps
Communication Functions	Maintenance communication, user communication, Modbus RTU communication (master/slave), data link communication (master station/slave station)
Connector	RJ45
Cable	RS232C: Shielded multicore RS485: Shielded twisted-pair
Maximum Cable Length	RS-232C: 5 m RS-485 : 200 m
Isolation between Internal Circuit	Not isolated

Serial Communication Module

Serial communication modules can be used by connecting them to the basic expansion side or expansion interface side of the CPU module.

This port can be used to communicate with RS232C/RS485 communication-compatible external devices such as computers, operator interfaces, and printers. Maintenance communication, user communication, Modbus RTU communication (master/slave), and data link communication (master station/slave station) are possible.

Type No.		FC6A-SIF52, FC6A-SIF524
Points		2
Electrical Characteristics		EIA RS232C/EIA RS485*1
Maximum Communication Speed		115,200 bps
Communication Functions		Maintenance communication, user communication, Modbus RTU communication (master/slave), data link communication (master station/slave station)
Maximum Cable Length		15 m (EIA RS232C)/1,200 m (EIA RS485)
Isolation between Internal Circuit	**Between Communication Port and Internal Circuit**	Photocoupler
	Between Ports	Transformer
Cable	**Recommended Cable**	RS-232C: 24 AWG shielded multicore cable RS-485: 24 AWG twisted-pair shielded cable

*1 Specify the values in the WindLDR **Function Area Settings**. The combination of "Data Bits: 7 bits" and "Parity: None" cannot be set.

RS232C Communication Cartridge and RS485 Communication Cartridge

The RS232C communication cartridge and RS485 communication cartridge can be used by connecting them to the following cartridge slots.

- Cartridge slot 1 and 2 of the All-in-One CPU module/CAN J1939 All-in-One CPU module
- Cartridge slot 1 and 2 of a cartridge base module added to the Plus CPU module
- Cartridge slot 3 of an HMI module added to the Plus CPU module

This port can be used to communicate with RS232C/RS485 communication-compatible external devices such as computers, operator interfaces, and printers. Maintenance communication, user communication, Modbus RTU communication (master/slave), and data link communication (master station/slave station) are possible.

Type No.		FC6A-PC1	FC6A-PC3
Electrical Characteristics		EIA RS232C	EIA RS485
Maximum Communication Speed		115,200 bps	115,200 bps
Communication Functions		Maintenance communication, user communication, Modbus RTU communication (master/slave), data link communication (master station/slave station)	
Maximum Cable Length		5 m	200 m
Isolation between Internal Circuit		Not isolated	Not isolated
Cable	**Recommended Cable**	Shielded multicore: 24 AWG	Shielded twisted-pair: 24 AWG

Bluetooth Communication Cartridge

The Bluetooth communication cartridge can be used by connecting it to the following cartridge slots.

- Cartridge slot 1 and 2 of the All-in-One CPU module/CAN J1939 All-in-One CPU module
- Cartridge slot 1 and 2 of a cartridge base module added to the Plus CPU module
- Cartridge slot 3 of an HMI module added to the Plus CPU module

Using Bluetooth, the FC6A Series MICROSmart can communicate with external devices that support Bluetooth communication, such as computers, smartphones, and barcode readers. Maintenance communication and user communication are possible.

Type No.	FC6A-PC4
Electrical Characteristics	Bluetooth ver.2.1 + EDR
Profile	SPP (Serial Port Profile) iAP (iPod Accessory Protocol)
Communication Functions	Maintenance communication, user communication
Transmission Distance	10 m (Class 2)

Ethernet Port 1 and 2

This port can be used to communicate with Ethernet communication-compatible external devices such as computers and operator interfaces.

Multiple connections are available for the Ethernet port, and different communication protocols can be used in those connections simultaneously. Each connection can be configured for maintenance communication (server), user communication (server/client), user communication (UDP), or Modbus TCP communication (server/client).

Yes: Can be used as a communication port.

No: Cannot be used as a communication port.

—: Not equipped with the CPU module.

<table>
<tr><th colspan="2" rowspan="2">Communication</th><th colspan="2">All-in-One CPU Module/
J1939 All-in-One CPU Module</th><th colspan="2">Plus CPU Module</th></tr>
<tr><th>Ethernet Port 1</th><th>Ethernet Port 2</th><th>Ethernet Port 1</th><th>Ethernet Port 2</th></tr>
<tr><td colspan="2">Maintenance Communication</td><td rowspan="4">Yes</td><td rowspan="6">—</td><td rowspan="5">Yes</td><td rowspan="4">Yes</td></tr>
<tr><td rowspan="3">Ethernet Communication</td><td>User Communication (server/client)</td></tr>
<tr><td>User Communication (UDP)</td></tr>
<tr><td>Modbus TCP Communication (server/client)</td></tr>
<tr><td colspan="2">BACnet Communication</td><td rowspan="2">No</td><td>No</td></tr>
<tr><td colspan="2">EtherNet/IP Communication</td><td>No</td><td>Yes</td></tr>
</table>

Communication Type	IEEE 802.3 compliant
Communication Speed	10BASE-T, 100BASE-TX
Number of Connections	All-in-One CPU module/CAN J1939 All-in-One CPU module: 8 Plus CPU module: 16
Communication Functions	Maintenance communication, user communication (server/client), user communication (UDP), Modbus TCP communication (server/client)
Connector	RJ45
Cable	CAT 5. or higher STP
Maximum Cable Length	100 m
Isolation between Internal Circuit	Pulse transformer isolated

Plus CPU module Ethernet port 1 supports the web server function, the send E-mail function, and the BACnet/IP communication function.

Web Server Function	Yes
Web Data Storage	FROM
Web Data Capacity[*1]	5.0 MB
Send E-mail Function	Yes
FTP Server/Client Function	Yes
BACnet/IP	B-ASC profile

*1 Among the web data capacity, available storage for the user depends on whether to use the system web page and the web page editor.

		System Web Page	
		Use	**Not Use**
Web Page Editor	**Use**	2.5 MB	3.0 MB
	Not Use	4.5 MB	5.0 MB

CAN Port

The CAN J1939 All-in-One CPU module can use this port to perform J1939 communication.

Communication Type	CAN bus communication
Communication Speed	250 kbps
Communication Functions	J1939 communication
Connector	FC6A-PMTE05PN02
Cable	SAE-J1939-11 : Shielded twisted-pair
	SAE-J1939-15 : Unshielded twisted-pair
Maximum Cable Length	SAE-J1939-11 : 40 m, stub 1 m maximum
	SAE-J1939-15 : 40 m, stub 3 m maximum
Terminating Resistance	120 Ω (0.5 W or higher)
Isolation between Internal Circuit	Power supply: Transformer isolated Signal: Galvanic isolation, photocoupler isolated

HMI-Ethernet Port

The HMI-Ethernet port can only be used when a CPU module and an HMI module are connected.
This port can be used to communicate with Ethernet communication-compatible external devices such as computers and operator interfaces.
This port has eight connections that can be used with Ethernet communication. Each connection can be configured for maintenance communication (server).
This port also supports the web server function and the send E-mail function.

Communication Type	IEEE 802.3 compliant
Communication Speed	10BASE-T, 100BASE-TX
Number of Connections	8 maximum
Communication Mode	Maintenance Communication
Web Server Function	Yes
Web Data Storage	FROM
Web Data Capacity[*1]	5.0 MB
Send E-mail Function	Yes
Connector	RJ45
Cable	CAT 5. or higher STP
Maximum Cable Length	100 m
Isolation between Internal Circuit	Pulse transformer isolated

*1 Among the web data capacity, available storage for the user depends on whether to use the system web page and the web page editor.

		System Web Page	
		Use	**Not Use**
Web Page Editor	**Use**	2.5 MB	3.0 MB
	Not Use	4.5 MB	5.0 MB

Caution

- When accessing the FC6A Series MICROSmart over the Internet, adequate security measures for the network to prevent unauthorized access are required. Be sure to consult your network administrator or Internet service provider. IDEC bears no responsibility for damages or problems caused due to security in Ethernet communication.
- Restrict the access to FC6A Series MICROSmart with IP addresses and ports by using appropriate measures such as the firewall.

List of CPU Modules and Communication Interfaces

The following are the communication interfaces that the CPU modules are equipped with or can be expanded with.
For the locations of the communication interfaces in each module, see Chapter 2 "Product Specifications" in the "FC6A Series MICROSmart User's Manual".

<table>
<tr><th>Type No.</th><th>USB Port</th><th>Serial Port 1</th><th>Communication Module Port 4 to 33*1</th><th>Communication Cartridge</th><th>Ethernet Port 1</th><th>Ethernet Port 2</th><th>CAN Port</th><th>HMI-Ethernet Port*4</th></tr>
<tr><td>FC6A-C16R1AE</td><td rowspan="35">1</td><td rowspan="18">1</td><td rowspan="25">6 maximum</td><td rowspan="11">1 maximum*2</td><td rowspan="35">1</td><td rowspan="25">—</td><td rowspan="18">—</td><td rowspan="35">1 maximum</td></tr>
<tr><td>FC6A-C16R1CE</td></tr>
<tr><td>FC6A-C16K1CE</td></tr>
<tr><td>FC6A-C16P1CE</td></tr>
<tr><td>FC6A-C16R1DE</td></tr>
<tr><td>FC6A-C16K1DE</td></tr>
<tr><td>FC6A-C16P1DE</td></tr>
<tr><td>FC6A-C24R1AE</td></tr>
<tr><td>FC6A-C24R1CE</td></tr>
<tr><td>FC6A-C24K1CE</td></tr>
<tr><td>FC6A-C24P1CE</td></tr>
<tr><td>FC6A-C40R1AE</td><td rowspan="14">2 maximum*2</td></tr>
<tr><td>FC6A-C40R1CE</td></tr>
<tr><td>FC6A-C40K1CE</td></tr>
<tr><td>FC6A-C40P1CE</td></tr>
<tr><td>FC6A-C40R1DE</td></tr>
<tr><td>FC6A-C40K1DE</td></tr>
<tr><td>FC6A-C40P1DE</td></tr>
<tr><td>FC6A-C40R1AEJ</td><td rowspan="17">—</td><td rowspan="7">1</td></tr>
<tr><td>FC6A-C40R1CEJ</td></tr>
<tr><td>FC6A-C40K1CEJ</td></tr>
<tr><td>FC6A-C40P1CEJ</td></tr>
<tr><td>FC6A-C40R1DEJ</td></tr>
<tr><td>FC6A-C40K1DEJ</td></tr>
<tr><td>FC6A-C40P1DEJ</td></tr>
<tr><td>FC6A-D16R1CEE</td><td rowspan="10">30 maximum</td><td rowspan="10">3 maximum*3</td><td rowspan="10">1</td><td rowspan="10">—</td></tr>
<tr><td>FC6A-D16R4CEE</td></tr>
<tr><td>FC6A-D16K1CEE</td></tr>
<tr><td>FC6A-D16K4CEE</td></tr>
<tr><td>FC6A-D16P1CEE</td></tr>
<tr><td>FC6A-D16P4CEE</td></tr>
<tr><td>FC6A-D32K3CEE</td></tr>
<tr><td>FC6A-D32K4CEE</td></tr>
<tr><td>FC6A-D32P3CEE</td></tr>
<tr><td>FC6A-D32P4CEE</td></tr>
</table>

*1 Serial communication modules can be used by connecting them to the basic expansion side or expansion interface side of the CPU module. The communication ports are assigned to port 4, port 5, and so on in order from the ports closest to the CPU module.
*2 The communication cartridges can be used by connecting them to cartridge slots 1 and 2 of the All-in-One CPU module/CAN J1939 All-in-One CPU module.
*3 Communication cartridges can be used by connecting them to cartridge slot 1 and 2 of a cartridge base module connected to the Plus CPU module and cartridge slot 3 of an HMI module connected to the Plus CPU module.
*4 The HMI-Ethernet port is available on the HMI module.

Allocations of the Communication Port Numbers

All-in-One CPU module, CAN J1939 All-in-One CPU module, and Plus CPU module support serial communication with the target device. Serial communication is possible with built-in interface or expanded interface of each CPU module. Interfaces supporting the serial communication are as follows:

All-in-One CPU module	:	Serial port 1, cartridge slots 1 and 2, communication module ports
CAN J1939 All-in-One CPU module	:	Cartridge slots 1 and 2, communication module ports
Plus CPU module	:	Cartridge slots 1 to 3, communication module ports

In order to use each interface as communication port, the communication parameters must be configured according to the communication parameters of the target device. Configure the communication parameters in **Communiction Ports** tab in the **Function Area Settings** dialog box. The allocation between each interface and communication port is described below.

For details on each setting, see "Communication Port Settings" on page 3-2.

■ Ports and Cartridge Slots 1 to 3 on the CPU Module and HMI Module

— : Not equipped with the CPU module.

No : Cannot be used as the serial communication ports.

<table>
<tr><th colspan="2" rowspan="2">CPU Module</th><th colspan="6">Port</th><th colspan="3">Cartridge Slot</th></tr>
<tr><th>Serial Port 1</th><th>USB Port</th><th>Ethernet Port 1</th><th>Ethernet Port 2</th><th>HMI-Ehternet Port</th><th>CAN Port</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td rowspan="3">All-in-One CPU module</td><td>16-I/O type</td><td rowspan="3">Port 1</td><td rowspan="6">No</td><td rowspan="6">No</td><td rowspan="4">—</td><td rowspan="6">No*1</td><td rowspan="3">—</td><td rowspan="4">Port 2
*2*5</td><td rowspan="2">—</td><td rowspan="4">—</td></tr>
<tr><td>24-I/O type</td></tr>
<tr><td>40-I/O type</td><td rowspan="2">Port 3
*2*5</td></tr>
<tr><td>CAN J1939 All-in-One CPU module</td><td>40-I/O type</td><td rowspan="3">—</td><td>No</td></tr>
<tr><td rowspan="2">Plus CPU module</td><td>Plus 16-I/O type</td><td rowspan="2">No</td><td rowspan="2">—</td><td rowspan="2">Port 1
*3*5</td><td rowspan="2">Port 2
*3*5</td><td rowspan="2">Port 3
*4*5</td></tr>
<tr><td>Plus 32-I/O type</td></tr>
</table>

*1 When the HMI module is connected to the CPU module and HMI-Ethernet port is added.
*2 When a communication cartridge is installed on the cartridge slot.
*3 When the cartridge base module is connected to the CPU module and a communication cartridge is installed on the cartridge slot.
*4 When the HMI module is connected to the CPU module and a communication cartridge is installed on the cartridge slot.
*5 The combination of "Data Bits: 7" and "Parity: None" is not possible.

Note: For the locations of serial port 1 and cartridge slots 1 and 2, and for how to wire serial port 1, communication modules, and communication cartridges, see Chapter 2 "Product Specifications" in the FC6A Series MICROSmart User's Manual.

■ Ports on Communication Modules

<table>
<tr><th colspan="2" rowspan="2">CPU Module</th><th colspan="6">Communication Module</th></tr>
<tr><th>1st</th><th>2nd</th><th>3rd</th><th>4th</th><th>···</th><th>15th</th></tr>
<tr><td rowspan="3">All-in-One CPU module</td><td>16-I/O type</td><td rowspan="6">Port 4, 5</td><td rowspan="6">Port 6, 7</td><td rowspan="6">Port 8, 9</td><td rowspan="4">*1</td><td rowspan="4">*1</td><td rowspan="4">*1</td></tr>
<tr><td>24-I/O type</td></tr>
<tr><td>40-I/O type</td></tr>
<tr><td>CAN J1939 All-in-One CPU module</td><td>40-I/O type</td></tr>
<tr><td rowspan="2">Plus CPU module</td><td>Plus 16-I/O type</td><td rowspan="2">Port 10, 11</td><td rowspan="2">···</td><td rowspan="2">Port 32, 33</td></tr>
<tr><td>Plus 32-I/O type</td></tr>
</table>

*1 Four or more communication modules cannot be connected to the All-in-One CPU module and CAN J1939 All-in-One CPU module.

Communication Functions Overview

The FC6A Series MICROSmart supports maintenance communication, user communication, Modbus communication, data link communication, J1939 communication, and BACnet communication.

This section describes an overview of and connection examples for the communication functions.

Maintenance Communication

The maintenance communication of the FC6A Series MICROSmart enables you to check the operating status and I/O status of the FC6A Series MICROSmart, monitor and change device values, and download and upload user programs with the PLC programming software WindLDR installed on a computer. For details on maintenance communication, see "Maintenance Communication" on page 4-1.

Supported ports[*1]

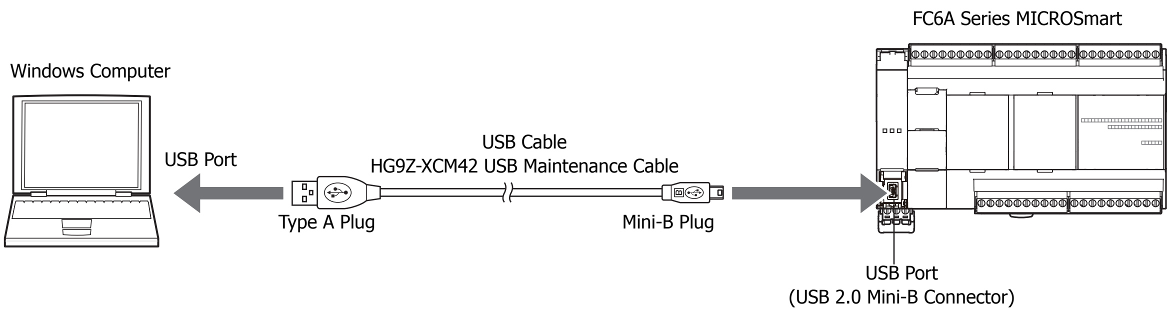
USB Port	Serial Port 1	Ethernet Port 1, 2 and HMI-Ethernet Port[*2]	Communication Cartridge and Communication Module	CAN Port
Yes	Yes	Yes	Yes	No

*1 Depending on the port that will be used, there are restrictions on the maintenance communication methods that can be used. For details on the restrictions, see the "Maintenance Communication" on page 4-1.
*2 Only maintenance communication can be used with the HMI-Ethernet port.

Note: When an HMI module is connected, maintenance communication can be performed by using the HMI-Ethernet port. For details, see Chapter 7 "HMI Function" in "FC6A Series MICROSmart User's Manual".

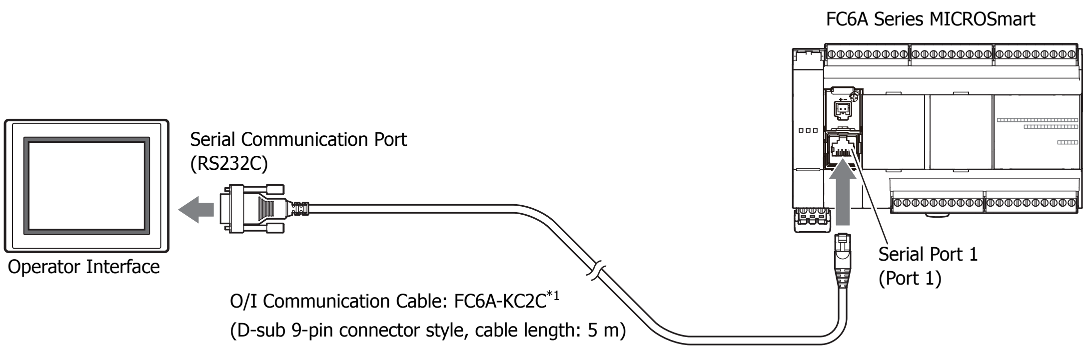
■ 1:1 Maintenance Communication Using USB Port

This example shows a 1:1 maintenance communication system in which a FC6A Series MICROSmart and a computer are connected with USB. The USB maintenance cable (HG9Z-XCM42) is used.



■ 1:1 Maintenance Communication Example with an IDEC Operator Interface Using Serial Port 1

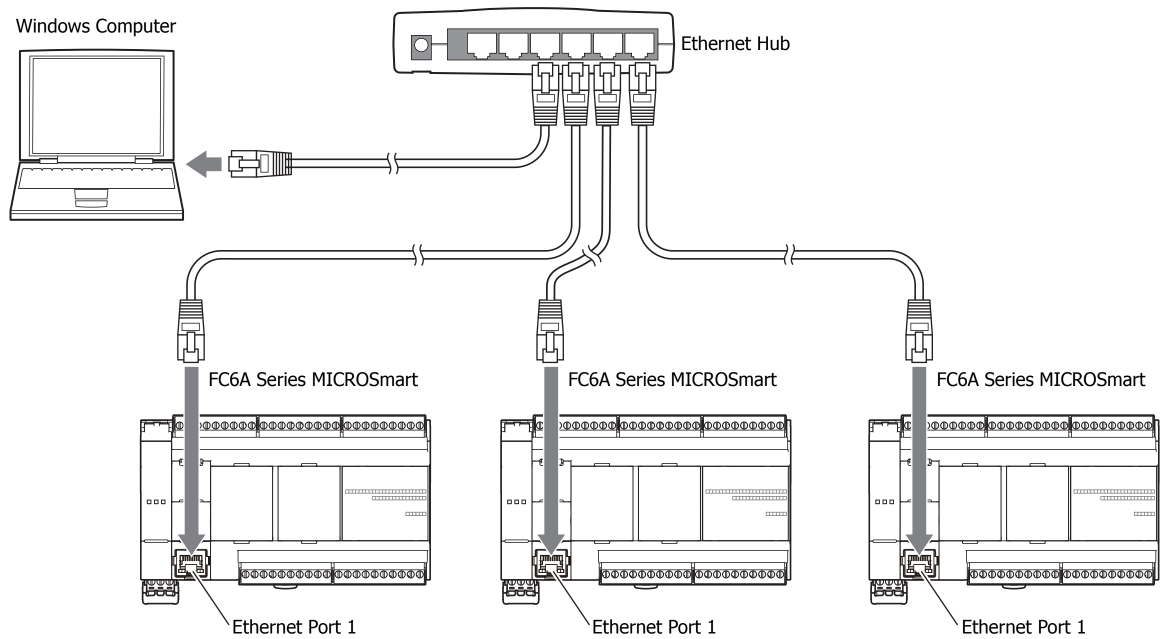
This example shows maintenance communication between the FC6A Series MICROSmart and an operator interface, as well as monitoring and changing FC6A Series MICROSmart device values using the operator interface. An IDEC operator interface is connected to serial port 1 of the FC6A Series MICROSmart.



*1 For details on O/I communication cables, see Appendix "Cables" in the "FC6A Series MICROSmart User's Manual".

■ **1:N Maintenance Communication Using Ethernet Port 1**

This example shows a 1:N maintenance communication system in which three FC6A Series MICROSmart and a computer are connected over Ethernet. The Ethernet cables are connected to the Ethernet port 1 of three FC6A Series MICROSmart, and those FC6A Series MICROSmart are connected to the computer via an Ethernet hub.



User Communication

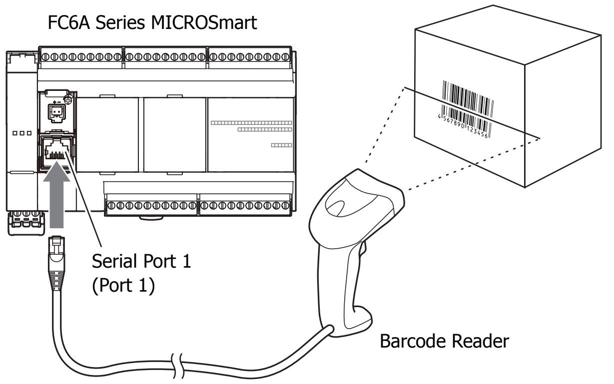
The user communication of the FC6A Series MICROSmart enables you to control external devices such as computers, printers, and barcode readers. For details on user communication, see "User Communication Instructions" on page 5-1.

Supported ports

USB Port	Serial Port 1	Ethernet Port 1 and 2	Communication Cartridge and Communication Module	CAN Port
No	Yes	Yes	Yes	No

■ **User Communication Using Serial Port 1**

This example shows a system in which a FC6A Series MICROSmart receives the data read by a barcode reader. A barcode reader is connected to port 1 of the FC6A Series MICROSmart.



Modbus Communication

The FC6A Series MICROSmart supports Modbus RTU protocol and can be used as either a Modbus communication master or slave. With Modbus communication, the FC6A Series MICROSmart can monitor and modify the data of inverters and temperature controllers.

Ethernet ports 1 and 2 also support the Modbus TCP communication protocol.

For details on Modbus communication, see "Modbus Communication" on page 6-1.

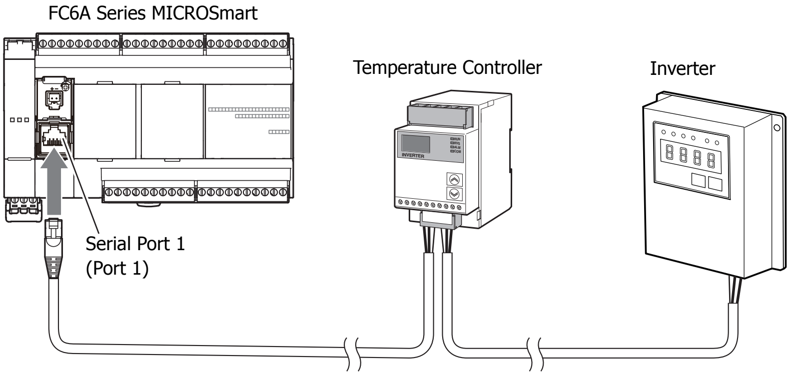
Supported ports

USB Port	Serial Port 1	Ethernet Port 1 and 2	Communication Cartridge*1 and Communication Module	CAN Port
No	Yes	Yes	Yes	No

*1 Only the RS-232C communication cartridge and RS-485 communication cartridge are supported.

■ Modbus RTU Communication Using Serial Port 1

This example shows a system in which a FC6A Series MICROSmart is communicating with a temperature controller and an inverter that support Modbus RTU. The A temperature controller is connected to port 1 of the FC6A Series MICROSmart.



Data Link Communication

The FC6A Series MICROSmart supports data link communication, and it can share data between CPU modules using serial port 1 and cartridge slots. The FC6A Series MICROSmart can also share data with FC5A Series and FC4A Series CPU modules. Configure the settings in WindLDR to enable distributed control of a maximum of 31 CPU modules.

For details about the data link communication, see "Data Link Communication" on page 7-1.

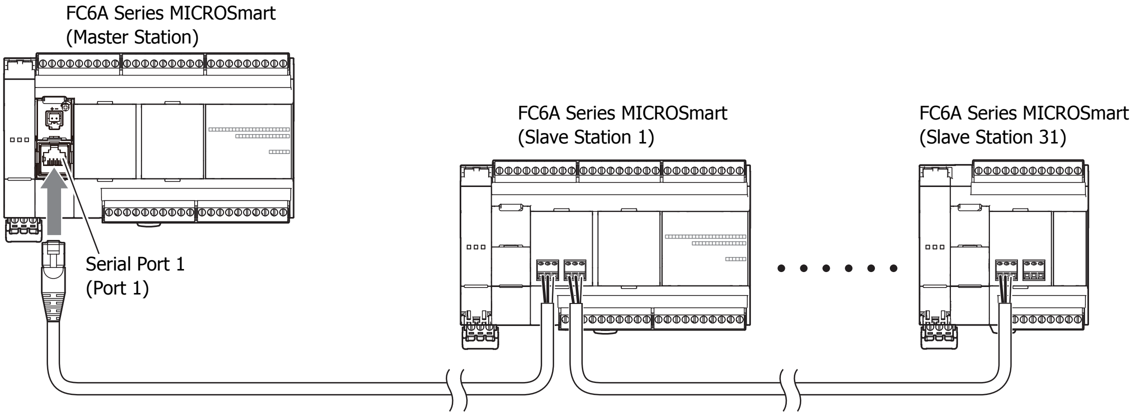
Supported ports

USB Port	Serial Port 1	Ethernet Port 1 and 2	Communication Cartridge*1 and Communication Module	CAN Port
No	Yes	No	Yes	No

*1 Only the RS-232C communication cartridge and RS-485 communication cartridge are supported.

■ Data Link Communication Using Serial Port 1

This example shows communication between multiple CPU modules with the FC6A Series MICROSmart as the master station. A slave station CPU module is connected to port 1 of the FC6A Series MICROSmart.



J1939 Communication

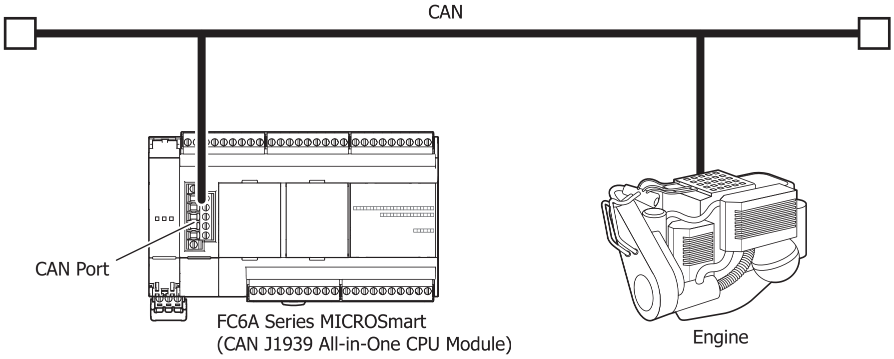
The CAN J1939 All-in-One CPU module can be connected to a J1939 communication network using the CAN port and it can communicate with other J1939 communication-compatible devices. Messages that conform to the SAE J1939 standard can be sent and received.
For details on J1939 communication, see "J1939 Communication" on page 8-1.

Supported ports

USB Port	Serial Port 1	Ethernet Port 1 and 2	Communication Cartridge	CAN Port
No	No	No	No	Yes

■ J1939 Communication Using CAN Port

This example shows the FC6A Series MICROSmart communicating with a J1939-compatible engine. The CAN port of the CAN J1939 All-in-One CPU module is connected to the engine.



BACnet Communication

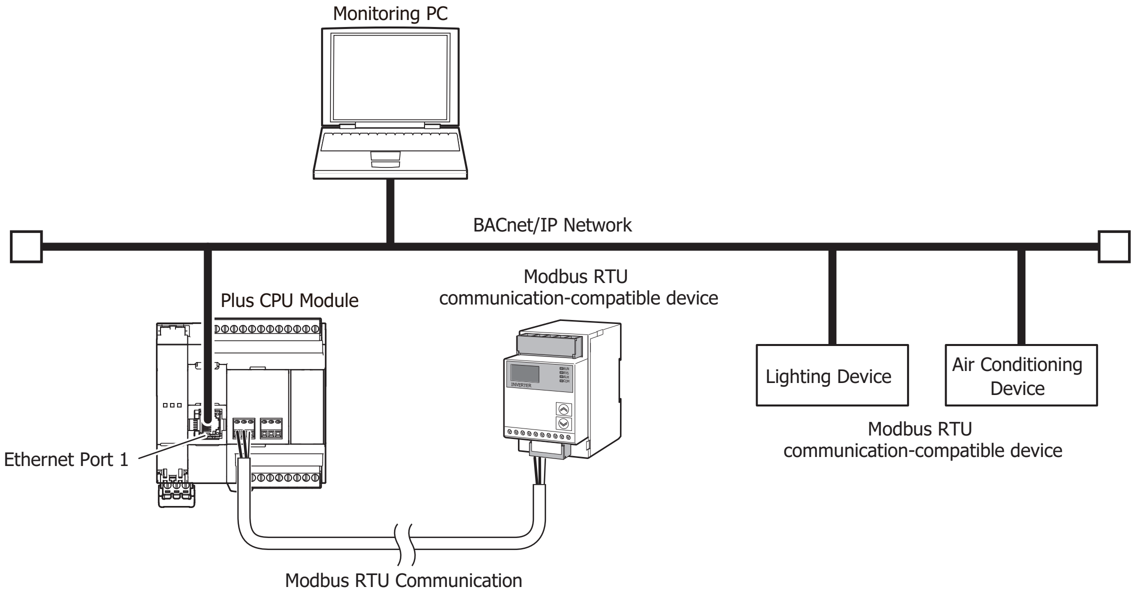
The Plus CPU module can be connected to a BACnet/IP network using Ethernet port 1 and communicate with other BACnet communication-compatible external devices. For details on BACnet communication, see "BACnet/IP" on page 15-1.

Supported ports

USB Port	Serial Port 1	Ethernet Port 1	Ethernet Port 2	Communication Cartridge and Communication Module	CAN Port
No	No	Yes	No	No	No

■ BACnet Communication Using Ethernet Port 1

This example shows the Plus CPU module aggregating information from Modbus RTU communication-compatible devices, communicating with a BACnet communication-compatible device connected to a BACnet/IP network, and making that information publicly available.



EtherNet/IP communication

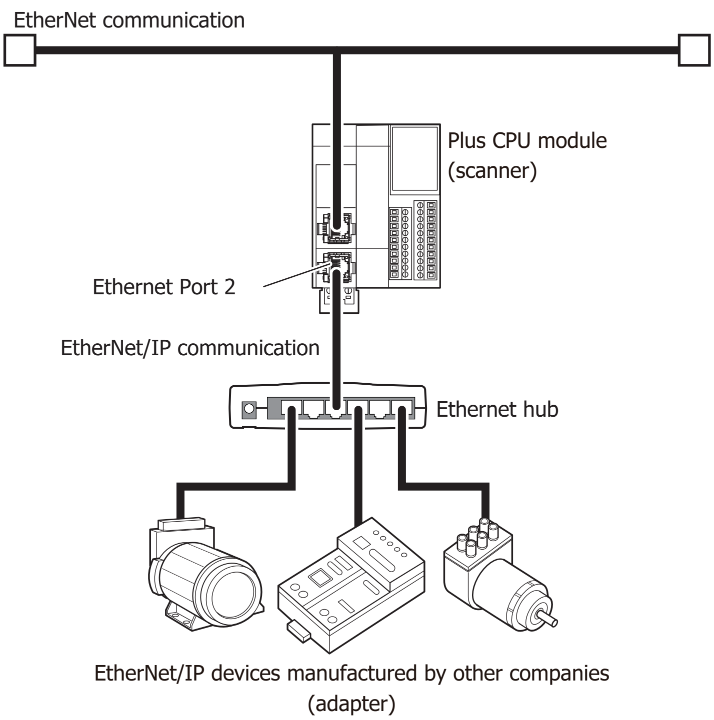
The Plus CPU module can be connected to an existing Ethernet network using Ethernet port 2 and communicate with other EtherNet/IP communication-compatible devices. EtherNet/IP communication uses standard Ethernet technologies, which allows networks to be built that include various Ethernet-compatible devices. For details on EtherNet/IP communication, see "EtherNet/IP Communication" on page 16-1.

Supported ports

USB Port	Serial Port 1	Ethernet Port 1	Ethernet Port 2	Communication Cartridge and Communication Module	CAN Port
No	No	No	Yes	No	No

■ EtherNet/IP Communication Using Ethernet Port 2

This example shows the Plus CPU module communicating with EtherNet/IP communication-compatible devices and controlling those devices.



2: DEVICES

This chapter provides detailed descriptions of the allocations of devices such as inputs, outputs, internal relays, registers, timers, and counters that are used in the basic and advanced instructions, as well as details about the allocations of special internal relays and special data registers.

Please use this chapter as a reference for the devices.

Note: The programing and operation of FC6A Series MICROSmart user programs requires specialist knowledge.
Take the time to develop a thorough understanding of the contents and programs in this manual before using the FC6A Series MICROSmart.

Device Addresses

All-in-One CPU Module/J1939 All-in-One CPU Module

Device	Symbol	Unit	Range (Points)		
			16-I/O Type	24-I/O Type	40-I/O Type
Inputs*1	I	Bit	I0 - I10 (9 points)	I0 - I15 (14 points)	I0 - I27 (24 points)
Expansion Inputs*1	I	Bit	I30 - I187 (128 points) I190 - I507*2 (256 points) I630 - I633*4 (4 points)	I30 - I307 (224 points) I310 - I627*3 (256 points) I630 - I633*4 (4 points)	I30 - I307 (224 points) I310 - I627*3 (256 points) I630 - I637*4 (8 points)
Output*1	Q	Bit	Q0 - Q6 (7 points)	Q0 - Q11 (10 points)	Q0 - Q17 (16 points)
Expansion Outputs*1	Q	Bit	Q30 - Q187 (128 points) Q190 - Q507*2 (256 points) Q630 - Q633*4 (4 points)	Q30 - Q307 (224 points) Q310 - Q627*3 (256 points) Q630 - Q633*4 (4 points)	Q30 - Q307 (224 points) Q310 - Q627*3 (256 points) Q630 - Q637*4 (8 points)
Internal Relay*1	M	Bit	M0 - M7997 (6,400 points) M10000 - M17497 (6,000 points)		
Special Internal Relay*1	M	Bit	M8000 - M8317 (256 points)		
Shift Register	R	Bit	R0 - R255 (256 points)		
Timer	T	Bit/Word	T0 - T1023 (1,024 points)		
Counter	C	Bit/Word	C0 - C511 (512 points)		
Data Register	D	Bit/Word	D0000 - D7999 (8,000 points) D10000 to D55999 (46,000 points)		
Special Data Register	D	Bit/Word	D8000 - D8499 (500 points)		

*1 The least significant digit of the device address is an octal number (0 to 7).
*2 I190 to I507 and Q190 to Q507 are devices that can only be used when expansion modules are connected at the expansion interface side using the expansion interface module (expander).
*3 I310 to I627 and Q310 to Q627 are devices that can only be used when expansion modules are connected at the expansion interface side using the expansion interface module (expander).
*4 I630 to I637 and Q630 to Q637 are devices that can be used only when I/O cartridges are connected.

Plus CPU module

Device	Symbol	Unit	Range (Points)	
			Plus 16-I/O Type	Plus 32-I/O Type
Inputs[*1]	I	Bit	I0 - I7 (8 points)	I0 - I17 (16 points)
Expansion Inputs[*1]	I	Bit	I30 - I307 (224 points) I310 - I627[*2] (256 points) I630 - I643[*3] (12 points) I1000 - I10597[*4] (2,016 points)	
Output[*1]	Q	Bit	Q0 - Q7 (8 points)	Q0 - Q17 (16 points)
Expansion Outputs[*1]	Q	Bit	Q30 - Q307 (224 points) Q310 - Q627[*2] (256 points) Q630 - Q643[*3] (12 points) Q1000 - Q10597[*4] (2,016 points)	
Internal Relay[*1]	M	Bit	M0 - M7997 (6,400 points) M10000 - M21247 (9,000 points)	
Special Internal Relay[*1]	M	Bit	M8000 - M9997 (1,600 points)	
Shift Register	R	Bit	R0 - R255 (256 points)	
Timer	T	Bit/Word	T0 - T1999 (2,000 points)	
Counter	C	Bit/Word	C0 - C511 (512 points)	
Data Register	D	Bit/Word	D0000 - D7999 (8,000 points) D10000 - D61999 (52,000 points)	
Non-retentive Data Register	D	Bit[*7]/Word	D70000 - D269999[*5] (200,000 points)	
Special Data Register	D	Bit/Word	D8000 - D8899 (900 points)	
Index Register[*6]	P	2 words	P0 - P15 (16 points)	

*1 The least significant digit of the device address is an octal number (0 to 7).
*2 I1310 to I627 and Q310 to Q627 are devices that can only be used when expansion modules are connected at the expansion interface side using the expansion interface module (expander). (Node 0)
*3 I630 to I643 and Q630 to Q643 are devices that can be used only when I/O cartridges are connected.
*4 I1000 to I10597 and Q1000 to Q10597 are devices that can be used only when expansion modules are connected using the expansion interface modules (remote master and slaves) and the expansion interface modules (expander). (Node 1 to 10)
*5 D70000 to D269999 cannot be designated as "Keep." Data register values are kept when you switch the PLC from Stop to Run, but those values are cleared at power up.
*6 The data type that can be used is L (Long) only.
*7 Usable only in scripts executed by the SCRPT instruction and as argument devices used in UMACRO instructions.

■ **Inputs (I), Expansion Inputs (I)**
Devices that input on/off information from external devices to the FC6A Series MICROSmart.

■ **Outputs (Q), Expansion Outputs (Q)**
Devices that output on/off information from the FC6A Series MICROSmart to external devices.

■ **Internal Relays (M)**
Bit devices used internally on the FC6A Series MICROSmart.

■ **Special Internal Relays (M)**
Bit devices used internally on the FC6A Series MICROSmart. Special functions are assigned to each bit.

■ **Shift Registers (R)**
Bit devices that are used with the SFR instruction and the SFRN instruction. The bit sequence of the data is shifted according to pulse input.

■ **Timer (T)**
Timers used internally in the FC6A Series MICROSmart. There are three devices: Timer bits (symbol: T, unit: bit), timer preset values (symbol: TP, unit: word), and timer current values (symbol: TC, unit: word).

These can be used as an on-delay timer or an off-delay timer. For details on timers (T), see Chapter 3 "Using Timer or Counter as Source Device" in the "FC6A Series MICROSmart Ladder Programming Manual".

■ **Counters (C)**
Counters used internally in the FC6A Series MICROSmart. There are three devices: Counter bits (symbol: C, unit: bit), counter preset values (symbol: CP, unit: word), and counter current values (symbol: CC, unit: word). These can be used as an adding counter or a reversible counter. For details on counters (C), see Chapter 3 "Using Timer or Counter as Source Device" in the "FC6A Series MICROSmart Ladder Programming Manual".

■ **Data Registers (D)**
Word devices that are used for writing numerical data internally in the FC6A Series MICROSmart. These can also be used as bit devices.

■ **Special Data Registers (D)**
Word devices that are used for writing numerical data internally in the FC6A Series MICROSmart. Special functions are assigned to each data register. These can also be used as bit devices.

Notes:

- Although the device symbol for internal relays (M0000 to M7997, M10000 to M21247) and special internal relays (M8000 to M9997) is the same ("M"), the device characteristics are different. Special functions are assigned to each bit of the special internal relays.
- Although the device symbol for the data registers (D0000 to D7999, D10000 to D61999, D70000 to D269999) and special data registers (D8000 to D8899) is the same ("D"), the device characteristics are different. Special functions are assigned to each special data register.

Special Internal Relay

Special Internal Relay Device Addresses

⚠Warning	Do not write to data in the area marked as reserved in the special internal relays list. Otherwise the system may not operate correctly.

Note: R/W is an abbreviation for read/write.
The notation for the R/W field is as follows.
R/W: The device can be both read from and written to
R: Read-only
W: Write-only

Device Address	Description		When Stopped	Power OFF	R/W
M8000	Start Control		Maintained	Maintained	R/W
M8001	1-s Clock Reset		Cleared	Cleared	R/W
M8002	All Outputs OFF		Cleared	Cleared	R/W
M8003	Carry (Cy) or Borrow (Bw)		Cleared	Cleared	R/W
M8004	User Program Execution Error		Cleared	Cleared	R/W
M8005	Communication Error		Maintained	Cleared	R/W
M8006	Communication Prohibited Flag (When Data Link Master)		Maintained	Maintained	R/W
M8007	Initialization Flag (When Data Link Master)/Stop Communication Flag (When Data Link Slave)		Cleared	Cleared	R/W
M8010	Status LED Operation		Operating	Cleared	R/W
M8011 M8012	**— Reserved —**		—	—	—
M8013	Calendar/Clock Data Write/Adjust Error Flag		Operating	Cleared	R/W
M8014	Calendar/Clock Data Read Error Flag		Operating	Cleared	R/W
M8015	**— Reserved —**		—	—	—
M8016	Calendar Data Write Flag		Operating	Cleared	R/W
M8017	Clock Data Write Flag		Operating	Cleared	R/W
M8020	Calendar/Clock Data Write Flag		Operating	Cleared	R/W
M8021	Clock Data Adjust Flag		Operating	Cleared	R/W
M8022	User Communication Receive Instruction Cancel Flag (Port 1)		Cleared	Cleared	R/W
M8023	User Communication Receive Instruction Cancel Flag (Port 2)		Cleared	Cleared	R/W
M8024	BMOV/WSFT Executing Flag		Maintained	Maintained	R/W
M8025	Maintain Outputs While Stopped		Maintained	Cleared	R/W
M8026	User Communication Receive Instruction Cancel Flag (Port 3)		Cleared	Cleared	R/W
M8027	High-speed Counter (Group 1/I0)	Count Direction Flag	Maintained	Cleared	R/W
M8030		Comparison Output Reset	Cleared	Cleared	R/W
M8031		Gate Input	Maintained	Cleared	R/W
M8032		Reset Input	Maintained	Cleared	R/W
M8033	User Communication Receive Instruction Cancel Flag (Port 4)		Cleared	Cleared	R/W
M8034	High-speed Counter (Group 3/I3)	Comparison Output Reset	Cleared	Cleared	R/W
M8035		Gate Input	Maintained	Cleared	R/W
M8036		Reset Input	Maintained	Cleared	R/W
M8037	**— Reserved —**		—	—	—
M8040	High-speed Counter (Group 4/I4)	Comparison Output Reset	Cleared	Cleared	R/W
M8041		Gate Input	Maintained	Cleared	R/W
M8042		Reset Input	Maintained	Cleared	R/W
M8043	High-speed Counter (Group 5/I6)	Count Direction Flag	Maintained	Cleared	R/W
M8044		Comparison Output Reset	Cleared	Cleared	R/W
M8045		Gate Input	Maintained	Cleared	R/W
M8046		Reset Input	Maintained	Cleared	R/W
M8047 M8050	**— Reserved —**		—	—	—

Device Address	Description		When Stopped	Power OFF	R/W
M8051	High-speed Counter (Group 2/I1)	Comparison Output Reset	Cleared	Cleared	R/W
M8052		Gate Input	Maintained	Cleared	R/W
M8053		Reset Input	Maintained	Cleared	R/W
M8054		Comparison ON Status	Maintained	Cleared	R
M8055		Overflow	Maintained	Cleared	R
M8056	**— Reserved —**		—	—	—
M8057	High-speed Counter (Group 6/I7)	Comparison Output Reset	Cleared	Cleared	R/W
M8060		Gate Input	Maintained	Cleared	R/W
M8061		Reset Input	Maintained	Cleared	R/W
M8062		Comparison ON Status	Maintained	Cleared	R
M8063		Overflow	Maintained	Cleared	R
M8064 to M8067	**— Reserved —**		—	—	—
M8070	SD Memory Card Mount Status		Maintained	Cleared	R
M8071	Accessing SD Memory Card		Maintained	Cleared	R
M8072	Unmount SD Memory Card		Operating	Cleared	R/W
M8073	Function Switch Status		Operating	Cleared	R
M8074	Battery Voltage Measurement Flag		Operating	Cleared	R/W
M8075 to M8077	**— Reserved —**		—	—	—
M8080	Data Link Slave 1 Communication Completed Relay (When Data Link Master)		Operating	Cleared	R
M8081	Data Link Slave 2 Communication Completed Relay		Operating	Cleared	R
M8082	Data Link Slave 3 Communication Completed Relay		Operating	Cleared	R
M8083	Data Link Slave 4 Communication Completed Relay		Operating	Cleared	R
M8084	Data Link Slave 5 Communication Completed Relay		Operating	Cleared	R
M8085	Data Link Slave 6 Communication Completed Relay		Operating	Cleared	R
M8086	Data Link Slave 7 Communication Completed Relay		Operating	Cleared	R
M8087	Data Link Slave 8 Communication Completed Relay		Operating	Cleared	R
M8090	Data Link Slave 9 Communication Completed Relay		Operating	Cleared	R
M8091	Data Link Slave 10 Communication Completed Relay		Operating	Cleared	R
M8092	Data Link Slave 11 Communication Completed Relay		Operating	Cleared	R
M8093	Data Link Slave 12 Communication Completed Relay		Operating	Cleared	R
M8094	Data Link Slave 13 Communication Completed Relay		Operating	Cleared	R
M8095	Data Link Slave 14 Communication Completed Relay		Operating	Cleared	R
M8096	Data Link Slave 15 Communication Completed Relay		Operating	Cleared	R
M8097	Data Link Slave 16 Communication Completed Relay		Operating	Cleared	R
M8100	Data Link Slave 17 Communication Completed Relay		Operating	Cleared	R
M8101	Data Link Slave 18 Communication Completed Relay		Operating	Cleared	R
M8102	Data Link Slave 19 Communication Completed Relay		Operating	Cleared	R
M8103	Data Link Slave 20 Communication Completed Relay		Operating	Cleared	R
M8104	Data Link Slave 21 Communication Completed Relay		Operating	Cleared	R
M8105	Data Link Slave 22 Communication Completed Relay		Operating	Cleared	R
M8106	Data Link Slave 23 Communication Completed Relay		Operating	Cleared	R
M8107	Data Link Slave 24 Communication Completed Relay		Operating	Cleared	R
M8110	Data Link Slave 25 Communication Completed Relay		Operating	Cleared	R
M8111	Data Link Slave 26 Communication Completed Relay		Operating	Cleared	R
M8112	Data Link Slave 27 Communication Completed Relay		Operating	Cleared	R
M8113	Data Link Slave 28 Communication Completed Relay		Operating	Cleared	R
M8114	Data Link Slave 29 Communication Completed Relay		Operating	Cleared	R
M8115	Data Link Slave 30 Communication Completed Relay		Operating	Cleared	R
M8116	Data Link Slave 31 Communication Completed Relay		Operating	Cleared	R
M8117	Data Link All Slaves Communication Completed Relay		Operating	Cleared	R
M8120	Initialize Pulse		Cleared	Cleared	R
M8121	1-s Clock		Operating	Cleared	R
M8122	100-ms Clock		Operating	Cleared	R

Device Address	Description		When Stopped	Power OFF	R/W
M8123	10-ms Clock		Operating	Cleared	R
M8124	Timer/Counter Preset Value Changed		Maintained	Cleared	R
M8125	In-operation Output		Cleared	Cleared	R
M8126	1 Scan ON After Run-Time Download Completes		Cleared	Cleared	R
M8127	**— Reserved —**		—	—	—
M8130	High-speed Counter (Group 1/I0)	Reset Status	Maintained	Cleared	R
M8131		Comparison ON Status	Maintained	Cleared	R
M8132	**— Reserved —**		—	—	—
M8133	High-speed Counter (Group 3/I3)	Comparison ON Status	Maintained	Cleared	R
M8134	High-speed Counter (Group 4/I4)	Comparison ON Status	Maintained	Cleared	R
M8135	High-speed Counter (Group 5/I6)	Reset Status	Maintained	Cleared	R
M8136		Comparison ON Status	Maintained	Cleared	R
M8137	Interrupt Input I0 Status (Group 1/I0)	(ON: Allowed, OFF: Prohibited)	Cleared	Cleared	R
M8140	Interrupt Input I1 Status (Group 2/I1)		Cleared	Cleared	R
M8141	Interrupt Input I3 Status (Group 3/I3)		Cleared	Cleared	R
M8142	Interrupt Input I4 Status (Group 4/I4)		Cleared	Cleared	R
M8143	Interrupt Input I6 Status (Group 5/I6)		Cleared	Cleared	R
M8144	Timer Interrupt Status		Cleared	Cleared	R
M8145	User Communication Receive Instruction Cancel Flag (Port 5)		Cleared	Cleared	R/W
M8146	User Communication Receive Instruction Cancel Flag (Port 6)		Cleared	Cleared	R/W
M8147	User Communication Receive Instruction Cancel Flag (Port 7)		Cleared	Cleared	R/W
M8150	Comparison Result 1		Maintained	Cleared	R
M8151	Comparison Result 2		Maintained	Cleared	R
M8152	Comparison Result 3		Maintained	Cleared	R
M8153	Catch Input ON/OFF Status	Group 1/I0	Maintained	Cleared	R
M8154		Group 2/I1	Maintained	Cleared	R
M8155		Group 3/I3	Maintained	Cleared	R
M8156		Group 4/I4	Maintained	Cleared	R
M8157		Group 5/I6	Maintained	Cleared	R
M8160		Group 6/I7	Maintained	Cleared	R
M8161	High-speed Counter (Group 1/I0)	Overflow	Maintained	Cleared	R
M8162		Underflow	Maintained	Cleared	R
M8163	High-speed Counter (Group 5/I6)	Overflow	Maintained	Cleared	R
M8164		Underflow	Maintained	Cleared	R
M8165	High-speed Counter (Group 3/I3)	Overflow	Maintained	Cleared	R
M8166	High-speed Counter (Group 4/I4)	Overflow	Maintained	Cleared	R
M8167	Interrupt Input I7 Status (Group 6/I7)	(ON: Allowed, OFF: Prohibited)	Maintained	Cleread	R
M8170	User Communication Receive Instruction Cancel Flag (Port 8)		Cleared	Cleared	R/W
M8171	**— Reserved —**		—	—	—
M8172	Transistor Source Output Overcurrent Detection	Group 1	Operating	Cleared	R
M8173		Group 2	Operating	Cleared	R
M8174		Group 3	Operating	Cleared	R
M8175		Group 4	Operating	Cleared	R
M8176	User Communication Receive Instruction Cancel Flag (Port 9)		Cleared	Cleared	R/W
M8177 to M8183	**— Reserved —**		—	—	—
M8184	Change HMI Module Network Settings Trigger		Operating	Cleared	R/W
M8185	In Daylight Saving Time Period		Operating	Cleared	R
M8186	Ethernet Port 1 Executing Auto Ping		Operating	Cleared	R
M8187	Ethernet Port 1 Auto Ping Stop Flag		Operating	Cleared	R/W
M8190	Change CPU Module Ethernet Port 1 Network Settings Trigger		Operating	Cleared	R/W
M8191	SNTP Acquisition Flag		Operating	Cleared	R/W

Device Address	Description		When Stopped	Power OFF	R/W
M8192	Interrupt Input I0 Edge	On: Rising Edge Off: Falling Edge	Cleared	Cleared	R
M8193	Interrupt Input I3 Edge		Cleared	Cleared	R
M8194	Interrupt Input I4 Edge		Cleared	Cleared	R
M8195	Interrupt Input I6 Edge		Cleared	Cleared	R
M8196	Interrupt Input I7 Edge		Cleared	Cleared	R
M8197	Interrupt Input I1 Edge		Cleared	Cleared	R
M8200	User Communication Receive Instruction Cancel Flag	Connection 1	Cleared	Cleared	R/W
M8201		Connection 2	Cleared	Cleared	R/W
M8202		Connection 3	Cleared	Cleared	R/W
M8203		Connection 4	Cleared	Cleared	R/W
M8204		Connection 5	Cleared	Cleared	R/W
M8205		Connection 6	Cleared	Cleared	R/W
M8206		Connection 7	Cleared	Cleared	R/W
M8207		Connection 8	Cleared	Cleared	R/W
M8210	**— Reserved —**		—	—	—
M8211	HMI Module Send E-mail Server Settings Initialization		Operating	Cleared	R/W
M8212	Connection Status (ON: Connected, OFF: Not Connected)	Connection 1	Operating	Cleared	R
M8213		Connection 2	Operating	Cleared	R
M8214		Connection 3	Operating	Cleared	R
M8215		Connection 4	Operating	Cleared	R
M8216		Connection 5	Operating	Cleared	R
M8217		Connection 6	Operating	Cleared	R
M8220		Connection 7	Operating	Cleared	R
M8221		Connection 8	Operating	Cleared	R
M8222	Disconnect User Communication Connection	Connection 1	Operating	Cleared	R/W
M8223		Connection 2	Operating	Cleared	R/W
M8224		Connection 3	Operating	Cleared	R/W
M8225		Connection 4	Operating	Cleared	R/W
M8226		Connection 5	Operating	Cleared	R/W
M8227		Connection 6	Operating	Cleared	R/W
M8230		Connection 7	Operating	Cleared	R/W
M8231		Connection 8	Operating	Cleared	R/W
M8232	HMI Module Connection Information Reference Connection Status		Operating	Cleared	R
M8233 to M8247	**— Reserved —**		—	—	—
M8250	Download from SD Memory Card Execution Flag		Operating	Cleared	R/W
M8251	Upload to SD Memory Card Execution Flag		Operating	Cleared	R/W
M8252	Executing SD Memory Card Download		Operating	Cleared	R
M8253	Executing SD Memory Card Upload		Operating	Cleared	R
M8254	SD Memory Card Download/Upload Execution Completion Output		Operating	Cleared	R
M8255	SD Memory Card Download/Upload Execution Error Output		Operating	Cleared	R
M8256 M8257	**— Reserved —**		—	—	—
M8260	Write Recipe Execution Flag		Operating	Cleared	R/W
M8261	Read Recipe Execution Flag		Operating	Cleared	R/W
M8262	Executing Write Recipe		Operating	Cleared	R/W
M8263	Executing Read Recipe		Operating	Cleared	R/W
M8264	Recipe Execution Completed Output		Operating	Cleared	R/W
M8265	Recipe Execution Error Output		Operating	Cleared	R/W
M8266 to M8297	**— Reserved —**		—	—	—
M8300	J1939 Communication Permitted Flag		Cleared	Cleared	R/W
M8301	J1939 Online Status		Cleared	Cleared	R
M8302	J1939 Local Station Address Confirmation Status		Cleared	Cleared	R
M8303	J1939 Communication Error Output		Cleared	Cleared	R

Device Address	Description		When Stopped	Power OFF	R/W
M8304	J1939 Communication Bus Off Occurrence Output		Cleared	Cleared	R
M8305 to M8310	**— Reserved —**		—	—	—
M8311	ESC+Key Input (Up)	ESC+Key Input (▲)	Cleared	Cleared	R
M8312	ESC+Key Input (Down)	ESC+Key Input (▼)	Cleared	Cleared	R
M8313	ESC+Key Input (Left)	ESC+Key Input (◀)	Cleared	Cleared	R
M8314	ESC+Key Input (Right)	ESC+Key Input (▶)	Cleared	Cleared	R
M8315 to M8319	**— Reserved —**		—	—	—
M8320	Initialize Expansion Interface Remote Master/Slave		Operating	Cleared	R/W
M8321 to M8330	**— Reserved —**		—	—	—
M8331	Ethernet Port 2 Executing Auto Ping		Operating	Cleared	R
M8332	Ethernet Port 2 Auto Ping Stop Flag		Operating	Cleared	R/W
M8333	Change CPU Module Ethernet Port 2 Network Settings Trigger		Operating	Cleared	R/W
M8334	User Communication Receive Instruction Cancel Flag	Connection 9	Cleared	Cleared	R/W
M8335		Connection 10	Cleared	Cleared	R/W
M8336		Connection 11	Cleared	Cleared	R/W
M8337		Connection 12	Cleared	Cleared	R/W
M8340		Connection 13	Cleared	Cleared	R/W
M8341		Connection 14	Cleared	Cleared	R/W
M8342		Connection 15	Cleared	Cleared	R/W
M8343		Connection 16	Cleared	Cleared	R/W
M8344	Ethernet Port 1 Send E-mail Server Settings Initialization		Operating	Cleared	R/W
M8345	Connection Status (ON: Connected, OFF: Not Connected)	Connection 9	Operating	Cleared	R
M8346		Connection 10	Operating	Cleared	R
M8347		Connection 11	Operating	Cleared	R
M8350		Connection 12	Operating	Cleared	R
M8351		Connection 13	Operating	Cleared	R
M8352		Connection 14	Operating	Cleared	R
M8353		Connection 15	Operating	Cleared	R
M8354		Connection 16	Operating	Cleared	R
M8355	Disconnect User Communication Connection	Connection 9	Operating	Cleared	R/W
M8356		Connection 10	Operating	Cleared	R/W
M8357		Connection 11	Operating	Cleared	R/W
M8360		Connection 12	Operating	Cleared	R/W
M8361		Connection 13	Operating	Cleared	R/W
M8362		Connection 14	Operating	Cleared	R/W
M8363		Connection 15	Operating	Cleared	R/W
M8364		Connection 16	Operating	Cleared	R/W
M8365	User Communication Receive Instruction Cancel Flag (Port 10)		Cleared	Cleared	R/W
M8366	User Communication Receive Instruction Cancel Flag (Port 11)		Cleared	Cleared	R/W
M8367	User Communication Receive Instruction Cancel Flag (Port 12)		Cleared	Cleared	R/W
M8370	User Communication Receive Instruction Cancel Flag (Port 13)		Cleared	Cleared	R/W
M8371	User Communication Receive Instruction Cancel Flag (Port 14)		Cleared	Cleared	R/W
M8372	User Communication Receive Instruction Cancel Flag (Port 15)		Cleared	Cleared	R/W
M8373	User Communication Receive Instruction Cancel Flag (Port 16)		Cleared	Cleared	R/W
M8374	User Communication Receive Instruction Cancel Flag (Port 17)		Cleared	Cleared	R/W
M8375	User Communication Receive Instruction Cancel Flag (Port 18)		Cleared	Cleared	R/W
M8376	User Communication Receive Instruction Cancel Flag (Port 19)		Cleared	Cleared	R/W
M8377	User Communication Receive Instruction Cancel Flag (Port 20)		Cleared	Cleared	R/W
M8380	User Communication Receive Instruction Cancel Flag (Port 21)		Cleared	Cleared	R/W
M8381	User Communication Receive Instruction Cancel Flag (Port 22)		Cleared	Cleared	R/W
M8382	User Communication Receive Instruction Cancel Flag (Port 23)		Cleared	Cleared	R/W
M8383	User Communication Receive Instruction Cancel Flag (Port 24)		Cleared	Cleared	R/W

Device Address	Description		When Stopped	Power OFF	R/W
M8384	User Communication Receive Instruction Cancel Flag (Port 25)		Cleared	Cleared	R/W
M8385	User Communication Receive Instruction Cancel Flag (Port 26)		Cleared	Cleared	R/W
M8386	User Communication Receive Instruction Cancel Flag (Port 27)		Cleared	Cleared	R/W
M8387	User Communication Receive Instruction Cancel Flag (Port 28)		Cleared	Cleared	R/W
M8390	User Communication Receive Instruction Cancel Flag (Port 29)		Cleared	Cleared	R/W
M8391	User Communication Receive Instruction Cancel Flag (Port 30)		Cleared	Cleared	R/W
M8392	User Communication Receive Instruction Cancel Flag (Port 31)		Cleared	Cleared	R/W
M8393	User Communication Receive Instruction Cancel Flag (Port 32)		Cleared	Cleared	R/W
M8394	User Communication Receive Instruction Cancel Flag (Port 33)		Cleared	Cleared	R/W
M8395 to M8447	**— Reserved —**		—	—	—
M8450	BACnet Communication Bit		Operating	Cleared	R/W
M8451 to M8457	**— Reserved —**		—	—	—
M8460	EtherNet/IP Communication Bit		Cleared	Cleared	R/W
M8461 to M8597	**— Reserved —**		—	—	—
M8600	High-speed Counter (Group 3/I3)	Reset Status	Maintained	Cleared	R
M8601		Underflow	Maintained	Cleared	R
M8602		Count Direction Flag	Maintained	Cleared	R
M8603 to M8997	**— Reserved —**		—	—	—

Supplementary Descriptions of the Special Internal Relays

■ M8000: Start Control

M8000 controls the run/stop status of the FC6A Series MICROSmart. The CPU is set to run when M8000 is turned on, and the CPU is set to off when M8000 is turned off. See Chapter 1 "Start/Stop Operation" in the "FC6A Series MICROSmart Ladder Programming Manual". However, the function switch, stop input, and reset input have precedence over start control. M8000 maintains its status when the CPU is powered down. When data to be maintained during a power failure disappears after the CPU has been off for a period longer than the battery backup duration, the CPU restarts operation as selected in Configuration > Run/Stop Control > Run/Stop Selection at Keep Data Error. For details, see Chapter 5 "Run/Stop Selection at Keep Data Error" in the "FC6A Series MICROSmart User’s Manual".

■ M8001: 1-s Clock Reset

While M8001 is on, M8121 (1-s clock) is always off.

■ M8002: All Outputs OFF

While M8002 is on, all outputs are off. The self-holding circuit created in the ladder program is also off.

■ M8003: Carry/Borrow

When a carry (Cy) or borrow (Bw) results from executing an addition or subtraction instruction, M8003 is turned on. For details, see Chapter 3 "Carry and Borrow" in the "FC6A Series MICROSmart Ladder Programming Manual".

■ M8004: User Program Execution Error

When an error occurs while executing a user program, M8004 is turned on. For details on user program execution errors, see Chapter 13 "User Program Execution Error" in the "FC6A Series MICROSmart User’s Manual".

■ M8005: Communication Error

When an error occurs during data link communication, M8005 is turned on. The state is retained even when the error is cleared.

■ M8006: Communication Prohibited Flag (When Data Link Master)

During data link communication, communication is stopped while M8006 is on.

■ M8007: Initialization Flag (When Data Link Master)/Stop Communication Flag (When Data Link Slave)

When data link master: When this flag is turned on in the run status, the data link is initialized just once to check the connection status. Use this when the slave configured in the data link is powered at a timing slower than the master.

When data link slave : This flag is turned on when communication from the master is interrupted for 10 s or longer. This flag is turned off when communication can be normally received.

■ **M8010: Status LED Operation**
While M8010 is on, the status LED [STAT] is turned on. While off, the status LED [STAT] is turned off.

■ **M8013: Calendar/Clock Data Write/Adjust Error Flag**
When the clock writing or clock adjustment processing could not be executed normally, M8013 is turned on. It is turned off when the processing completes normally.

■ **M8014: Calendar/Clock Data Read Error Flag**
When an error occurs while calendar/clock data is read from the internal clock to the special data registers (D8008 to D8021), M8014 is turned on. It is turned off when reading completes normally.

■ **M8016: Calendar Data Write Flag**
When M8016 is turned on after writing data to the calendar data (write-only) special data registers (D8015 to D8018), the calendar data (year, month, day, day of the week) is written to the internal clock.

■ **M8017: Clock Data Write Flag**
When M8017 is turned on after writing data to the clock data (write-only) special data registers (D8019 to D8021), the clock data (hours, minutes, seconds) is written to the internal clock.

■ **M8020: Calendar/Clock Data Write Flag**
When M8020 is turned on after writing data to the calendar/clock data (write-only) special data registers (D8015 to D8021), the calendar data (year, month, day, day of the week) and the clock data (hours, minutes, seconds) is written to the internal clock.

■ **M8021: Clock Data Adjust Flag**
When M8021 is turned on, the clock is adjusted with respect to seconds.

- When M8021 is turned on, if the seconds are between 0 and 29, the seconds will be set to 0 and the minutes remain the same.
- When M8021 is turned on, if the seconds are between 30 and 59, the seconds will be set to 0 and 1 will be added to the minutes.

■ **M8022: User Communication Receive Instruction Cancel Flag (Port 1)**
While M8022 is on, user communication (receive instruction) executing on Port 1 is canceled.

■ **M8023: User Communication Receive Instruction Cancel Flag (Port 2)**
While M8023 is on, user communication (receive instruction) executing on Port 2 is canceled.

■ **M8024: BMOV/WSFT Executing Flag**
While the WSFT (word shift) instruction or the BMOV (block move) instruction is executing, M8024 is turned on. When completed, M8024 is turned off.

■ **M8025: Maintain Outputs While Stopped**
When the FC6A Series MICROSmart is stopped while running with M8025 turned on, the outputs maintain their runtime status. When set to run again, M8025 is automatically turned off.

■ **M8026: User Communication Receive Instruction Cancel Flag (Port 3)**
While M8026 is on, user communication (receive instruction) executing on Port 3 is canceled.

■ **M8033, M8145 to M8147, M8170, M8176, M8365 to M8394: User Communication Receive Instruction Cancel Flag (Port 4 to 33)**
While these flags are on, user communication (receive instruction) being executed on the corresponding port is canceled.
M8033 = User Communication Receive Instruction Cancel Flag (Port 4)
M8145 = User Communication Receive Instruction Cancel Flag (Port 5)
M8146 = User Communication Receive Instruction Cancel Flag (Port 6)
M8147 = User Communication Receive Instruction Cancel Flag (Port 7)
M8170 = User Communication Receive Instruction Cancel Flag (Port 8)
M8176 = User Communication Receive Instruction Cancel Flag (Port 9)
M8365 = User Communication Receive Instruction Cancel Flag (Port 10)
M8366 = User Communication Receive Instruction Cancel Flag (Port 11)
M8367 = User Communication Receive Instruction Cancel Flag (Port 12)
M8370 = User Communication Receive Instruction Cancel Flag (Port 13)
M8371 = User Communication Receive Instruction Cancel Flag (Port 14)
M8372 = User Communication Receive Instruction Cancel Flag (Port 15)
M8373 = User Communication Receive Instruction Cancel Flag (Port 16)
M8374 = User Communication Receive Instruction Cancel Flag (Port 17)

M8375 = User Communication Receive Instruction Cancel Flag (Port 18)
M8376 = User Communication Receive Instruction Cancel Flag (Port 19)
M8377 = User Communication Receive Instruction Cancel Flag (Port 20)
M8380 = User Communication Receive Instruction Cancel Flag (Port 21)
M8381 = User Communication Receive Instruction Cancel Flag (Port 22)
M8382 = User Communication Receive Instruction Cancel Flag (Port 23)
M8383 = User Communication Receive Instruction Cancel Flag (Port 24)
M8384 = User Communication Receive Instruction Cancel Flag (Port 25)
M8385 = User Communication Receive Instruction Cancel Flag (Port 26)
M8386 = User Communication Receive Instruction Cancel Flag (Port 27)
M8387 = User Communication Receive Instruction Cancel Flag (Port 28)
M8390 = User Communication Receive Instruction Cancel Flag (Port 29)
M8391 = User Communication Receive Instruction Cancel Flag (Port 30)
M8392 = User Communication Receive Instruction Cancel Flag (Port 31)
M8393 = User Communication Receive Instruction Cancel Flag (Port 32)
M8394 = User Communication Receive Instruction Cancel Flag (Port 33)

■ **M8027 to M8032, M8034 to M8036, M8040 to M8046, M8051 to M8055, M8057 to M8063, M8130, M8131, M8133 to M8136, M8161 to M8167, M8600 to M8602: Special Internal Relays for High-speed Counter**

Special internal relays used for the high-speed counter. For details, see Chapter 5 "High-Speed Counter" in the "FC6A Series MICROSmart User’s Manual".
M8027 to M8032, M8130, M8131, M8161, M8162 = High-speed counter (group 1/I0)
M8034 to M8036, M8133, M8165, M8600 to M8602 = High-speed counter (group 3/I3)
M8040 to M8042, M8134, M8166 = High-speed counter (group 4/I4)
M8043 to M8046, M8135, M8136, M8163, M8164 = High-speed counter (group 5/I6)
M8051 to M8055 = High-speed counter (group 2/I1)
M8057 to M8063 = High-speed counter (group 6/I7)

■ **M8070: SD Memory Card Mount Status**

When an SD memory card is inserted in the FC6A Series MICROSmart and it has been recognized and can be used, M8070 is turned on. M8070 is turned off if no SD memory card has been inserted or if it is not recognized.

■ **M8071: Accessing SD Memory Card**

M8071 is turned on while the SD memory card is being accessed. It is turned off when access has finished.

■ **M8072: Unmount SD Memory Card**

When M8072 is turned on, access to the SD memory card is stopped. To make an SD memory card accessible that has had access to it stopped, insert the card once again.

■ **M8073: Function Switch Status**

This relay indicates the status of the function switch on the front of the CPU module.
M8073 is on when the function switch is 1. M8073 is off when the function switch is 0.

■ **M8074: Battery Voltage Measurement Flag**

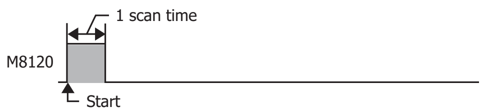
This relay indicates the battery voltage measurement status of the backup battery.
When M8074 is turned on, the battery voltage starts being measured, and it is turned off when the measurement has finished.

■ **M8080 to M8117: Data Link Communication Completed Relay**

Special internal relays used for data link communication. For details, see "Data Link Communication" on page 7-1.

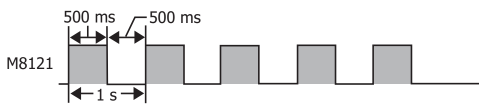
■ **M8120: Initialize Pulse**

When operation (RUN) starts, M8120 is turned on for a period of one scan.



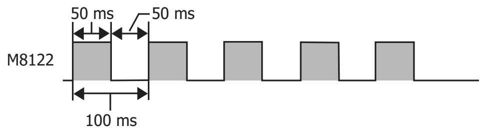
■ **M8121: 1-s Clock**

While M8001 is off, M8121 generates clock pulses in a 1 s cycle, with a duty ratio of 1:1 (500 ms on and 500 ms off).



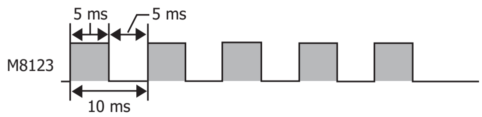
■ M8122: 100-ms Clock

M8122 generates clock pulses in a 100 ms cycle, with a duty ratio of 1:1 (50 ms on and 50 ms off).



■ M8123: 10-ms Clock

M8123 generates clock pulses in a 10 ms cycle, with a duty ratio of 1:1 (5 ms on and 5 ms off).



■ M8124: Timer/Counter Preset Value Changed

When timer or counter preset values are changed, M8124 is turned on. When a user program is transferred or when the changed data is cleared, M8124 is turned off.

■ M8125: In-operation Output

M8125 is on during the run status.

■ M8126: Scan ON After Run-Time Download Completes

When the user program is changed during the run status (run-time download), after the download completes, M8126 is turned on for one scan only when the user program starts operation.

■ M8137 to M8143, M8167: Interrupt Input Status

These relays are turned on when the corresponding user interrupt is allowed. These relays are turned off when the user interrupt is prohibited.

M8137 = Interrupt input I0 status
M8140 = Interrupt input I1 status
M8141 = Interrupt input I3 status
M8142 = Interrupt input I4 status
M8143 = Interrupt input I6 status
M8167 = Interrupt input I7 status

■ M8144: Timer Interrupt Status

When the timer interrupt is allowed, M8144 is turned on. When the timer interrupt is prohibited, M8144 is turned off.

■ M8150 to M8152: Comparison Result

M8150 to M8152 turn on according to the comparison results of the CMP= (Compare(=)) instruction and the ICMP>= (Interval Compare) instruction.

For details, see Chapter 6 "Special Internal Relays M8150, M8151, and M8152 in ICMP>=" in the "FC6A Series MICROSmart Ladder Programming Manual".

■ M8153 to M8160: Catch Input ON/OFF Status

When the rising or falling input edge of the input contact specified as the catch input is detected during a scan, the input contact status is captured. The detectable edges are once per scan.

M8153 = Group 1/I0 status
M8154 = Group 2/I1 status
M8155 = Group 3/I3 status
M8156 = Group 4/I4 status
M8157 = Group 5/I6 status
M8160 = Group 6/I7 status

■ M8172 to M8175: Transistor Source Output Overcurrent Detection

When overcurrent output occurs in a transistor protection source output on the CPU module, a special internal relay (M8172 to M8175) is turned on. The following special internal relays have been allocated with four outputs set as one group. If overcurrent output occurs in any of the special internal relays, it is turned on.

Even if the overcurrent output has been cleared, these special internal relays are not reset to off. To reset a special internal relay to off, create programming to do so in the ladder program.

M8172 = Group 1 (Q0 to Q3) status
M8173 = Group 2 (Q4 to Q7) status
M8174 = Group 3 (Q10 to Q13) status
M8175 = Group 4 (Q14 to Q17) status

■ M8184: Change HMI Module Network Settings Trigger

When M8184 is turned on, the values stored in D8437 to D8456 are set as the HMI module IP address.
The IP address is not set just by changing the values of D8437 to D8456. For details on changing the HMI module network settings, see "Network settings by HMI module special data registers" on page 3-6.

■ M8185: In Daylight Saving Time Period

When the daylight saving time function is enabled, M8185 is on during the daylight saving time period. When outside of the daylight saving time period, M8185 is off.

When the daylight saving time function is disabled, M8185 is off.

■ M8186: Ethernet Port 1 Executing Auto Ping

M8186 is on when the auto ping on Ethernet port 1 is operating. M8186 is off when auto ping is stopped. For details on auto ping, see "Auto Ping Function" on page 3-22.

■ M8187: Ethernet Port 1 Auto Ping Stop Flag

While M8187 is on, the auto ping on Ethernet port 1 is stopped. While M8187 is off, auto ping is executed. At that time, auto ping is executed from the smallest remote host number specified in the remote host list, regardless of the previous end status.

■ M8190: Change CPU Module Ethernet Port 1 Network Settings Trigger

When M8190 is turned on, the values stored in D8303 to D8323 are set as the IP Settings/DNS Settings of Ethernet port 1 on the CPU module.
The IP address is not set just by changing the values of D8303 to D8323. For details on changing the IP Settings/DNS Settings of Ethernet port 1 on the CPU module, see "Network settings by special data registers" on page 3-4.

■ M8191: SNTP Acquisition Flag

When M8191 is turned on, the time information is acquired from the SNTP server.

■ M8192 to M8197: Interrupt Input Edge

These relays turn on when an interrupt occurs with the rising edge of an interrupt input. They turn off when an interrupt occurs with the falling edge of an interrupt input.
M8192 = Interrupt input I0 edge
M8193 = Interrupt input I3 edge
M8194 = Interrupt input I4 edge
M8195 = Interrupt input I6 edge
M8196 = Interrupt input I7 edge
M8197 = Interrupt input I1 edge

■ M8200 to M8207, M8334 to M8343: User Communication Receive Instruction Cancel Flag

When M8200 to M8207 or M8334 to M8343 are turned on, the user communication receive instruction being executed is stopped.
M8200 = User communication receive instruction being executed on client connection 1
M8201 = User communication receive instruction being executed on client connection 2
M8202 = User communication receive instruction being executed on client connection 3
M8203 = User communication receive instruction being executed on client connection 4
M8204 = User communication receive instruction being executed on client connection 5
M8205 = User communication receive instruction being executed on client connection 6
M8206 = User communication receive instruction being executed on client connection 7
M8207 = User communication receive instruction being executed on client connection 8
M8334 = User communication receive instruction being executed on client connection 9
M8335 = User communication receive instruction being executed on client connection 10
M8336 = User communication receive instruction being executed on client connection 11
M8337 = User communication receive instruction being executed on client connection 12
M8340 = User communication receive instruction being executed on client connection 13
M8341 = User communication receive instruction being executed on client connection 14
M8342 = User communication receive instruction being executed on client connection 15
M8343 = User communication receive instruction being executed on client connection 16

■ M8211: HMI Module Send E-mail Server Settings Initialization

When M8211 is turned on, the settings for the send E-mail server on the HMI-Ethernet port are initialized.

- **M8212 to M8221, M8345 to M8354: Connection Status**

  While connected to a network device via the maintenance communication server, user communication server/client, or Modbus TCP server/client, the connection status is turned on. While not connected to a network device, the connection status is turned off.
  M8212 = Connection 1
  M8213 = Connection 2
  M8214 = Connection 3
  M8215 = Connection 4
  M8216 = Connection 5
  M8217 = Connection 6
  M8220 = Connection 7
  M8221 = Connection 8
  M8345 = Connection 9
  M8346 = Connection 10
  M8347 = Connection 11
  M8350 = Connection 12
  M8351 = Connection 13
  M8352 = Connection 14
  M8353 = Connection 15
  M8354 = Connection 16

- **M8222 to M8231, M8355 to M8364: Disconnect User Communication Connection**

  When connected to a remote host via user communication, the corresponding connection is disconnected when M8222 to M8231 is turned on.
  M8222 = Connection 1
  M8223 = Connection 2
  M8224 = Connection 3
  M8225 = Connection 4
  M8226 = Connection 5
  M8227 = Connection 6
  M8230 = Connection 7
  M8231 = Connection 8
  M8355 = Connection 9
  M8356 = Connection 10
  M8357 = Connection 11
  M8360 = Connection 12
  M8361 = Connection 13
  M8362 = Connection 14
  M8363 = Connection 15
  M8364 = Connection 16
  These relays are enabled only when a user communication client is used.

- **M8232: HMI Module Connection Information Reference Connection Status**

  M8232 is turned on when there is a connection with the connection number specified by D8429. M8232 is turned off when there is no connection.

- **M8250: Download from SD Memory Card Execution Flag**

  When M8250 is turned on, a ZLD file is downloaded from the SD memory card. The file that will be downloaded is the ZLD file specified in the autoexec.ini file.

- **M8251: Upload to SD Memory Card Execution Flag**

  When M8251 is turned on, a ZLD file is uploaded to the SD memory card.
  A ZLD file is created with the file name specified in the autoexec.ini file.

- **M8252: Executing SD Memory Card Download**

  M8252 is turned on when starting execution of the download from the SD memory card, and when the download has completed, it is turned off.

- **M8253: Executing SD Memory Card Upload**

  M8253 is turned on when starting execution of the upload to the SD memory card, and when the upload has completed, it is turned off.

■ **M8254: SD Memory Card Download/Upload Execution Completed Output**

M8254 is turned off when starting execution of the download from the SD memory card or the upload to the SD memory card, and when the download or upload has completed, it is turned on.

■ **M8255: SD Memory Card Download/Upload Execution Error Output**

M8255 is updated when execution of the download from the SD memory card or the upload to the SD memory card has completed. M8255 is turned on when D8255 (Download/Upload Execution Status) is a value other than 0.

■ **M8260: Write Recipe Execution Flag**

When M8260 is turned on, write recipe is executed.

■ **M8261: Read Recipe Execution Flag**

When M8261 is turned on, read recipe is executed.

■ **M8262: Executing Write Recipe**

M8262 is turned on when the write recipe processing starts, and it is turned off when the processing has completed. M8262 also is turned off when read recipe processing starts.

■ **M8263: Executing Read Recipe**

M8263 is turned on when the read recipe processing starts, and it is turned off when the processing has completed. M8263 also is turned off when write recipe processing starts.

■ **M8264: Recipe Execution Completed Output**

M8264 is turned off when starting to read or write a recipe, and it is turned on when reading or writing a recipe has completed.

■ **M8265: Recipe Execution Error Output**

M8265 is turned on when recipe execution has completed and D8264 (Recipe Execution Status) is a value other than 0. For details on recipes, see Chapter 11 "Recipe Function" in the "FC6A Series MICROSmart User's Manual".

■ **M8300 to M8304: J1939 Communication**

Special data registers used in J1939 communication. For details, see "J1939 Communication" on page 8-1.

■ **M8311 to M8314: Key Input Status**

While the ESC button and direction buttons on the HMI module are simultaneously pressed, the corresponding special internal relays are turned on. When the keys are not pressed, the relays are turned off.

M8311 = ESC key + Up (▲) key

M8312 = ESC key + Down (▼) key

M8313 = ESC key + Left (◀) key

M8314 = ESC key + Right (▶) key

■ **M8320: Initialize Expansion Interface Remote Master/Slave**

When M8320 is turned on, expansion interface remote master and slave modules along with the expansion modules that are connected to the expansion interface remote slave modules are initialized. One the initialization is finished, M8320 is automatically turned off and the I/O refresh at the expansion interface remote slave modules will resume.

■ **M8331: Ethernet Port 2 Executing Auto Ping**

M8331 is on when the auto ping on Ethernet port 2 is operating. M8331 is off when auto ping is stopped. For details on auto ping, see "Auto Ping Function" on page 3-22.

■ **M8332: Ethernet Port 2 Auto Ping Stop Flag**

While M8332 is on, the auto ping on Ethernet port 2 is stopped. While M8332 is off, auto ping is executed. At that time, auto ping is executed from the smallest remote host number specified in the remote host list, regardless of the previous end status.

■ **M8333: Change CPU Module Ethernet Port 2 Network Settings Trigger**

When M8333 is turned on, the values stored in D8630 to D8650 are set as the IP Settings/DNS Settings of Ethernet port 2 on the CPU module. The IP address is not set just by changing the values of D8630 to D8650. For details on changing the IP Settings/DNS Settings of Ethernet port 2 on the CPU module, see "Network settings by special data registers" on page 3-4.

■ **M8344: Ethernet Port 1 Send E-mail Server Settings Initialization**

When M8344 is turned on, the settings for the send E-mail server on the Ethernet port 1 are initialized.

■ **M8450: BACnet Communication Bit**

While M8450 is on, BACnet communication will be performed.

■ **M8460: EtherNet/IP Communication Bit**

This special internal relay permits or prohibits EtherNet/IP communication.

OFF: Prohibit EtherNet/IP communication

ON: Permit EtherNet/IP communication

Special Data Register

Special Data Register Device Addresses

Warning	Do not write to data in the area marked as reserved in the special data registers list. Otherwise the system may not operate correctly.

Note: R/W is an abbreviation for read/write.
The notation for the R/W field is as follows.
R/W: The device can be both read from and written to
R: Read-only
W: Write-only

Device Address	Description		Update Timing	R/W
D8000	Quantity of Inputs		When I/O initialized	R
D8001	Quantity of Outputs		When I/O initialized	R
D8002	CPU Module Type Information		Power-up	R
D8003 D8004	**— Reserved —**		—	-
D8005	General Error Code		When error occurred	R/W
D8006	User Program Execution Error Code		When error occurred	R
D8007	**— Reserved —**		—	-
D8008	Calendar/Clock Current Data (Read only)	Year	Every 500 ms	R
D8009		Month	Every 500 ms	R
D8010		Day	Every 500 ms	R
D8011		Day of the Week	Every 500 ms	R
D8012		Hour	Every 500 ms	R
D8013		Minute	Every 500 ms	R
D8014		Second	Every 500 ms	R
D8015	Calendar/Clock New Data (Write only)	Year	—	W
D8016		Month	—	W
D8017		Day	—	W
D8018		Day of the Week	—	W
D8019		Hour	—	W
D8020		Minute	—	W
D8021		Second	—	W
D8022	Scan Time Data	Constant Scan Time Preset Value (1 to 1,000 ms)	—	W
D8023		Scan Time Current Value (ms)	Every scan	R
D8024		Scan Time Maximum Value (ms)	At occurrence	R
D8025		Scan Time Minimum Value (ms)	At occurrence	R
D8026	Communication Mode Information (Port 1 to Port 3)		Every scan	R
D8027 D8028	**— Reserved —**		—	-
D8029	System Software Version		Power-up	R
D8030	Communication Cartridge Information		Power-up	R
D8031	Optional Device Connection Information		Power-up	R
D8032	Interrupt Input Jump Destination Label No. (I1)		—	R/W
D8033	Interrupt Input Jump Destination Label No. (I3)		—	R/W
D8034	Interrupt Input Jump Destination Label No. (I4)		—	R/W
D8035	Interrupt Input Jump Destination Label No. (I6)		—	R/W
D8036	Timer Interrupt Jump Destination Label No.		—	R/W
D8037	Number of Connected I/O Modules		When I/O initialized	R
D8038 D8039	**— Reserved —**		—	—
D8040	Slave Number (Port 4)		—	R/W
D8041	Slave Number (Port 5)		—	R/W
D8042	Slave Number (Port 6)		—	R/W
D8043	Slave Number (Port 7)		—	R/W

Device Address	Description	Update Timing	R/W
D8044	Slave Number (Port 8)	—	R/W
D8045	Slave Number (Port 9)	—	R/W
D8046 to D8051	**— Reserved —**	—	—
D8052	J1939 Communication Error Code	Every scan	R/W
D8053 to D8055	**— Reserved —**	—	—
D8056	Battery Voltage	—	R
D8057	Analog Volume (AI0)	Every scan	R
D8058	Built-in Analog Input (AI1)	Every scan	R
D8059	Analog Input Status AI0	Every scan	R
D8060	Analog Input Status AI1	Every scan	R
D8061 to D8066	**— Reserved —**	—	—
D8067	Backlight ON Time	—	R/W
D8068	**— Reserved —**	—	—
D8069	Slave 1 Communication Status/Error (When Data Link Master Mode) Slave Communication Status/Error (When Data Link Slave Mode)	When error occurred	R
D8070	Slave 2 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8071	Slave 3 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8072	Slave 4 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8073	Slave 5 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8074	Slave 6 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8075	Slave 7 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8076	Slave 8 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8077	Slave 9 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8078	Slave 10 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8079	Slave 11 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8080	Slave 12 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8081	Slave 13 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8082	Slave 14 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8083	Slave 15 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8084	Slave 16 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8085	Slave 17 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8086	Slave 18 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8087	Slave 19 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8088	Slave 20 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8089	Slave 21 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8090	Slave 22 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8091	Slave 23 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8092	Slave 24 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8093	Slave 25 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8094	Slave 26 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8095	Slave 27 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8096	Slave 28 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8097	Slave 29 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8098	Slave 30 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8099	Slave 31 Communication Status/Error (When Data Link Master Mode)	When error occurred	R
D8100	Slave Number (Port 1)	—	R/W
D8101	**— Reserved —**	—	—
D8102	Slave Number (Port 2)	—	R/W
D8103	Slave Number (Port 3)	—	R/W
D8104	Control Signal Status (Port 1 to 5)	Every scan	R
D8105	RS232C DSR Input Control Signal Option (Port 1 to 5)	Every scan	R/W
D8106	RS232C DTR Output Control Signal Option (Port 1 to 5)	Every scan	R/W

Device Address	Description			Update Timing	R/W
D8107 to D8119	**— Reserved —**			—	—
D8120	HMI Module Information		Type ID/Status	—	R
D8121			System Software Version	—	R
D8122	Cartridge Slot 1 Information		Type ID/Status	—	R
D8123			System Software Version	—	R
D8124	Cartridge Slot 2 Information		Type ID/Status	—	R
D8125			System Software Version	—	R
D8126	Cartridge Slot 3 Information		Type ID/Status	—	R
D8127			System Software Version	—	R
D8128 to D8169	**— Reserved —**			—	—
D8170	Analog I/O Cartridge I/O AI2/AQ2			Every scan	R
D8171	Analog I/O Cartridge I/O AI3/AQ3			Every scan	R
D8172	Analog I/O Cartridge Status AI2/AQ2			Every scan	R
D8173	Analog I/O Cartridge Status AI3/AQ3			Every scan	R
D8174	Analog I/O Cartridge I/O AI4/AQ4			Every scan	R
D8175	Analog I/O Cartridge I/O AI5/AQ5			Every scan	R
D8176	Analog I/O Cartridge Status AI4/AQ4			Every scan	R
D8177	Analog I/O Cartridge Status AI5/AQ5			Every scan	R
D8178	Analog I/O Cartridge I/O AI6/AQ6			Every scan	R
D8179	Analog I/O Cartridge I/O AI7/AQ7			Every scan	R
D8180	Analog I/O Cartridge Status AI6/AQ6			Every scan	R
D8181	Analog I/O Cartridge Status AI7/AQ7			Every scan	R
D8182 to D8191	**— Reserved —**			—	—
D8192	High-speed Counter (Group 2/I1)	High Word	Current Value/Frequency Measurement (I1) Current Value	Every scan	R
D8193		Low Word		Every scan	R
D8194		High Word	Preset Value	—	R/W
D8195		Low Word		—	R/W
D8196		High Word	Reset Value	—	R/W
D8197		Low Word		—	R/W
D8198	High-speed Counter (Group 6/I7)	High Word	Current Value/Frequency Measurement (I7) Current Value	Every scan	R
D8199		Low Word		Every scan	R
D8200		High Word	Preset Value	—	R/W
D8201		Low Word		—	R/W
D8202		High Word	Reset Value	—	R/W
D8203		Low Word		—	R/W
D8204	Control Signal Status (Port 6 to 9)			Every scan	R
D8205	RS232C DSR Input Control Signal Option (Port 6 to 9)			Every scan	R/W
D8206	RS232C DTR Output Control Signal Option (Port 6 to 9)			Every scan	R/W
D8207 to D8209	**— Reserved —**			—	—
D8210	High-speed Counter (Group 1/I0)	High Word	Current Value/Frequency Measurement (I0) Current Value	Every scan	R
D8211		Low Word		Every scan	R
D8212		High Word	Preset Value	—	R/W
D8213		Low Word		—	R/W
D8214	Interrupt Input Jump Destination Label No. (I7)			—	R/W
D8215	Interrupt Input Jump Destination Label No. (I0)			—	R/W
D8216	High-speed Counter (Group 1/I0)	High Word	Reset Value	—	R/W
D8217		Low Word		—	R/W
D8218	High-speed Counter (Group 3/I3)	High Word	Current Value/Frequency Measurement (I3) Current Value	Every scan	R
D8219		Low Word		Every scan	R
D8220		High Word	Preset Value	—	R/W
D8221		Low Word		—	R/W

Device Address	Description			Update Timing	R/W
D8222	High-speed Counter (Group 4/I4)	High Word	Current Value/Frequency Measurement (I4) Current Value	Every scan	R
D8223		Low Word		Every scan	R
D8224		High Word	Preset Value	—	R/W
D8225		Low Word		—	R/W
D8226	High-speed Counter (Group 5/I6)	High Word	Current Value/Frequency Measurement (I6) Current Value	Every scan	R
D8227		Low Word		Every scan	R
D8228		High Word	Preset Value	—	R/W
D8229		Low Word		—	R/W
D8230 D8231	**— Reserved —**			**—**	**—**
D8232	High-speed Counter (Group 5/I6)	High Word	Reset Value	—	R/W
D8233		Low Word		—	R/W
D8234	High-speed Counter (Group 3/I3)	High Word	Reset Value	—	R/W
D8235		Low Word		—	R/W
D8236	High-speed Counter (Group 4/I4)	High Word	Reset Value	—	R/W
D8237		Low Word		—	R/W
D8238	**— Reserved —**			**—**	**—**
D8239	Absolute Position Control Status			Every scan	R
D8240	Absolute Position Counter 1	High Word	Absolute Position	Every scan	R
D8241		Low Word		Every scan	R
D8242	Absolute Position Counter 2	High Word	Absolute Position	Every scan	R
D8243		Low Word		Every scan	R
D8244	Absolute Position Counter 3	High Word	Absolute Position	Every scan	R
D8245		Low Word		Every scan	R
D8246	Absolute Position Counter 4	High Word	Absolute Position	Every scan	R
D8247		Low Word		Every scan	R
D8248 D8249	**— Reserved —**			**—**	**—**
D8250	Read SD Memory Card Capacity			Every scan	R
D8251	Read SD Memory Card Free Capacity			Every scan	R
D8252 D8253	**— Reserved —**			**—**	**—**
D8254	SD Memory Card Download/Upload Execution Information			When processing has completed	R
D8255	SD Memory Card Download/Upload Execution Status			When processing has completed	R
D8256 to D8259	**— Reserved —**			**—**	**—**
D8260	Recipe Block Number			—	R/W
D8261	Recipe Execution Block Number			When recipe execution has completed	R
D8262	Recipe Execution Channel No.			When recipe execution has completed	R
D8263	Recipe Execution Operation			When recipe execution has completed	R
D8264	Recipe Execution Status			When recipe execution has completed	R
D8265	Recipe Execution Error Information			When recipe execution has completed	R
D8266 D8267	**— Reserved —**			**—**	**—**
D8268	Remote Host Number of Connection 1 (1 to 255)			—	R/W
D8269	Remote Host Number of Connection 2 (1 to 255)			—	R/W
D8270	Remote Host Number of Connection 3 (1 to 255)			—	R/W

Device Address	Description		Update Timing	R/W
D8271	Remote Host Number of Connection 4 (1 to 255)		—	R/W
D8272	Remote Host Number of Connection 5 (1 to 255)		—	R/W
D8273	Remote Host Number of Connection 6 (1 to 255)		—	R/W
D8274	Remote Host Number of Connection 7 (1 to 255)		—	R/W
D8275	Remote Host Number of Connection 8 (1 to 255)		—	R/W
D8276 D8277	**— Reserved —**		**—**	**—**
D8278	Communication Mode Information (Client Connection)	Connection 1 to 4	—	R
D8279		Connection 5 to 8	—	R
D8280 to D8283	**— Reserved —**		**—**	**—**
D8284	Communication Mode Information (HMI Connection)	HMI Connection 1 to 4	—	R
D8285		HMI Connection 5 to 8	—	R
D8286 to D8302	**— Reserved —**		**—**	**—**
D8303	CPU Module Ethernet Port 1 IP Settings/DNS Settings Switching		—	R/W
D8304	CPU Module Ethernet Port 1 IP Address (Write-only)		—	W
D8305			—	W
D8306			—	W
D8307			—	W
D8308	CPU Module Ethernet Port 1 Subnet Mask (Write-only)		—	W
D8309			—	W
D8310			—	W
D8311			—	W
D8312	CPU Module Ethernet Port 1 Default Gateway (Write-only)		—	W
D8313			—	W
D8314			—	W
D8315			—	W
D8316	CPU Module Ethernet Port 1 Preferred DNS Server (Write-only)		—	W
D8317			—	W
D8318			—	W
D8319			—	W
D8320	CPU Module Ethernet Port 1 Alternate DNS Server (Write-only)		—	W
D8321			—	W
D8322			—	W
D8323			—	W
D8324	CPU Module Ethernet Port 1 MAC Address (Current Value Read-only)		Every 1 s	R
D8325			Every 1 s	R
D8326			Every 1 s	R
D8327			Every 1 s	R
D8328			Every 1 s	R
D8329			Every 1 s	R
D8330	CPU Module Ethernet Port 1 IP Address (Current Value Read-only)		Every 1 s	R
D8331			Every 1 s	R
D8332			Every 1 s	R
D8333			Every 1 s	R
D8334	CPU Module Ethernet Port 1 Subnet Mask (Current Value Read-only)		Every 1 s	R
D8335			Every 1 s	R
D8336			Every 1 s	R
D8337			Every 1 s	R
D8338	CPU Module Ethernet Port 1 Default Gateway (Current Value Read-only)		Every 1 s	R
D8339			Every 1 s	R
D8340			Every 1 s	R
D8341			Every 1 s	R

Device Address	Description	Update Timing	R/W
D8342	CPU Module Ethernet Port 1 Preferred DNS Server (Current Value Read-only)	Every 1 s	R
D8343		Every 1 s	R
D8344		Every 1 s	R
D8345		Every 1 s	R
D8346	CPU Module Ethernet Port 1 Alternate DNS Server (Current Value Read-only)	Every 1 s	R
D8347		Every 1 s	R
D8348		Every 1 s	R
D8349		Every 1 s	R
D8350	Connection 1 Connected IP Address	Every 1 s	R
D8351		Every 1 s	R
D8352		Every 1 s	R
D8353		Every 1 s	R
D8354	Connection 2 Connected IP Address	Every 1 s	R
D8355		Every 1 s	R
D8356		Every 1 s	R
D8357		Every 1 s	R
D8358	Connection 3 Connected IP Address	Every 1 s	R
D8359		Every 1 s	R
D8360		Every 1 s	R
D8361		Every 1 s	R
D8362	Connection 4 Connected IP Address	Every 1 s	R
D8363		Every 1 s	R
D8364		Every 1 s	R
D8365		Every 1 s	R
D8366	Connection 5 Connected IP Address	Every 1 s	R
D8367		Every 1 s	R
D8368		Every 1 s	R
D8369		Every 1 s	R
D8370	Connection 6 Connected IP Address	Every 1 s	R
D8371		Every 1 s	R
D8372		Every 1 s	R
D8373		Every 1 s	R
D8374	Connection 7 Connected IP Address	Every 1 s	R
D8375		Every 1 s	R
D8376		Every 1 s	R
D8377		Every 1 s	R
D8378	Connection 8 Connected IP Address	Every 1 s	R
D8379		Every 1 s	R
D8380		Every 1 s	R
D8381		Every 1 s	R
D8382	HMI Module MAC Address (Current Value Read-only)	Every 1 s	R
D8383		Every 1 s	R
D8384		Every 1 s	R
D8385		Every 1 s	R
D8386		Every 1 s	R
D8387		Every 1 s	R
D8388	HMI Module IP Address (Current Value Read-only)	Every 1 s	R
D8389		Every 1 s	R
D8390		Every 1 s	R
D8391		Every 1 s	R
D8392	HMI Module Subnet Mask (Current Value Read-only)	Every 1 s	R
D8393		Every 1 s	R
D8394		Every 1 s	R
D8395		Every 1 s	R

Device Address	Description		Update Timing	R/W
D8396	HMI Module Default Gateway (Current Value Read-only)		Every 1 s	R
D8397			Every 1 s	R
D8398			Every 1 s	R
D8399			Every 1 s	R
D8400	HMI Module Preferred DNS Server (Current Value Read-only)		Every 1 s	R
D8401			Every 1 s	R
D8402			Every 1 s	R
D8403			Every 1 s	R
D8404	HMI Module Alternate DNS Server (Current Value Read-only)		Every 1 s	R
D8405			Every 1 s	R
D8406			Every 1 s	R
D8407			Every 1 s	R
D8408 to D8412	**— Reserved —**		—	—
D8413	Time Zone Offset		—	R/W
D8414	SNTP Operation Status		—	R
D8415	SNTP Access Elapsed Time		—	R
D8416 to D8428	**— Reserved —**		—	—
D8429	HMI Module Connection Information Reference	Connection No.	—	R/W
D8430		Connected IP Address	Every 1 s	R
D8431			Every 1 s	R
D8432			Every 1 s	R
D8433			Every 1 s	R
D8434		Connected Port No.	Every 1 s	R
D8435 D8436	**— Reserved —**		—	—
D8437	HMI Module IP Address (Write-only)		—	W
D8438			—	W
D8439			—	W
D8440			—	W
D8441	HMI Module Subnet Mask (Write-only)		—	W
D8442			—	W
D8443			—	W
D8444			—	W
D8445	HMI Module Default Gateway (Write-only)		—	W
D8446			—	W
D8447			—	W
D8448			—	W
D8449	HMI Module Preferred DNS Server (Write-only)		—	W
D8450			—	W
D8451			—	W
D8452			—	W
D8453	HMI Module Alternate DNS Server (Write-only)		—	W
D8454			—	W
D8455			—	W
D8456			—	W
D8457	EMAIL Instruction Detailed Error Information (HMI-Ethernet Port)		—	R
D8458 to D8469	**— Reserved —**		—	—
D8470	Expansion Module Slot 1 Information	Type ID/Status	—	R
D8471		System Software Version/Position Information	—	R
D8472	Expansion Module Slot 2 Information	Type ID/Status	—	R
D8473		System Software Version/Position Information	—	R

Device Address	Description		Update Timing	R/W
D8474	Expansion Module Slot 3 Information	Type ID/Status	—	R
D8475		System Software Version/Position Information	—	R
D8476	Expansion Module Slot 4 Information	Type ID/Status	—	R
D8477		System Software Version/Position Information	—	R
D8478	Expansion Module Slot 5 Information	Type ID/Status	—	R
D8479		System Software Version/Position Information	—	R
D8480	Expansion Module Slot 6 Information	Type ID/Status	—	R
D8481		System Software Version/Position Information	—	R
D8482	Expansion Module Slot 7 Information	Type ID/Status	—	R
D8483		System Software Version/Position Information	—	R
D8484	Expansion Module Slot 8 Information	Type ID/Status	—	R
D8485		System Software Version/Position Information	—	R
D8486	Expansion Module Slot 9 Information	Type ID/Status	—	R
D8487		System Software Version/Position Information	—	R
D8488	Expansion Module Slot 10 Information	Type ID/Status	—	R
D8489		System Software Version/Position Information	—	R
D8490	Expansion Module Slot 11 Information	Type ID/Status	—	R
D8491		System Software Version/Position Information	—	R
D8492	Expansion Module Slot 12 Information	Type ID/Status	—	R
D8493		System Software Version/Position Information	—	R
D8494	Expansion Module Slot 13 Information	Type ID/Status	—	R
D8495		System Software Version/Position Information	—	R
D8496	Expansion Module Slot 14 Information	Type ID/Status	—	R
D8497		System Software Version/Position Information	—	R
D8498	Expansion Module Slot 15 Information	Type ID/Status	—	R
D8499		System Software Version/Position Information	—	R
D8500	Expansion Module Slot 16 Information	Type ID/Status	—	R
D8501		System Software Version/Position Information	—	R
D8502	Expansion Module Slot 17 Information	Type ID/Status	—	R
D8503		System Software Version/Position Information	—	R
D8504	Expansion Module Slot 18 Information	Type ID/Status	—	R
D8505		System Software Version/Position Information	—	R
D8506	Expansion Module Slot 19 Information	Type ID/Status	—	R
D8507		System Software Version/Position Information	—	R
D8508	Expansion Module Slot 20 Information	Type ID/Status	—	R
D8509		System Software Version/Position Information	—	R
D8510	Expansion Module Slot 21 Information	Type ID/Status	—	R
D8511		System Software Version/Position Information	—	R
D8512	Expansion Module Slot 22 Information	Type ID/Status	—	R
D8513		System Software Version/Position Information	—	R
D8514	Expansion Module Slot 23 Information	Type ID/Status	—	R
D8515		System Software Version/Position Information	—	R
D8516	Expansion Module Slot 24 Information	Type ID/Status	—	R
D8517		System Software Version/Position Information	—	R
D8518	Expansion Module Slot 25 Information	Type ID/Status	—	R
D8519		System Software Version/Position Information	—	R
D8520	Expansion Module Slot 26 Information	Type ID/Status	—	R
D8521		System Software Version/Position Information	—	R
D8522	Expansion Module Slot 27 Information	Type ID/Status	—	R
D8523		System Software Version/Position Information	—	R
D8524	Expansion Module Slot 28 Information	Type ID/Status	—	R
D8525		System Software Version/Position Information	—	R
D8526	Expansion Module Slot 29 Information	Type ID/Status	—	R
D8527		System Software Version/Position Information	—	R

Device Address	Description		Update Timing	R/W
D8528	Expansion Module Slot 30 Information	Type ID/Status	—	R
D8529		System Software Version/Position Information	—	R
D8530	Expansion Module Slot 31 Information	Type ID/Status	—	R
D8531		System Software Version/Position Information	—	R
D8532	Expansion Module Slot 32 Information	Type ID/Status	—	R
D8533		System Software Version/Position Information	—	R
D8534	Expansion Module Slot 33 Information	Type ID/Status	—	R
D8535		System Software Version/Position Information	—	R
D8536	Expansion Module Slot 34 Information	Type ID/Status	—	R
D8537		System Software Version/Position Information	—	R
D8538	Expansion Module Slot 35 Information	Type ID/Status	—	R
D8539		System Software Version/Position Information	—	R
D8530	Expansion Module Slot 36 Information	Type ID/Status	—	R
D8541		System Software Version/Position Information	—	R
D8542	Expansion Module Slot 37 Information	Type ID/Status	—	R
D8543		System Software Version/Position Information	—	R
D8544	Expansion Module Slot 38 Information	Type ID/Status	—	R
D8545		System Software Version/Position Information	—	R
D8546	Expansion Module Slot 39 Information	Type ID/Status	—	R
D8547		System Software Version/Position Information	—	R
D8548	Expansion Module Slot 40 Information	Type ID/Status	—	R
D8549		System Software Version/Position Information	—	R
D8550	Expansion Module Slot 41 Information	Type ID/Status	—	R
D8551		System Software Version/Position Information	—	R
D8552	Expansion Module Slot 42 Information	Type ID/Status	—	R
D8553		System Software Version/Position Information	—	R
D8554	Expansion Module Slot 43 Information	Type ID/Status	—	R
D8555		System Software Version/Position Information	—	R
D8556	Expansion Module Slot 44 Information	Type ID/Status	—	R
D8557		System Software Version/Position Information	—	R
D8558	Expansion Module Slot 45 Information	Type ID/Status	—	R
D8559		System Software Version/Position Information	—	R
D8560	Expansion Module Slot 46 Information	Type ID/Status	—	R
D8561		System Software Version/Position Information	—	R
D8562	Expansion Module Slot 47 Information	Type ID/Status	—	R
D8563		System Software Version/Position Information	—	R
D8564	Expansion Module Slot 48 Information	Type ID/Status	—	R
D8565		System Software Version/Position Information	—	R
D8566	Expansion Module Slot 49 Information	Type ID/Status	—	R
D8567		System Software Version/Position Information	—	R
D8568	Expansion Module Slot 50 Information	Type ID/Status	—	R
D8569		System Software Version/Position Information	—	R
D8570	Expansion Module Slot 51 Information	Type ID/Status	—	R
D8571		System Software Version/Position Information	—	R
D8572	Expansion Module Slot 52 Information	Type ID/Status	—	R
D8573		System Software Version/Position Information	—	R
D8574	Expansion Module Slot 53 Information	Type ID/Status	—	R
D8575		System Software Version/Position Information	—	R
D8576	Expansion Module Slot 54 Information	Type ID/Status	—	R
D8577		System Software Version/Position Information	—	R
D8578	Expansion Module Slot 55 Information	Type ID/Status	—	R
D8579		System Software Version/Position Information	—	R
D8580	Expansion Module Slot 56 Information	Type ID/Status	—	R
D8581		System Software Version/Position Information	—	R

Device Address	Description		Update Timing	R/W
D8582	Expansion Module Slot 57 Information	Type ID/Status	—	R
D8583		System Software Version/Position Information	—	R
D8584	Expansion Module Slot 58 Information	Type ID/Status	—	R
D8585		System Software Version/Position Information	—	R
D8586	Expansion Module Slot 59 Information	Type ID/Status	—	R
D8587		System Software Version/Position Information	—	R
D8588	Expansion Module Slot 60 Information	Type ID/Status	—	R
D8589		System Software Version/Position Information	—	R
D8590	Expansion Module Slot 61 Information	Type ID/Status	—	R
D8591		System Software Version/Position Information	—	R
D8592	Expansion Module Slot 62 Information	Type ID/Status	—	R
D8593		System Software Version/Position Information	—	R
D8594	Expansion Module Slot 63 Information	Type ID/Status	—	R
D8595		System Software Version/Position Information	—	R
D8596	Expansion Interface Remote Master Module Slot Information	Type ID/Status	—	R
D8597		System Software Version/ Number of Connected Expansion Interface Remote Slave Modules	—	R
D8598	Expansion Interface Remote Slave Module (Node 1) Slot Information	Type ID/Status	—	R
D8599		System Software Version/ Expansion Module Connection Information	—	R
D8600	Expansion Interface Remote Slave Module (Node 2) Slot Information	Type ID/Status	—	R
D8601		System Software Version/ Expansion Module Connection Information	—	R
D8602	Expansion Interface Remote Slave Module (Node 3) Slot Information	Type ID/Status	—	R
D8603		System Software Version/ Expansion Module Connection Information	—	R
D8604	Expansion Interface Remote Slave Module (Node 4) Slot Information	Type ID/Status	—	R
D8605		System Software Version/ Expansion Module Connection Information	—	R
D8606	Expansion Interface Remote Slave Module (Node 5) Slot Information	Type ID/Status	—	R
D8607		System Software Version/ Expansion Module Connection Information	—	R
D8608	Expansion Interface Remote Slave Module (Node 6) Slot Information	Type ID/Status	—	R
D8609		System Software Version/ Expansion Module Connection Information	—	R
D8610	Expansion Interface Remote Slave Module (Node 7) Slot Information	Type ID/Status	—	R
D8611		System Software Version/ Expansion Module Connection Information	—	R
D8612	Expansion Interface Remote Slave Module (Node 8) Slot Information	Type ID/Status	—	R
D8613		System Software Version/ Expansion Module Connection Information	—	R
D8614	Expansion Interface Remote Slave Module (Node 9) Slot Information	Type ID/Status	—	R
D8615		System Software Version/ Expansion Module Connection Information	—	R
D8616	Expansion Interface Remote Slave Module (Node 10) Slot Information	Type ID/Status	—	R
D8617		System Software Version/ Expansion Module Connection Information	—	R
D8618	Refresh Time of Expansion Interface Remote Master/Slave Modules Current Value		Every scan	R
D8619	Refresh Time of Expansion Interface Remote Master/Slave Modules Maximum Value		Every scan	R
D8620 to D8629	— **Reserved** —		—	—
D8630	CPU Module Ethernet Port 2 IP Settings/DNS Settings Switching		—	R/W
D8631	CPU Module Ethernet Port 2 IP Address (Write-only)		—	W
D8632			—	W
D8633			—	W
D8634			—	W

Device Address	Description	Update Timing	R/W
D8635	CPU Module Ethernet Port 2 Subnet Mask (Write-only)	—	W
D8636		—	W
D8637		—	W
D8638		—	W
D8639	CPU Module Ethernet Port 2 Default Gateway (Write-only)	—	W
D8640		—	W
D8641		—	W
D8642		—	W
D8643	CPU Module Ethernet Port 2 Preferred DNS Server (Write-only)	—	W
D8644		—	W
D8645		—	W
D8646		—	W
D8647	CPU Module Ethernet Port 2 Alternate DNS Server (Write-only)	—	W
D8648		—	W
D8649		—	W
D8650		—	W
D8651	CPU Module Ethernet Port 2 MAC Address (Current Value Read-only)	Every 1 s	R
D8652		Every 1 s	R
D8653		Every 1 s	R
D8654		Every 1 s	R
D8655		Every 1 s	R
D8656		Every 1 s	R
D8657	CPU Module Ethernet Port 2 IP Address (Current Value Read-only)	Every 1 s	R
D8658		Every 1 s	R
D8659		Every 1 s	R
D8660		Every 1 s	R
D8661	CPU Module Ethernet Port 2 Subnet Mask (Current Value Read-only)	Every 1 s	R
D8662		Every 1 s	R
D8663		Every 1 s	R
D8664		Every 1 s	R
D8665	CPU Module Ethernet Port 2 Default Gateway (Current Value Read-only)	Every 1 s	R
D8666		Every 1 s	R
D8667		Every 1 s	R
D8668		Every 1 s	R
D8669	CPU Module Ethernet Port 2 Preferred DNS Server (Current Value Read-only)	Every 1 s	R
D8670		Every 1 s	R
D8671		Every 1 s	R
D8672		Every 1 s	R
D8673	CPU Module Ethernet Port 2 Alternate DNS Server (Current Value Read-only)	Every 1 s	R
D8674		Every 1 s	R
D8675		Every 1 s	R
D8676		Every 1 s	R
D8677	Connection 9 Connected IP Address	Every 1 s	R
D8678		Every 1 s	R
D8679		Every 1 s	R
D8680		Every 1 s	R
D8681	Connection 10 Connected IP Address	Every 1 s	R
D8682		Every 1 s	R
D8683		Every 1 s	R
D8684		Every 1 s	R
D8685	Connection 11 Connected IP Address	Every 1 s	R
D8686		Every 1 s	R
D8687		Every 1 s	R
D8688		Every 1 s	R

Device Address	Description	Update Timing	R/W
D8689	Connection 12 Connected IP Address	Every 1 s	R
D8690		Every 1 s	R
D8691		Every 1 s	R
D8692		Every 1 s	R
D8693	Connection 13 Connected IP Address	Every 1 s	R
D8694		Every 1 s	R
D8695		Every 1 s	R
D8696		Every 1 s	R
D8697	Connection 14 Connected IP Address	Every 1 s	R
D8698		Every 1 s	R
D8699		Every 1 s	R
D8700		Every 1 s	R
D8701	Connection 15 Connected IP Address	Every 1 s	R
D8702		Every 1 s	R
D8703		Every 1 s	R
D8704		Every 1 s	R
D8705	Connection 16 Connected IP Address	Every 1 s	R
D8706		Every 1 s	R
D8707		Every 1 s	R
D8708		Every 1 s	R
D8709 to D8716	**— Reserved —**	—	—
D8717	Control Signal Status (Port 10 to 13)	Every scan	R
D8718	RS232C DSR Input Control Signal Option (Port 10 to 13)	Every scan	R/W
D8719	RS232C DTR Output Control Signal Option (Port 10 to 13)	Every scan	R/W
D8720	Control Signal Status (Port 14 to 17)	Every scan	R
D8721	RS232C DSR Input Control Signal Option (Port 14 to 17)	Every scan	R/W
D8722	RS232C DTR Output Control Signal Option (Port 14 to 17)	Every scan	R/W
D8723	Control Signal Status (Port 18 to 21)	Every scan	R
D8724	RS232C DSR Input Control Signal Option (Port 18 to 21)	Every scan	R/W
D8725	RS232C DTR Output Control Signal Option (Port 18 to 21)	Every scan	R/W
D8726	Control Signal Status (Port 22 to 25)	Every scan	R
D8727	RS232C DSR Input Control Signal Option (Port 22 to 25)	Every scan	R/W
D8728	RS232C DTR Output Control Signal Option (Port 22 to 25)	Every scan	R/W
D8729	Control Signal Status (Port 26 to 29)	Every scan	R
D8730	RS232C DSR Input Control Signal Option (Port 26 to 29)	Every scan	R/W
D8731	RS232C DTR Output Control Signal Option (Port 26 to 29)	Every scan	R/W
D8732	Control Signal Status (Port 30 to 33)	Every scan	R
D8733	RS232C DSR Input Control Signal Option (Port 30 to 33)	Every scan	R/W
D8734	RS232C DTR Output Control Signal Option (Port 30 to 33)	Every scan	R/W
D8735	Slave Number (Port 10)	—	R/W
D8736	Slave Number (Port 11)	—	R/W
D8737	Slave Number (Port 12)	—	R/W
D8738	Slave Number (Port 13)	—	R/W
D8739	Slave Number (Port 14)	—	R/W
D8740	Slave Number (Port 15)	—	R/W
D8741	Slave Number (Port 16)	—	R/W
D8742	Slave Number (Port 17)	—	R/W
D8743	Slave Number (Port 18)	—	R/W
D8744	Slave Number (Port 19)	—	R/W
D8745	Slave Number (Port 20)	—	R/W
D8746	Slave Number (Port 21)	—	R/W
D8747	Slave Number (Port 22)	—	R/W
D8748	Slave Number (Port 23)	—	R/W
D8749	Slave Number (Port 24)	—	R/W

Device Address	Description		Update Timing	R/W
D8750	Slave Number (Port 25)		—	R/W
D8751	Slave Number (Port 26)		—	R/W
D8752	Slave Number (Port 27)		—	R/W
D8753	Slave Number (Port 28)		—	R/W
D8754	Slave Number (Port 29)		—	R/W
D8755	Slave Number (Port 30)		—	R/W
D8756	Slave Number (Port 31)		—	R/W
D8757	Slave Number (Port 32)		—	R/W
D8758	Slave Number (Port 33)		—	R/W
D8759	EMAIL Instruction Detailed Error Information (Ethernet Port 1)		—	R
D8760	Communication Mode Information (Client Connection)	Connection 9 to 12	—	R
D8761		Connection 13 to 16	—	R
D8762 to D8773	**— Reserved —**		—	—
D8774	Remote Host Number of Connection 9 (1 to 255)		—	R/W
D8775	Remote Host Number of Connection 10 (1 to 255)		—	R/W
D8776	Remote Host Number of Connection 11 (1 to 255)		—	R/W
D8777	Remote Host Number of Connection 12 (1 to 255)		—	R/W
D8778	Remote Host Number of Connection 13 (1 to 255)		—	R/W
D8779	Remote Host Number of Connection 14 (1 to 255)		—	R/W
D8780	Remote Host Number of Connection 15 (1 to 255)		—	R/W
D8781	Remote Host Number of Connection 16 (1 to 255)		—	R/W
D8782	BACnet Operation Status		—	R
D8783	BACnet Error Information		—	R
D8784 to D8789	**— Reserved —**		—	—
D8790	EtherNet/IP Operation Status		—	R
D8791	EtherNet/IP Error Information		—	R
D8792 to D8899	**— Reserved —**		—	—

Supplementary Descriptions of the Special Data Registers

■ D8000: Quantity of Inputs

The number of FC6A Series MICROSmart inputs is written to this register.
The total number of inputs for the CPU module and the expansion modules is written to this register.

■ D8001: Quantity of Outputs

The number of FC6A Series MICROSmart outputs is written to this register.
The total number of outputs for the CPU module and the expansion modules is written to this register.

■ D8002: CPU Module Type Information

CPU module type information is written this register.
0 (00h): 16-I/O type
1 (01h): 24-I/O type
2 (02h): 40-I/O type (All-in-One CPU module)
18 (12h): 40-I/O type (CAN J1939 All-in-One CPU module)
32 (20h): Plus 16-I/O type
33 (21h): Plus 32-I/O type

■ D8005: General Error Code

FC6A Series MICROSmart general error information is written to this register. When a general error occurs, the bit corresponding to the error occurred turns on. The general error and user program execution error can be cleared by writing "1" to the most significant bit of D8005. For details on general error codes, see Chapter 13 "General Error Codes" in the "FC6A Series MICROSmart User's Manual".

■ D8006: User Program Execution Error Code

FC6A Series MICROSmartuser program execution error information is written to this register. When a user program execution error occurs, the error code corresponding to the error that occurred is written to this register. For details on user program execution errors, see Chapter 13 "User Program Execution Error" in the "FC6A Series MICROSmart User's Manual".

■ D8008 to D8021: Calendar/Clock Data

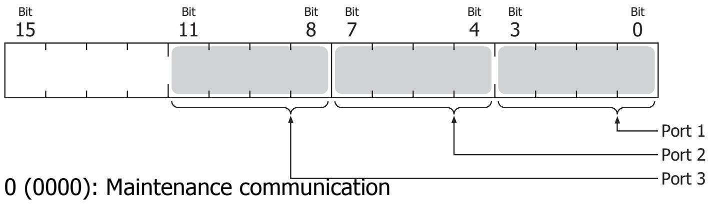
D8008 through D8021 are used for reading calendar/clock data from the internal clock and for writing calendar/clock data to the internal clock. For details on the calendar/clock data, see Chapter 5 "Clock Function" in the "FC6A Series MICROSmart User's Manual".

■ D8022 to D8025: Scan Time Data

D8022 through D8025 are special data registers for checking the scan time and configuring the constant scan time. For details on the scan time, see Chapter 5 "Constant Scan Time" in the "FC6A Series MICROSmart User's Manual".

■ D8026: Communication Mode Information (Port 1 to 3)

This register indicates communication mode information for Port 1 to Port 3.
The allocation of communication ports in the device (bit assignment) is as follows.



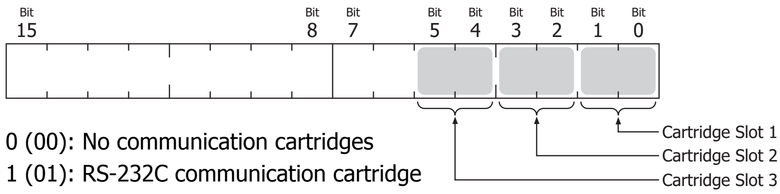
0 (0000): Maintenance communication
1 (0001): User communication
2 (0010): Modbus RTU master
3 (0011): Modbus RTU slave
4 (0100): Data link communication

■ D8029: System Software Version

The CPU module system software version number is written to this register.

■ D8030: Communication Cartridge Information

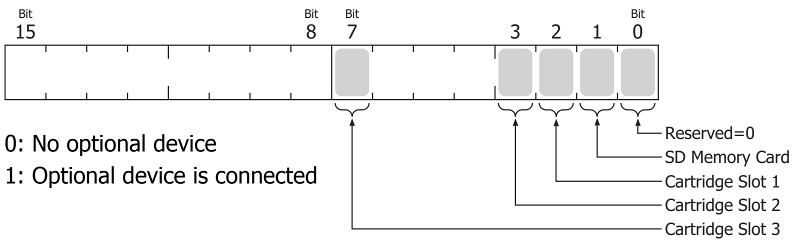
This register indicates information about the communication cartridges in Port 2 and Port 3.
The allocation of communication ports in the device (bit assignment) is as follows.



0 (00): No communication cartridges
1 (01): RS-232C communication cartridge
2 (10): RS-485 communication cartridge
3 (11): Bluetooth communication cartridge

■ D8031: Optional Device Connection Information

Optional device connection information is written to this register.
The allocation of bits in the device (bit assignment) is as follows.



0: No optional device
1: Optional device is connected

■ D8032 to D8035, D8214, D8215: Interrupt Input Jump Destination Label No.

Jump destination label numbers for interrupt inputs are written to these registers. When using interrupt inputs, write the label number that corresponds to the special data register that has been allocated to the interrupt input. For details on interrupt inputs, see Chapter 5 "Interrupt Input" in the "FC6A Series MICROSmart User's Manual".
D8032 = I1

D8033 = I3
D8034 = I4
D8035 = I6
D8214 = I7
D8215 = I0

■ **D8036: Timer Interrupt Jump Destination Label No.**
The jump destination label number when the timer interrupt occurs is written to this register. When using the timer interrupt, store the corresponding label number.
For details on the timer interrupt, see Chapter 5 "Timer Interrupt" in the "FC6A Series MICROSmart User's Manual".

■ **D8037: Number of Connected Expansion Modules**
The number of expansion modules connected to the CPU module (I/O modules, PID modules, and communication modules) is written to this register.

■ **D8052: J1939 Communication Error Code**
When an error occurs in J1939 communication, the error code is written to this register. For details on J1939 communication error codes, see "J1939 Communication Error Code (D8052)" on page 8-5.

■ **D8056: Battery Voltage**
The measurement result of the backup battery voltage is written to this register in 1 mV units.
0: Measurement error

■ **D8057: Analog Volume (AI0)**
The analog volume value is converted to a digital value and written to this register. For details, see Chapter 5 "Analog Potentiometer" in the "FC6A Series MICROSmart User's Manual".

■ **D8058: Built-in Analog Input (AI1)**
The analog input's analog value (0 to 10V DC) is converted to a digital value and written to this register. For details, see Chapter 5 "Analog Voltage Input" in the "FC6A Series MICROSmart User's Manual".

■ **D8059: Analog Input Status AI0**
The analog volume's analog input status is written to this register. For details, see Chapter 5 "Analog Potentiometer" in the "FC6A Series MICROSmart User's Manual".

■ **D8060: Analog Input Status AI1**
The analog input's analog input status is written to this register. For details, see Chapter 5 "Analog Voltage Input" in the "FC6A Series MICROSmart User's Manual".

■ **D8067: Backlight ON Time**
The backlight on time for the HMI module is written to this register. The backlight on time can be set by modifying the value of D8067 in a range between 1 and 65,535 s. When D8067 is 0 s, the backlight is always on. The backlight on time can also be changed with the HMI environment settings. For details, see "Master Station" on page 7-4.

■ **D8069 to D8099: Slave (1 to 31) Communication Status/Error**
Special data registers used in data link communication. For details, see "Master Station" on page 7-4.

■ **D8040 to D8045, D8100, D8102, D8103, D8735 to D8758: Slave Number (Port 1 to 33)**
The slave number is written to these registers when the Port 1 to 33 communication mode is maintenance communication, Modbus RTU slave, or data link communication. The slave number can be changed by changing the value of the corresponding device in **Function Area Settings**.

D8100 = Port 1 Slave No.
D8102 = Port 2 Slave No.
D8103 = Port 3 Slave No.

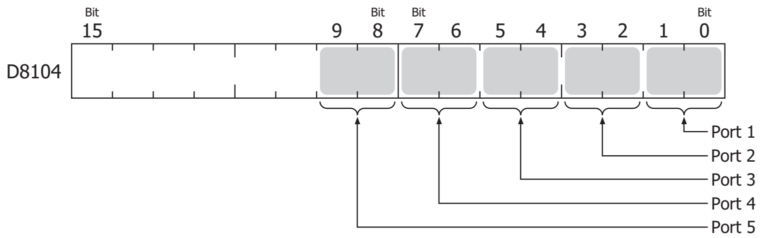
D8040 = Port 4 Slave No.
D8041 = Port 5 Slave No.
D8042 = Port 6 Slave No.
D8043 = Port 7 Slave No.
D8044 = Port 8 Slave No.
D8045 = Port 9 Slave No.
D8735 = Port 10 Slave No.
D8736 = Port 11 Slave No.
D8737 = Port 12 Slave No.
D8738 = Port 13 Slave No.
D8739 = Port 14 Slave No.
D8740 = Port 15 Slave No.
D8741 = Port 16 Slave No.
D8742 = Port 17 Slave No.
D8743 = Port 18 Slave No.
D8744 = Port 19 Slave No.
D8745 = Port 20 Slave No.
D8746 = Port 21 Slave No.
D8747 = Port 22 Slave No.
D8748 = Port 23 Slave No.
D8749 = Port 24 Slave No.
D8750 = Port 25 Slave No.
D8751 = Port 26 Slave No.
D8752 = Port 27 Slave No.
D8753 = Port 28 Slave No.
D8754 = Port 29 Slave No.
D8755 = Port 30 Slave No.
D8756 = Port 31 Slave No.
D8757 = Port 32 Slave No.
D8758 = Port 33 Slave No.

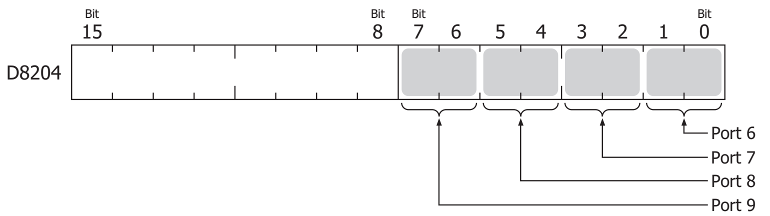
For details on communication modes, see the following in this manual.

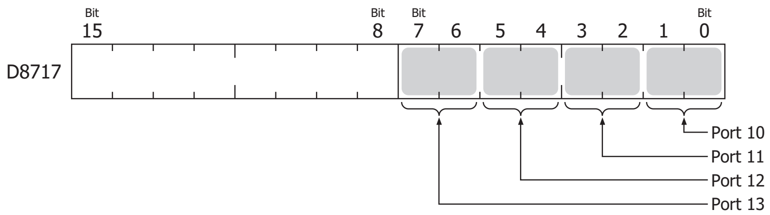
- Maintenance communication: "Maintenance Communication" on page 4-1
- Modbus RTU slave: "Modbus Communication" on page 6-1
- Data link communication: "Data Link Communication" on page 7-1

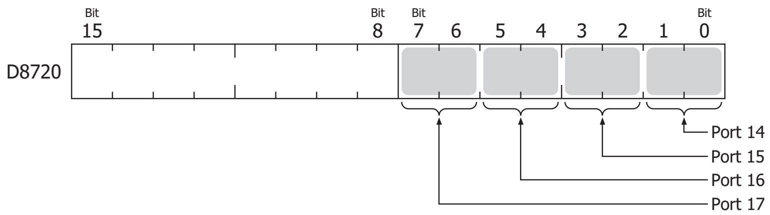
■ D8104, D8204, D8717, D8720, D8723, D8726, D8729, D8732: Control Signal Status (Port 1 to 33)

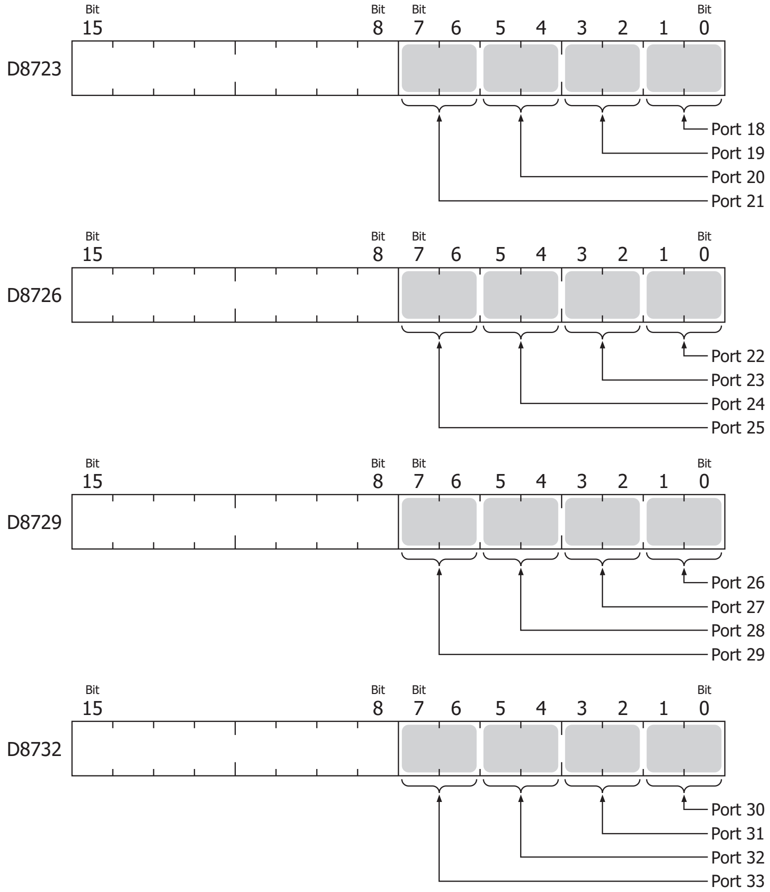
The signal statuses of the DSR and DTR controls lines are written to this register. This register is updated in END processing when stopped and while running. The allocation of communication ports in the device (bit assignment) is as follows.







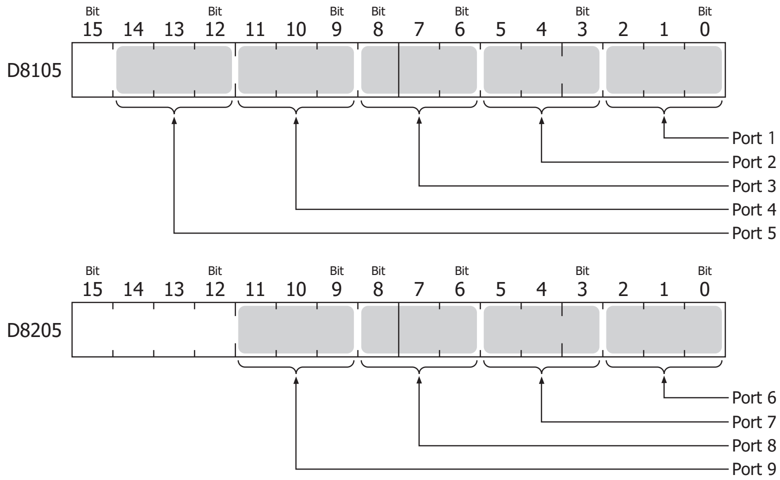


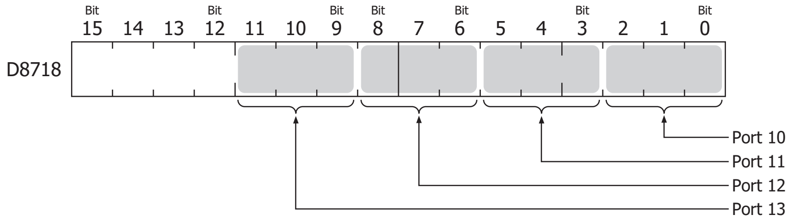


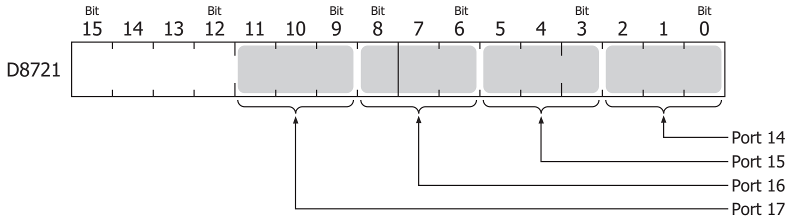
0 (00): DTR and DSR are both off.
1 (01): DTR is off and DSR is on.
2 (10): DTR is on and DSR is off.
3 (11): DTR and DSR are both on.

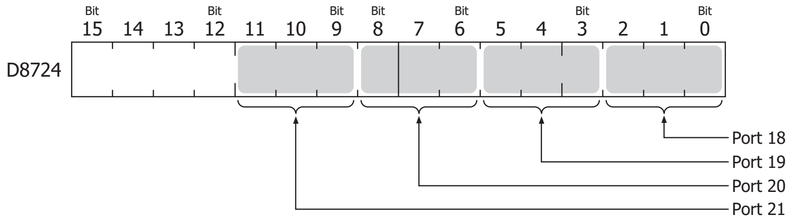
■ **D8105, D8205, D8718, D8721, D8724, D8727, D8730, D8733: RS232C DSR Input Control Signal Option (Port 1 to 33)**

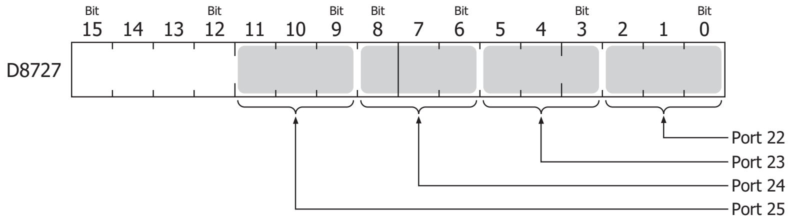
The signal statuses of the DSR and DTR controls lines are written to this register. This register is updated in END processing when stopped and while running. The allocation of communication ports in the device (bit assignment) is as follows.

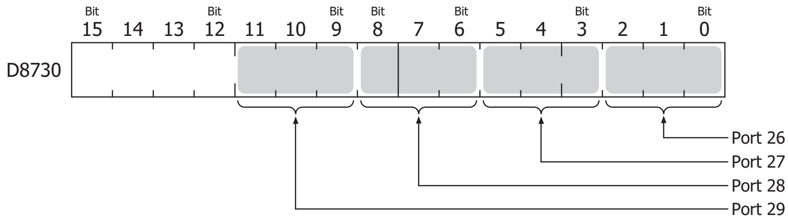


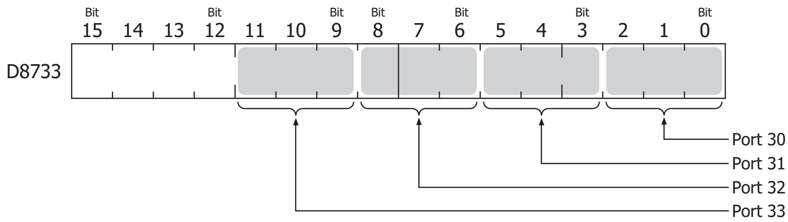






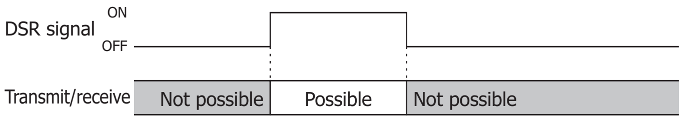




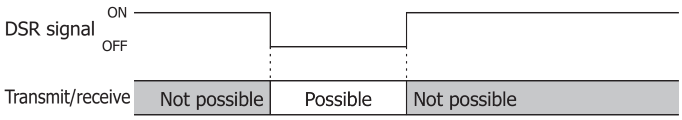


0 (000): The DSR signal status is not used for FC6A Series MICROSmart transmission control. Use this status when DSR signal control is not required.

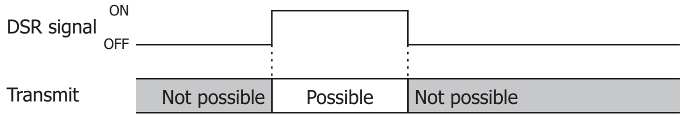
1 (001): When the DSR signal is on, the FC6A Series MICROSmart can transmit and receive.



2 (010): When the DSR signal is off, the FC6A Series MICROSmart can transmit and receive.



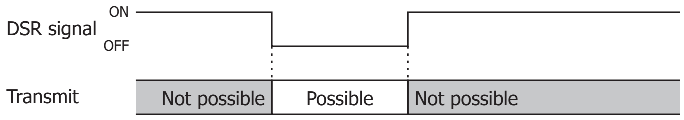
3 (011): When the DSR signal is on, transmission is possible (reception is always possible).



This is normally called "busy control" and is used for transmission control for devices with a slow processing speed such as printers.

(As viewed from the connected device, there is a limit on the data that can be input.)

4 (100): When the DSR signal is off, transmission is possible.

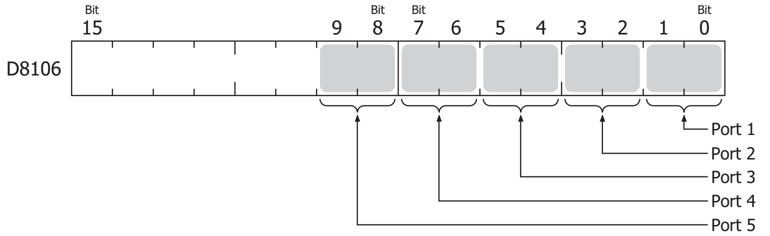


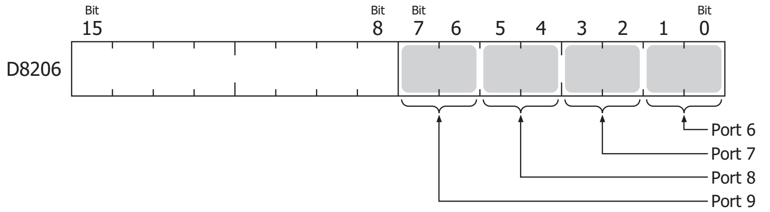
5 or higher: The operation is the same as the setting "000".

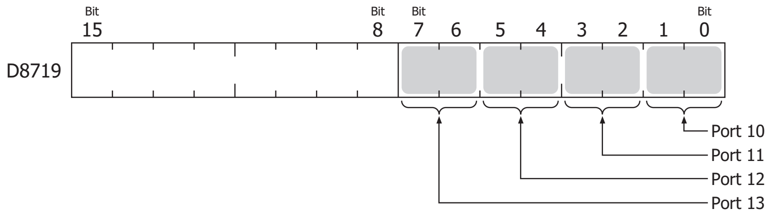
- **D8106, D8206, D8719, D8722, D8725, D8728, D8731, D8734: RS232C DTR Output Control Signal Option (Port 1 to 33)**

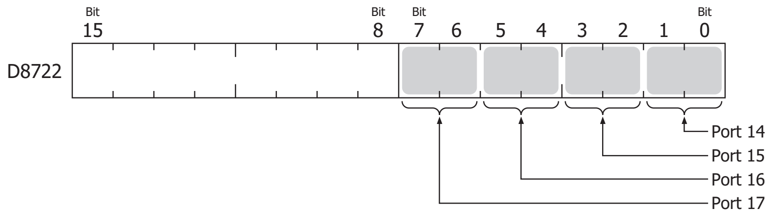
  This register is used when indicating the FC6A Series MICROSmart control status and the transmit/receive status to the connected device. This control line is an output signal from the FC6A Series MICROSmart to the connected device. This register is only valid during user communication.

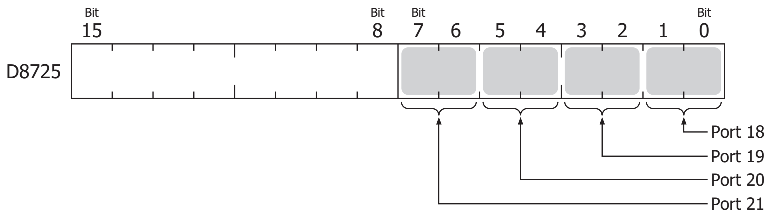
  The allocation of communication ports in the device (bit assignment) is as follows.

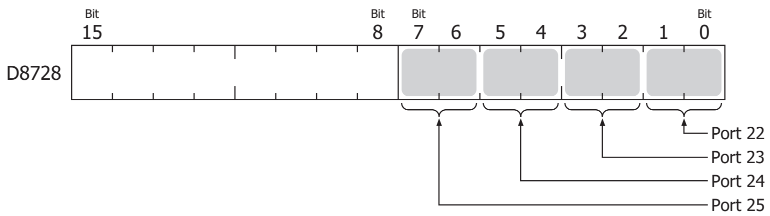


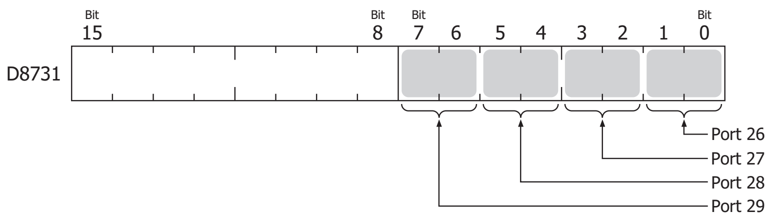


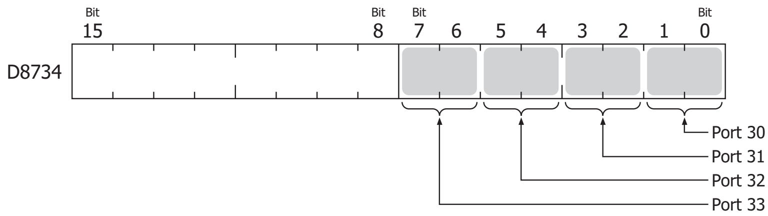




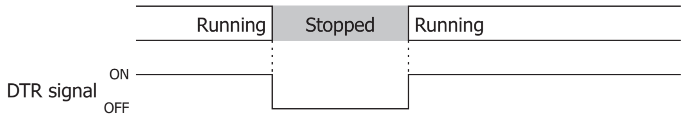






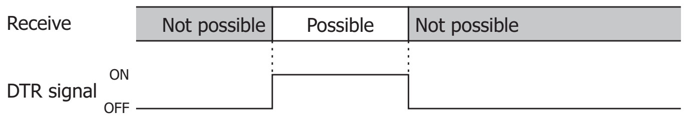


0 (00): The signal is on when the FC6A Series MICROSmart is set to run and off when stopped.
While running, the signal is always on regardless of transmitting or receiving data. Set this value when it is necessary to indicate the run status.



1 (01): Always off.

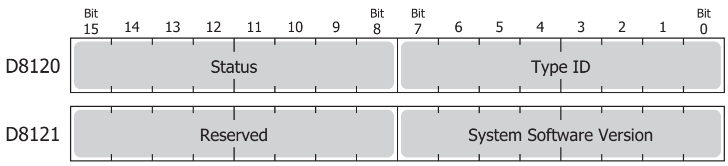
2 (10): Set this value when performing flow control for received data. The signal is on when data from the connected device can be received.
The signal is off when data cannot be received.



3 (11): The operation is the same as the setting "0".

■ D8120, D8121: HMI Module Information

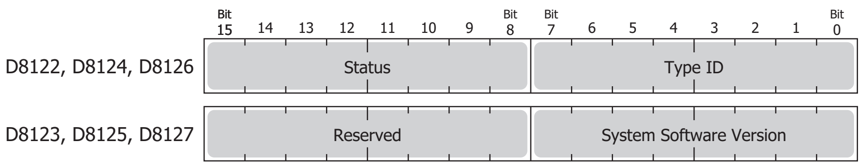
HMI module type information is written to these registers.
The allocation of information (bit assignment) is as follows.



The system software version displays the version of the software written to the device.
For details on the type IDs and statuses, see "List of Type IDs and Status" on page 2-47.

■ D8122 to D8127: Cartridge Slot Information

Cartridge Slot 1 to 3 information is written to these registers.
The allocation of information (bit assignment) is as follows.



D8122, D8123 = Cartridge Slot 1 Information
D8124, D8125 = Cartridge Slot 2 Information
D8126, D8127 = Cartridge Slot 3 Information
For details on the type IDs and statuses, see "List of Type IDs and Status" on page 2-47.

■ D8170, D8171, D8174, D8175, D8178, D8179: Analog I/O Cartridge I/O

Analog I/O values for the analog I/O cartridges are written to these registers.

For the analog input type : The analog values input to the analog I/O cartridge are converted to digital values and written to the registers.

For the analog output type : The digital values stored as digital values are converted to analog values and output from the analog I/O cartridges.

For details, see Chapter 10 "Analog I/O Cartridge" in the "FC6A Series MICROSmart User’s Manual".

D8170 = AI2/AQ2
D8171 = AI3/AQ3
D8174 = AI4/AQ4
D8175 = AI5/AQ5
D8178 = AI6/AQ6
D8179 = AI7/AQ7

■ D8172, D8173, D8176, D8177, D8180, D8181: Analog I/O Cartridge Status

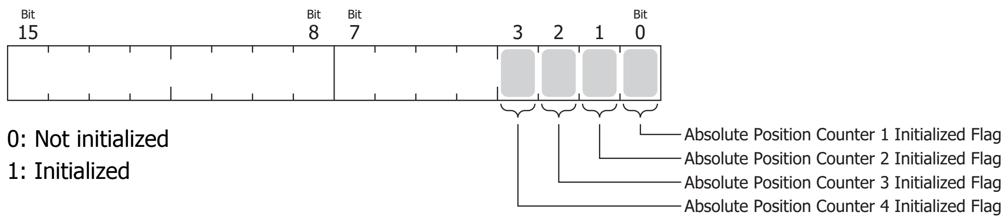
The analog status of analog I/O cartridges is written to these registers.
For details, see Chapter 10 "Analog I/O Cartridge" in the "FC6A Series MICROSmart User’s Manual".
D8172 = AI2/AQ2
D8173 = AI3/AQ3
D8176 = AI4/AQ4
D8177 = AI5/AQ5
D8180 = AI6/AQ6
D8181 = AI7/AQ7

■ D8192 to D8203, D8210 to D8213, D8216 to D8229, D8232 to D8237: High-speed Counter

These special data registers are used with the high-speed counter function and the frequency measurement function.
For details on high-speed counters, see Chapter 5 "High-Speed Counter" in the "FC6A Series MICROSmart User’s Manual".
D8210 to D8213, D8216, D8217 = High-speed counter (group 1/I0)
D8218 to D8221, D8234, D8235 = High-speed counter (group 3/I3)
D8222 to D8225, D8236, D8237 = High-speed counter (group 4/I4)
D8226 to D8229, D8232, D8233 = High-speed counter (group 5/I6)
D8192 to D8197 = High-speed counter (group 2/I1)
D8198 to D8203 = High-speed counter (group 6/I7)

■ D8239: Absolute Position Control Status

This register indicates the status of absolute position control.
The allocation of the absolute position counter initialized flags in the device (bit assignment) is as follows. For details on the absolute position control status, see Chapter 18 "ABS (Set Absolute Position)" in the "FC6A Series MICROSmart Ladder Programming Manual".



■ D8240 to D8247: Absolute Position Counter 1 to 4

The absolute position is written to these registers according to pulse output. The absolute position can be initialized with the ABS instruction.
When reversible control is set in the pulse output instructions (RAMP/ARAMP/ZRN/JOG), the absolute position is incremented or decremented according to the output result. For details on absolute position counters, see Chapter 18 "ABS (Set Absolute Position)" in the "FC6A Series MICROSmart Ladder Programming Manual".
D8240, D8241 = Absolute position counter 1
D8242, D8243 = Absolute position counter 2
D8244, D8245 = Absolute position counter 3
D8246, D8247 = Absolute position counter 4

■ **D8250: Read SD Memory Card Capacity**
This register indicates the capacity of the recognized SD or SDHC (maximum size 32 GB) memory card in megabytes.
It is 0 when no SD memory card has been inserted or if it is not recognized.

■ **D8251: Read SD Memory Card Free Capacity**
This register indicates the free capacity of the SD memory card in megabytes.
It is 0 when no SD memory card has been inserted or if it is not recognized.

■ **D8254: SD Memory Card Download/Upload Execution Information**
Information about the SD memory card download/upload that was executed.
For details, see Chapter 11 "SD Memory Card" in the "FC6A Series MICROSmart User's Manual".

■ **D8255: SD Memory Card Download/Upload Execution Status**
The status of the SD memory card download/upload that was executed.
For details, see Chapter 11 "SD Memory Card" in the "FC6A Series MICROSmart User's Manual".

■ **D8260: Recipe Block Number**
The recipe block number to read or write using special internal relays.
All channels for the specified block number are applicable.

■ **D8261: Recipe Execution Block Number**
The recipe block number of the recipe that was executed.
This is updated when the recipe block starts, and the status is retained when completed.

■ **D8262: Recipe Execution Channel No.**
The channel number of the recipe that was executed.
This is updated when the channel processing starts, and the status is retained when completed.

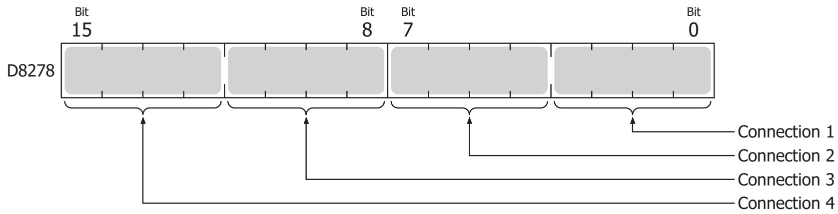
■ **D8263: Recipe Execution Operation**
Information about the operation of the recipe that was executed.
For details on recipes, see Chapter 11 "Recipe Function" in the "FC6A Series MICROSmart User's Manual".

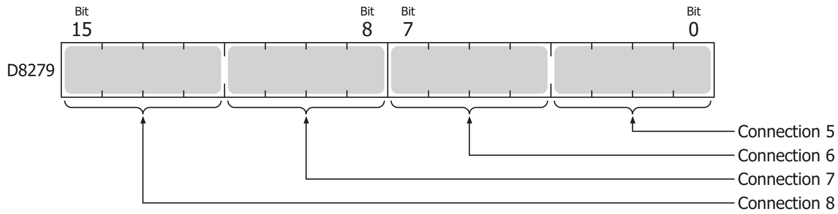
■ **D8264: Recipe Execution Status**
The status of the recipe that was executed.
For details on recipes, see Chapter 11 "Recipe Function" in the "FC6A Series MICROSmart User's Manual".

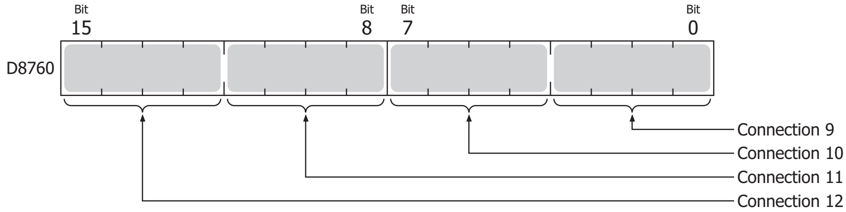
■ **D8265: Recipe Execution Error Information**
Error information about the recipe that was executed.
For details on recipes, see Chapter 11 "Recipe Function" in the "FC6A Series MICROSmart User's Manual".

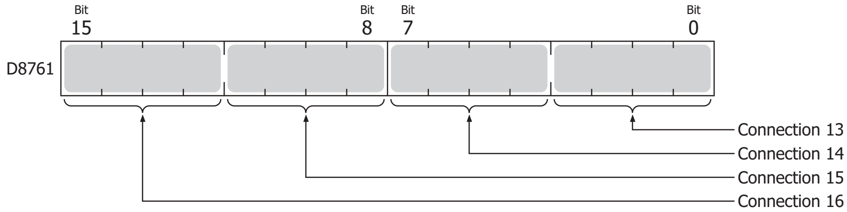
■ **D8268 to D8275, D8774 to D8781: Remote Host Number 1 to 255 (connection 1 to 16)**
Special data registers that change the communications device when user communications client is set. For details, see "Switching Remote Host Numbers" on page 5-43.

■ **D8278, D8279, D8760, D8761: Communication Mode Information (Client Connection) (Connection 1 to 16)**
D8278 = Indicates the communication mode of connections 1 through 4.
D8279 = Indicates the communication mode of connections 5 through 8.
D8760 = Indicates the communication mode of connections 9 through 12.
D8761 = Indicates the communication mode of connections 13 through 16.
The allocation of connections in the device (bit assignment) is as follows.





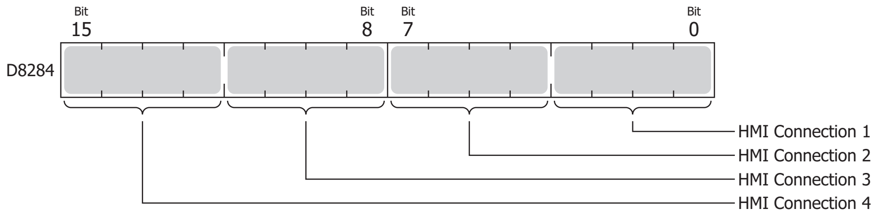


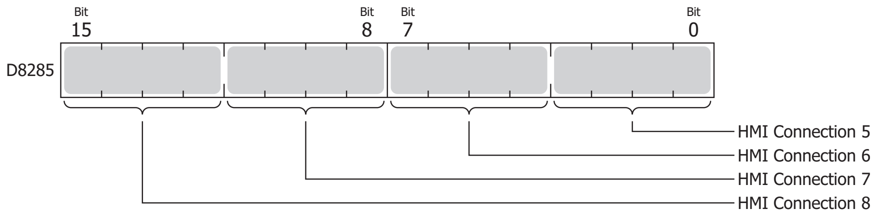


- Client connection (most significant bit = 0)
  0000: Unused
  0001: User Communication
  0010: Modbus TCP client
  0100: User communication UDP
- Server connection (most significant bit = 1)
  1000: Maintenance Communication
  1001: User Communication
  1010: Modbus TCP server

■ D8284, D8285: Communication Mode Information (HMI Connection) (HMI Connection 1 to 8)

D8284: Indicates the communication mode of HMI connections 1 through 4.
D8285: Indicates the communication mode of HMI connections 5 through 8.
The allocation of connections in the device (bit assignment) is as follows.





- Client connection (most significant bit = 0)
  0000: Unused
- Server connection (most significant bit = 1)
  1000: Maintenance Communication

■ D8303: CPU Module Ethernet Port 1 IP Settings/DNS Settings Switching

The IP settings/DNS settings for Ethernet Port 1 can be changed by writing one of the setting values in the table on the next page to D8303 and then turning on M8190.
To use this function, enable the following items in **Function Area Settings** of WindLDR.

- All-in-One CPU module/CAN J1939 All-in-One CPU module
  Enable D8303 (IP Settings / DNS Settings switching) on **Network Settings** in **Function Area Settings**
- Plus CPU module
  Enable D8303 (IP Settings / DNS Settings switching) on **Ethernet Port 1** in **Function Area Settings**

The meanings of the setting values are as follows.

Setting Value	IP Settings/DNS Settings
0	Conform to function area settings.
1	Enable DHCP.
2	Conform to special data register (D8304 to D8323) settings.

■ D8304 to D8307: CPU Module Ethernet Port 1 IP Address (Write-only)

These registers are used to write the CPU module's IP address.
IP address: To set as aaa.bbb.ccc.ddd, write the following.
D8304=aaa, D8305=bbb, D8306=ccc, D8307=ddd

■ D8308 to D8311: CPU Module Ethernet Port 1 Subnet Mask (Write-only)

These registers are used to write the CPU module's subnet mask.
Subnet mask: For aaa.bbb.ccc.ddd, write the following.
D8308=aaa, D8309=bbb, D8310=ccc, D8311=ddd

■ D8312 to D8315: CPU Module Ethernet Port 1 Default Gateway (Write-only)

These registers are used to write the CPU module's default gateway.
Default gateway: For aaa.bbb.ccc.ddd, write the following.
D8312=aaa, D8313=bbb, D8314=ccc, D8315=ddd

■ D8316 to D8319: CPU Module Ethernet Port 1 Preferred DNS Server (Write-only)

These registers are used to write the CPU module's preferred DNS server.
Preferred DNS server: For aaa.bbb.ccc.ddd, write the following.
D8316=aaa, D8317=bbb, D8318=ccc, D8319=ddd

■ D8320 to D8323: CPU Module Ethernet Port 1 Alternate DNS Server (Write-only)

These registers are used to write the CPU module's alternate DNS server.
Alternate DNS server: For aaa.bbb.ccc.ddd, write the following.
D8320=aaa, D8321=bbb, D8322=ccc, D8323=ddd

■ D8324 to D8329: CPU Module Ethernet Port 1 MAC Address (Current Value Read-only)

The CPU module's MAC address is written to the special data registers as follows.
Example: MAC address: AA-BB-CC-DD-EE-FF
D8324=AA, D8325=BB, D8326=CC, D8327=DD, D8328=EE, D8329=FF

■ D8330 to D8333: CPU Module Ethernet Port 1 IP Address (Current Value Read-only)

The CPU module's IP address is written to the special data registers as follows.
Example: The CPU module's own IP address aaa.bbb.ccc.ddd
D8330=aaa, D8331=bbb, D8332=ccc, D8333=ddd

■ D8334 to D8337: CPU Module Ethernet Port 1 Subnet Mask (Current Value Read-only)

The CPU module's subnet mask value is written to the special data registers as follows.
Example: Subnet mask: aaa.bbb.ccc.ddd
D8334=aaa, D8335=bbb, D8336=ccc, D8337=ddd

■ D8338 to D8341: CPU Module Ethernet Port 1 Default Gateway (Current Value Read-only)

The CPU module's default gateway address is written to the special data registers as follows.
Example: Default gateway: aaa.bbb.ccc.ddd
D8338=aaa, D8339=bbb, D8340=ccc, D8341=ddd

■ D8342 to D8345: CPU Module Ethernet Port 1 Preferred DNS Server (Current Value Read-only)

The CPU module's preferred DNS server address is written to the special data registers as follows.
Example: Preferred DNS server: aaa.bbb.ccc.ddd
D8342=aaa, D8343=bbb, D8344=ccc, D8345=ddd

■ D8346 to D8349: CPU Module Ethernet Port 1 Alternate DNS Server (Current Value Read-only)

The CPU module's alternate DNS server address is written to the special data registers as follows.
Example: Alternate DNS server: aaa.bbb.ccc.ddd
D8346=aaa, D8347=bbb, D8348=ccc, D8349=ddd

■ **D8350 to D8381, D8677 to D8708: Connection Connected IP Address**

The IP address of the connected device that is being accessed through a connection is written as follows.

Connection 1 Connected IP Address: For aaa.bbb.ccc.ddd
D8350=aaa, D8351=bbb, D8352=ccc, D8353=ddd

Connection 2 Connected IP Address: For aaa.bbb.ccc.ddd
D8354=aaa, D8355=bbb, D8356=ccc, D8357=ddd

Connection 3 Connected IP Address: For aaa.bbb.ccc.ddd
D8358=aaa, D8359=bbb, D8360=ccc, D8361=ddd

Connection 4 Connected IP Address: For aaa.bbb.ccc.ddd
D8362=aaa, D8363=bbb, D8364=ccc, D8365=ddd

Connection 5 Connected IP Address: For aaa.bbb.ccc.ddd
D8366=aaa, D8367=bbb, D8368=ccc, D8369=ddd

Connection 6 Connected IP Address: For aaa.bbb.ccc.ddd
D8370=aaa, D8371=bbb, D8372=ccc, D8373=ddd

Connection 7 Connected IP Address: For aaa.bbb.ccc.ddd
D8374=aaa, D8375=bbb, D8376=ccc, D8377=ddd

Connection 8 Connected IP Address: For aaa.bbb.ccc.ddd
D8378=aaa, D8379=bbb, D8380=ccc, D8381=ddd
Connection 9 Connected IP Address: For aaa.bbb.ccc.ddd
D8677=aaa, D8678=bbb, D8679=ccc, D8680=ddd
Connection 10 Connected IP Address: For aaa.bbb.ccc.ddd
D8681=aaa, D8682=bbb, D8683=ccc, D8684=ddd
Connection 11 Connected IP Address: For aaa.bbb.ccc.ddd
D8685=aaa, D8686=bbb, D8687=ccc, D8688=ddd
Connection 12 Connected IP Address: For aaa.bbb.ccc.ddd
D8689=aaa, D8690=bbb, D8691=ccc, D8692=ddd
Connection 13 Connected IP Address: For aaa.bbb.ccc.ddd
D8693=aaa, D8694=bbb, D8695=ccc, D8696=ddd
Connection 14 Connected IP Address: For aaa.bbb.ccc.ddd
D8697=aaa, D8698=bbb, D8699=ccc, D8700=ddd
Connection 15 Connected IP Address: For aaa.bbb.ccc.ddd
D8701=aaa, D8702=bbb, D8703=ccc, D8704=ddd
Connection 16 Connected IP Address: For aaa.bbb.ccc.ddd
D8705=aaa, D8706=bbb, D8707=ccc, D8708=ddd

■ **D8382 to D8387: HMI Module MAC Address (Current Value Read-only)**

The MAC address is written to the special data registers as follows.
Example: MAC address: AA-BB-CC-DD-EE-FF
D8382=AA, D8383=BB, D8384=CC, D8385=DD, D8386=EE, D8387=FF

■ **D8388 to D8391: HMI Module IP Address (Current Value Read-only)**

The HMI module's IP address is written to the special data registers as follows.
Example: HMI module IP address: aaa.bbb.ccc.ddd
D8388=aaa, D8389=bbb, D8390=ccc, D8391=ddd

■ **D8392 to D8395: HMI Module Subnet Mask (Current Value Read-only)**

The HMI module's subnet mask value is written to the special data registers as follows.
Example: HMI module subnet mask: aaa.bbb.ccc.ddd
D8392=aaa, D8393=bbb, D8394=ccc, D8395=ddd

■ **D8396 to D8399: HMI Module Default Gateway (Current Value Read-only)**

The HMI module's default gateway address is written to the special data registers as follows.
Example: HMI module default gateway: aaa.bbb.ccc.ddd
D8396=aaa, D8397=bbb, D8398=ccc, D8399=ddd

■ **D8400 to D8403: HMI Module Preferred DNS Server (Current Value Read-only)**

The HMI module's preferred DNS server address is written to the special data registers as follows.
Example: HMI module preferred DNS server: aaa.bbb.ccc.ddd
D8400=aaa, D8401=bbb, D8402=ccc, D8403=ddd

■ D8404 to D8407: HMI Module Alternate DNS Server (Current Value Read-only)

The HMI module's alternate DNS server address is written to the special data registers as follows.
Example: HMI module alternate DNS server: aaa.bbb.ccc.ddd
D8404=aaa, D8405=bbb, D8406=ccc, D8407=ddd

■ D8413: Time Zone Offset

The time zone configured in the function area settings can be finely adjusted in 15-minute increments.
For details, see "SNTP Settings" on page 3-11.

■ D8414: SNTP Operation Status

The operation status is written to this register when the time information is acquired through operation of M8191 (SNTP Time Acquisition Flag) or when time information acquisition was executed by automatic acquisition.
For details, see "SNTP Settings" on page 3-11.

■ D8415: SNTP Access Elapsed Time

The elapsed time in minutes from when the time information was last acquired from the SNTP server.
For details, see "SNTP Settings" on page 3-11.

■ D8429: HMI Module Connection Information Reference Connection Number

The connection information for the specified connection number is reflected in D8430 to D8434. When 0 is written, 0 is written to D8430 to D8434. If a connection number that does not exist is specified, 0 is written to D8430 to D8434.

■ D8430 to D8433: HMI Module Connection Information Reference Connected IP Address

The IP address of the terminal being accessed through the connection is written to the special data registers as follows.
Example: IP address to read: aaa.bbb.ccc.ddd
D8430=aaa, D8431=bbb, D8432=ccc, D8433=ddd

■ D8434: HMI Module Connection Information Reference Connected Port No.

The port number of the terminal being accessed through the connection is written to this register.

■ D8437 to D8440: HMI Module IP Address (Write-only)

These registers are used to write the HMI module's IP address.
HMI module IP address: To set as aaa.bbb.ccc.ddd, write the following.
D8437=aaa, D8438=bbb, D8439=ccc, D8440=ddd

■ D8441 to D8444: HMI Module Subnet Mask (Write-only)

These registers are used to write the HMI module's subnet mask.
HMI module subnet mask: For aaa.bbb.ccc.ddd, write the following.
D8441=aaa, D8442=bbb, D8443=ccc, D8444=ddd

■ D8445 to D8448: HMI Module Default Gateway (Write-only)

These registers are used to write the HMI module's default gateway.
HMI module default gateway: For aaa.bbb.ccc.ddd, write the following.
D8445=aaa, D8446=bbb, D8447=ccc, D8448=ddd

■ D8449 to D8452: HMI Module Preferred DNS Server (Write-only)

These registers are used to write the HMI module's preferred DNS server.
HMI module preferred DNS server: For aaa.bbb.ccc.ddd, write the following.
D8449=aaa, D8450=bbb, D8451=ccc, D8452=ddd

■ D8453 to D8456: HMI Module Alternate DNS Server (Write-only)

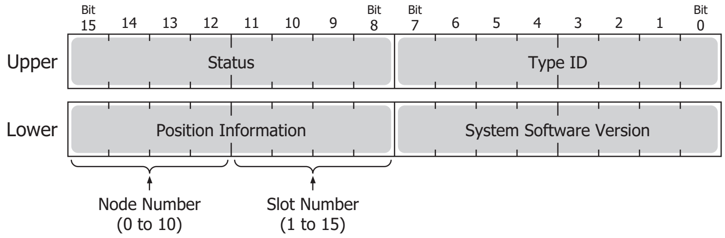
These registers are used to write the HMI module's alternate DNS server.
HMI module alternate DNS server: For aaa.bbb.ccc.ddd, write the following.
D8453=aaa, D8454=bbb, D8455=ccc, D8456=ddd

■ D8457: EMAIL Instruction Detailed Error Information (HMI-Ethernet Port)

Detailed error information for the EMAIL instruction is written to this register.
For details, see "Confirm the error detail of EMAIL instruction" on page 12-3.

■ D8470 to D8595: Expansion Module Slot Information

Expansion module type information is written to these registers.
The allocation of information (bit assignment) is as follows.



Upper	Lower	
D8470,	D8471	= Expansion Module Slot 1 Information
D8472,	D8473	= Expansion Module Slot 2 Information
D8474,	D8475	= Expansion Module Slot 3 Information
D8476,	D8477	= Expansion Module Slot 4 Information
D8478,	D8479	= Expansion Module Slot 5 Information
D8480,	D8481	= Expansion Module Slot 6 Information
D8482,	D8483	= Expansion Module Slot 7 Information
D8484,	D8485	= Expansion Module Slot 8 Information
D8486,	D8487	= Expansion Module Slot 9 Information
D8488,	D8489	= Expansion Module Slot 10 Information
D8490,	D8491	= Expansion Module Slot 11 Information
D8492,	D8493	= Expansion Module Slot 12 Information
D8494,	D8495	= Expansion Module Slot 13 Information
D8496,	D8497	= Expansion Module Slot 14 Information
D8498,	D8499	= Expansion Module Slot 15 Information
D8500,	D8501	= Expansion Module Slot 16 Information
D8502,	D8503	= Expansion Module Slot 17 Information
D8504,	D8505	= Expansion Module Slot 18 Information
D8506,	D8507	= Expansion Module Slot 19 Information
D8508,	D8509	= Expansion Module Slot 20 Information
D8510,	D8511	= Expansion Module Slot 21 Information
D8512,	D8513	= Expansion Module Slot 22 Information
D8514,	D8515	= Expansion Module Slot 23 Information
D8516,	D8517	= Expansion Module Slot 24 Information
D8518,	D8519	= Expansion Module Slot 25 Information
D8520,	D8521	= Expansion Module Slot 26 Information
D8522,	D8523	= Expansion Module Slot 27 Information
D8524,	D8525	= Expansion Module Slot 28 Information
D8526,	D8527	= Expansion Module Slot 29 Information
D8528,	D8529	= Expansion Module Slot 30 Information
D8530,	D8531	= Expansion Module Slot 31 Information
D8532,	D8533	= Expansion Module Slot 32 Information
D8534,	D8535	= Expansion Module Slot 33 Information
D8536,	D8537	= Expansion Module Slot 34 Information
D8538,	D8539	= Expansion Module Slot 35 Information
D8540,	D8541	= Expansion Module Slot 36 Information
D8542,	D8543	= Expansion Module Slot 37 Information
D8544,	D8545	= Expansion Module Slot 38 Information
D8546,	D8547	= Expansion Module Slot 39 Information

Upper Lower
D8548, D8549 = Expansion Module Slot 40 Information
D8550, D8551 = Expansion Module Slot 41 Information
D8552, D8553 = Expansion Module Slot 42 Information
D8554, D8555 = Expansion Module Slot 43 Information
D8556, D8557 = Expansion Module Slot 44 Information
D8558, D8559 = Expansion Module Slot 45 Information
D8560, D8561 = Expansion Module Slot 46 Information
D8562, D8563 = Expansion Module Slot 47 Information
D8564, D8565 = Expansion Module Slot 48 Information
D8566, D8567 = Expansion Module Slot 49 Information
D8568, D8569 = Expansion Module Slot 50 Information
D8570, D8571 = Expansion Module Slot 51 Information
D8572, D8573 = Expansion Module Slot 52 Information
D8574, D8575 = Expansion Module Slot 53 Information
D8576, D8577 = Expansion Module Slot 54 Information
D8578, D8579 = Expansion Module Slot 55 Information
D8580, D8581 = Expansion Module Slot 56 Information
D8582, D8583 = Expansion Module Slot 57 Information
D8584, D8585 = Expansion Module Slot 58 Information
D8586, D8587 = Expansion Module Slot 59 Information
D8588, D8589 = Expansion Module Slot 60 Information
D8590, D8591 = Expansion Module Slot 61 Information
D8592, D8593 = Expansion Module Slot 62 Information
D8594, D8595 = Expansion Module Slot 63 Information

The system software version displays the version of the software written to the device.
For details on the type IDs and statuses, see "List of Type IDs and Status" on page 2-47.

■ **D8596, D8597: Expansion Interface Remote Master Module Slot Information**

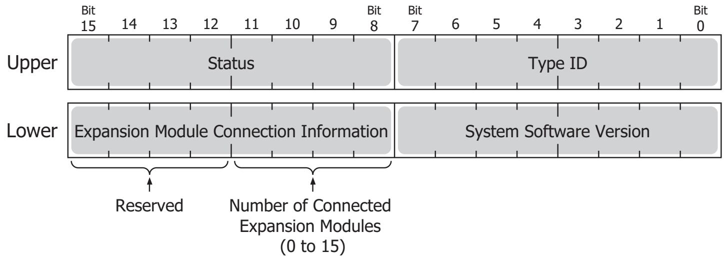
Expansion interface remote master module type information is written to these registers.
The allocation of information (bit assignment) is as follows.

	Bit 15 to Bit 8 (15 14 13 12 11 10 9 8)	Bit 7 to Bit 0 (7 6 5 4 3 2 1 0)
Upper	Status	Type ID
Lower	Number of Connected Expansion Interface Remote Slave Modules	System Software Version

The system software version displays the version of the software written to the device.
For details on the type IDs and statuses, see "List of Type IDs and Status" on page 2-47.

■ **D8598 to D8617: Expansion Interface Remote Slave Module Slot Information**

Expansion interface remote slave module type information is written to these registers.
The allocation of information (bit assignment) is as follows.



Upper	Lower	
D8598,	D8599	= Expansion Interface Remote Slave Module (Unit 1) Slot Information
D8600,	D8601	= Expansion Interface Remote Slave Module (Unit 2) Slot Information
D8602,	D8603	= Expansion Interface Remote Slave Module (Unit 3) Slot Information
D8604,	D8605	= Expansion Interface Remote Slave Module (Unit 4) Slot Information
D8606,	D8607	= Expansion Interface Remote Slave Module (Unit 5) Slot Information
D8608,	D8609	= Expansion Interface Remote Slave Module (Unit 6) Slot Information
D8610,	D8611	= Expansion Interface Remote Slave Module (Unit 7) Slot Information
D8612,	D8613	= Expansion Interface Remote Slave Module (Unit 8) Slot Information
D8614,	D8615	= Expansion Interface Remote Slave Module (Unit 9) Slot Information
D8616,	D8617	= Expansion Interface Remote Slave Module (Unit 10) Slot Information

The system software version displays the version of the software written to the device.
For details on the type ID and status, see "List of Type IDs and Status" on page 2-47.

■ D8618: Refresh Time of Expansion Interface Remote Master/Slave Modules Current Value

The current value of the I/O refresh time of the expansion modules connected with the expansion interface remote master/slave modules is stored in increments of 1 ms.

■ D8619: Refresh Time of Expansion Interface Remote Master/Slave Modules Maximum Value

The maximum value of the I/O refresh time of the expansion modules connected with the expansion interface remote master/slave modules is stored in increments of 1 ms.

■ D8630: CPU Module Ethernet Port 2 IP Settings/DNS Settings Switching

The IP settings/DNS settings for Ethernet Port 2 can be changed by writing one of the setting values in the following table to D8630 and then turning on M8333.
To use this function, enable **Enable D8630 (IP Settings / DNS Settings switching)** on **Ethernet Port 2** in **Function Area Settings** of WindLDR.
The meanings of the setting values are as follows.

Setting Value	IP Settings/DNS Settings
0	Conform to function area settings.
1	Enable DHCP.
2	Conform to special data register (D8631 to D8650) settings.

■ D8631 to D8634: CPU Module Ethernet Port 2 IP Address (Write-only)

These registers are used to write the IP address of Ethernet port 2 on the Plus CPU module.

IP address: To set as aaa.bbb.ccc.ddd, write the following.

D8631 = aaa, D8632 = bbb, D8633 = ccc, D8634 = ddd

■ D8635 to D8638: CPU Module Ethernet Port 2 Subnet Mask (Write-only)

These registers are used to write the subnet mask of Ethernet port 2 on the Plus CPU module.

Subnet mask: For aaa.bbb.ccc.ddd, write the following.

D8635 = aaa, D8636 = bbb, D8637 = ccc, D8638 = ddd

■ D8639 to D8642: CPU Module Ethernet Port 2 Default Gateway (Write-only)

These registers are used to write the default gateway of Ethernet port 2 on the Plus CPU module.

Default gateway: For aaa.bbb.ccc.ddd, write the following.

D8639 = aaa, D8640 = bbb, D8641 = ccc, D8642 = ddd

■ **D8643 to D8646: CPU Module Ethernet Port 2 Preferred DNS Server (Write-only)**

These registers are used to write the preferred DNS server of Ethernet port 2 on the Plus CPU module.

Preferred DNS server: For aaa.bbb.ccc.ddd, write the following.

D8643 = aaa, D8644 = bbb, D8645 = ccc, D8646 = ddd

■ **D8647 to D8650: CPU Module Ethernet Port 2 Alternate DNS Server (Write-only)**

These registers are used to write the alternate DNS server of Ethernet port 2 on the Plus CPU module.

Alternate DNS server: For aaa.bbb.ccc.ddd, write the following.

D8647 = aaa, D8648 = bbb, D8649 = ccc, D8650 = ddd

■ **D8651 to D8656: CPU Module Ethernet Port 2 MAC Address (Current Value Read-only)**

The MAC address of Ethernet port 2 on the Plus CPU module is written to the special data registers as follows.

Example: MAC address: AA-BB-CC-DD-EE-FF

D8651 = AA, D8652 = BB, D8653 = CC, D8654 = DD, D8655 = EE, D8656 = FF

■ **D8657 to D8660: CPU Module Ethernet Port 2 IP Address (Current Value Read-only)**

The IP address of Ethernet port 2 on the Plus CPU module is written to the special data registers as follows.

Example: The CPU module's own IP address: aaa.bbb.ccc.ddd

D8657 = aaa, D8658 = bbb, D8659 = ccc, D8660 = ddd

■ **D8661 to D8664: CPU Module Ethernet Port 2 Subnet Mask (Current Value Read-only)**

The subnet mask value of Ethernet port 2 on the Plus CPU module is written to the special data registers as follows.

Example: Subnet mask: aaa.bbb.ccc.ddd

D8661 = aaa, D8662 = bbb, D8663 = ccc, D8664 = ddd

■ **D8665 to D8668: CPU Module Ethernet Port 2 Default Gateway (Current Value Read-only)**

The default gateway address of Ethernet port 2 on the Plus CPU module is written to the special data registers as follows.

Example: Default gateway: aaa.bbb.ccc.ddd

D8665 = aaa, D8666 = bbb, D8667 = ccc, D8668 = ddd

■ **D8669 to D8672: CPU Module Ethernet Port 2 Preferred DNS Server (Current Value Read-only)**

The preferred DNS server address of Ethernet port 2 on the Plus CPU module is written to the special data registers as follows.

Example: Preferred DNS server: aaa.bbb.ccc.ddd

D8669 = aaa, D8670 = bbb, D8671 = ccc, D8672 = ddd

■ **D8673 to D8676: CPU Module Ethernet Port 2 Alternate DNS Server (Current Value Read-only)**

The alternate DNS server address of Ethernet port 2 on the Plus CPU module is written to the special data registers as follows.

Example: Alternate DNS server: aaa.bbb.ccc.ddd

D8673 = aaa, D8674 = bbb, D8675 = ccc, D8676 = ddd

■ **D8759: EMAIL Instruction Detailed Error Information (Ethernet Port 1)**

Detailed error information for the EMAIL instruction (Ethernet port 1) is written to this register.

For details, see "Confirm the error detail of EMAIL instruction" on page 12-3.

■ **D8782: BACnet Operation Status**

The operating status of BACnet communication is written to this special data register.

For details, see "Special Devices Used in BACnet/IP" on page 15-11.

■ **D8783: BACnet Error Information**

Information for the error that last occurred in BACnet communication is written to this special data register.

For details, see "Special Devices Used in BACnet/IP" on page 15-11.

■ **D8790: EtherNet/IP Operation Status**

The operating status of EtherNet/IP communication is written to this special data register.

For details, see "Special Devices Used in EtherNet/IP Communication" on page 16-13.

■ **D8791: BACnet Error Information**

Information for the error that last occurred in EtherNet/IP communication is written to this special data register.

For details, see "Special Devices Used in EtherNet/IP Communication" on page 16-13.

List of Type IDs and Status

■ Type ID

Expansion module and HMI module

Type ID		Type No.
Hexadecimal	Binary	
0x00	0000 0000	FC6A-N16B1, FC6A-N16B4, FC6A-N16B3
0x01	0000 0001	FC6A-R161, FC6A-R164, FC6A-T16K1, FC6A-T16K4, F6A-T16P1, FC6A-T16P4, F6A-T16K3, FCA-T16P3
0x02	0000 0010	FC6A-N32B3
0x03	0000 0011	FC6A-T32K3, FC6A-T32P3
0x04	0000 0100	FC6A-N08B1, FC6A-N08B4, FC6A-N08A11, FC6A-N08A14
0x05	0000 0101	FC6A-R081, FC6A-R084, FC6A-T08K1, FC6A-T08K4, FC6A-T08P1, FC6A-T08P4
0x06	0000 0110	FC6A-M08BR1, FC6A-M08BR4
0x07	0000 0111	FC6A-M24BR1, FC6A-M24BR4
0x18	0001 1000	FC6A-PH1
0x19	0001 1001	FC6A-EXM2, FC6A-EXM24
0x1A	0001 1010	FC6A-EXM1S, FC6A-EXM1S4
0x20	0010 0000	FC6A-J2C1, FC6A-J2C4
0x21	0010 0001	FC6A-J4A1, FC6A-J4A4
0x22	0010 0010	FC6A-J8A1, FC6A-J8A4
0x23	0010 0011	FC6A-K2A1, FC6A-K2A4
0x24	0010 0100	FC6A-K4A1, FC6A-K4A4
0x25	0010 0101	FC6A-L06A1, FC6A-L06A4
0x26	0010 0110	FC6A-L03CN1, FC6A-L03CN4
0x27	0010 0111	FC6A-J4CN1, FC6A-J4CN4
0x28	0010 1000	FC6A-J8CU1, FC6A-J8CU4
0x29	0010 1001	FC6A-F2M1, FC6A-F2M4
0x2A	0010 1010	FC6A-F2MR1, FC6A-F2MR4
0x2B	0010 1011	FC6A-J4CH1Y, FC6A-J4CH4Y
0x2C	0010 1100	FC6A-EXM1M
0x2E	0010 1110	FC6A-SIF52, FC6A-SIF524
0xFF	1111 1111	Not connected

Cartridge

Type ID		Type No.
Hexadecimal	Binary	
0x00	0000 0000	FC6A-PJ2A
0x01	0000 0001	FC6A-PK2AV
0x02	0000 0010	FC6A-PK2AW
0x03	0000 0011	FC6A-PJ2CP
0x06	0000 0110	FC6A-PC1
0x07	0000 0111	FC6A-PC3
0x09	0000 1001	FC6A-PTS4, FC6A-PTK4
0x0A	0000 1010	FC6A-PN4
0x0C	0000 1100	FC6A-PC4
0xFF	1111 1111	Not connected

■ Status

Status		Description
Hexadecimal	Binary	
0x00	0000 0000	Normal
0x81	1000 0001	Communication error (An error has occurred in the communication between the CPU module and an expansion module, HMI module, or cartridge.)
0x82	1000 0010	Unknown device detected (A device other than the FC6A Series MICROSmart is connected.)
0x83	1000 0011	Device setting error (No device is connected or the connected device is different from that set in the user program.)
0x84	1000 0100	Device writing error (The attempt to set the operation of the device failed.)
0x85	1000 0101	System update error (The system update failed.)
0x86	1000 0110	Expansion interface remote master module communication error (An error is occurring in the communication between the expansion interface remote master and slave modules.)

3: COMMUNICATION SETTINGS

Introduction

This chapter describes how to configure the settings to use the communication functions of the FC6A Series MICROSmart and examples of using these functions.
For functions that require advanced setup including the communication ports and network settings, first configure those settings in the **Function Area Settings** dialog box in WindLDR, and then download the user program to the FC6A Series MICROSmart.
For the way to configure settings related to CAN J1939 communication, see "J1939 Communication" on page 8-1.

Setting List

Function Name	Overview	Reference	Setup Location
Communication ports	Configures the communication function, parameters and the ports to match the communication device.	3-2	Function area settings
Network settings	Configures information for connecting the FC6A Series MICROSmart to a network.	3-3	
Network Management	Configures SNTP settings and the PING instruction timeout time.	3-11	
Connection settings	Communication mode and parameters for the Ethernet communication can be configured for each connection so that the FC6A Series MICROSmart can communicate with other network devices over the Ethernet.	3-14	
Remote host list	The remote host devices on the network that the FC6A Series MICROSmart communicates with can be registered and managed in the Remote Host List.	3-20	Remote host list
Auto Ping Function	Configures the auto ping function to monitor the network connection status of remote hosts.	3-22	

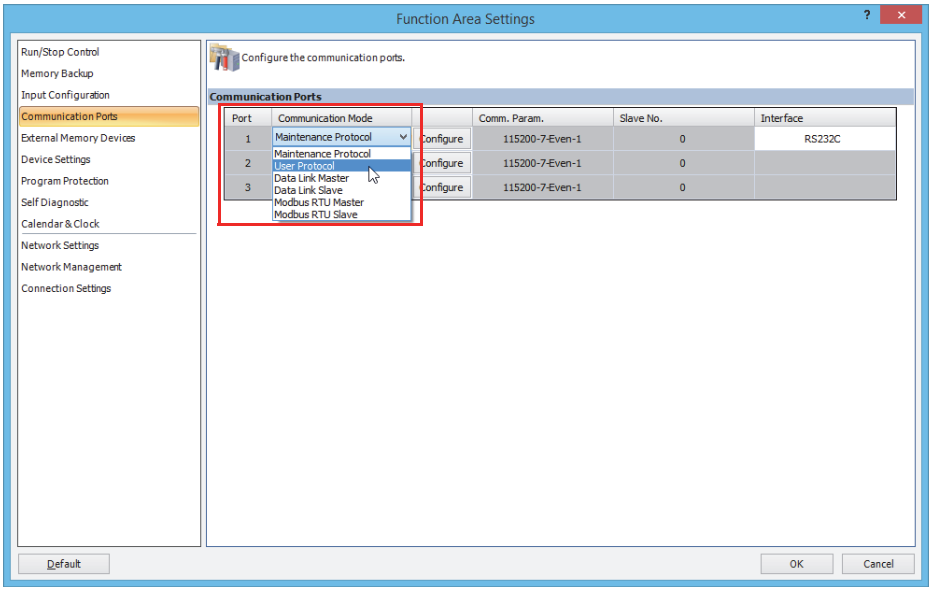
Communication Port Settings

This section describes how to configure the communication ports when using communication port 1 to 33 to communicate with communication devices.

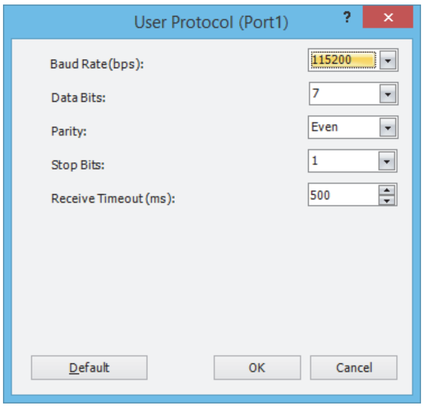
Programming WindLDR

Configure the communication format according to the communication specifications of the device.

1. From the WindLDR menu bar, select **Configuration** > **Communication Ports**.
   The **Function Area Settings** dialog box is displayed.
2. Click **Communication Mode** for the port to use and select the communication mode to use.
   The configuration dialog that corresponds to the communication mode is displayed.



3. Change the settings on the configuration dialog according to the communication format for the destination device.
   A user communication example is shown below.



4. Click **OK**.
   This concludes configuring the settings.

Network Settings

This section describes the settings for using Ethernet port 1, Ethernet port 2, or the HMI-Ethernet port to connect the FC6A Series MICROSmart to a network.

Description

The following table presents the Ethernet ports and configuration items that can be configured in the FC6A Series MICROSmart network settings in order for the FC6A Series MICROSmart to connect to an Ethernet network.

FC6A Series MICROSmart	Ethernet Port	Settings								
		IP Settings	DNS Settings	SNTP Settings	Ping Settings	E-mail Settings	Web Server Settings	BACnet Settings	FTP Settings	EtherNet/IP Settings
All-in-One CPU module	Ethernet port 1	Yes	Yes	Yes	Yes	—	—	—	—	—
Plus CPU module	Ethernet port 1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	—
	Ethernet port 2	Yes	Yes	—	Yes	—	—	—	—	Yes
HMI module	HMI-Ethernet port	Yes	Yes	—	—	Yes	Yes	—	—	—

Network Settings

IP Settings

The specification method for the FC6A Series MICROSmart IP address, subnet mask, and default gateway can be selected as the following three types.

Specification Method	Description
Obtain an IP Address Automatically (DHCP)	The FC6A Series MICROSmart will automatically acquire the IP address, subnet mask, and default gateway from the DHCP server when the user program download has finished and when the FC6A Series MICROSmart power is turned on. However, a DHCP server must be present on the network where the FC6A Series MICROSmart is located.
Use Special Data Register to Configure the IP Address	Specify the IP address, subnet mask, and default gateway of Ethernet port 1, Ethernet port 2, and HMI-Ethernet port using the special data registers. These values are set on the FC6A Series MICROSmart when the FC6A Series MICROSmart power is turned on and when special internal relay M8190, M8333, or M8184 is turned on.
Use the Following IP Address	Specify a fixed IP address, subnet mask, and default gateway. The values specified here are reflected in the FC6A Series MICROSmart when the user program download has finished.

Notes:

- The default FC6A Series MICROSmart settings are: IP address 192.168.1.5, subnet mask 255.255.255.0, default gateway 0.0.0.0.
- The FC6A Series MICROSmart IP address, subnet mask, and default gateway can be changed using the HMI module. For details, see Chapter 7 "HMI Function" in the "FC6A Series MICROSmart User's Manual".

DNS Settings

The specification method for the DNS server addresses can be selected as the following three types.

Specification Method	Description
Obtain DNS Server Address Automatically (DHCP)	The FC6A Series MICROSmart will automatically acquire the DNS server addresses from the DHCP server when the user program download has finished and when the FC6A Series MICROSmart power is turned on. However, a DHCP server must be present on the network where the FC6A Series MICROSmart is located.
Use Special Data Registers to Configure the DNS Server Addresses	Specify the preferred DNS server address and the alternate DNS server address of Ethernet port 1, Ethernet port 2, and HMI-Ethernet port using the special data registers. These values are set on the FC6A Series MICROSmart when the FC6A Series MICROSmart power is turned on and when special internal relay M8190, M8333, or M8184 is turned on.
Use the Following DNS Server Addresses	Specify a fixed preferred DNS server address and alternate DNS server address. The values specified here are reflected in the FC6A Series MICROSmart when the user program download has finished.

Notes:

- When an IP address cannot be obtained from the preferred DNS server, the alternate DNS server is accessed.
- The DNS server addresses can be changed using the HMI module. For details, see Chapter 7 "HMI Function" in the "FC6A Series MICROSmart User's Manual".

Network settings by special data registers

When configuring the IP Settings and DNS server addresses using special data registers, values stored in the special data registers are applied as those settings when the power to the FC6A Series MICROSmart is turned on or when special internal relay M8190 or M3333 is turned on.

Examples:

- When M8190 (Change CPU Module Ethernet Port 1 Network Settings Trigger) is turned on, values stored in D8304 to D8323 are applied as IP settings and DNS server addresses of Ethernet port 1.
- When M8333 (Change CPU Module Ethernet Port 2 Network Settings Trigger) is turned on, values stored in D8631 to D8650 are applied as IP settings and DNS server addresses of Ethernet port 2.

Special Data Register	Description	Read/Write
D8304-D8307	Setting value of IP address set for Ethernet port 1. Example: When the IP address is 192.168.0.1 D8304=192, D8305=168, D8306=0, D8307=1	R/W
D8308-D8311	Setting value of subnet mask set for Ethernet port 1. Example: When the subnet mask is 255.255.255.0 D8308=255, D8309=255, D8310=255, D8311=0	R/W
D8312-D8315	Setting value of default gateway set for Ethernet port 1. Example: When the default gateway is 192.168.0.24 D8312=192, D8313=168, D8314=0, D8315=24	R/W
D8316-D8319	Setting value of preferred DNS server address used by Ethernet port 1. Example: When the preferred DNS address is 192.168.0.100 D8316=192, D8317=168, D8318=0, D8319=100	R/W
D8320-D8323	Setting value of alternate DNS server address used by Ethernet port 1. Example: When the alternate DNS address is 192.168.0.101 D8320=192, D8321=168, D8322=0, D8323=101	R/W
D8631-D8634	Setting value of IP address set for Ethernet port 2. Example: When the IP address is 192.168.1.1 D8631=192, D8632=168, D8633=1, D8634=1	R/W
D8635-D8638	Setting value of subnet mask set for Ethernet port 2. Example: When the subnet mask is 255.255.255.0 D8635=255, D8636=255, D8637=255, D8638=0	R/W
D8639-D8642	Setting value of default gateway set for Ethernet port 2. Example: When the default gateway is 192.168.1.24 D8639=192, D8640=168, D8641=1, D8642=24	R/W
D8643-D8646	Setting value of preferred DNS server address used by Ethernet port 2. Example: When the preferred DNS address is 192.168.1.100 D8643=192, D8644=168, D8645=1, D8646=100	R/W
D8647-D8650	Setting value of alternate DNS server address used by Ethernet port 2. Example: When the alternate DNS address is 192.168.1.101 D8647=192, D8648=168, D8649=1, D8650=101	R/W

Checking the network settings

The current IP address, subnet mask, default gateway, and DNS server addresses of Ethernet port 1 and Ethernet port 2 of the FC6A Series MICROSmart can be checked with special data registers.

Special Data Register	Description	Read/Write
D8330-D8333	Current value of IP address used by Ethernet port 1. Example: When the IP address is 192.168.0.1 D8330=192, D8331=168, D8332=0, D8333=1	R
D8334-D8337	Current value of subnet mask used by Ethernet port 1. Example: When the subnet mask is 255.255.255.0 D8334=255, D8335=255, D8336=255, D8337=0	R
D8338-D8341	Current value of default gateway used by Ethernet port 1. Example: When the default gateway is 192.168.0.24 D8338=192, D8339=168, D8340=0, D8341=24	R
D8342-D8345	Current value of preferred DNS server address used by Ethernet port 1. Example: When the preferred DNS address is 192.168.0.100 D8342=192, D8343=168, D8344=0, D8345=100	R
D8346-D8349	Current value of alternate DNS server address used by Ethernet port 1. Example: When the alternate DNS address is 192.168.0.101 D8346=192, D8347=168, D8348=0, D8349=101	R
D8657-D8660	Current value of IP address used by Ethernet port 2. Example: When the IP address is 192.168.1.1 D8657=192, D8658=168, D8659=1, D8660=1	R
D8661-D8664	Current value of subnet mask used by Ethernet port 2. Example: When the subnet mask is 255.255.255.0 D8661=255, D8662=255, D8663=255, D8664=0	R
D8665-D8668	Current value of default gateway used by Ethernet port 2. Example: When the default gateway is 192.168.1.24 D8665=192, D8666=168, D8667=1, D8668=24	R
D8669-D8672	Current value of preferred DNS server address used by Ethernet port 2. Example: When the preferred DNS address is 192.168.1.100 D8669=192, D8670=168, D8671=1, D8672=100	R
D8673-D8676	Current value of alternate DNS server address used by Ethernet port 2. Example: When the alternate DNS address is 192.168.1.101 D8673=192, D8674=168, D8675=1, D8676=101	R

Network settings by HMI module special data registers

When configuring the IP Settings and DNS server addresses using special data registers, values stored in the special data registers are applied as those settings when the power to the FC6A Series MICROSmart is turned on or when special internal relay M8184 is turned on.

Example:

- When M8184 (Change HMI Module Network Settings Trigger) is turned on, values stored in D8437 to D8456 are applied as IP settings and DNS server addresses of HMI-Ethernet port.

Special Data Register	Description	Read/Write
D8437-D8440	The HMI module IP address setting value. Example: When the IP address is 192.168.0.1 D8437=192, D8438=168, D8439=0, D8440=1	R/W
D8441-D8444	The HMI module subnet mask setting value. Example: When the subnet mask is 255.255.255.0 D8441=255, D8442=255, D8443=255, D8444=0	R/W
D8445-D8448	The HMI module default gateway setting value. Example: When the default gateway is 192.168.0.24 D8445=192, D8446=168, D8447=0, D8448=24	R/W
D8449-D8452	The setting value of the preferred DNS server address used by the HMI module. Example: When the preferred DNS address is 192.168.0.100 D8449=192, D8450=168, D8451=0, D8452=100	R/W
D8453-D8456	The setting value of the alternate DNS server address used by the HMI module. Example: When the alternate DNS address is 192.168.0.101 D8453=192, D8454=168, D8455=0, D8456=101	R/W

Checking the HMI module network settings

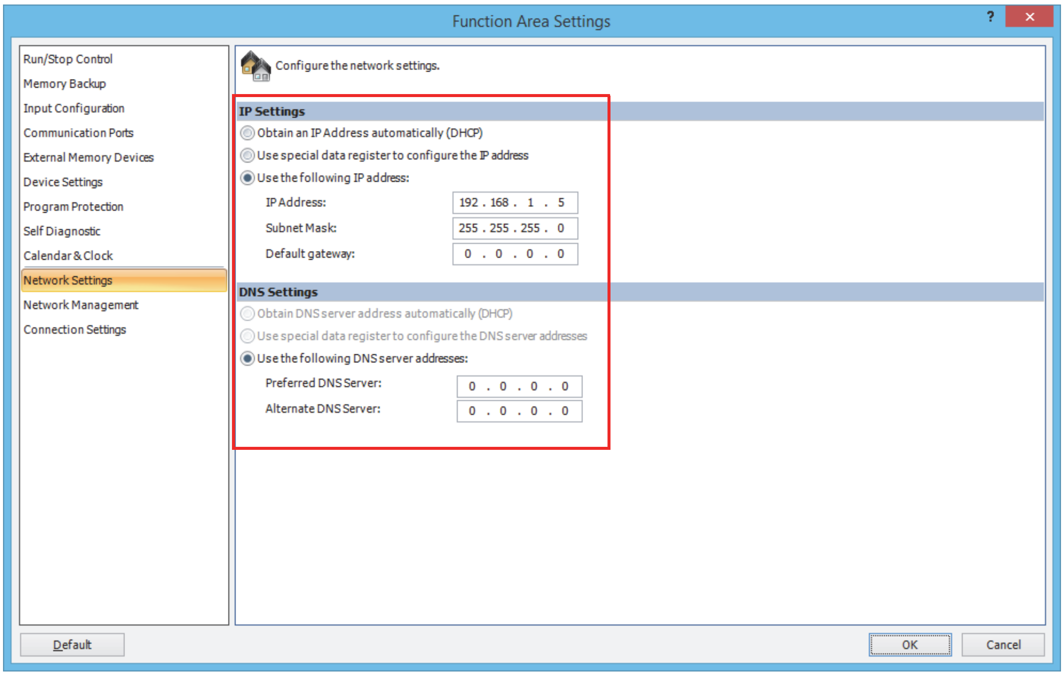
The current HMI module IP address, subnet mask, default gateway, and DNS server addresses can be checked with special data registers.

Special Data Register	Description	Read/Write
D8388-D8391	The HMI module IP address current value. Example: When the IP address is 192.168.0.1 D8388=192, D8389=168, D8390=0, D8391=1	R
D8392-D8395	The HMI module subnet mask current value. Example: When the subnet mask is 255.255.255.0 D8392=255, D8393=255, D8394=255, D8395=0	R
D8396-D8399	The HMI module default gateway current value. Example: When the default gateway is 192.168.0.24 D8396=192, D8397=168, D8398=0, D8399=24	R
D8400-D8403	The current value of the preferred DNS server address used by the HMI module. Example: When the preferred DNS address is 192.168.0.100 D8400=192, D8401=168, D8402=0, D8403=100	R
D8404-D8407	The current value of the alternate DNS server address used by the HMI module. Example: When the alternate DNS address is 192.168.0.101 D8404=192, D8405=168, D8406=0, D8407=101	R

Programming WindLDR

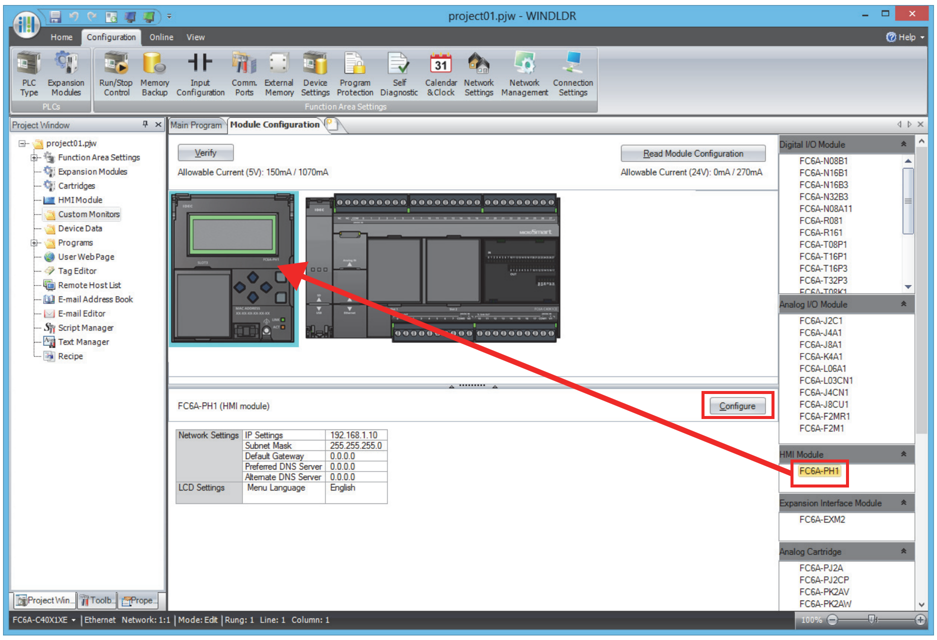
1. On the **Configuration** tab, in the **Function Area Settings** group, click the following item.
   - All-in-One CPU module/CAN J1939 All-in-One CPU module
     Network Settings
   - Plus CPU module
     Ethernet Port 1 to configure Ethernet port 1
     Ethernet Port 2 to configure Ethernet port 2

   The **Function Area Settings** dialog box appears.
2. Configure **IP Settings** and **DNS Settings**.

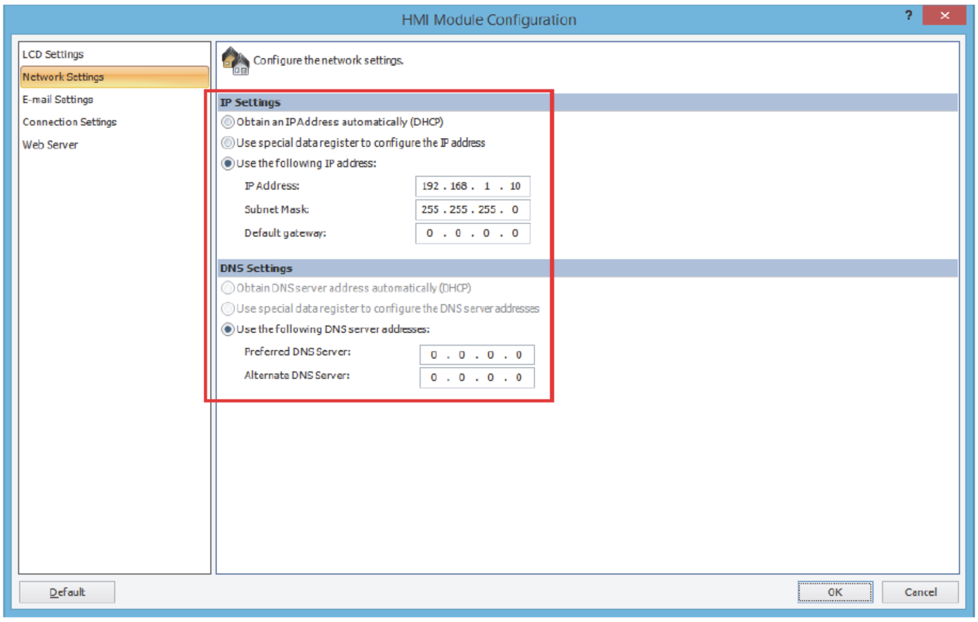


3. Click **OK**.
4. Use the Module Configuration Editor to configure the HMI module network settings.
   On the **Configuration** tab, in the **PLCs** group, select **Expansion Modules**.
5. Click the inserted HMI module in the module configuration area and click **Configure**.
   The **HMI Module Configuration** dialog box is displayed.

Note: You can also display the **HMI Module Configuration** dialog box by double-clicking **HMI Module** in the Project Window.



6. Click the **Network Settings** tab, and then configure the **IP Settings** and **DNS Settings**.



This concludes configuring the settings.

IP Settings and DNS Settings Switching

The IP settings and DNS settings for Ethernet port 1 and Ethernet port 2 of the CPU module can be switched by using special data registers (D8303 and D8630) and special internal relays (M8190 and M8333). The switching procedure is as follows.

Programming WindLDR

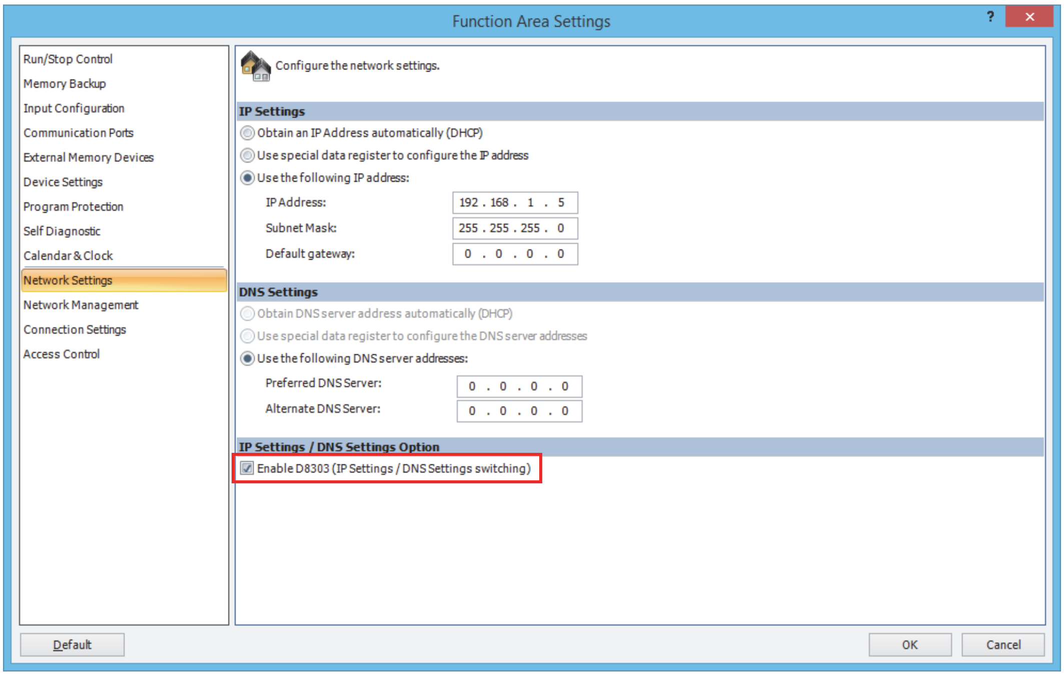
1. On the **Configuration** tab, in the **Function Area Settings** group, click the following item.
   - All-in-One CPU module/CAN J1939 All-in-One CPU module
     Network Settings
   - Plus CPU module
     Ethernet Port 1 to configure Ethernet port 1
     Ethernet Port 2 to configure Ethernet port 2

   The **Function Area Settings** dialog box appears.

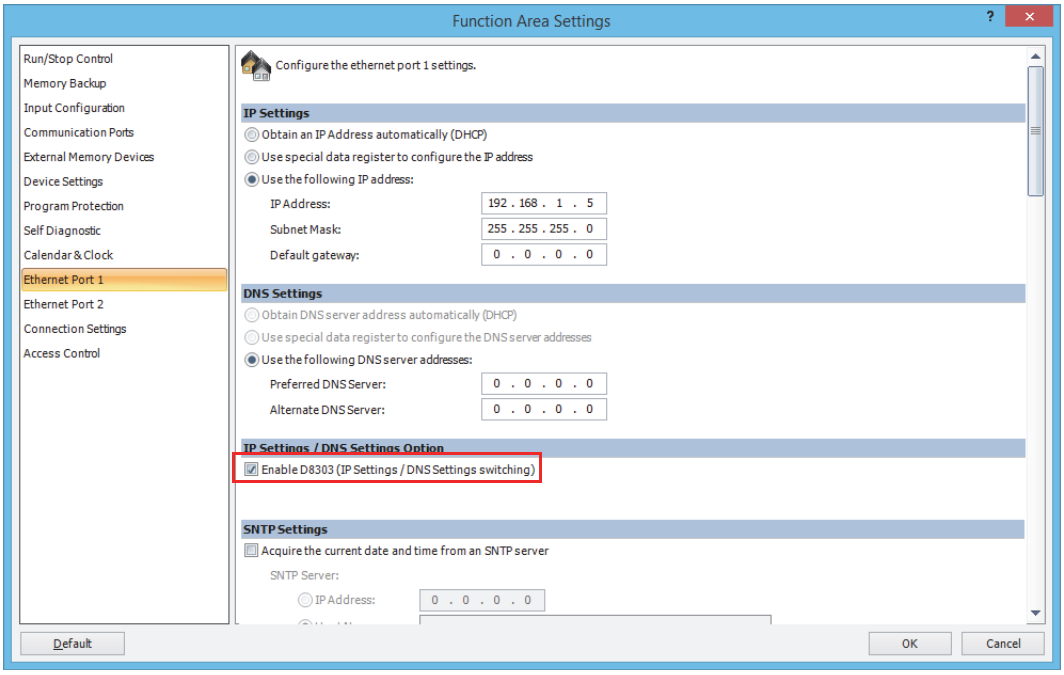
2. Configure **IP Settings** and **DNS Settings**.

3. To set Ethernet port 1, select the **Enable D8303 (IP Settings / DNS Settings switching)** check box.

- All-in-One CPU module/CAN J1939 All-in-One CPU module

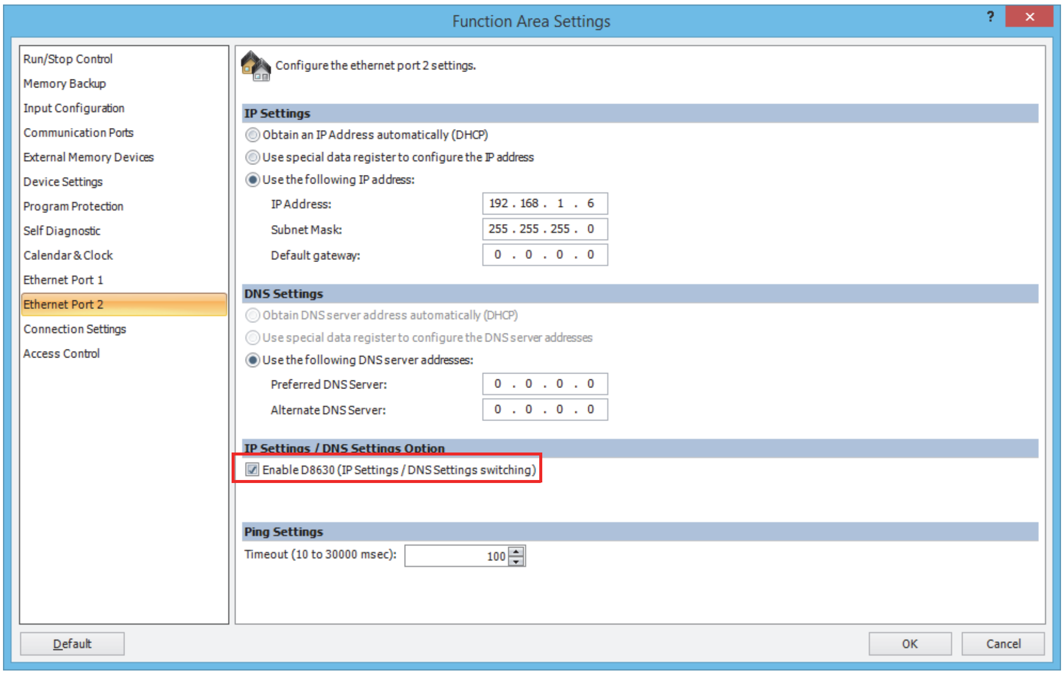


- Plus CPU module



4. To set Ethernet port 2, select the **Enable D8630 (IP Settings / DNS Settings switching)** check box.

- Plus CPU module



5. Click **OK**.

This concludes configuring the settings.

When the user program is downloaded, the IP settings and DNS settings are switched according to the setting value of D8303 or D8630.
The meanings of the setting values of D8303 and D8630 are as follows.

Meanings of the values stored in D8303 or D8630

Setting Value	IP Settings/DNS Settings
0	Conform to function area settings.
1	Enable DHCP.
2	Conform to special data register (D8304 to D8323 or D8631 to D8650) settings.

Notes:

- Regardless of the function area settings, the IP settings and DNS settings can be forcibly set using special data registers by setting the value of D8303 or D8630 to 2 (conform to special data register settings), and then switching M8190 or M8333 from off to on or turning the power supply off and then on again.
- Regardless of the function area settings, DHCP can be forcibly set by setting the value of D8303 or D8630 to 1 (enable DHCP), and then switching M8190 or M8333 from off to on or turning the power supply off and then on again.
- To restore the function area settings, set the value of D8303 or D8630 to 0, and then switch M8190 or M8333 from off to on or turn the power supply off and then on again.

Network Management

SNTP Settings

This section describes the settings to adjust the clock in the FC6A Series MICROSmart by acquiring the current time (GMT) from an SNTP server on the network.

Description

The FC6A Series MICROSmart acquires the current time (GMT) from an SNTP server on the network automatically or at the set interval and adjusts the internal clock according to the time zone setting. If the daylight savings time setting has been enabled, the current time (GMT) acquired from the SNTP server is corrected for daylight savings time during the daylight savings time period. The time zone can be adjusted and the operation status of the SNTP server can be checked using special data registers.

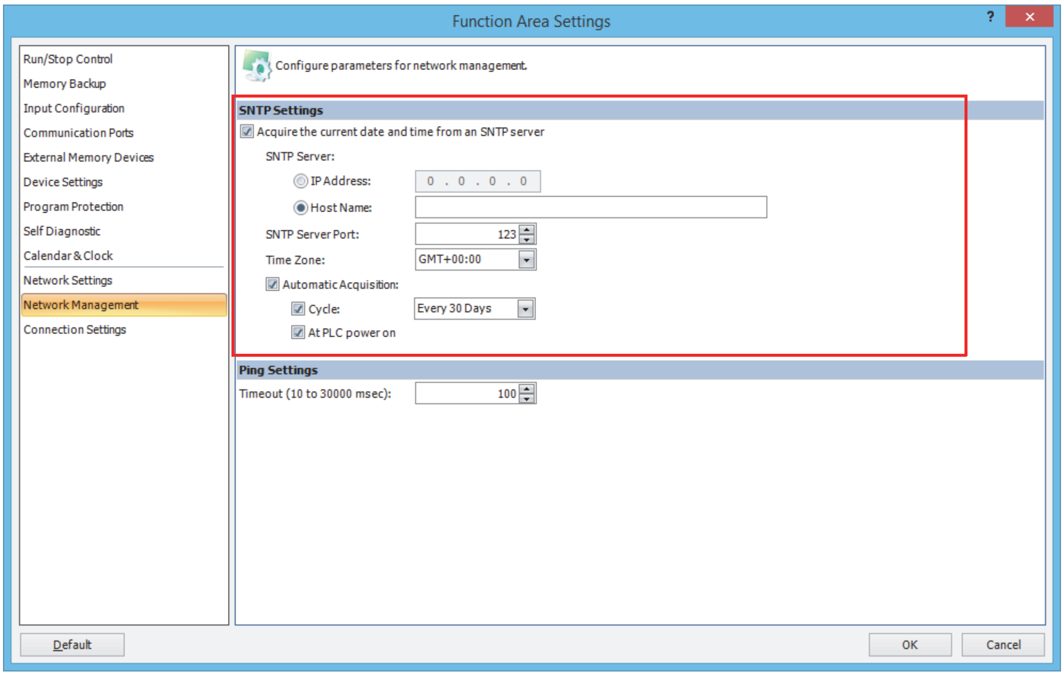
Programming WindLDR

Configure the SNTP server used to acquire the current time and the acquisition method of the current time.

1. On the WindLDR **Configuration** tab, in the **Function Area Settings** group, click the following item.

- All-in-One CPU module/CAN J1939 All-in-One CPU module
  Network Management
- Plus CPU module
  Ethernet Port 1

2. Select the **Acquire the current date and time from an SNTP server** check box.



3. Configure the SNTP server, time zone, and acquisition method of the current time.

Item	Setting Value
SNTP Server (IP Address)	Specify the IP address of the SNTP server used to acquire the current time. The format is "xxx.xxx.xxx.xxx". "xxx" stands for a numeric value from 0 to 255.
SNTP Server (Host Name)	Specify the host name of the SNTP server used to acquire the current time. The host name can be entered up to 40 single-byte alphanumeric characters.
SNTP Server Port	Specify the SNTP server port number (0 to 65535).
Time Zone	Select the difference of the regional time zone in regard to the standard time acquired from the SNTP server. The time zone can be selected in the range of GMT-12:00 to GMT+13:00. The time zone can be adjusted in 15 minute increments using special data register D8413.

4. To automatically acquire the current time, select the **Automatic Acquisition** check box.
When a keep data error occurs and when the user program is downloaded and automatic acquisition is changed from off to on, the current time is automatically acquired from the SNTP server.
To periodically acquire the current time at a timing other than the above, select the **Cycle** check box and select the cycle as "Every 10 Minutes", "Every Hour", "Every Day", or "Every 30 Days".
To acquire the current time from the SNTP server when the FC6A Series MICROSmart power is turned on, select the **At PLC power on** check box.
When the **Automatic Acquisition** check box is cleared, if special internal relay M8191 is turned from off to on, the current time is acquired from the SNTP server.

5. Click **OK**.

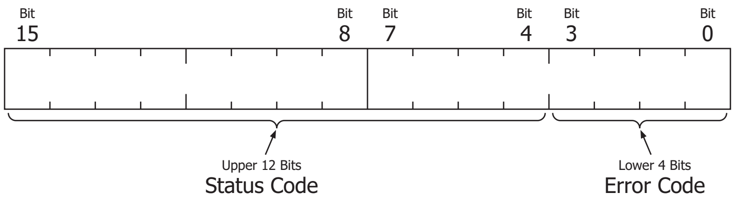
This concludes configuring the settings.

Adjusting the time zone (D8413)
The selected time zone (GMT-12:00 to GMT+13:00) can be adjusted in 15 minute increments with the value of special data register D8413 (time zone offset).

For example, if GMT+09:00 is selected, storing +1 in D8413 advances the time by 15 minutes to make the time zone "GMT+09:15". Storing -2 in D8413 sets the time back by -30 minutes to make the time zone "GMT+08:30".

Checking the operation status (D8414)
The SNTP Operation Status is stored in special data register D8414 (SNTP Operation Status). The operation status indicates the operation status (status code) and the error details (error code).

The operation status (status code) is stored in the upper 12 bits of D8414 and the error details (error code) is stored in the lower 4 bits of D8414.



Status Code	Operation Status	Status Description
0 (000000000000)	No operation	When there is no access to the SNTP server
32 (000000100000)	Waiting for response	When the SNTP server has been accessed and waiting for the response from the SNTP server
64 (000001000000)	Time information acquisition successful	When the response from the SNTP server was normally received

Error Code	Error Details
0 (0000)	Normal
2 (0010)	Timeout error
3 (0011)	The set SNTP server IP address could not be resolved by DNS
9 (1001)	Invalid data was received

Acquire the current time at an arbitrary timing (M8191)
When special internal relay M8191 (SNTP time acquisition flag) is turned on, the current time is acquired from the SNTP server.

Elapsed time since acquiring the current time (D8415)
The elapsed time (0 to 65,535) in minutes since last acquiring the current time from the SNTP server is stored in special data register D8415 (elapsed time since SNTP access). Since the values that can be stored are between 0 and 65,535, 65,535 minutes / 60 minutes / 24 hours = approximately 45 days, so the maximum value that can be measured is 45 days.

For example, when the date and time that the current time was last acquired from the SNTP server is January 1 at 12:00, if the value of D8415 is checked on January 1 at 15:00, the value stored is "180" because 3 hours = 180 minutes have elapsed.

D8145 is reset to 0 when the current time is successfully acquired, then the elapsed time count starts. If acquiring the current time from the SNTP server was not successful even once, the value of D8145 is not updated.

Ping Settings

This section describes the ping timeout setting for the PING instruction and when the auto ping function is being executed.

Description

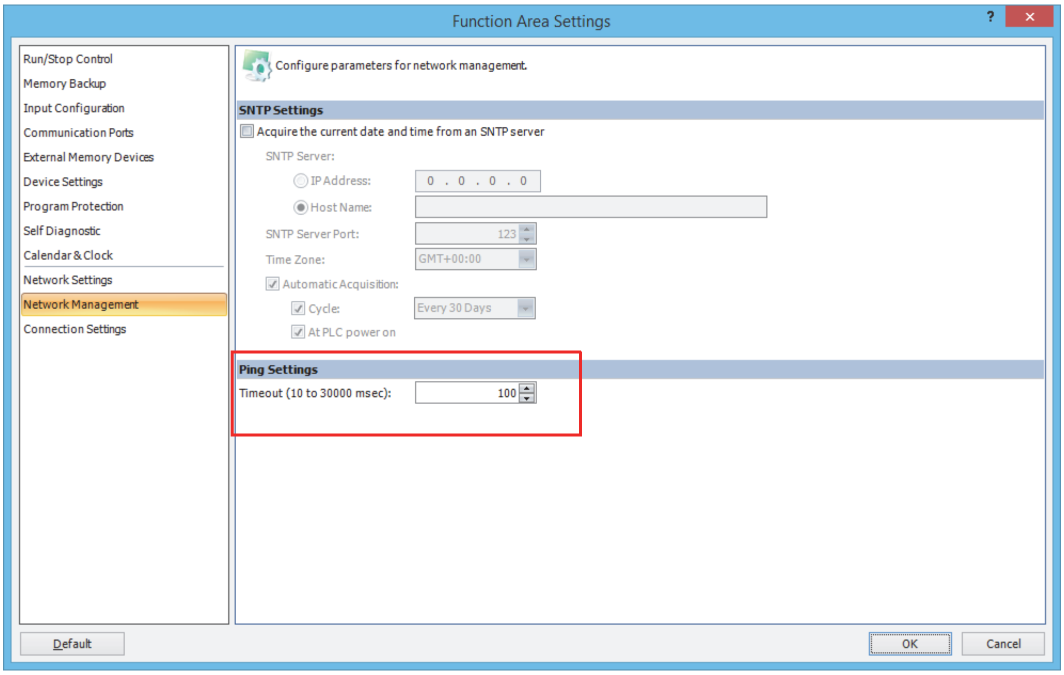
When sending pings with the PING instruction or auto ping function, a ping packet is sent to the specified remote host to check whether or not communication is possible at the IP level. Set the time from when this ping is sent until the timeout occurs.

Programming WindLDR

1. On the **Configuration** tab, in the **Function Area Settings** group, click the following item.

- All-in-One CPU module/CAN J1939 All-in-One CPU module
  Network Management
- Plus CPU module
  Ethernet Port 1 to configure Ethernet port 1
  Ethernet Port 2 to configure Ethernet port 2

2. With **Timeout (10 to 30000 msec)**, set the PING instruction timeout between 10 and 30,000 ms.
The default is 100 ms.



3. Click **OK**.

This concludes configuring the settings.

Connection Settings

This section describes the settings for client/server communication on TCP/IP communication using Ethernet port 1, Ethernet port 2, or the HMI-Ethernet port of the FC6A Series MICROSmart.

Applications

The All-in-One CPU module/CAN J1939 All-in-One CPU module is capable of maintenance communication, user communication, and Modbus TCP communication using a maximum of eight connections over Ethernet port 1.

The Plus CPU module is capable of maintenance communication, user communication, and Modbus TCP communication using a maximum of 16 connections over Ethernet port 1 and Ethernet port 2.

Those communications can be configured in the **Function Area Settings** dialog box.

Also, the HMI-Ethernet port can be used to expand a maximum of eight connections when using an HMI module.

However, the only communication function that is supported by the HMI-Ethernet port is maintenance communication.

Description of functions

The communication mode of each connection can be selected from six types of communication: maintenance communication server, user communication server, user communication client, Modbus TCP server, Modbus TCP client, or unused. User communication (UDP) is also available in the Plus CPU module, so the communication mode can be selected from the seven types. A maximum of eight connections can be configured for the All-in-One CPU module/CAN J1939 All-in-One CPU module. A maximum of 16 connections can be configured for the Plus CPU module.

To limit the access to the FC6A Series MICROSmart, IP address flittering can be used. By specifying the IP address that can access the FC6A Series MICROSmart, anonymous access can be prevented.

Communication Modes Available with the Ethernet Ports

The communication modes supported by each Ethernet port are as follows.

Communication Mode	All-in-One CPU Module/ CAN J1939 All-in-One CPU module	Plus CPU Module		HMI Module
	Ethernet Port 1	Ethernet Port 1	Ethernet Port 2	HMI-Ethernet Port
Maintenance communication server (Default)	X	X	X	X
User communication server	X	X	X	—
User communication client	X	X	X	—
User communication (UDP)	—	X	X	—
Modbus TCP server	X	X	X	—
Modbus TCP client	X	X	X	—
Unused	X	X	X	X

Number of Connections

CPU Module	Amount
All-in-One CPU module	8 maximum
Plus CPU module	16 maximum (A maximum of 255 remote devices can be registered when using the Modbus TCP client. Connections with a maximum of 128 remote hosts can be maintained during the communication.)
HMI module	8 maximum

Overview of Communication Modes

The following table presents an overview of the communication modes.

Communication Mode	Overview	Reference
Maintenance communication server (Default)	This mode allows monitoring and changing devices and user program uploading and downloading from WindLDR.	4-7
User communication server	This mode enables communication with client devices according to the ETXD (User Communication Transmit over Ethernet) instruction and the ERXD (User Communication Receive over Ethernet) instruction.	5-47
User communication client	This mode enables communication with server devices according to the ETXD (User Communication Transmit over Ethernet) instruction and the ERXD (User Communication Receive over Ethernet) instruction.	5-43

Communication Mode	Overview	Reference
User communication (UDP) *1	This mode enables communication with UDP communication devices according to the ETXD (User Communication Transmit over Ethernet) instruction and the ERXD (User Communication Receive over Ethernet) instruction.	5-50
Modbus TCP server	When the FC6A Series MICROSmart is configured as a Modbus TCP server, FC6A Series MICROSmart data monitoring and modifications can be executed from Modbus TCP client-compatible devices.	6-27
Modbus TCP client	Modbus TCP server-compatible device data monitoring and modifications can be executed from Modbus TCP clients.	6-22
Unused	The connection is not used. This option stops the internal control processing, which improves the responsiveness of the other connections.	—

*1 Can be selected for only the Plus CPU module.

Connection Status and Connected IP Address

The connection status of connections with remote hosts can be confirmed with special internal relays M8212 to M8231 and M8345 to M8364. When a connection with a remote host is established, the corresponding special internal relay is turned on. When the connection is disconnected, the corresponding special internal relay is turned off. The IP addresses of the remote hosts can be confirmed with special data registers D8350 to D8381.

Note: R/W is the abbreviation for read/write. When R/W, it can be read and written. When R, it can only be read. When W, it can only be written.

Special Internal Relays

Device Address	Description	Details	Read/Write
M8212	Connection 1 Status	While a connection is established with a remote host, the special internal relay will be on. When no connection is established, it will be off.	R
M8213	Connection 2 Status		
M8214	Connection 3 Status		
M8215	Connection 4 Status		
M8216	Connection 5 Status		
M8217	Connection 6 Status		
M8220	Connection 7 Status		
M8221	Connection 8 Status		
M8345*1	Connection 9 Status		
M8346*1	Connection 10 Status		
M8347*1	Connection 11 Status		
M8350*1	Connection 12 Status		
M8351*1	Connection 13 Status		
M8352*1	Connection 14 Status		
M8353*1	Connection 15 Status		
M8354*1	Connection 16 Status		
M8222	Connection 1 disconnected flag	When turned on, the connection to the server is forcibly disconnected to stop communication. When turned off, the connection to the server is established and communication is restarted. This flag is only valid in user communication client.	R/W
M8223	Connection 2 disconnected flag		
M8224	Connection 3 disconnected flag		
M8225	Connection 4 disconnected flag		
M8226	Connection 5 disconnected flag		
M8227	Connection 6 disconnected flag		
M8230	Connection 7 disconnected flag		
M8231	Connection 8 disconnected flag		
M8355*1	Connection 9 disconnected flag		
M8356*1	Connection 10 disconnected flag		
M8357*1	Connection 11 disconnected flag		
M8360*1	Connection 12 disconnected flag		
M8361*1	Connection 13 disconnected flag		
M8362*1	Connection 14 disconnected flag		
M8363*1	Connection 15 disconnected flag		
M8364*1	Connection 16 disconnected flag		

*1 Can be used only with the Plus CPU module.

Connected IP addresses

Special Data Register	Description		Read/Write
D8350-D8353	Connection 1 Connected IP Address	When communication has been established between an external device and a connection, the IP address of the connected external device is stored in special data registers as shown below. Example: IP Address: aaa.bbb.ccc.ddd, D8350=aaa, D8351=bbb, D8352=ccc, D8353=ddd	R
D8354-D8357	Connection 2 Connected IP Address		
D8358-D8361	Connection 3 Connected IP Address		
D8362-D8365	Connection 4 Connected IP Address		
D8366-D8369	Connection 5 Connected IP Address		
D8370-D8373	Connection 6 Connected IP Address		
D8374-D8377	Connection 7 Connected IP Address		
D8378-D8381	Connection 8 Connected IP Address		
D8677-D8680[*1]	Connection 9 Connected IP Address		
D8681-D8684[*1]	Connection 10 Connected IP Address		
D8685-D8688[*1]	Connection 11 Connected IP Address		
D8689-D8692[*1]	Connection 12 Connected IP Address		
D8693-D8696[*1]	Connection 13 Connected IP Address		
D8697-D8700[*1]	Connection 14 Connected IP Address		
D8701-D8704[*1]	Connection 15 Connected IP Address		
D8705-D8708[*1]	Connection 16 Connected IP Address		

*1 Can be used only with the Plus CPU module.

HMI module connection status information and connected IP address

The connection status for the HMI module connection number specified by D8429 is stored in special internal relays and special data registers.
The status of the connection to an HMI module external device can be checked with special internal relay M8232.
The relay is on when connected to the external device and off when not connected.
The connected IP address can be checked with special data registers D8430 to D8433.

Read specified target connection number

Special Internal Relays	Description		Read/Write
D8429	HMI module read target connection number	Reflects the connection information for the specified number to D8430 through D8434 and M8232. When 0 is specified, the target devices are all set to 0. If a connection not that does not exist is specified, the operation is the same as when 0 is specified.	R/W

Status information

Special Internal Relays	Description		Read/Write
M8232	HMI module connection status	On when the connection specified with D8429 is connected to an external device and off when there is no connection.	R

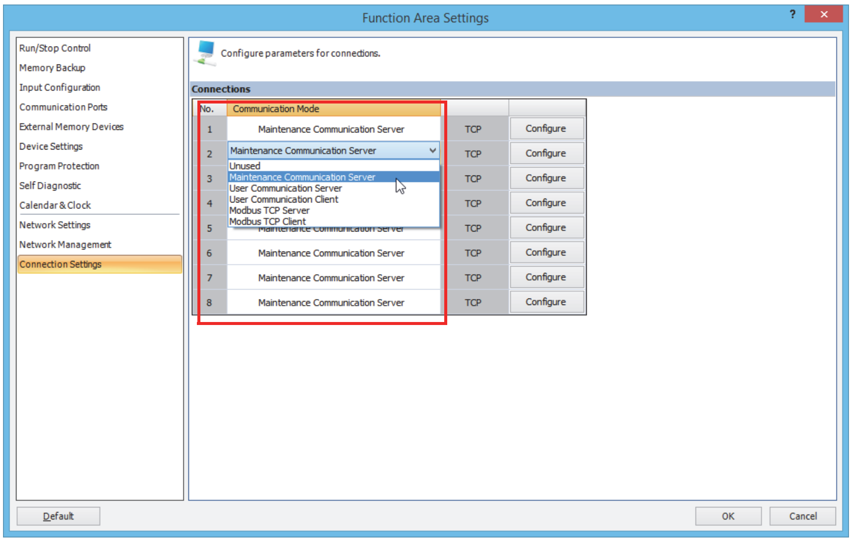
Connected IP Address

Special Internal Relays	Description		Read/Write
D8430-D8433	HMI connection Connected IP Address	When communication has been established between an external device and the connection specified by D8429, the IP address of the connected external device is stored in special data registers as shown below. Example: IP Address: aaa.bbb.ccc.ddd, D8430=aaa, D8431=bbb, D8432=ccc, D8433=ddd	R

Programming WindLDR

1. From the WindLDR menu bar, select **Configuration** > **Connection Settings**.
   The **Function Area Settings** dialog box is displayed.

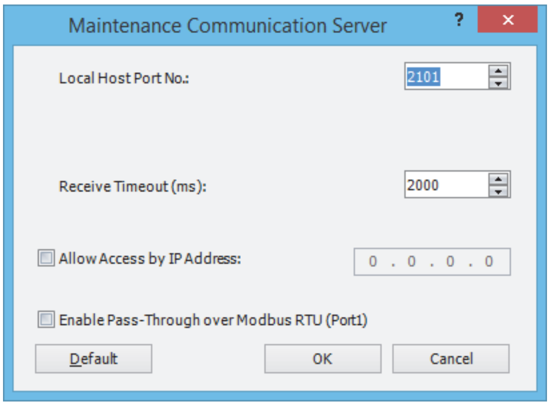
2. Click **Communication Mode** for the port to be used and select communication mode of the connection to use.



Configure the communication modes allocated to a maximum of eight connections for the All-in-One CPU module and 16 connections for the Plus CPU module that can be used as client or server. Each connection can be set to a different port number. Set connections that are not used to **Unused**.

A dialog box is displayed according to the communication mode.

3. Specify the parameters according to the communication format of the destination device.

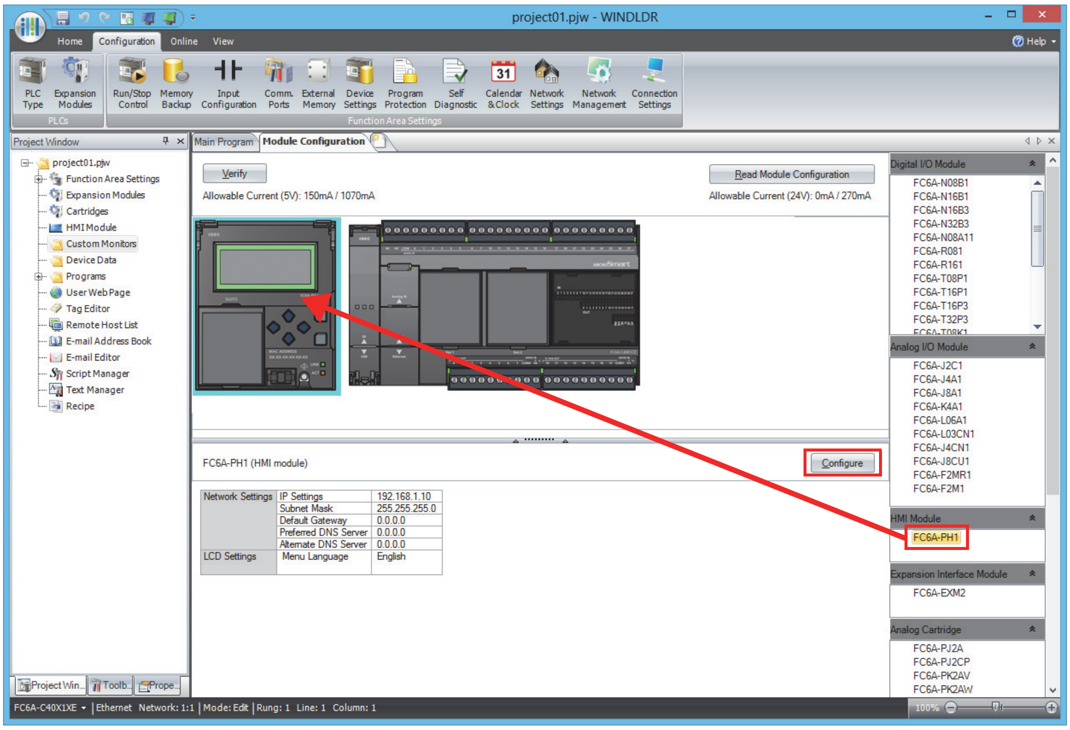


4. Click **OK**.

5. Use the Module Configuration Editor to configure the HMI module network settings.
   On the **Configuration** tab, in the **PLCs** group, select **Expansion Modules**.

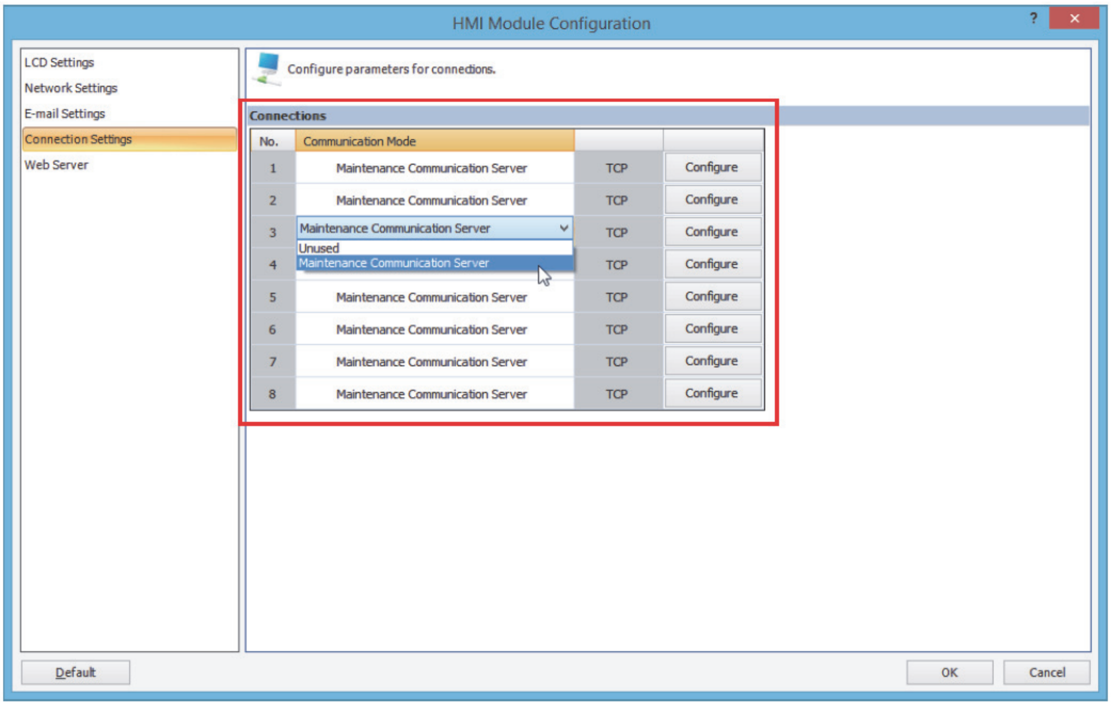
6. Click the inserted HMI module in the module configuration area and click **Configure**.
   The **HMI Module Configuration** dialog box is displayed.

Note: You can also display the **HMI Module Configuration** dialog box by double-clicking **HMI Module** in the Project Window.

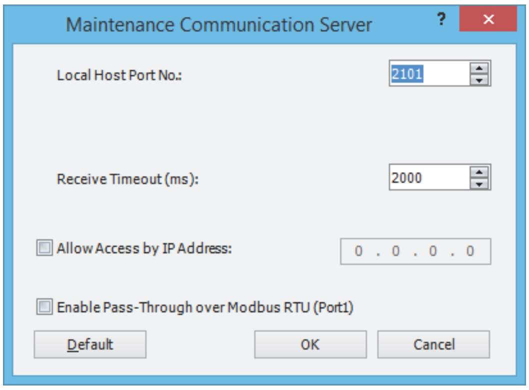


7. Click **Connection Settings**.

8. Click **Communication Mode** for the port to be used and select communication mode of the connection to use.



9. Specify the parameters according to the communication format of the destination device.



10. Click **OK**.

This concludes configuring the settings.

Remote Host List

This chapter describes how to configure a list of network devices (remote hosts) in the network to which the FC6A Series MICROSmart communicates.

Description

When the FC6A Series MICROSmart accesses and communicates with other network devices in the network, the remote host device should be specified. The remote host list is required to use the following functions:

- PING Instruction
- ETXD/ERXD Instructions (User communication over Ethernet)
- Modbus TCP Client

Description of Functions

The remote host consists of an **IP Address** or a **Host Name** and a **Port Number**.

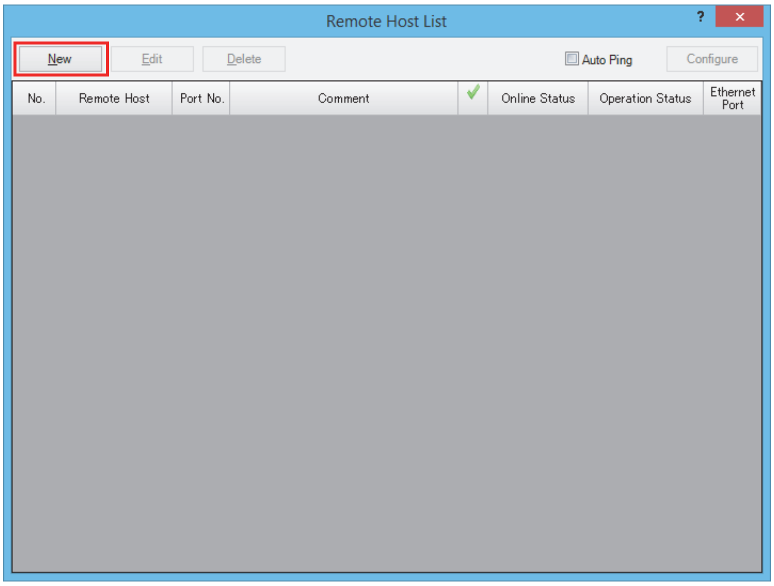
When a remote host is specified with an IP address, and the FC6A Series MICROSmart establishes connection with the remote host that has the specified IP address and the corresponding port number, then communication is started.

When a remote host is specified with a host name, the FC6A Series MICROSmart tries to obtain the IP address from the specified host name using the DNS server. If the IP address is successfully obtained, the FC6A Series MICROSmart establishes connection with the remote host that has the specified IP address and the corresponding port number, then communication is started.

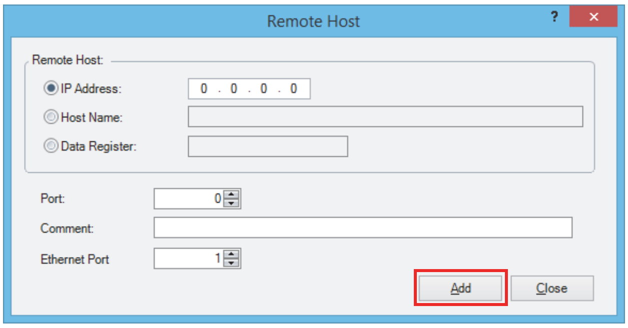
For details about DNS server settings, see Network Settings in the **Function Area Settings**.

Programming WindLDR

1. Click **Project Window** in the **Work Space** on the **View** tab.
   The Project Window is displayed on the left side of the screen.
2. Double-click on the Remote Host List in the **Project Window**.
   The **Remote Host List** dialog box appears.
3. Click **New** button.
   The **Remote Host List** dialog box appears.



4. Configure the parameters in the **Remote Host** dialog box and then click on **Add** button.



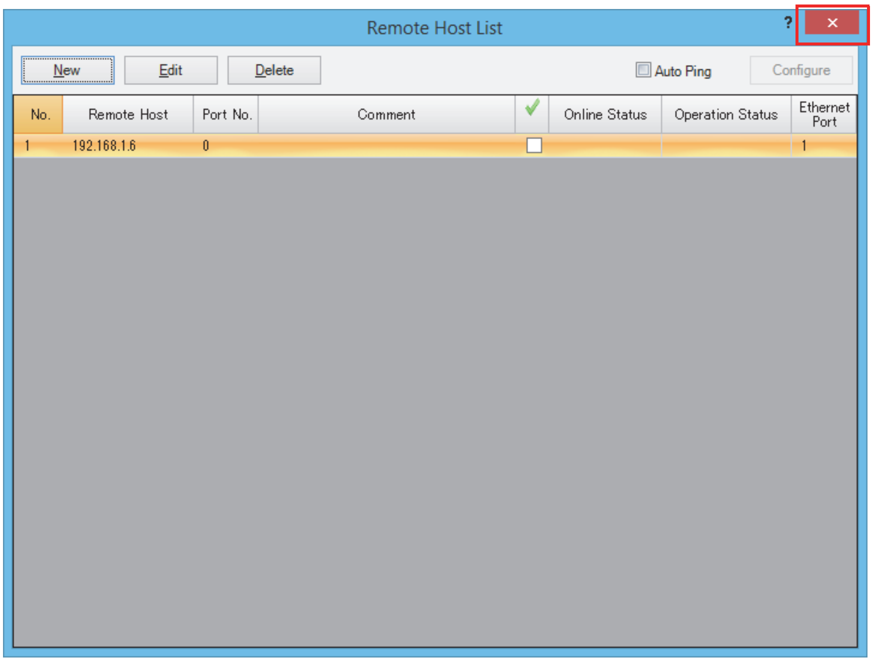
The remote host is composed of the following items.

Item	Setting Value
IP Address	Specify the remote host with an IP address. The FC6A Series MICROSmart will communicate by establishing a connection to the set IP address and port number.
Host Name	Specifies the remote host as a host name. Up to 40 single-byte alphanumeric characters can be entered.
Data Register	Specify the IP address of the remote host as data registers (4 words).
Port	Specify the port number of the remote host. This port number is the TCP communication port number. It differs from the numbers of the FC6A Series MICROSmart USB port and port 1 through port 3.
Comment	The comment for the remote host can be assigned. The contents or the lngth of the comment has no effect on the FC6A Series MICROSmart operation.
Ethernet Port	Select the Ethernet port to communicate with the remote host.

A new remote host will be added in the Remote Host List dialog box. If you want to add additional remote hosts, repeat the same procedure.

5. To periodically ping the specified remote host, configure auto ping.
For details, see "Auto Ping Function" on page 3-22.

6. After adding remote hosts has been completed, click on **Close** button.



7. If you want to delete an unused remote host, select that remote host in the **Remote Host List** dialog box and click on **Delete** button.

This concludes configuring the settings.

Note: Once a remote host is deleted, the remote host numbers of the following remote hosts are changed. As a result, there is an effect on the operation of the PING instruction, Modbus TCP client, and user communication client that refer to these remote host list numbers.

Auto Ping Function

This section describes the auto ping function that periodically pings the specified remote host.

Function Description

The function to periodically ping the specified remote host is called the auto ping function. This function can be used to check whether or not communication is possible with the specified remote host by sending a packet to that host.

The remote host is specified with the remote host list.

When auto ping is enabled and immediately after auto ping is enabled due to the FC6A Series MICROSmart power being turned on and a change in the user program, auto ping execution starts and the specified remote host numbers are pinged in order from the smallest number. There is no impact on the FC6A run and stop statuses and the ladder program.

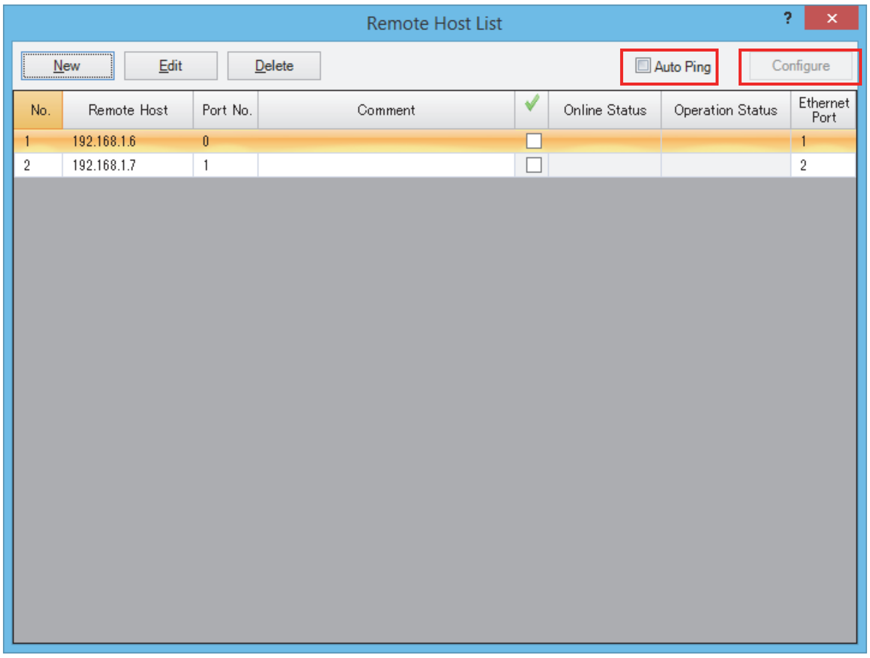
For Ethernet port 1, when M8186 (Ethernet port 1 executing auto ping) is on, auto ping will stop if M8187 (Ethernet port 1 auto ping stop flag) is turned on. When M8187 is turned off, auto ping execution starts.

For Ethernet port 2, when M8331 (Ethernet port 2 executing auto ping) is on, auto ping will stop if M8332 (Ethernet port 2 auto ping stop flag) is turned on. When M8332 is turned off, auto ping execution starts. Ethernet port 2 can be used with only the Plus CPU module.

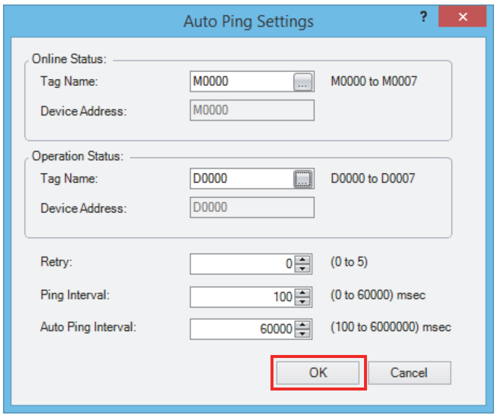
The results of the auto ping function can be linked with Modbus TCP client request processing. For linking with Modbus TCP clients, see "Modbus TCP Client" on page 6-22.

Programming WindLDR

1. In the **Remote Host List** dialog box, select the **Auto Ping** check box and click **Configure**.
   The **Auto Ping Settings** dialog box is displayed.



2. Specify the auto ping function items and click **OK**.

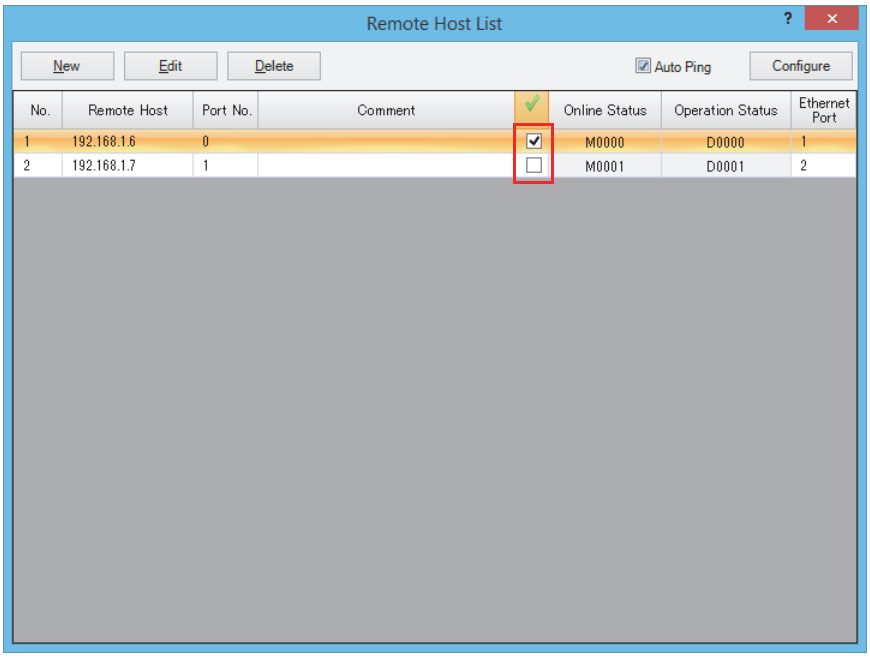


The auto ping function is composed of the following items.

Item	Setting Value
Online Status:	Specify the internal relay that stores the results of the pings sent with the auto ping function. If an internal relay is entered, the range of internal relays that will be used is displayed on the right side. The internal relays are all turned off when auto ping function execution starts. After the ping is sent, they are turned on when a response is received. • Ethernet port 1 When M8186 (Ethernet port 1 executing auto ping) is on and M8187 (Ethernet port 1 auto ping stop flag) is changed from off to on or the user program is changed and the Ethernet port 1 auto ping function is set from **Used** to **Not used**, the internal relay on/off status is left unchanged. When M8186 (Ethernet port 1 executing auto ping) is off and M8187 (Ethernet port 1 auto ping stop flag) is changed from on to off or the user program is changed and the Ethernet port 1 auto ping function is set from **Not used** to **Used**, the internal relays are all turned off. • Ethernet port 2[*1] When M8331 (Ethernet port 2 executing auto ping) is on and M8332 (Ethernet port 2 auto ping stop flag) is changed from off to on or the user program is changed and the Ethernet port 2 auto ping function is set from **Used** to **Not used**, the internal relay on/off status is left unchanged. When M8331 (Ethernet port 2 executing auto ping) is off and M8332 (Ethernet port 2 auto ping stop flag) is changed from on to off or the user program is changed and the Ethernet port 2 auto ping function is set from **Not used** to **Used**, the internal relays are all turned off. There is no impact on the operation of the PING instruction.
Operation Status:	Specify the data register that stores the operation status of the ping sent with the auto ping function. If a data register is entered, the range of data registers that will be used is displayed on the right side. The operation status indicates the operation status (status code) and the error details (error code). The status code is stored in the upper 12 bits and the error code is stored in the lower 4 bits. For the status code details, see "Operation status" on page 3-25. There is no impact on the operation of the PING instruction.
Retry:	Specify the retry count (0 to 5 times) of the ping sent with the auto ping function.
Ping Interval:	Specify the send interval (0 to 60,000 ms) of the pings sent with the auto ping function in 10 ms increments.
Auto Ping Interval:	Specify the interval (100 to 6,000,000 ms) until the next auto ping is executed after the auto ping was executed in 100 ms increments.

*1 Can be used only with the Plus CPU module.

3. In the **Remote Host List** dialog box, select the check box of the remote hosts to ping when auto ping executes.



4. Click **Close**.

This concludes configuring the settings.

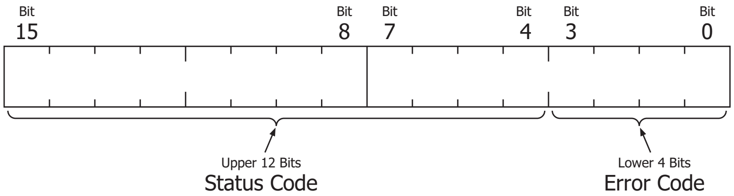
Operation status

The operation status indicates the operation status (status code) and the error details (error code).

The operation status (status code) is stored in the upper 12 bits of D8414 and the error details (error code) is stored in the lower 4 bits of D8414.

When the value of the operation status data register is divided by 16, the remainder is the error code.

If the value of the operation status data register is 66:
66 ÷ 16 = 4 with a remainder of 2, so the error code is 2.



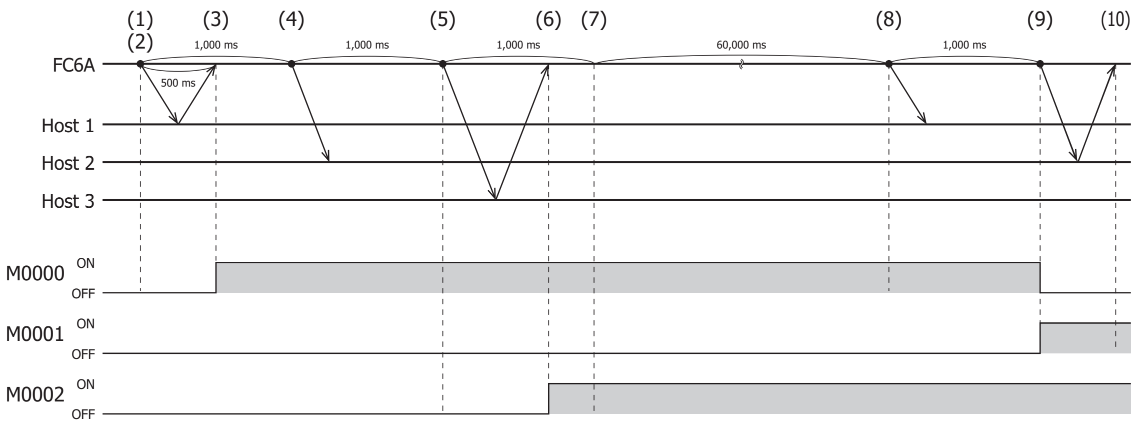
Status Code	Operation Status
16 (000000010000)	Status after ping executed and before the packet is sent
32 (000000100000)	Status after packet send processing has completed and waiting for a response from the host
64 (000001000000)	Status where receiving the response for the packet completed normally or a timeout error occurred and the next ping can be executed

Error Code	Error Details
0 (0000)	Normal
2 (0010)	Timeout error
3 (0011)	The destination host name could not be resolved by DNS

Auto ping operation example 1

This example describes the operation when auto ping is executed for three remote hosts under the following conditions.

Settings		Setting Details
Ping settings	Timeout	1,000 ms
Auto ping settings	Online Status	M0000
	Retry	0 times
	Ping Interval	1,000 ms
	Auto Ping Interval	60,000 ms

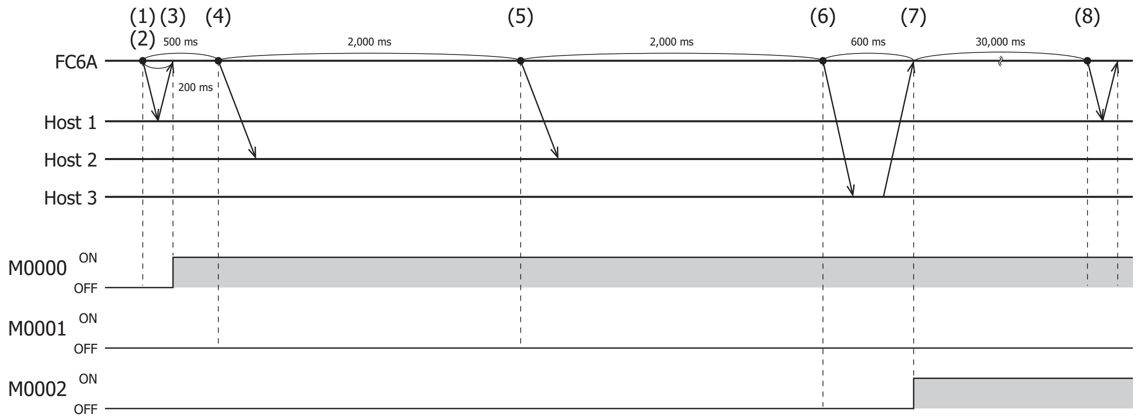


(1) : Auto ping operation starts
(2) to (3) : Response from Host 1 within the timeout
(4) : No response from Host 2 within the timeout
(5) to (6) : Response from Host 3 within the timeout
(7) to (8) : Auto Ping Interval
(8) : No response from Host 1 within the timeout
(9) to (10) : Response from Host 2 within the timeout

Auto ping operation example 2

This example describes the operation when auto ping is executed for three remote hosts under the following conditions.

Settings		Setting Details
Ping settings	Timeout	2,000 ms
Auto ping settings	Online Status	M0000
	Retry	1 time
	Ping Interval	500 ms
	Auto Ping Interval	30,000 ms



(1) : Auto ping operation starts
(2) to (3) : Response from Host 1 within the timeout
(4) to (5) : No response from Host 2 within the timeout, no response even for retries
When the timeout is longer than the ping interval, there is no ping interval wait time.
(6) : Response from Host 3 was late, but within the timeout
(7) to (8) : Auto Ping Interval

4: MAINTENANCE COMMUNICATION

Introduction

This chapter describes the FC6A Series MICROSmart maintenance communication function.

Maintenance communication is a communication protocol dedicated for IDEC programmable controllers. It is used when WindLDR or an IDEC operator interface communicates with the FC6A Series MICROSmart.

The FC6A Series MICROSmart supports maintenance communication on each communication port, allowing the optimum communication method to be selected for a variety of system configurations.

Maintenance Communication Functions

When performing maintenance communication with the FC6A Series MICROSmart, the following functions can be used:

Function	Description
Download user programs	User programs created in WindLDR can be downloaded to the FC6A Series MICROSmart. See Chapter 4 "Download Program" in the "FC6A Series MICROSmart User's Manual".
Upload user programs	User programs stored in the FC6A Series MICROSmart can be uploaded to WindLDR.
Monitor/change device values	The user program and the device values of the FC6A Series MICROSmart can be monitored and the device values can be changed using WindLDR. See Chapter 4 "Monitor Operation" in the "FC6A Series MICROSmart User's Manual".
Download system software	System software can be downloaded to the FC6A Series MICROSmart. See Appendix "Upgrade FC6A Series MICROSmart System Software" in the "FC6A Series MICROSmart User's Manual".

Note: To use maintenance communication, see Chapter 4 "Start WindLDR" in the "FC6A Series MICROSmart User's Manual" and perform setup.

Communication Ports Used For Maintenance Communication

Supported models, ports, and slots are as follows.

Port	All-in-One CPU Module			CAN J1939 All-in-One CPU Module	Plus CPU Module	
	16-I/O Type	24-I/O Type	40-I/O Type		Plus 16-I/O Type	Plus 32-I/O Type
USB Port	Yes	Yes	Yes	Yes	Yes	Yes
Port 1	Yes	Yes	Yes	—	Yes*2	Yes*2
Port 2	Yes*1	Yes*1	Yes*1	Yes*1	Yes*2	Yes*2
Port 3	—	—	Yes*1	Yes*1	Yes*3	Yes*3
Port 4 to 33	Yes (Port 4 to 9)*4				Yes (Port 4 to 33)*5	
Ethernet Port 1	Yes	Yes	Yes	Yes	Yes	Yes
Ethernet Port 2	—	—	—	—	Yes	Yes
HMI-Ethernet Port*6	Yes	Yes	Yes	Yes	Yes	Yes

*1 When a communication cartridge is connected.
*2 When a cartridge base module and communication cartridge are connected.
*3 When the HMI module and a communication cartridge are connected.
*4 Up to three communication modules can be connected to the All-in-One CPU module and CAN J1939 All-in-One CPU module, expanding the communication ports 4 to 9.
*5 Up to 15 communication modules can be connected to the Plus CPU module, expanding the communication ports 4 to 33.
*6 HMI-Ethernet port can be expanded by connecting the HMI module to the CPU module.

Maintenance communication methods that are supported by the communication ports are as follows.

Communication Method		USB Port	Port 1	Port 2 and Port 3	Port 4 to 33	Ethernet Port 1	Ethernet Port 2	HMI-Ethernet Port
Maintenance Communication	Download system software	Yes	Yes*1	No	No	Yes	Yes	Yes
	Download/upload user programs	Yes	Yes*1	No	No	Yes	Yes	Yes
	Run-time download	Yes	Yes*1	No	No	Yes	Yes	Yes
	Monitor/change device values	Yes	Yes	Yes	Yes	Yes	Yes	Yes

*1 Not supported with a communication cartridge.

Maintenance communication functions that are supported by the communication ports and slots are as follows.

All-in-One CPU module/CAN J1939 All-in-One CPU module

Communication Port	Standard/ Expansion	Communication Settings
USB Port	Standard	None
Port 1	Standard*1	Function area settings. See "Maintenance Communication via Port 1" on page 4-4.
Port 2	Expansion*2	When using a RS232C/RS485 communication cartridge: Function area settings. See "Maintenance Communication via Port 2 to 33" on page 4-12. When using a Bluetooth communication cartridge: Function area settings. See "Maintenance Communication via Port 1 to 3 (Bluetooth)" on page 4-15.
Port 3		
Port 4 to 9	Expansion*3	Function area settings. See "Maintenance Communication via Port 2 to 33" on page 4-12.
Ethernet Port 1	Standard	Function area settings. See "Maintenance Communication via Ethernet Port 1 and 2" on page 4-7.
Ethernet Port 2	—	—
HMI-Ethernet Port	Standard*4	Function area settings. See "Maintenance Communication via HMI-Ethernet port" on page 4-16.

*1 The CAN J1939 All-in-One CPU module does not have port 1.
*2 The ports can be expanded by connecting a communication cartridge to the CPU module.
*3 The ports can be expanded by using a communication module.
*4 The ports can be expanded by using the HMI module.

Plus CPU module

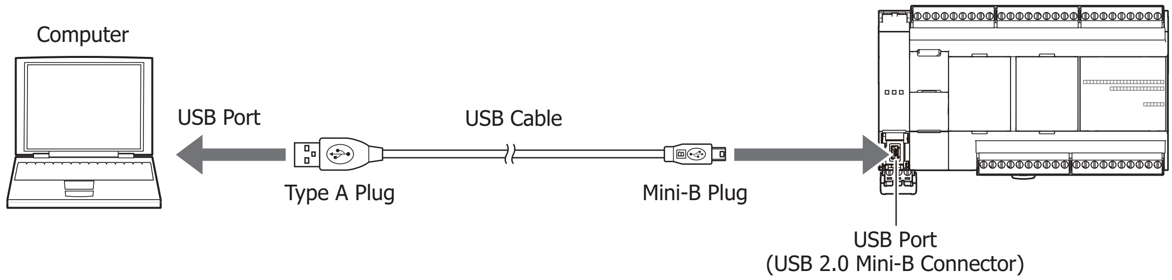
Communication Port	Standard/ Expansion	Communication Settings
USB Port	Standard	None
Port 1	Expansion*1	When using a RS232C/RS485 communication cartridge: Function area settings. See "Maintenance Communication via Port 1" on page 4-4. When using a Bluetooth communication cartridge: Function area settings. See "Maintenance Communication via Port 1 (Bluetooth)" on page 4-6.
Port 2		When using a RS232C/RS485 communication cartridge: Function area settings. See "Maintenance Communication via Port 2 to 33" on page 4-12. When using a Bluetooth communication cartridge: Function area settings. See "Maintenance Communication via Port 1 to 3 (Bluetooth)" on page 4-15.
Port 3	Expansion*2	
Port 4 to 33	Expansion*3	Function area settings. See "Maintenance Communication via Port 2 to 33" on page 4-12.
Ethernet Port 1	Standard	Function area settings. See "Maintenance Communication via Ethernet Port 1 and 2" on page 4-7.
Ethernet Port 2		
HMI-Ethernet Port	Expansion*4	Function area settings. See "Maintenance Communication via HMI-Ethernet port" on page 4-16.

*1 The ports can be expanded by connecting a communication cartridge to a cartridge base module.
*2 The ports can be expanded by connecting a communication cartridge to the HMI module.
*3 The ports can be expanded by using a communication module.
*4 The ports can be expanded by using the HMI module.

Maintenance Communication via USB Port

Using the USB port, it is possible to connect FC6A Series MICROSmart to a computer on which WindLDR is installed, and monitor and change device values, download and upload user programs, and download system software. Connect the computer and the FC6A Series MICROSmart using a USB cable (recommended cable: HG9Z-XCM42).

FC6A Series MICROSmart USB Port



Maintenance Communication Specifications

Item	Specifications/Functions
Cable	Recommended cable: USB maintenance cable (HG9Z-XCM42)
Maintenance communication functions	Monitor/change device values Download/upload user programs Download system software Run-time download

Detailed Operations for Maintenance Communication Functions

See the following in the "FC6A Series MICROSmart User's Manual".

- Monitor/change device values: Chapter 4 "Monitor Operation"
- Download/upload user programs: Chapter 4 "Download Program"
- Download system software: Appendix "Upgrade FC6A Series MICROSmart System Software"
- Run-time download: Chapter 5 "Online Edit"

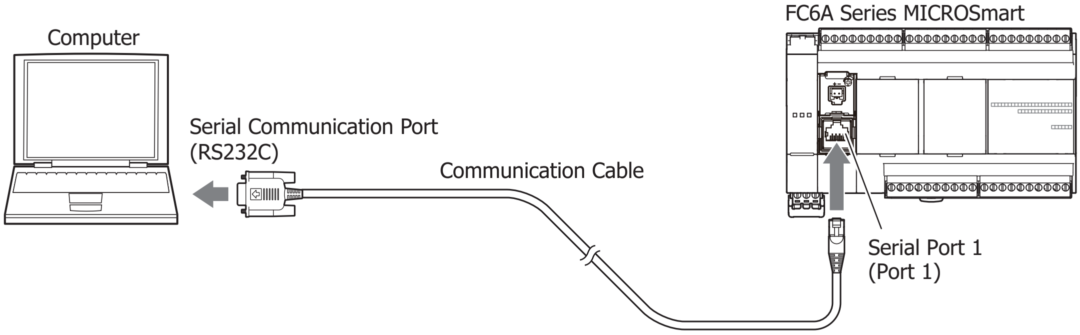
Maintenance Communication via Port 1

Computers in which WindLDR is installed or IDEC operator interfaces can be connected to the FC6A Series MICROSmart, and monitoring and changing device values, downloading or uploading the user program, and downloading the system software are possible.

Maintenance communication on port 1 is possible by using the serial port 1 of the All-in-One CPU module or using the communication cartridge installed to the cartridge slot 1 of the cartridge base module that is connected to the Plus CPU module.

For details on communication cables, see Appendix "Cables" in the "FC6A Series MICROSmart User's Manual".

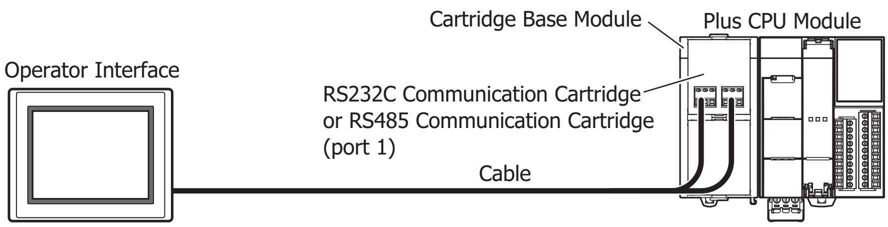
All-in-One CPU Module Port 1



Maintenance Communication Specifications

Item	Specifications/Functions
Cable	O/I communication cable (FC6A-KC2C)
Maintenance communication functions	Monitor/change device values Download/upload user programs Download system software Run-time download

Plus CPU Module Port 1



Maintenance Communication Specifications

Item	Specifications/Functions
Cable	RS232C communication cartridge: Shielded multicore cable RS485 communication cartridge: Shielded twisted-pair cable
Maintenance communication functions	Monitor/change device values

Note:

The following functions are not supported by maintenance communication using an RS232C communication cartridge or RS485 communication cartridge.

- Download/upload user programs
- Download system software
- Run-time download

Detailed Operations for Maintenance Communication Functions

See the following in the "FC6A Series MICROSmart User's Manual".

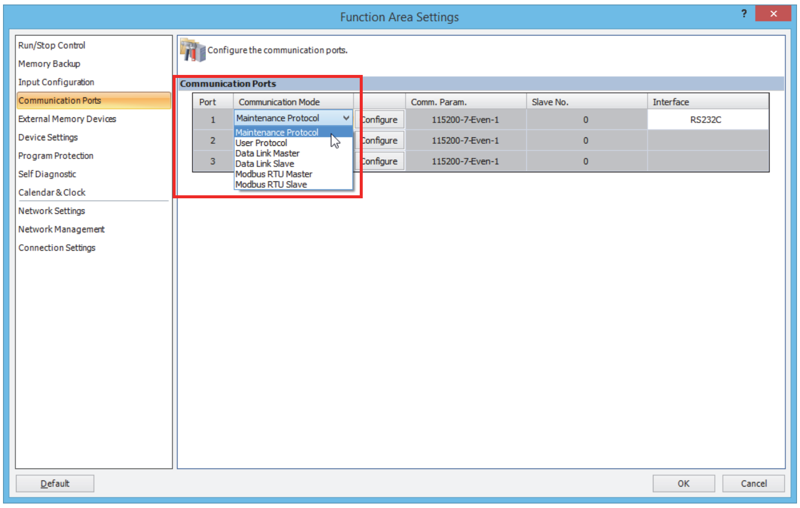
- Monitor/change device values: Chapter 4 "Monitor Operation"

Programming WindLDR

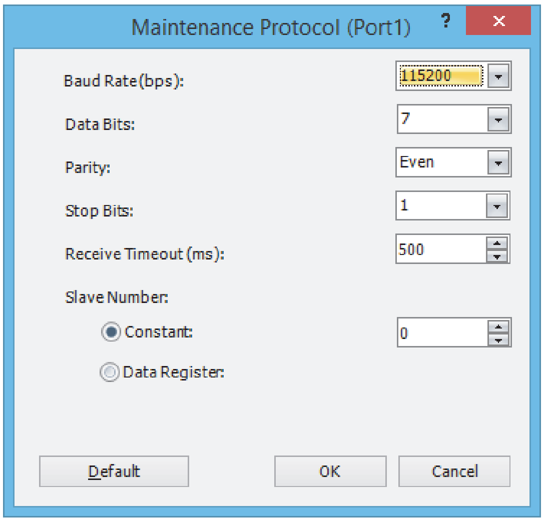
Configure the settings for maintenance communication.

1. From the WindLDR menu bar, select **Configuration** > **Comm**. **Ports**.
   The **Function Area Settings** dialog box appears.

2. Click **Communication Mode** for port 1 and select **Maintenance Protocol**.
The **Maintenance Protocol (Port1)** dialog box appears.



3. Configure the parameters to match the communication settings of the computer or operator interface.



Baud Rate (bps): 115200 bps (1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200)
Data Bits: 7 (7 or 8)
Parity: Even (None, Even, Odd)
Stop Bits: 1 (1 or 2)
Receive Timeout (ms): 500 (10 to 2550)
Slave Number: 0 (0 to 31)
Slave number can be specified by either a constant or a data register.

Type	Details
Constant	Set within the range of 0 to 31
Data register	Store the slave numbers 0 to 31 in the following special data registers Port 1: D8100 Port 2: D8102 Port 3: D8103 Port 4 to 9*1: D8040 to D8045 Port 10 to 33*1*2: D8735 to D8758

*1 When using a communication module.
*2 Plus CPU module only.

* Values not in parentheses are the default settings.

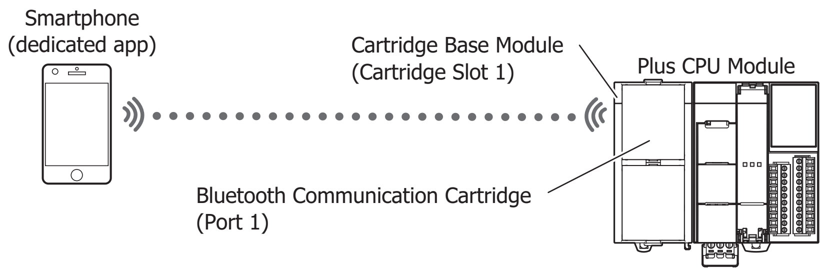
* Set **Data Bits** to **8** when downloading the system software over port 1.

4. Click **OK**.
Configuring the maintenance communication for the expansion communication port is now complete.

Maintenance Communication via Port 1 (Bluetooth)

Connect a Bluetooth communication cartridge to cartridge slot 1 of a cartridge base module added to the Plus CPU module to connect to computers or smartphones equipped with Bluetooth and allow device values to be monitored and changed.

Plus CPU Module Bluetooth



Maintenance Communication Specifications

Item	Specifications/Functions
Maintenance Communication Function	Monitor/change device values Download/upload user programs

Notes:

- The following functions are not supported by maintenance communication using a Bluetooth communication cartridge.
  - Download system software
  - Run-time download
- Bluetooth communication may fail and a download error may occur after a user program is downloaded and Bluetooth communication cartridge settings are changed.

Detailed Operations for Maintenance Communication Functions

See the following in the "FC6A Series MICROSmart User's Manual".

- Monitor/change device values: Chapter 4 "Monitor Operation"

Programming WindLDR

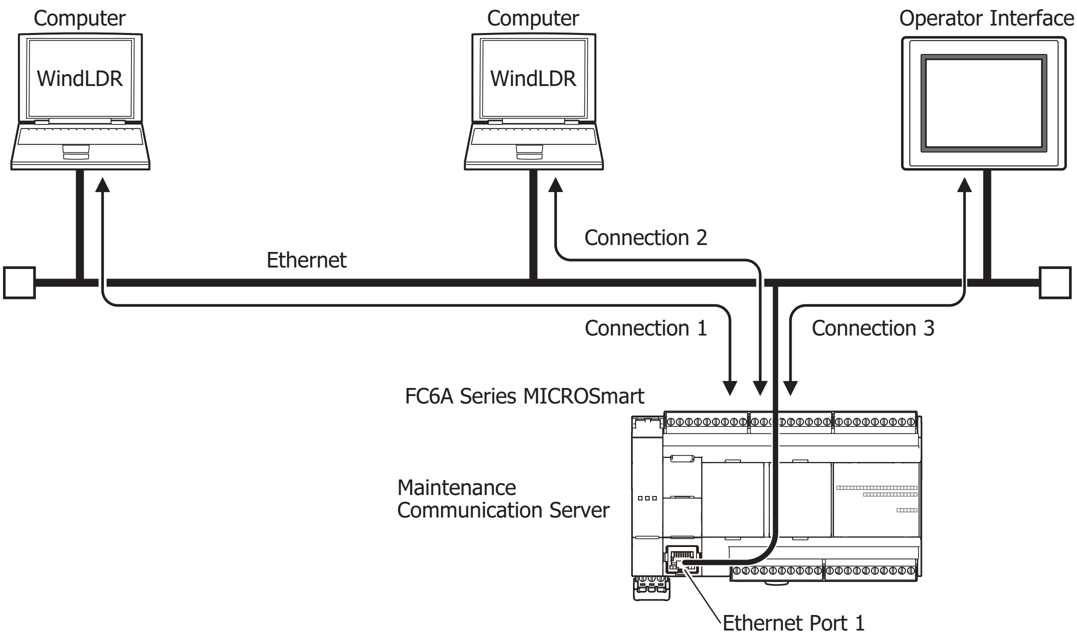
For details, see "Bluetooth Communication" on page 9-1.

Maintenance Communication via Ethernet Port 1 and 2

The FC6A Series MICROSmart can communicate with network devices such as computers or IDEC operator interfaces using Ethernet port 1 of the All-in-One CPU module/CAN J1939 All-in-One CPU module and Ethernet port 1 and 2 of the Plus CPU module. External devices on the network can monitor or change the device values and download or upload user programs.

The FC6A Series MICROSmart can use the maintenance communication server and other communications functions simultaneously by assigning a separate communication function, such as a maintenance communication server or Modbus TCP communication, to a maximum of eight connections on the All-in-One CPU module/CAN J1939 All-in-One CPU module and 16 connections on the Plus CPU module.

FC6A Series MICROSmart Ethernet Port 1 and 2



Maintenance Communication Specifications

Item	Specifications/Functions
Cable	LAN cable (Cat 5. STP)
Maintenance communication functions	Monitor/change device values Download/upload user programs Download system software Run-time download

Detailed Operations for Maintenance Communication Functions

See the following in the "FC6A Series MICROSmart User's Manual".

- Monitor/change device values: Chapter 4 "Monitor Operation"
- Download/upload user programs: Chapter 4 "Download Program"
- Download system software: Appendix "Upgrade FC6A Series MICROSmart System Software"
- Run-time download: Chapter 5 "Online Edit"

Programming WindLDR

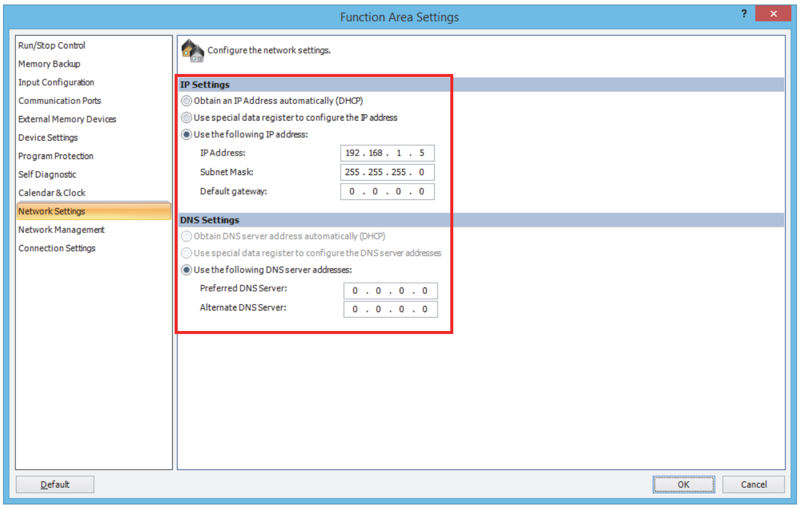
This section describes the procedures to configure the maintenance communication server for the Ethernet port 1 and 2, and to communicate with the FC6A Series MICROSmart via Ethernet.

Configure Maintenance Communication Server

1. On the **Configuration** tab, in the **Function Area Settings** group, click the following item.
   - All-in-One CPU module/CAN J1939 All-in-One CPU module
     Network Settings
   - Plus CPU module
     Ethernet Port 1 to configure Ethernet port 1
     Ethernet Port 2 to configure Ethernet port 2

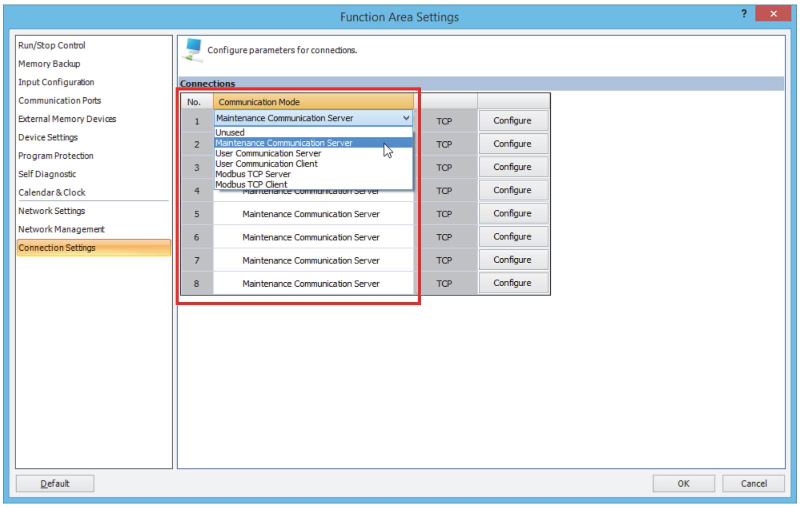
   The **Function Area Settings** dialog box appears.

2. Enter the IP address, subnet mask, and default gateway.

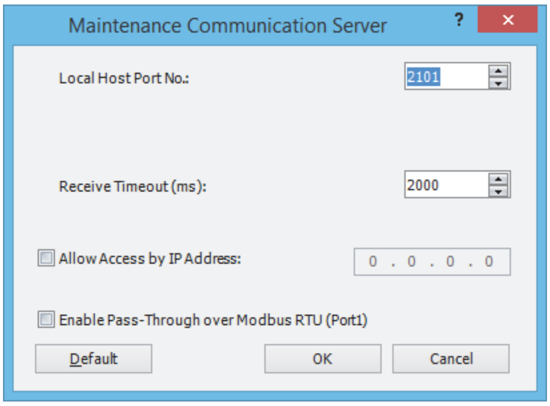


3. Click **Connection Settings**.

4. Click **Communication Mode** for the connection that will be used and select **Maintenance Communication Server**.
   The Maintenance Communication Server dialog box appears.



5. Set the parameters to match the communication settings of the PC or operator interface.



Local Host Port No.:	2101 (The port number that the FC6A Series MICROSmart uses for the maintenance communication server)
Receive Timeout (ms):	2000 (100 to 25500)
Allow Access by IP Address:	Disabled (Enabling this option makes it possible to prevent access from devices having any IP addresses other than the one entered.)

Enable Pass-Through over Modbus RTU (Port1):

Port 1 can be used as the pass-through port. In this case, set **Communication Mode** for port 1 to **Modbus RTU Master**. Specify a port number different from the port numbers of the other server connections as the local host port number of the connection used for the pass-through.

Note: The number of clients that can be connected to the FC6A Series MICROSmart simultaneously is one client per connection. If connections 1 to 8 are all set to the maintenance communication server, eight clients can connect to the FC6A Series MICROSmart at the same time.

6. Click **OK**.

This completes maintenance communications settings.

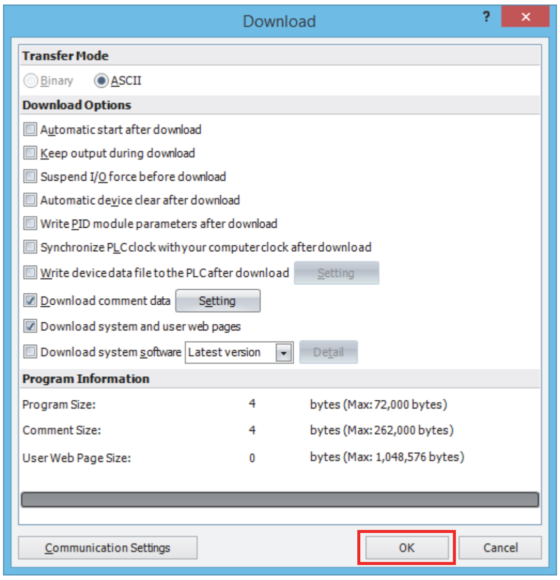
Download User Program and Confirm IP Addresses via USB Port

Before starting Ethernet communication, configure the function area settings and download the user program to the FC6A Series MICROSmart via USB.

7. Connect the PC and the FC6A Series MICROSmart using a USB cable.

8. From the WindLDR menu bar, select **Online** > **Transfer** > **Download**.
The **Download** dialog box appears.

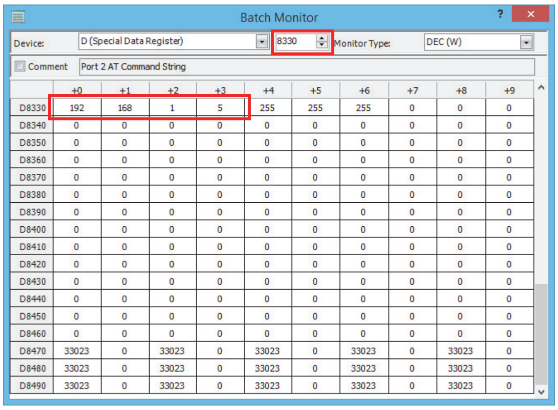
9. Click **OK**.
The user program is downloaded to the FC6A Series MICROSmart.



10. After the user program has been successfully downloaded, go to Monitor Mode to check the status of the FC6A Series MICROSmart. Select **Online** > **Monitor** > **Monitor** from the WindLDR menu bar.

11. From the WindLDR menu bar, select **Online** > **Monitor** > **Batch**.
The **Batch Monitor** dialog box appears.

12. Confirm that the IP address entered in step 2 is correctly shown in D8330 to D8333.



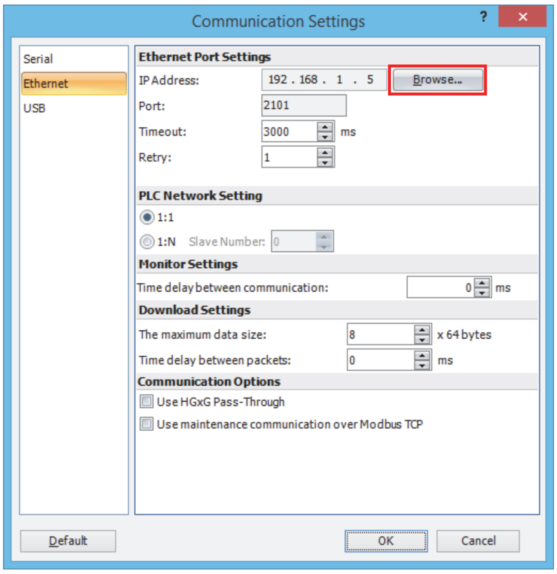
This concludes downloading the user program via the USB port and checking the IP address.

Monitor FC6A Series MICROSmart via the Ethernet Port 1 and 2

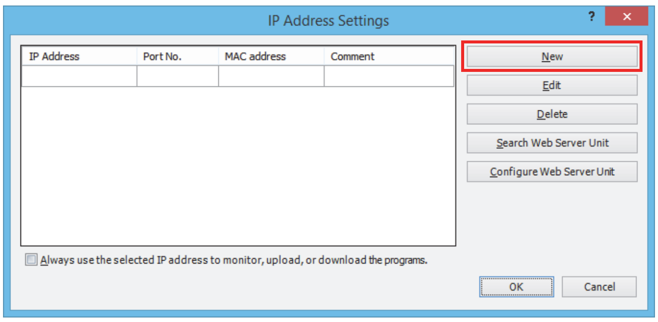
Monitor the FC6A Series MICROSmart via Ethernet using WindLDR.

13. From the WindLDR menu bar, select **Online** > **Communication** > **Set Up**.
The **Communication Settings** dialog box appears.

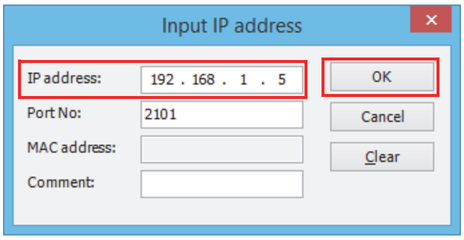
14. Select the **Ethernet** tab and click **Browse**.
The **IP Address Settings** dialog box appears.



15. Click **New**.
The **Input IP address** dialog box appears.

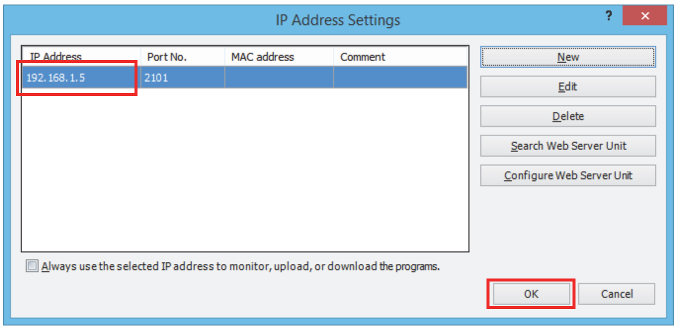


16. Enter the IP address entered in step 2 and click **OK**.



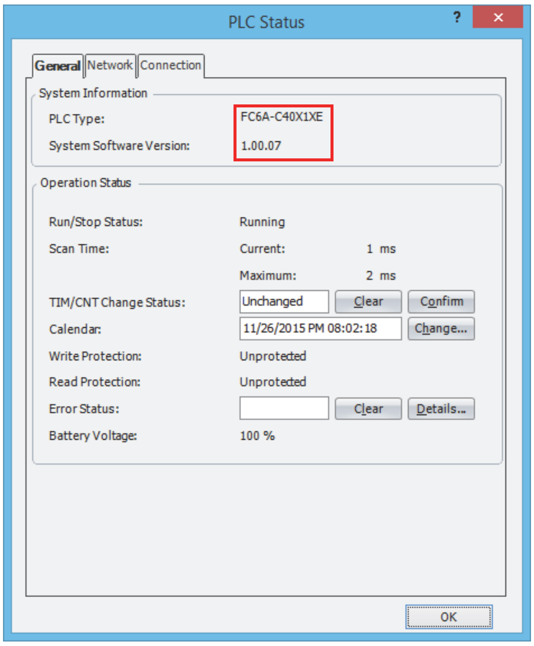
17. From the WindLDR menu bar, select **Online** > **Monitor** > **Monitor**.
The **IP Address Settings** dialog box appears.

18. Select the IP address you entered and click **OK**.



19. From the WindLDR menu bar, select **Online** > **PLC** > **Status**.
The **PLC Status** dialog box appears.

20. Check that the FC6A Series MICROSmart module type and system software version are displayed correctly.



Configuring the initial Ethernet setup for the maintenance communication server is now complete. You can download and upload user programs and monitor and change device values via Ethernet.

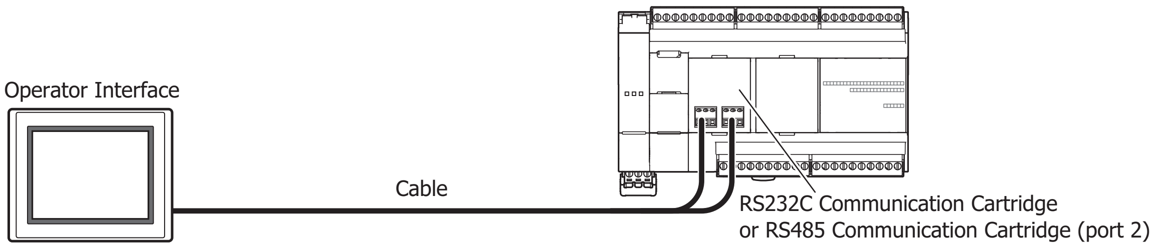
Maintenance Communication via Port 2 to 33

Expand the FC6A Series MICROSmart in the following manner to allow the FC6A Series MICROSmart to connect to a computer or operator interface with an RS232C or RS485 port and monitor and change device values.

- Connect an RS232C communication cartridge or RS485 communication cartridge to the cartridge slot of the All-in-One CPU module/CAN J1939 All-in-One CPU module.
- Connect an RS232C communication cartridge or RS485 communication cartridge to cartridge slot 2 of a cartridge base module added to the Plus CPU module.
- Connect an RS232C communication cartridge or RS485 communication cartridge to the cartridge slot of the HMI module added to the Plus CPU module.
- Connect a communication module to the FC6A Series MICROSmart.

For the communication cable, see Appendix "Cables" in the "FC6A Series MICROSmart User's Manual".

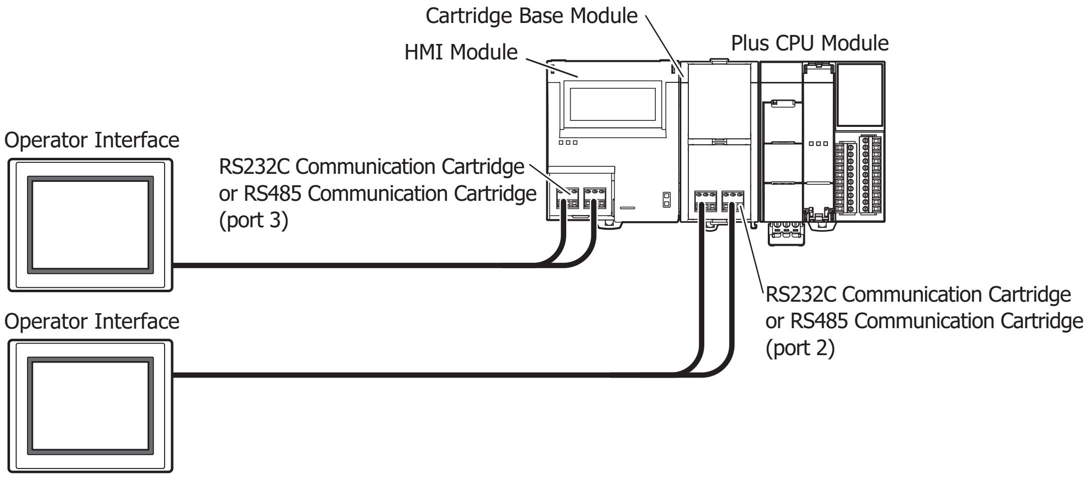
All-in-One CPU Module/CAN J1939 All-in-One CPU Module Port 2 and 3



Maintenance Communication Specifications

Item	Specifications/Functions
Cable	RS232C communication cartridge: Shielded multicore cable RS485 communication cartridge: Shielded twisted-pair cable
Maintenance Communication Function	Monitor/change device values

Plus CPU Module Port 2 and 3



Maintenance Communication Specifications

Item	Specifications/Functions
Cable	RS232C communication cartridge: Shielded multicore cable RS485 communication cartridge: Shielded twisted-pair cable
Maintenance Communication Function	Monitor/change device values

FC6A Series MICROSmart Port 4 to 33

The All-in-One CPU module and CAN J1939 All-in-One CPU module can be expanded with up to three communication modules to add ports 4 to 9.

The Plus CPU module can be expanded with up to 15 communication modules to add ports 4 to 33. For details on communication modules, see Chapter 2 "Communication Modules" in the "FC6A Series MICROSmart User's Manual".

Maintenance Communication Specifications

Item	Specifications/Functions
Cable	RS232C communication cartridge: Shielded multicore cable RS485 communication cartridge: Shielded twisted-pair cable
Maintenance Communication Function	Monitor/change device values

Detailed Operations for Maintenance Communication Functions

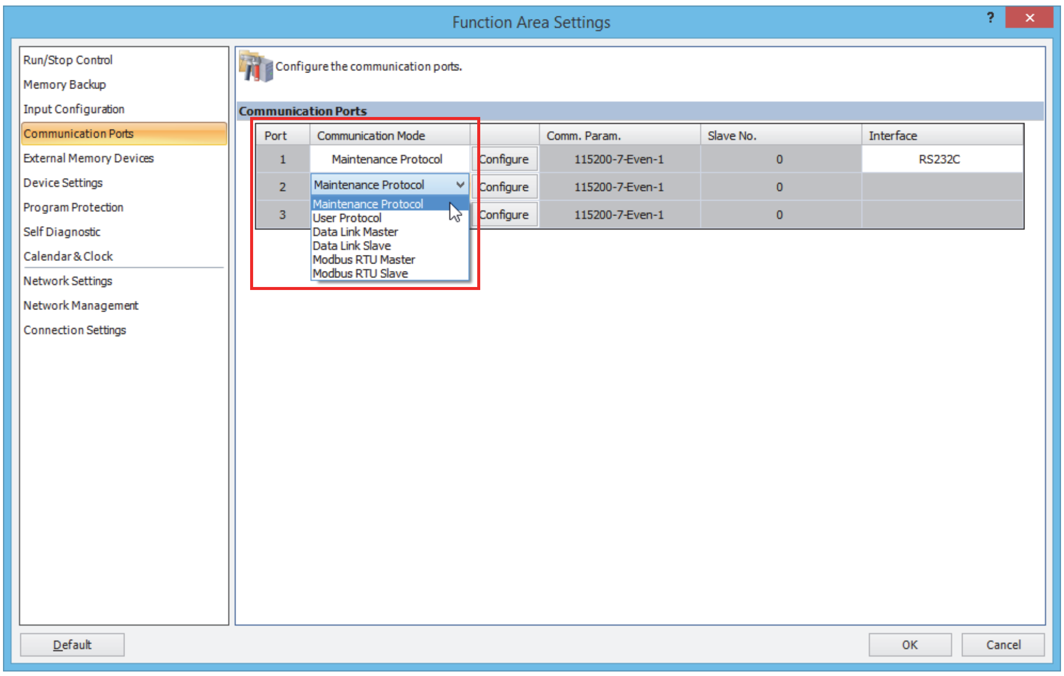
See the following in the "FC6A Series MICROSmart User's Manual".

- Monitor/change device values: Chapter 4 "Monitor Operation"

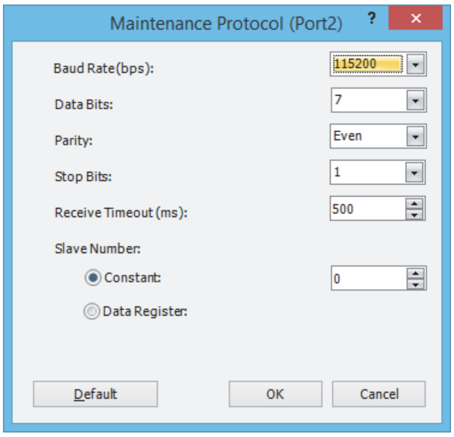
Programming WindLDR

Configure the settings for maintenance communication.

1. On the **Configuration** tab, in the **Function Area Settings** group, click **Comm. Ports**.
   The **Function Area Settings** dialog box is displayed.
2. Click **Communication Mode** for port 2 to 33 and select **Maintenance Protocol**.
   The **Maintenance Protocol (Port n)** dialog box is displayed.



3. Set the parameters to match the communication format used by the computer or operator interface.



Baud Rate (bps)	: 115200 bps (1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200)
Data Bits	: 7 (7 or 8)
Parity	: Even (None, Even, Odd)
Stop Bit	: 1 (1 or 2)
Receive Timeout	: 500 ms (10 ms to 2550 ms)
Slave Number	: 0 (0 to 31) Slave numbers are specified by constants or data registers.

Type	Details
Constant	Set within the range of 0 to 31
Data Register	Store the slave numbers 0 to 31 in the following special data registers Port 1: D8100 Port 2: D8102 Port 3: D8103 Port 4 to 9[*1]: D8040 to D8045 Port 10 to 33[*1*2]: D8735 to D8758

*1 When using a communication module.
*2 Plus CPU module only.

* Values not in parentheses are the default settings.

4. Click **OK**.

This concludes configuring the maintenance communications settings.

Maintenance Communication via Port 1 to 3 (Bluetooth)

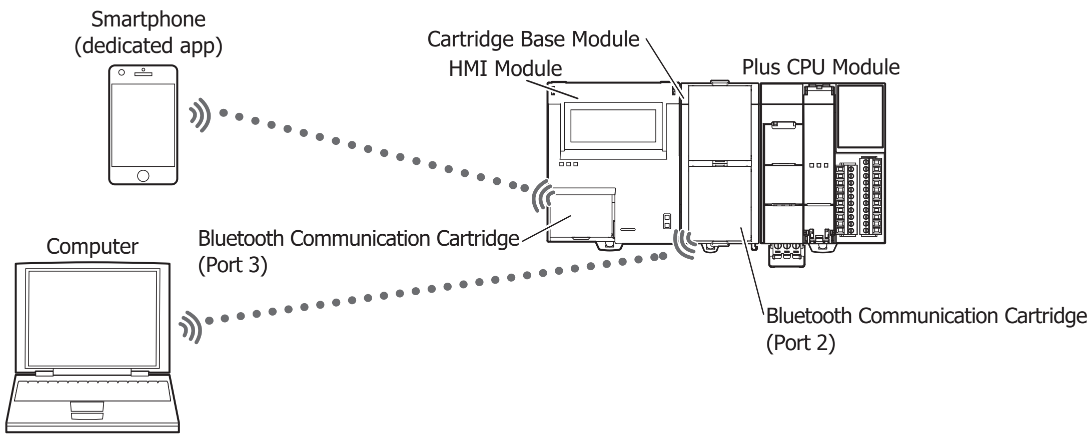
Computers or smart phones having Bluetooth interface can be connected to the FC6A Series MICROSmart, and monitoring and changing device values are possible. Bluetooth communication is possible by inserting a Bluetooth communication cartridge on the cartridge slot 1, 2, or 3 (Plus CPU module only).

All-in-One CPU Module/CAN J1939 All-in-One CPU Module Bluetooth

Maintenance Communication Specifications

Item	Specifications/Functions
Maintenance Communication Function	Monitor/change device values Download/upload user programs

Plus CPU Module Bluetooth



Maintenance Communication Specifications

Item	Specifications/Functions
Maintenance Communication Function	Monitor/change device values Download/upload user programs

Notes:

- The following functions are not supported by maintenance communication using a Bluetooth communication cartridge.
  - Download system software
  - Run-time download
- Bluetooth communication may fail and a download error may occur after a user program is downloaded and Bluetooth communication cartridge settings are changed.

Detailed Operations for Maintenance Communication Functions

For details, see the following in the "FC6A Series MICROSmart User's Manual".

- Monitor/change device values: Chapter 4 "Monitor Operation"

Programming WindLDR

For details, see "Bluetooth Communication" on page 9-1.

Maintenance Communication via HMI-Ethernet port

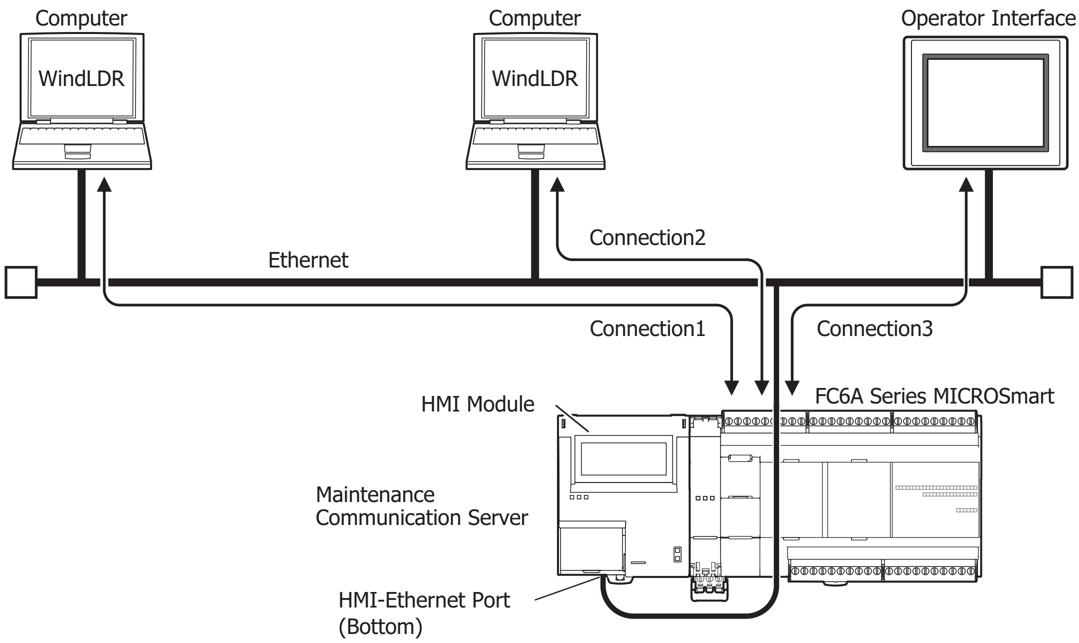
It is possible to use the HMI-Ethernet port of an HMI module connected to the CPU module to perform maintenance communication with network-enabled devices such as computers and operator interfaces.

It is possible to monitor and change FC6A Series MICROSmart device values from, download user programs from, and upload user programs to an external device connected to the network.

The HMI-Ethernet port can be used to add a maximum of eight connections to the connections of Ethernet port 1 (eight connections maximum), which is standard equipment on the All-in-One CPU module, and the connections of Ethernet port 1 and 2 (16 connections maximum), which are standard equipment on the Plus CPU module.

The maintenance communication server can be assigned to the expanded connections (up to eight) provided by the HMI-Ethernet port.

HMI Module HMI-Ethernet Port 1



Maintenance Communication Specifications

Item	Specifications/Functions
Cable	Ethernet cable (Cat 5. STP)
Maintenance communication functions	Monitor/change device values Download/upload user programs Download system software Run-time download

Detailed Operations for Maintenance Communication Functions

See the following in the "FC6A Series MICROSmart User’s Manual".

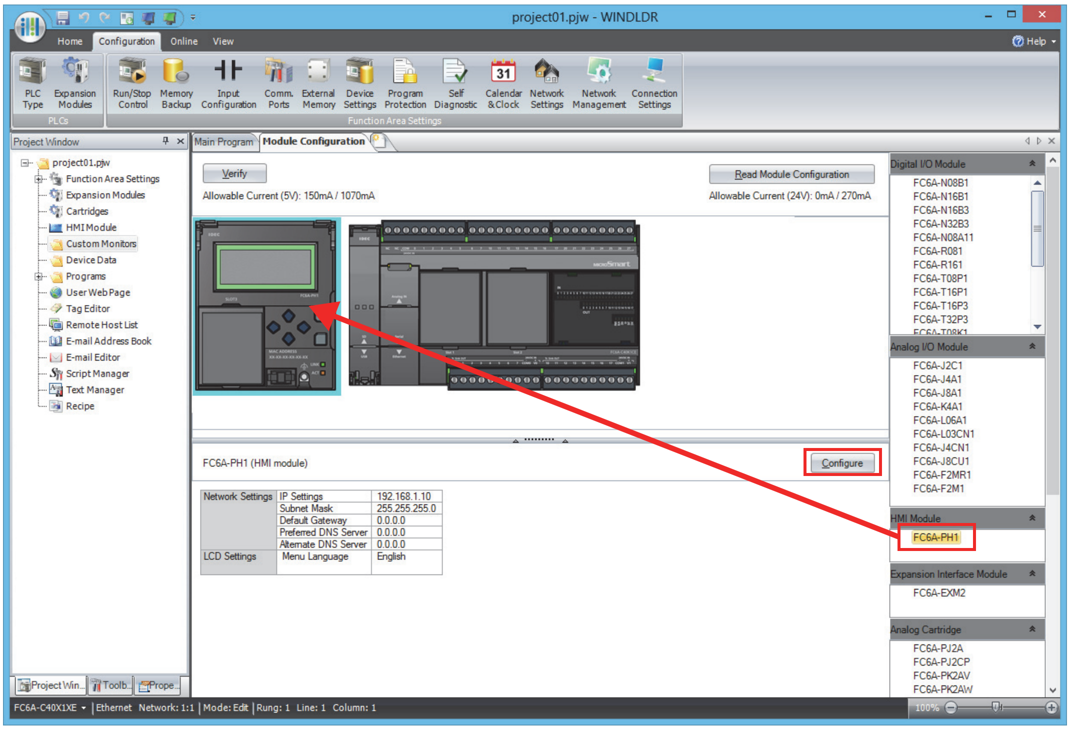
- Monitor/change device values: Chapter 4 "Monitor Operation"
- Download/upload user programs: Chapter 4 "Download Program"
- Download system software: Appendix "Upgrade FC6A Series MICROSmart System Software"
- Run-time download: Chapter 5 "Online Edit"

Programming WindLDR

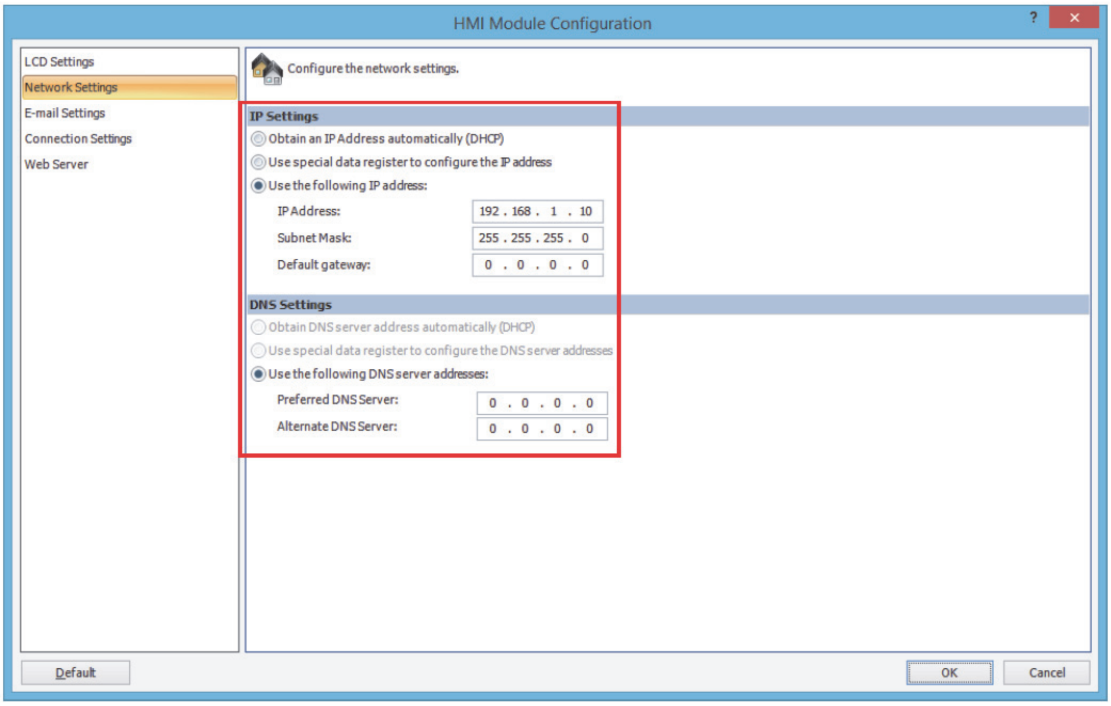
Configure the settings for maintenance communication.

1. Use the Module Configuration Editor to configure the HMI module network settings and connection settings. On the **Configuration** tab, in the **PLCs** group, select **Expansion Modules**.
2. Click the inserted HMI module in the module configuration area and click **Configure**. The **HMI Module Configuration** dialog box is displayed.

Note: You can also display the **HMI Module Configuration** dialog box by double-clicking **HMI Module** in the Project Window.

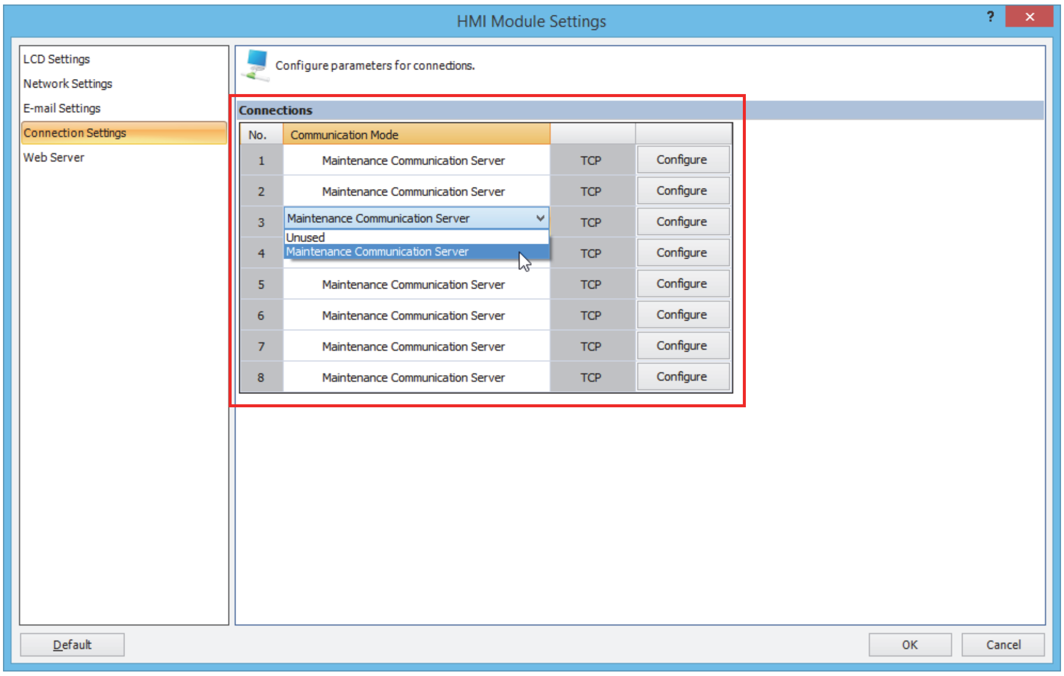


3. Click the **Network Settings** tab, and then configure the **IP Settings** and **DNS Settings**.

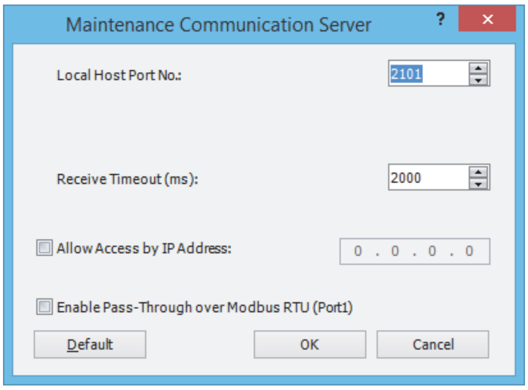


4. Click **Connection Settings**.

5. Click **Communication Mode** for the port to be used and select communication mode of the connection to use.



6. Set the parameters to match the communication format used by the computer or operator interface.



Local Host Port No.	:	2101 (The port number that the FC6A Series MICROSmart uses for the maintenance communication server)
Receive Timeout (ms)	:	2000 (100 to 25500)
Allow Access by IP Address	:	Disabled (Enabling this option makes it possible to prevent access from devices having any IP addresses other than the one entered.)

Note: The number of clients that can be connected to the FC6A Series MICROSmart simultaneously is one client per connection. If connections 1 to 8 are all set to the maintenance communication server, eight clients can connect to the FC6A Series MICROSmart at the same time.

7. Click **OK**.

This completes maintenance communications settings.

5: USER COMMUNICATION INSTRUCTIONS

Introduction

This chapter describes user communication to send and receive specific data by converting it to data types for external devices connected to the FC6A Series MICROSmart.

The user communication instructions are used to execute user communication.

The user communication instructions differ according to the communication interface that will be used.

- When performing user communication with an external device using port 1 to port 33 via serial communication (RS232C/RS485)
  When performing user communication with an external device using port 1 to port 3 via serial communication (Bluetooth)
  "TXD (Transmit)" on page 5-2
  "RXD (Receive)" on page 5-10
- When performing user communication with an external device using Ethernet port 1 and 2 via Ethernet communication
  "ETXD (User Communication Transmit over Ethernet)" on page 5-23
  "ERXD (User Communication Receive over Ethernet)" on page 5-23

Communication Ports Used in User Communication

Supported models and communication ports are as follows.

Communication Port	All-in-One CPU Module			CAN J1939 All-in-One CPU Module	Plus CPU Module	
	16-I/O Type	24-I/O Type	40-I/O Type		Plus 16-I/O Type	Plus 32-I/O Type
Port 1	Yes	Yes	Yes	—	Yes*2	Yes*2
Port 2	Yes*1	Yes*1	Yes*1	Yes*1	Yes*2	Yes*2
Port 3	—	—	Yes*1	Yes*1	Yes*3	Yes*3
Port 4 to 33	Yes (Port 4 to 9)*4				Yes (Port 4 to 33)*5	
Ethernet Port 1	Yes	Yes	Yes	Yes	Yes	Yes
Ethernet Port 2	—	—	—	—	Yes	Yes
HMI-Ethernet Port	—	—	—	—	—	—

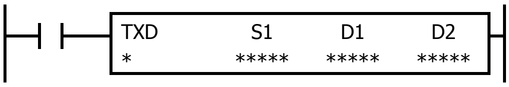
*1 When a communication cartridge is connected.
*2 When a cartridge base module and communication cartridge are connected.
*3 When the HMI module and a communication cartridge are connected.
*4 Up to three communication modules can be connected to the All-in-One CPU module and CAN J1939 All-in-One CPU module, expanding the communication ports 4 to 9.
*5 Up to 15 communication modules can be connected to the Plus CPU module, expanding the communication ports 4 to 33.

User Communication Settings

The user communication settings supported by each communication port are as follows.

- Serial communication with an external device connected to port 1 to port 33 (RS232C/RS485)
  Serial communication with an external device connected to port 1 to port 3 (Bluetooth)
  "User Communication via Serial Communication" on page 5-24
- Ethernet communication with an external device connected by the Ethernet port 1 and 2
  "User Communication via Ethernet Communication" on page 5-40

TXD (Transmit)



The transmit data is converted to the set data type and transmitted to the external device using port 1 to 33.

Valid Devices

Device	Function	I	Q	M	R	T	C	D	P	Constant	Repeat
S1 (Source 1)	Transmit data	—	—	—	—	—	—	X	—	X	—
D1 (Destination 1)	Transmit completion output	—	X	X*1	—	—	—	—	—	—	—
D2 (Destination 2)	Transmit status register	—	—	—	—	—	—	X*2	—	—	—

For valid device address ranges, see "Devices" on page 2-1.

*1 Special internal relays cannot be designated as D1.
*2 Special data registers cannot be used.

Transmit data designated by device S1 can be a maximum of 1,536 bytes.

When transmission is complete, an output or internal relay, designated by device D1, is turned on.

Destination 2 occupies two consecutive data registers starting with the device designated by D2. The transmit status data register stores the status of transmission and error code. The next data register stores the byte count of transmitted data. The same data registers cannot be used as transmit status registers for TXD1 through TXD33 instructions and receive status registers for RXD1 through RXD33 instructions.

The TXD instructions cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the FC6A Series MICROSmart. For details about the user program execution errors, see Chapter 3 "User Program Execution Errors" in the "FC6A Series MICROSmart Ladder Programming Manual".

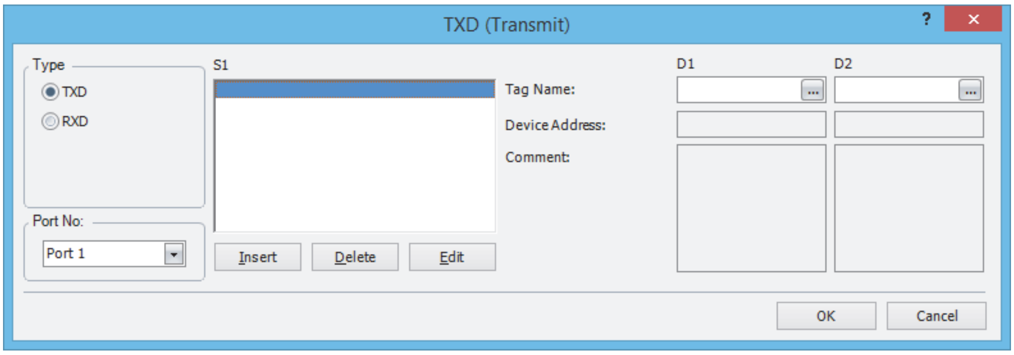
Notes:

- For details on user communication over serial communication, see "User Communication via Serial Communication" on page 5-24.
- For details on user communication over Bluetooth communication, see "User Communication via Serial Communication (Bluetooth)" on page 5-39.

Precautions for Programming TXD Instruction

- The FC6A Series MICROSmart has five formatting areas each for executing TXD1 through TXD33 instructions, so five instructions each of TXD1 through TXD33 can be processed at the same time. If inputs to more than five of the same TXD instructions are turned on at the same time, an error code is set to the transmit status data register, designated by device D2, in the excessive TXD instructions that cannot be executed.
- If the input for a TXD instruction turns on on while another TXD instruction is executed, the subsequent TXD instruction is executed 2 scan times after the preceding TXD instruction is completed.
- Since TXD instructions are executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used.
- The data register used to store the transmit/receive status and the transmit/receive data byte count must be different for the TXD instruction and the RXD instruction.
- The TXD instruction cannot be used in an interrupt program. If used, a user program execution error occurs, the execution of the instruction is canceled, and the next instruction is executed. For details about the user program execution errors, see Chapter 3 "User Program Execution Errors" in the "FC6A Series MICROSmart Ladder Programming Manual".

User Communication Transmit Instruction Dialog Box in WindLDR



Selections and Devices in Transmit Instruction Dialog Box

Type	**TXD**	Transmit instruction
	RXD	Receive instruction
Port No.	**Port 1 - Port 33**	Transmit user communication from port 1 (TXD1) through port 33 (TXD33)
S1	**Source 1**	Enter the data to transmit in this area. Transmit data can be constant values (character or hexadecimal), data registers, or BCC.
D1	**Destination 1**	Transmit completion output can be an output or internal relay.
D2	**Destination 2**	Transmit status register can be data register. The next data register stores the byte count of transmitted data.

Transmit Data

Transmit data is designated by source device S1 using constant values or data registers. BCC code can also be calculated automatically and appended to the transmit data. One TXD instruction can transmit a maximum of 1,500 bytes of data.

S1 (Source 1)

Transmit Data	Conversion Type	Transmit Digits (Bytes)	Repeat	BCC Calculation	Calculation Start Position
Constant (Character)	No conversion	1	—	—	—
Constant (Hexadecimal)					
Data Register	A: Binary to ASCII B: BCD to ASCII –: No conversion	1-4 1-5 1-2	1-99	—	—
BCC	A: Binary to ASCII –: No conversion	1-2	—	X: XOR A: ADD C: Add-2comp M: Modbus ASCII M: Modbus RTU	1-15

Note: Total byte count of the transmit data is shown in S1 of TXD instruction on the ladder diagram of WindLDR.

Designating Constant as S1

When a constant value is designated as source device S1, one-byte data is transmitted without conversion. The valid transmit data value depends on the data bits selected in the Communication Parameters dialog box. These data bits are set in **Configuration** > **Comm. Ports**, followed by selecting **User Protocol** in Port 1 through Port 33 list box and clicking the **Configure** button. When 7 data bits are selected as default, 00h through 7Fh is transmitted. When 8 data bits are selected, 00h through FFh is transmitted. Constant values are entered in character or hexadecimal notation into the source data.

Constant (Character)

Any character available on the computer keyboard can be entered. One character is counted as one byte.

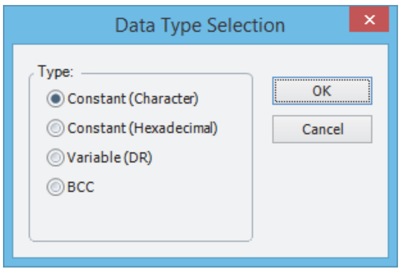
Constant (Hexadecimal)

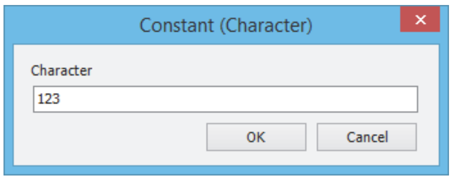
Use this option to enter the hexadecimal code of any ASCII character. ASCII control codes NUL (00h) through US (1Fh) can also be entered using this option.

Example:

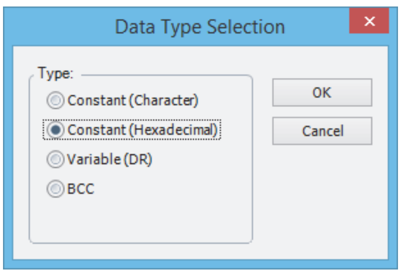
The following example shows two methods to enter 3-byte ASCII data "1" (31h), "2" (32h), "3" (33h).

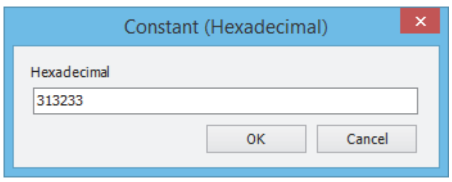
(1) Constant (Character)





(2) Constant (Hexadecimal)





Designating Data Register as S1

When a data register is designated as source device S1, conversion type and transmit digits must also be assigned. The data stored in the designated data register is converted and the assigned quantity of digits of the resultant data is transmitted. The conversion types that are available are: Binary to ASCII, BCD to ASCII, and no conversion.

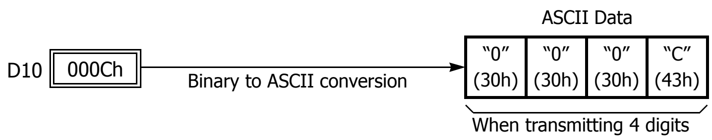
When repeat is designated, data of data registers as many as the repeat cycles are transmitted, starting with the designated data register. Repeat cycles can be up to 99.

Conversion Type

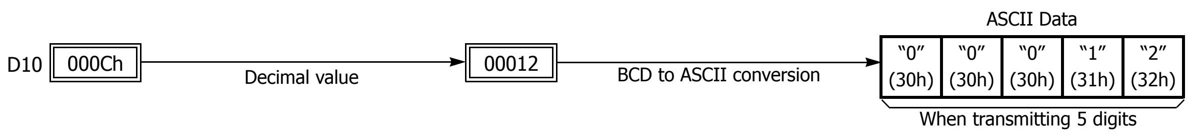
The transmit data is converted according to the designated conversion type as described below:

Example: D10 stores 000Ch (12)

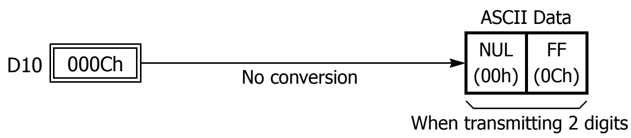
(1) Binary to ASCII conversion



(2) BCD to ASCII conversion



(3) No conversion

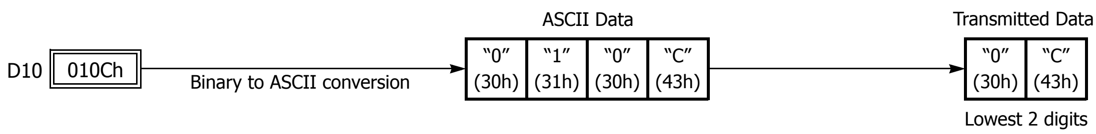


Transmit Digits (Bytes)

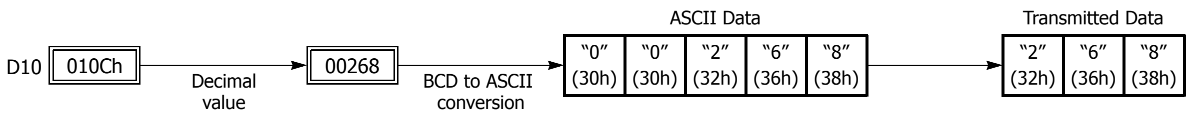
After conversion, the transmit data is taken out in specified digits. Possible digits depend on the selected conversion type.

Example: D10 stores 010Ch (268)

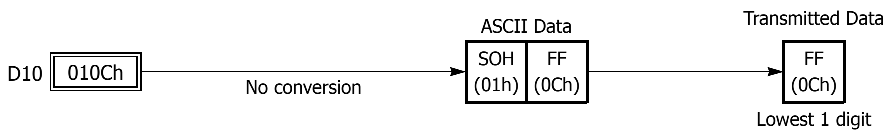
(1) Binary to ASCII conversion, Transmit digits = 2



(2) BCD to ASCII conversion, Transmit digits = 3



(3) No conversion, Transmit digits = 1



Repeat Cycles

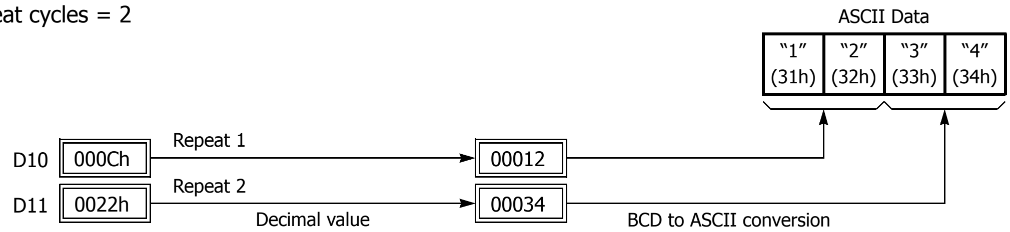
When a data register is assigned to repeat, as many consecutive data registers, as the repeat cycles, are used to transmit data in the same conversion type and transmit digits.

Example:

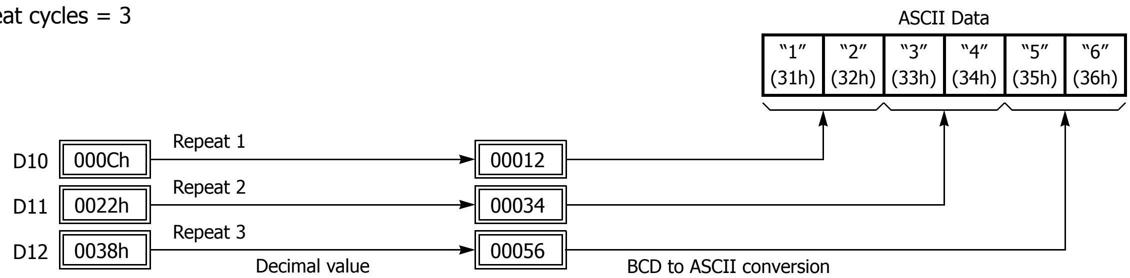
D10	000Ch	Data register No.:	D10
D11	0022h	Transmit digits:	2
D12	0038h	Conversion type:	BCD to ASCII

Data from data registers starting with D10 is converted in BCD to ASCII and is transmitted according to the designated repeat cycles.

(1) Repeat cycles = 2

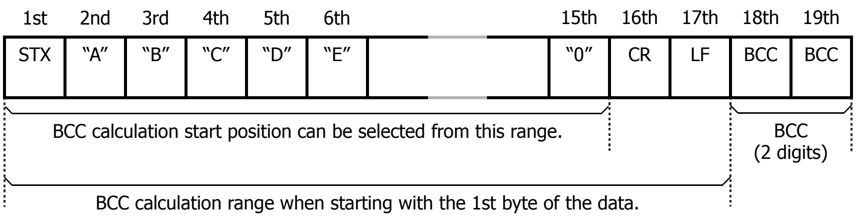


(2) Repeat cycles = 3



BCC (Block Check Character)

Block check characters can be appended to the transmit data. The start position for the BCC calculation can be selected from the first byte through the 15th byte. The BCC can be 1 or 2 digits.



BCC Calculation Start Position

The start position for the BCC calculation can be specified from the first byte through the 15th byte. The BCC is calculated for the range starting at the designated position up to the byte immediately before the BCC of the transmit data.

Example: Transmit data consists of 17 bytes plus 2 BCC digits.

(1) Calculation start position = 1

1st	2nd	3rd	4th	5th	6th		15th	16th	17th	18th	19th
STX	"A"	"B"	"C"	"D"	"E"		"0"	CR	LF	BCC	BCC

BCC calculation range (1st through 17th) — BCC (2 digits)

(2) Calculation start position = 2

1st	2nd	3rd	4th	5th	6th		15th	16th	17th	18th	19th
STX	"A"	"B"	"C"	"D"	"E"		"0"	CR	LF	BCC	BCC

BCC calculation range (2nd through 17th) — BCC (2 digits)

BCC Calculation Formula

BCC calculation formula can be selected from XOR (exclusive OR), ADD (addition), ADD-2comp, Modbus ASCII, or Modbus RTU.

Example: Conversion results of transmit data consist of 41h, 42h, 43h, and 44h.

ASCII Data

"A" (41h)	"B" (42h)	"C" (43h)	"D" (44h)

(1) BCC calculation formula = XOR
    Calculation result = 41h ⊕ 42h ⊕ 43h ⊕ 44h = 04h

(2) BCC calculation formula = ADD
    Calculation result = 41h + 42h + 43h + 44h = 10Ah → 0Ah (Only the last 1 or 2 digits are used as BCC.)

(3) BCC calculation formula = ADD-2comp
    Calculation result = FEh, F6h (2 digits without conversion)

(4) BCC calculation formula = Modbus ASCII
    Calculation result = 88 (ASCII)

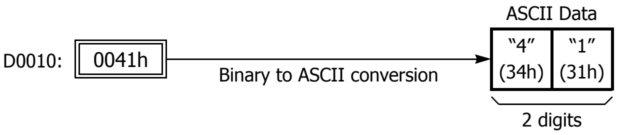
(5) BCC calculation formula = Modbus RTU
    Calculation result = 85h 0Fh (binary)

Conversion Type

The BCC calculation result is converted or not according to the designated conversion type as described below:

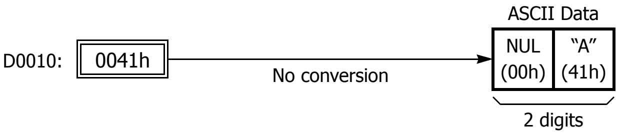
Example: BCC calculation result is 0041h.

(1) Binary to ASCII conversion



Note: On WindLDR, Modbus ASCII is defaulted to binary to ASCII conversion.

(2) No conversion



Note: On WindLDR, Modbus RTU is defaulted to no conversion.

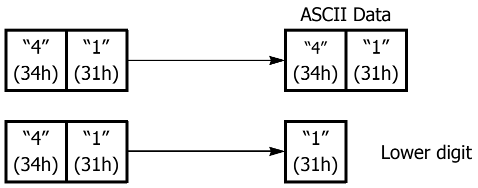
BCC Digits (Bytes)

The quantity of digits (bytes) of the BCC code can be selected from 1 or 2.

Example:

(1) BCC digits = 2

(2) BCC digits = 1



Note: On WindLDR, Modbus ASCII and Modbus RTU are defaulted to 2 digits.

D1 (Destination 1)

Set an internal relay or an output for the transmit completion output.

When the start input for a TXD instruction is turned on, preparation for transmission is initiated, followed by data transmission. When this sequence of operations is complete, the transmit completion output is turned on.

D2 (Destination 2)

Set the data register in which to store the transmit status and the transmit data byte count.

The transmit status is stored in D2+0 and the transmit data byte count is stored in D2+1.

D2+0 (Transmit Status)

Transmit Status Code	Status	Description
16	Preparing transmission	From turning on the start input for a TXD instruction, until the transmit data is stored in the internal transmit buffer
32	Transmitting data	From enabling data transmission by an END processing, until all data transmission is completed
48	Data transmission complete	From completing all data transmission, until the END processing is completed for the TXD instruction
64	Transmit instruction complete	All transmission operation is completed and the next transmission is made possible

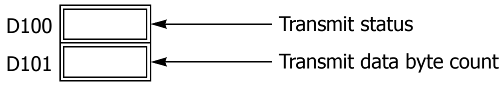
If the transmit status code is other than shown above, a transmit instruction error is suspected. See "User Communication Error" on page 5-52.

D2+1 (Transmit Digits (Bytes))

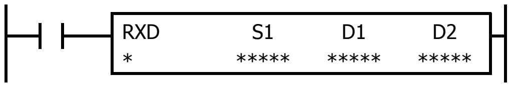
The transmit data byte count is stored in the data register after that set with D2 (D2+1). BCC data is also included in the transmit data byte count.

The data register next to the device designated for transmit status stores the byte count of data transmitted by the TXD instruction. When BCC is included in the transmit data, the byte count of the BCC is also included in the transmit data byte count.

Example: Data register D100 is designated as a device for transmit status.



RXD (Receive)



When input is on, data from an RS232C/RS485 remote terminal received by port 1 through port 33 is converted and stored in data registers according to the receive format assigned by S1.

Valid Devices

Device	Function	I	Q	M	R	T	C	D	P	Constant	Repeat
S1 (Source 1)	Receive format	—	—	—	—	—	—	X	—	X	—
D1 (Destination 1)	Receive completion output	—	X	X*1	—	—	—	—	—	—	—
D2 (Destination 2)	Receive status	—	—	—	—	—	—	X*2	—	—	—

For valid device address ranges, see "Devices" on page 2-1.

*1 Special internal relays cannot be designated as D1.
*2 Special data registers cannot be used.

Receive format designated by device S1 can be a maximum of 1,536 bytes.

When data receive is complete, an output or internal relay, designated by device D1, is turned on.

Destination 2 occupies two consecutive data registers starting with the device designated by D2. The receive status data register stores the status of data receive and error code. The next data register stores the byte count of received data. The same data registers can not be used as transmit status registers for TXD1 through TXD33 instructions and receive status registers for RXD1 through RXD33 instructions.

The RXD instructions cannot be used in an interrupt program. If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the FC6A Series MICROSmart. For details about the user program execution errors, see Chapter 3 "User Program Execution Errors" in the "FC6A Series MICROSmart Ladder Programming Manual".

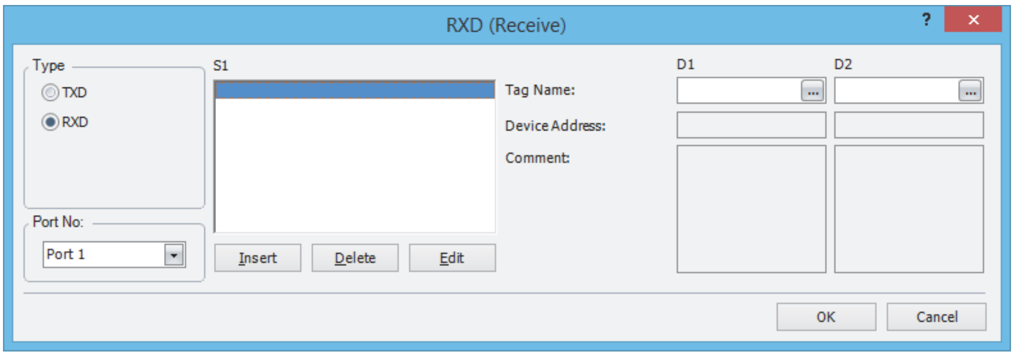
Notes:

- For details on user communication over serial communication, see "User Communication via Serial Communication" on page 5-24.
- For details on user communication over Bluetooth communication, see "User Communication via Serial Communication (Bluetooth)" on page 5-39.

Precautions for Programming the RXD Instruction

- The FC6A Series MICROSmart can execute a maximum of five instructions each of RXD1 through RXD33 that have a start delimiter at the same time. If a start delimiter is not programmed in RXD1 through RXD33 instructions, the FC6A Series MICROSmart can execute only one instruction each of RXD1 through RXD33 at a time. If the start input for a RXD1 through RXD33 instruction is turned on while another RXD1 through RXD33 instruction, without a start delimiter is executed, a user communication error occurs.
- Since RXD instructions are executed in each scan while input is on, a pulse input from a SOTU or SOTD instruction should be used.
- Once the input to the RXD instruction is turned on, the RXD is activated and ready for receiving incoming communication even after the input is turned off. When the RXD completes data receiving, the RXD is deactivated if the input to the RXD is off. Or, if the input is on, the RXD is made ready for receiving another communication. Special internal relays are available to deactivate all RXD instructions waiting for incoming communication. For user communication receive instruction cancel flags, see "User Communication Receive Instruction Cancel Flag" on page 5-22.

User Communication Receive Instruction Dialog Box in WindLDR



Selections and Devices in Receive Instruction Dialog Box

Type	**TXD**	Transmit instruction
	RXD	Receive instruction
Port No.	**Port 1 - Port 33**	Receive user communication to port 1 (RXD1) through port 33 (RXD33)

S1	**Source 1**	Enter the receive format in this area. The receive format can include a start delimiter, data register to store incoming data, constants, end delimiter, BCC, and skip.
D1	**Destination 1**	Receive completion output can be an output or internal relay.
D2	**Destination 2**	Receive status register can be data register. The next data register stores the byte count of received data.

Receive Format

Receive format, designated by source device S1, specifies data registers to store received data, data digits for storing data, data conversion type, and repeat cycles. A start delimiter and an end delimiter can be included in the receive format to discriminate valid incoming communication. When some characters in the received data are not needed, "skip" can be used to ignore a specified number of characters. BCC code can also be appended to the receive format to verify the received data. One RXD instruction can receive 1,500 bytes of data at the maximum.

S1 (Source 1)

Receive Format	Receive Digits (Bytes)	Conversion Type	Repeat	BCC Calculation	Calculation Start Position	Skip Bytes	Delimiter
Data Register	1-4 1-5 1-2	A:ASCII to Binary B:ASCII to BCD –:No conversion	1-99	—	—	—	Hex ASCII
Start Delimiter	—	No conversion	—	—	—	—	
End Delimiter	—	No conversion	—	—	—	—	
Constant for Verification	—	No conversion	—	—	—	—	
BCC	1-2	A:Binary to ASCII –:No conversion	—	X:XOR A:ADD C:Add-2comp M:Modbus ASCII M:Modbus RTU	1-15	—	
Skip	—	—	—	—	—	1-99	

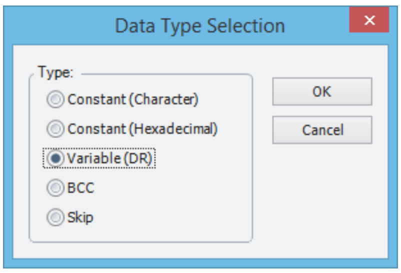
Note: Total byte count of the receive format is shown in S1 of RXD instruction on the ladder diagram of WindLDR.

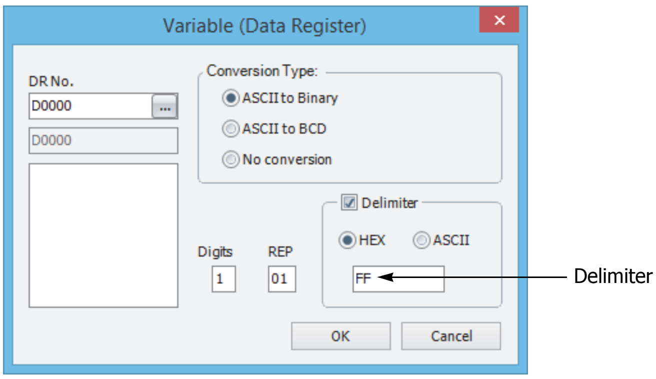
Designating Data Register as S1

When a data register is designated as source device S1, receive digits and conversion type must also be assigned. The received data is divided into blocks of specified receive digits, converted in a specified conversion type, and stored to the designated data registers. Conversion types are available in ASCII to Binary, ASCII to BCD, and no conversion.

When repeat is designated, received data is divided, converted, and stored into as many data registers as the repeat cycles, starting with the designated data register. There can be up to 99 repeat cycles.

When a data register is designated as source device S1, a delimiter can be included in the data register designation to end receiving communication.



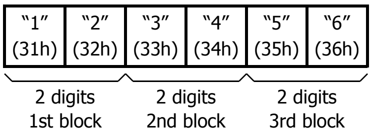


Receive Digits

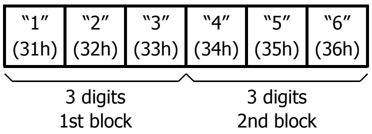
The received data is divided into blocks of specified receive digits before conversion as described below:

Example: Received data of 6 bytes are divided in different receive digits. (Repeat is also designated.)

(1) Receive digits = 2



(2) Receive digits = 3

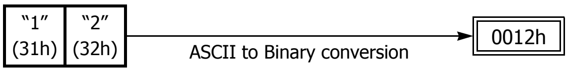


Conversion Type

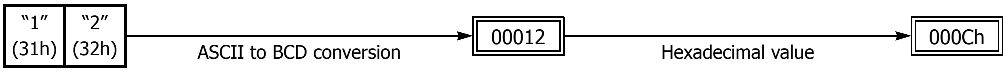
The data block of the specified receive digits is then converted according to the designated conversion type as described below:

Example: Received data has been divided into a 2-digit block.

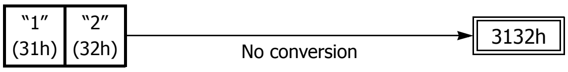
(1) ASCII to Binary conversion



(2) ASCII to BCD conversion



(3) No conversion

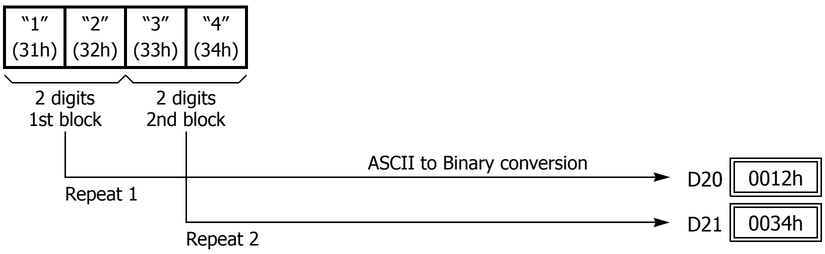


Repeat Cycles

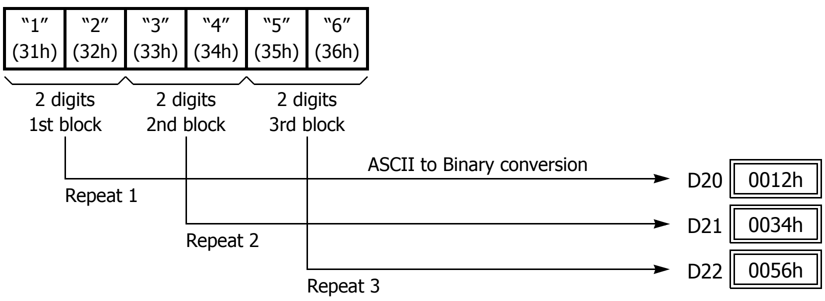
When a data register is assigned to repeat, the received data is divided and converted in the same way as specified, and the converted data is stored to as many consecutive data registers as repeat cycles.

Example: Received data of 6 bytes is divided into 2-digit blocks, converted in ASCII to Binary, and stored to data registers starting at D20.

(1) Repeat cycles = 2



(2) Repeat cycles = 3



Delimiter

A delimiter for the data register in the receive format can be assigned. Using a delimiter, incoming data of variable length can be received and stored to data registers.

Delimiter	How the incoming data is stored to data registers
Designated	The incoming data is stored to data registers until all the data specified with receive digits, conversion type, and repeat is processed or the specified delimiter is received.
No delimiter	The incoming data is stored to data registers until all the data specified with receive digits, conversion type, and repeat is processed.

Note: Delimiters for data registers can be used in the receive format of RXD instructions only.

Designating Constant as Start Delimiter

A start delimiter can be programmed at the first byte in the receive format of a RXD instruction; the FC6A Series MICROSmart will recognize the beginning of valid communication, although a RXD instruction without a start delimiter can also be executed.

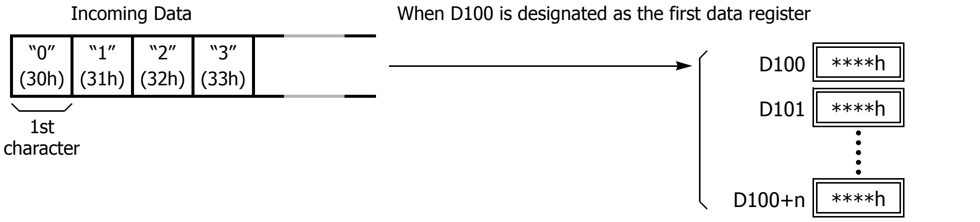
When a constant value is designated at the first byte of source device S1, the one-byte data serves as a start delimiter to start the processing of the received data.

A maximum of five instructions each of RXD1 through RXD3 with different start delimiters can be executed at the same time. When the first byte of the incoming data matches the start delimiter of a RXD instruction, the received data is processed and stored according to the receive format specified in the RXD instruction. If the first byte of the incoming data does not match the start delimiter of any RXD instruction that is executed, the FC6A Series MICROSmart discards the incoming data and waits for the next communication.

When a RXD instruction without a start delimiter is executed, any incoming data is processed continuously according to the receive format. Only one instruction each of RXD1 through RXD3 without a start delimiter can be executed at a time. If start inputs to two or more RXD instructions without a start delimiter are turned on simultaneously, the one at the smallest address is executed and the corresponding completion output is turned on.

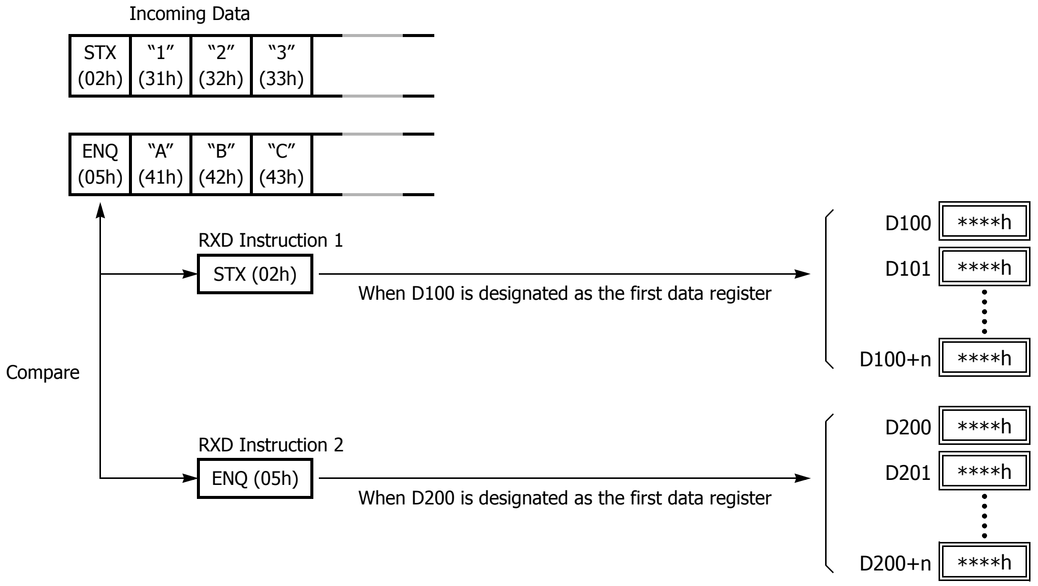
Example:

(1) When a RXD instruction without a start delimiter is executed



The incoming data is divided, converted, and stored to data registers according to the receive format.

(2) When RXD instructions with start delimiters STX (02h) and ENQ (05h) are executed



The incoming data is divided, converted, and stored to data registers according to the receive format.
Start delimiters are not stored to data registers.

Multi-byte Start Delimiter

A start delimiter can be programmed at the first bytes in the receive format of a RXD instruction; the FC6A Series MICROSmart will recognize the beginning of valid communication, although a RXD instruction without a start delimiter can also be executed. A maximum of 5 consecutive constants that are either character or hexadecimal from the first byte of the receive format are considered a multi-byte start delimiter.

If a RXD instruction with a start delimiter is executed and another RXD instruction with the same start delimiter is executed, user communication error code 5 is stored in the data register designated as the receive status of the second RXD instruction. When the error occurs, the second RXD instruction is canceled, and the first RXD instruction executed is kept.

If a multi-byte start delimiter is assigned, and the incoming data does not match the entire multi-byte start delimiter, the FC6A Series MICROSmart discards the incoming data and waits for the next communication.

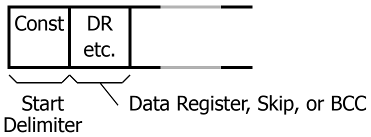
When the first one byte is received, a timer is started to monitor the interval between incoming data even when a multi-byte start delimiter is assigned. If data is not received in the period specified for the receive timeout value after receiving one byte of data, a receive timeout error occurs, and user communication error code 11 is stored in the status data register.

Examples: Multi-byte Start Delimiter

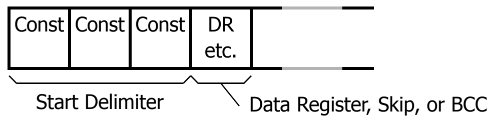
Multi-byte start delimiter is determined in the structure of the Receive Format. The following examples show how multi-byte start delimiter is determined.

- **Constants are followed by data register, skip, or BCC**

Receive Format



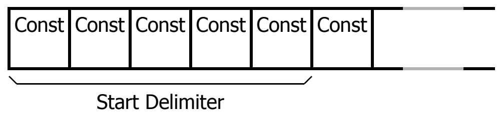
Receive Format



Note: Constants following data register, skip, or BCC are not considered start delimiter even if these are in the first five bytes of the receive format.

- **More than 5 constants are specified from the first byte**

Receive Format

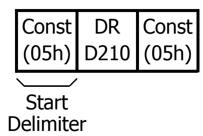


Note: Constants that are neither start delimiters or end delimiters are considered constants for verification. See "Constant for Verification" on page 5-17.

Example: Start Delimiter Duplication Error

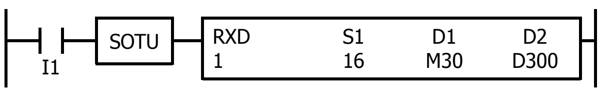
When input I0 is turned on, the first RXD instruction is executed and status code 32 is stored in the receive status D200, indicating the RXD instruction is waiting for the incoming data. When input I1 is turned on, another RXD instruction is executed, but since two RXD instructions have the same start delimiter, the second RXD instruction is not executed, and user communication error code 5 is stored in the receive status D300.

Receive Format S1

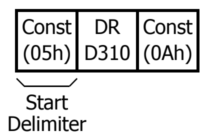


Communication port:	Port 1
Receive completion output:	M20
Receive status register:	D200
Receive data byte count:	D201

D200	32
M20	OFF



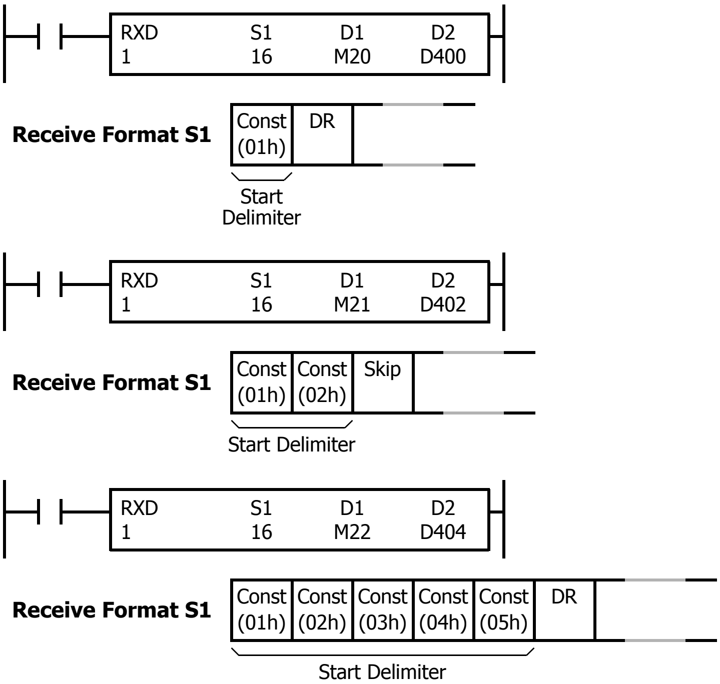
Receive Format S1



Communication port:	Port 1
Receive completion output:	M30
Receive status register:	D300
Receive data byte count:	D301

D300	5
M30	OFF

Note: If you execute two or more RXD instructions with multi-byte start delimiters simultaneously, the start delimiters of those RXD instructions must be unique from the others. When the length of the multi-byte start delimiters of the RXD instructions is different, the length of the shortest one is applied to check the duplicated start delimiters. The start delimiter of any of two RXD instructions in the following RXD instructions are considered the same.

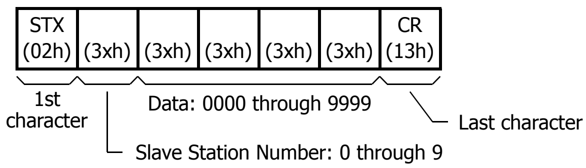


Example: Using Multi-byte Start Delimiter

The following example shows the advantages of using a multi-byte start delimiter rather than a single-byte start delimiter. A RXD instruction processes incoming data from the master station. The incoming data is sent to multiple slave stations 0 through 9, and the local slave station number is 1. Therefore, incoming data from the master station must be received only when the incoming data is sent for the slave station 1.

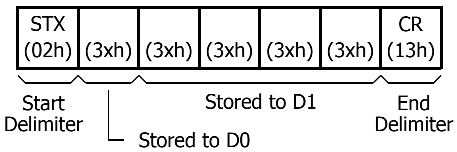
- **Incoming data**

Incoming data consists of start delimiter STX, a slave station number which can be 0 through 9, data 0000 through 9999, and end delimiter CR.



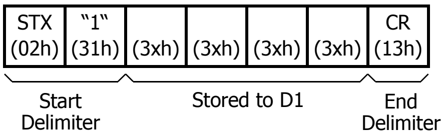
- **Single-byte start delimiter**

Only the first byte can be the start delimiter. The second byte of the incoming data, which is the slave station number, has to be stored to data register D0, and extra ladder programming is needed to see whether the slave station number of the incoming communication is 1 or not. Only when the slave station number is 1, received data stored in D1 is valid for the local PLC.



- **Multi-byte start delimiter**

First two bytes can be configured as a multi-byte start delimiter. The incoming data is processed according to the receive format only when the first two bytes of the incoming data match the start delimiter. Therefore, only the incoming data sent to slave station 1 is processed. No extra ladder programming is needed to check the slave station number.



Designating Constant as End Delimiter

An end delimiter can be programmed at the end of the receive format of a RXD instruction; the FC6A Series MICROSmart will recognize the end of valid communication, although RXD instructions without an end delimiter can also be executed.

When a constant value is designated at the end of source device S1, the one-byte data serves as an end delimiter to end the processing of the received data. If data bits are set to 7 bits, the end delimiters will be between 00h and 7Fh. If data bits are set to 8 bits, the end delimiters will be between 00h and FFh. Constant values are entered in character or hexadecimal notation into the source data. When using the same RXD instruction repeatedly in a user program, assign different end delimiters for each RXD instruction.

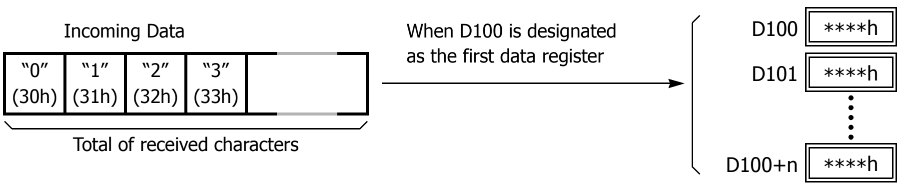
If a character in incoming data matches the end delimiter, the RXD instruction ends receiving data at this point and starts subsequent receive processing as specified. Even if a character matches the end delimiter at a position earlier than expected, the RXD instruction ends receiving data there.

If a BCC code is included in the receive format of a RXD instruction, an end delimiter can be positioned immediately before or after the BCC code. If a data register or skip is designated between the BCC and end delimiter, correct receiving is not ensured.

When a RXD instruction without an end delimiter is executed, data receiving ends when the specified bytes of data in the receive format, such as data registers and skips, have been received. In addition, data receiving also ends when the interval between incoming data characters exceeds the receive timeout value specified in the Communication Parameters dialog box whether the RXD has an end delimiter or not. The character interval timer is started when the first character of incoming communication is received and restarted each time the next character is received. When a character is not received within a predetermined period of time, timeout occurs and the RXD ends data receive operation.

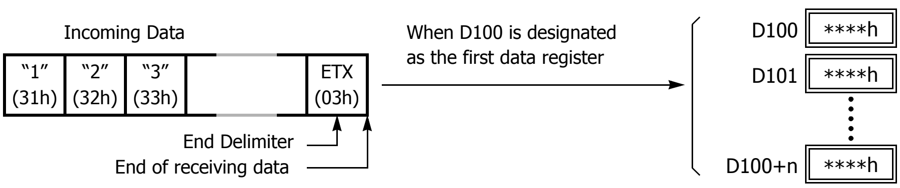
Example:

(1) When a RXD instruction without an end delimiter is executed



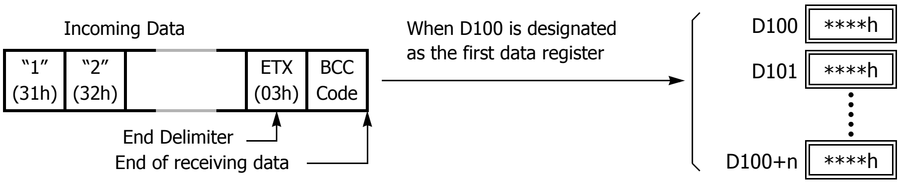
The incoming data is divided, converted, and stored to data registers according to the receive format.
Receive operation is completed when the total characters programmed in RXD are received.

(2) When a RXD instruction with end delimiter ETX (03h) and without BCC is executed



The incoming data is divided, converted, and stored to data registers according to the receive format.
The end delimiter is not stored to a data register.
Any data arriving after the end delimiter is discarded.

(3) When a RXD instruction with end delimiter ETX (03h) and one-byte BCC is executed



The incoming data is divided, converted, and stored to data registers according to the receive format.
The end delimiter and BCC code are not stored to data registers.
After receiving the end delimiter, the FC6A Series MICROSmart receives only the one-byte BCC code.

Constant for Verification

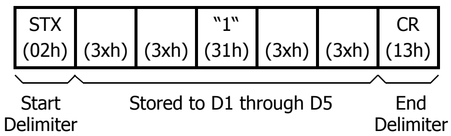
Constants excluding start and end delimiters can be configured in the receive format to verify the incoming data with the constants, which are either characters or hexadecimal values. As many constants can be configured for the verification as required. The verification result is stored in the receive status of the RXD instruction.

Example: Programming Constant for Verification

The following example shows the advantage of using constant for verification. The incoming data contains a constant value "1" in the middle, and that constant value needs to be verified to see whether the incoming data is valid.

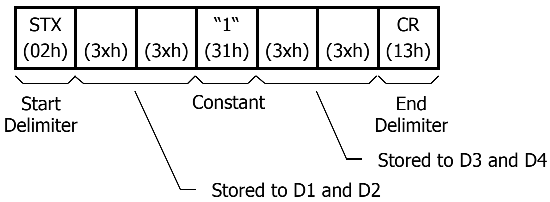
- **Using Data Register**

The incoming data including the constant value needs to be stored in data registers. When the RXD instruction completes receiving the incoming data, the receive status contains 64, meaning the RXD instruction has completed without errors, even if the constant value is not an expected value. Extra ladder programming is needed to see whether the constant value in the incoming data is correct or not.



- **Using Constant for Verification**

A constant to verify the constant value in the incoming data is designated in the receive format. If the constant value is not an expected value when the RXD instruction completes receiving the incoming data, the receive status contains 74, meaning the RXD instruction has completed but user communication error code 5 occurred. No extra ladder programming is needed to see whether the constant value in the received data is correct or not.



Note: When configuring constants, which are either characters or hexadecimal values, in the receive format, and the incoming data does not match the constants in the receive format, then a user communication error code is stored in the receive status. The error code contained in the receive status depends on whether the constants are used as a start delimiter or as constants for verification. If used as a start delimiter, user communication error code 7 is stored in the receive status, and the RXD instruction keeps waiting for valid incoming data. On the other hand, if used as constants for verification, the receive status contains 74, and the RXD instruction finishes the execution. To repeat receiving incoming communication, turn on the input for the RXD instruction.

- Start delimiter of incoming data does not match the receive format

Incoming Data	Const (02h)	(xxh)	(FFh)	(xxh)	(0Dh)	
Receive Format	Const (05h)	DR	Const (FFh)	DR	Const (0Dh)	Receive status: 7 Receive completion output: 0
	Start Delimiter					

RXD instruction keeps waiting for valid incoming data, and completes data receiving after receiving valid data including a correct start delimiter.

- Constant for verification of incoming data does not match the receive format

Incoming Data	(05h)	(xxh)	(0Fh)	(xxh)	(0Dh)	
Receive Format	Const (05h)	DR	Const (FFh)	DR	Const (0Dh)	Receive status: 74 Receive completion output: 1
			Constant for Verification			

Receive status stores 74 (= 64 + 10)

RXD instruction completes data receiving. To repeat receiving incoming communication, turn on the input for the RXD instruction.

Note: Constants configured in the beginning of receive formats have different functions as shown below:

- More than five constants are configured in the beginning of the receive format

Byte	1st	2nd	3rd	4th	5th	6th		200th
Receive Format	Const	Const	Const	Const	Const	Const		Const

Start Delimiter Up to 5th byte

Constant for Verification 6th byte and after

- Other than constants (data register, skip, or BCC) are included in the first five bytes of the receive format

Byte	1st	2nd	3rd	4th	5th	6th		200th
Receive Format	Const	DR etc.	Const	Const	Const			

Start Delimiter Consecutive first 5 bytes only

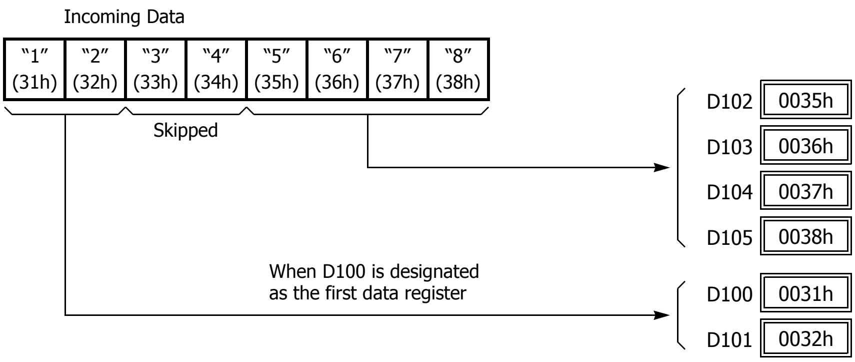
Data register, skip, or BCC

Constant for Verification Not consecutive in the first 5 bytes

Skip

When “skip” is designated in the receive format, a specified quantity of digits in the incoming data are skipped and not stored to data registers. A maximum of 99 digits (bytes) of characters can be skipped continuously.

Example: When a RXD instruction with skip for 2 digits starting at the third byte is executed



BCC (Block Check Character)

The FC6A Series MICROSmart has an automatic BCC calculation function to detect a communication error in incoming data. If a BCC code is designated in the receive format of a RXD instruction, the FC6A Series MICROSmart calculates a BCC value for a specified starting position through the position immediately preceding the BBC. It then compares the calculation result with the BCC code in the received incoming data. The start position for the BCC calculation can be specified from the first byte through the 15th byte. The BCC can be 1 or 2 digits.

When an end delimiter is not used in the RXD instruction, the BCC code must be positioned at the end of the receive format designated in Source 1 device. When an end delimiter is used, the BCC code must be immediately before or after the end delimiter. The FC6A Series MICROSmart reads a specified number of BCC digits in the incoming data according to the receive format to calculate and compare the received BCC code with the BCC calculation results.

BCC Calculation Start Position

The start position for the BCC calculation can be specified from the first byte through the 15th byte. The BCC is calculated for the range starting at the designated position up to the byte immediately before the BCC of the receive data.

Example: Received data consists of 17 bytes plus 2 BCC digits.

(1) Calculation start position = 1

1st	2nd	3rd	4th	5th	6th		15th	16th	17th	18th	19th
STX	"A"	"B"	"C"	"D"	"E"		"0"	CR	LF	BCC	BCC

BCC calculation range (1st–17th)

BCC (2 digits) (18th–19th)

(2) Calculation start position = 2

1st	2nd	3rd	4th	5th	6th		15th	16th	17th	18th	19th
STX	"A"	"B"	"C"	"D"	"E"		"0"	CR	LF	BCC	BCC

BCC calculation range (2nd–17th)

BCC (2 digits) (18th–19th)

BCC Calculation Formula

BCC calculation formula can be selected from XOR (exclusive OR), ADD (addition), ADD-2comp, Modbus ASCII, or Modbus RTU.

Example: Incoming data consists of 41h, 42h, 43h, and 44h.

(1) BCC calculation formula = XOR

Calculation result = 41h ⊕ 42h ⊕ 43h ⊕ 44h = 04h

(2) BCC calculation formula = ADD

Calculation result = 41h + 42h + 43h + 44h = 10Ah → 0Ah (Only the last 1 or 2 digits are used as BCC.)

(3) BCC calculation formula = ADD-2comp

Calculation result = FEh, F6h (2 digits without conversion)

(4) BCC calculation formula = Modbus ASCII

Calculation result = 88 (ASCII)

(5) BCC calculation formula = Modbus RTU

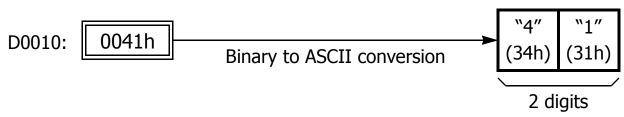
Calculation result = 85h 0Fh (binary)

Conversion Type

The BCC calculation result can be converted or not according to the assigned conversion type as described below:

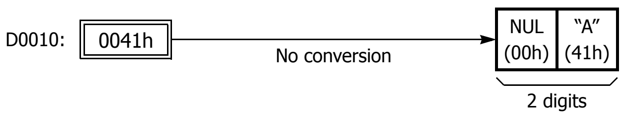
Example: BCC calculation result is 0041h.

(1) Binary to ASCII conversion



Note: On WindLDR, Modbus ASCII defaults to binary to ASCII conversion.

(2) No conversion

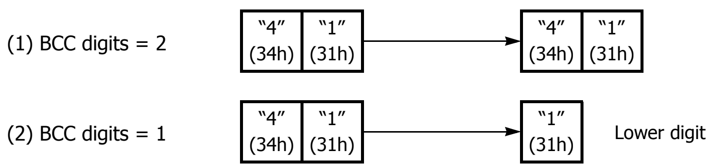


Note: On WindLDR, Modbus RTU defaults to no conversion.

BCC Digits (Bytes)

The quantity of digits (bytes) of the BCC code can be selected from 1 or 2.

Example:

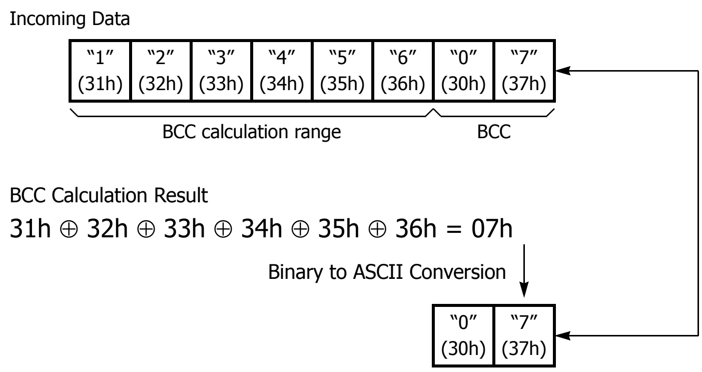


Note: On WindLDR, Modbus ASCII and Modbus RTU default is set to 2 digits.

Comparing BCC Codes

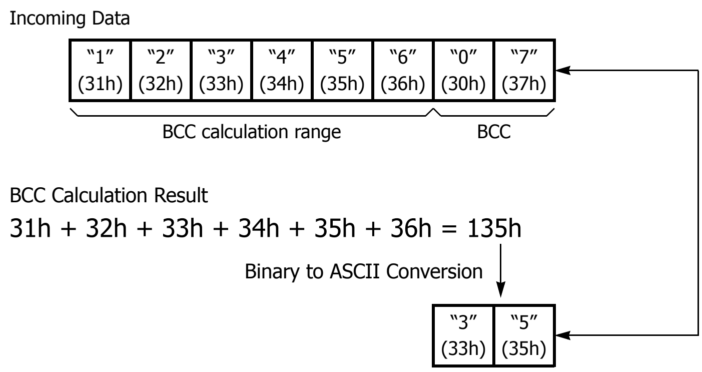
The FC6A Series MICROSmart compares the BCC calculation result with the BCC code in the received incoming data to check for any error in the incoming communication due to external noises or other causes. If a disparity is found in the comparison, an error code is stored in the data register designated as receive status in the RXD instruction. For user communication error code, see "User Communication Error" on page 5-52.

Example 1: BCC is calculated for the first byte through the sixth byte using the XOR format, converted in binary to ASCII, and compared with the BCC code appended to the seventh and eighth bytes of the incoming data.



Comparison result is true to indicate that data is received correctly.

Example 2: BCC is calculated for the first byte through the sixth byte using the ADD format, converted in binary to ASCII, and compared with the BCC code appended to the seventh and eighth bytes of the incoming data.



Comparison result is false.

Error code 9 is stored in the receive status data register.

D1 (Destination 1)

Set an internal relay or an output for the receive completion output.
When the start input for a RXD instruction is turned on, preparation for receiving data is initiated, followed by data conversion and storage. When the data receive operation sequence is complete, the designated output or internal relay is turned on.

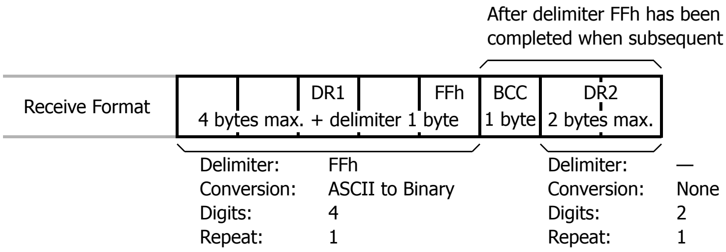
Conditions for Completion of Receiving Data

After starting to receive data, the RXD instruction can be completed in three ways depending on the designation of end delimiter and delimiter in the receive format.

End Delimiter	Delimiter	Conditions for Completion of Receiving Data
With	With or Without	When a specified byte count of data (digits × repeat) has been received or when an end delimiter is received. When a BCC exists immediately after the end delimiter, the BCC is received before ending data receiving.
Without	With	After the last constant (including delimiter) designated in the RXD instruction has been received, data receiving is completed when the subsequent byte count of data has been received.
Without	Without	When a specified byte count of data (digits × repeat) has been received.

Note: Whenever a receive timeout has occurred, data receiving stops arbitrarily.
Data receiving is complete when one of the above three conditions is met. To abort a RXD instruction, use the special internal relay for user communication receive instruction cancel flag. See "User Communication Receive Instruction Cancel Flag" on page 5-22.

Example: A RXD instruction does not have an end delimiter and has a delimiter programmed in the receive format for data registers.



D2 (Destination 2)

Set the data register in which to store the receive status and the receive data byte count.
The receive status is stored in D2+0 and the receive data byte count is stored in D2+1.

D2+0 (Receive Status)

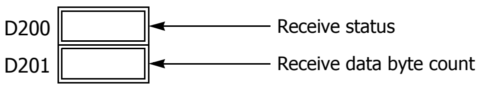
The receive status is stored in the data register set with D2. The receive status includes the reception operation status and the error information.

Receive Status Code	Status	Description
16	Preparing data receive	From turning on the start input for a RXD instruction to read the receive format, until the RXD instruction is enabled by an END processing
32	Receiving data	From enabling the RXD instruction by an END processing, until incoming data is received
48	Data receive complete	From receiving incoming data, until the received data is converted and stored in data registers according to the receive format
64	Receive instruction complete	All data receive operation is completed and the next data receive is made possible
128	User communication receive instruction cancel flag active	RXD instructions are cancelled by special internal relay for user communication receive instruction cancel flag, such as M8022, M8023, M8026, M8033, M8145 to M8147, M8170, M8176, or M8365 to M8394

If the receive status code is other than shown above, a receive instruction error is suspected. See "User Communication Error" on page 5-52.

D2+1 (Receive Digits (Bytes))

The data register next to the device designated for receive status stores the byte count of data received by the RXD instruction. When a start delimiter, end delimiter, and BCC are included in the received data, the byte counts for these codes are also included in the receive data byte count.
Example: Data register D200 is designated as a device for receive status.



User Communication Receive Instruction Cancel Flag

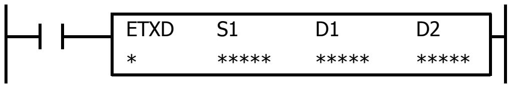
If user communication receive instruction cancel is turned on when the receive pre-processing for the user communication receive instruction has already been completed and data is being received (status code 32), all receive instructions for the corresponding port will be canceled. This is effective for canceling receive instruction execution when waiting a long time to receive data.
To activate a receive instruction that was canceled, turn off the user communication receive instruction cancel flag, and then turn on the receive instruction input conditions again.
User communication receive instruction cancel flags are allocated as follows to each communication port as a special internal relay.

Device Address	Description	CPU Stopped	Power OFF	R/W
M8022	User Communication Receive Instruction Cancel Flag (Port 1)	Cleared	Cleared	R/W
M8023	User Communication Receive Instruction Cancel Flag (Port 2)	Cleared	Cleared	R/W
M8026	User Communication Receive Instruction Cancel Flag (Port 3)	Cleared	Cleared	R/W
M8033	User Communication Receive Instruction Cancel Flag (Port 4)[*1]	Cleared	Cleared	R/W
M8145	User Communication Receive Instruction Cancel Flag (Port 5)[*1]	Cleared	Cleared	R/W
M8146	User Communication Receive Instruction Cancel Flag (Port 6)[*1]	Cleared	Cleared	R/W
M8147	User Communication Receive Instruction Cancel Flag (Port 7)[*1]	Cleared	Cleared	R/W
M8170	User Communication Receive Instruction Cancel Flag (Port 8)[*1]	Cleared	Cleared	R/W
M8176	User Communication Receive Instruction Cancel Flag (Port 9)[*1]	Cleared	Cleared	R/W
M8365	User Communication Receive Instruction Cancel Flag (Port 10)[*1*2]	Cleared	Cleared	R/W
M8366	User Communication Receive Instruction Cancel Flag (Port 11)[*1*2]	Cleared	Cleared	R/W
M8367	User Communication Receive Instruction Cancel Flag (Port 12)[*1*2]	Cleared	Cleared	R/W
M8370	User Communication Receive Instruction Cancel Flag (Port 13)[*1*2]	Cleared	Cleared	R/W
M8371	User Communication Receive Instruction Cancel Flag (Port 14)[*1*2]	Cleared	Cleared	R/W
M8372	User Communication Receive Instruction Cancel Flag (Port 15)[*1*2]	Cleared	Cleared	R/W
M8373	User Communication Receive Instruction Cancel Flag (Port 16)[*1*2]	Cleared	Cleared	R/W
M8374	User Communication Receive Instruction Cancel Flag (Port 17)[*1*2]	Cleared	Cleared	R/W
M8375	User Communication Receive Instruction Cancel Flag (Port 18)[*1*2]	Cleared	Cleared	R/W
M8376	User Communication Receive Instruction Cancel Flag (Port 19)[*1*2]	Cleared	Cleared	R/W
M8377	User Communication Receive Instruction Cancel Flag (Port 20)[*1*2]	Cleared	Cleared	R/W
M8380	User Communication Receive Instruction Cancel Flag (Port 21)[*1*2]	Cleared	Cleared	R/W
M8381	User Communication Receive Instruction Cancel Flag (Port 22)[*1*2]	Cleared	Cleared	R/W
M8382	User Communication Receive Instruction Cancel Flag (Port 23)[*1*2]	Cleared	Cleared	R/W
M8383	User Communication Receive Instruction Cancel Flag (Port 24)[*1*2]	Cleared	Cleared	R/W
M8384	User Communication Receive Instruction Cancel Flag (Port 25)[*1*2]	Cleared	Cleared	R/W
M8385	User Communication Receive Instruction Cancel Flag (Port 26)[*1*2]	Cleared	Cleared	R/W
M8386	User Communication Receive Instruction Cancel Flag (Port 27)[*1*2]	Cleared	Cleared	R/W
M8387	User Communication Receive Instruction Cancel Flag (Port 28)[*1*2]	Cleared	Cleared	R/W
M8390	User Communication Receive Instruction Cancel Flag (Port 29)[*1*2]	Cleared	Cleared	R/W
M8391	User Communication Receive Instruction Cancel Flag (Port 30)[*1*2]	Cleared	Cleared	R/W
M8392	User Communication Receive Instruction Cancel Flag (Port 31)[*1*2]	Cleared	Cleared	R/W
M8393	User Communication Receive Instruction Cancel Flag (Port 32)[*1*2]	Cleared	Cleared	R/W
M8394	User Communication Receive Instruction Cancel Flag (Port 33)[*1*2]	Cleared	Cleared	R/W

*1 When using a communication module.
*2 Plus CPU module only.

Note: "R/W" is the abbreviation for read/write. When R/W, it can be read and written. When R, it can only be read. When W, it can only be written.

ETXD (User Communication Transmit over Ethernet)



The ETXD instruction converts transmission data to the specified data type and sends it to the external device connected over Ethernet port 1 and 2.

When the input is on, the transmission data designated by S1 is transmitted to the device connected with the specified connection.

When transmission is complete, the device designated by D1 is turned on. Transmit status (the transmission status and error code) is stored to the device designated by D2.

The byte count of transmitted data is stored to D2+1.

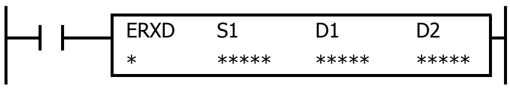
Apart from the connection settings, the settings of ETXD and TXD instructions are the same. For details on TXD instruction, see "TXD (Transmit)" on page 5-2.

The ETXD instruction cannot be used in an interrupt program.

If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the FC6A Series MICROSmart. For details about the user program execution errors, see Chapter 3 "User Program Execution Errors" in the "FC6A Series MICROSmart Ladder Programming Manual".

Note: For details on the user communication client and user communication server specifications and for details on user communication over Ethernet, see "User Communication via Ethernet Communication" on page 5-40.

ERXD (User Communication Receive over Ethernet)



The ERXD instruction receives data from an external device connected over Ethernet port 1 and 2, converts the received data in the specified format, and stores the converted data to data registers.

When the input is on, the received data designated by S1 is received from the device connected with the specified connection.

When all data has been received, the device designated by D1 is turned on. Receive status (the receive status and error code) is stored to the device designated by D2.

The byte count of received data is stored to D2+1.

When user communication receive instruction cancel flag (M8200 - M8207 and M8334 - M8343) is turned on while receiving incoming data, the execution of all active receive instructions for the corresponding connection is canceled.

Apart from the connection settings and the allocation of the user communication receive instruction cancel flags, the settings of ERXD and RXD instructions are the same. For details on RXD instruction, see "RXD (Receive)" on page 5-10.

The ERXD instruction cannot be used in an interrupt program.

If used, a user program execution error will result, turning on special internal relay M8004 and the ERR LED on the FC6A Series MICROSmart. For details about the user program execution errors, see Chapter 3 "User Program Execution Errors" in the "FC6A Series MICROSmart Ladder Programming Manual".

Note: For details on the user communication client and user communication server specifications and for details on user communication (UDP), see "User Communication via Ethernet Communication" on page 5-40.

User Communication via Serial Communication

With user communication via serial communication, the FC6A Series MICROSmart can send and receive data between external devices connected to port 1 to port 33, such as a printer or barcode reader, by using the TXD (user communication transmit) and RXD (user communication receive) instructions.

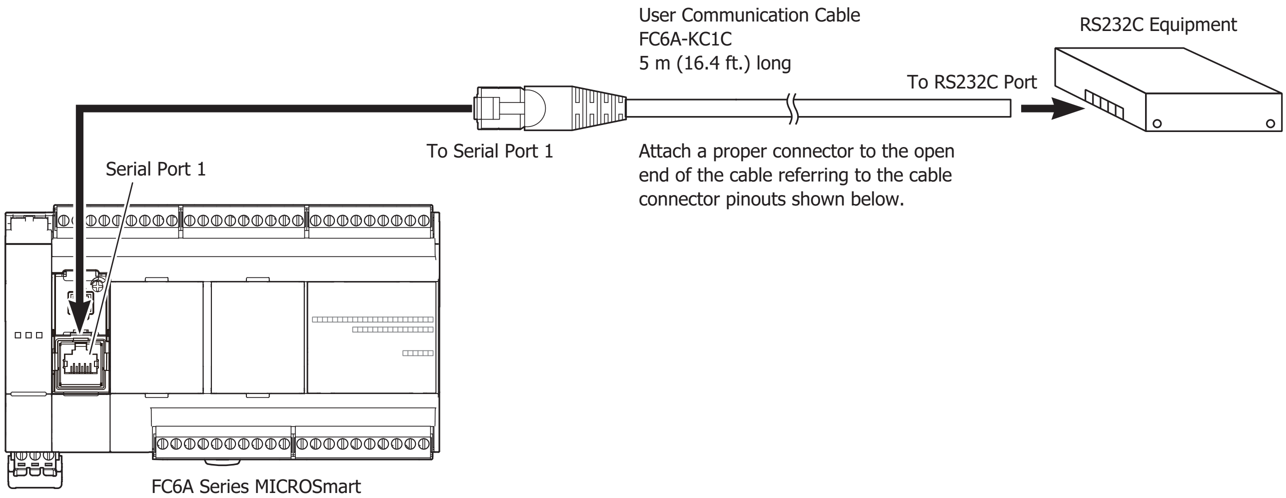
User Communication Overview

By installing a communication cartridge on the FC6A Series MICROSmart expansion communication port, the FC6A Series MICROSmart can communicate with two external devices simultaneously.

When using an RS485 communication cartridge, FC6A Series MICROSmart modules can communicate with a maximum of 31 RS485 devices using the user communication.

User communication transmit and receive instructions can be programmed to match the communication protocol of the equipment to communicate with. Possibility of communication using the user communication mode can be determined referring to the user communication mode specifications described below.

An RS232C equipment is connected to port 1 of the FC6A Series MICROSmart.



User Communication Mode Specifications

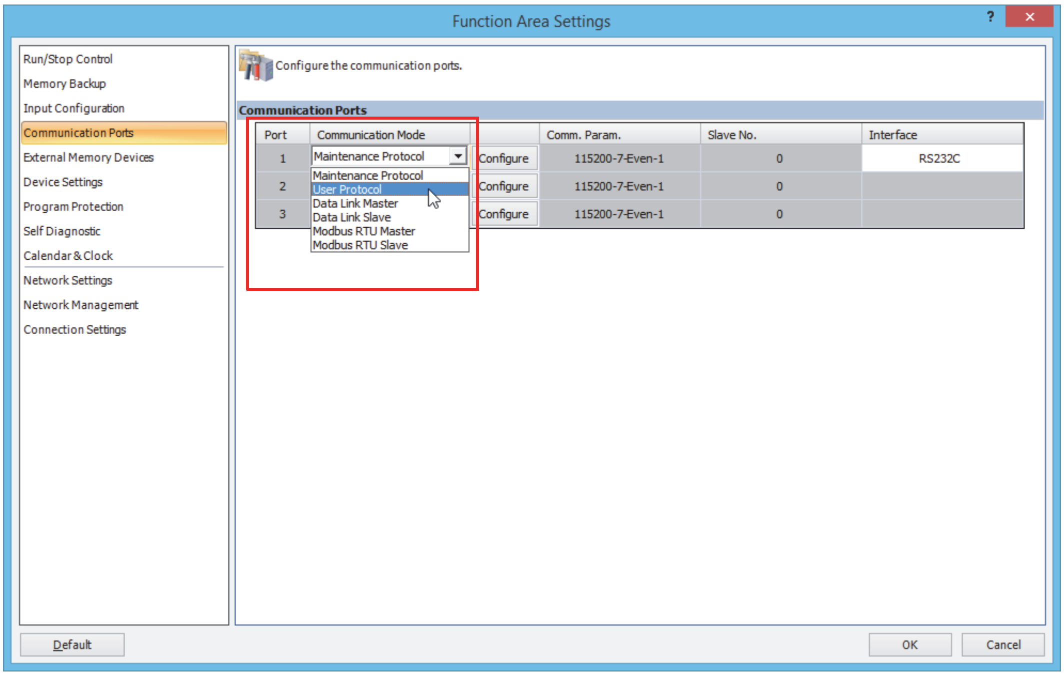
Type	RS232C User Communication	RS485 User Communication
Communication Port	Port 1 to port 33	Port 1 to port 33
Maximum Nodes	1 per port	31 maximum
Standards	EIA RS232C	EIA RS485
Baud Rate	1,200, 2,400, 4,800, 9,600, 19,200, 38,400, 57,600, 115,200 bps (Default: 115,200)	
Data Bits	7 or 8 bits (Default: 7)	
Parity	Odd, Even, None (Default: Even)	
Stop Bits	1 or 2 bits (Default: 1)	
Receive Timeout	10 to 2,540 ms (10 ms increments) or none (Receive timeout is disabled when 2,550 ms is selected.) The receive timeout has an effect when using RXD instructions.	
Communication Method	Start-stop synchronization system	
Maximum Cable Length	FC6A-PC1: 5 m FC6A-SIF52: 15 m	FC6A-PC3: 200 m FC6A-SIF52: 1,200 m
Maximum Transmit Data	1,500 bytes	
Maximum Receive Data	1,500 bytes	
BCC Calculation	XOR, ADD, ADD-2comp *, Modbus ASCII *, Modbus RTU * (* For calculation examples, see "BCC Calculation Examples" on page 5-58.)	

Programming WindLDR

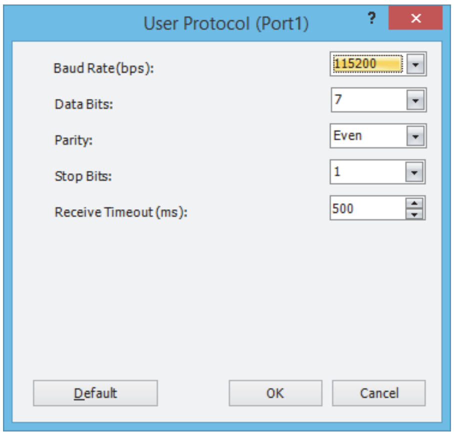
When using the user communication function to communicate with an external RS232C or RS485 device, set the communication parameters for the FC6A Series MICROSmart to match those of the external device.

Note: Since communication parameters in the Function Area Settings relate to the user program, the user program must be downloaded to the FC6A Series MICROSmart after changing any of these settings.

1. From the WindLDR menu bar, select **Configuration** > **Communication Ports**.
   The **Function Area Settings** dialog box for Communication Ports appears.



2. In the Communication Mode pull-down list for **Port 1** to **Port 3**, select **User Protocol**. (Click the **Configure** button when changing previous settings.)
   The **User Protocol** dialog box appears.



When **2550 ms** is selected in the **Receive Timeout** box, the receive timeout function is disabled.

3. Select communication parameters to the same values for the device to communicate with.
4. Click the **OK** button.

Programming TXD Instruction Using WindLDR

The following example demonstrates how to program a TXD instruction including a start delimiter, BCC, and end delimiter using WindLDR.

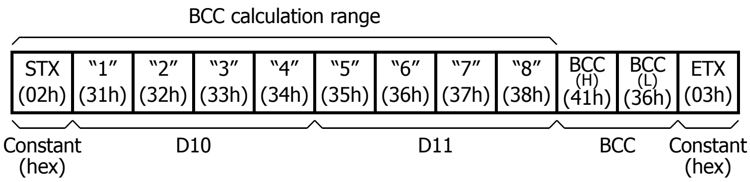
TXD sample program:

Communication port:	Port 1
Transmit completion output:	M10
Transmit status register:	D100
Transmit data byte count:	D101

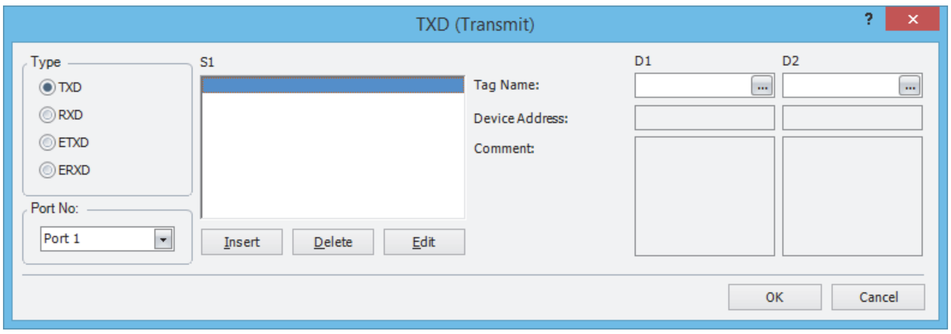
Data register contents:

D10	04D2h	= 1234
D11	162Eh	= 5678

Transmit data example:

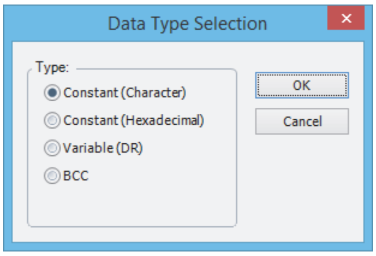


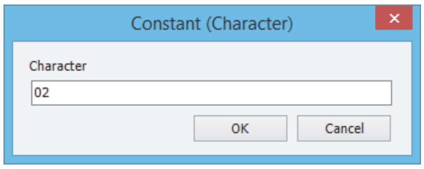
1. Start to program a TXD instruction. Move the cursor where you want to insert the TXD instruction, and type **TXD**. You can also insert the TXD instruction by clicking the User Communication icon in the menu bar and clicking where you want to insert the TXD instruction in the program edit area.
   The **TXD (Transmit)** dialog box appears.



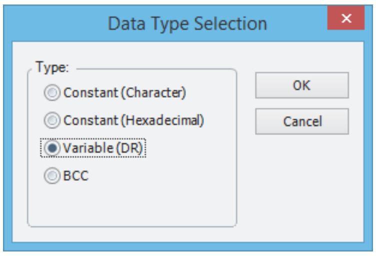
2. Check that **TXD** is selected in the **Type** box and select **Port 1** in the **Port No.** box. Then, click **Insert**.
   The **Data Type Selection** dialog box appears. You will program source device S1 using this dialog box.

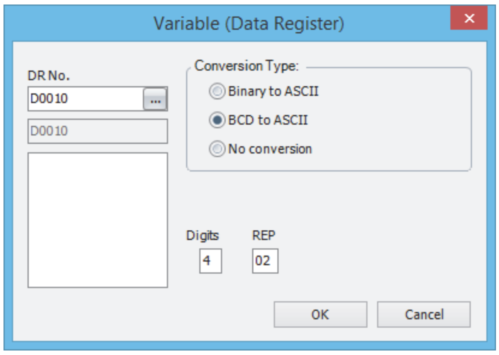
3. Click **Constant (Hexadecimal)** in the **Type** box and click **OK**. Next, in the **Constant (Hexadecimal)** dialog box, type **02** to program the start delimiter STX (02h). When finished, click **OK**.



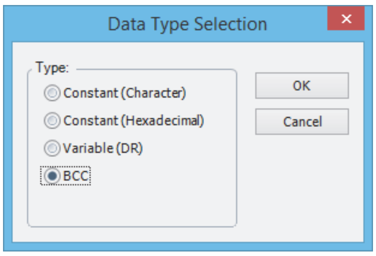


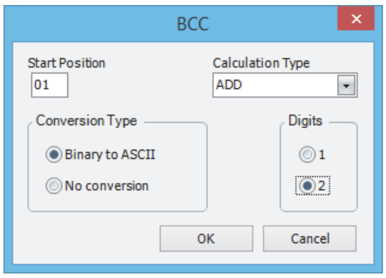
4. Since the **TXD (Transmit)** dialog box reappears, repeat the above procedure. In the **Data Type Selection** dialog box, click **Variable (DR)** and click **OK**. Next, in the **Variable (Data Register)** dialog box, type **D0010** in the **DR No.** box and click **BCD to ASCII** to select the BCD to ASCII conversion. Enter **4** in the **Digits** box (4 digits) and **2** in the **REP** box (2 repeat cycles). When finished, click **OK**.



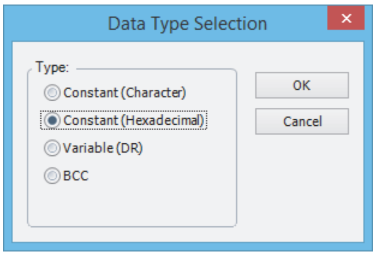


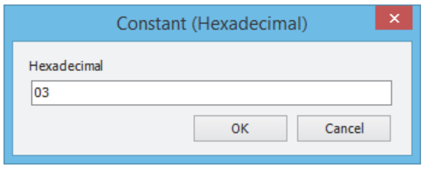
5. Again in the **Data Type Selection** dialog box, click **BCC** and click **OK**. Next, in the **BCC** dialog box, enter **1** in the **Calculation Start Position** box, select **ADD** for the **Calculate Type**, click **Binary to ASCII** for the **Conversion Type**, and click **2** for the **Digits**. When finished, click **OK**.



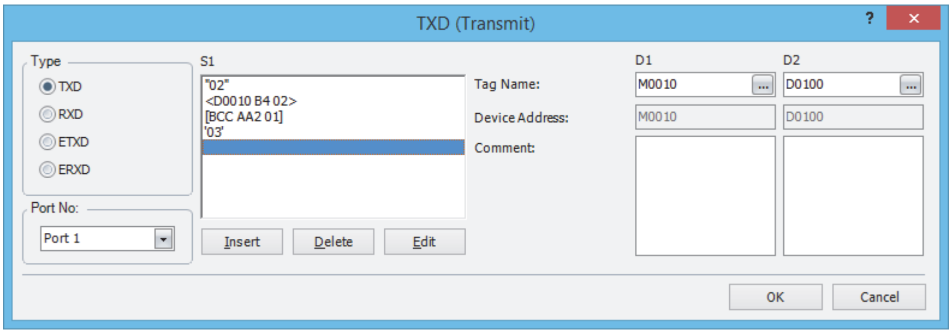


6. Once again in the **Data Type Selection** dialog box, click **Constant (Hexadecimal)** and click **OK**. Next, in the Constant (Hexadecimal) dialog box, type **03** to program the end delimiter ETX (03h). When finished, click **OK**.

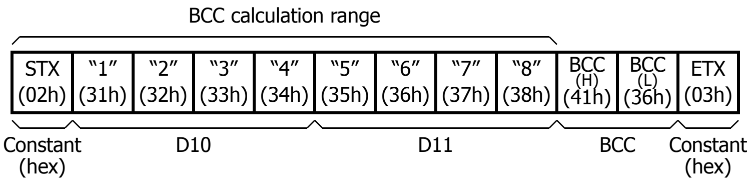




7. In the **TXD (Transmit)** dialog box, type **M0010** in the destination D1 box and type **D0100** in the destination D2 box. When finished, click **OK**.



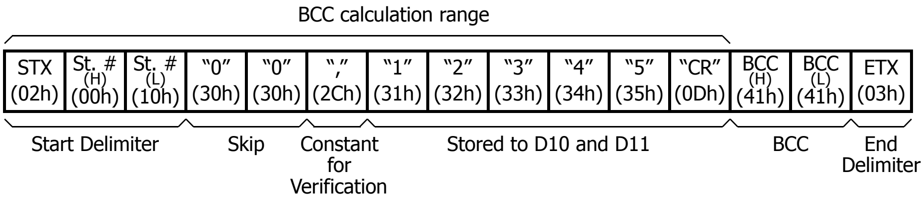
Programming of the TXD1 instruction is complete and the transmit data is specified as follows:



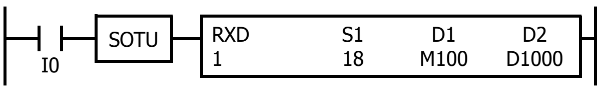
Programming RXD Instruction Using WindLDR

The following example demonstrates how to program an RXD instruction including a start delimiter, skip, constant for verification, BCC, and end delimiter using WindLDR. Converted data is stored to data registers D10 and D11. Internal relay M100 is used as destination D1 for the receive completion output. Data register D1000 is used as destination D2 for the receive status, and data register D1001 is used to store the receive data byte count.

Receive data example:

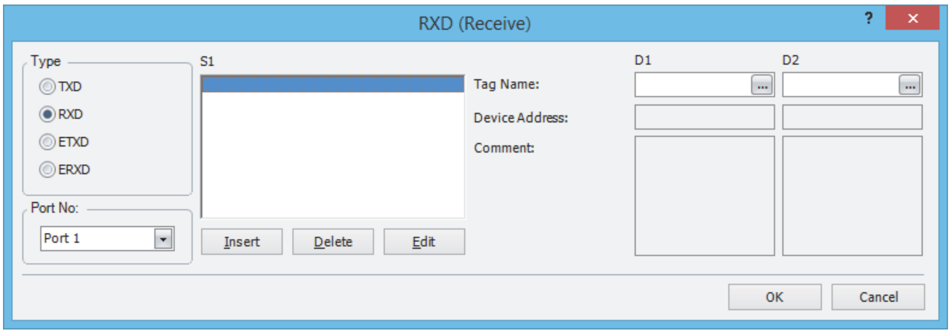


RXD sample program:



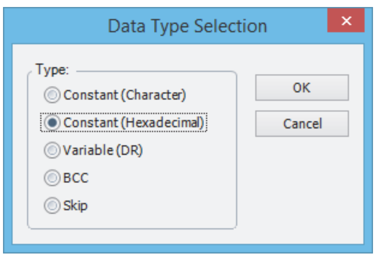
Communication port:	Port 1
Receive completion output:	M100
Receive status register:	D1000
Receive data byte count:	D1001

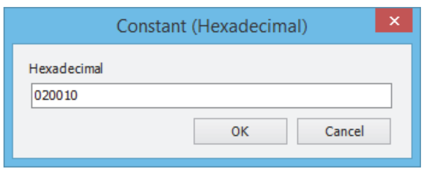
1. Start to program an RXD instruction. Move the cursor where you want to insert the RXD instruction, and type **RXD**. You can also insert the RXD instruction by clicking the User Communication icon in the menu bar and clicking where you want to insert the RXD instruction in the program edit area, then the Transmit dialog box appears. Click **RXD** to change the dialog box to the Receive dialog box.
   The **RXD (Receive)** dialog box appears.



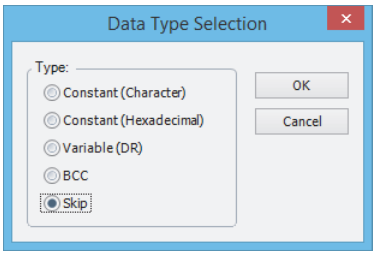
2. Check that **RXD** is selected in the Type box and select **Port 1** in the Port box. Then, click **Insert**.
   The **Data Type Selection** dialog box appears. You will program source device S1 using this dialog box.

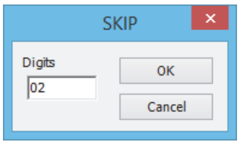
3. Click **Constant (Hexadecimal)** in the **Type** box and click **OK**. Next, in the **Constant (Hexadecimal)** dialog box, type **020010** to program the start delimiter STX (02h), Station No. H (00h), and Station No. L (10h). When finished, click **OK**.



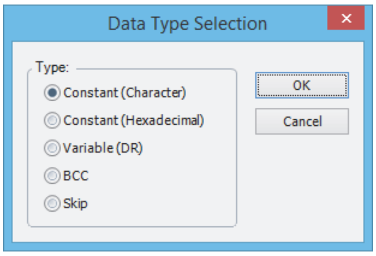


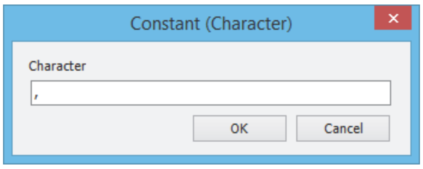
4. Since the **RXD (Receive)** dialog box reappears, repeat the above procedure. In the **Data Type Selection** dialog box, click **Skip** and click **OK**. Next, in the **Skip** dialog box, type **02** in the **Digits** box and click **OK**.



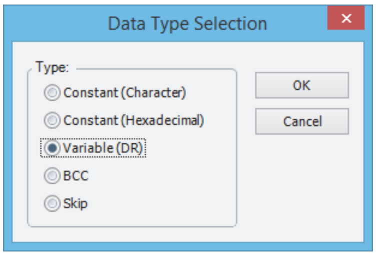


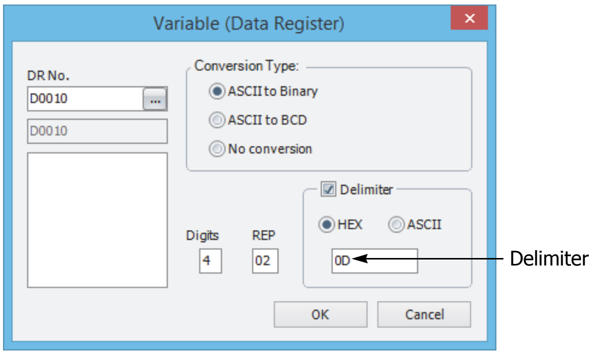
5. Again in the **Data Type Selection** dialog box, click **Constant (Character)** and click **OK**. Next, in the **Constant (Character)** dialog box, type **, (2Ch)** in the **Character** box to program a comma as a constant to verify. When finished, click **OK**.



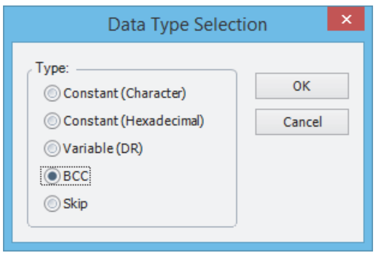


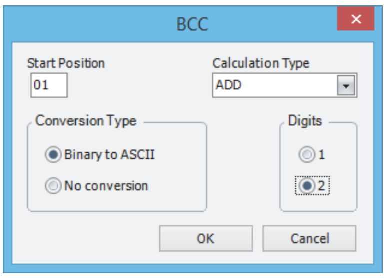
6. Again in the **Data Type Selection** dialog box, click **Variable (DR)** and click **OK**. Next, in the **Variable (Data Register)** dialog box, type **D10** in the **DR No.** box and click **ASCII to Binary** to select ASCII to binary conversion. Enter **4** in the **Digits** box (4 digits) and **2** in the **REP** box (2 repeat cycles). Click **Variable**, select **HEX**, and type **0D** to designate a delimiter. When finished, click **OK**.



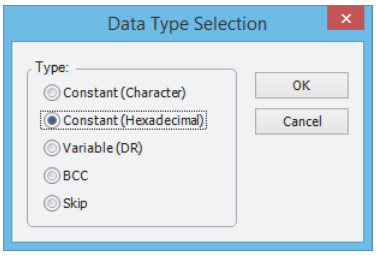


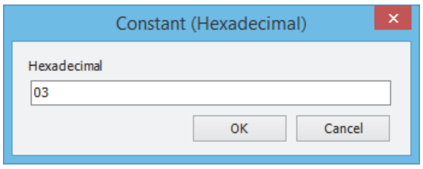
7. Again in the **Data Type Selection** dialog box, click **BCC** and click **OK**. Next, in the **BCC** dialog box, enter **01** in the **Calculation Start Position** box, select **ADD** for the **Calculation Type**, click **Binary to ASCII** for the **Conversion Type**, and click **2** for the **Digits**. When finished, click **OK**.



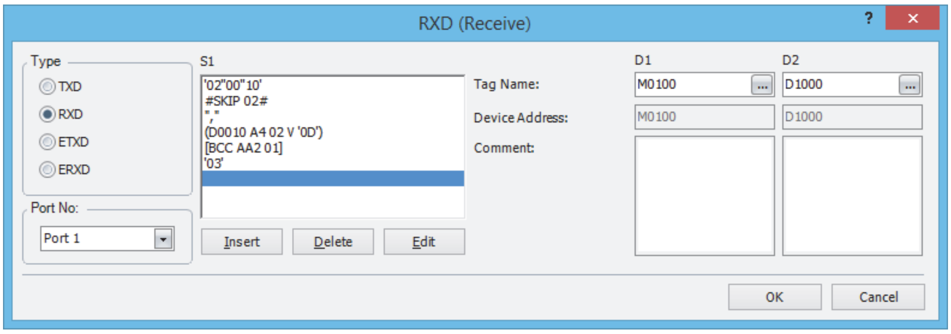


8. Once again in the **Data Type Selection** dialog box, click **Constant (Hexadecimal)** and click **OK**. Next, in the **Constant (Hexadecimal)** dialog box, type **03** to program the end delimiter ETX (03h). When finished, click **OK**.





9. In the **RXD (Receive)** dialog box, type **M0100** in the destination **D1** box and type **D1000** in the destination **D2** box. When finished, click **OK**.



Programming of the RXD instruction is complete and the receive data will be stored as follows:

D10	1234h	= 4660
D11	0005h	= 5

RS232C Line Control Signals

While the FC6A Series MICROSmart is in user communication mode, special data registers can be used to enable or disable DSR and DTR control signal options for port 1 through port 33.

Line control signals cannot be used with RS485 communication.

The RS-232C communication cartridge (FC6A-PC1) RS signal is an always on output signal.

In the maintenance communication mode, DSR has no effect and DTR remains on.

Special Data Registers for RS232C Line Control Signals

Special data registers D8104 through D8734 are allocated for RS232C line control signals.

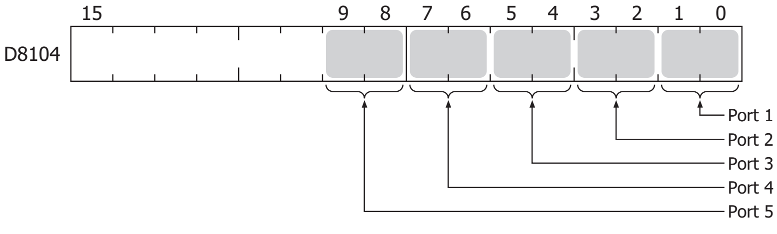
DR No.	Data Register Function	Data Register Value Updated	R/W
D8104	Control signal status (port 1 to port 5)	Every scan	R
D8105	DSR input control signal option (port 1 to port 5)	Every scan	R/W
D8106	DTR output control signal option (port 1 to port 5)	Every scan	R/W
D8204	Control signal status (port 6 to port 9)	Every scan	R
D8205	DSR input control signal option (port 6 to port 9)	Every scan	R/W
D8206	DTR output control signal option (port 6 to port 9)	Every scan	R/W
D8717	Control signal status (port 10 to port 13)	Every scan	R
D8718	DSR input control signal option (port 10 to port 13)	Every scan	R/W
D8719	DTR output control signal option (port 10 to port 13)	Every scan	R/W
D8720	Control signal status (port 14 to port 17)	Every scan	R
D8721	DSR input control signal option (port 14 to port 17)	Every scan	R/W
D8722	DTR output control signal option (port 14 to port 17)	Every scan	R/W
D8723	Control signal status (port 18 to port 21)	Every scan	R
D8724	DSR input control signal option (port 18 to port 21)	Every scan	R/W
D8725	DTR output control signal option (port 18 to port 21)	Every scan	R/W
D8726	Control signal status (port 22 to port 25)	Every scan	R
D8727	DSR input control signal option (port 22 to port 25)	Every scan	R/W
D8728	DTR output control signal option (port 22 to port 25)	Every scan	R/W
D8729	Control signal status (port 26 to port 29)	Every scan	R
D8730	DSR input control signal option (port 26 to port 29)	Every scan	R/W
D8731	DTR output control signal option (port 26 to port 29)	Every scan	R/W
D8732	Control signal status (port 30 to port 33)	Every scan	R
D8733	DSR input control signal option (port 30 to port 33)	Every scan	R/W
D8734	DTR output control signal option (port 30 to port 33)	Every scan	R/W

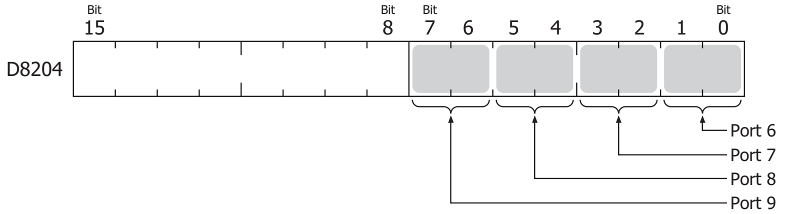
Control Signal Status

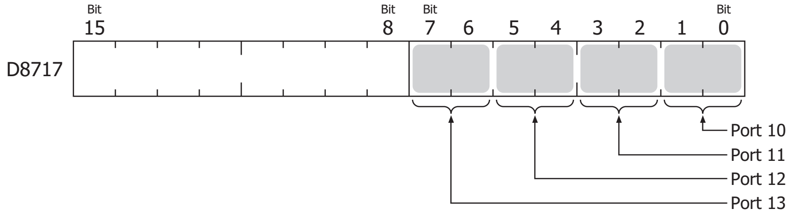
The signal statuses of the control signals (DSR and DTR) are stored in D8104, D8204, D8717, D8720, D8723, D8726, D8729, and D8732.

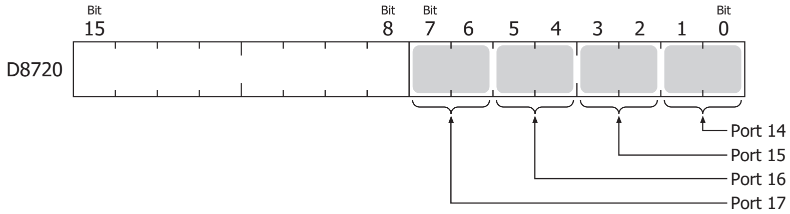
The statuses of the DSR and DTR signals are stored in the above devices that correspond to the communication port. The values in the devices are updated in END processing when stopped and while running.

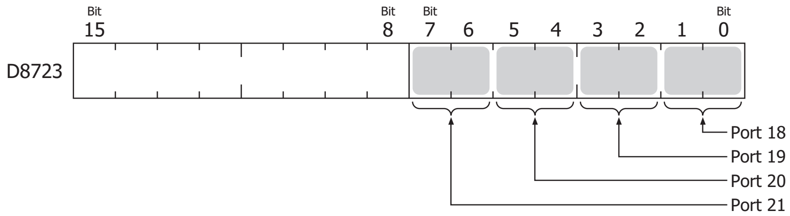
The allocation of communication ports in the device (bit assignment) is as follows.

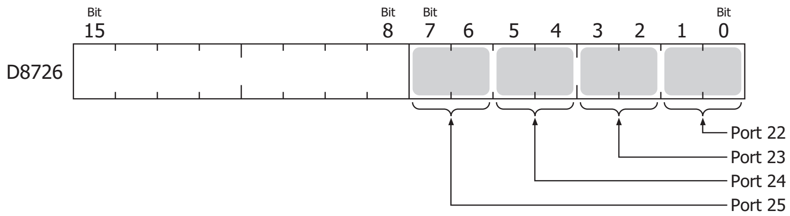


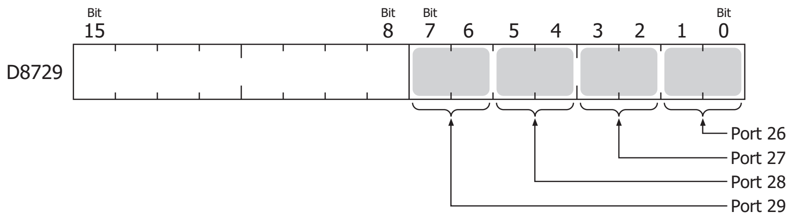


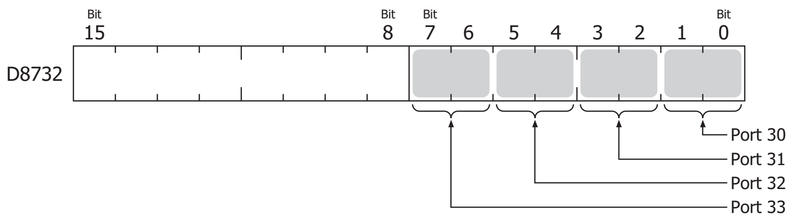












2-bit Binary Value	DTR	DSR	Description
00	OFF	OFF	Both DSR and DTR are off
01	OFF	ON	DSR is on
10	ON	OFF	DTR is on
11	ON	ON	Both DSR and DTR are on

DSR Control Signal Status in RUN and STOP Modes

Communication Mode	3-bit Binary Value	DSR (Input) Status	
		RUN Mode	STOP Mode
User Communication Mode	**000 (default)**	No effect	No effect (TXD/RXD disabled)
	001	ON: Enable TXD/RXD OFF: Disable TXD/RXD	No effect (TXD/RXD disabled)
	010	ON: Disable TXD/RXD OFF: Enable TXD/RXD	No effect (TXD/RXD disabled)
	011	ON: Enable TXD OFF: Disable TXD	No effect (TXD/RXD disabled)
	100	ON: Disable TXD OFF: Enable TXD	No effect (TXD/RXD disabled)
	≥ **101**	No effect	No effect (TXD/RXD disabled)
Maintenance Mode	—	No effect	No effect

DTR Control Signal Status in RUN and STOP Modes

Communication Mode	2-bit Binary Value	DTR (Output) Status	
		RUN Mode	STOP Mode
User Communication Mode	**00 (default)**	ON	OFF
	01	OFF	OFF
	10	RXD enabled: ON RXD disabled: OFF	OFF
	11	ON	OFF
Maintenance Mode	—	ON	ON

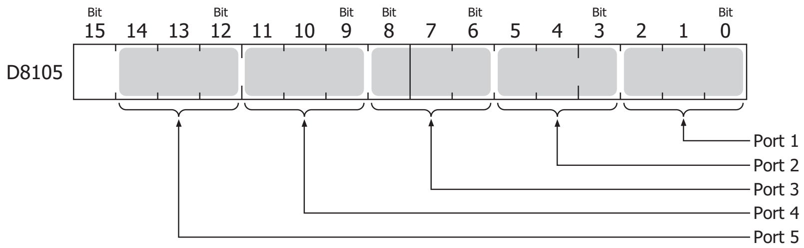
DSR Input Control Signal Option

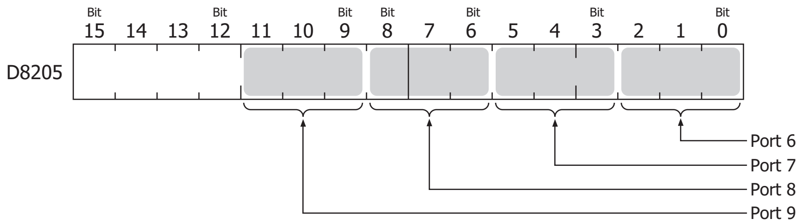
The external device uses the DSR signal to indicate that it is ready to receive data or to tell the FC6A Series MICROSmart its status, such as if it is transmitting valid data.

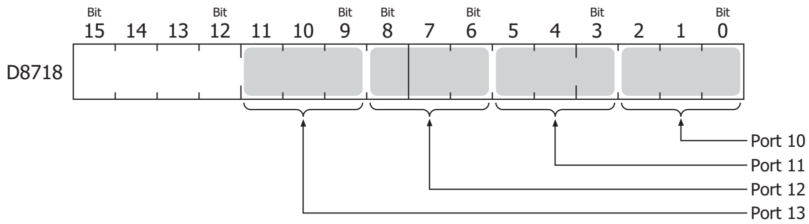
The FC6A Series MICROSmart determines whether or not to transmit and receive data depending on the status of the DSR signal of the external device.

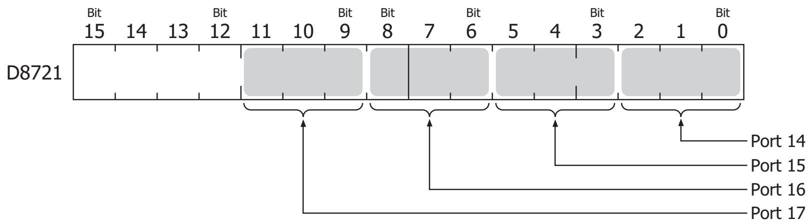
The DSR signal can be controlled by setting values in D8105, D8205, D8718, D8721, D8724, D8727, D8730, and D8733. These registers are only valid during user communication.

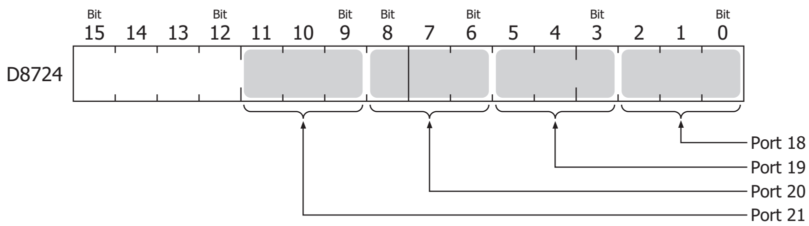
The allocation of communication ports in the devices (bit assignment) is as follows.

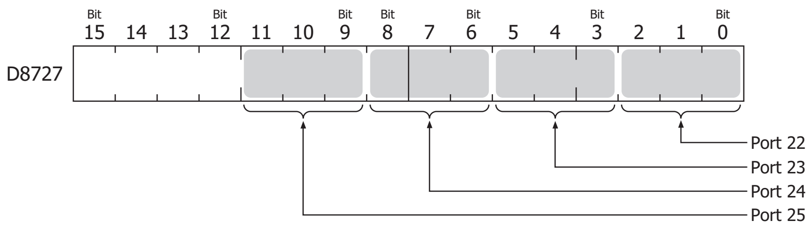


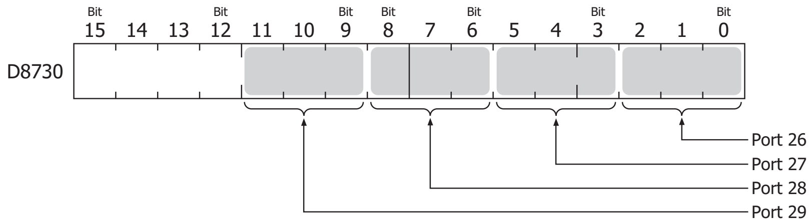


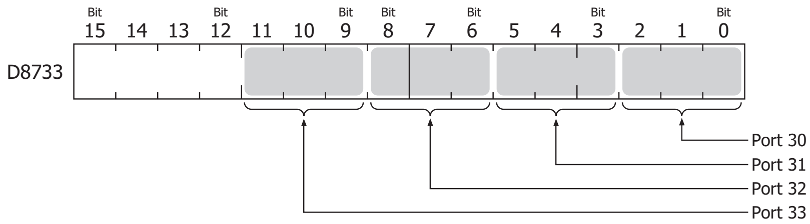


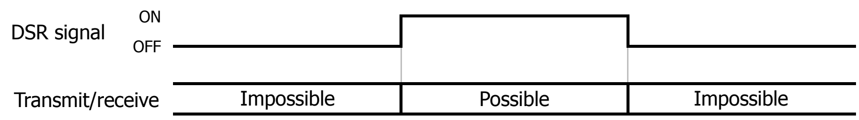
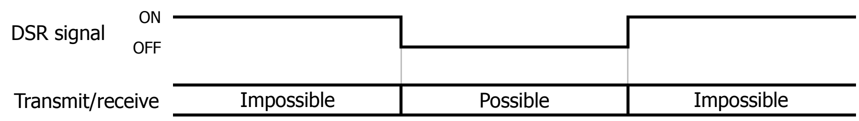
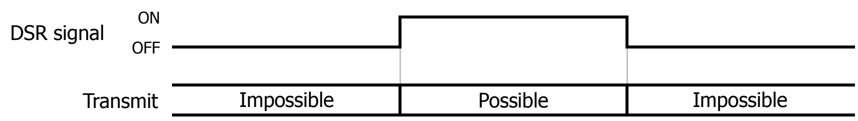
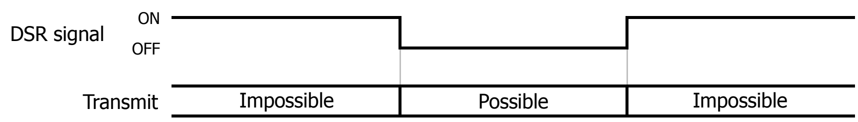










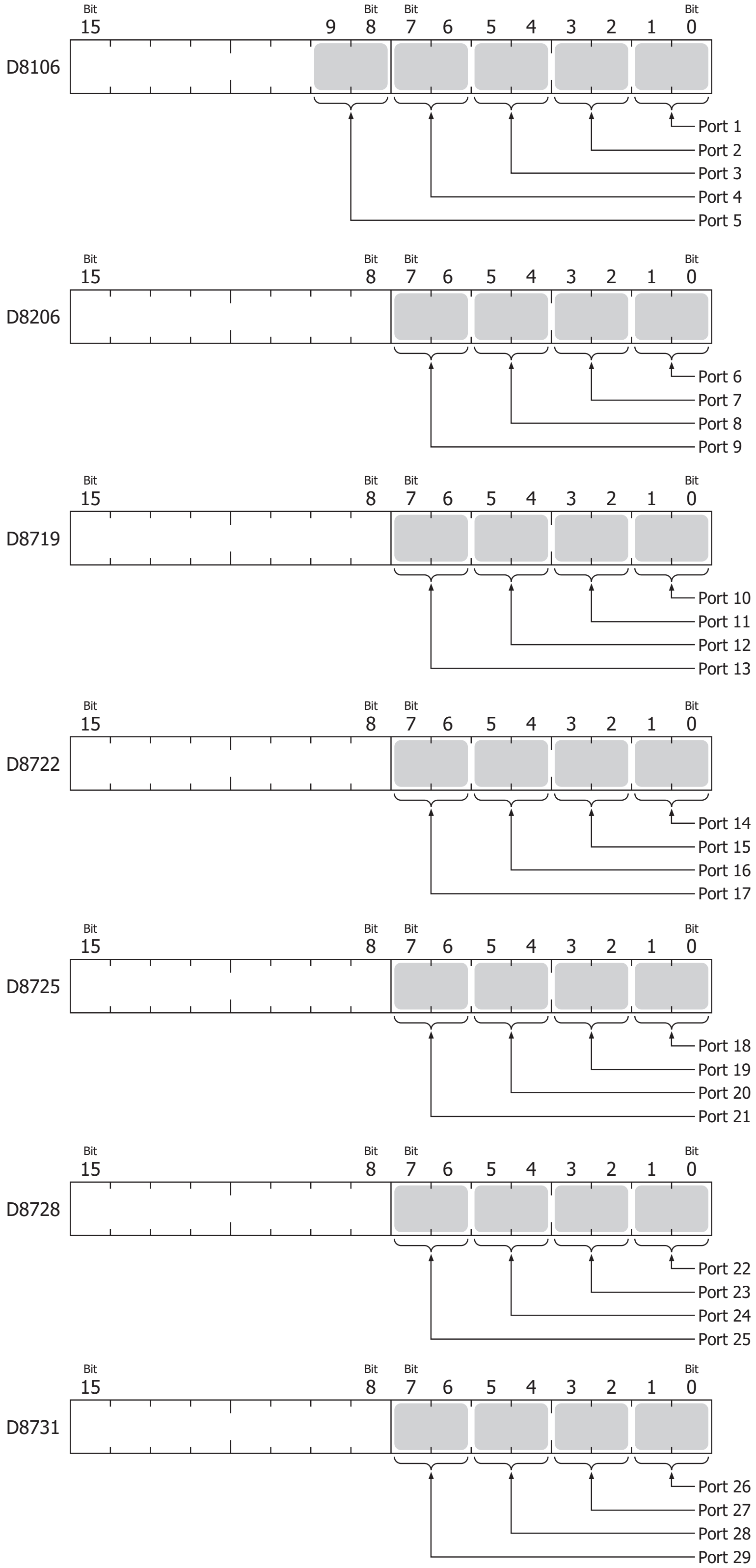
3-bit Binary Value	Description
000	DSR is not used for data flow control. Use this status if DSR signal control is not required.
001	When DSR is on, the FC6A Series MICROSmart can transmit and receive data. 
010	When DSR is off, the FC6A Series MICROSmart can transmit and receive data. 
011	When DSR is on, the FC6A Series MICROSmart can transmit data. This function is usually called “Busy Control” and is used for controlling transmission to a remote terminal with a slow processing speed, such as a printer. When the remote terminal is busy, data input to the remote terminal is restricted. 
100	When DSR is off, the FC6A Series MICROSmart can transmit data. 
≥ **101**	The operation is the same as the setting "000". DSR is not used for data flow control.

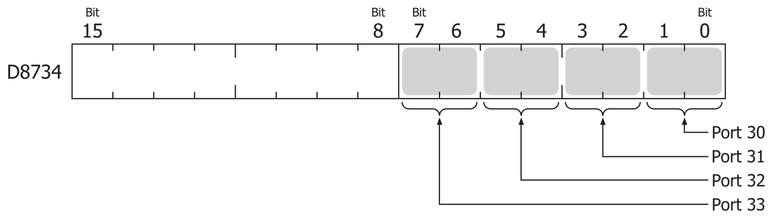
DTR Output Control Signal Option

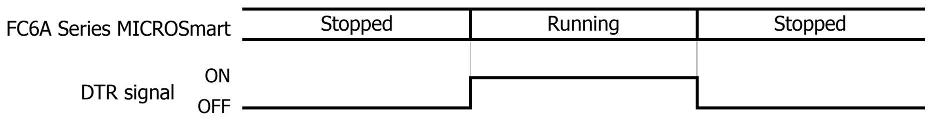
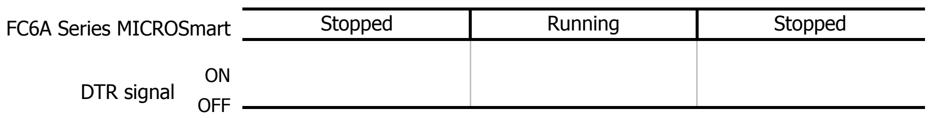
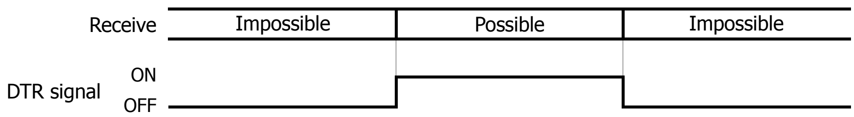
The FC6A Series MICROSmart tells the external device of its control status and transmit/receive status using the DTR signal.

The DTR signal can be controlled by setting values in D8106, D8206, D8719, D8722, D8725, D8728, D8731, and D8734. These registers are only valid during user communication.

The allocation of communication ports in the devices (bit assignment) is as follows.





<table>
<tr><th>2-bit Binary Value</th><th>Description</th></tr>
<tr><td>00</td><td>While the FC6A Series MICROSmart is running, DTR is on whether the FC6A Series MICROSmart is transmitting or receiving data. While the FC6A Series MICROSmart is stopped, DTR remains off. Use this option to indicate the FC6A Series MICROSmart operating status.

</td></tr>
<tr><td>01</td><td>Whether the FC6A Series MICROSmart is running or stopped, DTR remains off.

</td></tr>
<tr><td>10</td><td>While the FC6A Series MICROSmart can receive data, DTR is turned on. While the FC6A Series MICROSmart can not receive data, DTR remains off. Use this option when flow control of receive data is required.

</td></tr>
<tr><td>11</td><td>The operation is the same as the setting "00".</td></tr>
</table>

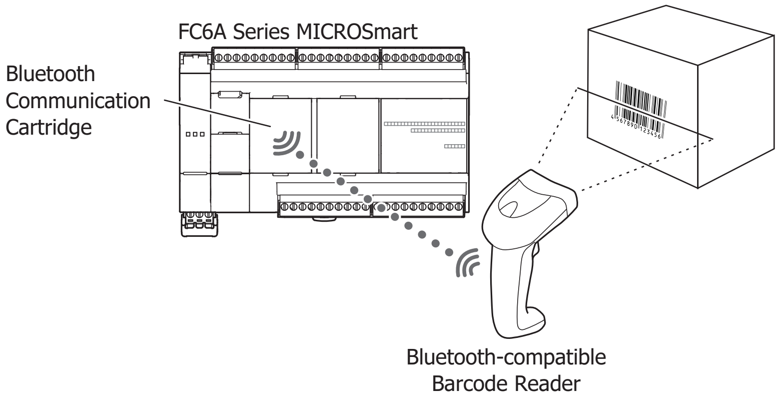
User Communication via Serial Communication (Bluetooth)

Connect a Bluetooth communication cartridge to the cartridge slot of the All-in-One CPU module/CAN J1939 All-in-One CPU module, the cartridge slot of a cartridge base module connected to the Plus CPU module, or the cartridge slot of the HMI module to use user communication over Bluetooth communication.

With user communication over Bluetooth communication, the FC6A MICROSmart can send and receive data between external devices connected over Bluetooth, such as barcode reader, using the TXD (user communication transmit) and RXD (user communication receive) instructions.

Example of Receiving Data Read with a Barcode Reader

Connect the Bluetooth communication cartridge to cartridge slot 1 of the FC6A Series MICROSmart and communicate with the barcode reader.



For details on communication specifications and WindLDR settings, see "Bluetooth Communication" on page 9-1.

User Communication via Ethernet Communication

This section describes the Ethernet user communication. Ethernet user communication works on TCP/UDP protocol. The FC6A Series MICROSmart can be used as a user communication client/server. With Ethernet user communication instructions (ETXD and ERXD instructions), the FC6A Series MICROSmart can exchange the data with devices on the network.

Except for the port number and the allocation of the user communication receive instruction cancel flags, Ethernet user communication instructions (ETXD and ERXD instructions) are identical to TXD and RXD instructions. For details about TXD and RXD instructions, see "TXD (Transmit)" on page 5-2 and "RXD (Receive)" on page 5-10.

Ethernet User Communication Overview

The FC6A Series MICROSmart can be used as an Ethernet user communication (TCP) client/server. The Plus CPU module can also be used for user communication (UDP). It can be used simultaneously with the maintenance communication server, Modbus TCP server, and Modbus TCP client.

When using the FC6A Series MICROSmart user communication client, the FC6A Series MICROSmart can access and communicate with the server devices using the protocol of the server device. A maximum of eight connections for the All-in-One CPU module/ CAN J1939 All-in-One CPU module and 16 connections for the Plus CPU module can be allocated to each type of communication.

User communication client functions and configuration are described in "To use the FC6A Series MICROSmart as a user communication client" on page 5-41. User communication server functions and configuration are described in "User Communication Server" on page 5-47.

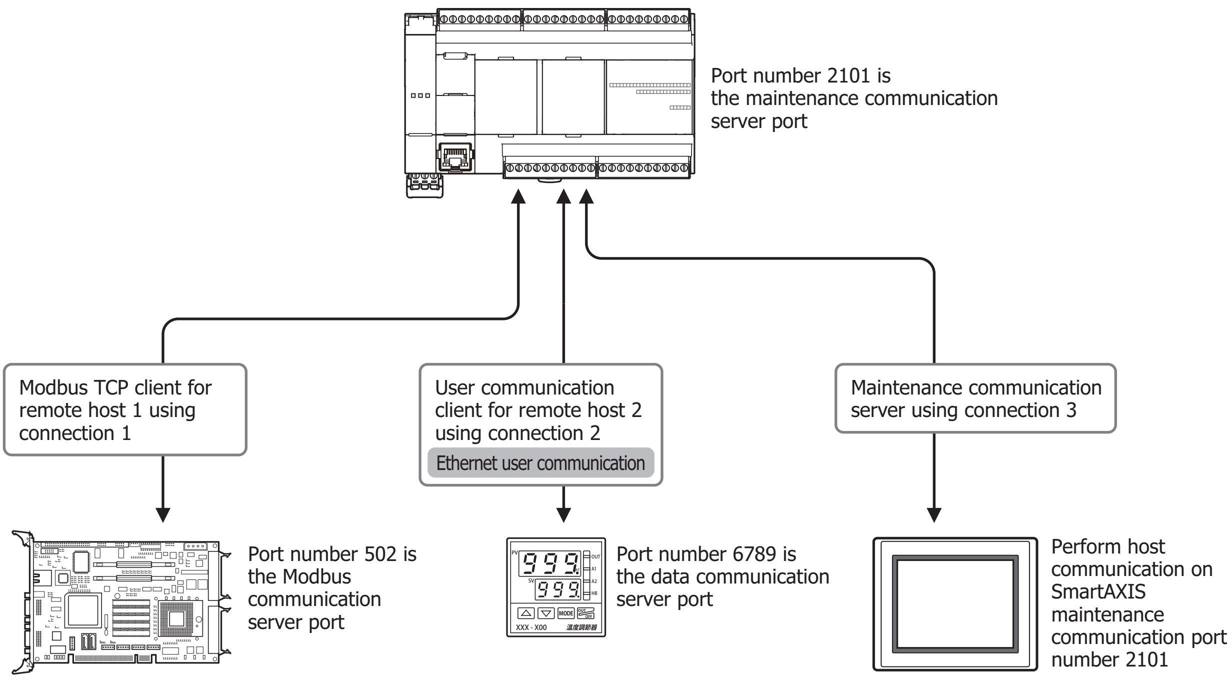
The FC6A Series MICROSmart supports the TCP/UDP protocol.

The FC6A Series MICROSmart can send data to and receive data from devices on a network by using the ETXD (User Communication Transmit over Ethernet) instruction and the ERXD (User Communication Receive over Ethernet) instruction.

The FC6A Series MICROSmart can be used as both an Ethernet user communication client and server.

Each of the eight connections possessed by the FC6A Series MICROSmart can be allocated to different types of communication. Ethernet user communication can simultaneously use the maintenance communication server, Modbus TCP server, and Modbus TCP client.

Ethernet communication example using three connections



FC6A Series MICROSmart function area settings connection settings

Connection	Communication Protocol	Other Settings
1	Modbus TCP client	Destination: Remote host 1
2	User communication client	Destination: Remote host 2
3	Maintenance communication server	Port number: 2101

Remote host table

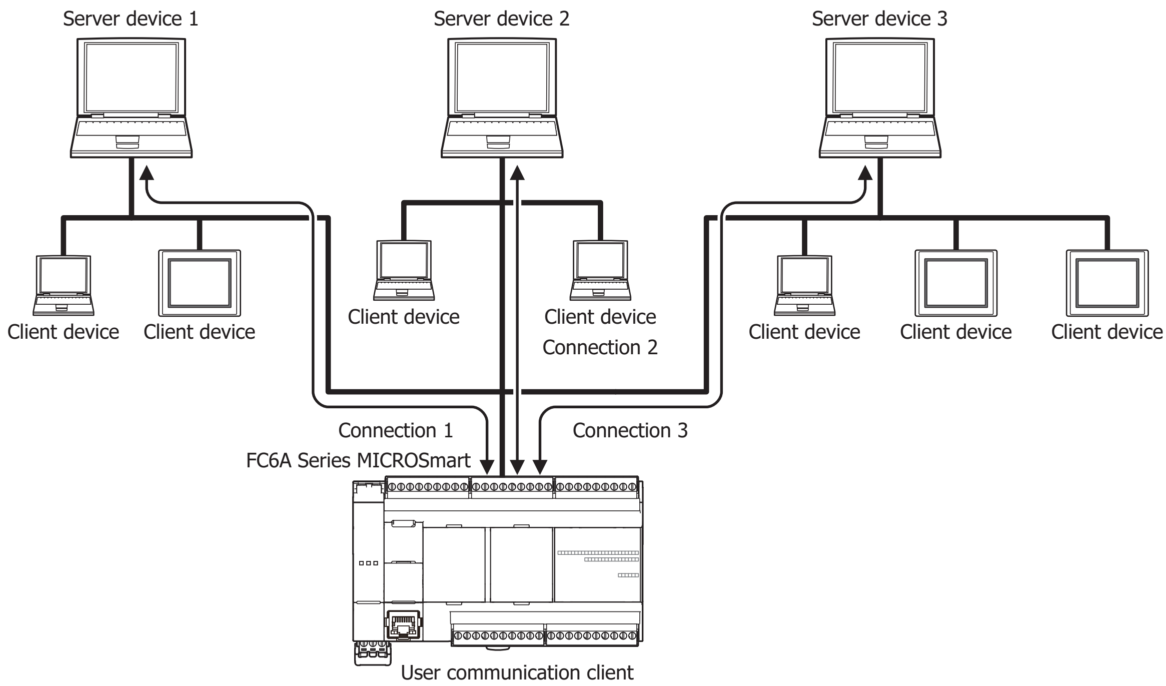
Remote Host Number	IP Address	Port Number
1	192.168.0.12	502
2	192.168.0.13	6789

To use the FC6A Series MICROSmart as a user communication client

Connect the FC6A Series MICROSmart to the server device via the network and communicate with the server device using the Ethernet user communication instructions.

A maximum of eight connections for the All-in-One CPU module/CAN J1939 All-in-One CPU module and 16 connections for the Plus CPU module can be allocated to user communication clients, which allows the FC6A Series MICROSmart to simultaneously connect to and communicate with different server devices.

When three connections are allocated to user communication client

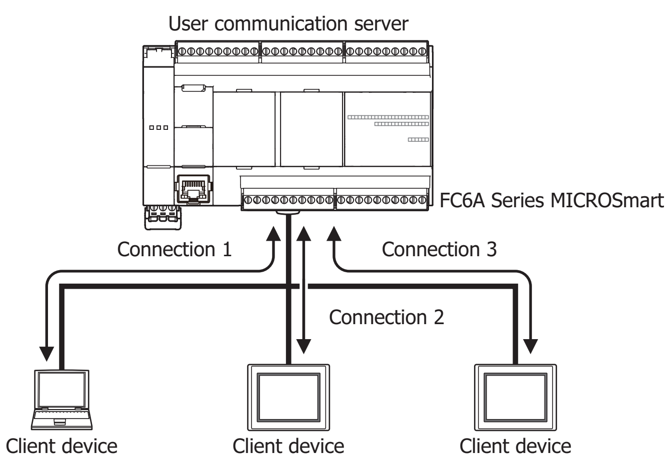


To use the FC6A Series MICROSmart as a user communication server

The client devices connect to the FC6A Series MICROSmart and the FC6A Series MICROSmart communicates with the client devices using Ethernet user communication instructions.

A maximum of eight connections for the All-in-One CPU module/CAN J1939 All-in-One CPU module and 16 connections for the Plus CPU module can be allocated to user communication servers, which enables FC6A Series MICROSmart to simultaneously connect to and communicate with a maximum of eight different client devices or a maximum of 16 different client devices, respectively.

When three connections are allocated to user communication server

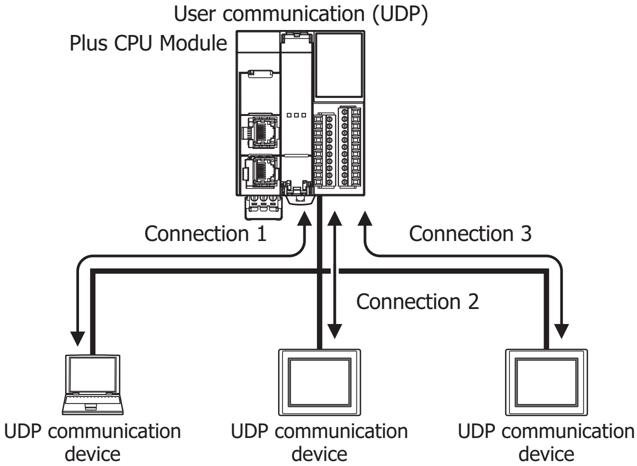


To use the FC6A Series MICROSmart for user communication (UDP)

The Plus CPU module can be connected to a UDP communication device and communicate with that device using the Ethernet user communication instructions.

The Plus CPU module can simultaneously connect to and communicate with a maximum of 16 UDP communication devices by allocating a maximum of 16 connections to user communication (UDP).

When three connections are allocated to user communication (UDP)



User Communication Client

When a client connection is configured as the user communication client, the FC6A Series MICROSmart communicates with the specified server device according to the settings configured in the ETXD and ERXD instructions that are programmed for the client connection. The remote host number and other communication settings can be configured in the Connection Settings tab in the Function Area Settings dialog box.

Specifications (User Communication Client)

Item	User Communication Client
Remote Host Number	1 to 255 Switch the remote host number with the special data registers (D8268 to D8275 and D8774 to D8781).
Establish Connection	• When ETXD/ERXD Instructions are executed • When the FC6A Series MICROSmart starts to run*1
Disconnect Connection	• When the FC6A Series MICROSmart is stopped • When special internal relays (M8222 to M8231 and M8355 to M8364) are turned on
Number of Remote Hosts that the FC6A Series MICROSmart Can Communicate Simultaneously	One remote host per a user communication client*2
Receive Timeout Time	100 to 25,500 ms (100 ms increments)*3

*1 Can be enabled or disabled in **Function Area Settings**, **Connection Settings**.
*2 A maximum of eight connections for the All-in-One CPU module/CAN J1939 All-in-One CPU module and 16 connections for the Plus CPU module can be allocated to user communication clients, which enables FC6A Series MICROSmart to simultaneously communicate with a maximum of eight server devices or a maximum of 16 server devices, respectively.
*3 Numeric values not in parentheses are the default settings.

Switching Remote Host Numbers

When user communication clients have been configured, the communication devices can be changed by using the special data registers (D8268 to D8275 and D8774 to D8781). Specify the remote host number of the communication devices to change to with the special data registers. If a value other than 1 to 255 is specified, the remote host number set in the Function Area Settings will be used.

Special Data Register	Description
D8268	Remote Host Number of Connection 1 (1 to 255)
D8269	Remote Host Number of Connection 2 (1 to 255)
D8270	Remote Host Number of Connection 3 (1 to 255)
D8271	Remote Host Number of Connection 4 (1 to 255)
D8272	Remote Host Number of Connection 5 (1 to 255)
D8273	Remote Host Number of Connection 6 (1 to 255)
D8274	Remote Host Number of Connection 7 (1 to 255)
D8275	Remote Host Number of Connection 8 (1 to 255)
D8774*1	Remote Host Number of Connection 9 (1 to 255)
D8775*1	Remote Host Number of Connection 10 (1 to 255)
D8776*1	Remote Host Number of Connection 11 (1 to 255)
D8777*1	Remote Host Number of Connection 12 (1 to 255)
D8778*1	Remote Host Number of Connection 13 (1 to 255)
D8779*1	Remote Host Number of Connection 14 (1 to 255)
D8780*1	Remote Host Number of Connection 15 (1 to 255)
D8781*1	Remote Host Number of Connection 16 (1 to 255)

*1 Plus CPU module only.

Establishing/Disconnecting User Communication Client Connections

When user communication clients are configured, connections are established on TCP protocol. The connections are established when ETXD/ERXD instructions are executed or when the FC6A Series MICROSmart starts running (See above)*1. Once a connection is established, the connection will be kept open until either the FC6A Series MICROSmart is stopped or a special internal relay (M8222 to M8231 and M8355 to M8364) allocated to the connection is turned on.

Special Internal Relay	Description	Operation
M8222	Connection 1 Disconnect	When the relay is turned on, the corresponding connection is disconnected.
M8223	Connection 2 Disconnect	
M8224	Connection 3 Disconnect	
M8225	Connection 4 Disconnect	
M8226	Connection 5 Disconnect	
M8227	Connection 6 Disconnect	
M8230	Connection 7 Disconnect	
M8231	Connection 8 Disconnect	
M8355*2	Connection 9 Disconnect	
M8356*2	Connection 10 Disconnect	
M8357*2	Connection 11 Disconnect	
M8360*2	Connection 12 Disconnect	
M8361*2	Connection 13 Disconnect	
M8362*2	Connection 14 Disconnect	
M8363*2	Connection 15 Disconnect	
M8364*2	Connection 16 Disconnect	

*1 Can be enabled or disabled in **Function Area Settings**, **Connection Settings**.
*2 Plus CPU module only.

User Communication Receive over Ethernet Instruction (ERXD) Cancel Flag

The allocation of the user communication receive instruction cancel flags for each client connection is shown in the table below. For details about the user communication receive instruction cancel flag, see "User Communication Receive Instruction Cancel Flag" on page 5-22.

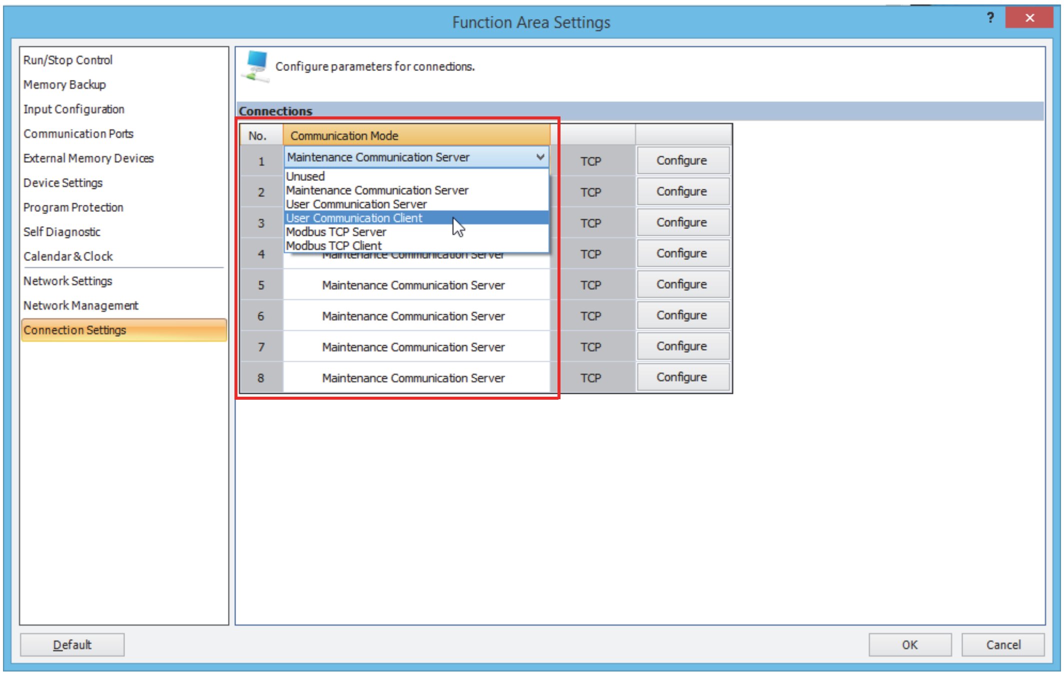
Special Internal Relay	Description
M8200	User Communication Receive Instruction Cancel Flag (Connection 1)
M8201	User Communication Receive Instruction Cancel Flag (Connection 2)
M8202	User Communication Receive Instruction Cancel Flag (Connection 3)
M8203	User Communication Receive Instruction Cancel Flag (Connection 4)
M8204	User Communication Receive Instruction Cancel Flag (Connection 5)
M8205	User Communication Receive Instruction Cancel Flag (Connection 6)
M8206	User Communication Receive Instruction Cancel Flag (Connection 7)
M8207	User Communication Receive Instruction Cancel Flag (Connection 8)
M8334*1	User Communication Receive Instruction Cancel Flag (Connection 9)
M8335*1	User Communication Receive Instruction Cancel Flag (Connection 10)
M8336*1	User Communication Receive Instruction Cancel Flag (Connection 11)
M8337*1	User Communication Receive Instruction Cancel Flag (Connection 12)
M8340*1	User Communication Receive Instruction Cancel Flag (Connection 13)
M8341*1	User Communication Receive Instruction Cancel Flag (Connection 14)
M8342*1	User Communication Receive Instruction Cancel Flag (Connection 15)
M8343*1	User Communication Receive Instruction Cancel Flag (Connection 16)

*1 Plus CPU module only.

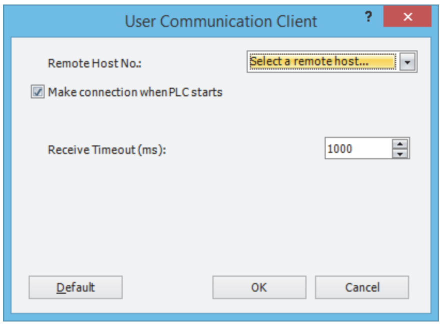
Programming WindLDR (User Communication Client)

To use the user communication client, configure the user client communication settings in the **Function Area Settings** dialog box and then download the user program to the FC6A Series MICROSmart.

1. Select **Configuration** from the WindLDR menu bar, and then click **Connection Settings**.
   The **Function Area settings** dialog box appears.
2. Select the **User Communication Client** as the communication mode for the client connection 1.

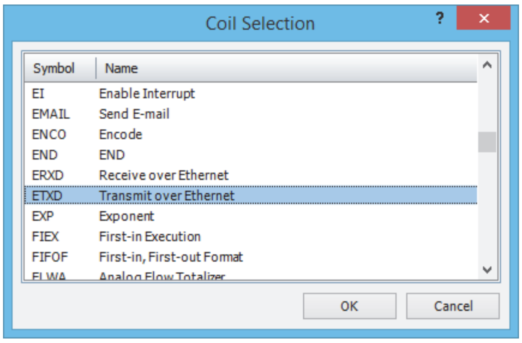


The **User Communication Client** dialog box appears.

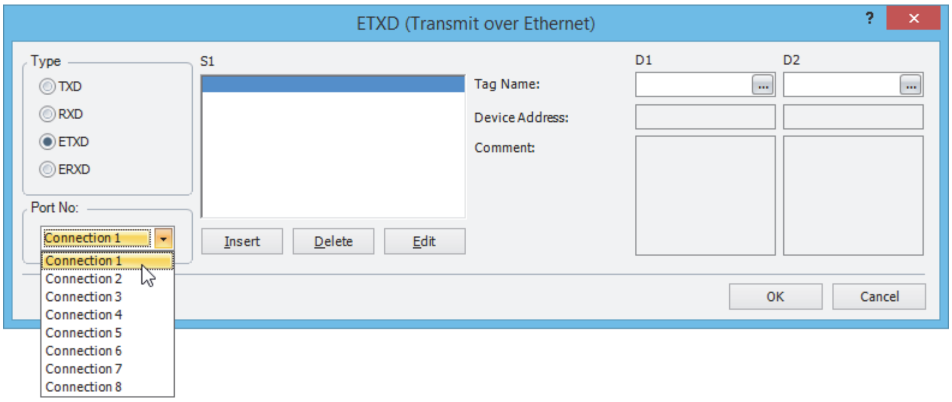


Configure the remote host number and receive timeout. If you want the FC6A Series MICROSmart to establish the connection when it starts to run, select "Make Connection when PLC starts." Click **OK** button to close the dialog box.

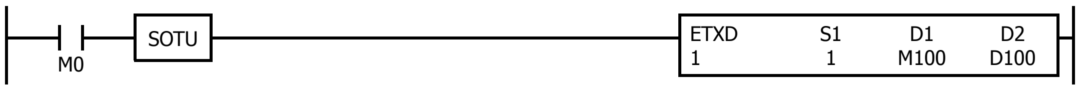
3. Edit the user program.
   To insert Ethernet user communication instructions to the ladder editor, select the Ethernet user communication instructions (ETXD or ERXD instructions) in the **Coil Selection** dialog box.



The **ETXD (Transmit over Ethernet)** dialog box appears.



Select **ETXD** (Transmit over Ethernet) to transmit data or **ERXD** (Receive over Ethernet) to receive data as the instruction type. Select the client connection from 1 through 8 and designate S1, D1, and D2. Click **OK** button to close the dialog box.



4. Download the user program.
   The settings for the user communication client have been finished.

The specifications of Ethernet user communication instructions (ETXD and ERXD instructions) are identical to TXD and RXD instructions. For details about TXD and RXD instructions, see "TXD (Transmit)" on page 5-2 and "RXD (Receive)" on page 5-10.

User Communication Server

When a server connection is configured as the user communication server, a client device can access and communicate with the FC6A Series MICROSmart. The FC6A Series MICROSmart communicates with the client device according to the settings configured in the ETXD and ERXD instructions that are programmed for the server connection. The local host number and other communication settings can be configured in the **Connection Settings** tab in the **Function Area Settings** dialog box.

Specifications (User Communication Server)

Item	User Communication Server
Local Host Port Number	2102 to 2117 (Can be changed between 0 and 65535)
Number of Clients That Can Simultaneously Communicate with the FC6A Series MICROSmart	One client per a user communication server*1
Receive Timeout Time	100 to 25,500 ms (100 ms increments)

*1 A maximum of eight connections for the All-in-One CPU module/CAN J1939 All-in-One CPU module and 16 connections for the Plus CPU module can be allocated to user communication servers, which enables a maximum of eight client devices or a maximum of 16 client devices to connect to and communicate with the FC6A Series MICROSmart, respectively.

User Communication Receive over Ethernet Instruction (ERXD) Cancel Flag

The allocation of the user communication receive instruction cancel flags for each server connection is shown in the table below. For details about the user communication receive instruction cancel flag, see "User Communication Receive Instruction Cancel Flag" on page 5-22.

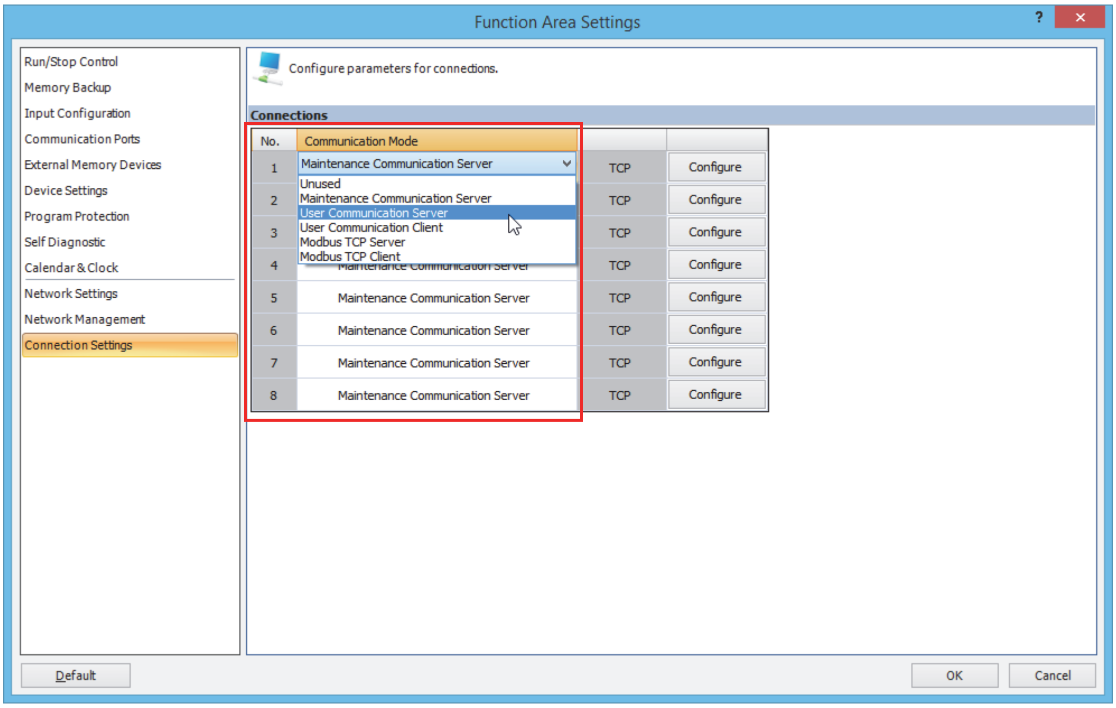
Device Address	Description
M8200	User Communication Receive Instruction Cancel Flag (Connection 1)
M8201	User Communication Receive Instruction Cancel Flag (Connection 2)
M8202	User Communication Receive Instruction Cancel Flag (Connection 3)
M8203	User Communication Receive Instruction Cancel Flag (Connection 4)
M8204	User Communication Receive Instruction Cancel Flag (Connection 5)
M8205	User Communication Receive Instruction Cancel Flag (Connection 6)
M8206	User Communication Receive Instruction Cancel Flag (Connection 7)
M8207	User Communication Receive Instruction Cancel Flag (Connection 8)
M8334*1	User Communication Receive Instruction Cancel Flag (Connection 9)
M8335*1	User Communication Receive Instruction Cancel Flag (Connection 10)
M8336*1	User Communication Receive Instruction Cancel Flag (Connection 11)
M8337*1	User Communication Receive Instruction Cancel Flag (Connection 12)
M8340*1	User Communication Receive Instruction Cancel Flag (Connection 13)
M8341*1	User Communication Receive Instruction Cancel Flag (Connection 14)
M8342*1	User Communication Receive Instruction Cancel Flag (Connection 15)
M8343*1	User Communication Receive Instruction Cancel Flag (Connection 16)

*1 Plus CPU module only.

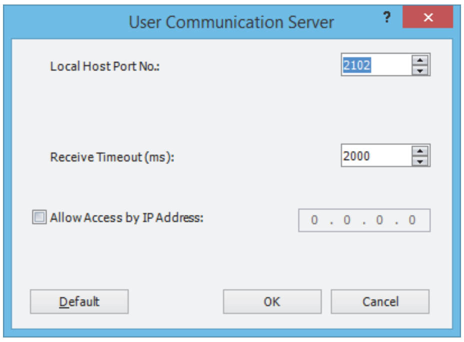
Programming WindLDR (User Communication Server)

To use the user communication server, configure the user communication server settings in the Function Area Settings dialog box and then download the user program to the FC6A Series MICROSmart.

1. Select **Configuration** from the WindLDR menu bar, and then click **Connection Settings**.
   The **Function Area Settings** dialog box appears.
2. Select the **User Communication Server** as the communication mode for the server connection 1.

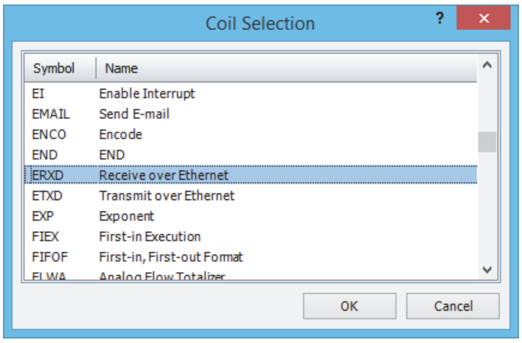


The **User Communication Server** dialog box appears.

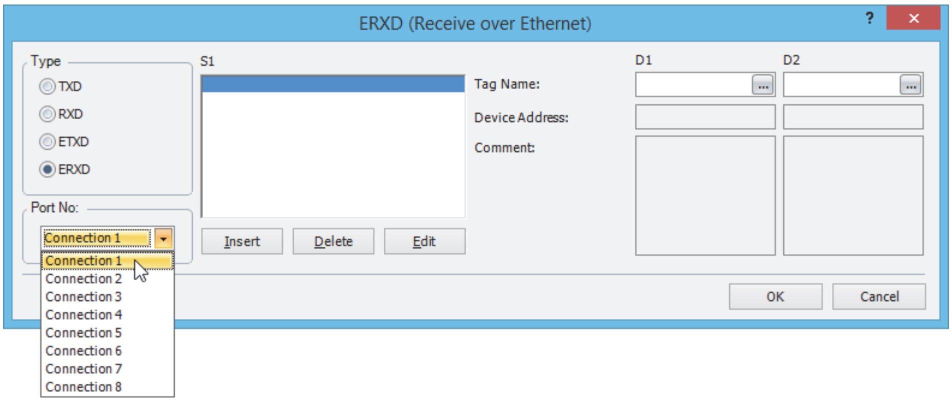


Configure the local host port number and receive timeout. If you want to restrict the access using IP address, configure the allowed IP address.

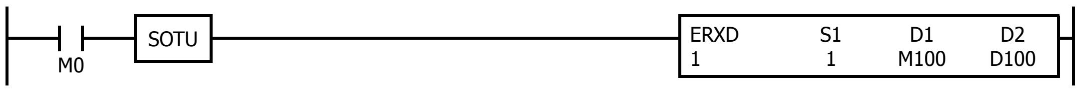
3. Edit the user program.
To insert Ethernet user communication instructions to the ladder editor, select the Ethernet user communication instructions (ETXD or ERXD instructions) in the **Coil Selection** dialog box.



The **ERXD (Receive over Ethernet)** dialog box appears.



Select **ETXD** (Transmit over Ethernet) to transmit data and **ERXD** (Receive over Ethernet) to receive data as the instruction type. Select the server connection from 1 through 16 and designate S1, D1, and D2. Click **OK** button to close the dialog box.



4. Download the user program.
The settings for the user communication server have been finished.

The specifications of Ethernet user communication instructions (ETXD and ERXD instructions) are identical to TXD and RXD instructions. For details about TXD and RXD instructions, see "TXD (Transmit)" on page 5-2 and "RXD (Receive)" on page 5-10.

User Communication (UDP)

The Plus CPU module can communicate with UDP communication devices through a user communication server according to the ETXD (User Communication Transmit over Ethernet) and ERXD (User Communication Receive over Ethernet) instructions set to the corresponding connection number.

The port number settings and the communication settings of the user communication server are configured on the **Configuration** tab, in the **Function Area Settings** group, under **Connection Settings**.

Specifications

Item	User Communication Server
Remote host number	1 to 255
Local host port number	0 to 65535
Allow access by IP address	0.0.0.0 to 255.255.255.255

Allocation of Receive Cancel Flags for ERXD (User Communication Receive over Ethernet) Instructions

The following table shows the allocation of receive cancel flags for each connection.

When preparations for receiving data is finished and the instruction is in Receiving Data state (status code 32), and the corresponding user communication receive instruction cancel flag is turned on, execution of all receive instructions for the corresponding connection will be canceled. This is effective when receiving data is taking too long time and you want to cancel the execution of the receive instruction.

To activate a receive instruction that is canceled, turn off the user communication receive cancel flag, and then turn on the input to the receive instruction again.

User communication receive instruction cancel flags are allocated in special internal relays for each connection as follows.

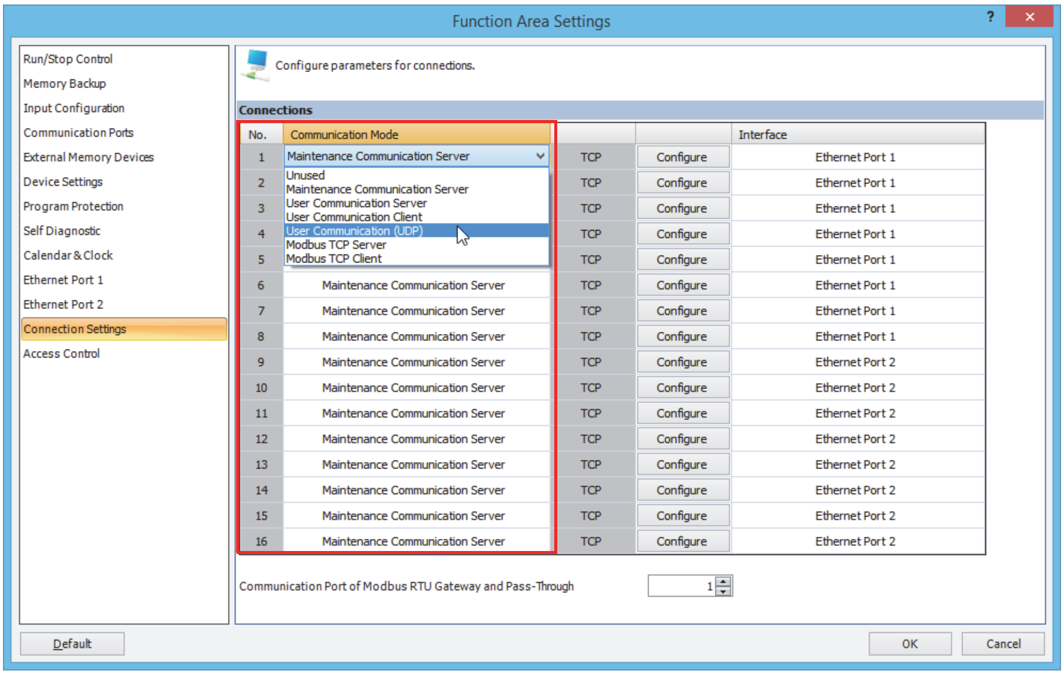
Special Internal Relay	Description
M8200	User Communication Receive Instruction Cancel Flag (Connection 1)
M8201	User Communication Receive Instruction Cancel Flag (Connection 2)
M8202	User Communication Receive Instruction Cancel Flag (Connection 3)
M8203	User Communication Receive Instruction Cancel Flag (Connection 4)
M8204	User Communication Receive Instruction Cancel Flag (Connection 5)
M8205	User Communication Receive Instruction Cancel Flag (Connection 6)
M8206	User Communication Receive Instruction Cancel Flag (Connection 7)
M8207	User Communication Receive Instruction Cancel Flag (Connection 8)
M8334	User Communication Receive Instruction Cancel Flag (Connection 9)
M8335	User Communication Receive Instruction Cancel Flag (Connection 10)
M8336	User Communication Receive Instruction Cancel Flag (Connection 11)
M8337	User Communication Receive Instruction Cancel Flag (Connection 12)
M8340	User Communication Receive Instruction Cancel Flag (Connection 13)
M8341	User Communication Receive Instruction Cancel Flag (Connection 14)
M8342	User Communication Receive Instruction Cancel Flag (Connection 15)
M8343	User Communication Receive Instruction Cancel Flag (Connection 16)

Programming WindLDR

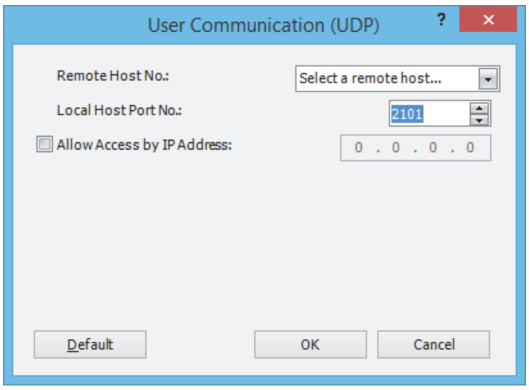
Configure the Plus CPU module for user communication (UDP).

The following example provides a explanation of the procedure to configure user communication (UDP) for connection 1 using WindLDR.

1. On the WindLDR **Configuration** tab, in the **Function Area Settings** group, click **Connection Settings**.
   The **Function Area Settings** dialog box is displayed.
2. Click **Communication Mode** for connection number 1 and select **User Communication (UDP)**.
   The **User Communication (UDP)** dialog box is displayed.



3. Configure **Remote Host No.** and **Local Host Port No**. To enable access restrictions by IP address, select the **Allow Access by IP Address** check box and configure the IP address to allow.
   For details on the remote host number, see "Remote Host List" on page 3-20.



4. Click **OK**.

This concludes configuring the settings for user communication (UDP).

Next, create a user program that uses the user communication instructions that correspond to the configured connection, and then download the user program to the Plus CPU module.

User Communication Error

When a user communication error occurs, an error code is stored in the data register designated as a transmit status in the TXD instruction or as a receive status in the RXD instruction. When multiple errors occur, the final error code overwrites all preceding errors and is stored in the status data register.

The status data register also contains transmit/receive status code. To extract a user communication error code from the status data register, divide the value by 16. The remainder is the user communication error code. See "D2 (Destination 2)" on page 5-8 for the Transmit Status and "D2 (Destination 2)" on page 5-21 for the Receive Status.

To correct the error, correct the user program by referring to the error causes described below:

User Communication Error Code

User Communication Error Code	Error Cause	Transmit/Receive Completion Output
1	Start inputs to more than 5 TXD instructions are on simultaneously.	Transmit completion outputs of the first 5 TXD instructions from the top of the ladder diagram are turned on.
2	Transmission destination busy timeout	The transmit completion output goes on.
3	Start inputs to more than 5 RXD instructions with a start delimiter are on simultaneously.	Among the first 5 RXD instructions from the top of the ladder diagram, receive completion outputs of RXD instructions go on if the start delimiter matches the first byte of the received data.
4	While a RXD instruction without a start delimiter is executed, another RXD instruction with or without a start delimiter is executed.	The receive completion output of the RXD instruction at a smaller address goes on.
5	While a RXD instruction with a start delimiter is executed, another RXD instruction with the same start delimiter is executed.	No effect on the receive completion output.
7	The first bytes of received data do not match the specified start delimiter.	No effect on the receive completion output. If incoming data with a matching start delimiter is received subsequently, the receive completion output goes on.
8	When ASCII to binary or ASCII to BCD conversion is specified in the receive format, any code other than 0 to 9 and A to F is received. (These codes are regarded as 0 during conversion.)	The receive completion output goes on.
9	BCC calculated from the RXD instruction does not match the BCC appended to the received data.	The receive completion output goes on.
10	Constants including the end delimiter code specified in the RXD instruction do not match the received constants.	The receive completion output goes on.
11	Receive timeout between characters (After receiving one byte of data, the next byte is not received in the period specified for the receive timeout value.)	The receive completion output goes on.
12	Overrun error (Before the receive processing is completed, the next data is received.)	The receive completion output goes off.
13	Framing error (Detection error of start bit or stop bit)	No effect on the completion output.
14	Parity check error (Error is found in the parity check.)	No effect on the completion output.
15	A user communication instruction was used even though the port settings or the connection settings were not set to user communication mode.	No effect on the completion output.

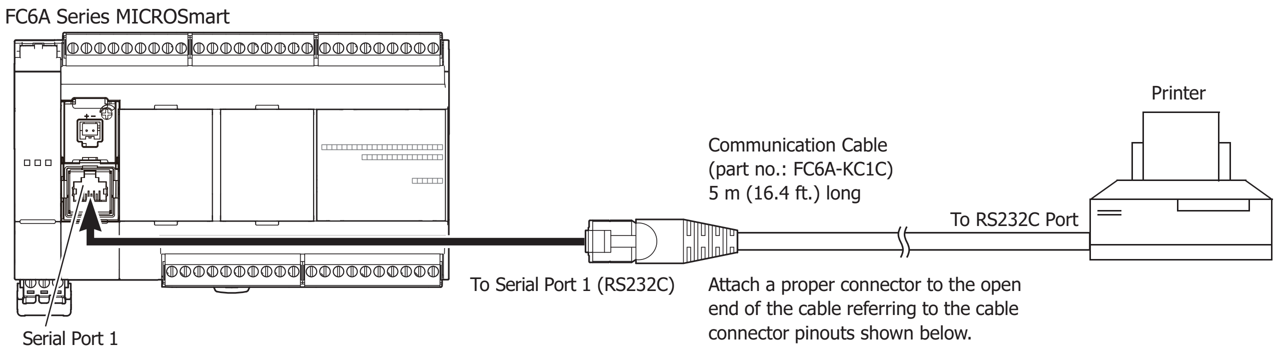
ASCII Character Code Table

Upper Bit / Lower Bit		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		NUL	DLE	SP	0	@	P	`	p								
	Decimal	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1		SOH	DC1	!	1	A	Q	a	q								
	Decimal	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
2		STX	DC2	″	2	B	R	b	r								
	Decimal	2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
3		ETX	DC3	#	3	C	S	c	s								
	Decimal	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
4		EOT	DC4	$	4	D	T	d	t								
	Decimal	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
5		ENQ	NAK	%	5	E	U	e	u								
	Decimal	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
6		ACK	SYN	&	6	F	V	f	v								
	Decimal	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
7		BEL	ETB	’	7	G	W	g	w								
	Decimal	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
8		BS	CAN	(	8	H	X	h	x								
	Decimal	8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
9		HT	EM	)	9	I	Y	i	y								
	Decimal	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
A		LF	SUB	*	:	J	Z	j	z								
	Decimal	10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
B		VT	ESC	+	;	K	[	k	{								
	Decimal	11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
C		FF	FS	,	<	L	\	l	|								
	Decimal	12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
D		CR	GS	-	=	M	]	m	}								
	Decimal	13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
E		SO	RS	.	>	N	^	n	~								
	Decimal	14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
F		SI	US	/	?	O	_	o									
	Decimal	15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

Sample Program – User Communication TXD

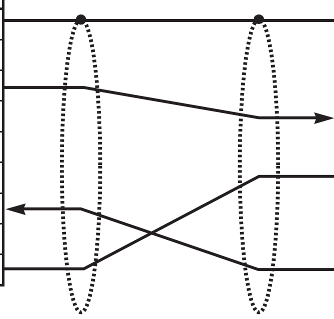
This example demonstrates a program to send data to a printer using the user communication TXD1 (transmit) instruction.

System Setup



Cable Connection and Pinouts

Description		Color	Pin
Shield		—	**Cover**
RXD	Receive Data	White/Orange	**1**
TXD	Transmit Data	Orange	**2**
DTR	Equipment Ready	White/Green	**3**
A		Blue	**4**
B		White/Blue	**5**
DSR	Data Set Ready	Green	**6**
NC	No Connection	White/Brown	**7**
SG	Signal Ground	Brown	**8**



Pin	Description	
	Shield	
1	NC	No Connection
2	NC	No Connection
3	DATA	Receive Data
4	NC	No Connection
5	SG	Signal Ground
6	NC	No Connection
7	NC	No Connection
8	BUSY	Busy Signal
9	NC	No Connection

The name of BUSY terminal differs depending on printers, such as DTR. The function of this terminal is to send a signal to remote equipment whether the printer is ready to print data or not. Since the operation of this signal may differ depending on printers, confirm the operation before connecting the cable.

> ⚠ **Caution** Do not connect any wiring to the NC (no connection) pins; otherwise, the FC6A Series MICROSmart and the printer may not work correctly and may be damaged.

Description of Operation

The data of counter C2 and data register D30 are printed every minute. A printout example is shown on the right.

Programming Special Data Register

Special data register D8105 is used to monitor the BUSY signal and to control the transmission of print data.

Special DR	Value	Description
D8105	3 (011)	While DSR is on (not busy), the FC6A Series MICROSmart sends data. While DSR is off (busy), the FC6A Series MICROSmart stops data transmission. If the off duration exceeds a limit (approx. 5 s), a transmission busy timeout error will occur, and the remaining data is not sent. The transmit status data register stores an error code. See "User Communication Error" on page 5-52.

Printout Example

```
--- PRINT TEST ---

11H 00M

CNT2...0050
D030...3854

--- PRINT TEST ---

11H 01M

CNT2...0110
D030...2124
```

The FC6A Series MICROSmart monitors the DSR signal to prevent the receive buffer of the printer from overflowing. For the DSR signal, see "DSR Input Control Signal Option" on page 5-35.

Setting User Communication Mode in WindLDR Function Area Settings

Since this example uses the Serial Port 1 (RS232C), select User Protocol for Port 1 in the **Function Area Settings** using WindLDR. See "Programming WindLDR" on page 5-25.

Setting Communication Parameters

Set the communication parameters to match those of the printer. See "Programming WindLDR" on page 5-25. For details of the communication parameters of the printer, see the user's manual for the printer. An example is shown below:

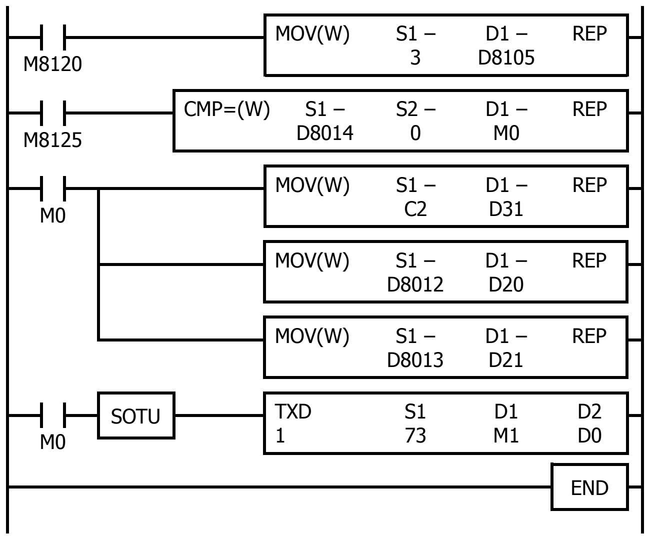
Communication Parameters:

Baud rate:	9,600 bps
Data bits:	8
Parity check:	None
Stop bits:	1

Note: The receive timeout value is used for the RXD instruction in the user communication mode. Since this example uses only the TXD instruction, the receive timeout value has no effect.

Ladder Diagram

The second data stored in special data register D8014 is compared with 0 using the CMP= (compare equal to) instruction. Each time the condition is met, the TXD1 instruction is executed to send the C2 and D30 data to the printer. A counting circuit for counter C2 is omitted from this sample program.



M8120 is the initialize pulse special internal relay.

3 → D8105 to enable the DSR option for busy control.

M8125 is the in-operation output special internal relay.

CMP=(W) compares the D8014 second data with 0.

When the D8014 data equals 0 second, M0 is turned on.

Counter C2 current value is moved to D31.

D8012 hour data is moved to D20.

D8013 minute data is moved to D21.

TXD1 is executed to send 73-byte data through the Serial Port 1 (RS232C) to the printer.

Details of S1 Settings in the Transmit Instruction

SP 20h	SP 20h	SP 20h	– 2Dh	– 2Dh	– 2Dh	SP 20h	P 50h	R 52h	I 49h	N 4Eh	T 54h	SP 20h	T 54h
E 45h	S 53h	T 54h	SP 20h	– 2Dh	– 2Dh	– 2Dh	CR 0Dh	LF 0Ah	CR 0Dh	LF 0Ah	SP 20h	SP 20h	SP 20h
D20 Conversion: BCD→ASCII Digits: 2 REP: 01													
H 48h	SP 20h												
D21 Conversion: BCD→ASCII Digits: 2 REP: 01													
M 4Dh	CR 0Dh	LF 0Ah	CR 0Dh	LF 0Ah									
SP 20h	SP 20h	SP 20h	C 43h	N 4Eh	T 54h	2 32h	. 2Eh	. 2Eh	. 2Eh				
D31 Conversion: BCD→ASCII Digits: 4 REP: 01													
CR 0Dh	LF 0Ah	SP 20h	SP 20h	SP 20h	D 44h	0 30h	3 33h	0 30h	. 2Eh	. 2Eh	. 2Eh		
D30 Conversion: BCD→ASCII Digits: 4 REP: 01													
CR 0Dh	LF 0Ah	CR 0Dh	LF 0Ah										

D20 hour data is converted from BCD to ASCII, and 2 digits are sent.

D21 minute data is converted from BCD to ASCII, and 2 digits are sent

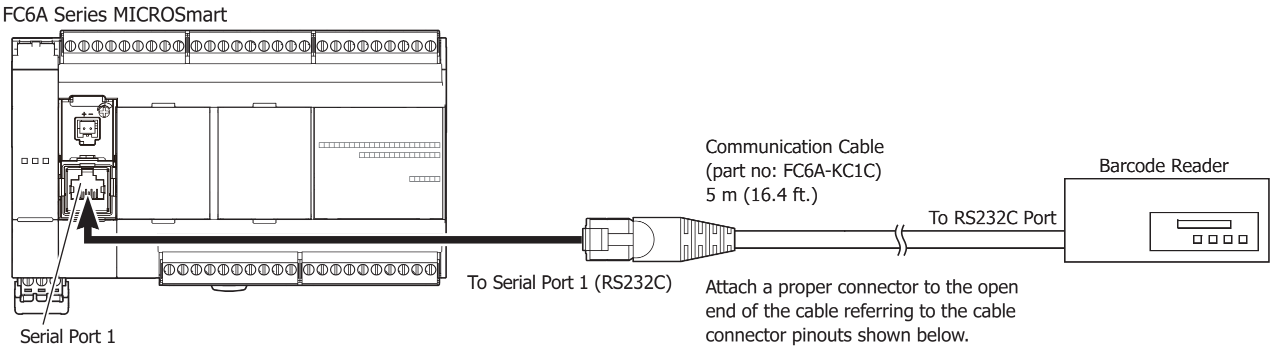
D31 counter C2 data is converted from BCD to ASCII, and 4 digits are sent.

D30 data is converted from BCD to ASCII, and 4 digits are sent.

Sample Program – User Communication RXD

This example demonstrates a program to receive data from a barcode reader with an RS232C port using the user communication RXD1 (receive) instruction.

System Setup



RJ45 Connector

Description		Color	Pin
Shield		—	**Cover**
RXD	Receive Data	White/Orange	**1**
TXD	Transmit Data	Orange	**2**
DTR	Equipment Ready	White/Green	**3**
A		Blue	**4**
B		White/Blue	**5**
DSR	Data Set Ready	Green	**6**
NC	No Connection	White/Brown	**7**
SG	Signal Ground	Brown	**8**

D-sub 25-pin Connector Pinouts

Pin	Description	
1	FG	Frame Ground
2	TXD1	Transmit Data
3	RXD1	Receive Data
7	GND	Ground

⚠ **Caution**	Do not connect any wiring to the NC (no connection) pins; otherwise, the FC6A Series MICROSmart and the barcode reader may not work correctly and may be damaged.

Description of Operation

A barcode reader is used to scan barcodes of 8 numerical digits. The scanned data is sent to the FC6A Series MICROSmart through the Serial Port 1 (RS232C) and stored to data registers. The upper 8 digits of the data are stored to data register D20 and the lower 8 digits are stored to data register D21.

Setting User Communication Mode in WindLDR Function Area Settings

Since this example uses the Serial Port 1 (RS232C), select User Protocol for Port 1 in the Function Area Settings using WindLDR. See "Programming WindLDR" on page 5-25.

Setting Communication Parameters

Set the communication parameters to match those of the barcode reader. See "Programming WindLDR" on page 5-25. For details of the communication parameters of the barcode reader, see the user's manual for the barcode reader. An example is shown below:

Communication Parameters:

Baud rate:	9,600 bps
Data bits:	7
Parity check:	Even
Stop bits:	1

Configuring Barcode Reader

The values shown below are an example of configuring a barcode reader. For actual settings, see the user's manual for the barcode reader.

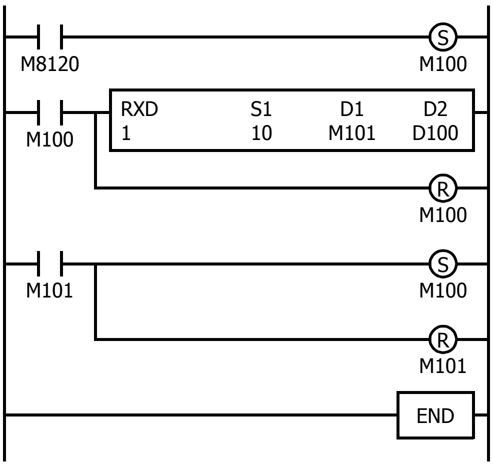
Synchronization Mode	Auto			
Read Mode	Single read or multiple read			
Communication Parameter	Baud rate: Parity check:	9,600 bps Even	Data bits: Stop bit:	7 1
Other Communication Settings	Header: Data echo back: Output timing: Data output filter: Sub serial:	02h No Output priority 1 No No	Terminator: BCR data output: Character suppress: Main serial input:	03h Yes No No
Comparison Preset Mode	Not used			

Device Addresses

M100	Input to start receiving barcode data
M101	Receive completion output for barcode data
M8120	Initialize pulse special internal relay
D20	Store barcode data (upper 4 digits)
D21	Store barcode data (lower 4 digits)
D100	Receive status data register for barcode data
D101	Receive data byte count data register

Ladder Diagram

When the FC6A Series MICROSmart starts operation, the RXD1 instruction is executed to wait for incoming data. When data receive is complete, the data is stored to data registers D20 and D21. The receive completion signal is used to execute the RXD1 instruction to wait for another incoming data.



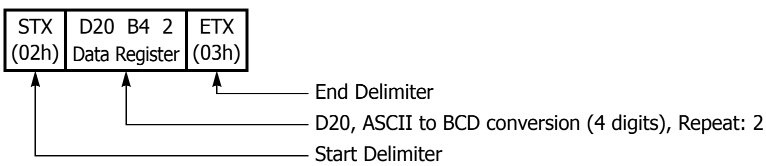
M8120 is the initialize pulse special internal relay used to set M100.

At the rising edge of M100, RXD1 is executed to be ready for receiving data.

Even after M100 is reset, RXD1 still waits for incoming data.

When data receive is complete, M101 is turned on, then M100 is set to execute RXD1 to receive the next incoming data.

RXD1 Data



BCC Calculation Examples

The FC6A Series MICROSmart can use three new BCC calculation formulas of ADD-2comp, Modbus ASCII, and Modbus RTU for transmit instructions TXD1, TXD2 and TXD3 and receive instructions RXD1, RXD2 and RXD3. These block check characters are calculated as described below.

ADD-2comp

Add the characters in the range from the BCC calculation start position to the byte immediately before the BCC, then invert the result bit by bit, and add 1.

1. Add the characters in the range from the BCC calculation start position to the byte immediately before the BCC.
2. Invert the result bit by bit, and add 1 (2's complement).
3. Store the result to the BCC position according to the designated conversion type (Binary to ASCII conversion or No conversion) and the designated quantity of BCC digits.

Example: Binary to ASCII conversion, 2 BCC digits

When the result of step **2** is 175h, the BCC will consist of 37h, 35h.

Modbus ASCII — Calculating the LRC (longitudinal redundancy check)

Calculate the BCC using LRC (longitudinal redundancy check) for the range from the BCC calculation start position to the byte immediately before the BCC.

1. Convert the ASCII characters in the range from the BCC calculation start position to the byte immediately before the BCC, in units of two characters, to make 1-byte hexadecimal data. (Example: 37h, 35h → 75h)
2. Add up the results of step **1**.
3. Invert the result bit by bit, and add 1 (2's complement).
4. Convert the lowest 1-byte data to ASCII characters. (Example: 75h → 37h, 35h)
5. Store the two digits to the BCC (LRC) position.
   If the BCC calculation range consists of an odd number of bytes, the BCC calculation results in an indefinite value. Modbus protocol defines that the BCC calculation range is an even number of bytes.

Modbus RTU — Calculating the CRC-16 (cyclic redundancy checksum)

Calculate the BCC using CRC-16 (cyclic redundancy checksum) for the range from the BCC calculation start position to the byte immediately before the BCC. The generation polynomial is: $X^{16} + X^{15} + X^2 + 1$.

1. Take the exclusive OR (XOR) of FFFFh and the first 1-byte data at the BCC calculation start position.
2. Shift the result by 1 bit to the right. When a carry occurs, take the exclusive OR (XOR) of A001h, then go to step **3**.
   If not, directly go to step **3**.
3. Repeat step **2**, shifting 8 times.
4. Take the exclusive OR (XOR) of the result and the next 1-byte data.
5. Repeat step **2** through step **4** up to the byte immediately before the BCC.
6. Swap the higher and lower bytes of the result of step **5**, and store the resultant CRC-16 to the BCC (CRC) position. (Example: 1234h → 34h, 12h)

6: MODBUS COMMUNICATION

Introduction

This chapter describes the Modbus communication functions for the FC6A Series MICROSmart.

Overview

For Modbus communication, the FC6A Series MICROSmart can be used as a Modbus RTU communication master and slave and Modbus TCP communication client and server to communicate with external devices.

Communication Ports Used in Modbus Communication

Supported models and communication ports are as follows.

Communication Port	All-in-One CPU Module			CAN J1939 All-in-One CPU Module	Plus CPU Module	
	16-I/O Type	24-I/O Type	40-I/O Type		Plus 16-I/O Type	Plus 32-I/O Type
Port 1	Yes	Yes	Yes	—	Yes*2	Yes*2
Port 2	Yes*1	Yes*1	Yes*1	Yes*1	Yes*2	Yes*2
Port 3	—	—	Yes*1	Yes*1	Yes*3	Yes*3
Port 4 to 33	Yes (Port 4 to 9)*4				Yes (Port 4 to 33)*5	
Ethernet Port 1	Yes	Yes	Yes	Yes	Yes	Yes
Ethernet Port 2	—	—	—	—	Yes	Yes
HMI-Ethernet Port	—	—	—	—	—	—

*1 When an RS232C communication cartridge or RS485 communication cartridge is connected.

*2 When a cartridge base module and an RS232C communication cartridge or RS485 communication cartridge are connected.

*3 When the HMI module and an RS232C communication cartridge or RS485 communication cartridge are connected.

*4 Up to three communication modules can be connected to the All-in-One CPU module and CAN J1939 All-in-One CPU module, expanding the communication ports 4 to 9.

*5 Up to 15 communication modules can be connected to the Plus CPU module, expanding the communication ports 4 to 33.

Modbus Communication Settings

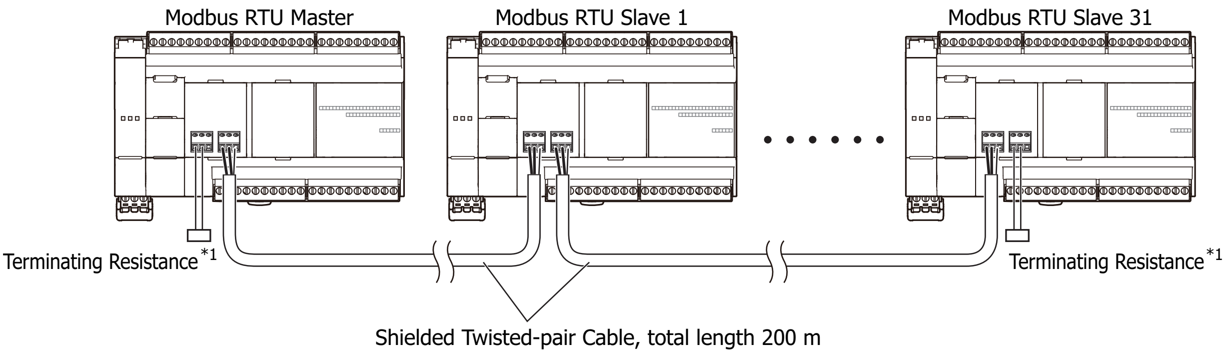
The Modbus communication settings supported by each communication port are as follows.

- Serial communication with an external device connected to port 1 to port 33 (RS232C/RS485)
  "Modbus RTU Communication via RS232C/RS485" on page 6-2
- Ethernet communication with an external device connected by the Ethernet port 1 and 2
  "Modbus TCP Communication via Ethernet Communication" on page 6-21

Modbus RTU Communication via RS232C/RS485

The FC6A Series MICROSmart supports the Modbus RTU protocol and can be used as the Modbus RTU master and the Modbus RTU slave. When configured as a Modbus RTU master, the FC6A Series MICROSmart can monitor and change the data of Modbus RTU slave devices. When the FC6A Series MICROSmart is configured as a Modbus RTU slave, the device data of the FC6A Series MICROSmart can be monitored and changed from the Modbus RTU master device.

For the Modbus RTU master function and how to configure it, see "Modbus RTU Master Communication" on page 6-3. For the Modbus RTU slave function and how to configure it, see "Modbus RTU Slave Communication" on page 6-9.



*1 When communication quality is unstable, add terminating resistance matched to the characteristic impedance to both ends. Use resistance with a rating of 1/2 W or higher.

Modbus RTU Master Communication

When configured as a Modbus RTU master, the FC6A Series MICROSmart sends communication requests to Modbus RTU slaves to read/write data. Each communication request is sent to a Modbus RTU slave according to the configured request table.

Modbus RTU master communication settings and request tables for Modbus RTU slave stations can be programmed using the WindLDR **Function Area Settings**. Communication with slave stations are performed in synchronism with user program execution, and the communication data are processed at the END processing in the order of request numbers specified in the request table. When request execution devices are designated, requests are executed only when the corresponding request execution device is turned on. When request execution devices are not designated, all requests are executed continuously.

Modbus RTU Master Communication Specifications

Item	Description
Baud Rate (bps)	9,600, 19,200, 38,400, 57,600, 115,200
Data Bits	8 bits (fixed)
Stop bits	1, 2 bits
Parity	Even, Odd, None
Slave Number	1 to 247 (0: broadcast slave number) *1
Maximum Number of Slaves	RS-232C: 1 RS-485: 31
Maximum Cable Length	Modbus Communication via RS-232C FC6A-PC1: 5 m FC6A-SIF52: 15 m Modbus Communication via RS-485 FC6A-PC3: 200 m FC6A-SIF52: 1,200 m
Receive Timeout *2	10 to 2,550 ms (in increments of 10 ms)
Timeout between Characters	10 ms
Transmission Wait Time	0 to 5,000 ms (in increments of 1 ms)
Retry Cycles	0 to 10

*1 A communication request becomes the broadcast when slave number 0 is specified. The broadcast communication request is received by all Modbus RTU slaves. Modbus RTU slave does not reply to the broadcast communication. Broadcast can be used to write the same data to all Modbus RTU slaves.

*2 Specifies the period of time before receiving a response frame from a slave.

Modbus RTU Master Communication Start and Stop

When request execution devices are designated in the Modbus RTU master request table, internal relays or data register bits as many as the request quantity are allocated to execute Modbus RTU master communication. The internal relays or data register bits are allocated in the order of requests. For example, when internal relay M0 is designated as the request execution device, M0 is allocated to request No. 1, M1 to request No. 2, and so on. To execute a request, turn on the corresponding request execution device.

When communication is completed, the request execution device turns off automatically. When it is required to send requests continuously, keep the corresponding request execution device on using a SET or OUT instruction.

When request execution devices are not designated, all requests programmed in the request table are executed continuously.

Communication Completion and Communication Error

Modbus RTU communication finishes when a read or write process is completed successfully or when a communication error occurs. Communication error occurs when communication failure has repeated more than the designated retry cycles or when the master station does not receive response within the designated receive timeout period. When a communication error occurs, the request is canceled and the next request is processed. When the error status data register is designated, the communication status of each request can be confirmed.

Note: Modbus master processes a maximum of one Modbus request per scan.

Communication Error Data

When Error Status is configured in the Request Table from the **Function Area Settings**, the error data of each request can be confirmed.

Use a Single DR for All Communication Requests	Error Data of Each Communication Request
Unchecked	Error data, the remote host number (high-order byte) and error code (low-order byte), of each request in the entire request table can be confirmed. Data registers as many as the quantity of requests are reserved for storing error data. When an error occurs for a request, error data is stored to the corresponding data register.
Checked	A single data register is shared by all requests. When an error occurs for a request, error data is stored to the data register and the old error data is overwritten.

Bit Allocation	
Remote Host Number (high-order byte)	1 to 255
Error Code (low-order byte)	00h: Normal completion 01h: Function code error (unsupported function code) 02h: Access destination error (address out of range, address+device quantity out of range) 03h: Device quantity error, 1-bit write data error (specified device quantity of 1-bit write is unsupported) 12h: Frame length error (frame length of transmitted request exceeds range) 13h: BCC error (BCC does not match) 14h: Slave number error (received slave number is invalid) 16h: Timeout error (timeout occurs)

Communication Error Data of Each Request

Error data of each request in the entire request table can be confirmed. To confirm error data of each request, select to use Error Status in the Request Table from the **Function Area Settings** and enter the data register number.

When Use a single DR for all communication requests is not selected, starting with the data register number, data registers as many as the quantity of requests are reserved for storing error data. When an error occurs for a request, an error code is stored to a corresponding data register.

When Use a single DR for all communication requests is selected, the same data register is shared by all requests. When an error occurs for a request, an error code is stored to the data register and the old value is overwritten.

Number of Requests in Modbus RTU Master

The number of requests that can be programmed in a request table is shown in the table below:

Port	Port 1 to Port 33
No. of Requests	255

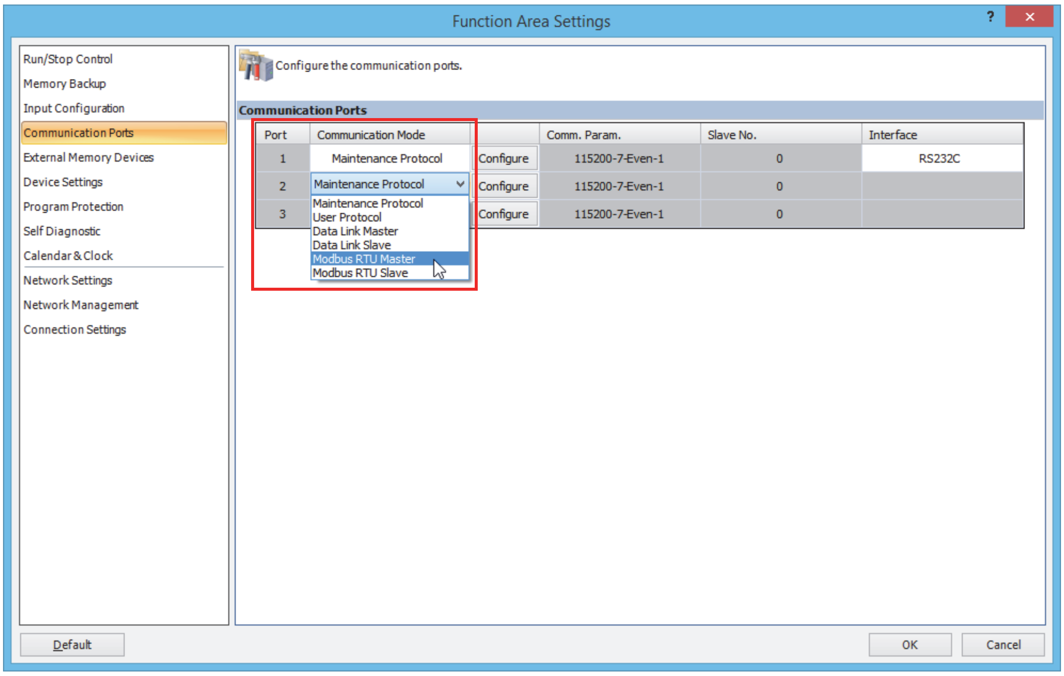
Note: 8 bytes of the user program area are needed per each request.

Programming Modbus RTU Master Using WindLDR

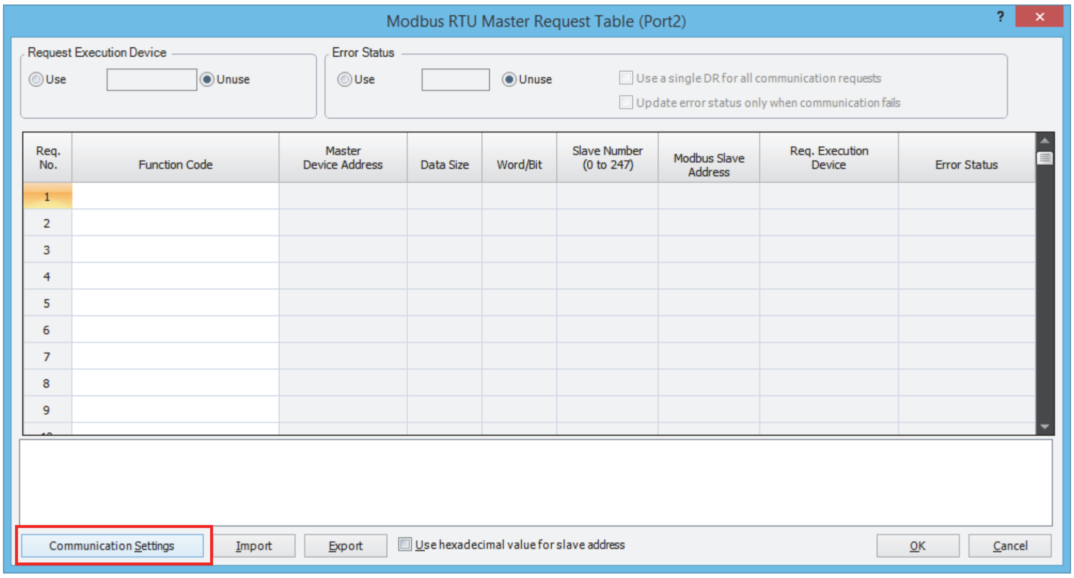
Modbus master communication is programmed for Modbus RTU mode using WindLDR. Since these settings relate to the user program, the user program must be downloaded to the FC6A Series MICROSmart after changing any of these settings.

1. From the WindLDR menu bar, select **Configuration** > **Communication Ports**.
The **Function Area Settings** dialog box for Communication Ports appears.

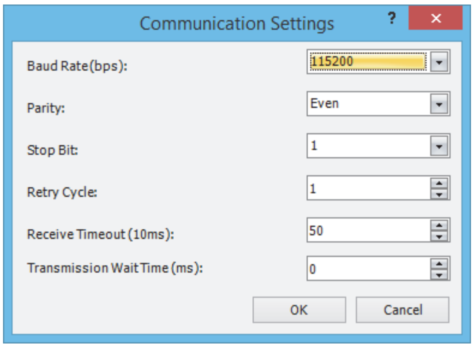
2. Click **Communication Mode** for the port to use and select **Modbus RTU Master**.



3. Click the **Configure** button for Port 2. The **Modbus RTU Master Request Table** appears.



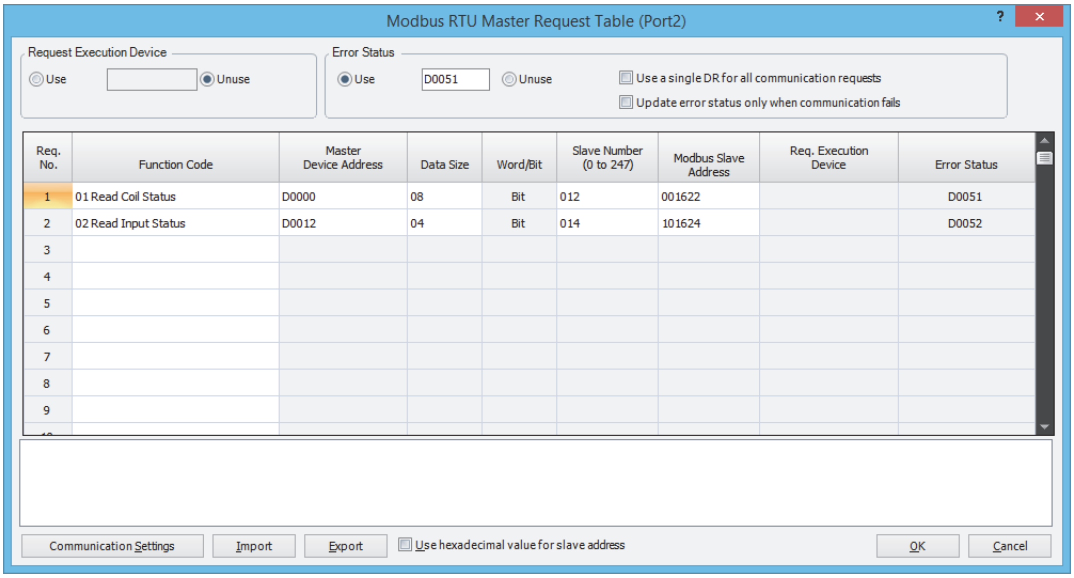
4. Click the **Communication Settings** button. The **Communication Settings** dialog box appears. Change settings, if required.



Baud Rate (bps)	9600, 19200, 38400, 57600, 115200
Parity	Even, Odd, None
Stop Bits	1 or 2
Retry Cycle	0 to 10
Receive Timeout	1 to 255 (×10 ms)
Transmission Wait Time	0 to 5000 (ms)

5. Click the **OK** button to return to the Modbus RTU Master Request Table. Designate requests under the **Function Code**. A maximum of 255 requests can be entered in one request table.

Choose to use **Req. Execution Device** and **Error Status** data registers if necessary. When using **Req. Execution Device** and **Error Status** data registers, enter the first number of the devices.



Notes for Editing the Request Table

Request execution devices and error status data registers are allocated in the order of request numbers. When deleting a request or changing the order of requests, the relationship of the request to the request execution devices and error status data register is changed. If the internal relay or data register is used in the user program, the device addresses must be changed accordingly. After completing the changes, download the user program again.

6. When editing of the Master Request Table is complete, click the **OK** button to save changes.

7. Download the user program to the FC6A Series MICROSmart.
Now, programming for the Modbus master is complete. Details about parameters and valid values are as follows.

Function Code

The Modbus RTU of the FC6A Series MICROSmart supports eight function codes as listed in the table below. Supported function codes and valid slave addresses vary with each Modbus slave device to communicate with. Configure the function codes according to the specifications of the Modbus slave devices.

Function Code	Data Size	Slave Address	FC6A Series MICROSmart as Modbus Slave
01 Read Coil Status	1 to 2,000 bits	000001 - 065535	Reads bit device statuses of Q (output), R (shift register), or M (internal relay).
02 Read Input Status	1 to 2,000 bits	100001 - 165535	Reads bit device statuses of I (input), T (timer contact), or C (counter contact).
03 Read Holding Registers	1 to 125 words	400001 - 465535	Reads word device data of D (data register), T (timer preset value), or C (counter preset value).
04 Read Input Registers	1 to 125 words	300001 - 365535	Reads word device data of T (timer current value) or C (counter current value).
05 Force Single Coil	1 bit	000001 - 065535	Changes a bit device status of Q (output), R (shift register), or M (internal relay).
06 Preset Single Register	1 word	400001 - 465535	Changes word device data of D (data register).
15 Force Multiple Coils	1 to 1,968 bits	000001 - 065535	Changes multiple bit device statuses of Q (output), R (shift register), or M (internal relay).
16 Preset Multiple Registers	1 to 123 words	400001 - 465535	Changes multiple word device data of D (data register).

Master Device Address

When function code 01, 02, 03, or 04 is selected to read data from Modbus slaves, designate the first data register or internal relay number to store the data received from the Modbus slave. When function code 05, 06, 15, or 16 is selected to write data to Modbus slaves, designate the first data register or internal relay number to store the data to write to the Modbus slave. Data registers and internal relays can be designated as the master device address.

Data Size and Word/Bit

Designate the quantity of data to read or write. The valid data size depends on the function code. When function code 01, 02, 05, or 15 is selected, designate the data size in bits. When function code 03, 04, 06, or 16 is selected, designate the data size in words. For valid data sizes, see the table above.

Slave No.

Designate slave numbers 0 through 247. The same slave number can be designated repeatedly for different request numbers which can be 1 through 255. In the Modbus communication, slave number 0 is used for a broadcast slave number. The broadcast can be used to write the same data to all Modbus slaves.

Slave Address

Designate data memory addresses of Modbus slaves. The valid slave address range depends on the function code. For valid slave addresses, see the table above. The allocations of memory addresses vary with each Modbus slave device. Refer to manuals for each Modbus slave device.

Request Execution Device

To use request execution devices, click the radio button for "Use" and designate the first internal relay in the Modbus RTU Master Request Table. Devices used for executing requests are automatically listed in the table. To execute a request, turn on the corresponding request execution device.

Data registers can also be designated as the Request Execution Device. When the first data register is designated as the Request Execution Device, data register bits as many as the number of requests are allocated from the least significant bit of the first data register. Data register bits assigned as the execution relays are automatically listed in the Request Table.

When request execution devices are not designated, all requests programmed in the Request Table are executed continuously.

Error Status Data Register

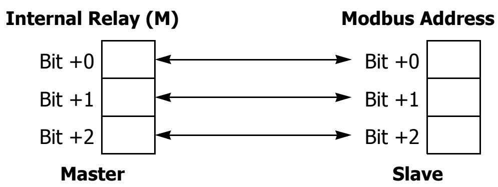
To use error status data registers, click the radio button for "Use" and designate the first data register in the Modbus RTU Master Request Table. Data registers used for storing error statuses are automatically listed in the table. When Use a single DR for all communication requests is selected, the first data register is shared by all requests.

Processing Requests

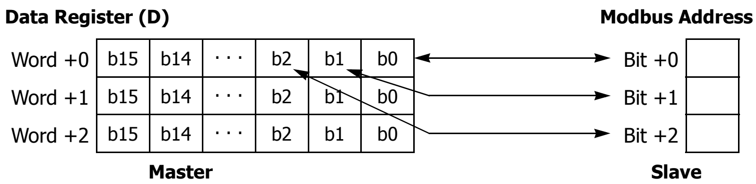
The data for Modbus communication are processed between the master and slaves as shown below.

Bit Data at Slaves (Function Codes 01, 02, 05, and 15)

- **Master Device Address: Internal Relay**

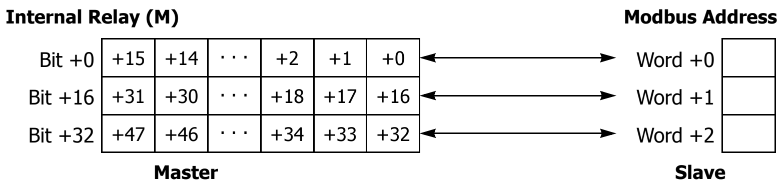


- **Master Device Address: Data Register**

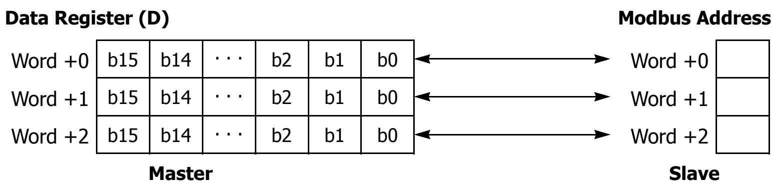


Word Data at Slaves (Function Codes 03, 04, 06, and 16)

- **Master Device Address: Internal Relay**



- **Master Device Address: Data Register**



Modbus RTU Slave Communication

Modbus slave communication can be configured by selecting Modbus RTU Slave for Port 1, Port 2 and Port 3 in the WindLDR **Function Area Settings**. When a Modbus RTU slave receives a request from the Modbus RTU master, the Modbus RTU slave reads or writes data according to the request. The request is processed at the END processing of the user program.

Modbus RTU slaves do not reply to the Modbus RTU master for the broadcast requests.

Modbus RTU Slave Communication Specifications

Item	Description	
Baud Rate (bps)	9,600, 19,200, 38,400, 57,600, 115,200	
Data Bits	8 bits (fixed)	
Stop bits	1, 2 bits	
Parity	Odd, even, none	
Slave Number	Constant	1 to 247
	Data register	Set the special data register values between 1 and 247 Port 1: D8100 Port 2: D8102 Port 3: D8103 Port 4 to 9*2: D8040 to D8045 Port 10 to 33*2*3: D8735 to D8758
Timeout between Characters*1	1.5 characters minimum*4	
Timeout between Frames *1	3.5 characters minimum*5	

*1 When timeout occurs, the FC6A Series MICROSmart discards the received data and waits for the first frame of the next valid communication.
*2 When using a communication module.
*3 Plus CPU module only.
*4 For communication at 19,200 bps or higher, the timeout between characters needs to be a minimum of 0.75 ms.
*5 For communication at 19,200 bps or higher, the timeout between frames needs to be a minimum of 1.75 ms.

Map of Slave Addresses for Modbus RTU Slaves

Modbus Device Name	Modbus Address Map (Decimal) *1	Communication Frame Address*2	FC6A Series MICROSmart Device*3	Applicable Function Code
Coil (000000 and above)	000001 - 000516	0000 - 0203	Q0 - Q643	1, 5, 15
	000701 - 000956	02BC - 03BB	R000 - R255	
	001001 - 003048	03E8 - 0BE7	M0000 - M2557	
	003049 - 007400	0BE8 - 1CE7	M2560 - M7997	
	009001 - 009256	2328 - 2427	M8000 - M8317	
	009257 - 010600	2428 - 2967	M8320 - M9997	
	011001 - 017000	2AF8 - 4267	M10000 - M17497	
	017001 - 020000	4268 - 4E1F	M17500 - M21247	
	030001 - 030480	7530 - 770F	Q1000 - Q1597	
	030481 - 030960	7710 - 78EF	Q2000 - Q2597	
	030961 - 031440	78F0 - 7ACF	Q3000 - Q3597	
	031441 - 031920	7AD0 - 7CAF	Q4000 - Q4597	
	031921 - 032400	7CB0 - 7E8F	Q5000 - Q5597	
	032401 - 032880	7E90 - 806F	Q6000 - Q6597	
	032881 - 033360	8070 - 824F	Q7000 - Q7597	
	033361 - 033840	8250 - 842F	Q8000 - Q8597	
	033841 - 034320	8430 - 860F	Q9000 - Q9597	
	034321 - 034800	8610 - 87EF	Q10000 - Q10597	

Modbus Device Name	Modbus Address Map (Decimal) *1	Communication Frame Address*2	FC6A Series MICROSmart Device*3	Applicable Function Code
Input Relay (100000 and above)	100001 - 100516	0000 - 0203	I0 - I643	2
	101001 - 101256	03E8 - 04E7	T000 - T255 (timer contact)	
	101501 - 101756	05DC - 06DB	C000 - C255 (counter contact)	
	102001 - 102768	07D0 - 0ACF	T256 - T1023 (timer contact)	
	102769 - 103744	0AD0 - 0E9F	T1024 - T1999 (timer contact)	
	104001 - 104256	0FA0 - 109F	C256 - C511 (counter contact)	
	130001 - 130480	7530 - 770F	I1000 - I1597	
	130481 - 130960	7710 - 78EF	I2000 - I2597	
	130961 - 131440	78F0 - 7ACF	I3000 - I3597	
	131441 - 131920	7AD0 - 7CAF	I4000 - I4597	
	131921 - 132400	7CB0 - 7E8F	I5000 - I5597	
	132401 - 132880	7E90 - 806F	I6000 - I6597	
	132881 - 133360	8070 - 824F	I7000 - I7597	
	133361 - 133840	8250 - 842F	I8000 - I8597	
	133841 - 134320	8430 - 860F	I9000 - I9597	
	134321 - 134800	8610 - 87EF	I10000 - I10597	
Input Register (300000 and above)	300001 - 300256	0000 - 00FF	T000 - T255 (timer current value)	4
	300501 - 300756	01F4 - 02F3	C000 - C255 (counter current value)	
	302001 - 302768	07D0 - 0ACF	T256 - T1023 (timer current value)	
	302769 - 303744	0AD0 - 0E9F	T1024 - T1999 (timer current value)	
	304001 - 304256	0FA0 - 109F	C256 - C511 (counter current value)	
Holding Register (400000 and above)	400001 - 408000	0000 - 1F3F	D0000 - D7999	3,6,16
	408001 - 408500	1F40 - 2133	D8000 - D8499	
	408501 - 408900	2134 - 22C3	D8500 - D8899	
	409001 - 409256	2328 - 2427	T000 - T255 (timer preset value)	3
	409501 - 409756	251C - 261B	C000 - C255 (counter preset value)	
	410001 - 456000	2710 - DABF	D10000 - D55999	3,6,16
	456001 - 462000	DAC0 - F22F	D56000 - D61999	
	462001 - 462768	F230 - F52F	T256 - T1023 (timer preset value)	3
	462769 - 463744	F530 - F8FF	T1024 - T1999 (timer preset value)	
	464001 - 464256	FA00 - FAFF	C256 - C511 (counter preset value)	

*1 Addresses generally used for Modbus communication. "Calculating Modbus Addresses for FC6A Series MICROSmart Devices" on page 6-11 shows the method to calculate slave addresses from FC6A Series MICROSmart devices.

*2 These 4-digit addresses are used in the communication frame. To calculate the address used in communication frame, extract lower 5 digits of the Modbus address, subtract 1 from the value, and convert the result into hexadecimal.

*3 Access within the device range for the FC6A Series MICROSmart type used.

Calculating Modbus Addresses for FC6A Series MICROSmart Devices

FC6A Series MICROSmart Device		Calculating Modbus Address	Calculation Example
I, Q, M	M XXXX X (2): Octal (1): Decimal	((1) − (4)) × 8 + (2) + (5) Minimum address Offset	Example: M325 (32 − 0) × 8 + 5 + 1001 = 1262 Modbus address: 1262 1262 − 1 = 1261 = 04ED Slave addresses in communication: 04ED
R, T, C, D	D XXXXX (3): Decimal	((3) − (4)) + (5) Minimum address Offset	Example: D756 (756 − 0) + 400001 = 400757 Modbus address: 400757 Extract lower 5 digits → 757 757 − 1 = 756 = 02F4 Slave addresses in communication: 02F4

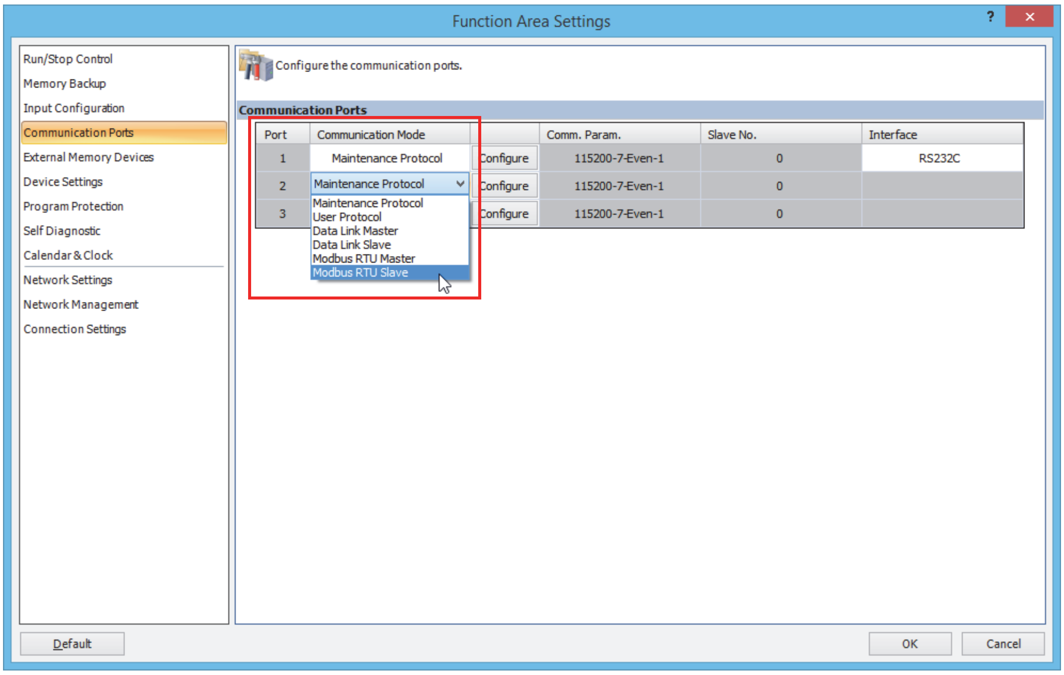
Modbus Device Name	FC6A Series MICROSmart Device	Minimum Address (4)	Offset (5)
Coil	Q0 - Q643	0	1
	R000 - R255	0	701
	M0000 - M2557	0	1001
	M2560 - M7997	256	3049
	M8000 - M8317	800	9001
	M8320 - M9997	832	9257
	M10000 - M17497	1000	11001
	M17500 - M21247	1750	17001
	Q1000 - Q1597	100	30001
	Q2000 - Q2597	200	30481
	Q3000 - Q3597	300	30961
	Q4000 - Q4597	400	31441
	Q5000 - Q5597	500	31921
	Q6000 - Q6597	600	32401
	Q7000 - Q7597	700	32881
	Q8000 - Q8597	800	33361
	Q9000 - Q9597	900	33841
	Q10000 - Q10597	1000	34321
Input Relay	I0 - I643	0	100001
	T000 - T255 (timer contact)	0	101001
	C000 - C255 (counter contact)	0	101501
	T256 - T1023 (timer contact)	256	102001
	T1024 - T1999 (timer contact)	1024	102769
	C256 - C511 (counter contact)	256	104001
	I1000 - I1597	100	130001
	I2000 - I2597	200	130481
	I3000 - I3597	300	130961
	I4000 - I4597	400	131441
	I5000 - I5597	500	131921
	I6000 - I6597	600	132401
	I7000 - I7597	700	132881
	I8000 - I8597	800	133361
	I9000 - I9597	900	133841
	I10000 - I10597	1000	134321
Input Register	T000 - T255 (timer current value)	0	300001
	C000 - C255 (counter current value)	0	300501
	T256 - T1023 (timer current value)	256	302001
	T1024 - T1999 (timer current value)	1024	302769
	C256 - C511 (counter current value)	256	304001

Modbus Device Name	FC6A Series MICROSmart Device	Minimum Address (4)	Offset (5)
Holding Register	D0000 - D7999	0	400001
	D8000 - D8499	8000	408001
	D8500 - D8899	8500	408501
	T000 - T255 (timer preset value)	0	409001
	C000 - C255 (counter preset value)	0	409501
	D10000 - D55999	10000	410001
	D56000 - D61999	56000	456001
	T256 - T1023 (timer preset value)	256	462001
	T1024 - T1999 (timer preset value)	256	462769
	C256 - C511 (counter preset value)	256	464001

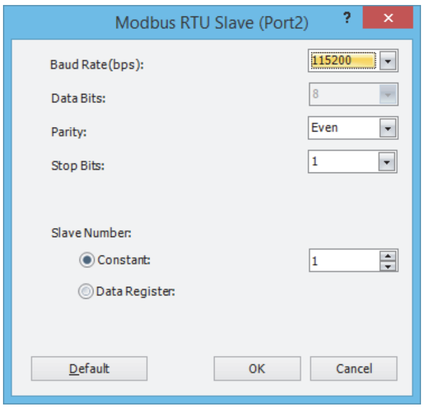
Programming Modbus Slave Using WindLDR

Modbus slave communication is programmed for Modbus RTU mode using WindLDR. Since these settings relate to the user program, the user program must be downloaded to the FC6A Series MICROSmart after changing any of these settings.

1. From the WindLDR menu bar, select **Configuration** > **Communication Ports**.
   The **Function Area Settings** dialog box for Communication Ports appears.
2. In the Communication Mode pull-down list for Port, select **Modbus RTU Slave**.



3. Click the **Configure** button. The **Modbus RTU Slave** dialog box appears. Change settings, if required.



Baud Rate (bps)	9600 19200 38400 57600 115200	
Data Bits	8	
Parity	Even, Odd, None	
Stop Bits	1 or 2	
Slave Number	Constant	1 to 247
	Data register	The value in D8100 (port 1), D8102 (port 2), D8103 (port 3), D8040 to D8045 (port 4 to 9*1), or D8735 to D8758 (port 10 to 33*1*2) is used.

*1 When using a communication module.
*2 Plus CPU module only.

4. Click the **OK** button to save changes.
5. Download the user program to the FC6A Series MICROSmart.
   Now, programming for the Modbus slave is complete. Details about parameters and valid values are as follows.

Communication Protocol

This section describes the communication frame format used for Modbus RTU communication.

Modbus RTU Mode Communication Format

Request from Modbus Master

Idle 3.5 characters	Slave No.	Function Code	Data	CRC	Idle 3.5 characters
	1 byte	1 byte		2 bytes	

ACK Reply from Modbus RTU Slave

Idle 3.5 characters	Slave No.	Function Code	Data	CRC	Idle 3.5 characters
	1 byte	1 byte		2 bytes	

NAK Reply from Modbus RTU Slave

Idle 3.5 characters	Slave No.	Function Code + 80H	Error Code	CRC	Idle 3.5 characters
	1 byte	1 byte	1 byte	2 bytes	

Note: Idle means no data flowing on the communication line.

Communication Frame Format

Modbus RTU mode requires a minimum of 3.5-character-long idle time between frames to determine the beginning of a frame. The FC6A Series MICROSmart Modbus master sends requests at idle intervals of 5 ms, which can be changed in the **Function Area Settings** dialog box.

Slave No.

The FC6A Series MICROSmart can be assigned slave numbers 1 through 247. In the 1:1 communication using RS232C, the same slave number must be set in the master and the FC6A Series MICROSmart.

Slave No. 0 is reserved for broadcast slave number and is used to write the same data to all Modbus RTU slaves. In this case, the Modbus RTU slaves do not send a reply to the master.

Modbus RTU Communication NG Reply Error Code

One of the following error codes is stored in NAK reply.

01h: Function code error (unsupported function code)

02h: Access destination error (address out of range, address+device quantity out of range)

03h: Device quantity error, 1-bit write data error

CRC

Modbus RTU mode uses CRC check codes.

- **Modbus RTU Mode — Calculating the CRC-16 (cyclic redundancy checksum)**

Calculate the BCC using CRC-16 for the range from the slave number to the byte immediately before the BCC. The generation polynomial is: $X^{16} + X^{15} + X^2 + 1$.

1. Take the exclusive OR (XOR) of FFFFh and the first 1-byte data at the slave number.
2. Shift the result by 1 bit to the right. When a carry occurs, take the exclusive OR (XOR) of A001h, then go to step **3**.
   If not, directly go to step **3**.
3. Repeat step **2**, shifting 8 times.
4. Take the exclusive OR (XOR) of the result and the next 1-byte data.
5. Repeat step **2** through step **4** up to the byte immediately before the BCC.
6. Swap the higher and lower bytes of the result of step **5**, and store the resultant CRC-16 to the BCC (CRC) position. (Example: 1234h → 34h, 12h)

Communication Format

This section describes the communication format for each function code from the slave number up to immediately before the check code.

Function Code 01 (Read Coil Status) and Function Code 02 (Read Input Status)

Function code 01 reads bit device statuses of Q (output), R (shift register), or M (internal relay). One through 2,000 consecutive bits can be read out.

Function code 02 reads bit device statuses of I (input), T (timer contact), or C (counter contact). One through 2,000 consecutive bits can be read out.

Communication Frame

Request from Modbus RTU Master

Slave No.	Function Code	Address	No. of Bits
xxh	01h / 02h	xxxxh	xxxxh

ACK Reply from Modbus RTU Slave

Slave No.	Function Code	Quantity of Data	First 8 Bits	Second 8 Bits		Last 8 Bits
xxh	01h / 02h	xxh	xxh	xxh		xxh

NAK Reply from Modbus RTU Slave

Slave No.	Function Code	Error Code
xxh	81h / 82h	xxh

Communication Example

Purpose	Read 15 bits starting at output Q10. Q10 → (1 – 0) × 8 + 0 + 1 = 9 Modbus address: 9 9 – 1 = 8 = 8h Communication frame address: 0008h
Condition	Slave No. 8 Q10 through Q27 binary data: 1234h

- **Modbus RTU Mode**

Request from Modbus RTU Master	08 01 0008 0010 (CRC)
ACK Reply from Modbus RTU Slave	08 01 02 34 12 (CRC)
NAK Reply from Modbus RTU Slave	08 81 xx (CRC)

Function Code 03 (Read Holding Registers) and Function Code 04 (Read Input Registers)

Function code 03 reads word device data of D (data register), T (timer preset value), or C (counter preset value). 1 through 125 consecutive words can be read out.

Function code 04 reads word device data of T (timer current value) or C (counter current value). 1 through 125 consecutive words can be read out.

Communication Frame

Request from Modbus RTU Master

Slave No.	Function Code	Address	No. of Words
xxh	03h / 04h	xxxxh	xxxxh

ACK Reply from Modbus RTU Slave

Slave No.	Function Code	Quantity of Data	First High Byte	First Low Byte		Last Low Byte
xxh	03h / 04h	xxh	xxh	xxh		xxh

NAK Reply from Modbus RTU Slave

Slave No.	Function Code	Error Code
xxh	83h / 84h	xxh

Communication Example

Purpose	Read 2 words starting at data register D1710. D1710 → (1710 – 0) + 400001 = 401711 Modbus address: 401711 Extract lower 5 digits → 1711 1711 – 1 = 1710 = 6AEh Communication frame address: 06AEh
Condition	Slave No. 8 D1710 data: 1234h D1711 data: 5678h

- **Modbus RTU Mode**

Request from Modbus RTU Master	08 03 06AE 0002 (CRC)
ACK Reply from Modbus RTU Slave	08 03 04 12 34 56 78 (CRC)
NAK Reply from Modbus RTU Slave	08 83 xx (CRC)

Function Code 05 (Force Single Coil)

Function code 05 changes a bit device status of Q (output), R (shift register), or M (internal relay).

Communication Frame

Request from Modbus RTU Master

Slave No.	Function Code	Address	OFF: 0000h ON: FF00h
xxh	05h	xxxxh	xxxxh

ACK Reply from Modbus RTU Slave

Slave No.	Function Code	Address	OFF: 0000h ON: FF00h
xxh	05h	xxxxh	xxxxh

NAK Reply from Modbus RTU Slave

Slave No.	Function Code	Error Code
xxh	85h	xxh

Communication Example

Purpose	Force internal relay M1320 on. M1320 → (132 – 0) × 8 + 0 + 1001 = 2057 Modbus address: 2057 2057 – 1 = 2056 = 808h Communication frame address: 0808h
Condition	Slave No. 8

- **Modbus RTU Mode**

Request from Modbus RTU Master	08 05 0808 FF00 (CRC)
ACK Reply from Modbus RTU Slave	08 05 0808 FF00 (CRC)
NAK Reply from Modbus RTU Slave	08 85 xx (CRC)

Function Code 06 (Preset Single Register)

Function code 06 changes word device data of D (data register).

Communication Frame

Request from Modbus RTU Master

Slave No.	Function Code	Address	New Data
xxh	06h	xxxxh	xxxxh

ACK Reply from Modbus RTU Slave

Slave No.	Function Code	Address	Acknowledge Data
xxh	06h	xxxxh	xxxxh

NAK Reply from Modbus RTU Slave

Slave No.	Function Code	Error Code
xxh	86h	xxh

Communication Example

Purpose	Write 8000 to data register D1708. D1708 → (1708 – 0) + 400001 = 401709 Modbus address: 401709 Extract lower 5 digits → 1709 1709 – 1 = 1708 = 6ACh Communication frame address: 06ACh
Condition	Slave No. 8

- **Modbus RTU Mode**

Request from Modbus RTU Master	08 06 06AC 1F40 (CRC)
ACK Reply from Modbus RTU Slave	08 06 06AC 1F40 (CRC)
NAK Reply from Modbus RTU Slave	08 86 xx (CRC)

Function Code 15 (Force Multiple Coils)

Function code 15 changes bit device statuses of Q (output), R (shift register), or M (internal relay). One through 1,968 consecutive bits can be changed.

Communication Frame

Request from Modbus RTU Master

Slave No.	Function Code	Address	No. of Bits	Quantity of Data	First 8 Bits	Second 8 Bits		Last 8 Bits
xxh	0Fh	xxxxh	xxxxh	xxh	xxh	xxh		xxh

ACK Reply from Modbus RTU Slave

Slave No.	Function Code	Address	No. of Bits
xxh	0Fh	xxxxh	xxxxh

NAK Reply from Modbus RTU Slave

Slave No.	Function Code	Error Code
xxh	8Fh	xxh

Communication Example

Purpose	Write the following bit statuses to internal relays M605 through M624. M605 (ON) M606 (ON) M607 (OFF) M610 (ON) M611 (OFF) M612 (ON) M613 (ON) M614 (OFF) M615 (OFF) M616 (ON) M617 (OFF) M620 (OFF) M621 (OFF) M622 (OFF) M623 (OFF) M624 (OFF) M605 (LSB) through M614 (MSB) binary data: 6B M615 (LSB) through M624 (MSB) binary data: 02 M605 $\rightarrow$ (60 – 0) $\times$ 8 + 5 + 1001 = 1486 Modbus address: 1486 1486 – 1 = 1485 = 5CDh Communication frame address: 05CDh
Condition	Slave No. 8

- **Modbus RTU Mode**

Request from Modbus RTU Master	08 0F 05CD 0010 02 6B 02 (CRC)
ACK Reply from Modbus RTU Slave	08 0F 05CD 0010 (CRC)
NAK Reply from Modbus RTU Slave	08 8F xx (CRC)

Function Code 16 (Preset Multiple Registers)

Function code 16 changes word device data of D (data register). One through 123 consecutive words can be changed.

Communication Frame

Request from Modbus RTU Master

Slave No.	Function Code	Address	No. of Words	Quantity of Data	First High Byte	First Low Byte		Last Low Byte
xxh	10h	xxxxh	xxxxh	xxh	xxh	xxh		xxh

ACK Reply from Modbus RTU Slave

Slave No.	Function Code	Address	No. of Words
xxh	10h	xxxxh	xxxxh

NAK Reply from Modbus RTU Slave

Slave No.	Function Code	Error Code
xxh	90h	xxh

Communication Example

Purpose	Write the following data to four data registers D1708 through D1711. D1708 (1234h) D1709 (5678h) D1710 (ABCDh) D1711 (EF01h) D1708 → (1708 – 0) + 400001 = 401709 Modbus address: 401709 Extract lower 5 digits → 1709 1709 – 1 = 1708 = 6ACh Communication frame address: 06ACh
Condition	Slave No. 8

- **Modbus RTU Mode**

Request from Modbus RTU Master	08 10 06AC 0004 08 12 34 56 78 AB CD EF 01 (CRC)
ACK Reply from Modbus RTU Slave	08 10 06AC 0004 (CRC)
NAK Reply from Modbus RTU Slave	08 90 xx (CRC)

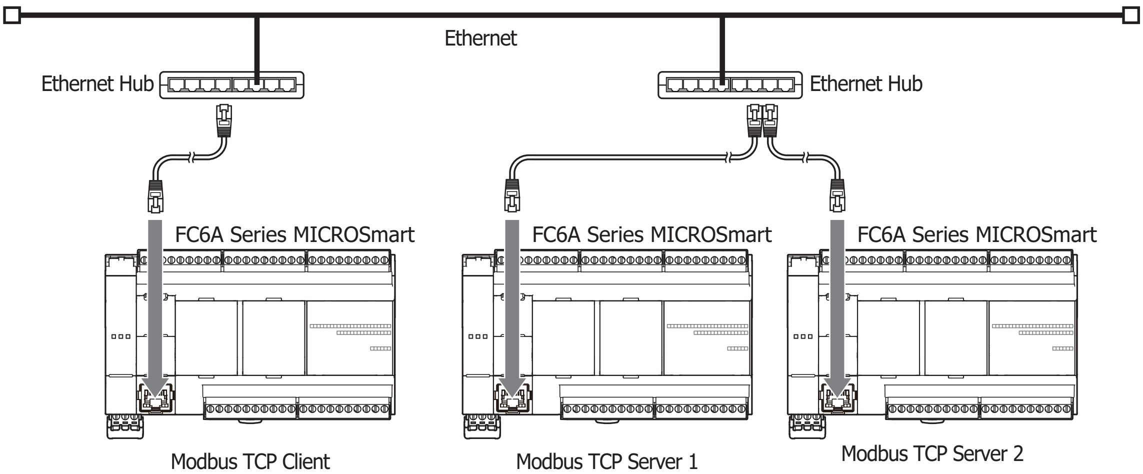
Modbus TCP Communication via Ethernet Communication

The FC6A Series MICROSmart supports Modbus TCP clients and Modbus TCP servers. Connect the Ethernet port 1 and Ethernet port 2 (Plus CPU module only) on the FC6A Series MICROSmart to enable the FC6A Series MICROSmart to communicate with Modbus TCP compliant devices.

When configured as a Modbus TCP client, the FC6A Series MICROSmart can monitor and change the data memory of the network devices supporting Modbus TCP server. A maximum of eight connections for the All-in-One CPU module/CAN J1939 All-in-One CPU module and 16 connections for the Plus CPU module can be allocated to Modbus TCP clients. Each connection can communicate with multiple Modbus TCP server devices (1 to 255).

When the FC6A Series MICROSmart is configured as a Modbus TCP server, the device data of the FC6A Series MICROSmart can be monitored and changed from Modbus TCP client devices. A maximum of eight connections for the All-in-One CPU module/CAN J1939 All-in-One CPU module and 16 connections for the Plus CPU module can be allocated to Modbus TCP servers.

For the Modbus TCP client function and how to configure it, see "Modbus TCP Client" on page 6-22. For the Modbus TCP server function and how to configure it, see "Modbus TCP Server" on page 6-27.



Modbus TCP Client

When configured as a Modbus TCP client, the FC6A Series MICROSmart sends communication requests to a Modbus TCP server to read/write data. Each communication request is sent to a Modbus Server according to the configured request table.

Modbus TCP client communication settings and request tables for Modbus TCP servers can be configured using the WindLDR **Function Area Settings**. The FC6A Series MICROSmart communicate with the Modbus TCP servers according to those settings.

Communication with Modbus TCP servers are performed in sync with the user program execution, and the communication data is processed at the END, in the order of request numbers specified in the request table. In Plus CPU module Modbus TCP client communication, the Plus CPU module can perform Modbus TCP communication for the first 128 requests without disconnecting the connection with the Modbus TCP server. For 129 and subsequent requests, the connection with the Modbus TCP server will be connected and disconnected each time Modbus TCP communication is performed.

Modbus TCP Client Specifications

Parameter	Modbus TCP Client
Slave Number	0-255
Maximum Number of Servers	8 (one server per one connection)
Receive Timeout *1	100 to 25,500 ms (in increments of 100 ms)
Transmission wait time	0 to 5,000 ms (0 ms)

*1 Specifies the period of time before receiving a response frame from a server.

Request Execution Device	Start/stop
Used	When request execution devices are designated in the Modbus TCP Client request table, internal relays or data register bits (as many as the request quantity) are allocated to execute Modbus TCP Client communication. For example, when internal relay M0 is designated as the request execution device, M0 is allocated to request No. 1, M1 to request No. 2, and so on. To execute a request, turn on the corresponding request execution device. When communication is completed, the request execution device turns off automatically. When it is required to send requests continuously, keep the corresponding request execution device on using a SET or OUT instruction. When request execution devices are not designated, all requests programmed in the request table are executed continuously.
Unused	The internal relays or data register bits are allocated in the order of requests.
Auto ping linking	Whether or not to send a request from the Modbus TCP client can be controlled with online status on/off via auto ping linking. This eliminates unnecessary timeouts by not sending requests to remote hosts that are not part of the network. For details on auto ping linking, see "Auto Ping Function" on page 3-22.

The FC6A Series MICROSmart operation when combining the request execution setting is as follows.

Request Execution Setting		Auto Ping Linking	
		Do Not Use	Use
Request execution device	Do not use	Always send all requests.	Send the applicable request only when the online status is on, and when it is off, do not send the applicable request.
	Use	Send the applicable request only when the communication execution device is on, and when it is off, do not send the applicable request.	Send the applicable request only when the online status and the communication execution device are both on, and do not send the applicable request in all other cases.

Communication Completion and Communication Error

Modbus communication finishes when a read or write process is completed successfully or when a communication error occurs.

A communication error occurs when communication failure has repeated three times. When a communication error occurs, the request is canceled and the next request is processed. When the error status data register is designated, the communication status of each request can be confirmed.

Communication Error Data

When Error Status is configured in the Request Table from the **Function Area Settings**, the error data of each request can be confirmed.

Use a single DR for all communication requests	Error data of each communication request
Unchecked	Error data, the remote host number (high-order byte) and error code (low-order byte), of each request in the entire request table can be confirmed. Data registers as many as the quantity of requests are reserved for storing error data. When an error occurs for a request, error data is stored to the corresponding data register.
Checked	A single data register is shared by all requests. When an error occurs for a request, error data is stored to the data register and the old error data is overwritten.

Bit Allocation	
Remote Host Number (high-order byte)	1 to 255
Error Code (low-order byte)	00h: Normal completion 01h: Function code error (unsupported function code) 02h: Access destination error (address out of range, address+device quantity out of range) 03h: Device quantity error, 1-bit write data error (specified device quantity of 1-bit write is unsupported) 12h: Frame length error (frame length of transmitted request exceeds range) 13h: BCC error (BCC does not match) 14h: Slave number error (received slave number is invalid) 16h: Timeout error (timeout occurs)

Modbus TCP Communication Request Table

A maximum of 255 requests can be configured in the Modbus TCP Client Request Table.

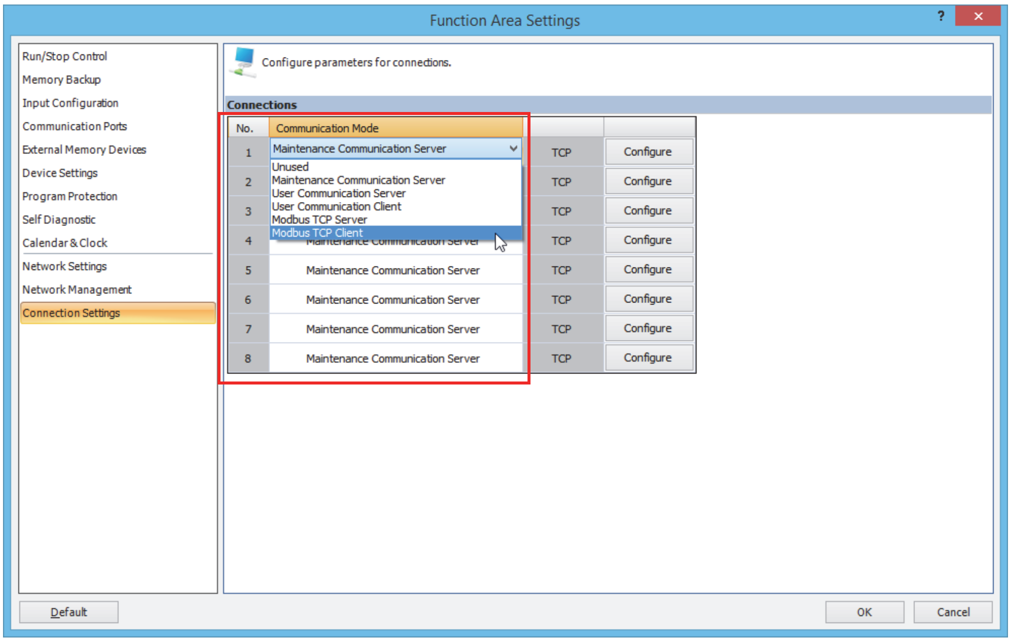
Notes:

- 10 bytes of the user program area are needed per each request.
- Request execution devices and error status data registers are allocated in the order of request numbers. When deleting a request or changing the order of requests, the relationship of the request to the request execution devices and error status data register is changed. If the allocated internal relays or data registers are used in the user program, those device addresses must be updated accordingly.

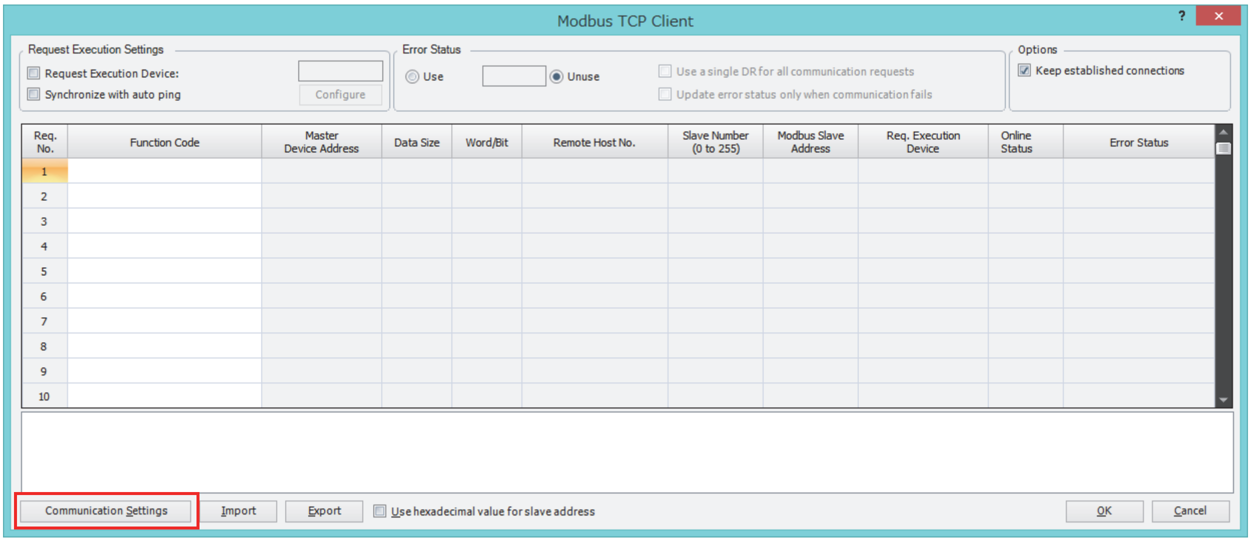
Programming WindLDR (Modbus TCP Client)

In order to use the Modbus TCP client, configure the Modbus TCP client in the **Function Area Settings** dialog box and then download the user program to the FC6A Series MICROSmart.

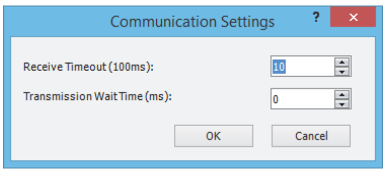
1. From the WindLDR menu bar, select **Configuration** > **Connection Settings**.
   The **Function Area Settings** dialog box appears.
2. Select **Modbus TCP Client** as the communication mode for the connection 1.



The **Modbus TCP Client** dialog box appears.



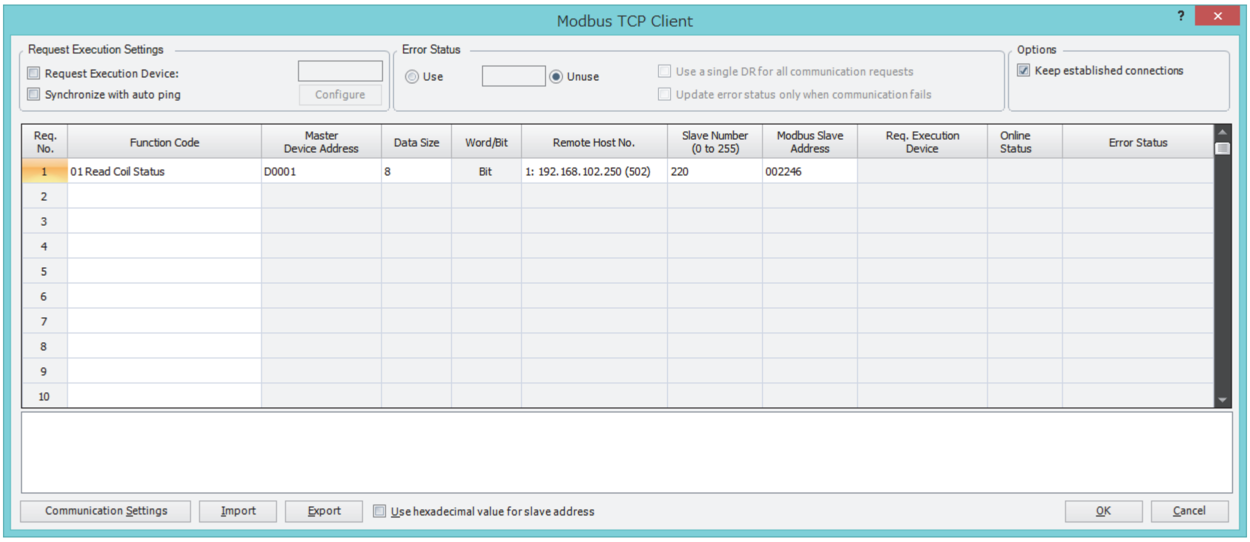
3. Click on the **Communication Settings** button.
   The **Communication Settings** dialog box appears. Configure the receive timeout and the transmission wait time.
   Click **OK**.



4. Configure the requests of the Modbus TCP Client.

A maximum of 255 requests can be entered in one request table. Specify the Modbus TCP servers with the remote host numbers. After all requests are configured, click **OK** button to close the dialog box.

Choose to use **Request Execution Devices** and **Error Status** data registers if necessary. When using **Request Execution Devices** and **Error Status** data registers, enter the first number of the devices.



5. Download the user program to the FC6A Series MICROSmart.

Programming for the Modbus TCP client is complete. Details about parameters and valid values are as follows.

Function Code

The Modbus TCP of the FC6A Series MICROSmart supports eight function codes as listed in the table below. Supported function codes and valid slave addresses vary with each Modbus server device to communicate with. Configure the function codes according to the specifications of the Modbus server devices.

Function Code	Data Size	Slave Address	FC6A Series MICROSmart as Modbus Slave
01 Read Coil Status	1 to 2,000 bits	000001 - 065535	Reads bit device statuses of Q (output), R (shift register), or M (internal relay).
02 Read Input Status	1 to 2,000 bits	100001 - 165535	Reads bit device statuses of I (input), T (timer contact), or C (counter contact).
03 Read Holding Registers	1 to 125 words	400001 - 465535	Reads word device data of D (data register), T (timer preset value), or C (counter preset value).
04 Read Input Registers	1 to 125 words	300001 - 365535	Reads word device data of T (timer current value) or C (counter current value).
05 Force Single Coil	1 bit	000001 - 065535	Changes a bit device status of Q (output), R (shift register), or M (internal relay).
06 Preset Single Register	1 word	400001 - 465535	Changes word device data of D (data register).
15 Force Multiple Coils	1 to 1,968 bits	000001 - 065535	Changes multiple bit device statuses of Q (output), R (shift register), or M (internal relay).
16 Preset Multiple Registers	1 to 123 words	400001 - 465535	Changes multiple word device data of D (data register).

Master Device Address

When function code 01, 02, 03, or 04 is selected to read data from Modbus servers, designate the first data register or internal relay number to store the data received from the Modbus server. When function code 05, 06, 15, or 16 is selected to write data to Modbus servers, designate the first data register or internal relay number to store the data to write to the Modbus server. Data registers and internal relays can be designated as the master device address.

Data Size and Word/Bit

Designate the quantity of data to read or write. The valid data size depends on the function code. When function code 01, 02, 05, or 15 is selected, designate the data size in bits. When function code 03, 04, 06, or 16 is selected, designate the data size in words. For valid data sizes, see "Function Code" on page 6-25.

Remote Host No.

Designate the remote host number configured in the Remote Host List dialog box. In the Remote Host List dialog box, IP address and port number are configured for each remote host. The default port number of Modbus TCP communication is 502. If the Modbus TCP server uses a different port number, configure that port number in the Remote Host List dialog box.

Slave No.

Designate slave numbers 0 through 255. The same slave number can be designated repeatedly for different request numbers which can be 1 through 255. If the Modbus TCP Server will not reference the slave number, set the slave number to any desired number.

Slave Address

Designate data memory addresses of Modbus servers. The valid slave address range depends on the function code. For valid slave addresses, see the table above. The allocations of memory addresses vary with each Modbus server device. Refer to manuals for each Modbus server device.

Request Execution Device

To use request execution devices, click the radio button for "Use" and designate the first internal relay or data register in the Modbus TCP Client Request Table. Internal relays or data register bits used for executing requests are automatically listed in the table. To execute a request, turn on the corresponding request execution device.

When request execution devices are not designated, all requests programmed in the Request Table are executed continuously.

Error Status Data Register

To use error status data registers, click the radio button for "Use" and designate the first data register in the Modbus TCP Client Request Table. Data registers used for storing error statuses are automatically listed in the table. When **Use a single DR for all communication requests** is selected, the first data register is shared by all requests.

Modbus TCP Server

When the FC6A Series MICROSmart is configured as the Modbus TCP server, Modbus TCP client devices can read/write data to the FC6A Series MICROSmart. When the FC6A Series MICROSmart receives a valid request from a Modbus TCP client device, the data is read or written according to the request received. The communication data received from Modbus TCP clients is processed at the END processing of the user program.

Modbus TCP Server Specifications

Parameter	Modbus TCP Server
Slave Number	Ignored
Response Time	1.5 ms
Number of Clients that can Access Simultaneously*1	All-in-One CPU module/CAN J1939 All-in-One CPU module: 8 (1 client per 1 connection) Plus CPU module: 16 (1 client per 1 connection)
Port Number	502 (can be changed between 0 and 65535)
Supported Function Code	01 Read Coil Status 02 Read Input Status 03 Read Holding Registers 04 Read Input Registers 05 Force Single Coil 06 Preset Single Register 15 Force Multiple oils 16 Preset Multiple Registers

*1 The number of connections when all eight connections for the All-in-One CPU module/CAN J1939 All-in-One CPU module and 16 connections for the Plus CPU module are set to Modbus TCP server.

Map of Slave Addresses for the Modbus TCP Server

Modbus TCP client can access the Modbus devices (Coil, Input Relay, Input Register, and Holding Register) of Modbus server to read or write the device data (I, Q, M, R, T, C, and D) of FC6A Series MICROSmart. Refer to the following table to configure the Modbus TCP clients.

Modbus Device Name	Slave Addresses	Slave Addresses in Communication*1	FC6A Series MICROSmart Device*2	Applicable Function Code
Coil (000000 and above)	000001 - 000516	0000 - 0203	Q0 - Q643	1,5,15
	000701 - 000956	02BC - 03BB	R000 - R255	
	001001 - 003048	03E8 - 0BE7	M0000 - M2557	
	003049 - 007400	0BE8 - 1CE7	M2560 - M7997	
	009001 - 009256	2328 - 2427	M8000 - M8317	
	009257 - 010600	2428 - 2967	M8320 - M9997	
	011001 - 017000	2AF8 - 4267	M10000 - M17497	
	017001 - 020000	4268 - 4E1F	M17500 - M21247	
	030001 - 030480	7530 - 770F	Q1000 - Q1597	
	030481 - 030960	7710 - 78EF	Q2000 - Q2597	
	030961 - 031440	78F0 - 7ACF	Q3000 - Q3597	
	031441 - 031920	7AD0 - 7CAF	Q4000 - Q4597	
	031921 - 032400	7CB0 - 7E8F	Q5000 - Q5597	
	032401 - 032880	7E90 - 806F	Q6000 - Q6597	
	032881 - 033360	8070 - 824F	Q7000 - Q7597	
	033361 - 033840	8250 - 842F	Q8000 - Q8597	
	033841 - 034320	8430 - 860F	Q9000 - Q9597	
	034321 - 034800	8610 - 87EF	Q10000 - Q10597	

Modbus Device Name	Slave Addresses	Slave Addresses in Communication*1	FC6A Series MICROSmart Device*2	Applicable Function Code
Input Relay (100000 and above)	100001 - 100516	0000 - 0203	I0 - I643	2
	101001 - 101256	03E8 - 04E7	T000 - T255 (timer contact)	
	101501 - 101756	05DC - 06DB	C000 - C255 (counter contact)	
	102001 - 102768	07D0 - 0ACF	T256 - T1023 (timer contact)	
	102769 - 103744	0AD0 - 0E9F	T1024 - T1999 (timer contact)	
	104001 - 104256	0FA0 - 109F	C256 - C511 (counter contact)	
	130001 - 130480	7530 - 770F	I1000 - I1597	
	130481 - 130960	7710 - 78EF	I2000 - I2597	
	130961 - 131440	78F0 - 7ACF	I3000 - I3597	
	131441 - 131920	7AD0 - 7CAF	I4000 - I4597	
	131921 - 132400	7CB0 - 7E8F	I5000 - I5597	
	132401 - 132880	7E90 - 806F	I6000 - I6597	
	132881 - 133360	8070 - 824F	I7000 - I7597	
	133361 - 133840	8250 - 842F	I8000 - I8597	
	133841 - 134320	8430 - 860F	I9000 - I9597	
	134321 - 134800	8610 - 87EF	I10000 - I10597	
Input Register (300000 and above)	300001 - 300256	0000 - 00FF	T000 - T255 (timer current value)	4
	300501 - 300756	01F4 - 02F3	C000 - C255 (counter current value)	
	302001 - 302768	07D0 - 0ACF	T256 - T1023 (timer current value)	
	302769 - 303744	0AD0 - 0E9F	T1024 - T1999 (timer current value)	
	304001 - 304256	0FA0 - 109F	C256 - C511 (counter current value)	
Holding Register (400000 and above)	400001 - 408000	0000 - 1F3F	D0000 - D7999	3,6,16
	408001 - 408500	1F40 - 2133	D8000 - D8499	
	408501 - 408900	2134 - 22C3	D8500 - D8899	
	409001 - 409256	2328 - 2427	T000 - T255 (timer preset value)	3
	409501 - 409756	251C - 261B	C000 - C255 (counter preset value)	
	410001 - 456000	2710 - DABF	D10000 - D55999	3,6,16
	456001 - 462000	DAC0 - F22F	D56000 - D61999	
	462001 - 462768	F230 - F52F	T256 - T1023 (timer preset value)	3
	462769 - 463744	F530 - F8FF	T1024 - T1999 (timer preset value)	
	464001 - 464256	FA00 - FAFF	C256 - C511 (counter preset value)	

*1 Slave addresses in communication are 4-digit values used in the address portion of the communication frame. Subtract 1 from the lower 5 digits of the slave address and store that value in hexadecimal. For details, see "Modbus RTU Slave Communication" on page 6-9.

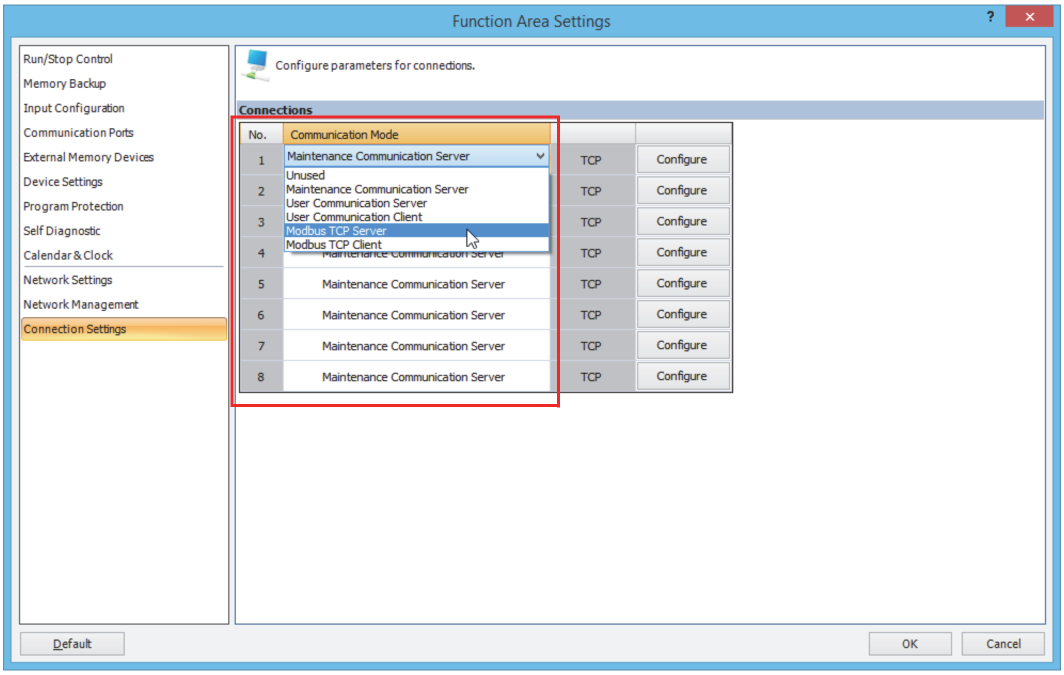
*2 Access within the device range for the FC6A Series MICROSmart type used.

Programming WindLDR (Modbus TCP Server)

In order to use Modbus TCP server, configure the Modbus TCP server in the **Function Area Settings** dialog box and then download the user program to the FC6A Series MICROSmart.

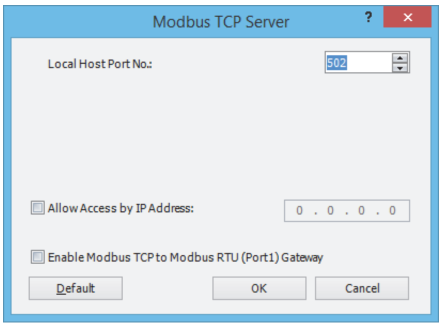
1. From the WindLDR menu bar, select **Configuration** > **Connection Settings**.
The **Function Area Settings** dialog box appears.

2. Select **Modbus TCP Server** as the communication mode for connection 1.



The **Modbus TCP Server** dialog box appears.

3. Configure the parameters and click on **OK** button.



Note: For details about each parameter, see the following page.

4. Download the user program to the FC6A Series MICROSmart.

Programming for the Modbus TCP server is complete.

Modbus TCP Server Communication Settings

Local Host Port No.

Configure the local host port number between 0 and 65,535. The same local host port number can be used with multiple connection numbers.

If the same port number is used with multiple connections, Modbus TCP clients (as many as the number of the connections) can communicate with FC6A Series MICROSmart through the same port number.

Allow Access by IP Address

You can set the IP address for a device to permit access. By setting the allowed IP address, only the device with the specified IP address can establish a connection with the FC6A Series MICROSmart and communicate with the Modbus TCP server. When the same local host port number is configured in multiple connections, all the allowed IP address settings are effective. If a connection in which the allowed IP address it not configured uses the same local host port number, arbitrary access is allowed through the port.

Example 1) If two connections use the same local port number and an allowed IP address is not configured for both connections, access from a total of two clients with any IP addresses is accepted.

Example 2) If two connections use the same local port number and 192.168.1.101 and 192.168.1.102 are configured as the allowed IP addresses, access from a total of two clients whose IP addresses are 192.168.1.101 and 192.168.1.102 is accepted.

Example 3) If connection 1 and 2 use the same local port number, an allowed IP address 192.168.1.101 is configured for connection 1, and the allowed IP address is not configured for connections, access from a total of two clients with any IP addresses is accepted.

Modbus TCP Communication Format

This section describes the communication format used for Modbus TCP client and server communication. Modbus TCP communication format starts with the Modbus TCP header followed by the RTU mode communication format without the idle 3.5 characters at both ends and CRC as shown below.

Modbus TCP Communication Format

Transaction ID	Protocol ID	Message Length (bytes)	Unit ID	Function Code	Data
2 bytes	2 bytes	2 bytes	1 byte	1 byte	N bytes

Modbus TCP Header (Transaction ID, Protocol ID, Message Length)

RTU Mode Communication Format

Idle	Slave No.	Function Code	Data	CRC	Idle
3.5 characters	1 byte	1 byte	N bytes	2 bytes	3.5 characters

Transaction ID

The Modbus TCP server (slave) returns the request ID sent from the client (master) without any change. When receiving the returned request ID, the client can confirm to which request the response was returned. When confirmation is not required, designate 0 as a transaction ID.

Protocol ID

Designate 0 to identify Modbus TCP protocol.

Message Length

Designate the length of the following message in bytes.

Unit ID

The ID for identifying the device. Store the slave number of the Modbus TCP server. The FC6A Series MICROSmart Modbus TCP server accepts and processes requests when the unit ID of the received request is not 0. When the unit ID is 0, the received request is processed as broadcast communication and no response is returned to the Modbus TCP client.

Function Code

Designate a function code, such as 01 (read coil status) and 02 (read input status).

Data

Designate required data for each function.

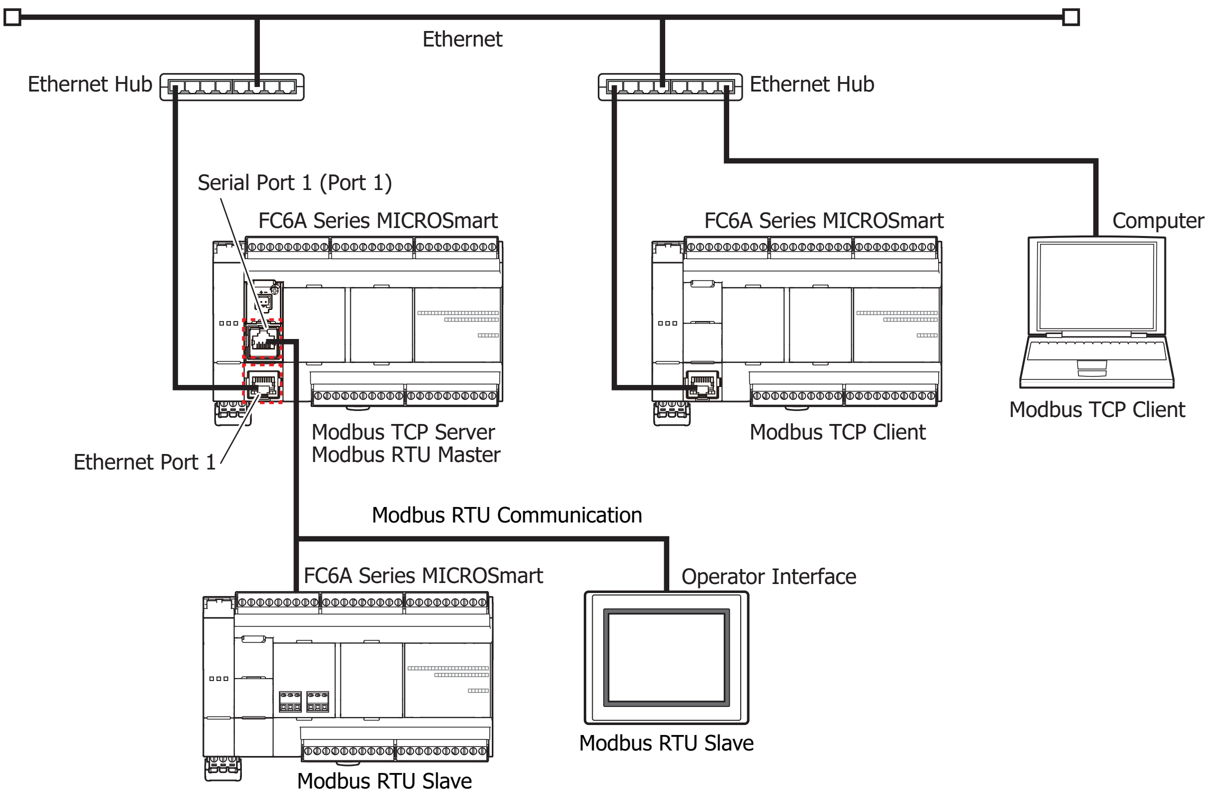
Modbus RTU Pass-Through Function

The Modbus RTU pass-through function allows a Modbus TCP client connected to a Modbus TCP network to access a Modbus RTU slave connected to a Modbus RTU network. Modbus RTU pass-through is supported in the All-in-One CPU module and the Plus CPU module but not supported in CAN J1939 All-in-One CPU module.

For the All-in-One CPU module, this function can read and write device information on a Modbus RTU slave device connected to serial port 1 of the FC6A Series MICROSmart from the Modbus TCP client device connected to Ethernet port 1.

The Modbus RTU pass-through function is only supported between Ethernet port 1 and the serial port 1 on the FC6A Series MICROSmart.

For the Plus CPU module, a Modbus RTU slave device connected to port 1 to 33 of the Plus CPU module can be selected and device information can be read and written from the Modbus TCP client device connected to Ethernet port 1 or 2. The Modbus RTU pass-through function is only supported between Ethernet port 1 and 2 and port 1 to 33 on the Plus CPU module.



Modbus TCP Communication Format in the Modbus RTU Pass-Through Function

The Modbus RTU pass-through function deletes the Modbus TCP headers from the Modbus TCP communication frame received by the Modbus TCP server and sends the frames with a CRC added to the "Unit ID", "Function code", and "Description" frames to the Modbus RTU slave specified by the unit ID.

The Modbus TCP communication format in the Modbus RTU pass-through function is as follows.

Modbus TCP Communication Format

Transaction ID	Protocol ID	Message Length (bytes)	Unit ID	Function Code	Description
2 bytes	2 bytes	2 bytes	1 byte	1 byte	N bytes

Modbus TCP Header

RTU Mode Communication Format

"Idle"	Slave No.	Function Code	Description	CRC	"Idle"
3.5 characters	1 byte	1 byte	N bytes	2 bytes	3.5 characters

Transaction ID
The Modbus TCP server returns the transaction ID from the client as is. The client can confirm to which request the response was returned. Enter 0 when there is no particular check to perform.

Protocol ID
This number indicates the Modbus TCP protocol and is 0.

Message Length
Represents the length of the message that follows in bytes.

Unit ID
The ID for identifying the device. It stores the slave number of the Modbus RTU slave. The Unit ID and subsequent frames are passed through to the Modbus RTU network.

If 255 is specified for the unit ID, the frames are processed by the Modbus TCP server (local station) and not passed through to the Modbus RTU slave.

Function Code
The function number such as reading or writing.

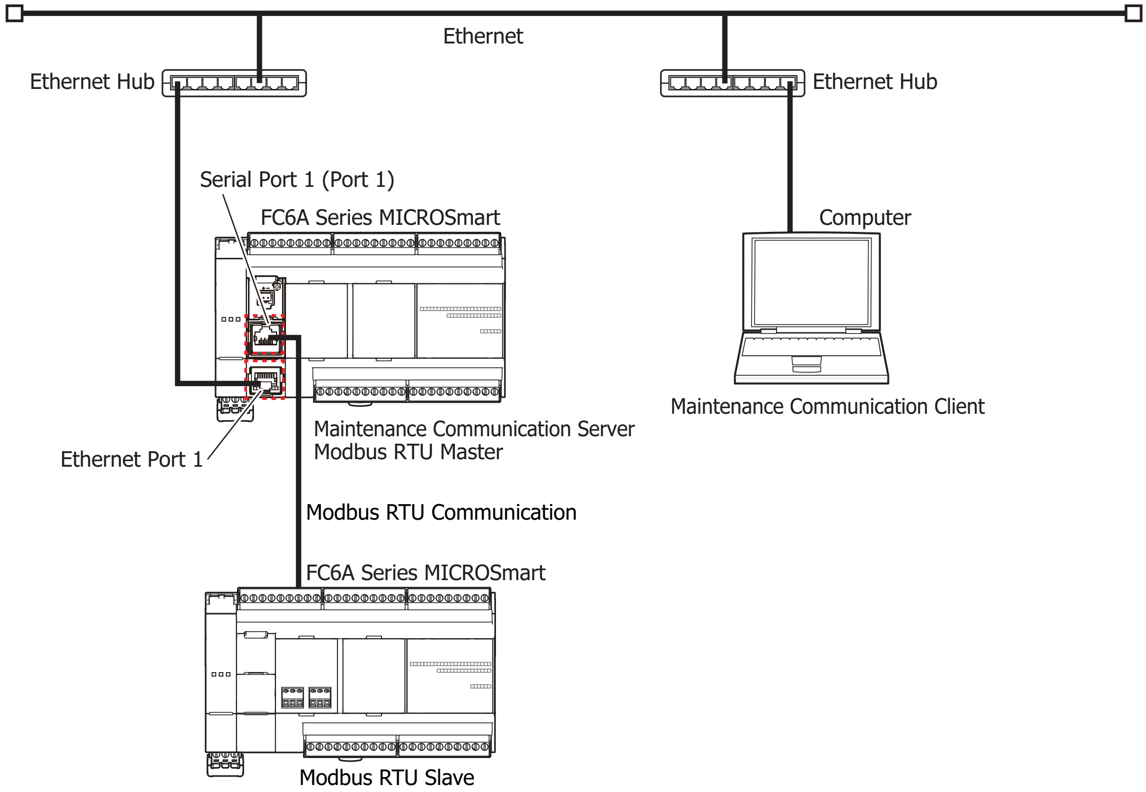
Description
The data required for processing.

Modbus RTU Pass-Through Function Specifications

Supported Port	**Ethernet Communication**	All-in-One CPU module: Ethernet port 1 Plus CPU module: Ethernet port 1 and 2
	Serial Communication	All-in-One CPU module: port 1 Plus CPU module: port 1 to 33
Communication Mode		Modbus TCP server (slave) and Modbus RTU master
Enable/disable Settings		Configure in **Function Area Settings**. * There is no limit on the number of server connections that can be enabled. * Can be used in combination with pass-through through the maintenance communication server.
Specifying the Pass-Through Destination		Specify the Modbus RTU slave for pass-through with the Modbus TCP communication frame unit ID. When the Modbus RTU pass-through function is enabled Unit ID (0 to 254): Modbus RTU pass-through is performed. Unit ID (255): Processed by the Modbus TCP server. When the Modbus RTU pass-through function is disabled Unit ID (0 to 255): Processed by the Modbus TCP server.
Available Number of Bytes for Communication		512 bytes * Data (a frame) that exceeds 512 bytes cannot be passed through with the pass-through function. When more than 512 bytes of data are received, that data including the previously received data is discarded.
Pass-Through Operation		When the unit ID (slave station number) is not 255, the FC6A Series MICROSmart issues a pass-through request to communication port and waits for a response from the Modbus RTU slave. When the unit ID is 255, the FC6A Series MICROSmart handles the request as communication to the local station and returns a response to the Modbus TCP client. When there are pass-through requests via the ModbusTCP Server, those requests are processed with a higher priority than the communication requests of the ModbusRTU master.

Modbus RTU Pass-Through Using the Maintenance Communication Server

When the FC6A Series MICROSmart is set as a maintenance communication server, it can access Modbus RTU slaves connected to a Modbus RTU network. From a maintenance communication client, such as WindLDR or Data File Manager installed in a remote computer, devices of an FC6A Series MICROSmart set as a Modbus RTU slave can be read and written and project data can be downloaded and uploaded through an FC6A Series MICROSmart (Modbus RTU Master) connected to Ethernet.



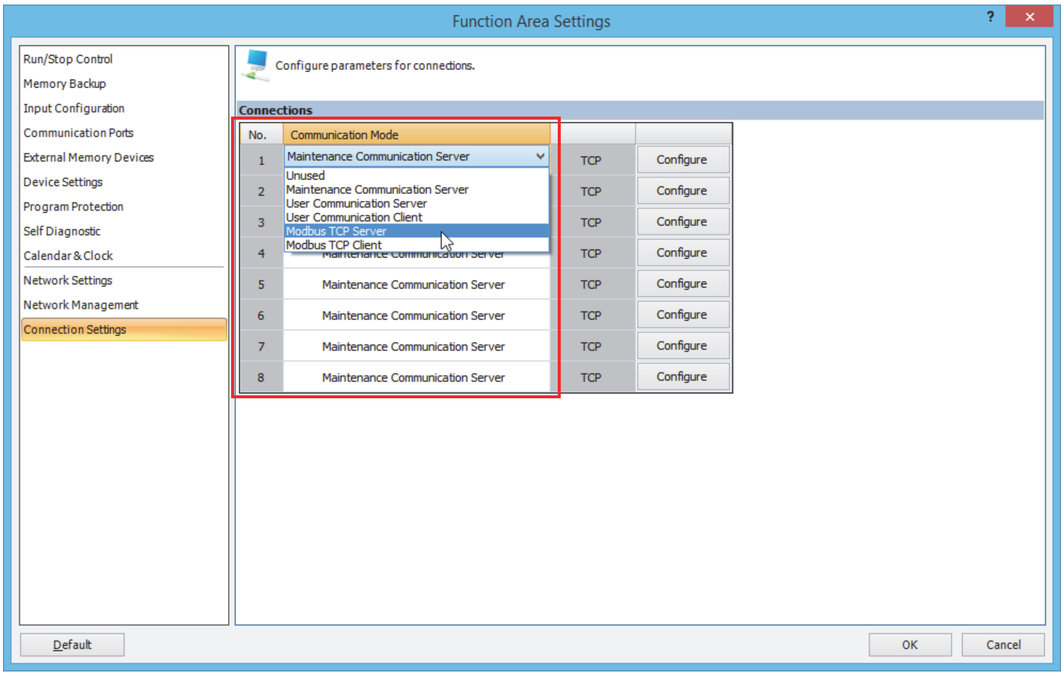
Modbus RTU Pass-Through Function Settings

Programming WindLDR

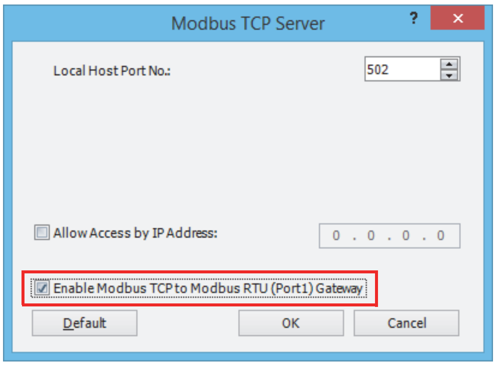
To use the Modbus RTU pass-through function, download a user program to the FC6A Series MICROSmart configured in **Function Area Settings**.

All-in-One CPU module

1. On the WindLDR **Configuration** tab, in the **Function Area Settings** group, click **Connection Settings**.
   The **Function Area Settings** dialog box is displayed.
2. Click **Communication Mode** for **Port 1** and select **Modbus TCP Server**.
   The **Modbus TCP Server** dialog box is displayed.

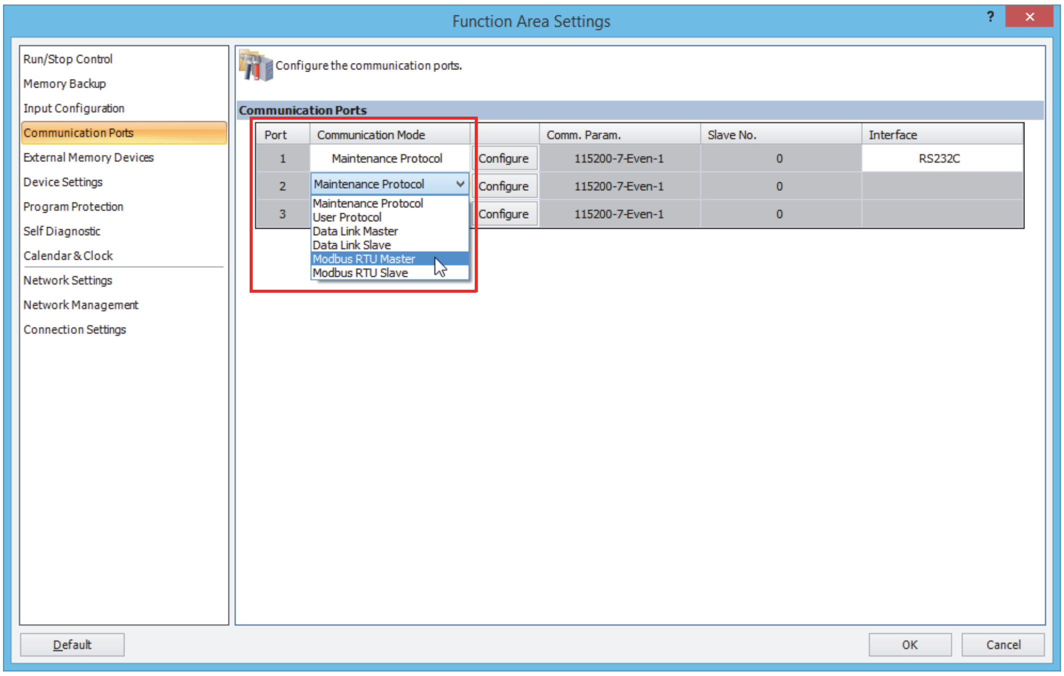


3. Select the **Enable Modbus TCP to Modbus RTU (Port1) Gateway** check box and configure the local host port number and the allow access by IP address.
   Specify a number that is different from the other server connections for the local host port number of the port that will use the pass-through function.



4. Click **OK**.

5. Click **Communication Mode** for **Port 1** and select **Modbus RTU Master**.
   The **Modbus RTU Master Request Table** dialog box is displayed.

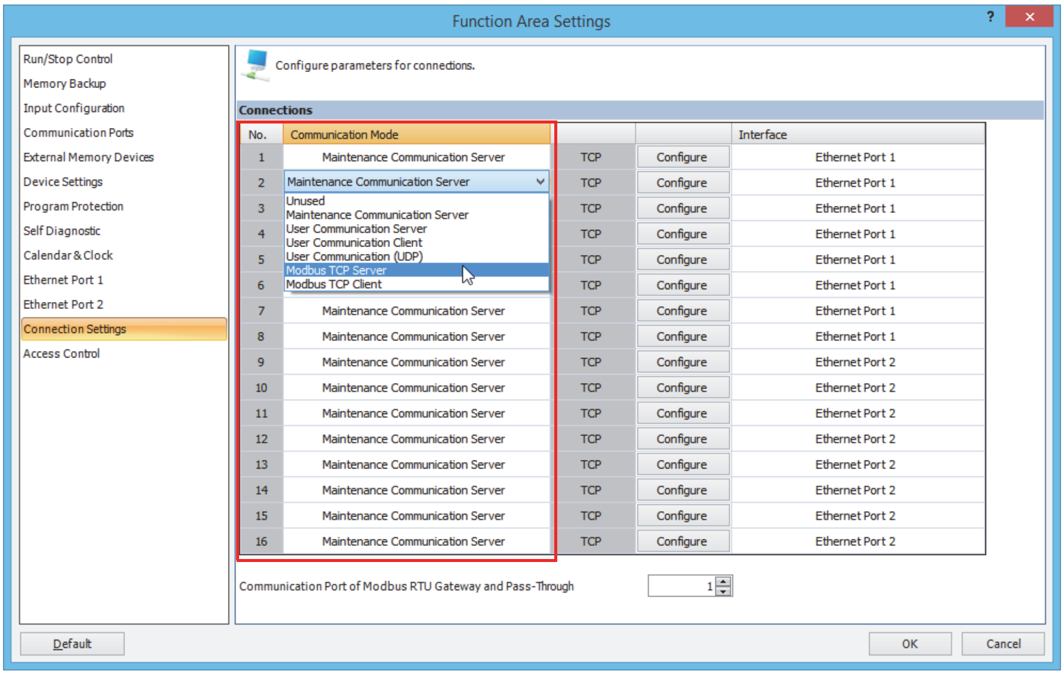


6. Configure the **Modbus RTU Master Request Table** dialog box items.
   For details, see "Modbus RTU Master Communication" on page 6-3.
7. Download the user program to the FC6A Series MICROSmart.

This concludes configuring the settings to use the Modbus RTU pass-through function.

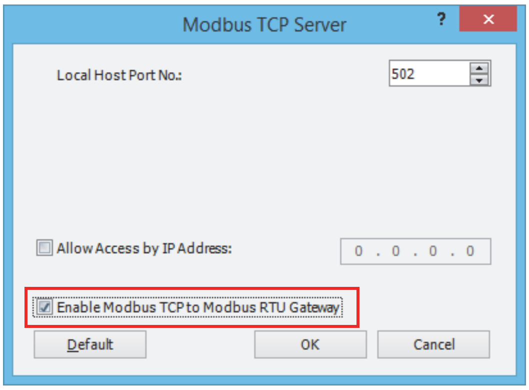
Plus CPU module

1. On the WindLDR **Configuration** tab, in the **Function Area Settings** group, click **Connection Settings**.
   The **Function Area Settings** dialog box is displayed.
2. Click **Communication Mode** for the connection number that will be used and select **Modbus TCP Server**.
   The **Modbus TCP Server** dialog box is displayed.



Note: For Modbus RTU pass-through using the maintenance communication server, select **Maintenance Communication Server**.
The **Maintenance Communication Server** dialog box is displayed.

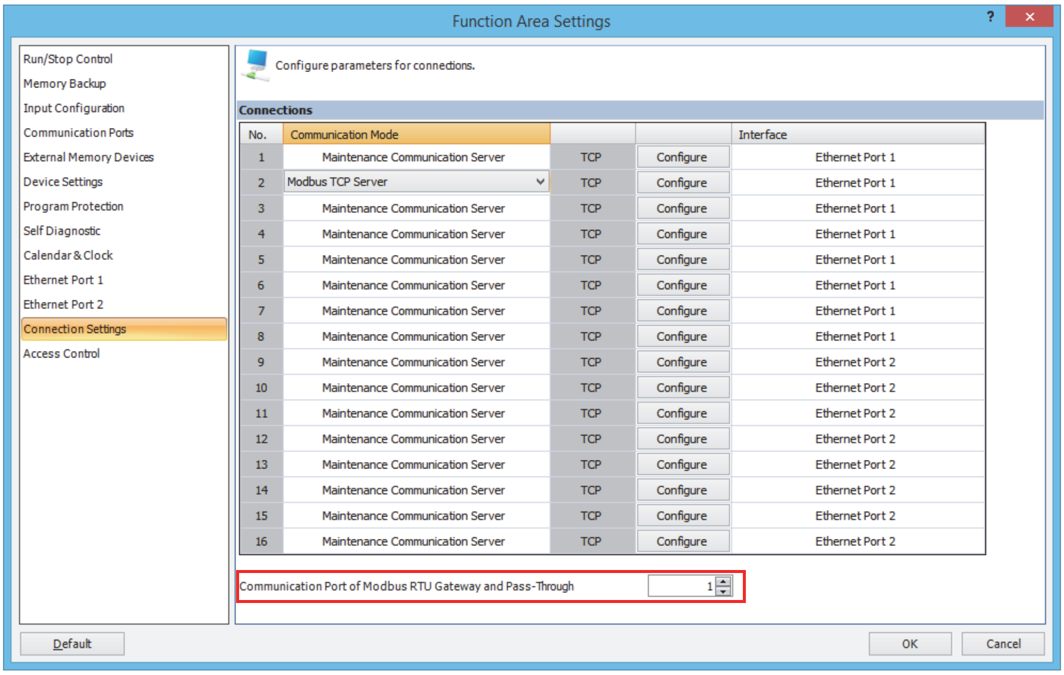
3. Select the **Enable Modbus TCP to Modbus RTU Gateway** check box and configure the local host port number and allow access by IP address.
   Set a number that is different from the other server connections for the local host port number of the port that will use the pass-through function.



Note: For the **Maintenance Communication Server** dialog box, select the **Enable Pass-Through over Modbus RTU** check box and configure the local host port number and allow access by IP address.

4. Click **OK**.

5. With **Communication Port of Modbus RTU Gateway and Pass-Through**, set the port number that will be used (1 to 33).



6. Download the user program to the FC6A Series MICROSmart.

This concludes configuring the settings to use the Modbus RTU pass-through function.

7: DATA LINK COMMUNICATION

Introduction

This chapter describes the data link communication function used to set up a distributed control system.

In data link communication, data can be exchanged between a master station and slave stations by connecting a maximum of 31 slave station FC6A Series MICROSmarts to a master station FC6A Series MICROSmart. The master station has a total of 13 words worth of data registers for each slave station: 6 words for sending data to the slave station, 6 words for receiving data from the slave station, and 1 word for the communication status/error. Each slave station has a total of 13 words worth of data registers: 6 words for sending data to the master station, 6 words for receiving data from the master station, and 1 word for the communication status/error.

Data link communication proceeds independently of the user program execution, and the data registers for the data link communication are updated at the END processing.

The FC6A Series MICROSmart data link function is compatible with the data link functions of the MICROSmart FC5A Series and FC4A Series.

One CPU module can be either a master station or a slave station. Data link master and slave cannot be used at the same time.

Communication Ports Used in Data Link Communication

Supported models and communication ports are as follows.

Communication Port	All-in-One CPU Module			CAN J1939 All-in-One CPU Module	Plus CPU Module	
	16-I/O Type	24-I/O Type	40-I/O Type		Plus 16-I/O Type	Plus 32-I/O Type
Port 1	Yes	Yes	Yes	—	Yes*2	Yes*2
Port 2	Yes*1	Yes*1	Yes*1	Yes*1	Yes*2	Yes*2
Port 3	—	—	Yes*1	Yes*1	Yes*3	Yes*3
Port 4 to 33	Yes (3 maximum)*4				Yes (15 maximum)*5	

*1 When an RS232C communication cartridge or RS485 communication cartridge is connected.
*2 When a cartridge base module and an RS232C communication cartridge or RS485 communication cartridge are connected.
*3 When the HMI module and an RS232C communication cartridge or RS485 communication cartridge are connected.
*4 Up to three communication modules can be connected to the All-in-One CPU module and CAN J1939 All-in-One CPU module, expanding the communication ports 4 to 9.
*5 Up to 15 communication modules can be connected to the Plus CPU module, expanding the communication ports 4 to 33.

Data Link Specifications

Mode	Details
Electric Specifications	Compliance with EIA RS485
Baud Rate	19,200, 38,400, 57,600 bps
Synchronization	Start-stop synchronization Start bit: 1 Data bits: 7 Parity: Even Stop bit: 1
Communication Cable	Shielded twisted pair cable
Maximum Cable Length	200 m (656 feet)
Maximum Slave Stations	31 slave stations
Transmit/Receive Data	Transmit data: 186 words maximum, Receive data: 186 words maximum 0 through 6 words each for transmission and receiving per slave station

Data Updating

Mode	Details
Effect on Scan Time	Communication processing between masters and slaves is not synchronized with the user program, and there is no effect on the scan time.
Data Update Timing	Data updating for both masters and slaves is performed in END processing. The update timing can be checked with the communication completed flag.
Master-compatible Models	FC6A Series MICROSmart (FC4A Series MICROSmart/FC5A Series MICROSmart)
Slave-compatible Models	FC6A Series MICROSmart (FC4A Series MICROSmart/FC5A Series MICROSmart)

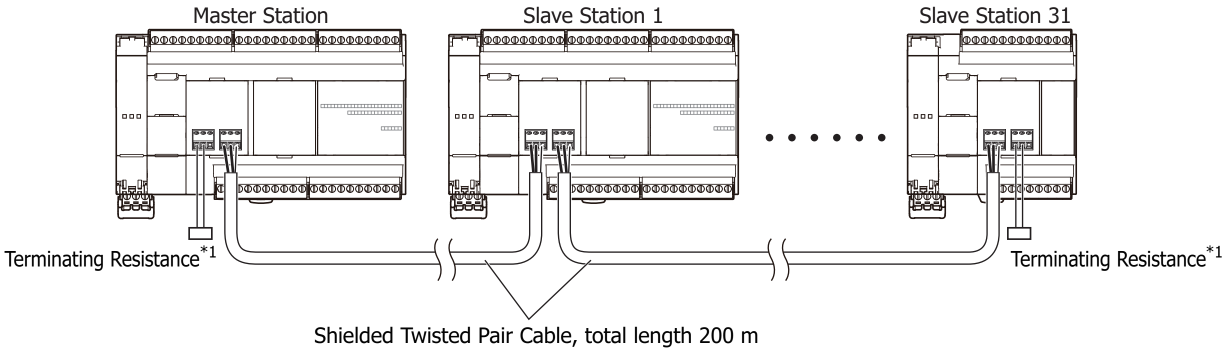
The master station sends the appropriate amount of data (in units of words) to the slave stations. The slave stations that received this transmission send an amount of data matching the number of words received to the master station. In this way data is exchanged. The data refresh is performed on the master and slave stations during the END processing. The master station can only exchange data with one slave station during one communication cycle.

If 31 slave stations are connected, the master station requires 31 communication cycles to exchange data with all slave stations.

The master station turns on the communication complete relay for only one scan after the completion of the communication cycle with a slave station.

Data Link System Setup

The data link function can be used by setting port 1 of the FC6A Series MICROSmart to RS-485 or by using a RS-485 communication cartridge in port 2 or port 3.



*1 When communication quality is unstable, add terminating resistance matched to the characteristic impedance to both ends. Use resistance with a rating of 1/2 W or higher.

Data Register Allocation for Transmit/Receive Data

The master station has 12 data registers assigned for data communication with each slave station. Each slave station has 12 data registers assigned for data communication with the master station. When data is set in data registers at the master station assigned for data link communication, the data is sent to the corresponding data registers at a slave station. When data is set in data registers at a slave station assigned for data link communication, the data is sent to the corresponding data registers at the master station.

Master Station

Slave Station Number	Data Register	Transmit/Receive Data	Slave Station Number	Data Register	Transmit/Receive Data
Slave 1	D900-D905	Transmit data to slave 1	Slave 17	D1092-D1097	Transmit data to slave 17
	D906-D911	Receive data from slave 1		D1098-D1103	Receive data from slave 17
Slave 2	D912-D917	Transmit data to slave 2	Slave 18	D1104-D1109	Transmit data to slave 18
	D918-D923	Receive data from slave 2		D1110-D1115	Receive data from slave 18
Slave 3	D924-D929	Transmit data to slave 3	Slave 19	D1116-D1121	Transmit data to slave 19
	D930-D935	Receive data from slave 3		D1122-D1127	Receive data from slave 19
Slave 4	D936-D941	Transmit data to slave 4	Slave 20	D1128-D1133	Transmit data to slave 20
	D942-D947	Receive data from slave 4		D1134-D1139	Receive data from slave 20
Slave 5	D948-D953	Transmit data to slave 5	Slave 21	D1140-D1145	Transmit data to slave 21
	D954-D959	Receive data from slave 5		D1146-D1151	Receive data from slave 21
Slave 6	D960-D965	Transmit data to slave 6	Slave 22	D1152-D1157	Transmit data to slave 22
	D966-D971	Receive data from slave 6		D1158-D1163	Receive data from slave 22
Slave 7	D972-D977	Transmit data to slave 7	Slave 23	D1164-D1169	Transmit data to slave 23
	D978-D983	Receive data from slave 7		D1170-D1175	Receive data from slave 23
Slave 8	D984-D989	Transmit data to slave 8	Slave 24	D1176-D1181	Transmit data to slave 24
	D990-D995	Receive data from slave 8		D1182-D1187	Receive data from slave 24
Slave 9	D996-D1001	Transmit data to slave 9	Slave 25	D1188-D1193	Transmit data to slave 25
	D1002-D1007	Receive data from slave 9		D1194-D1199	Receive data from slave 25
Slave 10	D1008-D1013	Transmit data to slave 10	Slave 26	D1200-D1205	Transmit data to slave 26
	D1014-D1019	Receive data from slave 10		D1206-D1211	Receive data from slave 26
Slave 11	D1020-D1025	Transmit data to slave 11	Slave 27	D1212-D1217	Transmit data to slave 27
	D1026-D1031	Receive data from slave 11		D1218-D1223	Receive data from slave 27
Slave 12	D1032-D1037	Transmit data to slave 12	Slave 28	D1224-D1229	Transmit data to slave 28
	D1038-D1043	Receive data from slave 12		D1230-D1235	Receive data from slave 28
Slave 13	D1044-D1049	Transmit data to slave 13	Slave 29	D1236-D1241	Transmit data to slave 29
	D1050-D1055	Receive data from slave 13		D1242-D1247	Receive data from slave 29
Slave 14	D1056-D1061	Transmit data to slave 14	Slave 30	D1248-D1253	Transmit data to slave 30
	D1062-D1067	Receive data from slave 14		D1254-D1259	Receive data from slave 30
Slave 15	D1068-D1073	Transmit data to slave 15	Slave 31	D1260-D1265	Transmit data to slave 31
	D1074-D1079	Receive data from slave 15		D1266-D1271	Receive data from slave 31
Slave 16	D1080-D1085	Transmit data to slave 16	—		
	D1086-D1091	Receive data from slave 16			

If any slave stations are not connected, master station data registers which are assigned to the vacant slave stations can be used as ordinary data registers.

Slave Station

Data	Data Register	Transmit/Receive Data
Slave Station Data	D900-D905	Transmit data to master station
	D906-D911	Receive data from master station

Slave station data registers D912 through D1271 can be used as ordinary data registers.

Special Data Registers for Data Link Communication Error

In addition to data registers assigned for data communication, the master station has 31 special data registers and each slave station has one special data register to store data link communication error codes. If any communication error occurs in the data link system, communication error codes are set to a corresponding data register for link communication error at the master station and to data register D8069 at the slave station. For details of link communication error codes, see below.

If a communication error occurs in the data link communication system, the data is resent two times. If the error still exists after three attempts, then the error code is set to the data registers for data link communication error. Since the error code is not communicated between the master and slave stations, error codes must be cleared individually.

Master Station

Special Data Register	Data Link Communication Error Data	Special Data Register	Data Link Communication Error Data
D8069	Slave station 1 communication error	D8085	Slave station 17 communication error
D8070	Slave station 2 communication error	D8086	Slave station 18 communication error
D8071	Slave station 3 communication error	D8087	Slave station 19 communication error
D8072	Slave station 4 communication error	D8088	Slave station 20 communication error
D8073	Slave station 5 communication error	D8089	Slave station 21 communication error
D8074	Slave station 6 communication error	D8090	Slave station 22 communication error
D8075	Slave station 7 communication error	D8091	Slave station 23 communication error
D8076	Slave station 8 communication error	D8092	Slave station 24 communication error
D8077	Slave station 9 communication error	D8093	Slave station 25 communication error
D8078	Slave station 10 communication error	D8094	Slave station 26 communication error
D8079	Slave station 11 communication error	D8095	Slave station 27 communication error
D8080	Slave station 12 communication error	D8096	Slave station 28 communication error
D8081	Slave station 13 communication error	D8097	Slave station 29 communication error
D8082	Slave station 14 communication error	D8098	Slave station 30 communication error
D8083	Slave station 15 communication error	D8099	Slave station 31 communication error
D8084	Slave station 16 communication error	—	—

If any slave stations are not connected, master station data registers which are assigned to the vacant slave stations can be used as ordinary data registers.

Slave Station

Special Data Register	Data Link Communication Error Data
D8069	Slave station communication error

Note: Slave station data registers D8070 through D8099 can be used as ordinary data registers.

Data Link Communication Error Code

When an error occurs during the data link, the data is sent again (the operation is retried) up to twice.

If an error occurs on the third attempt to send the data, the error number is set in the data register for the communication status/error of the master station and the corresponding slave station.

Error Code	Error Details
1h (1)	Overrun error (data is received when the receive data registers are full)
2h (2)	Framing error (failure to detect start or stop bit)
4h (4)	Parity error (an error was found by the parity check)
8h (8)	Receive timeout (line disconnection)
10h (16)	BCC (block check character) error (disparity with data received up to BCC)
20h (32)	Retry cycle over (error occurred in all 3 trials of communication)
40h (64)	I/O definition quantity error (discrepancy of transmit/receive station number or data quantity)

When more than one error is detected in the data link system, the total of error codes is indicated. For example, when framing error (error code 2h) and BCC error (error code 10h) are found, error code 12h (18) is stored.

Data Link Communication between Master and Slave Stations

The master station has 6 data registers assigned to transmit data to a slave station and 6 data registers assigned to receive data from a slave station. The quantity of data registers for data link can be selected from 0 through 6 using WindLDR. The following examples illustrate how data is exchanged between the master and slave stations when 2 or 6 data registers are used for data link communication with each of 31 slave stations.

Example 1: Transmit Data 2 Words and Receive Data 2 Words

Master Station		Slave Stations		
D8069	Communication Error	D8069	Communication Error	Slave Station 1
D900 - D901	Transmit Data	D900 - D901	Transmit Data	
D906 - D907	Receive Data	D906 - D907	Receive Data	
D8070	Communication Error	D8069	Communication Error	Slave Station 2
D912 - D913	Transmit Data	D900 - D901	Transmit Data	
D918 - D919	Receive Data	D906 - D907	Receive Data	
D8071	Communication Error	D8069	Communication Error	Slave Station 3
D924 - D925	Transmit Data	D900 - D901	Transmit Data	
D930 - D931	Receive Data	D906 - D907	Receive Data	
D8072	Communication Error	D8069	Communication Error	Slave Station 4
D936 - D937	Transmit Data	D900 - D901	Transmit Data	
D942 - D943	Receive Data	D906 - D907	Receive Data	
⋮	⋮	⋮	⋮	⋮
D8098	Communication Error	D8069	Communication Error	Slave Station 30
D1248 - D1249	Transmit Data	D900 - D901	Transmit Data	
D1254 - D1255	Receive Data	D906 - D907	Receive Data	
D8099	Communication Error	D8069	Communication Error	Slave Station 31
D1260 - D1261	Transmit Data	D900 - D901	Transmit Data	
D1266 - D1267	Receive Data	D906 - D907	Receive Data	

Example 2: Transmit Data 6 Words and Receive Data 6 Words

Master Station		Slave Stations		
D8069	Communication Error	D8069	Communication Error	Slave Station 1
D900 - D905	Transmit Data	D900 - D905	Transmit Data	
D906 - D911	Receive Data	D906 - D911	Receive Data	
D8070	Communication Error	D8069	Communication Error	Slave Station 2
D912 - D917	Transmit Data	D900 - D905	Transmit Data	
D918 - D923	Receive Data	D906 - D911	Receive Data	
D8071	Communication Error	D8069	Communication Error	Slave Station 3
D924 - D929	Transmit Data	D900 - D905	Transmit Data	
D930 - D935	Receive Data	D906 - D911	Receive Data	
D8072	Communication Error	D8069	Communication Error	Slave Station 4
D936 - D941	Transmit Data	D900 - D905	Transmit Data	
D942 - D947	Receive Data	D906 - D911	Receive Data	
⋮	⋮	⋮	⋮	⋮
D8098	Communication Error	D8069	Communication Error	Slave Station 30
D1248 - D1253	Transmit Data	D900 - D905	Transmit Data	
D1254 - D1259	Receive Data	D906 - D911	Receive Data	
D8099	Communication Error	D8069	Communication Error	Slave Station 31
D1260 - D1265	Transmit Data	D900 - D905	Transmit Data	
D1266 - D1271	Receive Data	D906 - D911	Receive Data	

Note: When using port 2 or port 3, the communication status and error are stored in the registers configured in **Function Area Settings**.

Special Internal Relays for Data Link Communication

Special internal relays M8005 through M8007 and M8080 through M8117 are assigned for the data link communication.

M8005 Data Link Communication Error

When an error occurs during communication in the data link system, M8005 turns on. The M8005 status is maintained when the error is cleared and remains on until M8005 is reset using WindLDR or until the CPU is turned off. The cause of the data link communication error can be checked using **Online** > **Monitor** > **Monitor**, followed by **Online** > **Status** > **Error Status: Details**. See "Data Link Communication Error Code" on page 7-5.
When performing communication on port 2 or port 3, this function is disabled.

M8006 Data Link Communication Prohibit Flag (Master Station)

When M8006 at the master station is turned on in the data link system, data link communication is stopped. When M8006 is turned off, data link communication resumes. The M8006 status is maintained when the CPU is turned off and remains on until M8006 is reset using WindLDR.
When M8006 is on at the master station, M8007 is turned on at slave stations in the data link system.

M8007 Data Link Communication Initialize Flag (Master Station)
Data Link Communication Stop Flag (Slave Station)

M8007 has a different function at the master or slave station of the data link communication system.

Master station: Data link communication initialize flag

When M8007 at the master station is turned on during operation, the link configuration is checked to initialize the data link system. When a slave station is powered up after the master station, turn M8007 on to initialize the data link system. After a data link system setup is changed, M8007 must also be turned on to ensure correct communication.

Slave station: Data link communication stop flag

When a slave station does not receive communication data from the master station for 10 seconds or more in the data link system, M8007 turns on. When a slave station does not receive data in 10 seconds after initializing the data link system, M8007 also turns on at the slave station. When the slave station receives correct communication data, M8007 turns off.

M8080-M8116 Slave Station Communication Completion Relay (Master Station)

Special internal relays M8080 through M8116 are used to indicate the completion of data refresh. When data link communication with a slave station is complete, a special internal relay assigned for the slave station is turned on for one scan time at the master station.

Special Internal Relay	Slave Station Number	Special Internal Relay	Slave Station Number	Special Internal Relay	Slave Station Number
M8080	Slave Station 1	**M8092**	Slave Station 11	**M8104**	Slave Station 21
M8081	Slave Station 2	**M8093**	Slave Station 12	**M8105**	Slave Station 22
M8082	Slave Station 3	**M8094**	Slave Station 13	**M8106**	Slave Station 23
M8083	Slave Station 4	**M8095**	Slave Station 14	**M8107**	Slave Station 24
M8084	Slave Station 5	**M8096**	Slave Station 15	**M8110**	Slave Station 25
M8085	Slave Station 6	**M8097**	Slave Station 16	**M8111**	Slave Station 26
M8086	Slave Station 7	**M8100**	Slave Station 17	**M8112**	Slave Station 27
M8087	Slave Station 8	**M8101**	Slave Station 18	**M8113**	Slave Station 28
M8090	Slave Station 9	**M8102**	Slave Station 19	**M8114**	Slave Station 29
M8091	Slave Station 10	**M8103**	Slave Station 20	**M8115**	Slave Station 30
—	—	—	—	**M8116**	Slave Station 31

M8080 Communication Completion Relay (Slave Station)

When data link communication with a master station is complete, special internal relay M8080 at the slave station is turned on for one scan time.

M8117 All Slave Station Communication Completion Relay

When data link communication with all slave stations is complete, special internal relay M8117 at the master station is turned on for one scan time. M8117 at slave stations does not go on.

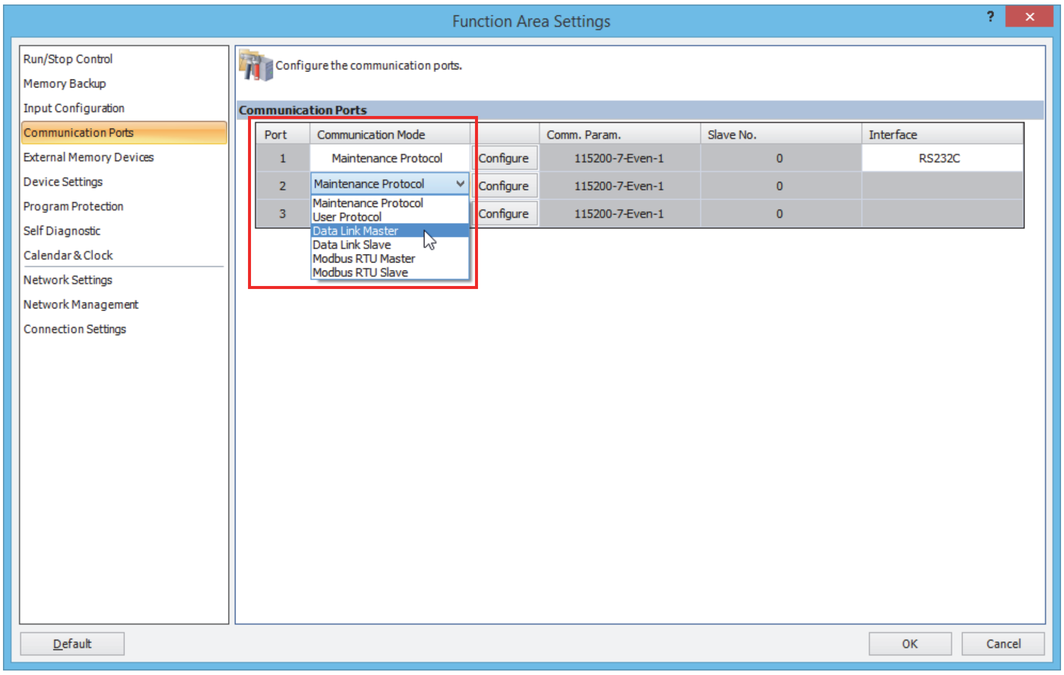
Programming WindLDR

The Communication page in the **Function Area Settings** is used to program the data link master and slave stations.

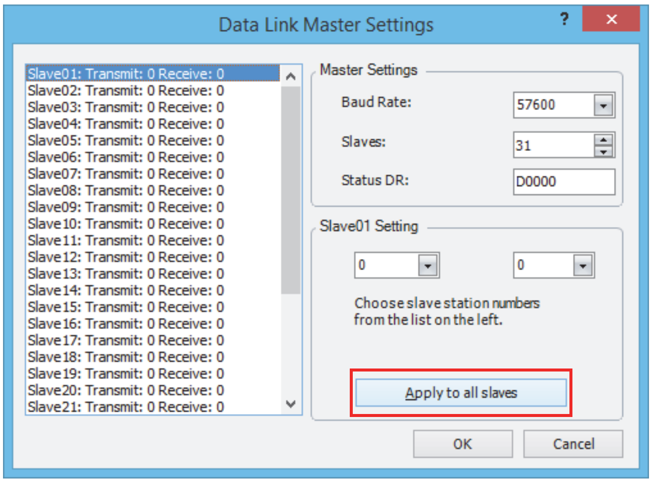
Since these settings relate to the user program, the user program must be downloaded to the CPU module after changing any of these settings.

Data Link Master Station

1. From the WindLDR menu bar, select **Configuration** > **Comm. Ports**.
   The **Function Area Settings** dialog box for Communication Ports appears.
2. Click **Communication Mode** for the port to use and select **Data Link Master**.
   The **Data Link Master Settings** dialog box appears.



3. Set the Baud Rate, Slaves, and Status DR under Master Settings.
4. Select a slave number from the list, and then set the number of words to transmit/receive.
   To make the same settings for all slave stations in the list, click **Apply to all slaves**.



Number of words to transmit:	The number of words worth of data to transmit from the master station to the slave stations
Number of words to receive:	The number of words worth of data that the master station receives from each slave station

5. Click **OK**.
6. Create the user program.
7. Transfer the user program.

This concludes configuring the settings.

Note: When you use data link communication, you have to turn on the slave stations first before turning on the master station. If you turn on the master station before turning on the slave stations, the master station cannot recognize the slave stations. If the master station cannot recognize the slave stations, you have to initialize the data link. To initialize the data link, click **Initialize Data Link** in WindLDR or turn on special internal relay M8007 in the user program.

1. Select **Online** > **Monitor** > **Monitor** > **Start Monitor**.
   WindLDR enters monitor mode.
2. Select **Online** > **PLC** > **Initialize** > **Initialize Data Link**.
   The slave stations are recognized by the master station.

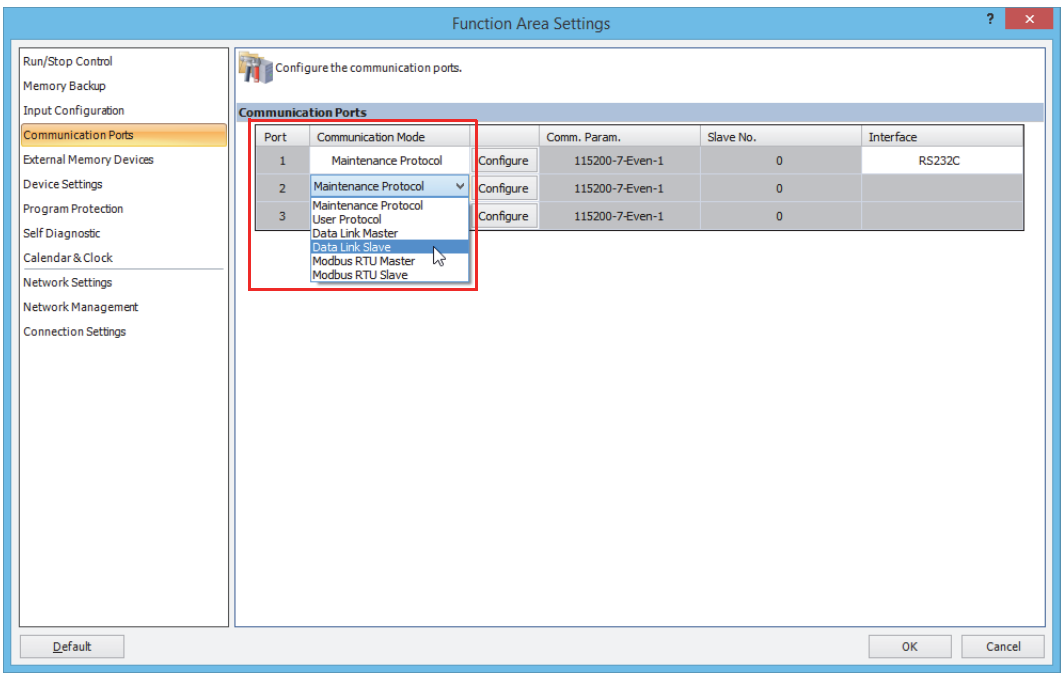
If you cannot perform WindLDR maintenance communication, insert the following program into the master station's user program.
Stop and then run the master station to have it recognize the slave stations.

M8120: Initialize pulse
M8007: Communication initialization flag

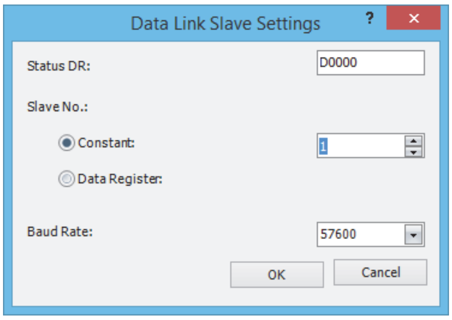
When operation (RUN) starts, M8007 is turned on for a period of one scan, and the data link is initialized.

Data Link Slave Station

1. From the WindLDR menu bar, select **Configuration** > **Comm. Ports**.
   The **Function Area Settings** dialog box for Communication Ports appears.
2. Click **Communication Mode** for the port to use and select **Data Link Slave**.
   The **Data Link Slave Settings** dialog box appears.



3. Set the Slave No. and the Baud Rate.
Select **Constant** or **Data Register** under **Slave No**. If you select **Constant**, set the slave No. If you select **Data Register**, write the slave No. into the special data register.



4. Click **OK**.

5. Create the user program.

6. Transfer the user program.

This concludes configuring the settings.

Note: If you select **Data Register** under **Slave No.** in the **Data Link Slave Settings** dialog box, you can change the slave No. of the slave station without downloading the user program by writing the slave No. into the following special data registers.

Communication Port	Special Data Register
Port 1	D8100
Port 2	D8102
Port 3	D8103
Port 4 to 9*1	D8040 to D8045
Port 10 to 33*1*2	D8735 to D8758

*1 When communication modules are connected.
*2 Plus CPU module only.

Note: After making changes, correctly initialize the master station and set the slave stations.
Use one of the following methods to initialize the master station.
(1) Turn off the master station, and then turn it back on.
(2) Set M8007 on the master station.
(3) Select **Online** > **PLC** > **Initialize** > **Initialize Data Link**.
The valid slave No. are 1 to 31. If you specify any other value, the station number set in the **Function Area Settings** will be used.

Data Refresh

In the data link communication, the master station communicates with only one slave station in one communication cycle. When a slave station receives a communication from the master station, the slave station returns data stored in data registers assigned for data link communication. After receiving data from slave stations, the master station stores the data into data registers allocated to each slave station. The process of updating data into data registers is called refresh. When the maximum 31 slave stations are connected, the master station requires 31 communication cycles to communicate with all slave stations.

Mode	Separate Refresh Mode
Scan Time	Since the communication between the master station and slave stations proceeds independently of the user program scanning, the scan time is not affected.
Data Refresh Timing	At both master and slave stations, received data is refreshed at the END processing. Refresh completion can be confirmed with communication completion special internal relays M8080 through M8117.
Applicable Master Station	FC6A Series MICROSmart (FC4A/FC5A MICROSmart)
Applicable Slave Station	FC6A Series MICROSmart (FC4A/FC5A MICROSmart)

Both master and slave stations refresh communication data at the END processing. When data refresh is complete, communication completion special internal relays M8080 through M8116 (slave station communication completion relay) go on at the master station for one scan time after the data refresh. At each slave station, special internal relay M8080 (communication completion relay) goes on.

When the master station completes communication with all slave stations, special internal relay M8117 (all slave station communication completion relay) goes on at the master station for one scan time.

Total Refresh Time at Master Station for Communication with All Slave Stations (Trfn)

The master station requires the following time to refresh the transmit and receive data for communication with all slave stations, that is the total of refresh times.

[Baud Rate 19,200 bps] Trfn = ∑ Trf = ∑ {4.2 ms + 2.4 ms × (Transmit Words + Receive Words) + 1 scan time}

[Baud Rate 38,400 bps] Trfn = ∑ Trf = ∑ {2.2 ms + 1.3 ms × (Transmit Words + Receive Words) + 1 scan time}

[Baud Rate 57,600 bps] Trfn = ∑ Trf = ∑ {1.6 ms + 0.9 ms × (Transmit Words + Receive Words) + 1 scan time}

Refresh Time

When data link communication is performed with such parameters as transmit words 6, receive words 6, slave stations 8, and average scan time 20 ms, then the total refresh time Trf8 for communication with all eight slave stations will be:

[Baud Rate 19,200 bps] Trf8 = {4.2 ms + 2.4 ms × (6 + 6) + 20 ms} × 8 = 424.0 ms

[Baud Rate 38,400 bps] Trf8 = {2.2 ms + 1.3 ms × (6 + 6) + 20 ms} × 8 = 302.4 ms

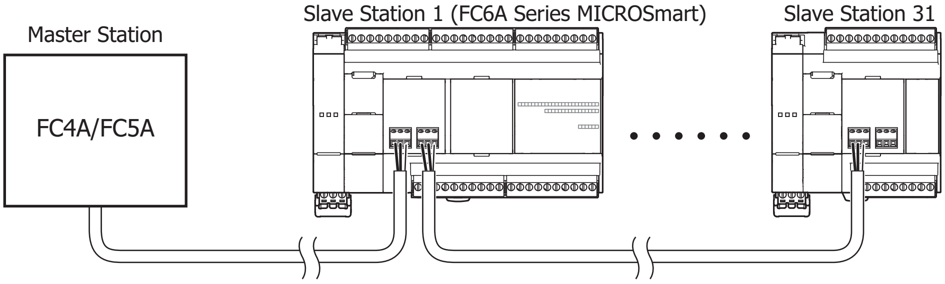
[Baud Rate 57,600 bps] Trf8 = {1.6 ms + 0.9 ms × (6 + 6) + 20 ms} × 8 = 259.2 ms

Data Link with Other PLCs

The FC6A Series MICROSmart can exchange data by communicating with IDEC FC4A Series and FC5A Series MICROSmart (data link).

Data Link with FC4A/FC5A

FC4A/FC5A Settings	MICROSmart Settings
Transmit data: 6 words Receive data: 6 words Baud rate: 19,200 or 38,400 bps	Set the data link station slave station to 1 to 31.



8: J1939 COMMUNICATION

This chapter describes J1939 communication in the CAN J1939 All-in-One CPU module.

The following abbreviations are used in the text to simplify the descriptions.

Abbreviation	Official Name
CAN	Controller Area Network
ECU	Electronic Controller Unit
NIECU	Network Interconnection ECU
CA	Controller Application
PDU	Protocol Data Unit
PGN	Parameter Group Number
SA	Source Address
DA	Destination Address
SAE	Society of Automotive Engineers
SPN	Suspect Parameter Number

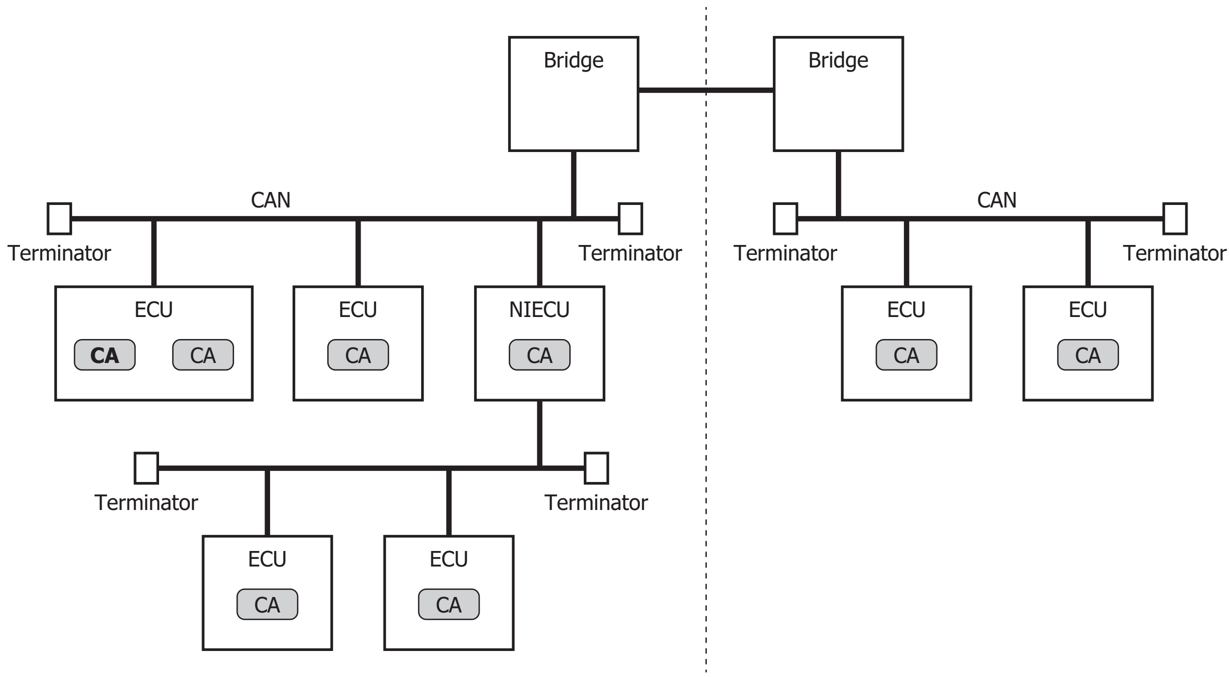
Overview of J1939 Communication over CAN

The CAN J1939 All-in-One CPU module is compatible with CAN communication based on SAE J1939. The CAN J1939 All-in-One CPU module can communicate with other J1939 communication-compatible devices by connecting the CAN port to a J1939 communication network. For details on J1939 communication, refer to the SAE J1939 standard.

SAE J1939 Overview

J1939 Communication Network

A J1939 communication network is composed of ECUs (engine, brake, etc.) and NIECUs (repeaters, routers, etc.), and each ECU has one or more CAs. A CA is assigned an address between 0 and 253 and a unique 64-bit ID (NAME), and messages can be exchanged between CAs using the addresses. If there is an address conflict between CAs, the conflict will be resolved based on the priority indicated in NAME.



Device Address

Addresses are defined between 0 and 255.

However, 254 (null address) is the result of an address conflict and is only used when an address cannot be obtained. 255 (global address) is used when broadcasting messages to all CAs without specifying a specific CA.

Address	Description	Details
0 to 253	Address range allocated to CAs	For a portion of addresses, the use has been determined according to the type of connected device. For details, refer to the SAE J1939 standard.
254	Null address	Set as the result of an address conflict when an address between 0 and 253 could not be obtained.
255	Global address	Used when transmitting a message as a broadcast.

NAME (64 bits total)

The 64-bit NAME is split into multiple fields. The content for a portion of the fields is already defined in the standard. NAME must be set to a value that is not duplicated by another device on the same J1939 communication network. For details on the definition of each bit, refer to the SAE J1939 standard.

Field Name	Bit Length	SAE Defined*1	Item
Arbitrary Address Capable	1 (high order)	—	Dynamic address support
Industry Group	3	Yes	Industry group
Vehicle System Instance	4	—	Vehicle system instance
Vehicle System	7	Yes	Vehicle system
Reserved	1	Yes	Reserved (fixed as 0)
Function	8	Yes	Function
Function Instance	5	—	Function instance
ECU Instance	3	—	ECU instance
Manufacturer Code	11	Yes	Manufacturer code
Identify Number	21 (low order)	—	ID number

*1 "Yes" items are defined in the standard. Set those items according to the used network and purpose.

Note: If the network that will be used is not known, set NAME as the maximum value of each field.

The CAN J1939 All-in-One CPU module resolves address conflicts with CAs according to the value of the Arbitrary Address Capable bit.

If the Arbitrary Address Capable bit is 0

The CAN J1939 All-in-One CPU module operates as a fixed address CA. If there is a CA that has the same address as the local address, the CAN J1939 All-in-One CPU module can exchange messages with the set local address if its priority is high.
If the CAN J1939 All-in-One CPU module priority is low, the local address is 254 (null address).

If the Arbitrary Address Capable bit is 1

The CAN J1939 All-in-One CPU module operates as a dynamic address CA. If there are CAs that duplicate the local address, the local address is compared to that of the corresponding CA, and if the CAN J1939 All-in-One CPU module priority is lower, it repeatedly attempts to acquire another address. When the local address has been determined, messages can be exchanged.
If the local address was not determined by the last address, it is set to 254 (null address).

PGN

CAs exchange parameter information such as the engine RPM and the on/off status of switches. These parameters are defined in advance and assigned unique ID numbers (SPNs). The parameters are also grouped according to content, and these parameter groups are assigned unique ID numbers (PGN).

CAs exchange messages for each PGN.

Example: SPNs that make up PGN 1792 General Purpose Valve Pressure*1

SPN	Parameter Name	Data Size
4086	Valve Load Sense Pressure	2 bytes
4087	Valve Pilot Pressure	1 byte
4088	Valve Assembly Load sense Pressure	2 bytes
4089	Valve Assembly Supply Pressure	2 bytes

*1 From J1939-71

Message

CAs use the data frame in the CAN extended frame format that holds the 29-bit CAN ID to exchange messages. The CAN ID is composed of the priority, PGN, and source address. The PGN is composed of the EDP, DP, PDU Format (PDUF), and PDU Specific (PDUS).

Message

SOF	29 Bits CAN ID	RTR	6 Bits Control	8 Bytes Data	16 Bits CRC	2 Bits ACK	7 Bits EOF	IFS

3 Bits Priority	18 Bits PGN	8 Bits Source Address

1 Bit EDP	1 Bit DP	8 Bits PDU Format	8 Bits PDU Specific

PDU 1 format

When PDUF is 00h to EFh, the PGN is defined in the format called the PDU1 format. The PGN is defined with EDP, DP, and PDUF, and it is handled as a PGN for one-to-one communication. At this time, the destination address is stored in the PDUS. However, when the PDUS is specified as 255 (global address), the PGN is handled as a broadcast PGN, not as one-to-one communication.

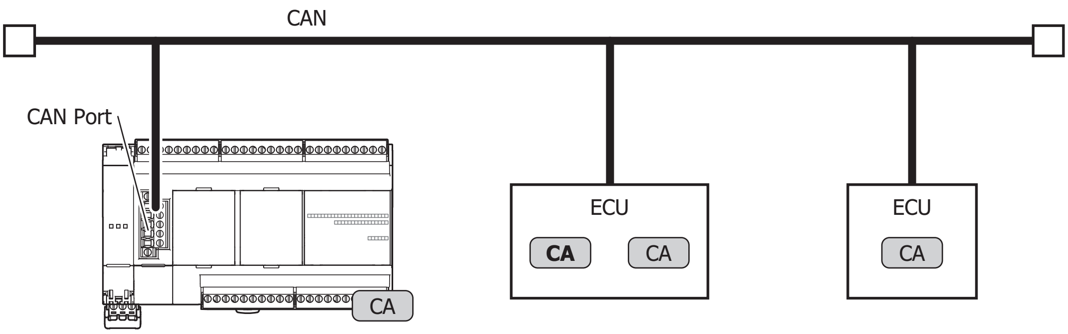
PDU 2 format

When PDUF is F0h to FFh, the PGN is defined in the format called the PDU2 format. The PGN is defined with EDP, DP, PDUF, and PDUS, and it is handled as a PGN for broadcast communication.

Since the data that can be stored in the CAN data frame is a maximum of 8 bytes, when the PGN data is 9 bytes or longer, messages are split into 8-byte packets and exchanged as multi-packet messages.

Overview of J1939 Communication Supported by the All-in-One CPU Module

The CAN J1939 All-in-One CPU module can connect to a J1939 communication network as an ECU that has one CA. It is compatible with the network management function that manages addresses between CAs so address conflicts can be resolved and dynamic addresses can be assigned.



FC6A Series MICROSmart
(CAN J1939 All-in-One CPU Module) (ECU)

Specifications

Item	Description	Comments
Supported SAE J1939	SAE J1939-11: Physical Layer, 250 K bits/s, Twisted Shielded Pair SAE J1939-15: Reduced Physical Layer, 250 K bits/s, Un-Shielded Twisted Pair SAE J1939-21: Data Link Layer SAE J1939-71: Vehicle Application Layer SAE J1939-73: Application Layer - Diagnostics SAE J1939-75: Application Layer - Generator Sets and Industrial SAE J1939-81: Network Management	
Baud Rate	250 k [bps]	
Transmit Methods	Event transmission, cycle transmission	
Receive Methods	Receive in ladder END, with cycle monitoring	
Maximum Number of Nodes	128 nodes	The number of CAs that can be managed with address management
Maximum Number of Send Message	100	
Maximum Number of Received Messages	200	
Maximum Length of Transmitted Message	252 bytes	
Maximum Length of Receive Message	252 bytes	
Network Management	Enabled	Resolves address conflicts, monitors the addresses of neighboring CAs

Special PGNs Handled by the CAN J1939 All-in-One CPU Module

A portion of PGNs are used internally to control the MICROSmart. These PGNs cannot be received, even when they are set as receive messages.

PGNs that cannot be set as receive messages

PGN	Parameter Name	Description
59392 (E800h)	Acknowledgment	Used for responses between CAs.
59904 (EA00h)	Request	Used to request PGN transmission.
60160 (EB00h)	Data Transfer Message	Used to exchange multi-packet messages 9 bytes or longer.
60416 (EC00h)	Connection Management Message	
60928 (EE00h)	Address Claim	Used for address management on the network.

Communication Control

Checking if J1939 communication is permitted/prohibited and checking the communication status can be performed with special internal relays and special data registers.

Note: R/W stands for read/write and allows reading and writing. R is read-only. W is write-only.

Special Internal Relay Allocations

Special Internal Relays	Description	Setting Timing	R/W
M8300	J1939 Communication Permitted Flag		R/W
M8301	J1939 Online Status	Every scan	R
M8302	J1939 Local Station Address Confirmation Status	Every scan	R
M8303	J1939 Communication Error Output	Every scan	R
M8304	J1939 Communication Bus Off Occurrence Output	Every scan	R

J1939 Communication Permitted Flag (M8300)

Controls whether J1939 communication is permitted or prohibited. Turn on M8300 to start J1939 communication after initialization processing has been performed.

OFF: Communication prohibited (default)

ON: Communication permitted

Note: If there is no response from any CAs to a message sent in initialization processing, the transmission will be repeatedly retried and Error Passive (error code 2002) will occur. Turn on M8300 after CAs on the network are able to send and receive data.

J1939 Online Status (M8301)
Indicates the J1939 communication online status. While offline, messages cannot be exchanged because the CAN J1939 All-in-One CPU module is not connected to a J1939 communication network.

OFF: Offline

ON: Online

J1939 Local Station Address Confirmation Status (M8302)
Indicates the local address confirmation status during J1939 communication. This relay turns on when the local address is between 0 and 253. While online, this relay turns off immediately after starting communication and when an address conflict has occurred and the local address is 254 (null address). When M8302 is off, messages cannot be exchanged using the ladder program.

OFF:Local address unconfirmed

ON:Local address confirmed

J1939 Communication Error Output (M8303)
Indicates the status of a J1939 communication error. M8303 turns on when a value other than "0" is stored in D8052 (J1939 Communication Error Code).

OFF: No communication error

ON: Communication error has occurred

J1939 Communication Bus Off Occurrence Output (M8304)
Indicates the bus off status during J1939 communication. Bus off is the status where devices cannot participate in communication on the bus. All transmit and receive operations are prohibited. The CAN J1939 All-in-One CPU module has an internal transmit error counter and receive error counter, and when these counters reach a certain value, the bus off status is set.

When bus off occurs, the CAN J1939 All-in-One CPU module stops communication and goes offline. To restart communication, M8300 must be turned on to go back online. When M8300 is turned on, M8304 is turned off.

OFF: Bus off has not occurred

ON: Bus off has occurred

Special Data Register Allocations

Special Data Register	Function	Setting Timing	R/W
D8052	J1939 Communication Error Code	When error occurred	R/W

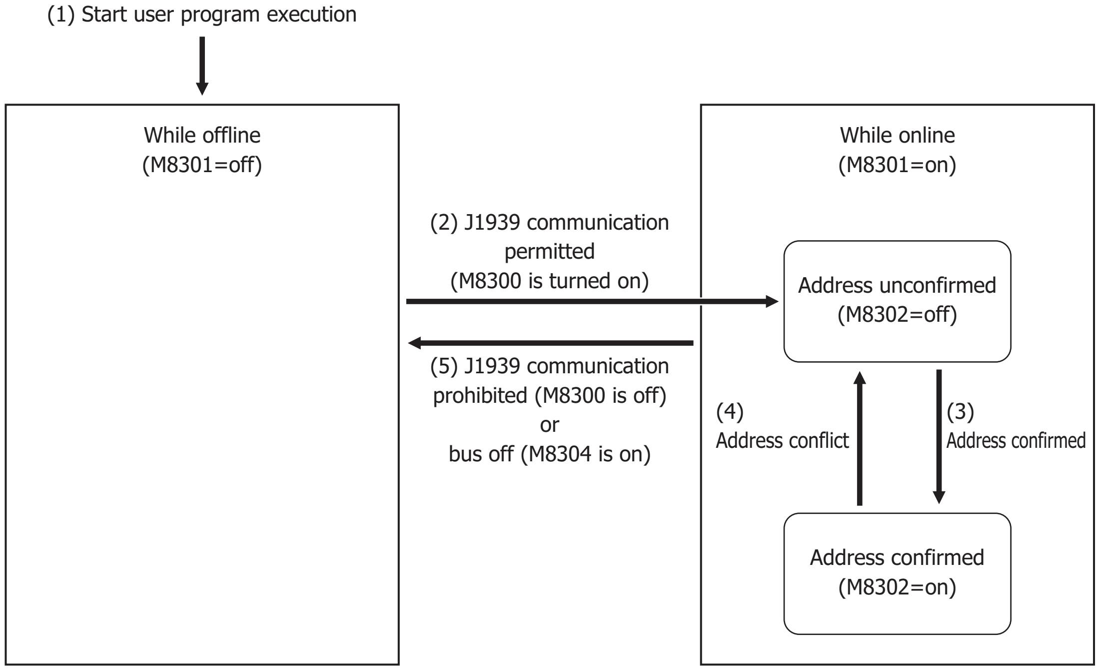
J1939 Communication Error Code (D8052)
This register stores the J1939 communication error code. To initialize this register, write "0".

Details on the J1939 communication error code are as follows.

Status Code	Description	Details
0	Normal	
100	Multi-packet transmit/receive error	An unexpected BAM frame was received.
101	Multi-packet transmit/receive error	An unexpected RTS frame was received.
102	Multi-packet transmit/receive error	An unexpected CTS frame was received.
103	Multi-packet transmit/receive error	An unexpected EOM frame was received.
104	Multi-packet transmit/receive error	An unexpected Connection Abort frame was received.
105	Multi-packet transmit/receive error	An unexpected DT frame was received.
106	Multi-packet transmit/receive error	The length of the multi-packet transmitted/received message was out of range.
107	Multi-packet transmit/receive error	Failed to transmit the Connection Abort message.
110	NACK transmission failure	NACK could not be transmitted for a PGN transmit request.
200	Internal transmit queue overflow	The limit of messages that can be sent at one time has been exceeded.
201	Internal receive queue overflow	The limit of messages that can be received at one time has been exceeded.
1000	Local address unconfirmed	The local address is 254.
2000	CAN communication initialization error	The CAN controller could not be initialized.
2001	CAN communication initialization error	The CAN controller could not be reset.
2002	Error Passive	An error/warning interrupt has occurred, and the CAN controller has entered the Error Passive state.
2003	Error Active	The CAN controller has entered the Error Active state.
2004	Data overrun error	The data overrun interrupt occurred.
3000	Bus off error	Bus off occurred on the CAN.

Communication Control Status Transitions

J1939 communication status and initialization processing transitions are as follows.



(1) The offline status is set immediately after starting execution of the user program.

(2) When M8300 is turned on, the online status is set and initialization processing is performed.

In initialization processing, initialization messages are sent to CAs, duplicate addresses are checked, and conflicts are resolved for those duplicate addresses.

Before confirming the local address in initialization processing, the address is the address unconfirmed status (M8302=off).

(3) When a local address is confirmed between 0 and 253, the status is the address confirmed status (M8302=on).

Messages can be exchanged between the MICROSmart and CAs. If the local address is 254, the status remains the address unconfirmed status.

(4) When an address conflict occurs and the local address is 254 (null address), the status is the address unconfirmed status.

(5) When M8300 is turned off, communication processing is stopped and the offline status is set.

The offline status is also set when bus off occurs during communication (M8304 is turned on). M8300 remains off at this time.

To set the online status again, turn off M8300 and then turn it on again.

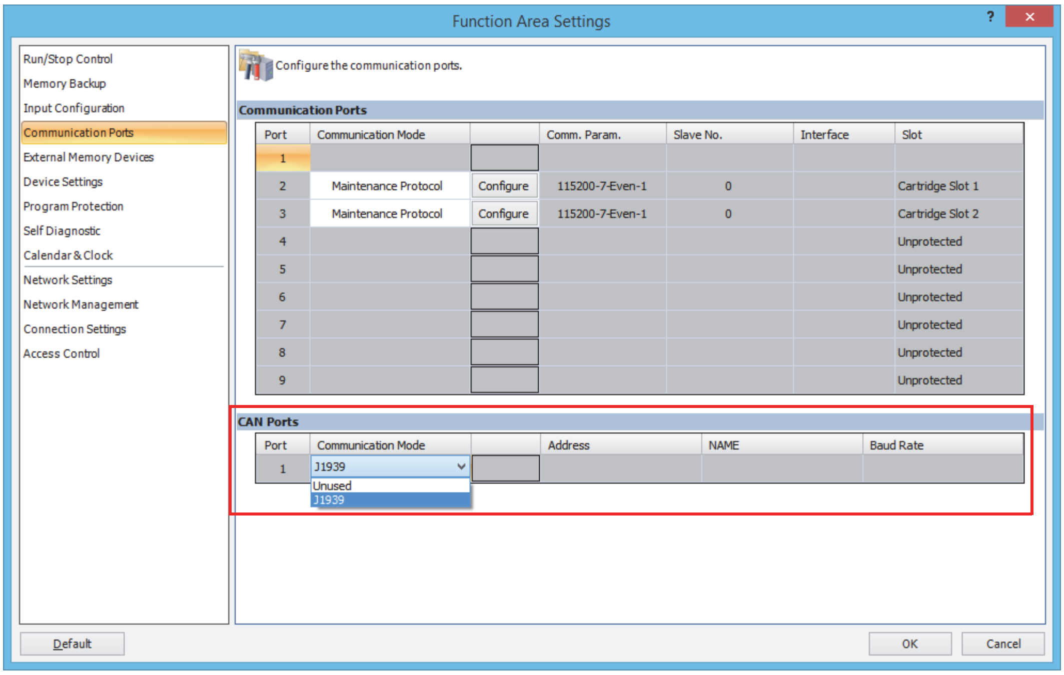
J1939 Communication Settings

This section describes the operation procedure and details about the items related to J1939 communication.

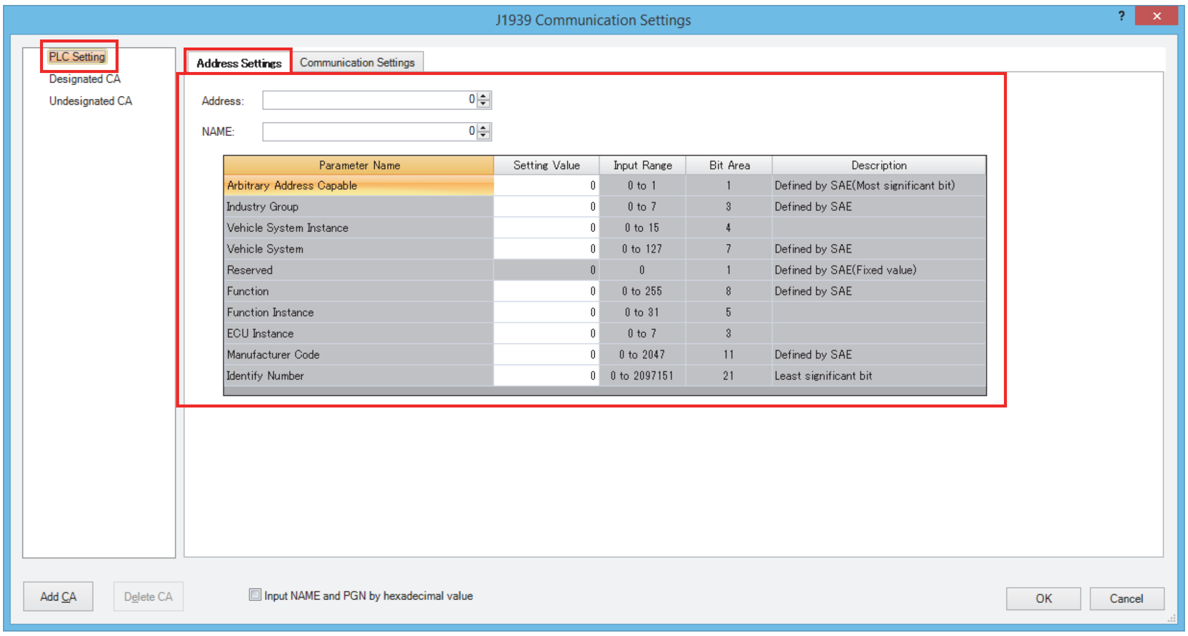
Programming WindLDR

Configure J1939 communication according to the usage environment.

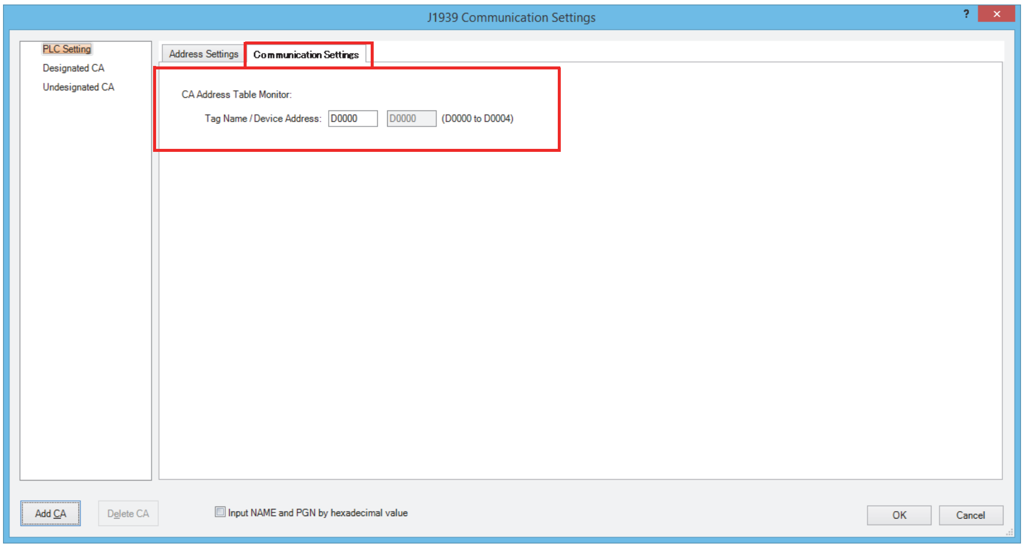
1. On the **Configuration** tab, in the **Function Area Settings** group, click **Comm. Port.**
   The **Function Area Settings** dialog box is displayed.
2. Select **J1939** from the CAN port group communication mode.
   The **J1939 Communication Settings** dialog box is displayed.



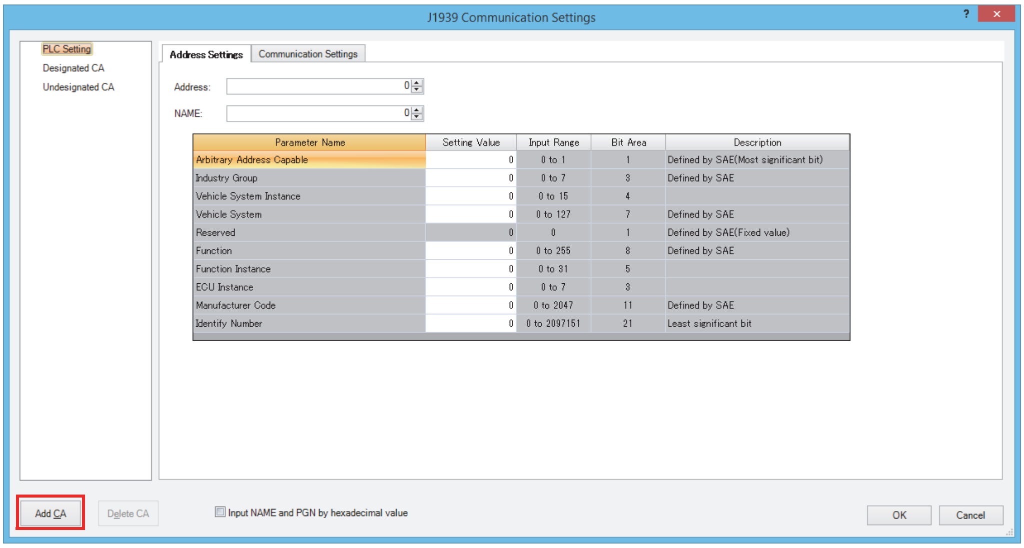
3. Under **PLC Setting**, on the **Address Settings** tab, set the local address, device name (NAME), and CA address table.
   Set the local address and device name (NAME).



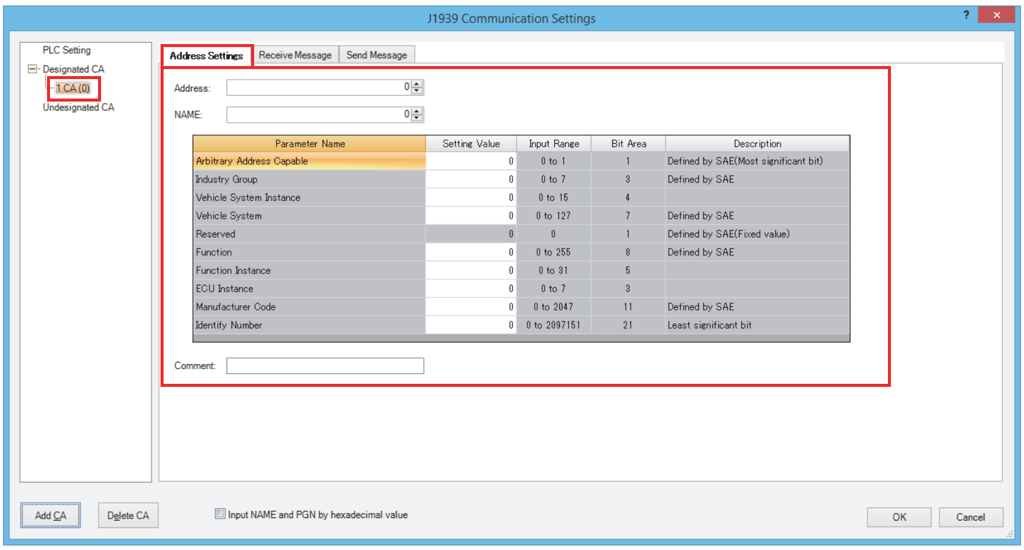
4. Click the **Communication Settings** tab and set the device address to store the CA address table.



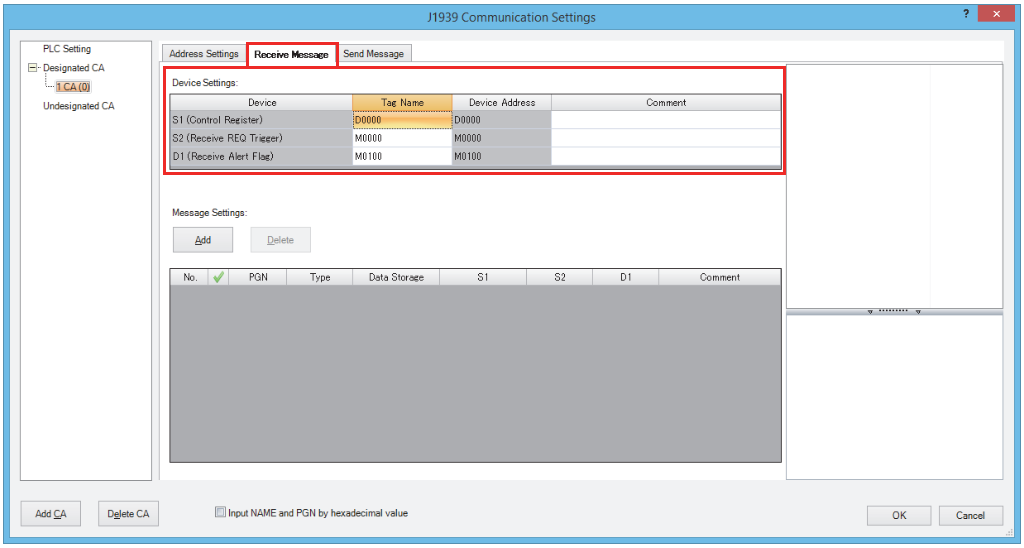
5. Click **Add CA** to add a node to configure a CA.



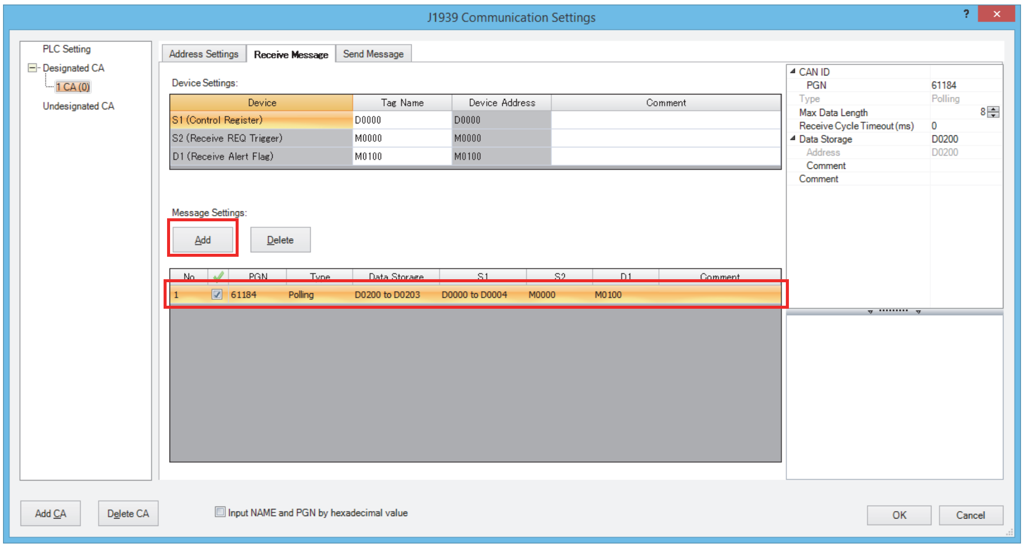
6. Click the node that corresponds to the CA to configure and click the **Address Settings** tab. Configure the address and the device name for the CA.



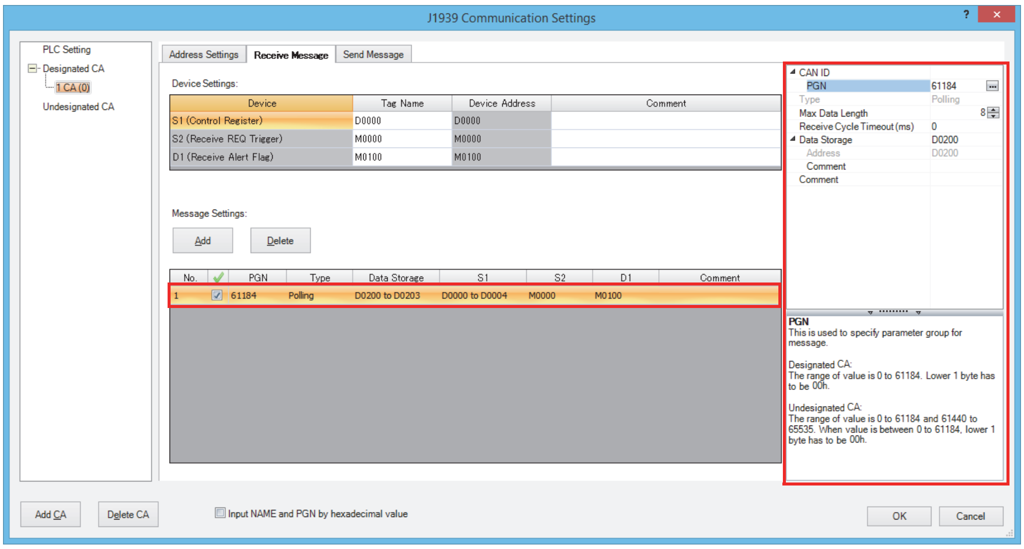
7. Click the **Receive Message** tab and first configure the devices that will used with received messages.



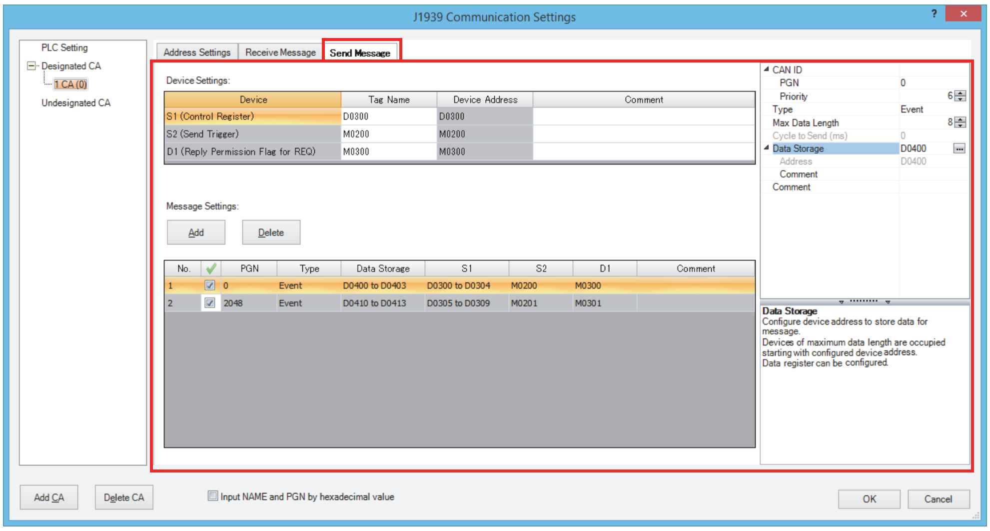
8. Click **Add** to add a received message.



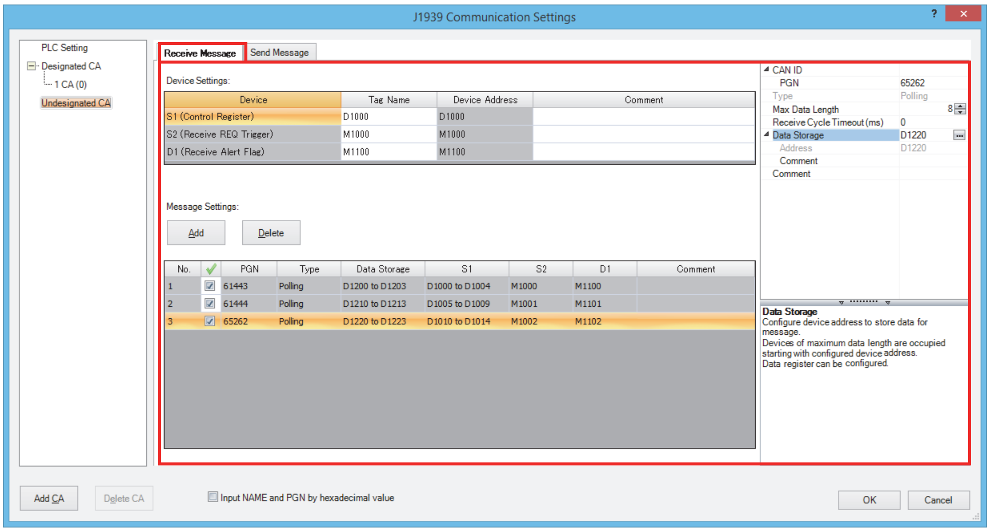
9. Select any of the cells for the message settings to display the parameters for the messages on the right side. Configure the items.

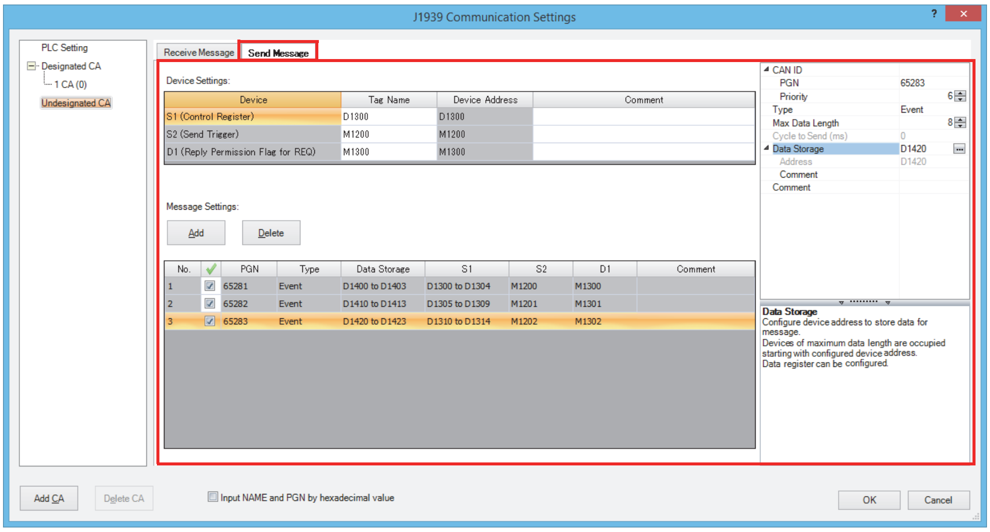


10. Click the **Send Message** tab and configure it in the same manner as the **Receive Message** tab.



11. Click **Undesignated CA**. Configure the **Receive Message** tab and the **Send Message** tab in the same manner as those under **Designated CA**.





12. Click **OK**.

This concludes configuring the settings.

J1939 Communication Parameters

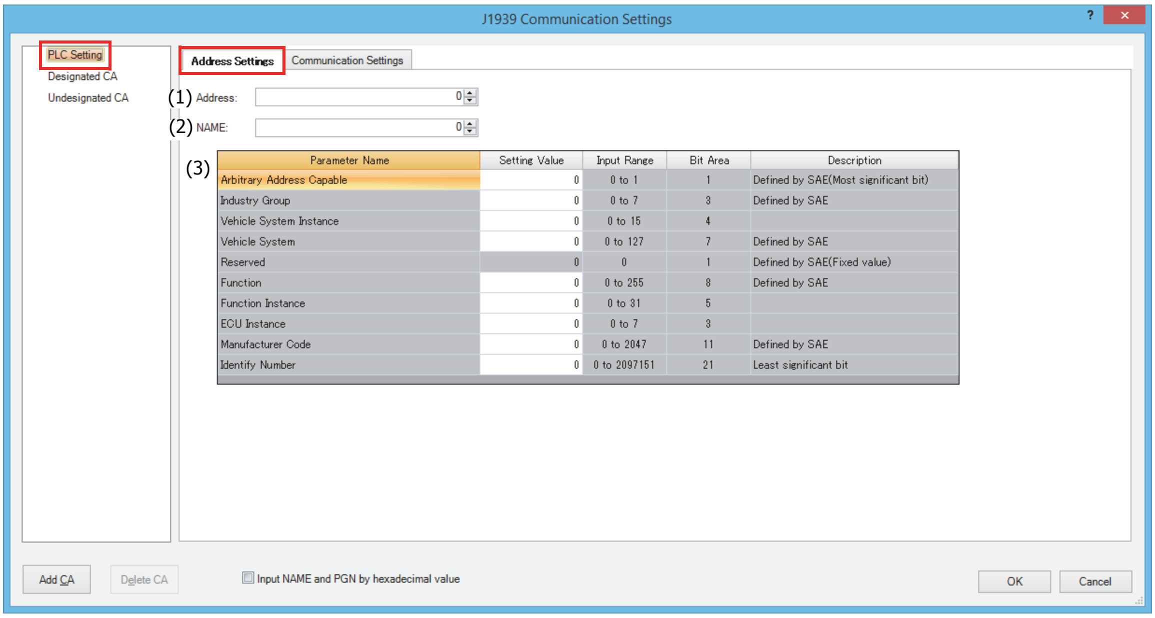
This section describes the **J1939 Settings** dialog box parameter settings required to use J1939 communication.

PLC Setting

These settings configure the CAN J1939 All-in-One CPU module local address, local NAME, and CA address table.

■ Address Settings tab

This tab configures the CAN J1939 All-in-One CPU module local address and local NAME.



(1) Address

Sets the initial value of the CAN J1939 All-in-One CPU module local address between 0 and 253.
When **Input NAME and PGN by hexadecimal value** check box is selected, enter them as hexadecimal values.
Example: When hexadecimal: 01, F0, etc.
When an address conflict occurs with a CA connected to the same network, the local address is reacquired according to the Arbitrary Address Capable bit of the local NAME.

(2) NAME

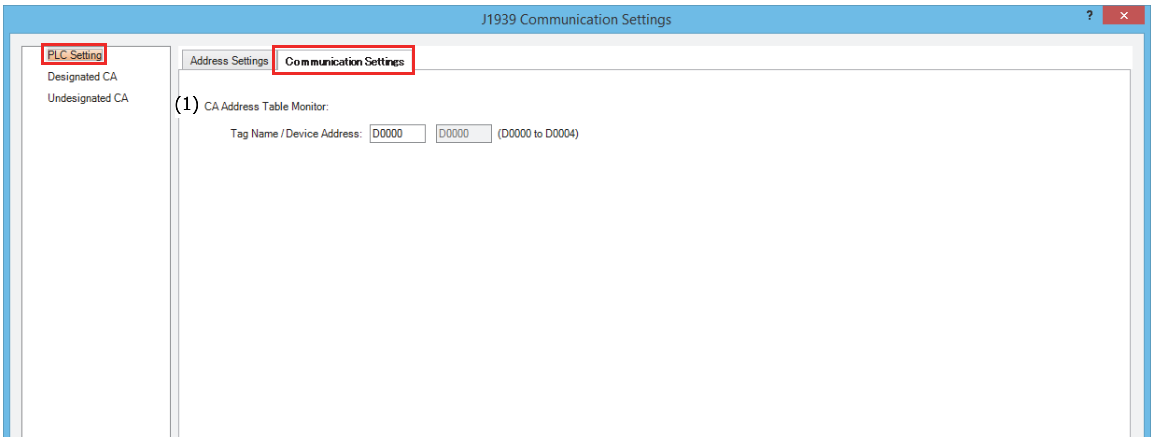
Sets the CAN J1939 All-in-One CPU module local NAME in 64 bits.
When **Input NAME and PGN by hexadecimal value** check box is selected, enter them as hexadecimal values.
Example: When hexadecimal: 123456789ABCDEF1, 0000FFFFFFFFFF0A, etc.
Set this value so that it is not duplicated by other CAs on the same network. For details, see "NAME (64 bits total)" on page 8-2.
The device name (NAME) is composed of the set values in the device name field list (3). When the device name (NAME) is changed, those changes are reflected under **Setting Value** in the device name field list.

(3) Device name field list

Displays each field of the device name (NAME) (2). When a **Setting Value** is changed, the result is reflected in the device name (NAME).

■ **Communication Settings tab**

This tab configures the CA address table monitor. The CA address table monitor is the table that manages the CA addresses registered by the CAN J1939 All-in-One CPU module (local) and **Designated CA**.



(1) CA Address Table Monitor

Sets the starting address of the data registers that will store the CA address table.

The values set in WindLDR are stored in the CA address table as the initial values, and when the CAN J1939 All-in-One CPU module address changes due to an address conflict, the CA address table is updated with that. When address information is received from a CA, the address of the CA that matches the NAME is updated. The CAN J1939 All-in-One CPU module exchanges messages based on the addresses in the CA address table.

CA Address Table Monitor Definition

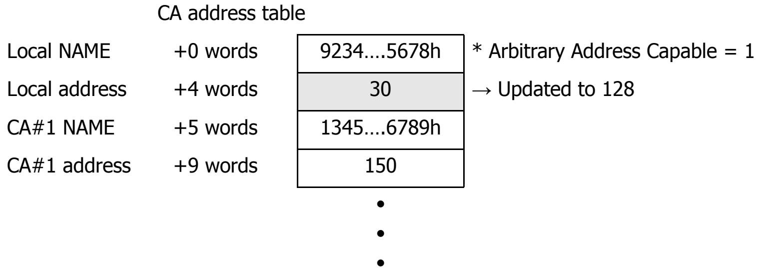
CA	Storage Destination	Function	Setting Details
Local	Starting number+0	Local NAME	Stores the CAN J1939 All-in-One CPU module NAME.
	Starting number+1		
	Starting number+2		
	Starting number+3		
	Starting number+4	Local address[*1]	Stores the CAN J1939 All-in-One CPU module address. If the address changes while online, exchanging messages after that is performed based on the new address.
CA#1 (5 words)	Starting number+5	CA#1 NAME[*2]	Stores the CA#1 NAME.
	Starting number+6		
	Starting number+7		
	Starting number+8		
	Starting number+9	CA#1 address[*2*3]	Stores the CA#1 address. When NAME has been set, the address may change when online. If the address changes while online, exchanging messages after that is performed based on the new address.
⋮	⋮	⋮	
CA#N[*3] (5 words)	Starting number+5xN	CA#N NAME[*2]	Stores the CA#N NAME.
	Starting number+5xN+1		
	Starting number+5xN+2		
	Starting number+5xN+3		
	Starting number+5xN+4	CA#N address[*2*3]	Stores the CA#N address. When NAME has been set, the address may change when online. If the address changes while online, exchanging messages after that is performed based on the new address.

*1 If an address cannot be assigned due to address conflicts with CAs on the network, the address is 254 (null address). In this case, messages cannot exchanged.

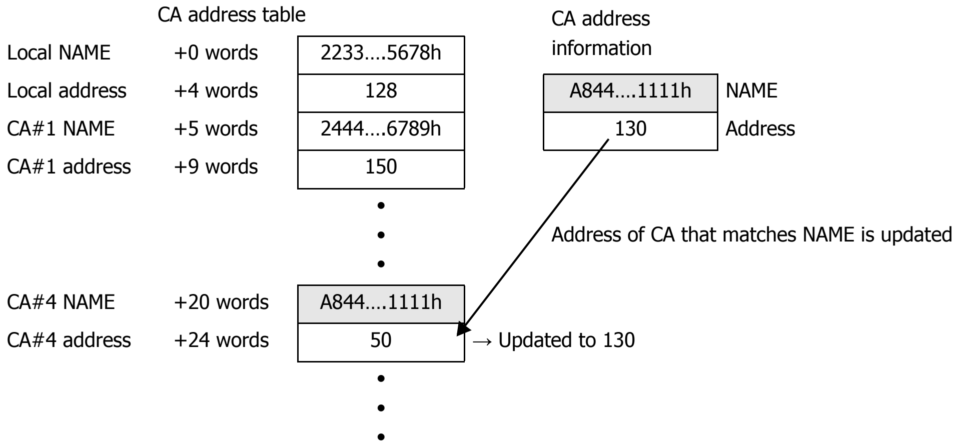
*2 If the address is outside the range of 0 to 253, messages cannot be exchanged with the corresponding CA.

*3 N is 1 to 253.

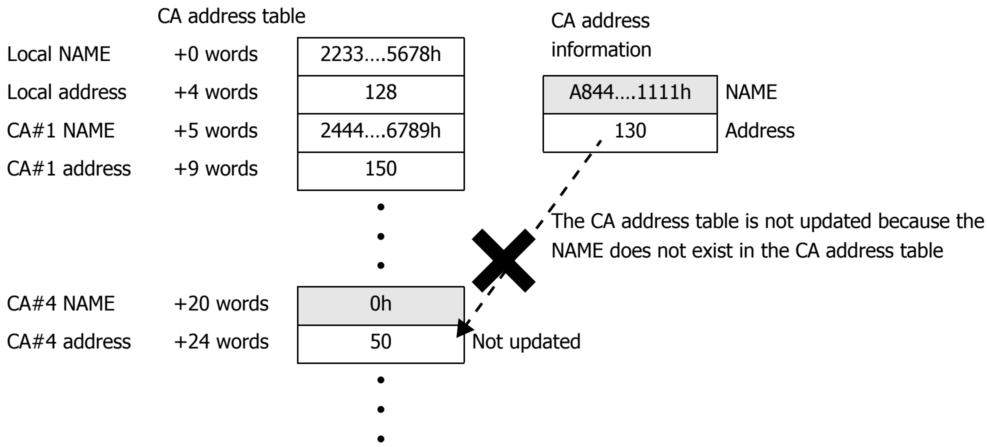
Example 1: When updated by local address conflict



Example 2: When updated by receiving CA address information



Example 3: When CA#4 NAME is set to 0 in Example 2

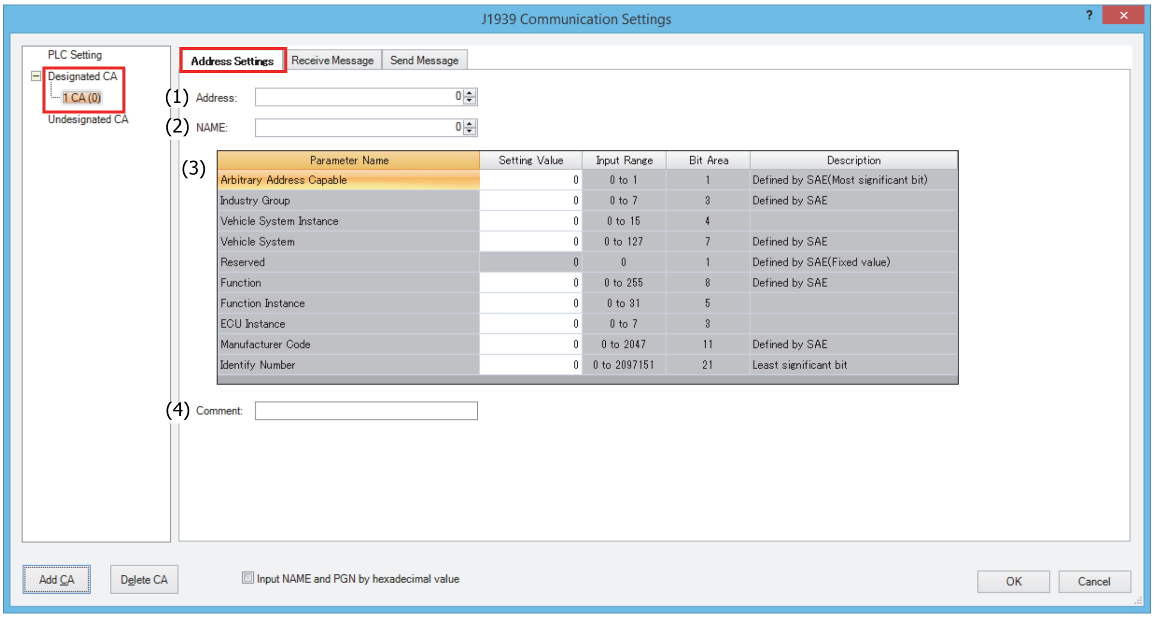


Designated CA

For each added CA, these settings configure the address and NAME of the CA to exchange messages with and the transmitted and received messages.

■ Address Settings tab

This tab configures the address and NAME of the CA selected on the CA list.



(1) Address

Sets the address of the CA to exchange messages with from between 0 and 254.
When **Input NAME and PGN by hexadecimal value** check box is selected, enter them as hexadecimal values.
Example: When hexadecimal: 01, F0, etc.

(2) NAME

Sets the CA NAME in 64 bits. If CA NAME is not defined, set it to 0.
When **Input NAME and PGN by hexadecimal value** check box is selected, enter them as hexadecimal values.
Example: When hexadecimal: 123456789ABCDEF1, 0000FFFFFFFFFF0A, etc.
The device name (NAME) is composed of the set values in the device name field list (3). When the device name (NAME) is changed, those changes are reflected under **Setting Value** in the device name field list.

(3) Device name field list

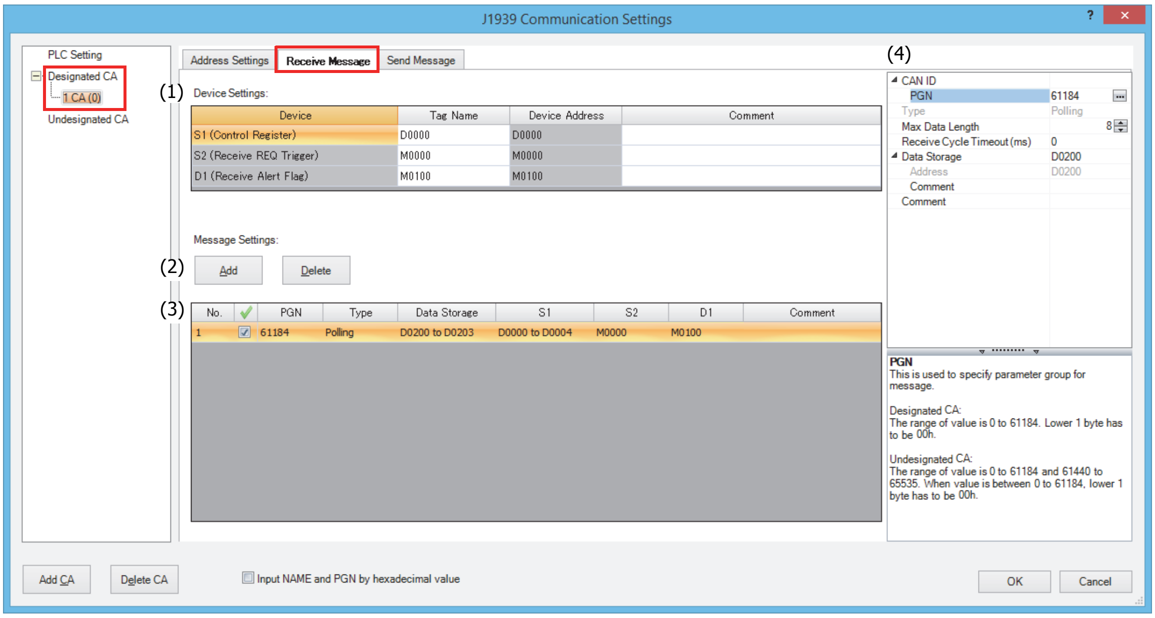
Displays each field of the device name (NAME) (2). When a **Setting Value** is changed, the result is reflected in the device name (NAME).

(4) Comment

Sets a comment for the CA as a maximum of 127 characters. This is reflected in the CA list under **Designated CA**.

■ Receive Message tab

This tab configures the messages that the CAN J1939 All-in-One CPU module receives from the CA selected in the CA list.



(1) Device Settings

Sets the data registers that will be used by the received messages.

Settings	Description
Tag Name	Specifies the tag name of the device or the device address.
Device Address	Shows the device address that corresponds to the tag name.
Comment	Shows the comment for the device address. This item can be edited.

S1 (Control Register)

Sets the starting address of the data registers that will store the receive results.

Storage Destination	Item		Description
Starting number+0	Received data length		This register stores the length of the received message data. If the message that will be received exceeds 8 bytes, a multi-packet message split into 8-byte packets will be received. After the multi-packet message has finished being received, the split data is combined and stored in the data storage destination. If the length of the received message data is longer than the maximum data length, the maximum data length is stored and 100 is stored in the receive results status.
Starting number+1	**Reserved**		
Starting number+2	Receive results status		This register stores the receive results. To initialize this register, write "0". For details on the receive results status, see "Receive results status" on page 8-16.
Starting number+3	Receive action flag		Stores the action flag for receiving.
	Bit 0	Receive buffer overwrite flag	When the internal receive buffer was overwritten by the next message before the message data stored in the receive buffer could be written to the data storage destination, this register turns on for one scan.
	Bits 1 to 15	**Reserved**	
Starting number+4	**Reserved**		

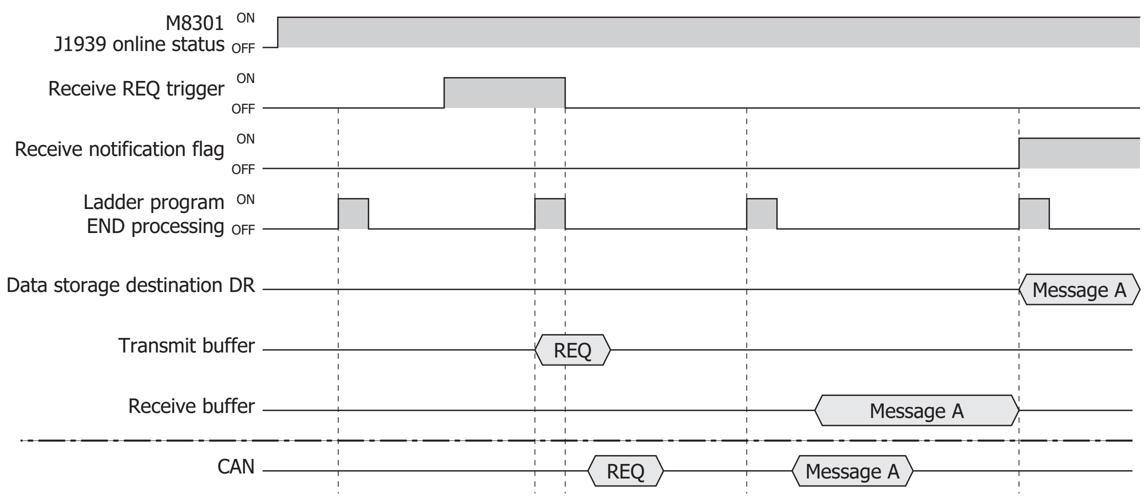
Receive results status

Status Code	Description	Details
0	Normal	
10	Receive cycle timeout	The set time for the receive cycle timeout has been exceeded since the previous message was received.
100	Received data length error	The length of the received message data was longer than the maximum data length.
101	Received data length error	The received message was 252 bytes or longer.
102	Multi-packet receive error	A message with a data length of 9 bytes or longer could not be received.

S2 (Receive REQ Trigger)

This function issues a request to the corresponding CA to transmit a message. When the receive REQ trigger is turned on, PGN 59904 (Request) will be sent to the CA in ladder program END processing.

Receive REQ trigger usage example



D1 (Receive Alert Flag)

Sets the flag for notification of a received message. This register is on for one scan when a message is received.

(2) Add button/Delete button

Click **Add** to add a received message to the message list (3).
To delete a message, select the message to delete in the message list, and then click **Delete**.

(3) Message list

Displays the content configured by the message parameter settings (4). The received message can also be enabled or disabled. The message is enabled when the check box is selected and disabled when the check box is cleared.

Notes:

Click the check box column header to perform the following operations.

- If all received message check boxes are on, all of the check boxes are cleared.
- If any received message check boxes are off, all of the check boxes are selected.

(4) Message parameter settings

These settings configure the details of the received message selected in the message list.

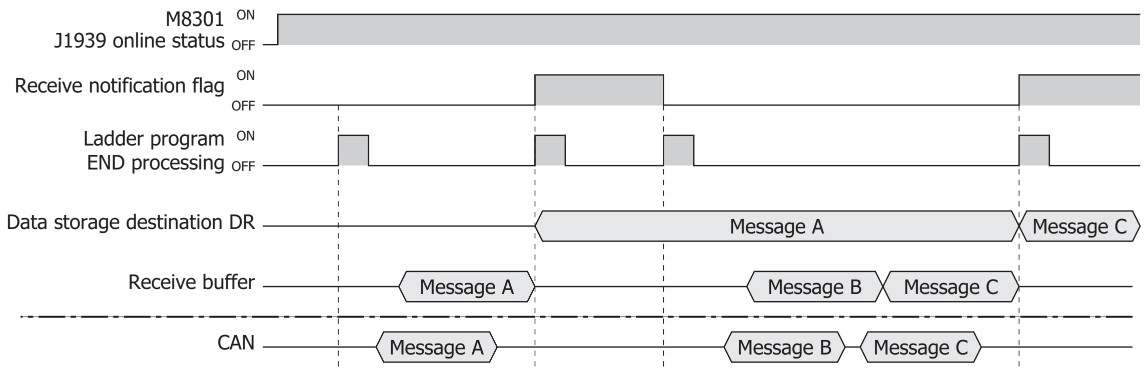
CAN ID

PGN: Sets the PGN of the message to receive. The range of the PGN that can be set is 0 to 61,184 (0000h to EF00h) (PDU1 format). Set the low order 1 byte (PDU Specific) that indicates the destination address in PDU1 format to 00h.

Click **...** and a PGN can be selected in the displayed PGN Manager. For details, see "PGN Manager" on page 8-29.

Type
The receive method for messages is fixed as "Polling".
Received message are temporarily saved to the internal receive buffer, and the contents of the receive buffer are written to the data registers set as the data storage destination in the ladder program END processing. At this time, the receive notification flag is turned on for only one scan.
If the same PGN message is received from the same CA multiple times during the period from the ladder program END processing until the next END processing, the content of the last received message is written to the data storage destination.

Polling reception example



Max Data Length
Sets the maximum data length of the PGN to receive. The maximum value that can be set is 252 bytes.

Receive Cycle Timeout (ms)
Sets the time to monitor the receive message cycle that periodically receives messages in 10 ms increments. The range that can be set is 0 to 655,350. If set to 0, the receive cycle is not monitored.

Data Storage
Sets the data registers that will store the received message. The maximum data length is used starting from the set device.

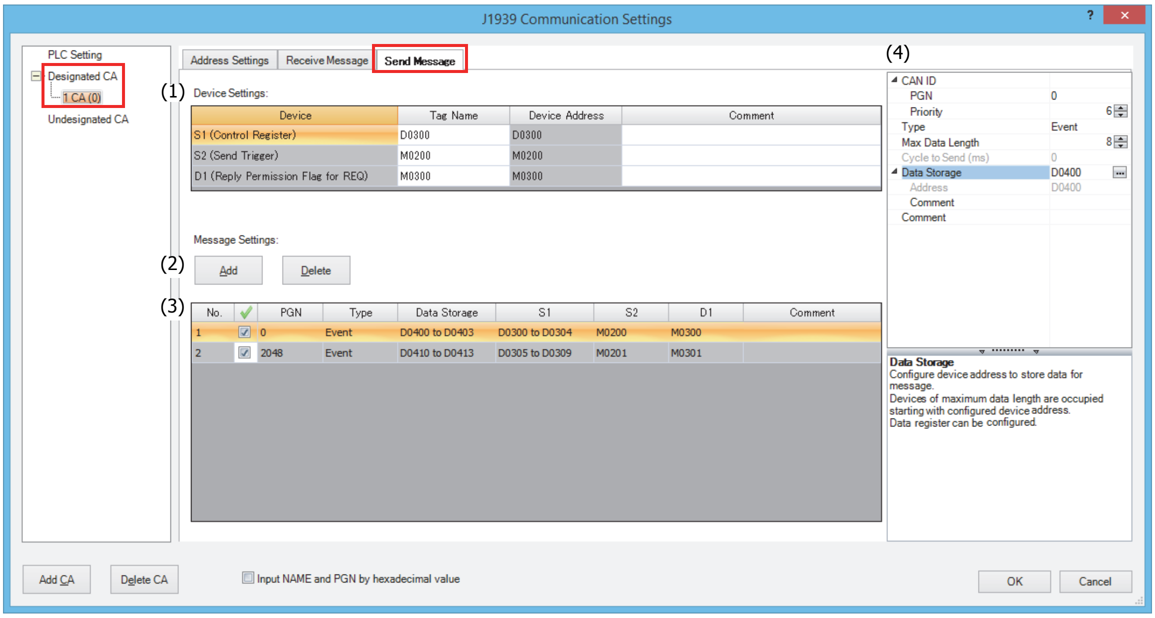
Address: Shows the device address that corresponds to the tag name of the device or the device address.

Comment: Shows the comment that corresponds to the tag name of the device or the device address. This item can be edited.

Comment
Sets the comment for the received message. The comment that can be set is a maximum of 127 bytes.

■ Send Message tab

This tab configures the messages that the CAN J1939 All-in-One CPU module transmits to the CA selected in the CA list.



(1) Device Settings

Sets the data registers that will be used by the sent messages.

Settings	Description
Tag name	Specifies the tag name of the device or the device address.
Device Address	Shows the device address that corresponds to the tag name.
Comment	Shows the comment for the device address. This item can be edited.

S1 (Control Register)

Sets the starting address of the data registers that will store the transmit settings and transmit results.

Storage Destination	Item		Description
Starting number+0	Transmit data length		This register stores the length of the message data to transmit. Starting from the data storage destination, data in the length of the transmit data length will be transmitted to the CA. When 0 is stored for the transmit data length, data of the maximum data length will be transmitted to the CA. This cannot be set larger than the maximum data length. If the message that will be transmitted exceeds 8 bytes, the message is split into 8-byte packets and transmitted as a multi-packet message.
Starting number+1	**Reserved**		
Starting number+2	Transmit results status		This register stores the transmit results. To initialize this register, write "0". For details on the transmit results status, see "Transmit results status" on page 8-19.
Starting number+3	Transmit action flag		Stores the action flag for transmitting.
	Bit 0	REQ received flag	This flag is turned on when a request PGN from a CA was received. This flag is turned off when the corresponding PGN is transmitted.
	Bits 1 to 15	**Reserved**	
Starting number+4	**Reserved**		

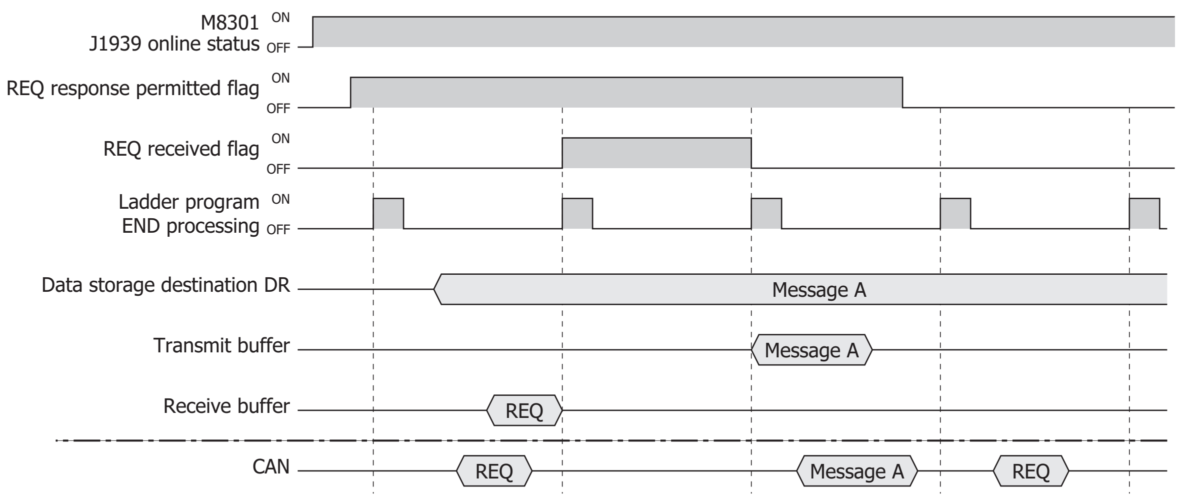
Transmit results status

Status Code	Description	Details
0	Normal	
1	Multi-packet message transmitted	A message with a data length of 9 bytes or longer has finished being transmitted.
10	Transmit data length error	The length of the transmitted message data was longer than the maximum data length.
20	Internal transmit queue overflow	The limit on the number of messages that can be queued for transmission has been reached.
100	Multi-packet transmit error	A message with a data length of 9 bytes or longer could not be transmitted.
200	Local address unconfirmed	The message could not be transmitted because the local address is unconfirmed.
201	Destination address unconfirmed	The message could not be transmitted because the destination CA address is unconfirmed.

S2 (Send Trigger)
Sets the transmit trigger when "Event transmission" is selected for **Type**. When the transmit trigger is turned on, the message is transmitted in ladder program END processing. After the message is transmitted, the transmit trigger is turned off.

D1 (Reply Permission Flag for REQ)
This function transmits a PGN in response to a transmit request PGN received from the corresponding CA while the REQ response permitted flag is on. When PGN 59904 (Request) is received from a CA, the transmit action flag's REQ received flag is turned on, and in the next END processing a message for the corresponding PGN will be transmitted.

Request response example



(2) Add button/Delete button
Click **Add** to add a transmitted message to the message list (3).
To delete a message, select the message to delete in the message list, and then click **Delete**.

(3) Message list
Displays the content configured by the message parameter settings (4). The send message can also be enabled or disabled. The message is enabled when the check box is selected and disabled when the check box is cleared.

Note:
Click the check box column header to perform the following operations.
- If all transmitted message check boxes are on, all of the check boxes are cleared.
- If any transmitted message check boxes are off, all of the check boxes are selected.

(4) Message parameter settings

These settings configure the details of the transmitted message selected in the message list.

CAN ID

PGN: Sets the PGN of the message to transmit. The range of the PGN that can be set is 0 to 61,184 (0000h to EF00h) (PDU1 format). Set the low order 1 byte (PDU Specific) that indicates the destination address in PDU1 format to 00h.

Click **...** and a PGN can be selected in the displayed PGN Manager. For details, see "PGN Manager" on page 8-29.

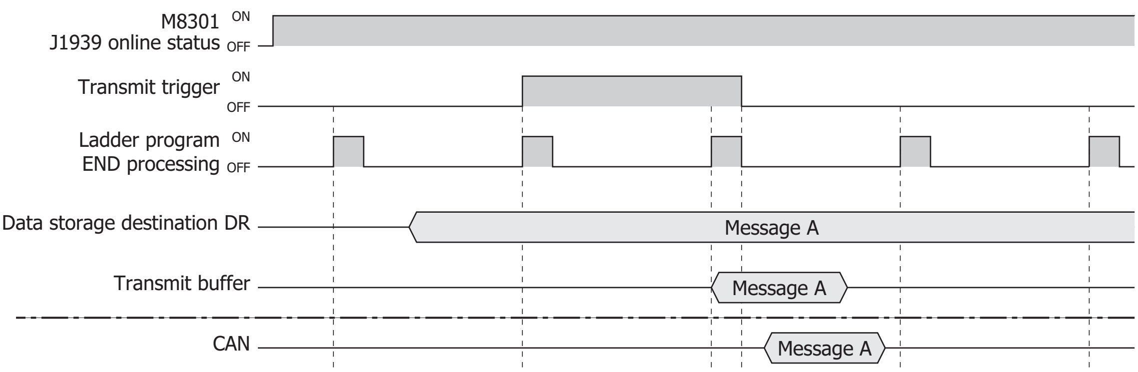
Priority: Sets the priority for the message to transmit between 0 and 7.

Type

Sets the transmit method for the message. Select as "Event" or "Cycle".

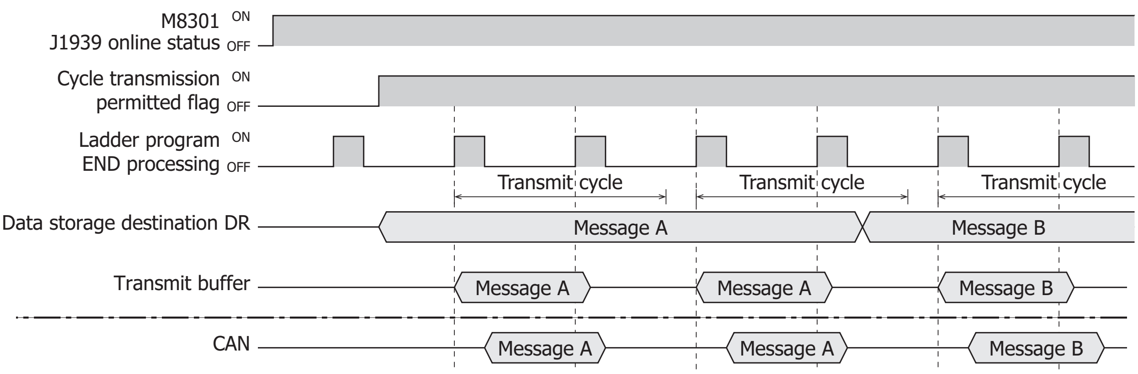
Event: When the transmit trigger is turned on, the transmit data stored in the data storage destination is transmitted in END Processing.

Designated CA event transmission example



Cycle: While the cycle transmission permitted flag is on and when the transmit cycle occurs, the transmit data stored in the data storage destination is transmitted in END processing. The actual transmit cycle is affected by the ladder program scan time.

Designated CA cycle transmission example



Max Data Length

Sets the maximum data length of the PGN to transmit. The maximum value that can be set is 252 bytes.

Cycle to Send (ms)

Sets the transmit cycle in 10 ms increments when "Cycle transmission" is selected for **Type**. The range of the value that can be set is 10 to 655,350.

Data Storage
Sets the data registers that will store the transmitted message. The maximum data length is used starting from the set device. When transmitting a message, the data is first copied from the data storage destination to the internal transmit buffer and then transmitted.

Address: Shows the device address that corresponds to the tag name of the device or the device address.

Comment: Shows the comment that corresponds to the tag name of the device or the device address. This item can be edited.

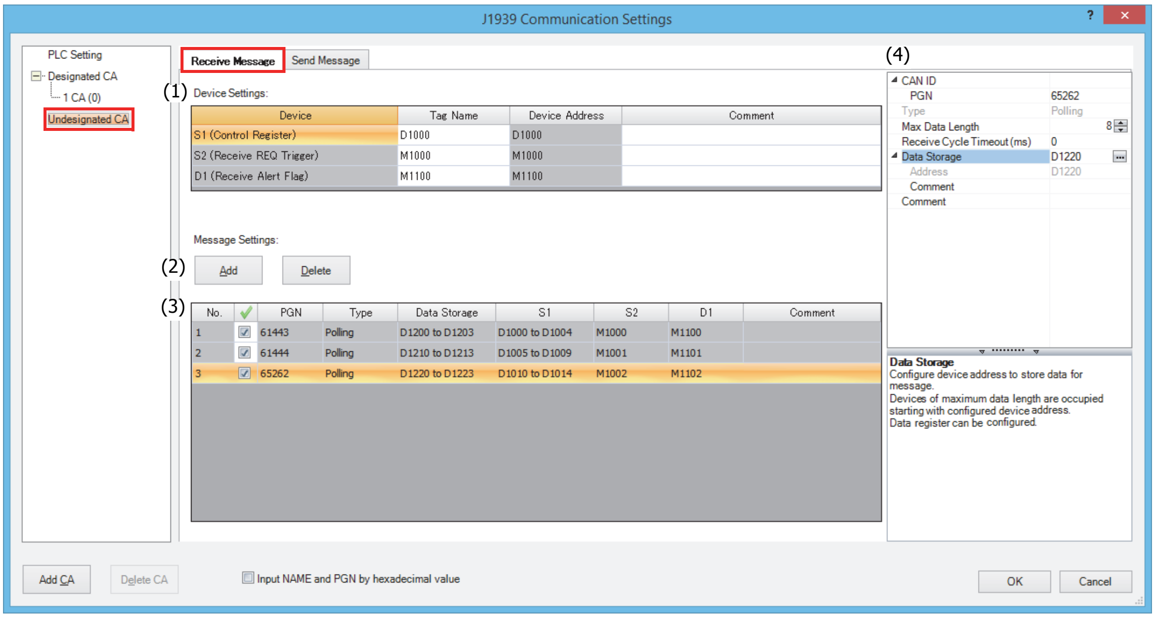
Comment
Sets the comment for the transmitted message. The comment that can be set is a maximum of 127 bytes.

Undesignated CA

These settings configure messages that are exchanged as broadcasts without specifying specific CAs.

■ Receive Message tab

This tab configures messages that are transmitted from other CAs to the J1939 communication network as broadcasts.



(1) Device Settings

Sets the data registers that will be used by the received messages.

Settings	Description
Tag name	Specifies the tag name of the device or the device address.
Device address	Shows the device address that corresponds to the tag name.
Comment	Shows the comment for the device address. This item can be edited.

S1 (Control Register)

Sets the starting address of the data registers that will store the receive results.

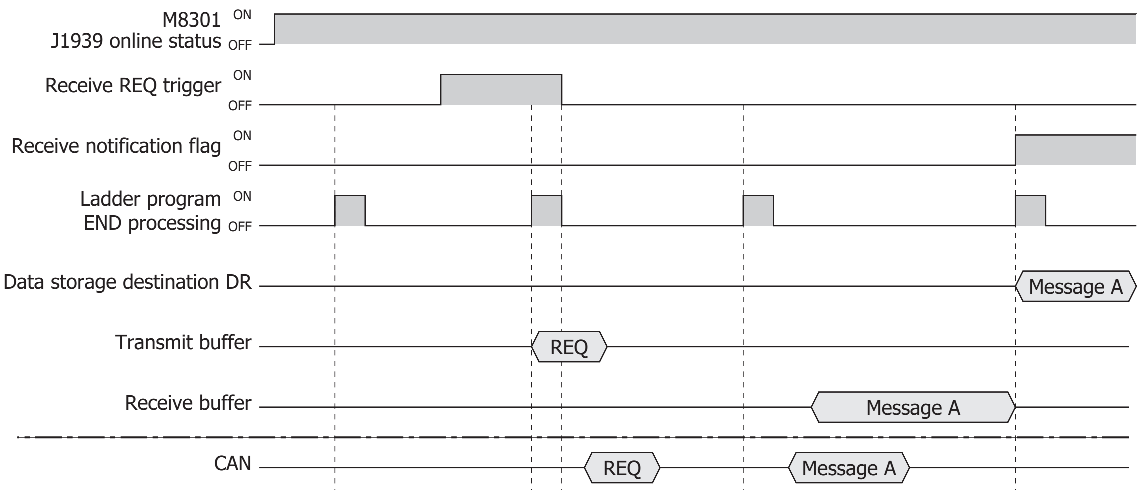
Storage Destination	Item		Description
Starting number+0	Received data length		This register stores the length of the received message data. If the message that will be received exceeds 8 bytes, a multi-packet message split into 8-byte packets will be received. After the multi-packet message has finished being received, the split data is combined and stored in the data storage destination. If the length of the received message data is longer than the maximum data length, the maximum data length is stored and 100 is stored in the receive results status.
Starting number+1	Source address		This register stores the source address of the received message.
Starting number+2	Receive results status		This register stores the receive results. To initialize this register, write "0". For details on the receive results status, see "Receive results status" on page 8-23.
Starting number+3	Receive action flag		Stores the action flag for receiving.
	Bit 0	Receive buffer overwrite flag	When the internal receive buffer was overwritten by the next message before the message data stored in the receive buffer could be written to the data storage destination, this register turns on for one scan.
	Bits 1 to 15	**Reserved**	
Starting number+4	**Reserved**		

Receive results status

Status Code	Description	Details
0	Normal	
10	Receive cycle timeout	The set time for the receive cycle timeout has been exceeded since the previous message was received.
100	Received data length error	The length of the received message data was longer than the maximum data length.
101	Received data length error	The received message was 252 bytes or longer.
102	Multi-packet receive error	A message with a data length of 9 bytes or longer could not be received.

S2 (Receive REQ Trigger)
This function issues a PGN transmit request as a broadcast to the J1939 communication network. If the receive REQ trigger is turned on, PGN 599034 (Request) is transmitted as a broadcast in ladder program END processing.

Receive REQ trigger usage example



D1 (Receive Alert Flag)
Sets the flag for notification of a received message. This register is on for one scan when a message is received.

(2) Add button/Delete button
Click **Add** to add a received message row to the message list (3).
To delete a message, select the message to delete in the message list, and then click **Delete**.

(3) Message list
Displays the content configured by the message parameter settings (4). The received message can also be enabled or disabled. The message is enabled when the check box is selected and disabled when the check box is cleared.

Notes:
Click the check box column header to perform the following operations.

- If all received message check boxes are on, all of the check boxes are cleared.
- If any received message check boxes are off, all of the check boxes are selected.

(4) Message parameter settings
These settings configure the details of the received message selected in the message list.

CAN ID

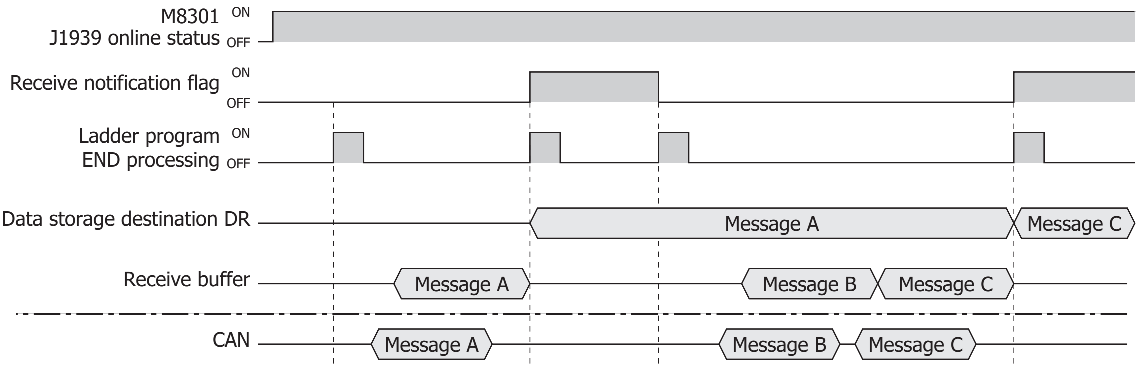
PGN: Sets the PGN of the message to receive in 2 bytes. The range of the PGN that can be set is 0 to 61,184 (0000h to EF00h) (PDU1 format) and 61,440 to 65,535 (F000h to FFFFh) (PDU2 format). Set the low order 1 byte (PDU Specific) that indicates the destination address in PDU1 format to 00h.

When the PGN is set in the PDU1 format range, only messages with a destination address of 255 (global address) can be received.

Click **...** and a PGN can be selected in the displayed PGN Manager. For details, see "PGN Manager" on page 8-29.

Type
The receive method for messages is fixed as "Polling".
Received message are temporarily saved to the internal receive buffer, and the contents of the receive buffer are written to the data registers set as the data storage destination in the ladder program END processing. At this time, the receive notification flag is turned on for only one scan.
If the same PGN message is received from the same CA multiple times during the period from the ladder program END processing until the next END processing, the content of the last received message is written to the data storage destination.

Polling reception example



Max Data Length
Sets the maximum data length of the PGN to receive. The maximum value that can be set is 252 bytes.

Receive Cycle Timeout (ms)
Sets the time to monitor the receive message cycle that periodically receives messages in 10 ms increments. The range that can be set is 0 to 655,350. If set to 0, the receive cycle is not monitored.

Data Storage
Sets the data registers that will store the received message. The maximum data length is used starting from the set device.

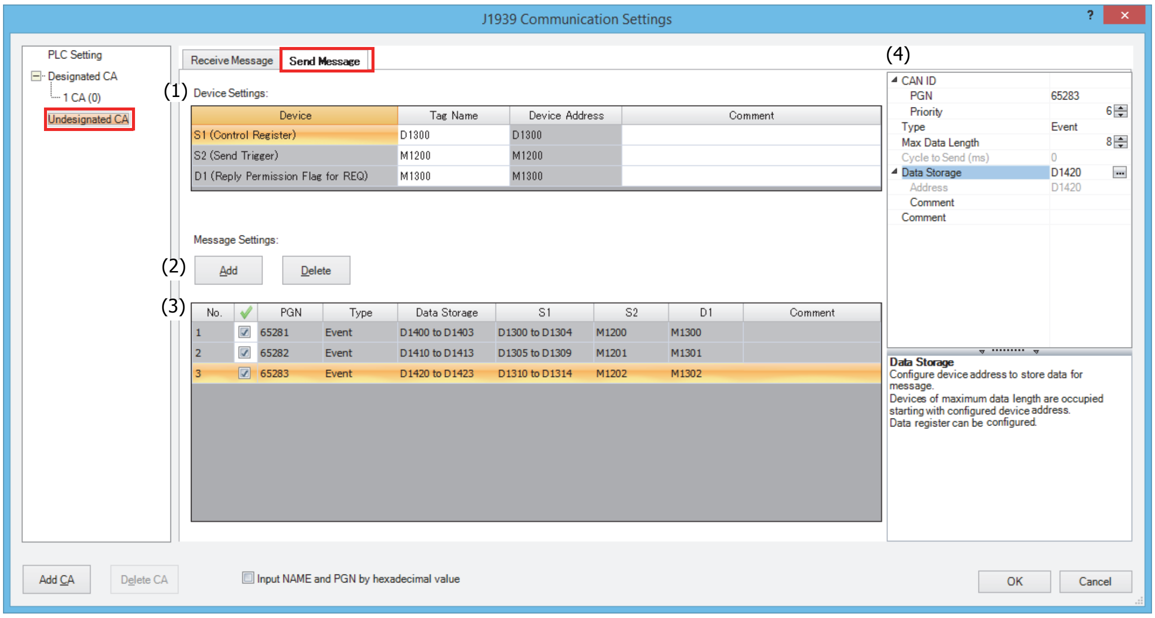
Address: Shows the device address that corresponds to the tag name of the device or the device address.

Comment: Shows the comment that corresponds to the tag name of the device or the device address. This item can be edited.

Comment
Sets the comment for the received message. The comment that can be set is a maximum of 127 bytes.

■ Send Message tab

This tab configures messages that the CAN J1939 All-in-One CPU module transmits to the J1939 communication network as broadcasts.



(1) Device Settings

Sets the data registers that will be used by the sent messages.

Settings	Description
Tag Name	Specifies the tag name of the device or the device address.
Device Address	Shows the device address that corresponds to the tag name.
Comment	Shows the comment for the device address. This item can be edited.

S1 (Control Register)

Sets the starting address of the data registers that will store the transmit settings and transmit results.

Storage Destination	Item		Description
Starting number+0	Transmit data length		This register stores the length of the message data to transmit. Starting from the data storage destination, data in the length of the transmit data length will be transmitted to the CA. When 0 is stored for the transmit data length, data of the maximum data length will be transmitted to the CA. This cannot be set larger than the maximum data length. If the message that will be transmitted exceeds 8 bytes, the message is split into 8-byte packets and transmitted as a multi-packet message.
Starting number+1	Request response destination address		Stores the source address of the received request PGN as the PGN response destination address.
Starting number+2	Transmit results status		This register stores the transmit results. To initialize this register, write "0". For details on the transmit results status, see "Transmit results status" on page 8-26.
Starting number+3	Transmit action flag		Stores the action flag for transmitting.
	Bit 0	REQ received flag	This flag is turned on when a request PGN from a CA was received. This flag is turned off when the corresponding PGN is transmitted.
	Bits 1 to 15	**Reserved**	
Starting number+4	**Reserved**		

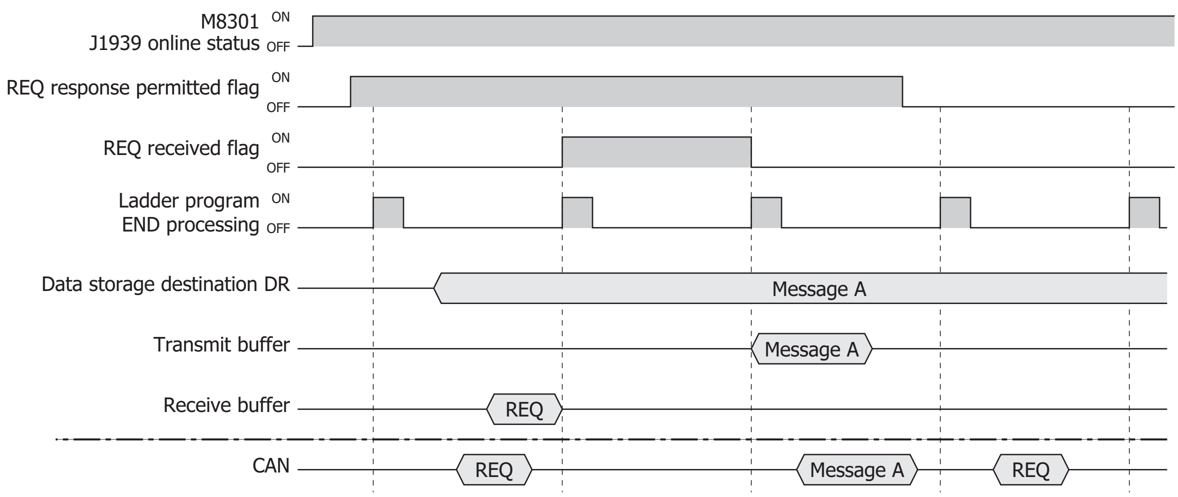
Transmit results status

Status Code	Description	Details
0	Normal	
1	Multi-packet message transmitted	A message with a data length of 9 bytes or longer has finished being transmitted.
10	Transmit data length error	The length of the transmitted message data was longer than the maximum data length.
20	Internal transmit queue overflow	The limit on the number of messages that can be queued for transmission has been reached.
100	Multi-packet transmit error	A message with a data length of 9 bytes or longer could not be transmitted.
200	Local address unconfirmed	The message could not be transmitted because the local address is unconfirmed.
201	Destination address unconfirmed	The message could not be transmitted because the destination CA address is unconfirmed.

S2 (Send Trigger)
Sets the transmit trigger when "Event transmission" is selected for **Type**. When the transmit trigger is turned on, the message is transmitted in ladder program END processing. After the message is transmitted, the transmit trigger is turned off.

D1 (Reply Permission Flag for REQ)
This function responds to the transmit request in a message transmitted as a broadcast. When PGN 59904 (Request) transmitted as a broadcast is received, if the REQ response permitted flag for the corresponding PGN is on, the PGN 59904 (Request) source address is stored in the request response destination address, and the transmit action flag's REQ received flag is turned on. In the next END processing, the requested PGN message is transmitted to the request response destination address.

Request response example



(2) Add button/Delete button

Click **Add** to add a transmitted message to the message list (3).
To delete a message, select the message to delete in the message list, and then click **Delete**.

(3) Message list

Displays the content configured by the message parameter settings (4). The send message can also be enabled or disabled. The message is enabled when the check box is selected and disabled when the check box is cleared.

Notes:
Click the check box column header to perform the following operations.

- If all transmitted message check boxes are on, all of the check boxes are cleared.
- If any transmitted message check boxes are off, all of the check boxes are selected.

(4) Message parameter settings

These settings configure the details of the transmitted message selected in the message list.

CAN ID

PGN: Sets the PGN of the message to transmit. The range of the PGN that can be set is 0 to 61,184 (0000h to EF00h) (PDU1 format) and 61,440 to 65,535 (F000h to FFFFh) (PDU2 format). Set the low order 1 byte (PDU Specific) that indicates the destination address in PDU1 format to 00h. When the PGN is set in the PDU1 format range, it is transmitted as a message with a destination address of 255 (global address).

Click **...** and a PGN can be selected in the displayed PGN Manager. For details, see "PGN Manager" on page 8-29.

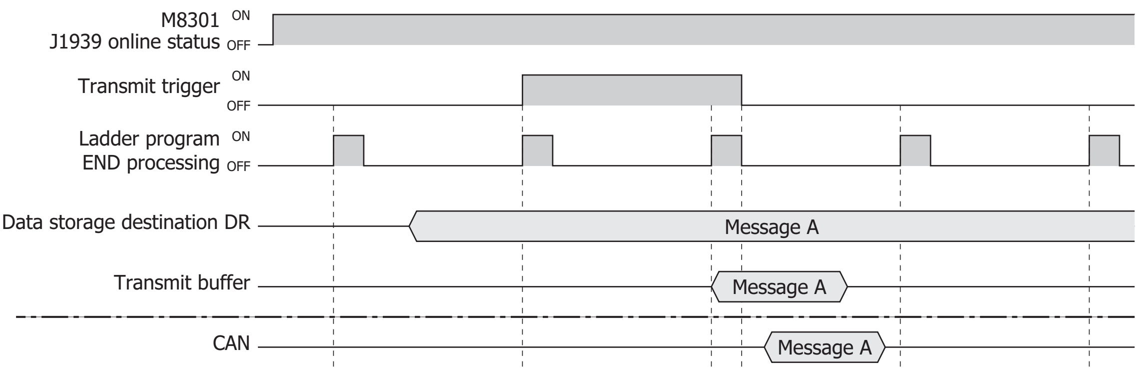
Priority: Sets the priority for the message to transmit between 0 and 7.

Type

Sets the transmit method for the message. Select as "Event" or "Cycle".

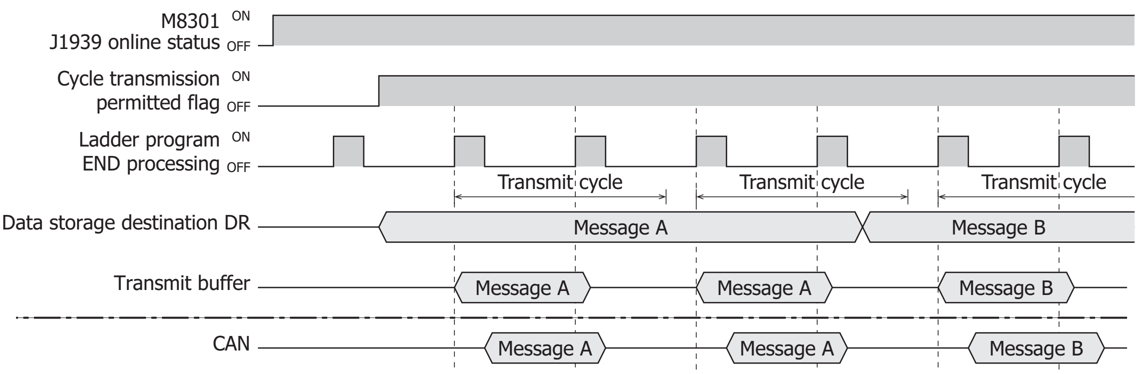
Event: When the transmit trigger is turned on, the transmit data stored in the data storage destination is transmitted in END Processing.

Undesignated CA event transmission example



Cycle: While the cycle transmission permitted flag is on and when the transmit cycle occurs, the transmit data stored in the data storage destination is transmitted in END processing. The actual transmit cycle is affected by the ladder program scan time.

Undesignated CA cycle transmission example



Max Data Length

Sets the maximum data length of the PGN to transmit. The maximum value that can be set is 252 bytes.

Cycle to Send (ms)

Sets the transmit cycle in 10 ms increments when "Cycle transmission" is selected for **Type**. The range of the value that can be set is 10 to 655,350.

Data Storage
Sets the data registers that will store the transmitted message. The maximum data length is used starting from the set device. When transmitting a message, the data is first copied from the data storage destination to the internal transmit buffer and then transmitted.

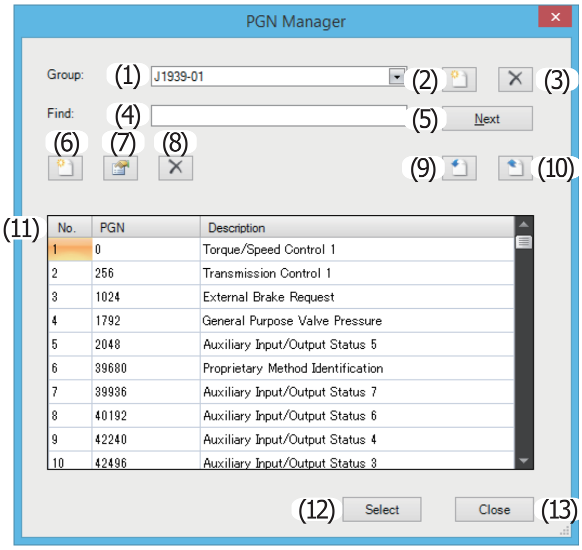
Address: Shows the device address that corresponds to the tag name of the device or the device address.

Comment: Shows the comment that corresponds to the tag name of the device or the device address. This item can be edited.

Comment
Sets the comment for the transmitted message. The comment that can be set is a maximum of 127 bytes.

PGN Manager

PGNs for transmitted and received messages are collectively managed with PGN Manager.



(1) Group list

Displays the registered PGN groups.
When the group is changed, the PGNs registered to the selected group are displayed in the PGN list (11).
J1939 has been registered in advance. A maximum of 20 groups including **J1939** can be registered.

(2) Add group button

Adds a group.
Click this button to display PGN Group Editor. Add a new group with PGN Group Editor. For details, see "PGN Group Editor" on page 8-31.
The added group is added to the group list.

(3) Delete group button

Deletes a group.
Select a group from the group list and click this button to delete it.
The default group **J1939** cannot be deleted.

(4) Find

Enter a string to find in the PGN list as a maximum of 128 bytes.

(5) Next

Finds the string entered in **Find** in the PGN list. Enter a string in **Find** and click this button.

(6) Add PGN button

Adds a PGN to the group.
Select the group in the group list where the PGN will be added and click this button to display PGN Editor.
Add a new PGN with PGN Editor. For details, see "PGN Editor" on page 8-31.
The added PGN is added to the PGN list.

(7) Edit PGN button

Edits a PGN.
Select a registered group in the group list that contains the PGN to edit, and then select the PGN to edit in the PGN list. Click this button to display PGN Editor.
Edit the PGN in PGN Editor. For details, see "PGN Editor" on page 8-31.
The content of the PGN selected in the PGN is updated.

(8) Delete PGN button

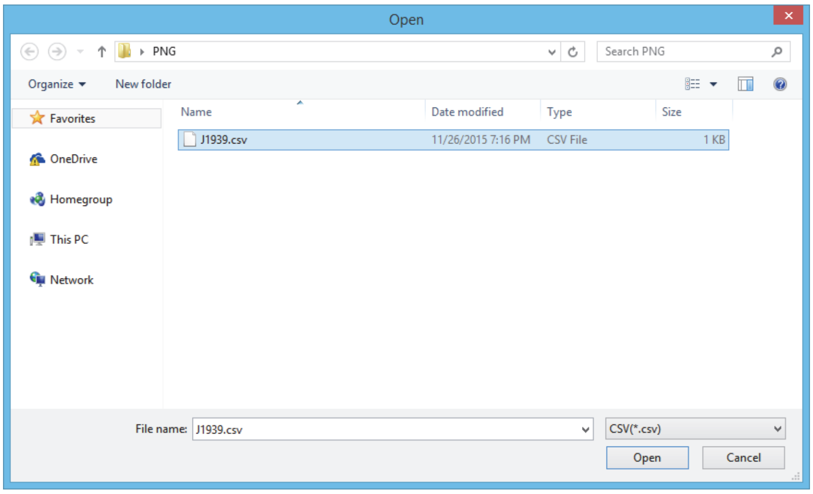
Deletes a PGN.
Select a registered group in the group list that contains the PGN to delete, and then select the PGN to delete in the PGN list. Click this button to delete the PGN.

(9) Import PGNs button

Imports PGNs saved as a CSV file.

Select the group where the PGNs will be imported in the group list. PGNs that are already registered to the selected group are all deleted before the import is executed.

Click this button to display the **Open** dialog box.

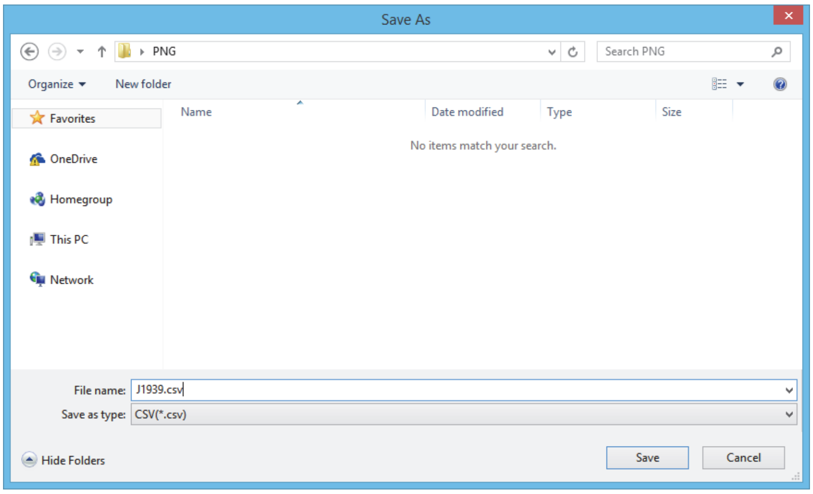


Select the CSV file where the PGNs are saved and click **Open** to import the PGNs in that file to the PGN list for the group selected in the group list.

(10) Export PGNs button

Exports the PGNs registered to the group selected in the group list as a CSV file.

Click this button to display the **Save As** dialog box.



Select the location to save the file, enter a file name, and then click **Save** to save the PGNs registered to the group as a CSV file.

(11) PGN list

Displays the PGNs registered to the group selected in the group list.

No.: Displays the control number for the registered PGN (1 to 65,535).

PGN: Displays the PGN.

Name: Displays the PGN name.

(12) Select button

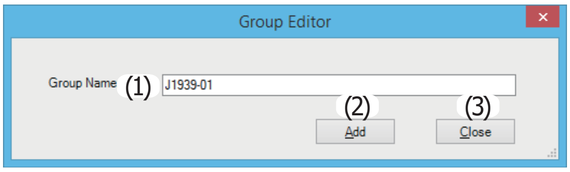
Closes PGN Manager and sets **CAN ID** in the calling message parameter to the PGN selected in the PGN list.

(13) Close button

Closes PGN Manager.

■ **PGN Group Editor**

Adds a new group.



(1) Group Name

Enter the group name for the group that will be used to register PGNs as a maximum of 64 bytes.

(2) Add button

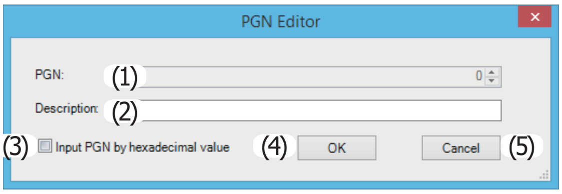
Closes PGN Group Editor and adds the group to the PGN Manager group list.

(3) Close button

Closes PGN Group Editor.

■ **PGN Editor**

Adds new PGNs and edits existing PGNs.



(1) PGN

Enter the PGN to register as a maximum of 5 characters.

When the **Input PGN by hexadecimal value** check box is cleared, enter this as a decimal value. When the check box is selected, enter this is a hexadecimal value.

Example: When hexadecimal: 0001, FF00, etc.

(2) Description

Enter the name of the PGN to register as a maximum of 128 bytes.

(3) Input PGN by hexadecimal value check box

Specifies the PGN entry format.

When cleared, enter the value as a decimal value. When selected, enter the value as a hexadecimal value.

(4) OK button

Closes PGN Editor and adds the PGN to the PGN Manager group list.

(5) Cancel button

Closes PGN Editor.

9: BLUETOOTH COMMUNICATION

This chapter describes the Bluetooth communication of the FC6A Series MICROSmart.

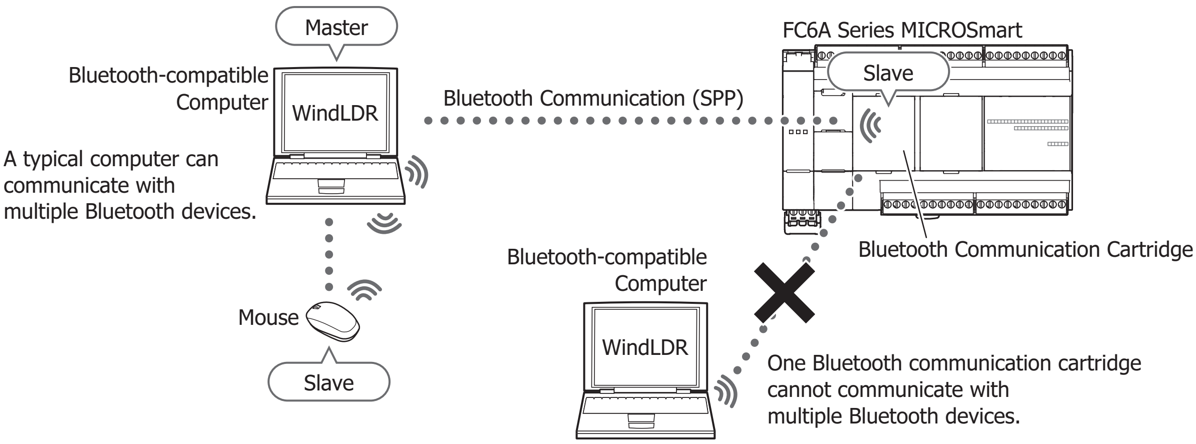
Overview

Bluetooth communication can be used by connecting a Bluetooth communication cartridge to the FC6A Series MICROSmart. Bluetooth communication cartridge can be connected to the three cartridge slots, cartridge slots 1, 2, and 3 (Plus CPU module only). Bluetooth communication cartridges can have the serial communication with Bluetooth devices that support SPP profile.

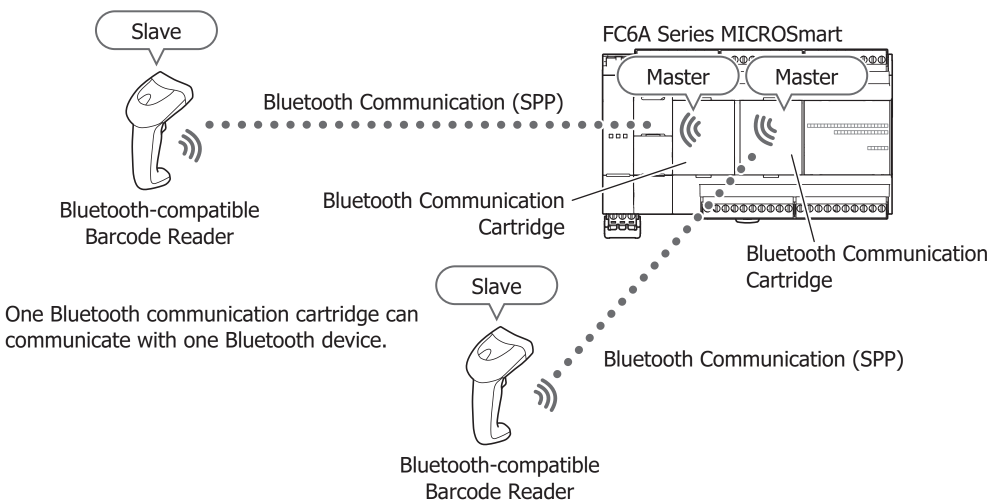
The FC6A Series MICROSmart with a connected Bluetooth communication cartridge can connect to a Bluetooth network as a master or slave. Regardless of whether the FC6A Series MICROSmart is a master or slave, the FC6A Series MICROSmart can connect to and communicate with only one Bluetooth device per Bluetooth communication cartridge.

On a Bluetooth network, the source of the connection is called the master and the destination of the connection is called the salve.

Example of Computer and FC6A Series MICROSmart Connection



Example of FC6A Series MICROSmart and Bluetooth Device Connection



Wireless Communication Specifications

Type No.	FC6A-PC4
Bluetooth Standard	Bluetooth ver.2.1 + EDR
Profile	SPP (Serial Port Profile) iAP (iPod Accessory Protocol)
Frequency Range	2,402 MHz to 2,480 MHz
Wireless Transmission Distance[*1]	10 m (Class 2)
Multi-point Function	8 units
Communication Protocol	Maintenance communication protocol, user communication protocol
Bluetooth Wireless Approved Regions[*2]	Japan, China, USA, Canada, Australia, New Zealand, Europe

*1 Connection effective range is affected by obstacles (human, metal, wall) and wave signal condition. Make sure to confirm the connection status before actual operation.

*2 Depending on countries or regions, evaluations on the device equipped with FC6A may be necessary.

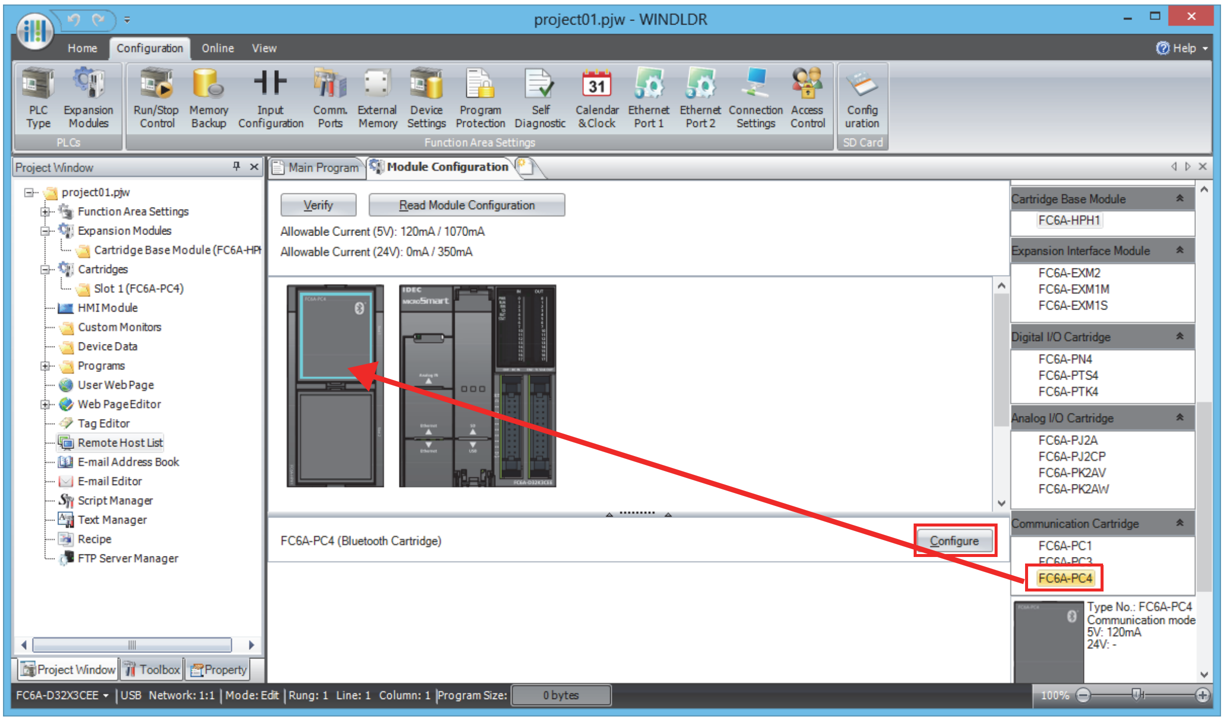
Note: Communication performance (required time) in maintenance communication is as follows.

- User program upload equivalent to 10,000 steps: 2 minutes 40 seconds approx.
- User program download equivalent to 10,000 steps: 2 minutes 30 seconds approx.
- User program upload equivalent to 20,000 steps: 5 minutes 20 seconds approx.
- User program download equivalent to 20,000 steps: 5 minutes approx.
- 100 KB CSV file retrieval: 1 minute 30 seconds approx.
- 200 KB CSV file retrieval: 3 minutes approx.

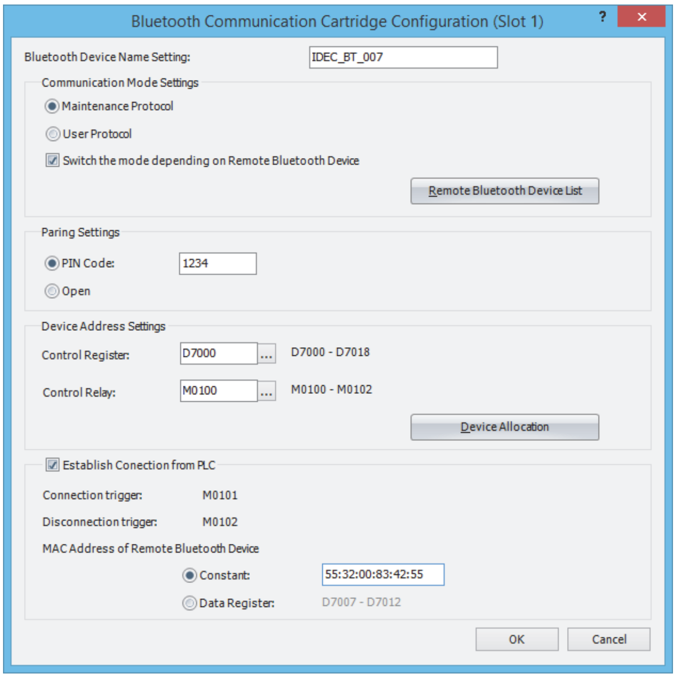
Configuring the Bluetooth Communication Cartridge

Programming WindLDR

1. The Bluetooth communication cartridge is configured with the Module Configuration Editor.
   On the **Configuration** tab, in the **PLCs** group, select **Expansion Modules**.
2. Click the Bluetooth communication cartridge inserted in the module configuration area and click **Configure**.
   The **Bluetooth Communication Cartridge Configuration** dialog box is displayed.



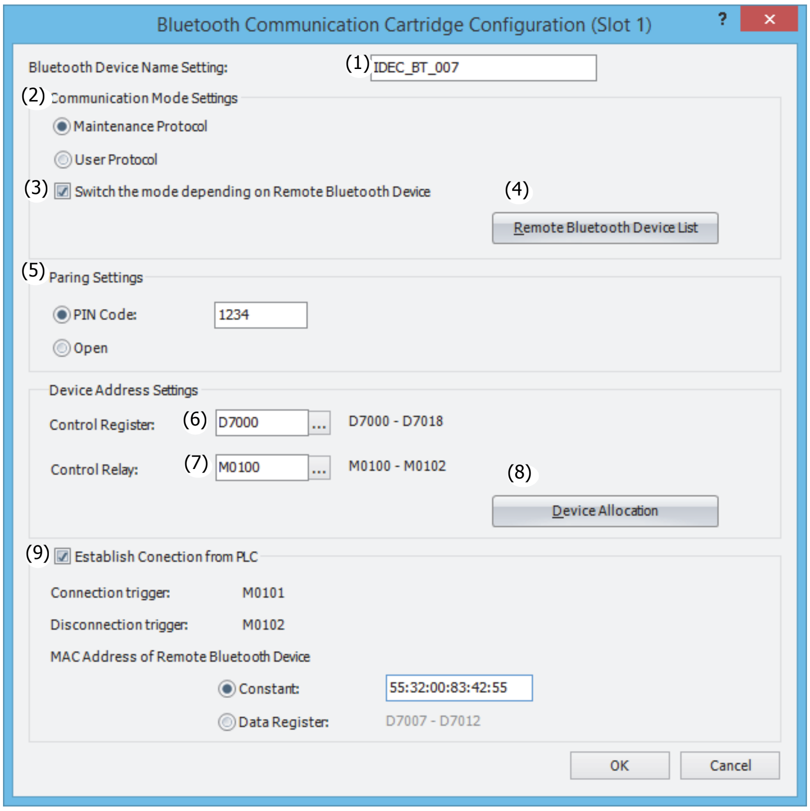
3. Configure the Bluetooth communication cartridge settings.



4. Download the user program to the FC6A Series MICROSmart.

This concludes configuring the Bluetooth communication cartridge.

Bluetooth Communication Cartridge Configuration



(1) Bluetooth Device Name Setting

Set the name that will be displayed on a computer or smartphone when Bluetooth devices in the area are detected from that computer or smartphone.

(2) Communication Mode Settings

Select the communication mode of the FC6A Series MICROSmart in Bluetooth communication from **Maintenance Protocol** or **User Protocol**.

When Switching the mode depending on Remote Bluetooth Device (3) is selected, the communication mode configured in (3) is applied with a higher priority.

(3) Switch the mode depending on Remote Bluetooth Device

Select this check box to switch the communication mode of the FC6A Series MICROSmart according to the connected Bluetooth device.

For example, use this option to switch to maintenance communication when the FC6A Series MICROSmart connects to a smart phone and to switch to user communication when the FC6A Series MICROSmart connects to a barcode reader after it is disconnected from the smart phone.

If this check box is not selected or this check box is selected but no Bluetooth devices are registered in the **Remote Bluetooth Device List** dialog box, the FC6A Series MICROSmart will operate with the communication mode set by Communication Mode Settings (2).

When Switching the mode depending on Remote Bluetooth Device (3) is configured and FC6A Series MICROSmart establishes a connection with a Bluetooth device, the FC6A Series MICROSmart determines the communication mode. Therefore, the FC6A Series MICROSmart does not receive the SPP data for a few seconds from when the connection is established. SPP communication is possible when the communication mode is changed and the Bluetooth communication cartridge (cartridge slot) status in the control register (6) is changed.

(4) Remote Bluetooth Device List button

MAC address and communication mode for the remote Bluetooth devices to connect for Switching the mode depending on Remote Bluetooth Device (3) can be configured.

Click this button to display the **Remote Bluetooth Device List** dialog box.

The MAC address, communication mode, and comment of each Bluetooth device are displayed in the dialog box. For details, see "Remote Bluetooth Device List" on page 9-6.

(5) Pairing Settings

Select the security when making a Bluetooth connection from the following.

PIN Code : Requires the input of the configured PIN code when making a connection with the Bluetooth device. The communicated content is encrypted.
This pairing method is typically used with devices up to Bluetooth v2.0.

Open : The communicated content is not encrypted.
This setting is used to communicate with devices up to Bluetooth v2.0 with no security.

(6) Control Registers

Specify the first data register of the control registers storing MAC addresses of the remote Bluetooth device and the Bluetooth communication cartridge connected to the FC6A Series MICROSmart. Special data registers cannot be specified.

Device Address	Item	Description	R/W
Starting number+0	Status of Bluetooth communication cartridge (cartridge slot)	Stores the status and communication mode of the Bluetooth communication cartridge. 0: No operation 255: In operation, such as waiting for Bluetooth connection and disconnection 1: Maintenance communication 2: User communication The default setting is 0 (No operation).	R
Starting number+1 Starting number+2 Starting number+3 Starting number+4 Starting number+5 Starting number+6	MAC address of Bluetooth communication cartridge (cartridge slot)	Stores the MAC address of the Bluetooth communication cartridge connected to the cartridge slot. Example: MAC address: AA-BB-CC-DD-EE-FF Starting number+1=AA, starting number+2=BB, starting number+3=CC, starting number+4=DD, starting number+5=EE, starting number+6=FF	R
Starting number+7 Starting number+8 Starting number+9 Starting number+10 Starting number+11 Starting number+12	MAC address of the remote Bluetooth device with which the Bluetooth communication cartridge makes a connection (cartridge slot)	Store the MAC address of the remote Bluetooth device to connect. Example: MAC address: AA-BB-CC-DD-EE-FF Starting number+1=AA, starting number+2=BB, starting number+3=CC, starting number+4=DD, starting number+5=EE, starting number+6=FF	R/W
Starting number+13 Starting number+14 Starting number+15 Starting number+16 Starting number+17 Starting number+18	**—Reserved—**		—

(7) Control Relays

Specify the first internal relay of the control relays storing the Bluetooth communication cartridge (cartridge slot) communication status and connection and disconnection triggers for the Establish Connection (9). Special internal relays cannot be specified.

Device Address	Item	Description	R/W
Starting number+0	Status of Bluetooth communication cartridge (cartridge slot)	0: Cannot communicate with the remote Bluetooth device 1: Can communicate with the remote Bluetooth device	R
Starting number+1	Connection trigger of Bluetooth communication cartridge (cartridge slot)	0: Do not process 1: Connect the connection target Bluetooth device	R/W
Starting number+2	Disconnection trigger of Bluetooth communication cartridge (cartridge slot)	0: Do not process 1: Disconnect the connection target Bluetooth device	R/W

(8) Device Allocation button

Click this button to display the **Device Allocation** dialog box.

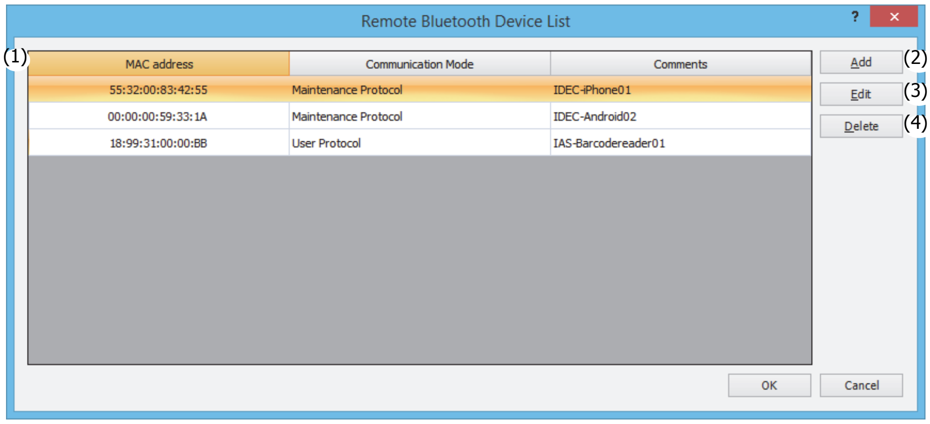
The dialog box displays a table with the Bluetooth communication cartridge settings and corresponding data registers and internal relays. For details, see "Device Allocations" on page 9-7.

(9) Establish Connection from PLC

Select this check box to connect to and disconnect from the remote Bluetooth device when FC6A Series MICROSmart is the master. When this check box is selected, iOS devices cannot be connected.

Connection Trigger	:	The device address of the control relay, which is configured in (7), + 1 is displayed as the connection trigger. When this is turned off to on, a connection is made with the remote Bluetooth device configured in **MAC Address of Remote Bluetooth Device**.
Disconnection Trigger	:	The device address of the control relay, which is configured in (7), + 2 is displayed as the disconnection trigger. When this is turned off to on, a connection with the remote Bluetooth device configured in **MAC Address of Remote Bluetooth Device** is disconnected.
MAC Address of Remote Bluetooth Device	:	The remote Bluetooth device to connect can be specified with either **Constant** or **Data Register**. When **Constant** is selected, MAC address configured in the **Bluetooth Communication Cartridge Configuration** dialog box becomes the target to connect. When **Data Register** is selected, MAC address configured with the first data register +7 to +12 of the control register (6) becomes the target to connect.

Remote Bluetooth Device List



(1) Bluetooth Device List

Displays the MAC address, communication mode, and comment of each Bluetooth device.

(2) Add

Click this button to display the **Remote Bluetooth Device Settings** dialog box. This dialog box is used to enter the MAC address of the connected Bluetooth device, communication mode to switch to at the start of the connection, and comment.

A maximum of 20 devices can be registered. For details, see "Remote Bluetooth Device Settings" on page 9-7.

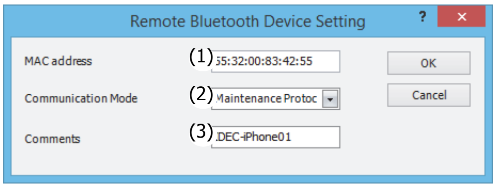
(3) Edit

Select a Bluetooth device in the Bluetooth Device List (1) and click this button to display the **Remote Bluetooth Device Settings** dialog box. This dialog box is used to edit the MAC address of the connected Bluetooth device, communication mode to switch to at the start of the connection, and comment. For details, see "Remote Bluetooth Device Settings" on page 9-7.

(4) Delete

Select a Bluetooth device in the Bluetooth Device List (1) and click this button to delete the Bluetooth device.

Remote Bluetooth Device Settings



(1) MAC address

Set the MAC address of the connected Bluetooth device.

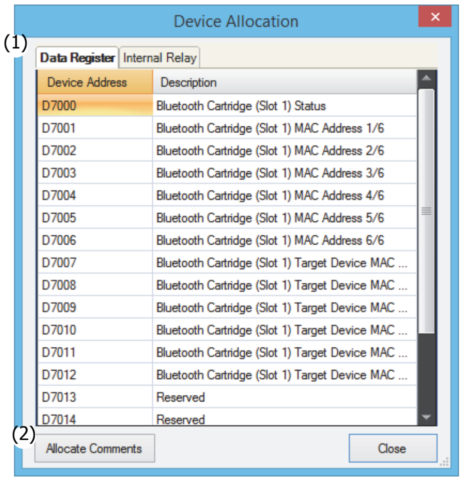
(2) Communication Mode

Select the communication mode to switch to at the start of the connection from **Maintenance Protocol** or **User Protocol**.

(3) Comments

Enter the MAC address of the connected Bluetooth device.

Device Allocations



(1) Device Allocations List

Displays a table with the Bluetooth communication cartridge settings, data registers, and internal relays.

(2) Allocate Comments button

Click this button to set the content in the table as the comments of the devices.

Configuring and Communicating with Bluetooth Devices

Maintenance Communication with a Computer

This section describes how to perform maintenance communication with a computer using the Bluetooth communication.

This enables you to have a computer, in which WindLDR is installed, connect with the Bluetooth communication cartridge to monitor and change device values and to upload and download the user program.

The FC6A Series MICROSmart waits for the connection as the slave and the computer acts as the master to make the connection on the Bluetooth network.

FC6A Series MICROSmart Settings

1. Connect the Bluetooth communication cartridge to the FC6A Series MICROSmart.

2. Configure the Bluetooth communication cartridge with WindLDR.
For details, see "Programming WindLDR" on page 9-3.

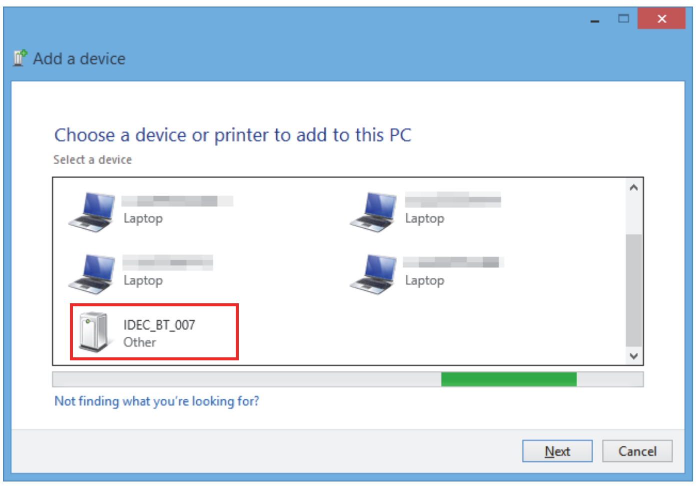
Example: **Bluetooth Communication Cartridge Configuration** dialog box settings

Settings		Description
Bluetooth Device Name Setting		IDEC_BT_007
Communication Mode		Maintenance Communication
Pairing Setting		PIN Code "1234"
Device Settings	Control Registers	D7000
	Control Relays	M0100

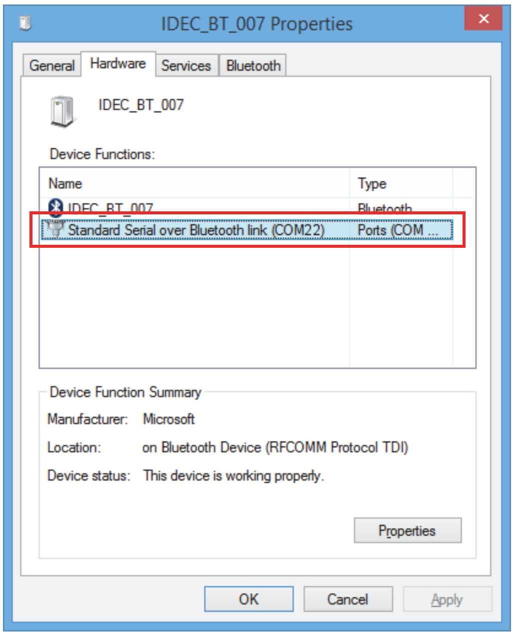
3. Download the user program to the FC6A Series MICROSmart.

Computer Settings

1. Search for the name of the Bluetooth communication cartridge with the computer.
The name is set with the **Bluetooth Device Name Setting** in the **Bluetooth Communication Cartridge Configuration** dialog box in WindLDR.
(IDEC_BT_007)

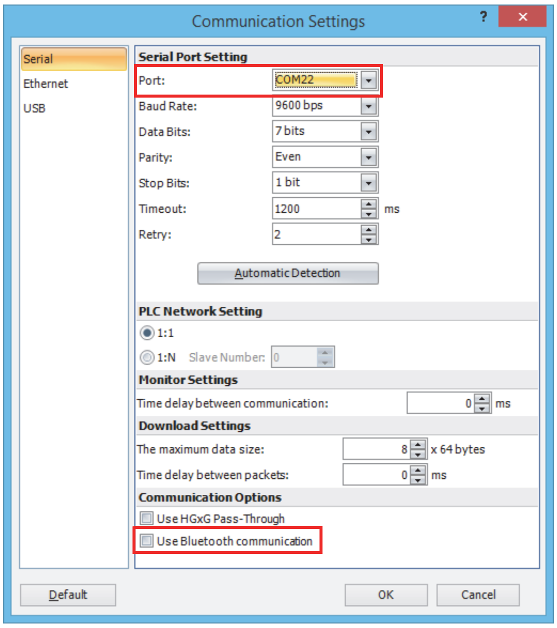


2. Connect to the Bluetooth communication cartridge and create a virtual COM port. When connecting the first time, pairing is necessary and entering the PIN code "1234" is required.



Checking Bluetooth Communication

1. Perform the serial communication using the virtual COM port.
   Select **Use Bluetooth communication**.

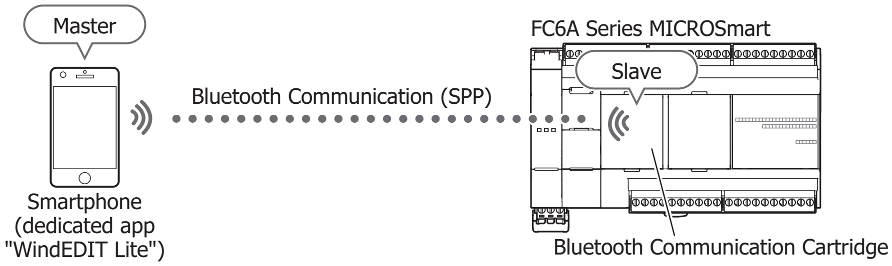


2. Run the monitor in WindLDR.

3. Check the value of the device that indicates the Bluetooth communication status.
   Check if the Bluetooth communication cartridge status (M0100) is 1. The Bluetooth communication cartridge (cartridge slot) status is the first internal relay + 0 of the control relays configured in **Device Address Settings** in **Bluetooth Communication Cartridge Configuration** dialog box.

Maintenance Communication with a Smartphone

This section describes how to perform maintenance communication with a smartphone using the Bluetooth communication. This enables you to have a smart phone, in which the dedicated app is installed, connect with the Bluetooth communication cartridge to monitor and change device values and to upload and download the user program.
The FC6A Series MICROSmart waits for the connection as the slave and the smartphone acts as the master to make the connection on the Bluetooth network.



Unspecified access can be prevented by using password authentication to restrict access to the FC6A Series MICROSmart.

FC6A Series MICROSmart Settings

1. Connect the Bluetooth communication cartridge to the FC6A Series MICROSmart.

2. Configure the Bluetooth communication cartridge with WindLDR.
For details, see "Programming WindLDR" on page 9-3.

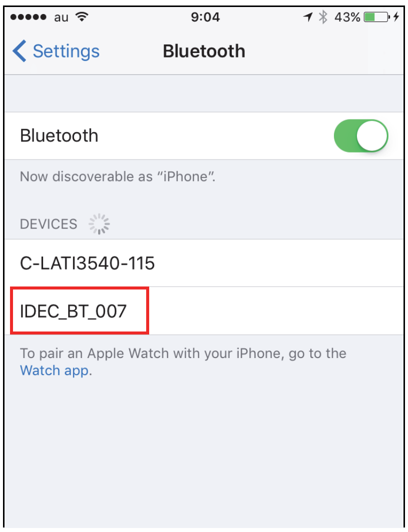
Example: **Bluetooth Communication Cartridge Configuration** dialog box settings

Settings		Description
Bluetooth Device Name Setting		IDEC_BT_007
Communication Mode		Maintenance Communication
Pairing Setting		PIN Code "1234"
Device Settings	Control Registers	D7000
	Control Relays	M0100

3. Download the user program to the FC6A Series MICROSmart.

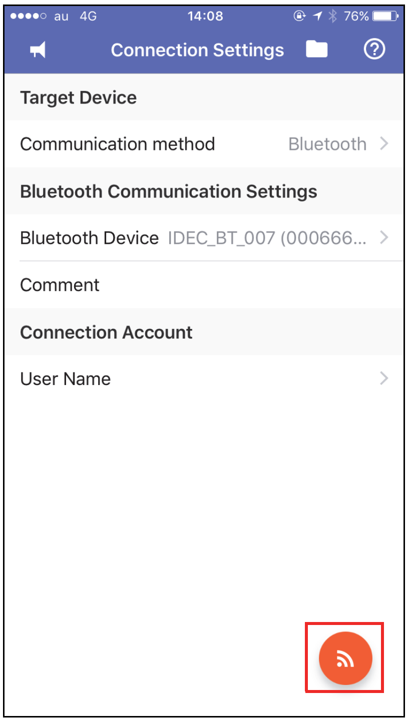
Smartphone Settings

1. Search for the name of the Bluetooth communication cartridge with a smart phone.
The name is set with the **Bluetooth Device Name Setting** in the **Bluetooth Communication Cartridge Configuration** dialog box in WindLDR.
(IDEC_BT_007)



Note: The search screen for the Bluetooth communication devices depends on each smart phone to use.

2. Connect to the Bluetooth communication cartridge from the dedicated app.



Checking Bluetooth Communication

1. Check the device value that indicates the Bluetooth communication status with the dedicated app **WindEDIT Lite**. Check if the Bluetooth communication cartridge status (M0100) is 1. The Bluetooth communication cartridge (cartridge slot) status is the first internal relay + 0 of the control relays configured in **Device Address Settings** in **Bluetooth Communication Cartridge Configuration** dialog box.

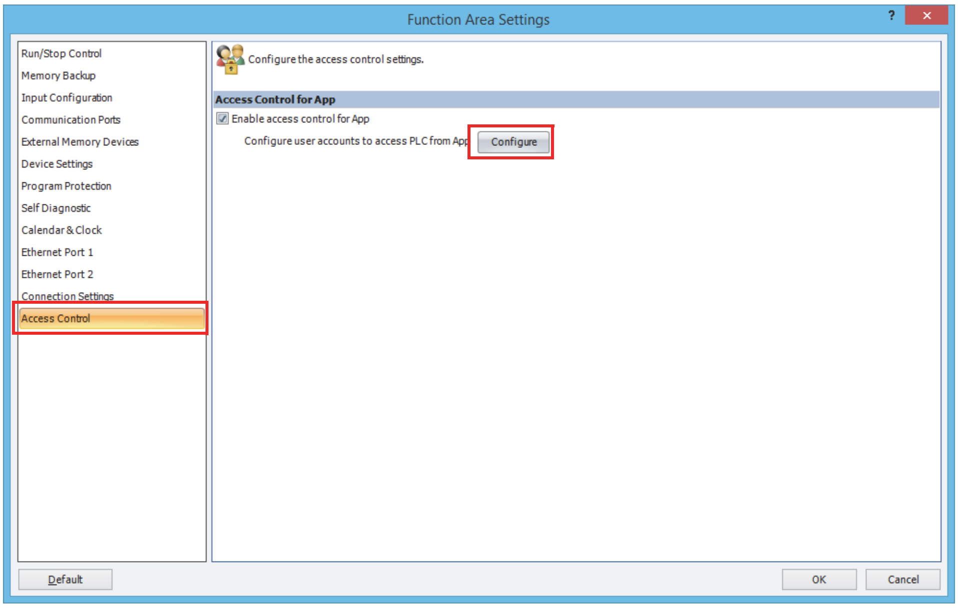
Note: For how to use the dedicated app **WindEDIT Lite**, see the "WindEDIT Lite User's Manual".

User Account Settings

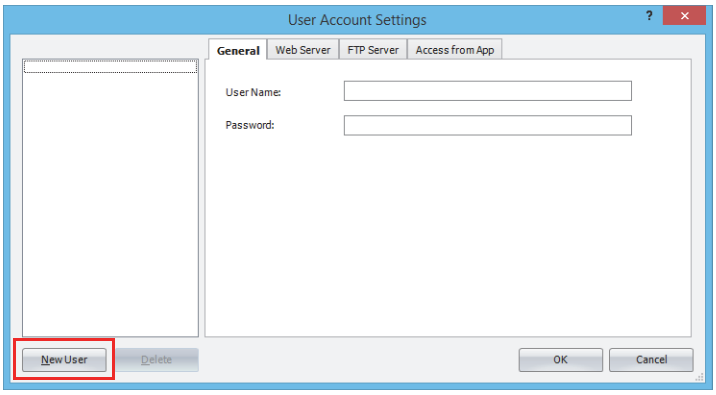
Creating User Accounts

● Operation procedure

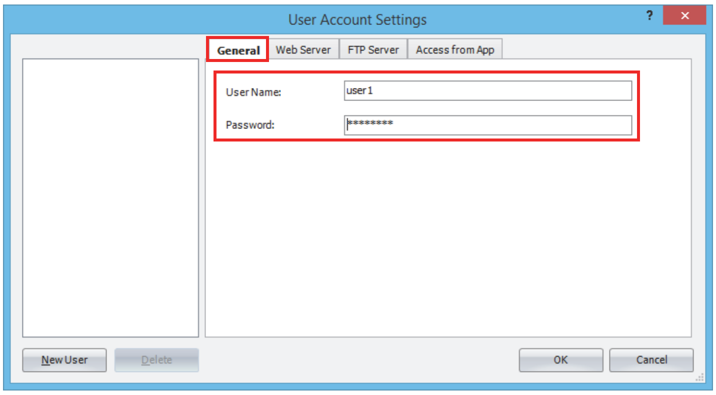
1. In the **Function Area Settings** dialog box, under **Access Control**, select the **Enable access control for App** check box.
2. Click **Configure**.
   The **User Account Settings** dialog box is displayed.



3. Click **New User**.
   A user account is created in the list.

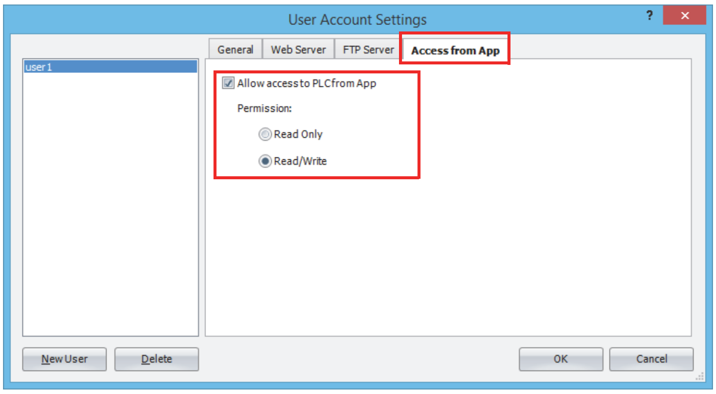


4. On the **General** tab, set **User Name** and **Password**.



5. Click the **Access from App** tab and select the **Allow access to PLC from App** check box.

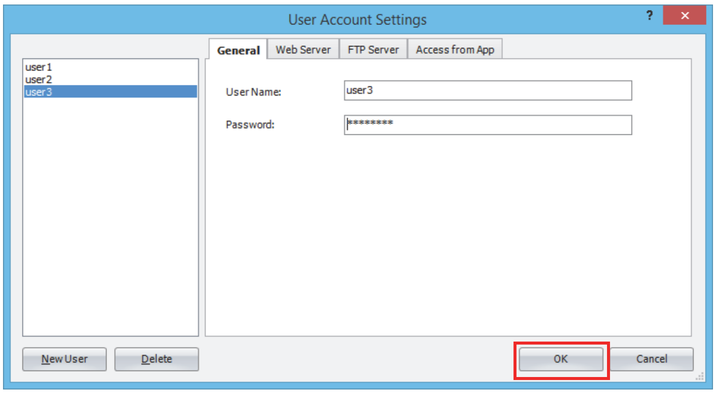
6. Under **Permission**, select **Read Only** or **Read/Write**.



7. Repeat the steps 2 to 6 and create the required number of user accounts.

Note: If you select a user account on the list, you can edit the settings of the selected user account on the **General** and **Access from App** tabs.

8. Click **OK**.

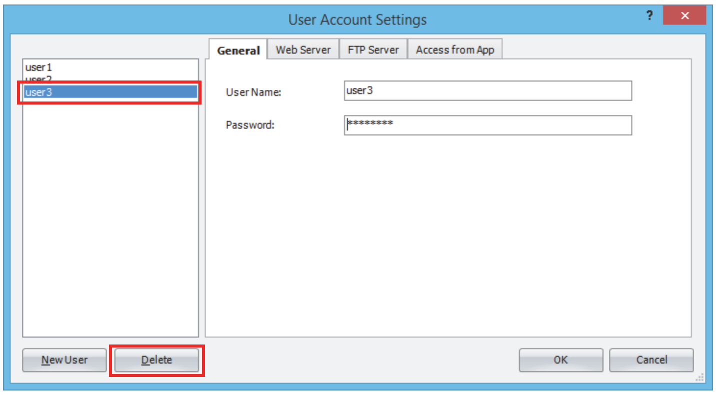


This concludes creating user accounts.

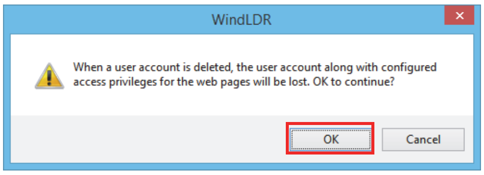
Deleting a Created User Account

● Operation procedure

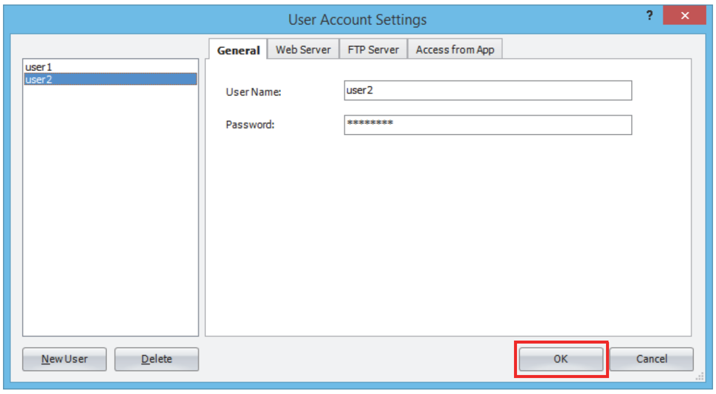
1. Select the user account on the list and click **Delete**.
   A confirmation message is displayed.



2. Click **OK**.



3. Click **OK**.



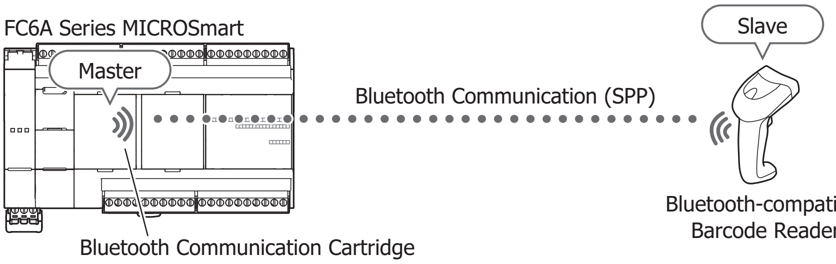
This concludes deleting a user account.

User Communication with a Barcode Reader

This section describes how to perform user communication with a barcode reader using the Bluetooth communication.

The FC6A Series MICROSmart can connect to a supported barcode reader and perform user communication.

The configuration method depends on the specifications of the barcode reader to use. In the following description, the barcode reader waits for the connection as the slave and the FC6A Series MICROSmart acts as the master to make the connection on the Bluetooth network.



Note: The recommended barcode reader is IDEC's GBT4400.

FC6A Series MICROSmart Settings

1. Connect the Bluetooth communication cartridge to the FC6A Series MICROSmart.
2. Configure the Bluetooth communication cartridge with WindLDR.
   For details, see "Programming WindLDR" on page 9-3.

   Example: **Bluetooth Communication Cartridge Configuration** dialog box settings

Settings		Description
Bluetooth Device Name Setting		IDEC_BT_007
Communication Mode		User Communication
Pairing Setting		PIN code "1234"
Device Settings	Control Registers	D7000
	Control Relays	M0100
Establish Connection from PLC		Selected*1

*1 The MAC address of the barcode reader is required for the connection.

3. Download the user program to the FC6A Series MICROSmart.

Barcode Reader Settings

1. Set the PIN code on the barcode reader.
   Configure the barcode reader by reading the PIN code setup barcode. This will depend on the specifications of the barcode reader. The PIN code is set with the **Pairing Settings** in the **Bluetooth Communication Cartridge Configuration** dialog box in WindLDR.
   (1234)
2. Have the barcode reader wait for a connection.
   Configure the barcode reader by having it read the setup barcode for the wait for connection. This will depend on the specifications of the barcode reader.

Checking Bluetooth Communication

1. Turn on the connection trigger on the FC6A Series MICROSmart.
   Turn on the Bluetooth communication cartridge connection trigger. The Bluetooth communication cartridge (cartridge slot) connection trigger is the first internal relay + 1 of the control relays configured in **Device Address Settings** in **Bluetooth Communication Cartridge Configuration** dialog box. The status code of 255 is stored in D7000, which is the first control register + 0: Bluetooth communication cartridge status (cartridge slot).
2. Run the monitor in WindLDR.

3. Check the device value that indicates the Bluetooth communication status.
Check if the Bluetooth communication cartridge status (M0100) is 1. The Bluetooth communication cartridge (cartridge slot) status is the first internal relay + 0 of the control relays configured in **Device Address Settings** in **Bluetooth Communication Cartridge Configuration** dialog box.

10: FTP SERVER/CLIENT

This chapter describes FTP server and client of the Plus CPU module.

FTP Server

The FTP server is available with the Plus CPU module only.

FTP Server Overview

What the FTP Server Can Do

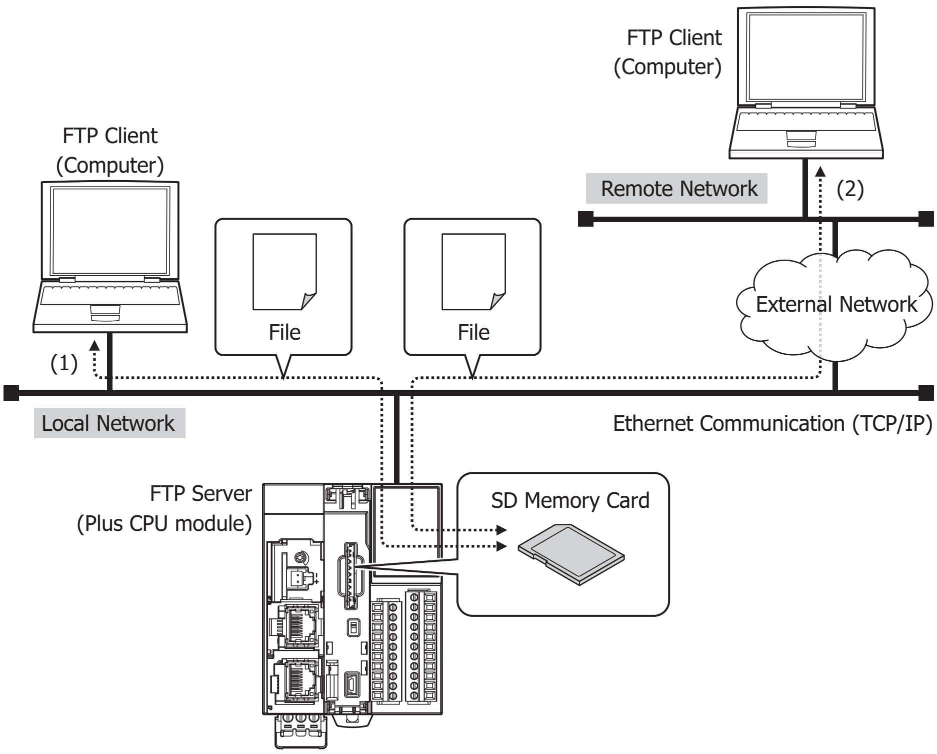
The FTP server enables the Plus CPU module to act as an FTP server to communicate with FTP clients, such as a computer.

FTP clients can perform the following actions.

- Read files from the SD memory card inserted into the Plus CPU module.
- Write files to the SD memory card inserted into the Plus CPU module.

Example of System Configuration

The following diagram shows an example of a system configuration for using the FTP server.



Connect Ethernet port 1 of the Plus CPU module to the network.

(1) Access the Plus CPU module from an FTP client connected to the network, and read files from or write files to the SD memory card inserted into the Plus CPU module.

(2) If the network is connected to an external network, configure the gateway and router of the local network for FTP clients connected via a remote network. Access the Plus CPU module from a remote FTP client, and read files from or write files to the SD memory card inserted into the Plus CPU module. Contact the administrator of the network to which the Plus CPU module is connected for information about the configurations for the gateway and router.

Files Handled by the FTP Server

All files on the SD memory card inserted into the Plus CPU module can be accessed with the FTP server. This section shows the folder and file structure of the SD memory card.

SD Memory Card

Arbitrary: File names and folder names that can be specified by the function area settings or instruction parameters.
Fixed: File names and folder names that are fixed by the system.

- FCDATA01 (Fixed)
  - autoexec.ini (Fixed) — The configuration file that defines actions taken when the Plus CPU module is turned on and when special internal relays are turned on to upload or download. This file is created using WindLDR, but it can also be edited using a text editor such as Notepad.
  - DATALOG (Fixed) — The folder used by the DLOG instruction.
    - RESULT (Arbitrary) — Log files are saved here. The folder name is specified by S1 in the DLOG instruction. The desired folder name up to 8 single-byte alphanumeric characters can be set.
      - 20111202.csv — The CSV log file that is created when the DLOG instruction is executed. For the Plus CPU module, a folder is created and given the date (YYYYMMDD) as its name and then files are saved in that folder. For details, see Chapter 25 "DLOG (Data Log)" in the "FC6A Series MICROSmart Ladder Programming Manual". (Example: 20111202.csv)
  - PROGRAM (Fixed) — The folder used for downloading and uploading the user program. ZLD files are saved here.
    - fc6a01.zld (Arbitrary) — A zld file consists of the user program, program comments, and the system software data for the CPU module. This file can be created by saving the zld file in WindLDR or by uploading the user program to the SD memory card.
  - RECIPE (Fixed) — The folder used by the recipe function. Recipe files are saved here.
    - RCP0001.csv (Arbitrary) — The files in which recipe values are saved. Any file name up to 7 single-byte alphanumeric characters can be set. By default, the file names are automatically assigned as RCP**** (where **** is 0001 to 9999). (Example: RCP0001.csv)
  - TRACE (Fixed) — The folder used by the TRACE instruction.
    - RESULT (Arbitrary) — Log files are saved here. The folder name is specified by S1 in the TRACE instruction. The desired file name up to 8 single-byte alphanumeric characters can be set.
      - 20111202.csv — The CSV log file that is created when the TRACE instruction is executed. For the Plus CPU module, a folder is created and given the date (YYYYMMDD) as its name and then files are saved in that folder. For details, see Chapter 25 "TRACE (Data Trace)" in the "FC6A Series MICROSmart Ladder Programming Manual". (Example: 20111202.csv)
  - WEBPAGE (Fixed) — The folder used by the web server. For details, see "Save Locations of Web Pages" on page 13-2.

Supported Commands and Connection Methods

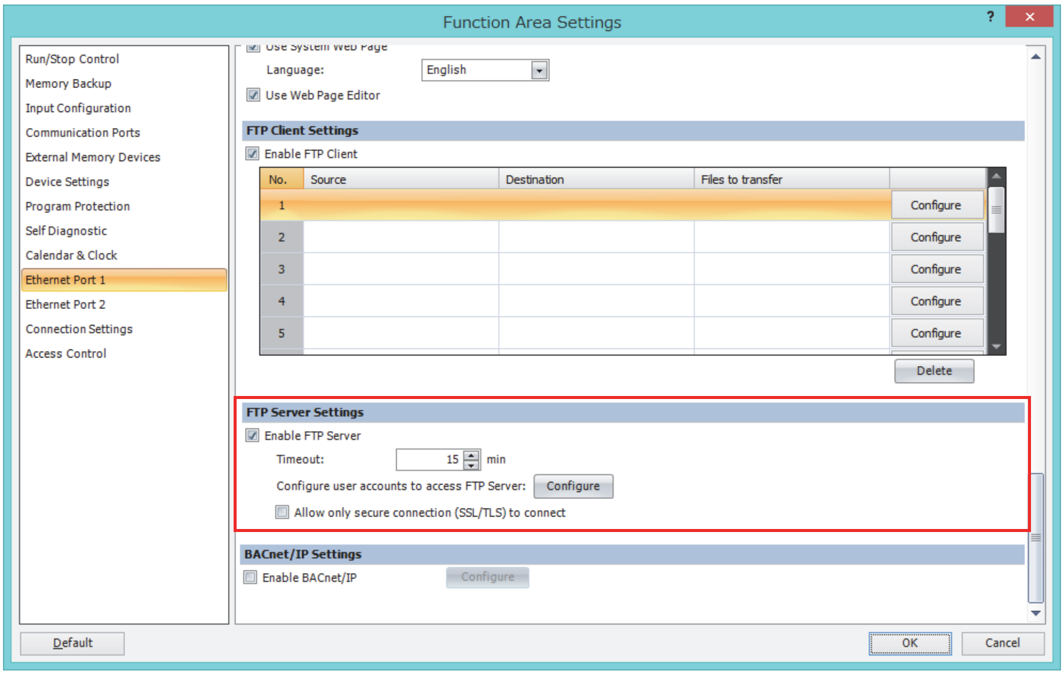
- Commands defined in RFC 959 are supported.
- Active mode and passive mode are supported.
- FTPS (FTP over SSL/TLS) is supported. Explicit mode is supported.
- Only one FTP client connection is allowed at one time.

The recommended FTP client software are as follows.

- FFFTP
- FileZilla
- Core FTP Lite

Programming WindLDR

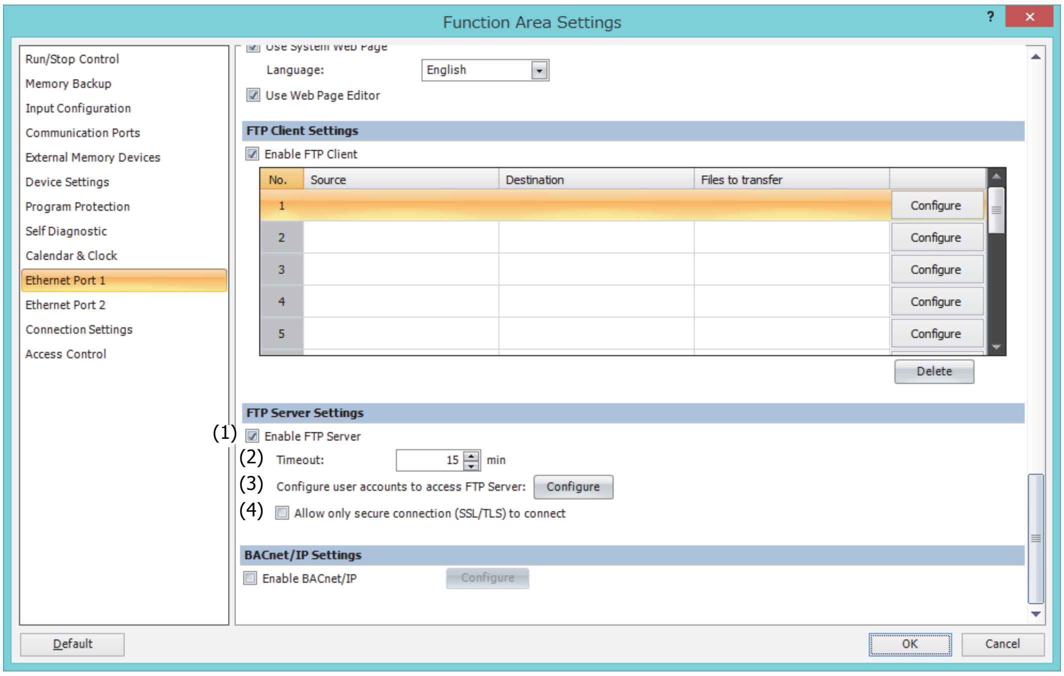
1. On the **Configuration** tab, in the **Function Area Settings** group, click **Ethernet Port 1**.
2. Under **FTP Server Settings**, select the **Enable FTP Server** check box.
3. Configure the FTP server settings.



4. Download the user program to the Plus CPU module.

This concludes configuring the FTP server settings.

FTP Server Settings



(1) Enable FTP Server

To use the FTP server function, select this check box and set the user account. The files on the SD memory card inserted into the Plus CPU module can be read and written from an FTP client. The port number used for the FTP server is 21.

(2) Timeout

Specify the duration of time to disconnect the FTP client. When an FTP client is logged in to the Plus CPU module and there is no communication between the FTP client and the Plus CPU module for the specified duration of time, the connection is automatically disconnected. (Default: 15 minutes)

(3) Configure user accounts to access FTP Server

Click **Configure** and create user accounts in the **User Account Settings** dialog box. For details, see "User Account Settings" on page 10-5.

(4) Allow only secure connection (SSL/TLS) to connect

Select this check box to allow logging in to the FTP server over SSL or TLS communication only.

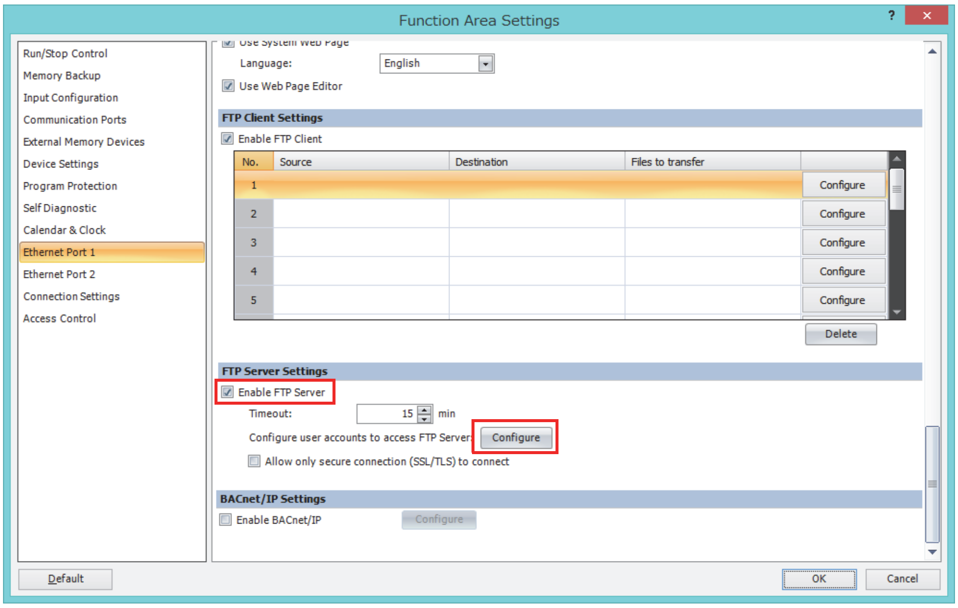
User Account Settings

Creating User Accounts

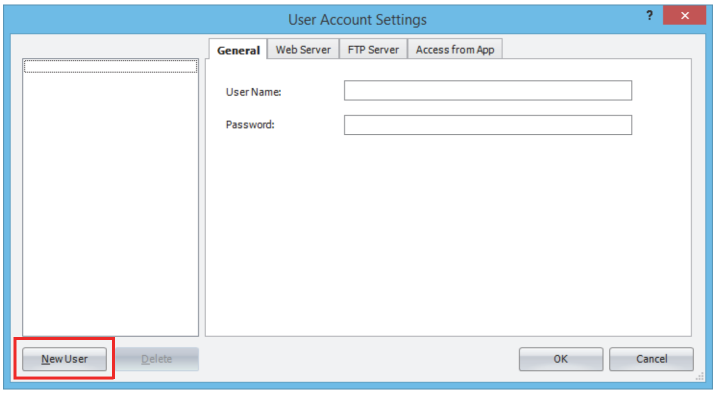
● Operation procedure

1. In the **Function Area Settings** dialog box, under **FTP Server Settings**, select the **Enable FTP Server** check box.

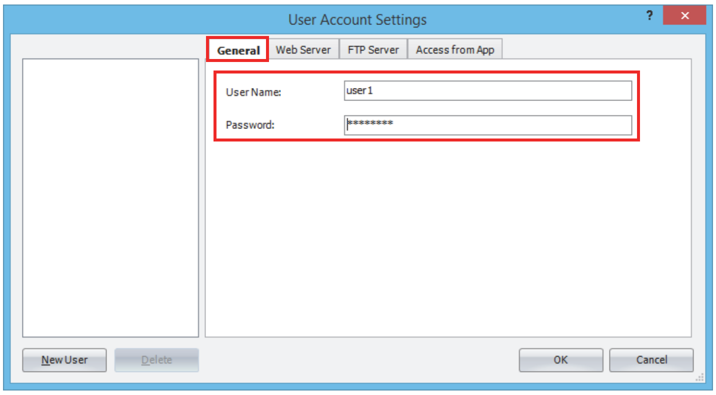
2. Click **Configure**.
The **User Account Settings** dialog box is displayed.



3. Click **New User**.
A user account is created in the list.

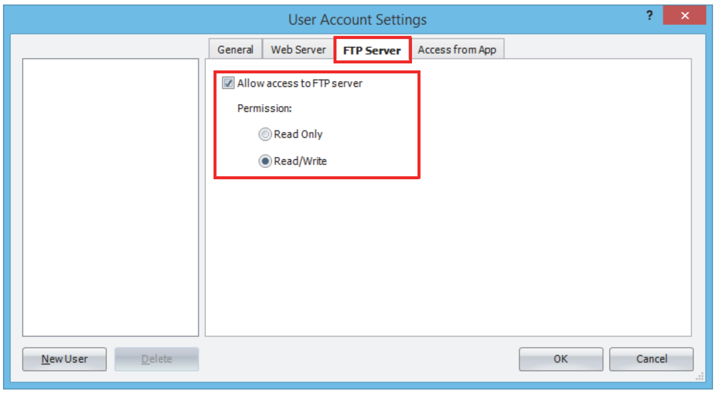


4. On the **General** tab, set **User Name** and **Password**.



5. Click the **FTP Server** tab and select the **Allow access to FTP server** check box.

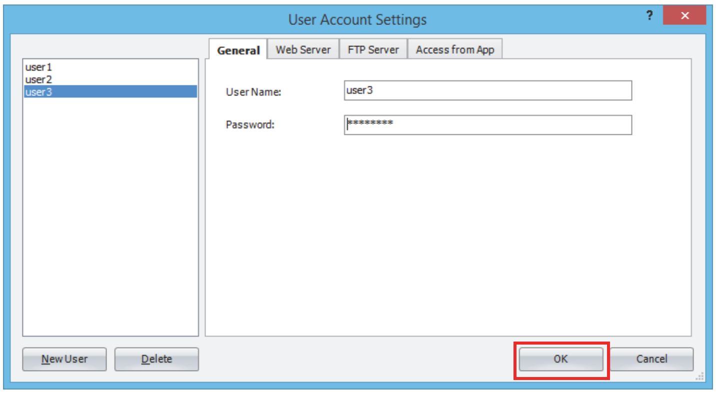
6. Under **Permission**, select **Read Only** or **Read/Write**.



7. Repeat the steps 2 to 6 and create the required number of user accounts.

Note: If you select a user account on the list, you can edit the settings of the selected user account on the **General** and **FTP Server** tabs.

8. Click **OK**.

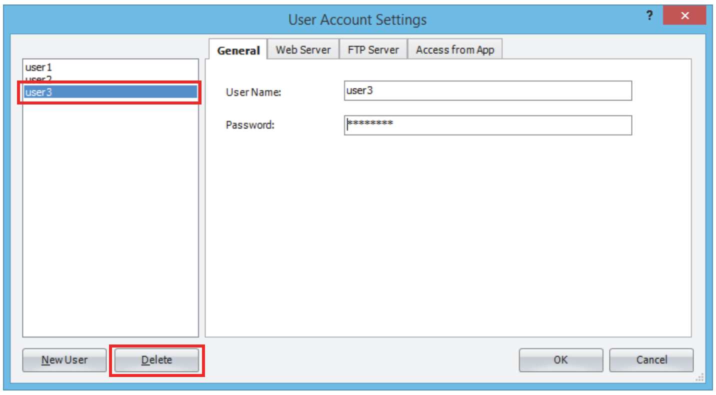


This concludes creating user accounts.

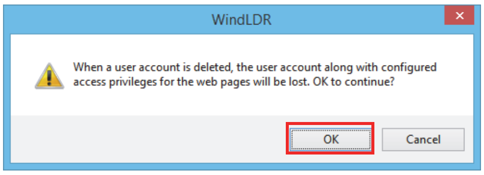
Deleting a Created User Account

● Operation procedure

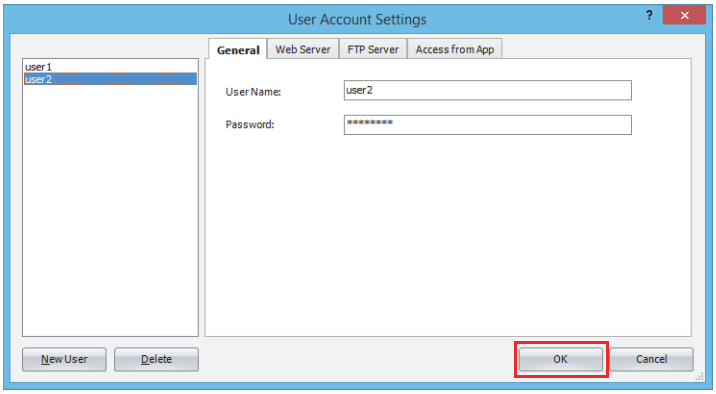
1. Select the user account on the list and click **Delete**.
A confirmation message is displayed.



2. Click **OK**.



3. Click **OK**.



This concludes deleting a user account.

FTP Client

The FTP client is available with the Plus CPU module only.

FTP Client Overview

What the FTP Client Can Do

The FTP client allows the Plus CPU module to act as an FTP client and copy or move files between the SD memory card inserted into the Plus CPU module and an FTP server.

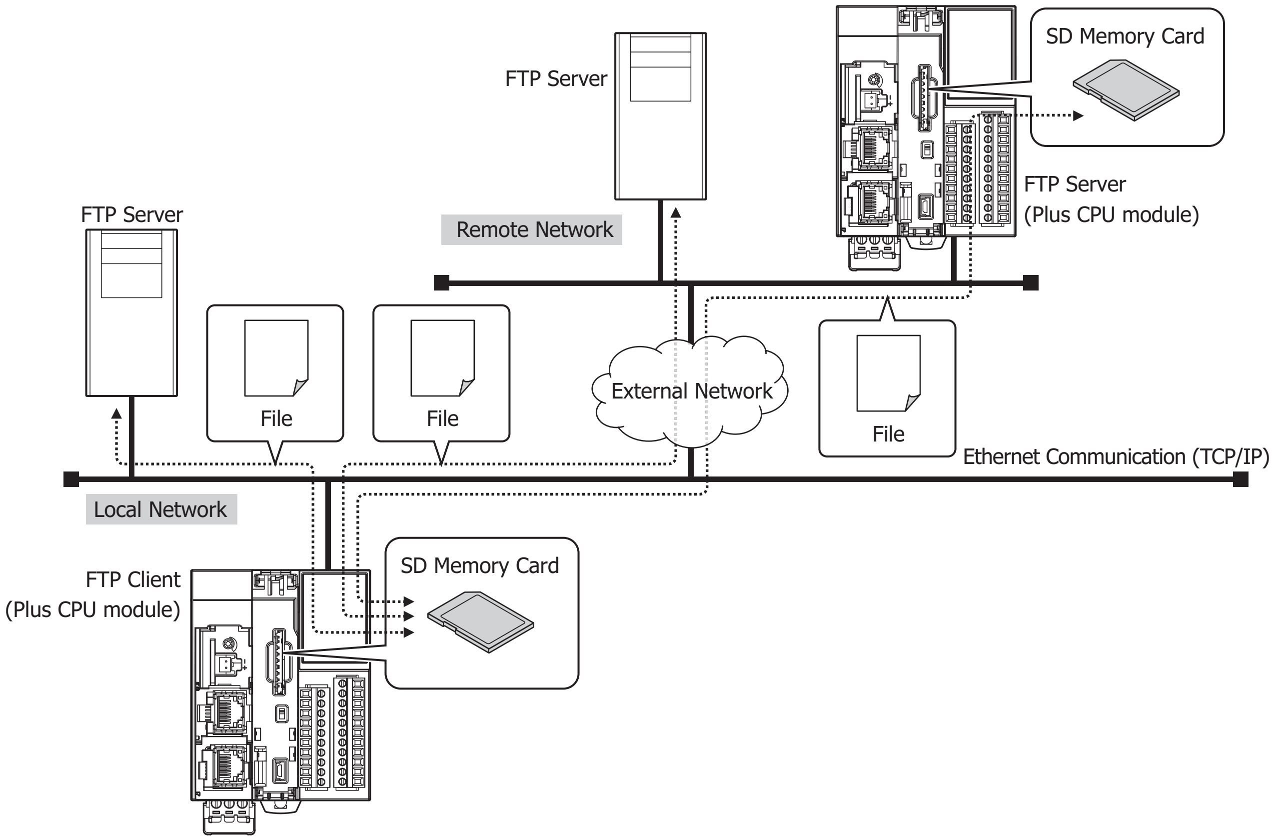
The Plus CPU module can perform the following actions as an FTP client.

- Copy or move files from the SD memory card inserted into the Plus CPU module to an FTP server.
- Copy files from an FTP server to the SD memory card inserted into the Plus CPU module.

Example of System Configuration

The following diagram shows an example of a system configuration for using the FTP client.

Connect Ethernet port 1 of the Plus CPU module to the network.



Configure the Ethernet settings of the Plus CPU module (IP address, subnet mask, and default gateway), and connect it to the local network.

Register the destination FTP server to the Plus CPU module.

When the operating conditions for the FTP client function (file transfer settings) are satisfied, the following content is executed.

- Copy or move files from the SD memory card inserted into the Plus CPU module to an FTP server.
- Copy files from an FTP server to the SD memory card inserted into the Plus CPU module.

Files Handled by the FTP Client

All files on the SD memory card inserted into the Plus CPU module can be accessed with the FTP client. This section shows the folder and file structure of the SD memory card.

SD Memory Card

Arbitrary: File names and folder names that can be specified by the function area settings or instruction parameters.
Fixed: File names and folder names that are fixed by the system.

- **FCDATA01** (Fixed)
  - **autoexec.ini** (Fixed) — The configuration file that defines actions taken when the Plus CPU module is turned on and when special internal relays are turned on to upload or download. This file is created using WindLDR, but it can also be edited using a text editor such as Notepad.
  - **DATALOG** (Fixed) — The folder used by the DLOG instruction.
    - **RESULT** (Arbitrary) — Log files are saved here. The folder name is specified by S1 in the DLOG instruction. The desired folder name up to 8 single-byte alphanumeric characters can be set.
      - **20111202.csv** — The CSV log file that is created when the DLOG instruction is executed. For the Plus CPU module, a folder is created and given the date (YYYYMMDD) as its name and then files are saved in that folder. For details, see Chapter 25 "DLOG (Data Log)" in the "FC6A Series MICROSmart Ladder Programming Manual". (Example: 20111202.csv)
  - **PROGRAM** (Fixed) — The folder used for downloading and uploading the user program. ZLD files are saved here.
    - **fc6a01.zld** (Arbitrary) — A zld file consists of the user program, program comments, and the system software data for the CPU module. This file can be created by saving the zld file in WindLDR or by uploading the user program to the SD memory card.
  - **RECIPE** (Fixed) — The folder used by the recipe function. Recipe files are saved here.
    - **RCP0001.csv** (Arbitrary) — The files in which recipe values are saved. Any file name up to 7 single-byte alphanumeric characters can be set. By default, the file names are automatically assigned as RCP**** (where **** is 0001 to 9999). (Example: RCP0001.csv)
  - **TRACE** (Fixed) — The folder used by the TRACE instruction.
    - **RESULT** (Arbitrary) — Log files are saved here. The folder name is specified by S1 in the TRACE instruction. The desired file name up to 8 single-byte alphanumeric characters can be set.
      - **20111202.csv** — The CSV log file that is created when the TRACE instruction is executed. For the Plus CPU module, a folder is created and given the date (YYYYMMDD) as its name and then files are saved in that folder. For details, see Chapter 25 "TRACE (Data Trace)" in the "FC6A Series MICROSmart Ladder Programming Manual". (Example: 20111202.csv)
  - **WEBPAGE** (Fixed) — The folder used by the web server. For details, see "Save Locations of Web Pages" on page 13-2.

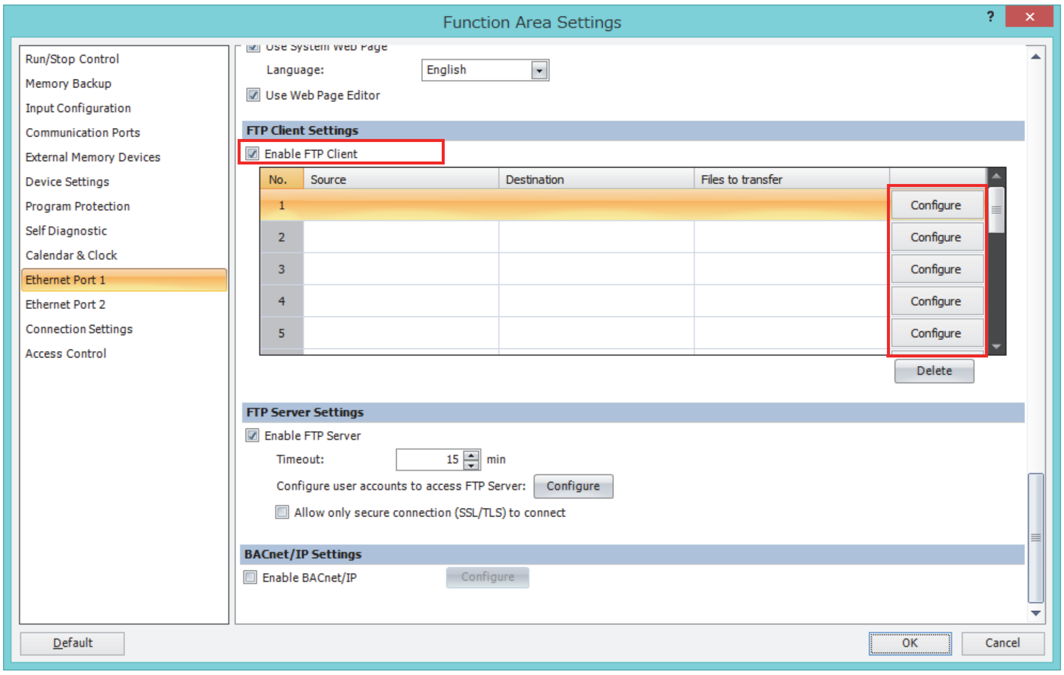
Supported Commands and Connection Methods

- Commands defined in RFC 959 are supported.
- Active mode and passive mode are supported.
- FTPS (FTP over SSL/TLS) is supported.

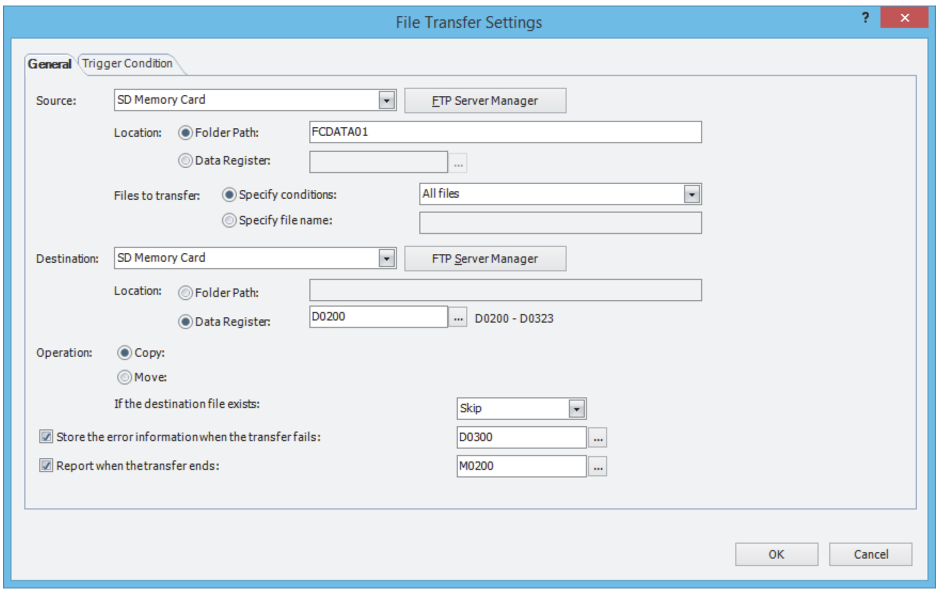
Note: The recommended FTP server software is Microsoft Internet Information Services.

Programming WindLDR

1. On the **Configuration** tab, in the **Function Area Settings** group, click **Ethernet Port 1**.
2. Select the **Enable FTP Client** check box.
3. Click **Configure** for the FTP client to configure on the FTP client registration list.
   The **File Transfer Settings** dialog box is displayed.



4. Configure the settings on the **General** and **Trigger Condition** tabs.



5. Click **OK**.

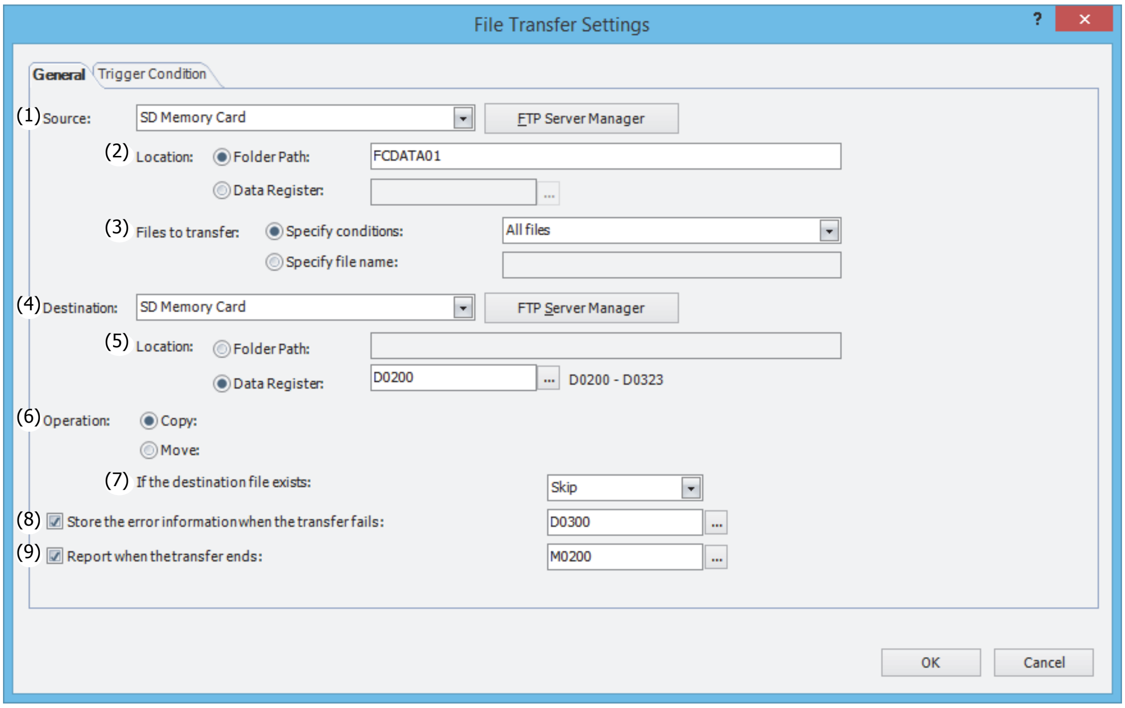
6. Repeat the steps 3 to 5 and create the required number of FTP client settings.

7. Download the user program to the Plus CPU module.

This concludes configuring the FTP client settings.

File Transfer Settings

General tab



(1) Source

Select **SD Memory Card** or **(FTP Server)** where the files that will be copied or moved are saved.

[Number]+[.(period)]+[Server name] as set in FTP Server Manager is displayed for (FTP Server).

Example: When [Number] is 1 and [Server name] is TestServer1

"1.TestServer1"

Click **FTP Server Manager** to display the FTP Server Manager. The FTP Server Manager can be used to add and modify FTP servers. For details, see "FTP Server Manager" on page 10-17.

(2) Location

Select the specification method of the location where the files that will be copied or moved are saved.

Folder Path/File Path : Specify the folder path or file path of the location where the files that will be copied or moved are saved with a character string. The maximum length of the character string is 247 single-byte alphanumeric characters.

Example: To copy or move files in the "DATALOG" folder in the "FCDATA01" folder on the SD memory card

FCDATA01\DATALOG

Data Register : Specify a word device for reading the data that will be used as the folder path of the location where the files that will be copied or moved are saved. The values are read in order starting from the specified data register, and the data up to the NULL terminating character (0x00) is handled as the character data of the folder path or file path.

(3) Files to transfer

Select the specification method of the files that will be copied or moved.

Specify conditions : Specify the files that will be copied or moved with a condition. Select the condition from the following.

All files, Latest file only, Files updated within 24 Hours, Files updated within 7 days

Specify file name : Specify the file that will be copied or moved with a character string. The maximum length of the character string is 247 single-byte alphanumeric characters.

(4) Destination

Select **SD Memory Card** or **(FTP Server)** where the copied or moved files will be saved.

[Number]+[.(period)]+[Server name] as set in FTP Server Manager is displayed for (FTP Server).

Example: When [Number] is 1 and [Server name] is TestServer1

"1.TestServer1"

Click **FTP Server Manager** to display the FTP Server Manager. The FTP Server Manager can be used to add and modify FTP servers. For details, see "FTP Server Manager" on page 10-17.

(5) Location

Select the specification method of the destination where the copied or moved files will be saved.

Folder Path : Specify the folder path where the copied or moved files will be saved with a character string. The maximum length of the character string is 247 single-byte alphanumeric characters.

Example: When saving the copied or moved files to the "DATALOG" folder in the "20170423" folder on the FTP server

20170423\DATALOG

Data Register : Specify a word device for reading the data that will be used as the folder path of the location where the copied or moved files will be saved. The values are read in order starting from the specified data register, and the data up to the NULL terminating character (0x00) is handled as the character data of the folder path.

(6) Operation

Select the transfer method of the files from **Copy** or **Move**. Only **Copy** can be selected when **(FTP Server)** is selected as **Source**.

(7) If the destination file exists

Select the processing method when a file with the same name already exists in the transfer destination from **Skip** or **Overwrite**.

(8) Store the error information when the transfer fails

To store the error information in a device address when an error occurs while copying or moving files, select this check box and specify the device address to which the error information will be written. When copying or moving files is started, zero is written to the device address. If an error occurs, one is written to the corresponding bit. The following error information is stored in each bit of the specified device address.

Bit Position	Error Information	Details	Countermeasure
0	SD memory card access error	• SD memory card specified as the source or destination is not inserted. • SD memory card specified as the source or destination cannot not be accessed.	• Insert an SD memory card that can be accessed. • Mount the SD memory card.
1	SD memory card read/write error	• The contents of the folder or file cannot be read from the SD memory card specified as the source or destination. • Creating a folder, writing a file, or deleting file cannot be done in the SD memory card specified as the source or the destination.	• Insert an SD memory card that can be read/ written. • Insert an SD memory card that has sufficient available capacity.
2	FTP server connection error	FTP server that is specified as the transfer source or destination cannot be accessed within the time specified as the connection timeout.	• Confirm that LAN cable is properly connected. • Confirm that the network settings of the Plus CPU module are correct. • Confirm that the IP address or port number of the FTP server are correct.

Bit Position	Error Information	Details	Countermeasure
3	FTP server authentication error	The user name or password is incorrect and the FTP server cannot be accessed.	Confirm that the user name or password is correct.
4	FTP server command error	An error was returned against the command sent to the FTP server.	Contact administrator of the FTP server.
5	FTP server transfer error	There is no response from the FTP server within the time specified as the transfer timeout.	Contact administrator of the FTP server.
6	**— Reserved —**	—	—
7	Other errors	Other errors	Contact administrator of the FTP server.
8 to 15	**— Reserved —**	—	—

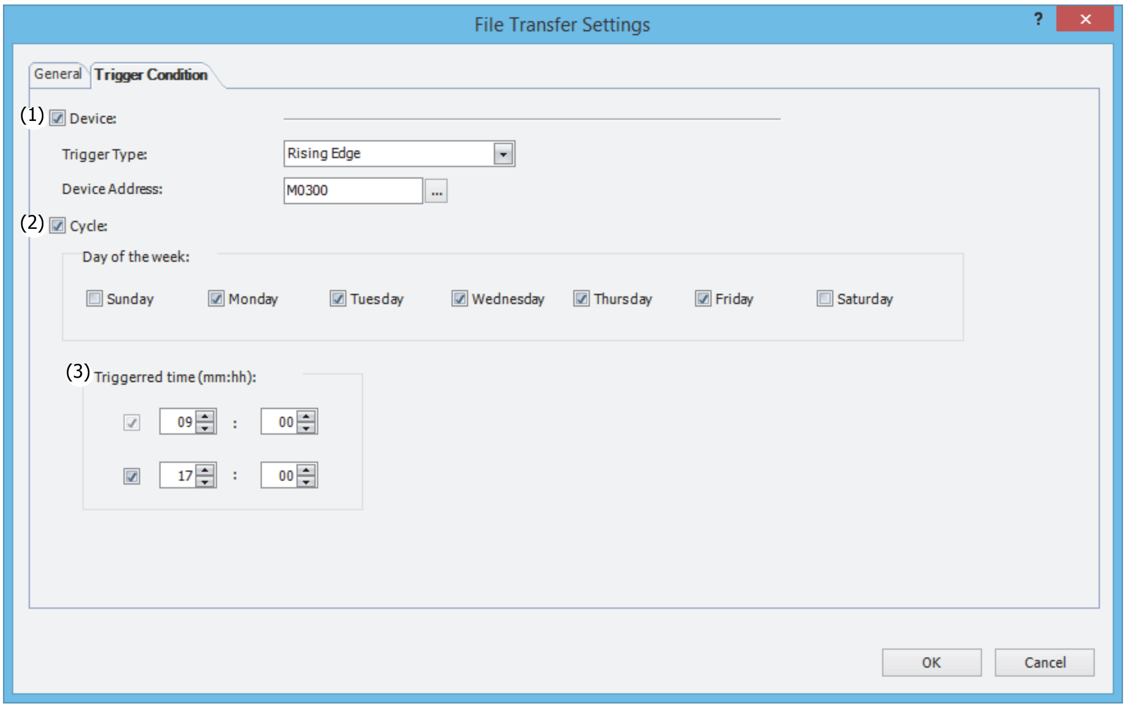
(9) Report when the transfer ends

To provide a notification when copying or moving files is finished, select this check box and specify the device address to which the notification will be written.

Notes:

- Files cannot be copied or moved from one FTP server to another FTP server.
- Files cannot be copied or moved between SD memory cards inserted into the Plus CPU modules.
- Subfolders cannot be copied or moved.
- The following single-byte characters cannot be used in folder and file paths.
  ; : * ? " < > |
- When **Data Register** is selected and the folder path limitation is exceeded or invalid characters are used, the folder path is determined as follows:
  - If the folder or file path string exceeds the maximum number of device address points (no NULL), the character string stored from the top device address to the maximum device address becomes the folder path.
  - If the folder path contains a character that cannot be used, the character string up to that invalid character becomes the folder path.

Trigger Condition tab



(1) Device

Select this check box to specify the condition for copying or moving files with a device address.

Select **Rising Edge** or **Falling Edge** and specify the device to act as the condition. Only an internal relay can be used.

(2) Cycle

To specify the condition to copy or move files with days of the week and time, select this check box and the check boxes for the days of the week to execute the file transfer.

(3) Triggerred Time

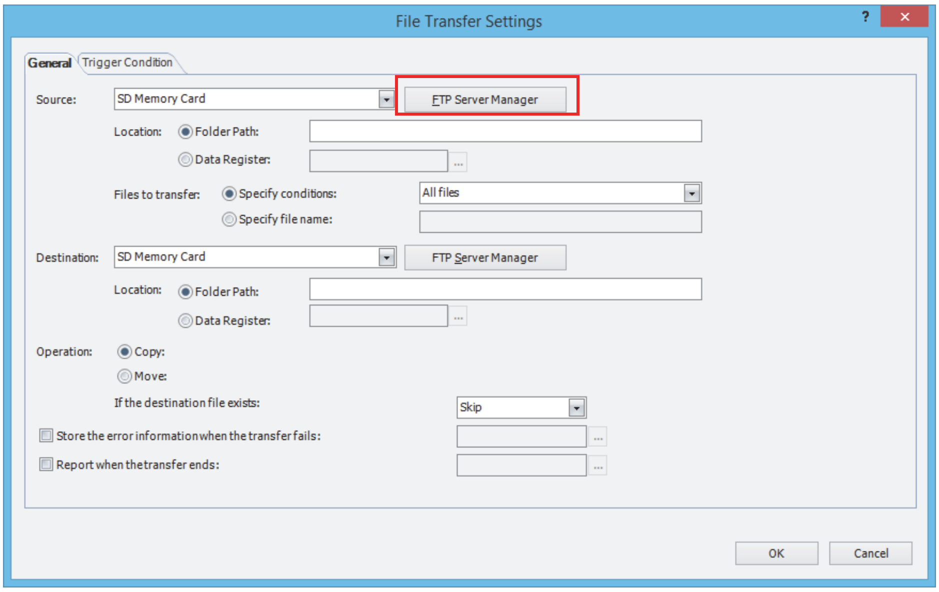
To specify the condition to copy or move files with a time, select this check box and specify the time. Up to two times can be set between 0:00 and 24:00 (24-hour time).

FTP Server Settings

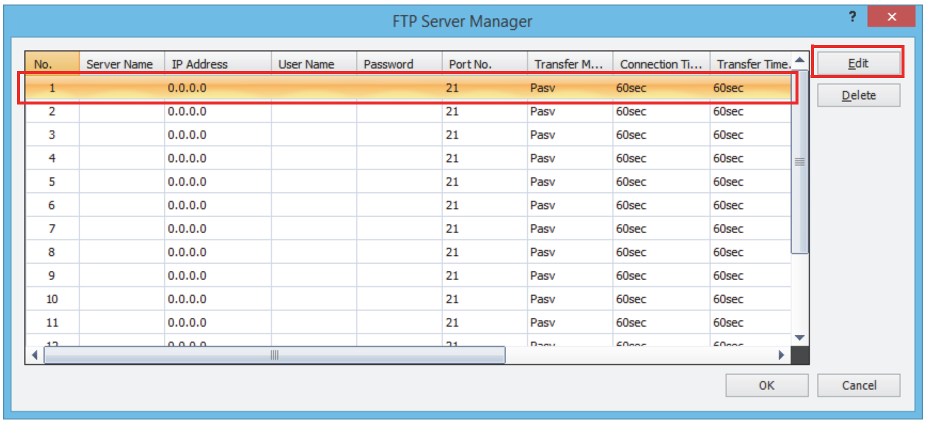
Configure the destination FTP servers.

• Operation procedure

1. In the **File Transfer Settings** dialog box, on the **General** tab, click **FTP Server Manager**.
   The FTP Server Manager is displayed.

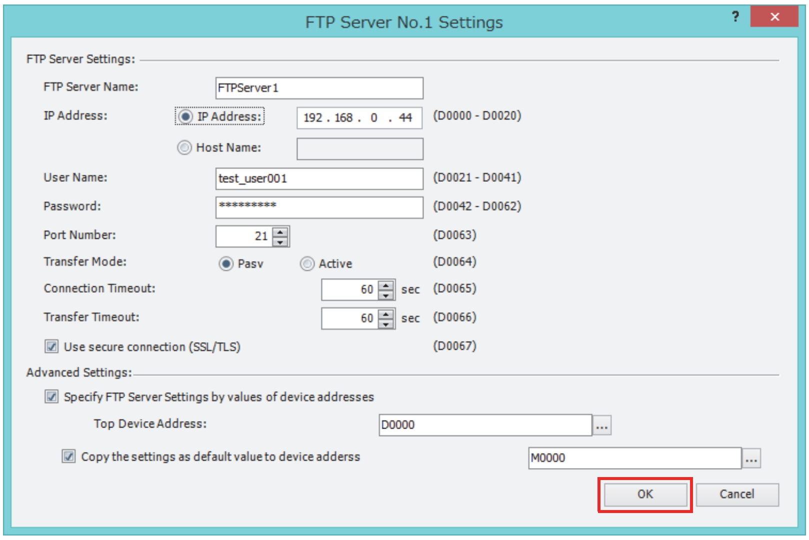


2. Select the number to use to register the FTP server on the FTP server list and click **Edit**.
   The **FTP Server No. n Settings** dialog box for the selected number is displayed (where n: Number in FTP Server Manager).



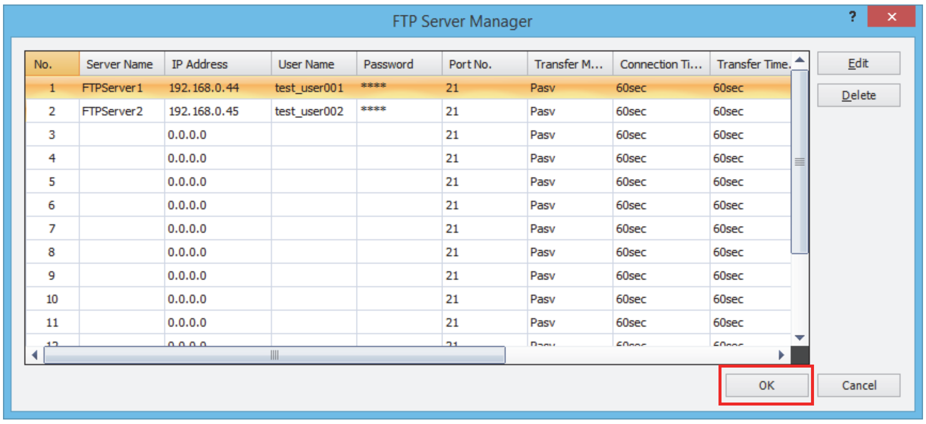
Note: To delete a registered FTP server, select the FTP server in the FTP server list, and then click **Delete**.

3. Configure each setting of the FTP server and click **OK**.



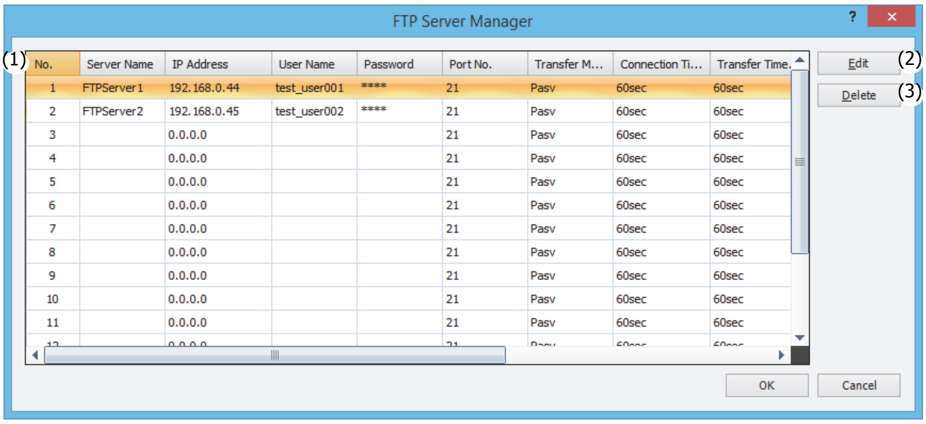
4. Repeat the steps 2 to 3 to register the required number of FTP servers.

5. Click **OK**.



This concludes configuring the FTP server settings.

FTP Server Manager



(1) FTP Server List

FTP server settings corresponding to each number are listed. Double click a cell to display the **FTP Server No. n Settings** dialog box (where n: Number in FTP Server Manager) where the settings can be edited. For details, see "Individual FTP Server Settings" on page 10-18.

No.	:	Displays the number (1 to 16) used to manage the FTP server settings.
Server Name	:	Displays the name of the FTP server.
IP Address	:	Displays the IP address of the FTP server.
User Name	:	Displays the user name for the FTP server.
Password	:	Displays the password for the FTP server masked with * (asterisks).
Port No.	:	Displays the port number of the FTP server.
Transfer Mode	:	Displays the transfer mode of the FTP server.
Connection Timeout	:	Displays the time to automatically stop the connection process if there is no response from the FTP server after **Trigger Condition** set in **File Transfer Settings** is satisfied and the connection is attempted.
Transfer Timeout	:	Displays the time to automatically stop the file transfer process if there is no response from the FTP server after **Trigger Condition** set in **File Transfer Settings** is satisfied and the data connection is established.
Top Device Address	:	Displays the Top device address when the FTP server settings are specified with device address values.

Copy the settings as default value to device address:

Displays the bit device that acts as the condition to copy the FTP server settings as initial values.

(2) Edit button

Registers or changes the settings of the selected number.

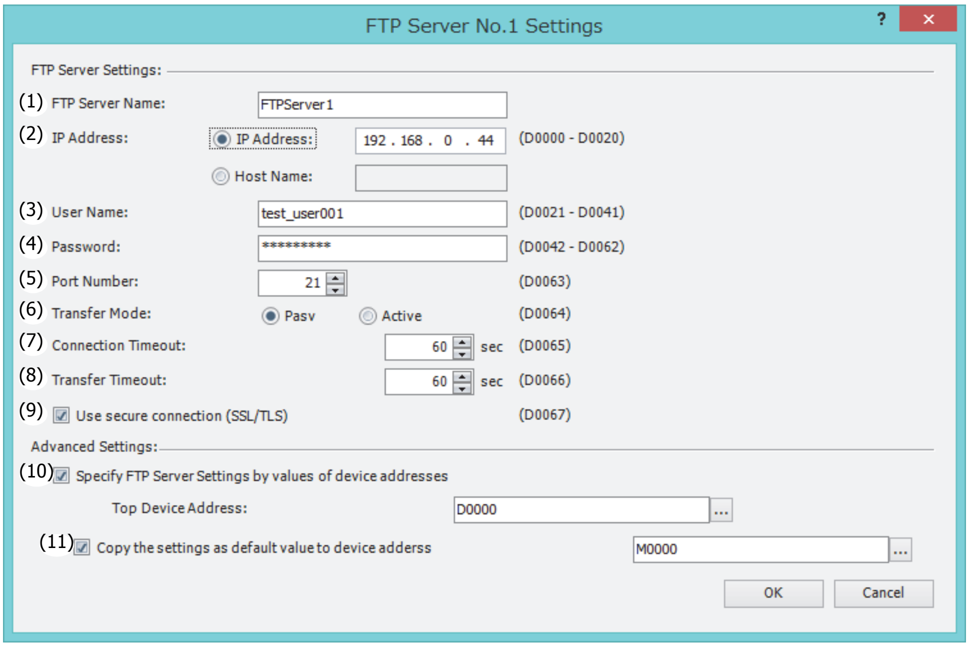
Select a number in the FTP Server List (1) and click this button to display the **FTP Server No. n Settings** dialog box. The settings of the **FTP Server No. n Settings** dialog box are reflected in the selected number.

For details, see "Individual FTP Server Settings" on page 10-18.

(3) Delete button

Deletes the settings with the number selected in FTP Server List (1).

Individual FTP Server Settings



(1) FTP Server Name

Enter the name of the FTP server. The maximum length of the string is 40 characters.

The default is "FTPServern". (n: FTP Server Manager number)

(2) IP Address

Select the identification method of the FTP server.

IP Address : Enter the IP address of the FTP server.
The format is "xxx.xxx.xxx.xxx". "xxx" stands for a numeric value from 0 to 255.

Host Name : Enter the host name of the FTP server.
The maximum length of the host name is 40 single-byte characters. Only alphanumeric characters and symbols can be used.

(3) User Name

Enter the name of the user account for the FTP server.

The maximum length of the user name is 40 single-byte characters. Only alphanumeric characters and symbols can be used.

(4) Password

Enter the password for the FTP server.

The maximum length of the password is 40 single-byte characters. Only alphanumeric characters and symbols can be used.

(5) Port Number

Specify the port number (0 to 65535) of the FTP server.

(6) Transfer Mode

Select the transfer mode of the FTP server.

Pasv : Establishes the data connection using passive mode.

Active : Establishes the data connection using active mode.

(7) Connection Timeout

Specify the time (10 to 300 seconds) to automatically stop the connection process if there is no response from the FTP server after **Trigger Condition** set in **File Transfer Settings** is satisfied and the connection is attempted.

(8) Transfer Timeout

Specify the time (10 to 300 seconds) to automatically stop the file transfer process if there is no response from the FTP server after **Trigger Condition** set in **File Transfer Settings** is satisfied and the data connection is established.

(9) Use secure connection (SSL/TLS)

Select this check box when SSL or TLS communication is required by the FTP server to use.

(10) Specify FTP Server Settings by values of device addresses

Select this check box to specify the **FTP Server Settings** with device addresses.

Specify the device addresses to use with **Top Device Address**. The settings in the **FTP Server Settings** are allocated to device addresses starting from the address number of the specified device address.

(11) Copy the settings as default value to device address

Select this check box to copy the settings in the **FTP Server Settings** to the device addresses as the initial values.

Specify a device address acting as the condition for copying the settings.

When 1 is written to this device address, the values set in the **FTP Server Settings** are written to the allocated device addresses starting from the address number of the device address set with **Top Device Address**.

Address Allocations When Specifying the FTP Server with Device Address Values

When the **Specify FTP Server Settings by values of device addresses** check box is selected, the settings in the **FTP Server Settings** are allocated to device addresses starting from the address number of the device address set with **Top Device Address**. The details of allocated device addresses are as follows.

Settings	Address Number	Number of Words	Data Type
IP Address or Host Name	+0 to +20	21*1*2	Numeric value
User Name	+21 to +41	21*2*3	Character string
Password	+42 to +62	21*2*3	Character string
Port Number	+63	1	Numeric value
Transfer Mode (0: Pasv, 1: Active)	+64	1	Numeric value
Connection Timeout (second)	+65	1	Numeric value
Transfer Timeout (second)	+66	1	Numeric value
Use Secure Connection (SSL/TLS) (0: SSL/TLS is used, 1: SSL/TLS is not used)	+67	1	Numeric value

*1 For an IP address, only the first four words are used, and the remaining 17 words are a reserved region.

*2 The 21st word is handled as the NULL terminating character (0x00), regardless of the value of the data register.

*3 When a character string to use is shorter than 20 words, add the NULL terminating character (0x00) as the end of the character string.

11: SEND PING FUNCTION

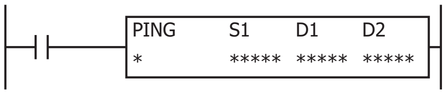
Introduction

This chapter describes the PING instruction that sends a ping packet to the specified remote host to check if communication is possible at the Internet Protocol (IP) layer.

Use the PING instruction to execute the send PING function.

PING (Ping)

The PING instruction sends a ping packet to the specified remote host. For the Plus CPU module, Ethernet port 1 or 2 can be selected.



Operation Details

When the input to the PING instruction is on, the PING instruction sends a ping packet to the remote host specified by S1. Once the FC6A Series MICROSmart receives the reply from the remote host, the completion output specified by D1 is turned on, and the operation status (operation transition state and error code) is stored in the device specified by D2. D2+1 is reserved for the system.

The PING instruction is executed when the input is on, and it sends a PING request. While that input remains on, the same PING instruction is executed after the ping is sent. To avoid this, use the PING instruction in combination with the SOTU (single output up) instruction or the SOTD (single output down) instruction so that the PING instruction input is on for only one scan. For the SOTU instruction and the SOTD instruction, see Chapter 4 "SOTU and SOTD (Single Output Up and Down)" in the "FC6A Series MICROSmart Ladder Programming Manual".

In order to use the PING instruction, the FC6A Series MICROSmart remote host list must be created in advance. For details about the remote host list, see "Remote Host List" on page 3-20.

Valid Devices

	Device	Function	I	Q	M	R	T	C	D	P	Constant	Repeat
S1	Source 1	Remote host number	-	-	-	-	-	-	X	-	1 - 255	-
D1	Destination 1	Completion output	-	X	X*1	-	-	-	-	-	-	-
D2	Destination 2	Operation status	-	-	-	-	-	-	X	-	-	-

*1 Special internal relays cannot be used.

Devices in PING Instructions

S1 (Source 1)

Specify the remote host number. A constant or data register can be specified.

D1 (Destination 1)

Specify an internal relay or output as the completion output bit. When the FC6A Series MICROSmart receives a reply from the remote host, the completion output bit is turned on. The completion output bit is also turned on when there is no reply from the remote host within the specified timeout that is configured in the **Function Area Settings**. For details about the timeout setting, see "Ping Settings" on page 3-13.

D2 (Destination 2)

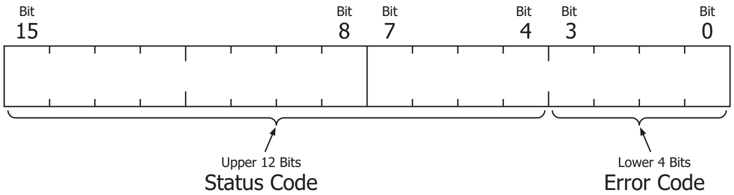
Specify a data register as the operation status. Destination 2 occupies two consecutive data registers starting with the data register specified by D2. The operation status is stored in D2. D2+1 is reserved for the system. Specify the starting number so that the device range is not exceeded.

Notes:

- When a data register is specified as S1, do not change the data register value while PING instruction is executed.
- The FC6A Series MICROSmart does not respond to a ping packet that is sent by itself.

The operation status includes the operation transition state (status code) and the error detail (error code). The status code can be obtained by masking the least 4 bits of the operation status. The error code is stored in the least 4 bits of the operation status. When the value of the operation status data register is divided by 16, the remainder is the error code.

If the value of the operation status data register is 66:
66 ÷ 16 = 4 with a remainder of 2, so the error code is 2.



Status Code	Operation Transition State	Description
16 (000000010000)	Preparing transmission	The interval from turning on the PING instruction input to when the ping packet is sent to the specified remote host
32 (000000100000)	Waiting for response	From sending the ping packet to the specified remote host, until the response is received
64 (000001000000)	PING instruction complete	The execution of the PING instruction is completed, allowing for the next transmission to be processed

Note: If the status code is anything other than the codes listed above, an error of PING instruction is suspected.

Error Code	Error Details	Completion Output
0 (0000)	Normal	ON (The status code is 64)
1 (0001)	-	-
2 (0010)	Timeout error	ON
3 (0011)	The host name of the target remote host could not be resolved with DNS	ON
4 (0100)	The specified remote host does not exist in the remote host list	ON
5 (0101)	Multiple PING instructions are executed simultaneously	ON
6 (0110)	Parameter error	ON

12: SEND E-MAIL FUNCTION

Introduction

This chapter describes the EMAIL instruction that sends preregistered e-mails.

Overview

Ethernet port 1 of the Plus CPU module and the HMI-Ethernet port of the HMI module support the send e-mail function.

Up to 255 types of e-mails can be sent by executing the EMAIL instruction.

EMAIL Instruction (Send E-mail)

The EMAIL instruction sends an e-mail.

Operation Details

When the input is on, e-mail of the e-mail number set by S1 is sent from Ethernet port 1 of the Plus CPU module or the HMI-Ethernet port of the HMI module. Once the FC6A Series MICROSmart receives a response from the sending e-mail server, the completion output specified by D1 is turned on, and the operation status (operation transition state and error code) is stored in the device specified by D2. D2+1 is reserved for the system.

The EMAIL instruction is executed when the input to the instruction is on. When this input continues to be on, the same EMAIL instruction is executed after the e-mail is sent. To avoid this, use the EMAIL instruction in combination with the SOTU (single output up) instruction or the SOTD (single output down) instruction so that the EMAIL instruction start input is on for only one scan. For the SOTU instruction and the SOTD instruction, see Chapter 4 "SOTU and SOTD (Single Output Up and Down)" in the "FC6A Series MICROSmart Ladder Programming Manual".

To use the EMAIL instruction, you must first configure the mail server settings and create the FC6A Series MICROSmart e-mails in advance. When using the HMI-Ethernet port of the HMI module, click the HMI module that has been inserted into the module configuration area, click **Configure**, and configure the e-mail settings in the displayed **HMI Module Configuration** dialog box. When using Ethernet port 1 of the Plus CPU module, configure the e-mail settings under **Ethernet Port 1** in the **Function Area Settings**.

Valid Devices

	Device	Function	I	Q	M	R	T	C	D	P	Constant	Repeat
S1	Source 1	E-mail number	-	-	-	-	-	-	X*1	-	1 - 255	-
D1	Destination 1	Completion output	-	X	X*2	-	-	-	-	-	-	-
D2	Destination 2	Operation status	-	-	-	-	-	-	X*1	-	-	-

*1 Special data registers cannot be used.

*2 Special internal relays cannot be used.

Devices in EMAIL Instruction

Ethernet Port Number Settings

Configure the settings of the Ethernet port that will be used to send e-mails. Configure Ethernet port 1 of the Plus CPU module or the HMI-Ethernet port of the HMI module.

S1 (Source 1)

Specify the e-mail number. A constant or data register can be specified.

D1 (Destination 1)

Specify an internal relay or output as the completion output bit. When the FC6A Series MICROSmart receives a reply from the sending e-mail server, the completion output bit is turned on. The completion output bit is also turned on when there is no reply from the sending e-mail server within the timeout.

D2 (Destination 2)

Specify a data register as the operation status. Destination 2 occupies two consecutive data registers starting with the data register specified by D2. The operation status is stored in D2. D2+1 is reserved for the system. Data registers D0000 to D7998, and D10000 to 55998 can be specified. Specify the starting number so that the device range is not exceeded.

Note: When a data register is specified as S1, do not change the data register value while EMAIL instruction is executed.

The operation status includes the operation transition state (status code) and the error detail (error code). The status code can be obtained by masking the least 4 bits of the operation status. The error code is stored in the least 4 bits of the operation status.

When the value of the operation status data register is divided by 16, the remainder is the error code.

If the value of the operation status data register is 66:
66 ÷ 16 = 4 with a remainder of 2, so the error code is 2.

Status Code	Operation Transition State	Description
16 (000000010000)	Preparing transmission	From turning on the start input for an EMAIL instruction, until the e-mail is sent out to the sending e-mail server.
32 (000000100000)	Waiting for response	From sending the e-mail to the sending e-mail server, until the response is received.
64 (000001000000)	EMAIL instruction complete	The execution of the EMAIL instruction is completed, allowing for the next transmission to be processed.

If the status code is anything other than the codes listed above, an error of EMAIL instruction is suspected.

Error Code	Error Details	Completion Output
0 (0000)	Normal	ON (The status code is 64)
1 (0001)	HMI module connection error • The HMI module is not connected • The HMI module is not operating correctly	ON
2 (0010)	Timeout error • Port number may be incorrect.	ON
3 (0011)	• The host name of the sending e-mail server could not be resolved with DNS. • The sending e-mail server could not be found. • The SMTP server does not support the authentication function. • Ethernet cable may be disconnected or broken. FC6A Series MICROSmart may not be connected to the network properly.	ON
4 (0100)	The specified e-mail is not configured.	ON
5 (0101)	Multiple EMAIL instructions are executed simultaneously.	ON
6 (0110)	Parameter error • If **Use data registers to configure E-mail settings** is enabled, the settings are not stored in data registers.	ON
7 (0111)	Other errors [*1]	ON
8 (1000)	Converting a data register value failed [*2]	ON
9 (1001)	The size of one log file that will be attached to the e-mail has exceeded the maximum file size configured for the DLOG/TRACE instruction.[*3]	ON
10 (1010)	The specified log file that will be attached does not exist.[*3]	ON
11 (1011)	The number of attached log files exceeds seven.[*4]	ON

*1 The detail of the error code 7 is stored in special data register D8457 (HMI-Ethernet port) or D8759 (Ethernet port 1).
*2 When converting a data register value fails, ---- is inserted in the e-mail body instead of data register value.

*3 For error code 9 or 10, the corresponding log file will not be attached to the e-mail, and the e-mail will be sent without an attachment.
For the configuration of the maximum log file size, see Chapter 5 "Functions and Settings" in the "FC6A Series MICROSmart User's Manual".
*4 E-mails in which seven files are attached will be sent.

E-mails can be sent to the recipients by the relay of the sending e-mail (SMTP) servers. Even when the value stored in D2 is 64 after the execution of an EMAIL instruction, the recipients of the e-mail may not be able to receive the e-mail if one of the following conditions is met. The FC6A Series MICROSmart also cannot judge that e-mails cannot be received.

- The SMTP server that relays the e-mails is not functioning normally.
- The e-mail recipient filters the in-coming e-mails by specifying the e-mail address or e-mail domain.

Special Data Registers

Confirm the error detail of EMAIL instruction

When the error code stored in D2 of EMAIL instruction is 7, the response from sending e-mail server can be confirmed with special data register. The allocation of special data registers is given in the following table.

Ethernet Port Used to Send the E-mail	Special Data Register That Stores the Detailed Error Code of the E-mail
HMI-Ethernet port of the HMI module	D8457
Ethernet port 1 of the Plus CPU module	D8759

The error code that the sending e-mail servers return could vary with each sending e-mail server. For details about the error code, contact the administrator of the sending e-mail server. An example of the error codes is as follows.

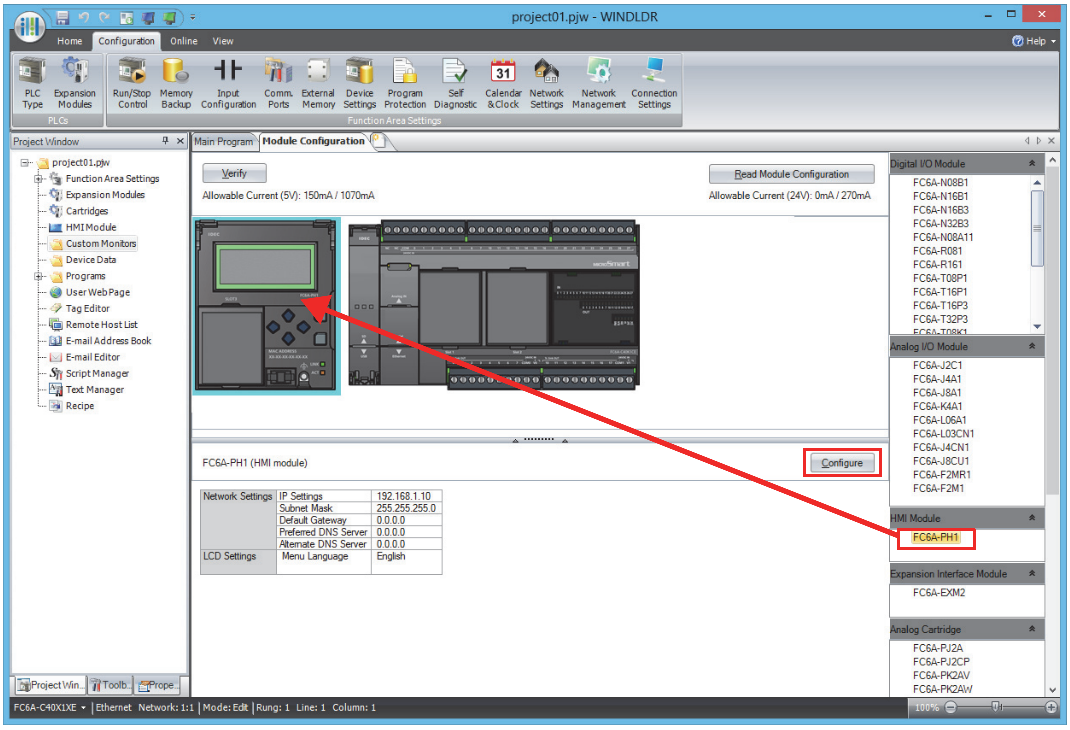
Error Code	Description	Possible Cause
451	The requested action is canceled	Sending e-mail server is not working normally
452	Insufficient system storage	The e-mail server storage is insufficient
500	Syntax error, unrecognizable command	The command is not recognized by the e-mail server
501	Parameter or argument syntax error	Invalid command parameter
502	Command not implemented	The command is not implemented on the e-mail server
504	Command parameter not implemented	The command parameter is not implemented on the e-mail server
521	Email not received	Email assumed to be spam
530	Access denied	The **Authentication is required to send e-mail** check box is not selected
535	Authentication error	Account name or password for the authentication is incorrect
550	The mailbox cannot be used	The command was refused due to the server policy
552	Allocated capacity has been exceeded	The message is too long
554	Transaction failure (or a no SMTP server response when starting the connection)	Sending e-mail server is not working normally

Programming WindLDR

1. Use the Module Configuration Editor to configure the HMI module EMAIL settings.
On the **Configuration** tab, in the **PLCs** group, select **Expansion Modules**.

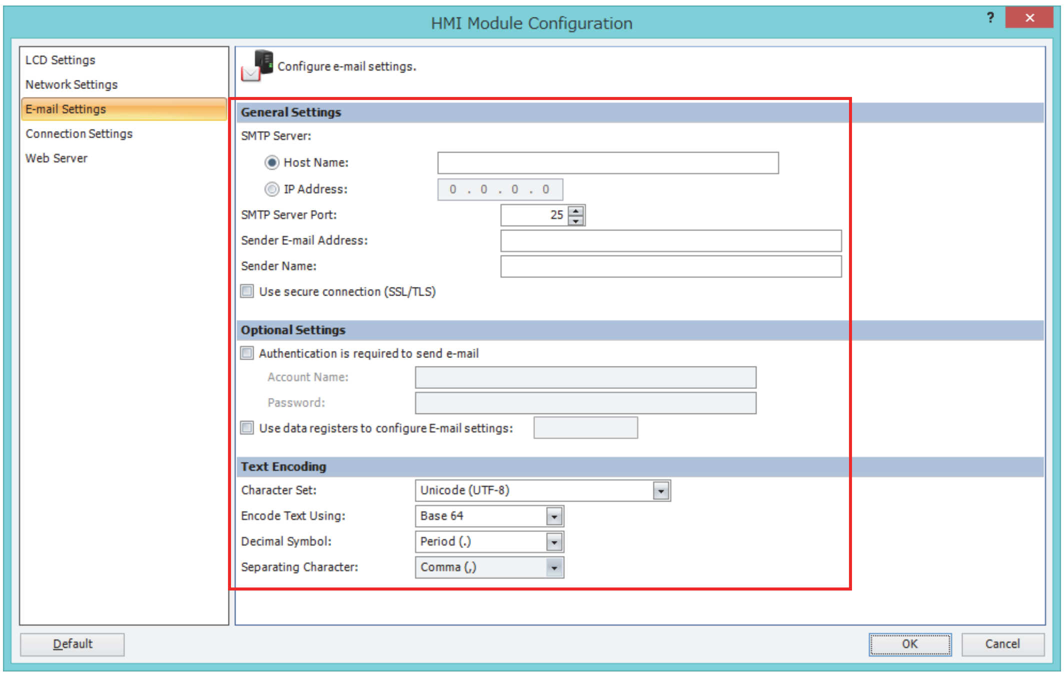
2. Click the inserted HMI module in the module configuration area and click **Configure**.
The **HMI Module Configuration** dialog box is displayed.

Note: You can also display the **HMI Module Configuration** dialog box by double-clicking **HMI Module** in the Project Window.



3. Click **E-mail Settings**.

4. Configure the settings in General Settings, Optional Settings, and Text Encoding.



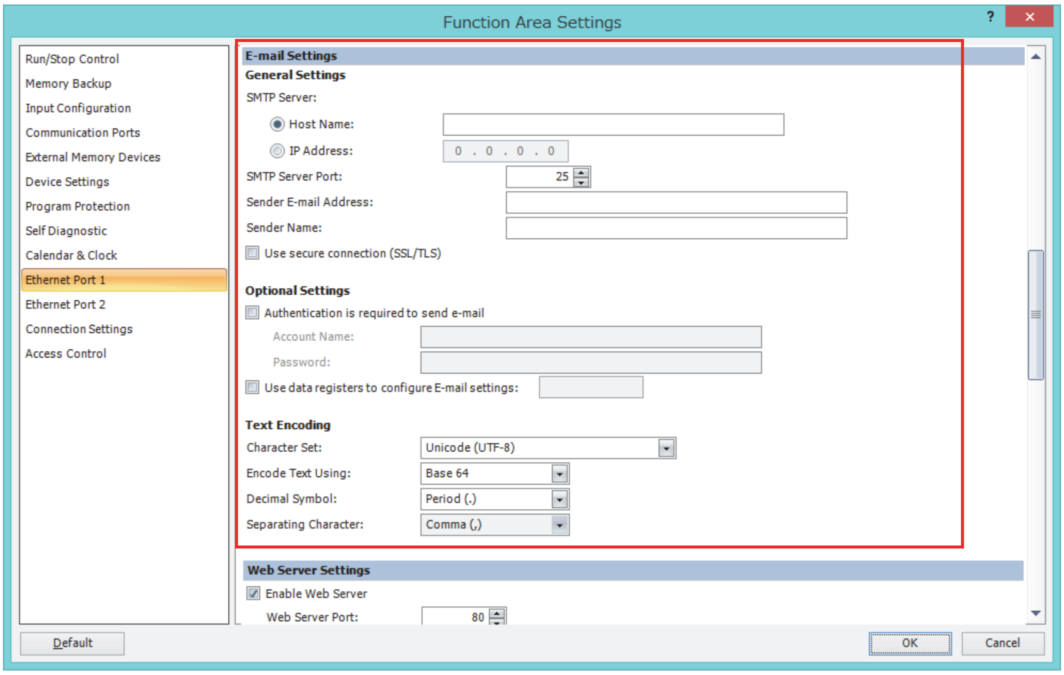
5. Download the user program to the FC6A Series MICROSmart.

This concludes configuring the e-mail settings.

Ethernet port 1 of the Plus CPU module

1. On the **Configuration** tab, in the **Function Area settings**, select **Ethernet Port 1**.

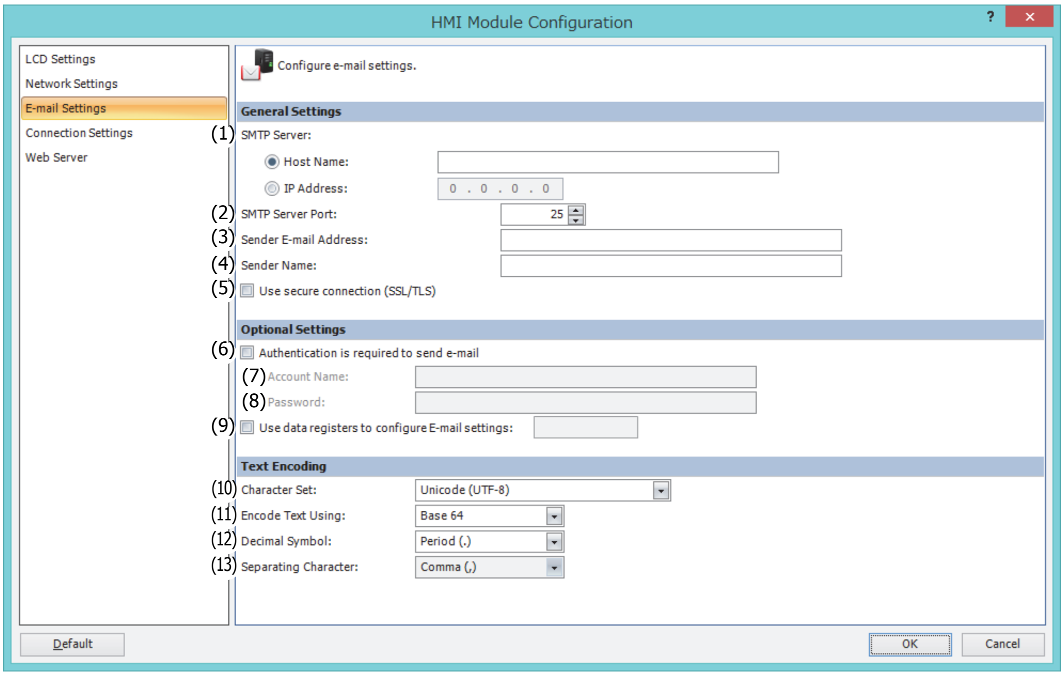
2. In **E-mail Settings**, configure the settings under General Settings, Optional Settings, and Text Encoding.



3. Download the user program to the Plus CPU module.

This concludes configuring the e-mail settings.

■ General Settings



(1) SMTP Server

Specify the IP address or the host name of the sending e-mail server that is used to send e-mails. A maximum of 40 ASCII characters can be entered.

(2) SMTP Server Port

Specify the port number of the sending e-mail server. Normally, SMTP uses port 25, SMTP-AUTH uses port 587, and SMTPs uses port 465.

(3) Sender E-mail Address

Specify the e-mail address that is included in the sender field of the e-mails to be sent from the FC6A Series MICROSmart. A maximum of 40 ASCII characters can be entered.

(4) Sender Name

Specify the name that is included in the sender field of the e-mails to be sent from the FC6A Series MICROSmart. A maximum of 40 ASCII characters can be entered.

(5) Use secure connection (SSL/TLS)

When SSL or TLS communication is required with the sending e-mail server that is used, select this check box.

Notes:

- When specifying the SMTP server with an IP address, 0 in the high order digits of the values is ignored. For example, when the IP address "192.168.1.234" and "192.168.001.234" are considered as the same IP address.
- When specifying the SMTP server with an SMTP server name, the host name of the SMTP server must be resolved using DNS. For DNS and DNS server settings, see "Network Settings" on page 3-3.

Configuration Example: The sender e-mail address and sender name are grouped and shown in the sender field of the e-mail. If the sender e-mail address is test@sample.com and the sender name is Test Mail, the text shown in the sender field will be as follows:

Test Mail <test@sample.com>

■ Optional Settings: SMTP Authentication

Depending on the SMTP server, SMTP-AUTH or SMTPs authentication may be required to send e-mails.

The FC6A Series MICROSmart supports SMTP-AUTH and SMTPs authentication, so if the SMTP server requires authentication, it can send e-mails using this function.

Contact the administrator of the SMTP server to confirm if it requires authentication.

(6) Authentication is required to send e-mail

Check if the sending e-mail server requires the SMTP authentication with the login method.

(7) Account Name

Specify the account name used for the login authentication. A maximum of 40 ASCII characters can be entered.

(8) Password

Specify the password used for the login authentication. A maximum of 40 ASCII characters can be entered.

(9) Use data registers to configure E-mail settings

When the check box for this option is selected, the basic settings and authentication settings can be configured using strings and numeric values stored in data registers. E-mails are sent using the settings stored in 107 words of data registers starting from the specified data register.

For each setting item, the start address and end address as well as details about the setting value are as follows.

Setting Item	Data Type	Number of Used Words	Start Address of Setting Item	End Address of Setting Item	Setting Value
SMTP Server	String (40 characters)	21*1	Starting data register	Starting data register+20	Specify the IP address or the host name of the sending e-mail server that is used to send e-mails as a string.*2
SMTP Server Port	Decimal value	1	Starting data register+21		Specify the port number of the sending e-mail server that is used to send e-mails as a decimal value.
Sender E-mail Address	String (40 characters)	21*1	Starting data register+22	Starting data register+42	Specify the e-mail address that is included in the sender field of the e-mails to be sent from the FC6A Series MICROSmart as a string.*2
Sender Name	String (40 characters)	21*1	Starting data register+43	Starting data register+63	Specify the name that is included in the sender field of the e-mails to be sent from the FC6A Series MICROSmart as a string.*2
Authentication is required to send e-mail	Decimal value	1	Starting data register+64		When authentication is required by the sending e-mail server that is used, specify 1. When authentication is not required, specify 0.
Account Name	String	21*1	Starting data register+65	Starting data register+85	Specify the account name that is used with authentication as a string.*2
Password	String	21*1	Starting data register+86	Starting data register+106	Specify the password that is used with authentication as a string.*2
Use secure connection (SSL/TLS)	Decimal value	1	Starting data register+107		When SSL or TLS communication is required with the sending e-mail server that is used, specify 1. When SSL or TLS communication is not required, specify 0.

*1 The string set to "Number of used words - 1" words is valid as the setting value. In order to represent the end of the string, the final word is handled as if 0000h was stored in it, regardless of the actual values in the data registers.

*2 If the set string is shorter than "Number of used words - 1", fill the data after the string with 00h.

Configuration Example: The e-mail settings shown below are configured by the data registers.

SMTP Server	: smtp.example.com
SMTP Server Port	: 587
Sender E-mail Address	: test@example.com
Sender Name	: Test
Authentication is required to send e-mail	: Checked
Account Name	: test_account
Password	: test_password
Use secure connection (SSL/TLS)	: Checked

When D100 is specified as the starting data register, set the values for data registers D100 to D207 as follows.

Setting Item	Start Address of Setting Item	End Address of Setting Item	Setting Value										
SMTP Server	D100	D120	**Data Register**	D100	D101	D102	D103	D104	D105	D106	D107	D108	D109 to D120
			ASCII	's' 'm'	't' 'p'	'.' 'e'	'x' 'a'	'm' 'p'	'l' 'e'	'.' 'c'	'o' 'm'	'\0' '\0'	'\0' '\0'
			Value (hexadecimal)	736Dh	7470h	2E65h	7861h	6D70h	6C65h	2E63h	6F6Dh	0000h	0000h
SMTP Server Port	D121		**Data Register**	D121									
			Value (decimal)	587									
Sender E-mail Address	D122	D142	**Data Register**	D122	D123	D124	D125	D126	D127	D128	D129	D130	D131 to D142
			ASCII	't' 'e'	's' 't'	'@' 'e'	'x' 'a'	'm' 'p'	'l' 'e'	'.' 'c'	'o' 'm'	'\0' '\0'	'\0' '\0'
			Value (hexadecimal)	7465h	7374h	4065h	7861h	6D70h	6C65h	2E63h	6F6Dh	0000h	0000h
Sender Name	D143	D163	**Data Register**	D143	D144	D145	D146 to D163						
			ASCII	'T' 'e'	's' 't'	'\0' '\0'	'\0' '\0'						
			Value (hexadecimal)	5465h	7374h	0000h	0000h						
Authentication is required to send e-mail	D164		**Data Register**	D164									
			Value (decimal)	1									
Account Name	D163	D185	**Data Register**	D165	D166	D167	D168	D169	D170	D171	D172 to D185		
			ASCII	't' 'e'	's' 't'	'_' 'a'	'c' 'c'	'o' 'u'	'n' 't'	'\0' '\0'	'\0' '\0'		
			Value (hexadecimal)	7465h	7374h	5F61h	6363h	6F75h	6E74h	0000h	0000h		
Password	D186	D206	**Data Register**	D186	D187	D188	D189	D190	D191	D192	D193 to D206		
			ASCII	't' 'e'	's' 't'	'_' 'p'	'a' 's'	's' 'w'	'o' 'r'	'd' '\0'	'\0' '\0'		
			Value (hexadecimal)	7465h	7374h	5F70h	6173h	7377h	6F72h	6400h	0000h		
Use secure connection (SSL/TLS)	D207		**Data Register**	D207									
			Value (decimal)	1									

Note: Strings by data registers are composed of 1 or more consecutive data registers. 1 word of data is handled as 2 bytes and is used in high-order byte to low-order byte order, and the terminating character is 00h.

Special Data Registers/Special Internal Relays

Initialize sending e-mail server settings

The values set in the **Function Area settings** can be reflected to the corresponding data registers as the initial values by using special internal relay M8211.

Special Internal Relays

Device Address	Description	Details
M8211	Initialize HMI module sending e-mail server settings	When this relay is turned on, the values set in the function area settings are set as the initial values in the target data registers.
M8344	Initialize Ethernet port 1 sending e-mail server settings	When this relay is turned on, the values set in the function area settings are set as the initial values in the target data registers.

Note: Strings can be stored in data registers. Strings start from the specified data register. The end of each string is specified with 00h which is stored in the high-order byte or the low order byte of the value in a data register. 1 word of data is handled as 2 bytes and is used in high-order byte to low-order byte order.

■ Text Encoding

(10) Character Set

The character set for the e-mail subject, body, and attached file can be specified.

ASCII: Specify when e-mail subject and body consist of ASCII characters only.

Japanese (ISO-2022-JP): Specify when e-mail subject and body consist of ASCII and Japanese characters. The character set for the attached file is Shift_JIS.

Chinese (GB2312): Specify when e-mail subject and body consist of Chinese characters.

Western European (ISO-8859-1): Specify when e-mail subject and body consist of western European characters.

Unicode (UTF-8): Specify when using Unicode characters.

In general, any characters can be used using Unicode. Depending on sending e-mail servers that relay the e-mails or the mailer used by the recipients, the e-mails sent out from the FC6A Series MICROSmart may not reach the recipients or may not be viewed correctly in the mailer.

When ASCII is used, e-mails can be viewed in the same manner in any mailer though only ASCII characters can be used.

Contact the administrator of the sending e-mail server to confirm the supported character sets. Select the appropriate character set for the mailer of the recipients.

(11) Encode Text Using

The encoding format for e-mail body can be specified. Depending on the sending e-mail servers that relay the e-mails, the e-mails in which 8-bit characters are used cannot be sent. In such case, Base64 encoding can be used to convert the 8-bit characters into 7-bit characters.

None: The e-mail body is not encoded.

Base 64: The e-mail body is encoded with Base64 format.

(12) Decimal Symbol

The decimal symbol for the floating-point values can be specified. When the data type of a data register embedded in the e-mail body is float, data register value is converted and shown as a floating-point value in the e-mail body.

Period (.): Period '.' (2Eh) is used.

Comma (,): Comma ',' (2Ch) is used.

(13) Separating Character

The separator for the attached CSV file is automatically determined by the selected **Decimal Symbol**.

Decimal Symbol	Separator Character
Period '.' (2Eh)	Comma ',' (2Ch)
Comma ',' (2Ch)	Semicolon ';' (3Eh)

Note: The text encoding settings of HMI-Ethernet port and Ethernet port 1 are the identical. For example, the configured text encoding settings in Email Settings of the HMI Module Configuration dialog box are automatically reflected to the text encoding settings for the Ethernet port 1.

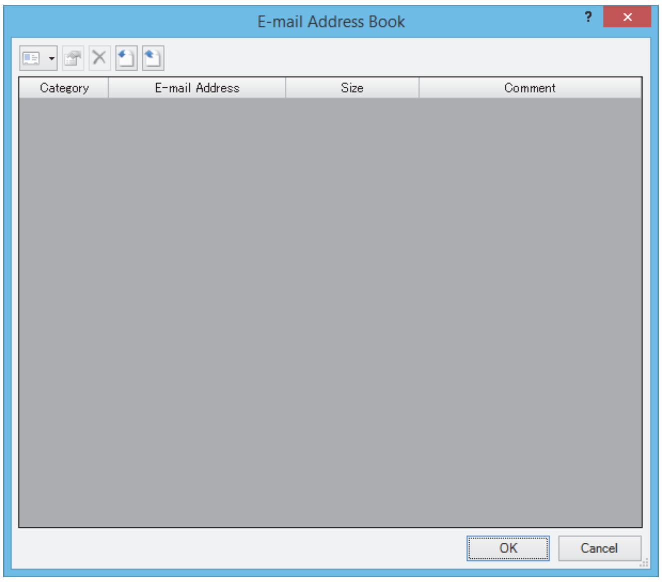
E-mail Address Book

The e-mail addresses and e-mail address groups can be configured in E-mail Address Book dialog box. E-mail addresses can be grouped into an e-mail address group. The e-mail address group can be used to specify a group of e-mail addresses in each e-mail.

A maximum of 255 combined e-mail addresses and e-mail address groups can be registered.

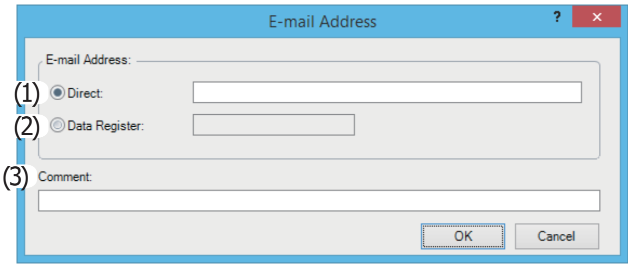
Programming WindLDR

1. Double-click on the **E-mail Address Book** in the **Project Window**.
The **E-mail Address Book** dialog box appears.



2. Click the ▼ next to the **New E-mail Address**, and then click **New E-mail Address**. Alternatively, select an existing E-mail address, and then click the **Edit** button.
The **E-mail Address** dialog box appears.

3. Configure the settings.



(1) Direct
Specify the e-mail address to register in the E-mail Address Book dialog box. A maximum of 40 ASCII characters can be entered.

(2) Data Register
Specify the data register to store the e-mail address. When sending an e-mail, the string stored in a maximum of 30 words of data registers starting from the specified data register is used as e-mail address for the E-mail recipients (To and CC). When specifying an e-mail address with a string using data registers, the length of the e-mail address is considered as 60 bytes fixed. When specifying all e-mail addresses with data registers, a maximum of eight e-mail addresses can be set for the E-mail recipients (To and CC).

(3) Comment
The comment for the e-mail address can be assigned. The contents or the length of the comment has no effect on the CPU module operation.

Configuration Example: To send an e-mail to test@example.com by specifying the data register D100, store the values in the data registers as follows.

Data Register	D100	D101	D102	D103	D104	D105	D106	D107	D108
ASCII	't' 'e'	's' 't'	'@' 'e'	'x' 'a'	'm' 'p'	'l' 'e'	'.' 'c'	'o' 'm'	'\0' '\0'
Value (hexadecimal)	7465h	7374h	4065h	7861h	6D70h	6C65h	2E63h	6F6Dh	0000h

To send an e-mail to you@example.com by specifying the data register D123, store the values in the data registers as follows.

Data Register	D123	D124	D125	D126	D127	D128	D129	D130
ASCII	y' 'o'	'u' '@'	'e' 'x'	'a' 'm'	'p' 'l' '	'e' '.'	'c' 'o'	'm' '\0'
Value (hexadecimal)	796Fh	7540h	6578h	616Dh	706Ch	652Eh	636Fh	6D00h

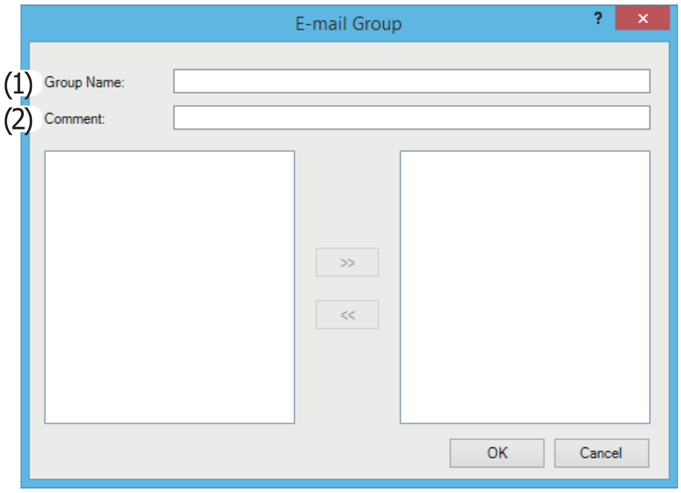
Notes:

- Strings by data registers are composed of 1 or more consecutive data registers. 1 word of data is handled as 2 bytes and is used in high-order byte to low-order byte order, and the terminating character is 00h.
- If 00h is not included in the 30 words (60 bytes) from the data register specified as the e-mail address, all 60 bytes are used as the e-mail address, and the FC6A Series MICROSmart operates as if the 61st byte is 00h.

4. Click **OK**.

5. Click the ▼ next to the **New E-mail Address**, and then click **New E-mail Group**. Alternatively, select an existing E-mail address, and then click the **Edit** button.
The **E-mail Group** dialog box appears.

6. Configure the settings.



(1) Group Name

Specify the name of e-mail address group to register in the E-mail Address Book dialog box.

(2) Comment

The comment for the e-mail address group can be assigned. The contents or the length of the comment has no effect on the CPU module operation.

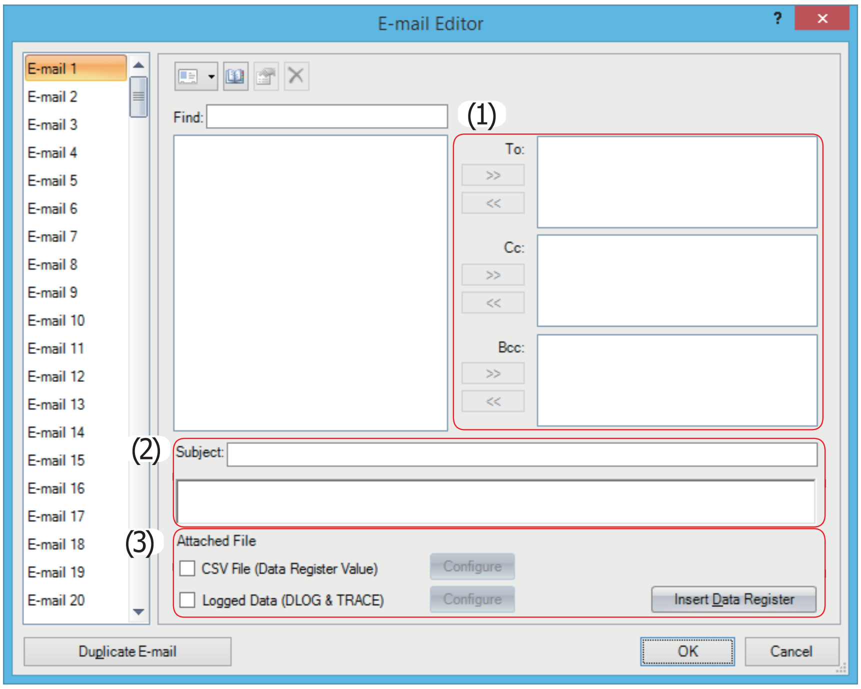
The e-mail addresses that have not been added to the e-mail address group are shown in the list box on the left. The e-mails addresses are listed in the order that they were registered in the E-mail Address Book dialog box. In order to add an e-mail address to the e-mail address group, select the e-mail address to add and click on the **>>** button. The selected e-mail address is moved to the list box on the right and listed to the bottom of the e-mails.
The e-mail addresses for the e-mail address group are shown on the list box on the right. The e-mail addresses are listed in the order that they were added to the e-mail address group. In order to remove an e-mail address from the e-mail address group, select the e-mail address to remove and click on the **<<** button. The selected e-mail address is moved to the list box on the left and listed to the bottom of the e-mails.
In order to delete unused e-mail addresses or e-mail address groups from the E-mail address book, select the e-mail address or e-mail address group to delete and click Delete button in the E-mail Address Book dialog box. When the deleted e-mail address is used in the e-mail address groups, the deleted e-mail address is removed from all e-mail address groups.

7. Click **OK**.

This concludes configuring the settings.

E-mail Editor

The e-mails can be configured in E-mail Editor dialog box. The following five parameters should be configured for each e-mail.



Settings in E-mail Editor

(1) E-mail recipients (To, Cc, and Bcc)

E-mail address or e-mail group can be specified as the recipients. The maximum size of texts for **To**, **Cc**, or **Bcc** is 512 bytes. Comma (,) is inserted as a separating character between e-mail addresses. For example, the total size of an e-mail group "ccc" containing two e-mail addresses "aa@example.com" (15 bytes including a comma) and "bbb@example.com" (16 bytes including a comma) is 31 bytes. If the size of e-mail addresses is 30 bytes, 16 e-mail addresses can be specified in **To**, **Cc**, and **Bcc** respectively. An e-mail can be sent to 32 e-mail addresses simultaneously.

(2) E-mail subject and body

The maximum size of texts is 256 bytes for **Subject** and 4,096 bytes for **e-mail body**. The e-mail subject and body are composed from the following elements.

- Strings that use **character set** under **E-mail Settings** in the **Function Area settings**
- Numeric value strings of the values of data registers that have been embedded in the body by inserting data registers
- Spaces and newlines

The size of the **e-mail subject** and **body** is the text that is composed of the combination of the above elements.

The result of encoding the composed text according to **Text encoding** under **E-mail Settings** in the **Function Area settings** is sent as the e-mail.

When only single-byte ASCII characters are used, approximately 200 characters can be entered for e-mail subject and 3,500 characters for e-mail body. When multi-byte characters are used, approximately 100 characters can be entered for e-mail subject and 2,000 characters for e-mail body

(3) Attached File

One CSV file can be attached to an e-mail. The attached file can include text and data register values. The file name of the CSV file is fixed as "data.csv". The maximum size of the attached file is 4,096 bytes.
The attached file composed from the following elements.

- Strings that use **character set** under **E-mail settings** in the **Function Area settings**
- Numeric value strings of the values of data registers that have been embedded in the body by inserting data registers
- Spaces, separator characters, and newlines

The size of the **attached file** is the text that is composed of the combination of the above elements.

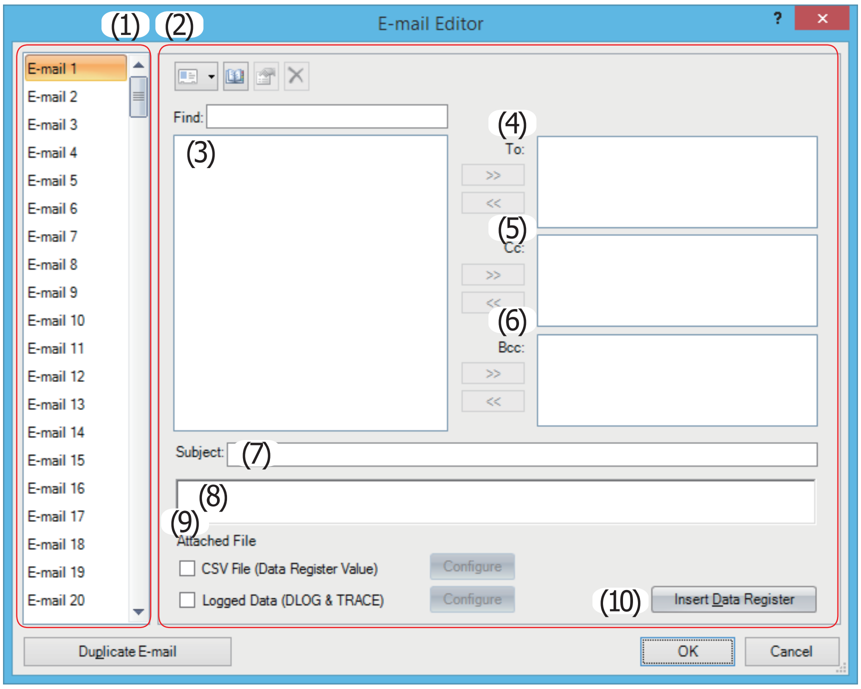
The result of encoding the composed text according to **Text encoding** under **E-mail Settings** in the **Function Area settings** is attached to the e-mail.

When sending e-mails from Ethernet port 1 of the Plus CPU module, log files created with the DLOG and TRACE instructions can be attached to e-mails.

Programming WindLDR

1. Double-click on the **E-mail Editor** in the **Project Window**.
   The **E-mail Editor** dialog box appears.

2. Create an e-mail.



(1) E-mail
Displays the registered e-mails.

(2) E-mail content
Displays the content of the selected e-mail.
This area is composed of the following six elements: **To**, **Cc**, **Bcc**, **Subject**, **Body**, **Attached File**.

(3) E-mail addresses
Displays the e-mail addresses registered in the E-mail Address Book.
Select the e-mail address to add to **To**, **Cc**, or **Bcc** and click **>>** on the appropriate side to add that e-mail address to **To**, **Cc**, or **Bcc**.
Of the e-mail addresses in the **To**, **Cc**, or **Bcc** boxes, select the e-mail address to delete and click **<<** on the appropriate side to delete that e-mail address from **To**, **Cc**, or **Bcc**.

(4) To/(5) Cc/(6) Bcc
Specifies the addresses of the e-mail to send.

(7) Subject
Specifies the subject of the e-mail to send.

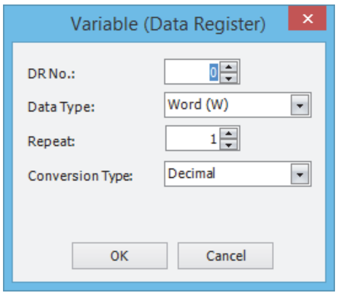
(8) Body
Specifies the e-mail body.
The e-mail body can be written in multiple lines.

(9) Attached File
Select this check box to attach a file to the e-mail.

CSV File (Data Register Value)	:	The attached file (CSV file) that includes text and data register values is configured with the Attached File Editor. Click **Edit** to display the Attached File Editor. The layout and items can be freely configured in a range that fits in the file size limit. For details, see "Attached File Editor" on page 12-15.
Logged Data (DLOG & TRACE)	:	For the Plus CPU module, log files created with the DLOG and TRACE instructions can be attached to e-mails. Configure these settings in **Attach Log Data** dialog box. Click **Configure** to display the **Attach Log Data** dialog box. For details, see "Attach Log Data" on page 12-18.

(10) Insert Data Register

The **Variable (Data Register)** dialog box appears.



The data register values can be embedded in the e-mail body when the FC6A Series MICROSmart sends e-mails.

Item		Description
DR Number		Specify the data register number.
Data Type	Word (W)	The 1-word value stored in the specified data register is converted to an unsigned 16 bits value.
	Integer (I)	The 1-word value stored in the specified data register is converted to a signed 15 bits value.
	Double (D)	The 2-word value stored in the two consecutive data registers starting with the specified data register is converted to an unsigned 32 bits value.
	Long (L)	The 2-word value stored in the two consecutive data registers starting with the specified data register is converted to a signed 31 bits value.
	Float (F)	The 2-word value stored in the two consecutive data registers starting with the specified data register is converted to a floating-point value according to IEEE754 format. Digits can be specified between 1 and 7.
	String (S)	Values in the configured data registers are encoded to a character string with the character set for the text encoding. The length of character string Characters.

When word (W) or double (D) is selected, the conversion type can be specified. For example, when a data register value is 4660 (1234h), the data register value is converted and embedded in the e-mail body as follows:

Decimal: 4660

Hexadecimal: 1234

When the repeat is configured, the values in the consecutive data registers can be embedded in the e-mail body. A space is inserted between the data register values. For example, when the data register values are (D100) = 1234h, (D101) = 5678h, and (D102) = ABCDh, and the DR number is 100, the data type is Word, the repeat is 3, and the conversion type is hexadecimal, the following text will be embedded in the e-mail body:

1234 5678 ABCD

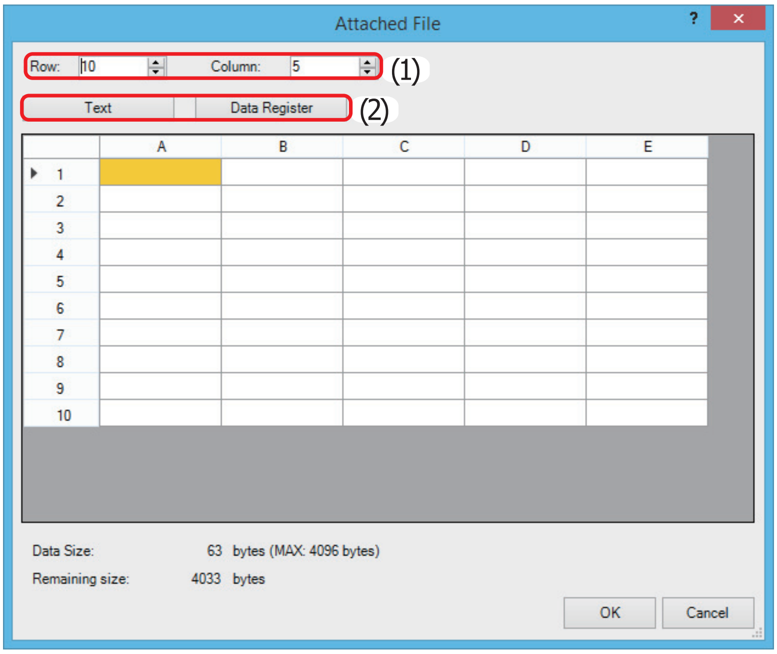
The text created in the E-mail Editor is encoded and sent according to the **Encoding method** settings specified on **E-mail Settings** in **Function Area Settings**.

3. Click **OK**.

Attached File Editor

Edit the content of the CSV file attached to the e-mail.

The desired text and data register values can be included in the CSV file. The file name of the CSV file is fixed as "data.csv".



Settings in Attached File Editor

(1) Row, Column

The number of rows and columns in the CSV file can be changed. The number of rows and columns can be specified between 1 and 64. When the editing range is shrunk, the setting values outside the editing range are cleared.

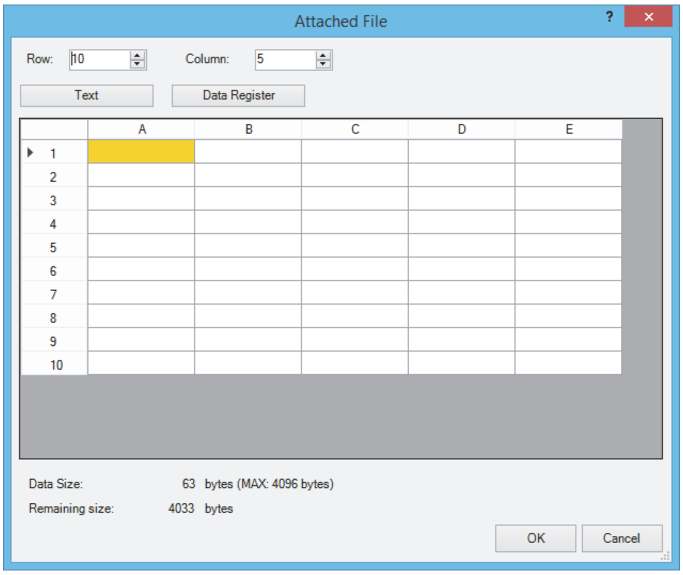
(2) Text, Data Register

Set the text and data registers displayed in the table to determine the content of the attached file. The file size of the CSV file is a maximum of 4,096 bytes.

Note: The character set, separator, and decimal symbol configured in the Function Area Settings are applied to all attached files. For those settings, see "Programming WindLDR" on page 12-4 in this chapter.

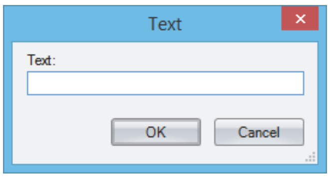
Programming WindLDR

1. Select the **Attached File** check box in the **E-mail Editor** and click **Edit**.
   The **Attached File** dialog box appears.
2. Specify **Row** and **Column**.
   Set the number of rows and columns in the CSV file to determine the editing range.



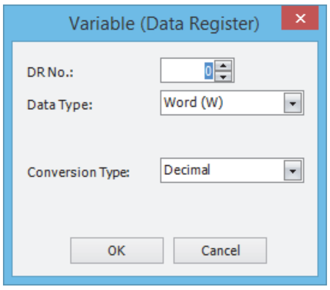
The content of the CSV file can be modified at the upper part of the dialog. The current file size and the remaining size that can be used for editing the content are displayed at the bottom of the dialog. You can expand the grid area by changing the size of the dialog. The file size includes separators and line breaks to be contained in the CSV file.

3. Select any cell and click **Text**.
   The **Text** dialog box appears.
4. Enter the desired text in the selected cell.
   The maximum length of the text that can be specified is 63 bytes.



5. Click **OK**.
   You are returned to the **Attached File** dialog box.
6. Select any cell and click **Data Register**.
   The **Variable (Data Register)** dialog box appears.

7. Configure the settings.



The value of the data register when the e-mail is sent can be included in the selected cell.

Item		Description
DR Number		Specify the data register number.
Data Type	Word (W)	The 1-word value stored in the specified data register is converted to an unsigned 16 bits value.
	Integer (I)	The 1-word value stored in the specified data register is converted to an unsigned 16 bits value.
	Double (D)	The 2-word value stored in the two consecutive data registers starting with the specified data register is converted to an unsigned 32 bits value.
	Long (L)	The 2-word value stored in the two consecutive data registers starting with the specified data register is converted to a signed 31 bits value.
	Float (F)	The 2-word value stored in the two consecutive data registers starting with the specified data register is converted to a floating-point value according to IEEE754 format. Digits can be specified between 1 and 7.

When word (W) or double (D) is selected, the conversion type can be specified. For example, when a data register value is 4660 (1234h), the data register value is converted and embedded in the e-mail body as follows:

Decimal: 4660

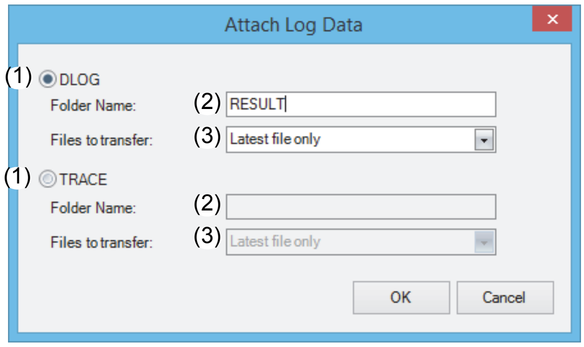
Hexadecimal: 1234

8. Click **OK**.
You are returned to the **Attached File** dialog box.

9. Click **OK**.

Attach Log Data

Set the log files that will be attached to the e-mail.



Settings

(1) Log File Type

Select the log file type to attach.

DLOG and **TRACE**

(2) Folder Name

Specify the folder name of the target folder to which the log files are saved by the DLOG or TRACE instructions. Up to 8 characters can be entered.

(3) Files to transfer

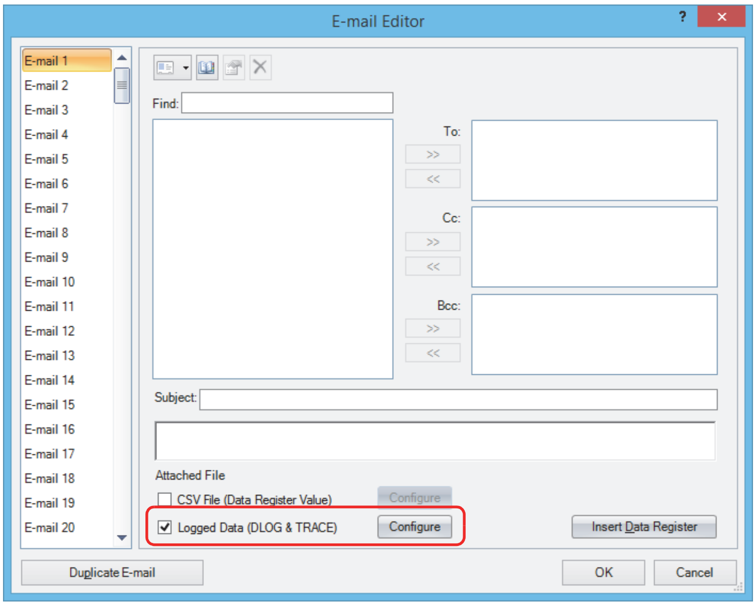
Select the specification method for the log files to attach from the following.

- Files updated within 24 hours
- Files updated within 7 days
- Latest file only

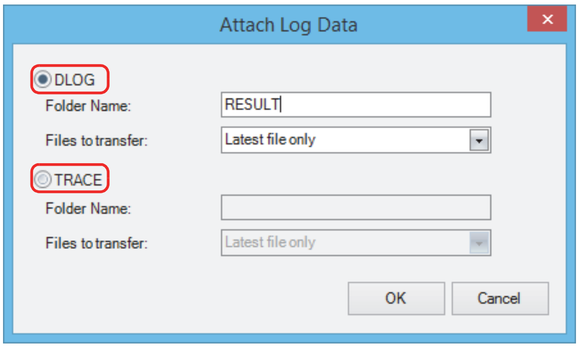
Note: The maximum number of log files that can be attached is seven. If the number of the log files exceeds seven, eighth or later log files are not attached and the error code 11 is stored in D2 of the EMAIL instruction.

Programming WindLDR

1. In the **E-mail Editor**, select the **Log Data (DLOG & TRACE)** check box and click **Configure**.
The **Attach Log Data** dialog box is displayed.



2. Select the type of the log file to attach to the e-mail from **DLOG** or **TRACE**.



3. With **Folder Name**, specify the folder name of the target folder to which the log files are saved. Up to 8 characters can be entered.

4. With **Files to transfer**, select the specification method for the log files from the following.

- Files updated within 24 hours
- Files updated within 7 days
- Latest file only

5. Click **OK**.

This concludes configuring the settings.

13: WEB SERVER

Introduction

This chapter describes the Web server functions in the FC6A Series MICROSmart.

Overview

Ethernet port 1 of the Plus CPU module and the HMI-Ethernet port of the HMI module support the Web server functions. The Web server enables you to access the FC6A Series MICROSmart using web browser on your PC in order to monitor the status or change the device data. Users can also download web page data that was freely created and build easy-to-use web sites for each project.

When using the Plus CPU module, the internal memory, SD memory card, or internal memory of the HMI module can be selected for storing web pages.

On the system web page, the PLC status can be monitored and data register values can be changed. The FC6A Series MICROSmart provides CGI so that you can access data register values of the FC6A Series MICROSmart from a web browser using JavaScript. The system library is also provided so that you can show data register values on your web page using system library parts, such as numerical input, bar graph, or trend graph, without special knowledge of JavaScript.

The Web server is protected with basic authentication. Access from people who do not have user name and password can be prevented.

The tree structure of web page

The tree of structure of the web page of web server in the FC6A Series MICROSmart is shown below:

/index.html:	System top page. Automatically generated by WindLDR.
/system/:	The root folder of the system web page. Automatically generated by WindLDR.
/system/index.html:	The top page of the system web page.
/user/:	User web page folder.

The imported files or folders in the Project Window of WindLDR are added under the root folder of the web page. If the IP address of the FC6A Series MICROSmart is 192.168.1.5, enter the following URL on the web browser to access the top page of the system web page.

http://192.168.1.5/system/index.html

When a folder is specified as URL, index.html in the folder is displayed. For example, the following URL opens the same page with the above URL.

http://192.168.1.5/system/

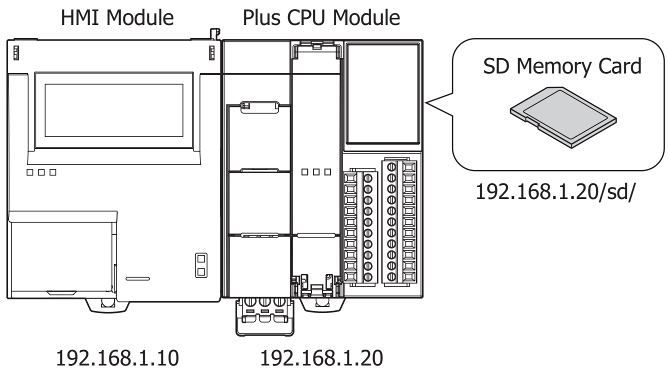
Save Locations of Web Pages

With the Plus CPU module, you can select the save location when downloading web pages.

- Internal memory of the Plus CPU module (5 MB maximum).
- SD memory card inserted into the Plus CPU module (32 GB maximum). However, the maximum size of one file is 4 GB.
- Internal memory of the HMI module (5 MB maximum).

The URL when accessing downloaded web pages will depend on the save location.

For example, the URLs are set as follows when the IP address of the Plus CPU module is 192.168.1.20 and the IP address of the HMI module connected to the Plus CPU module is 192.168.1.10.

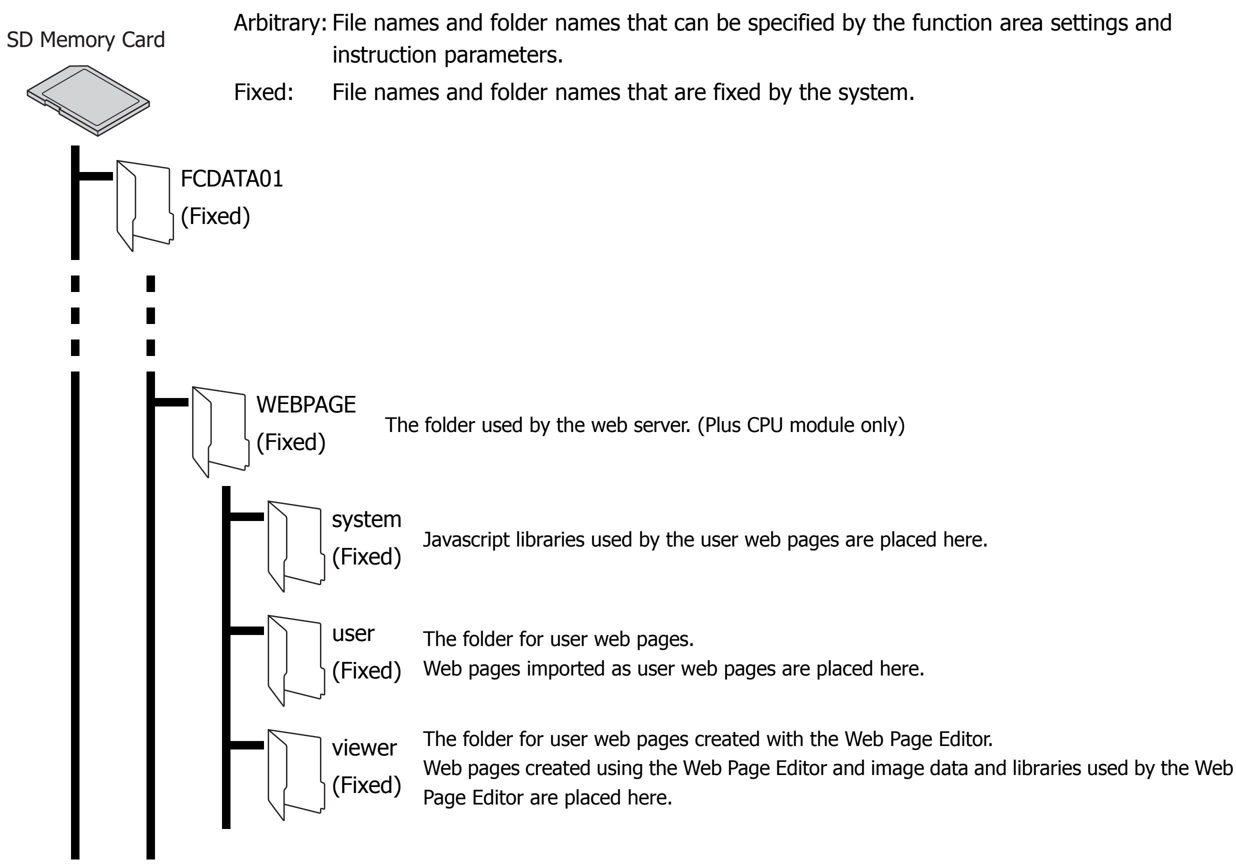


Save Location	URL When Accessing Web Pages
Internal memory of the Plus CPU module	http://192.168.1.20/
SD memory card inserted into the Plus CPU module	http://192.168.1.20/sd/
Internal memory of the HMI module connected to the Plus CPU module	http://192.168.1.10/

Locations of Web Pages Saved to SD Memory Card

When the save location is set to the SD memory card, downloaded web pages are saved to specific folders on the SD memory card.

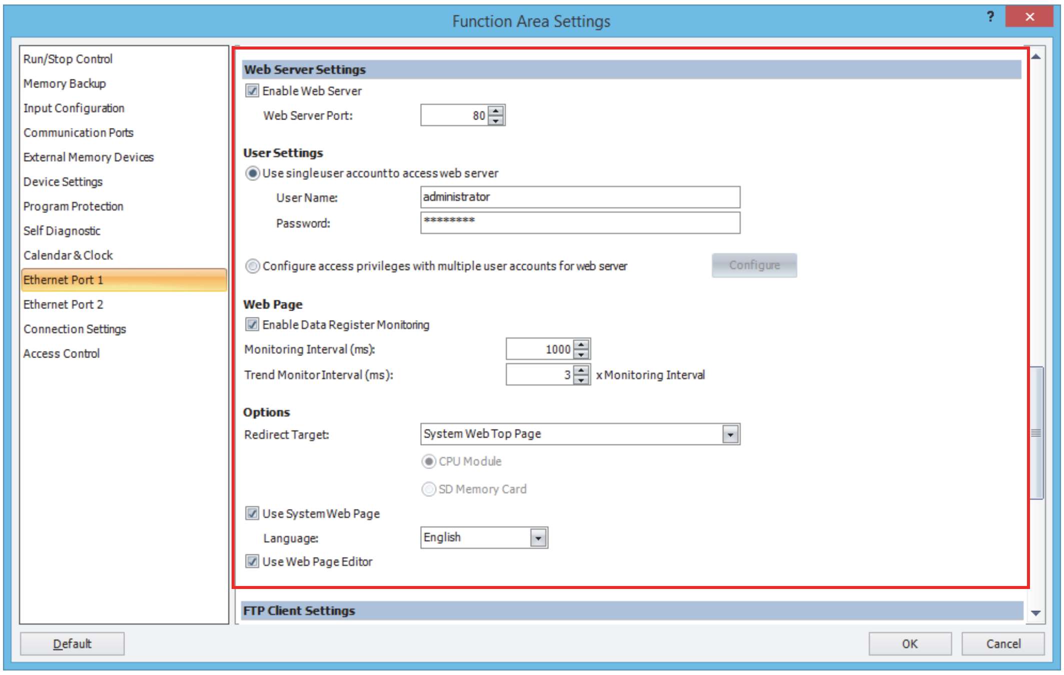
Only user web pages (including web pages created with the Web Page Editor) can be downloaded. The save locations on the SD memory card are as follows.



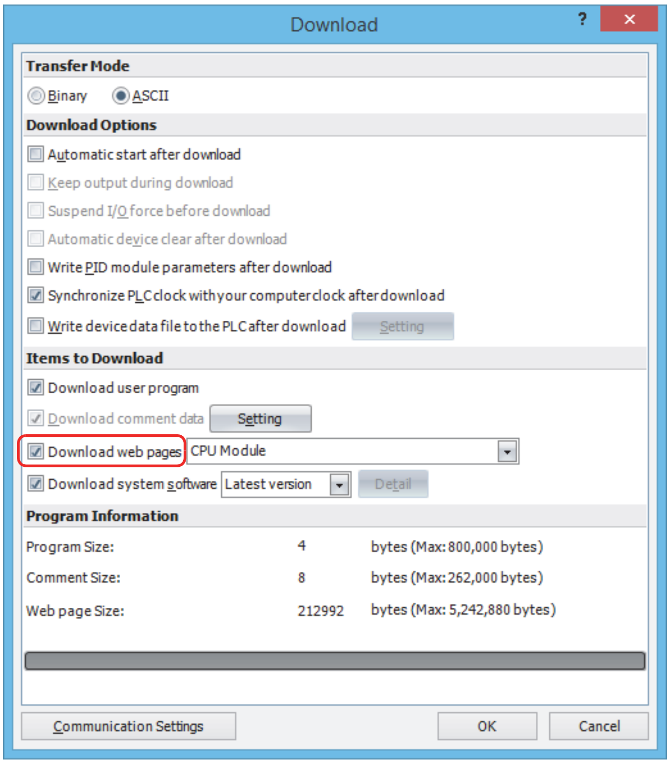
Programming WindLDR

Plus CPU Module Web Server Settings

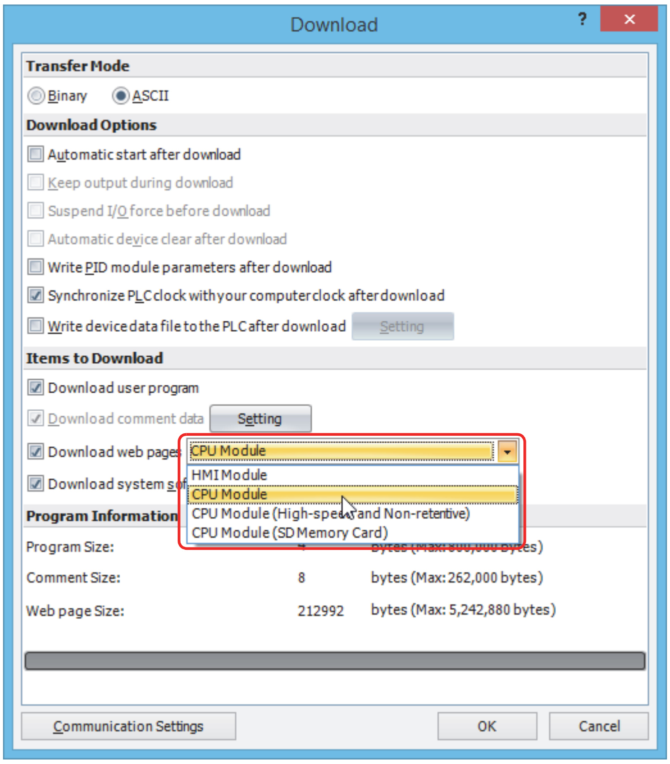
1. On the **Configuration** tab, in the **Function Area Settings** group, click **Ethernet Port 1**.
2. Under **Web Server Settings**, configure the web server settings and the settings for the web page and options.



3. Download the user program to the Plus CPU module.
   From the WindLDR menu bar, select **Online** > **Transfer** > **Download**.
   The **Download** dialog box appears.
4. Select the **Download web pages** check box.



5. Select the save location of web pages from the following.



No Download	: Web pages will not be downloaded.
CPU Module	: Downloaded web pages will be saved to the internal memory of the Plus CPU module.
CPU Module (High-speed and Non-retentive)	: Downloaded web pages will be saved to the internal memory of the Plus CPU module. This option allows web pages to be downloaded faster than when **CPU Module** is selected, but the downloaded web pages will be deleted when the power to the Plus CPU module is turned off. Select this option when you will repeatedly edit and download web pages, such as to check the operation of created web pages.
CPU Module (SD Memory Card)	: Downloaded web pages will be saved to the SD memory card inserted into the Plus CPU module.
HMI Module	: Downloaded web pages will be saved to the internal memory of the HMI module connected to the Plus CPU module.

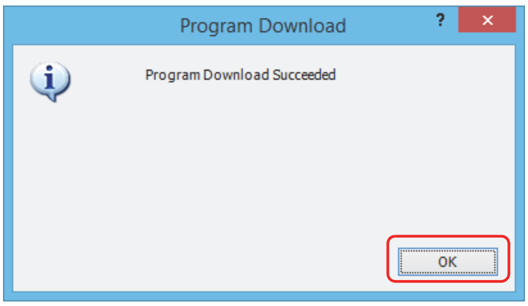
Notes:

- Web pages will not be downloaded when no SD memory card is inserted into the Plus CPU module but **CPU Module (SD Memory Card)** is selected.
- Web pages will not be downloaded when HMI module is not connected to the Plus CPU module but **HMI Module** is selected.

6. Click **OK**.

The web pages will be downloaded to the selected save location.

When the following message appears, the download has completed successfully. Click **OK**.



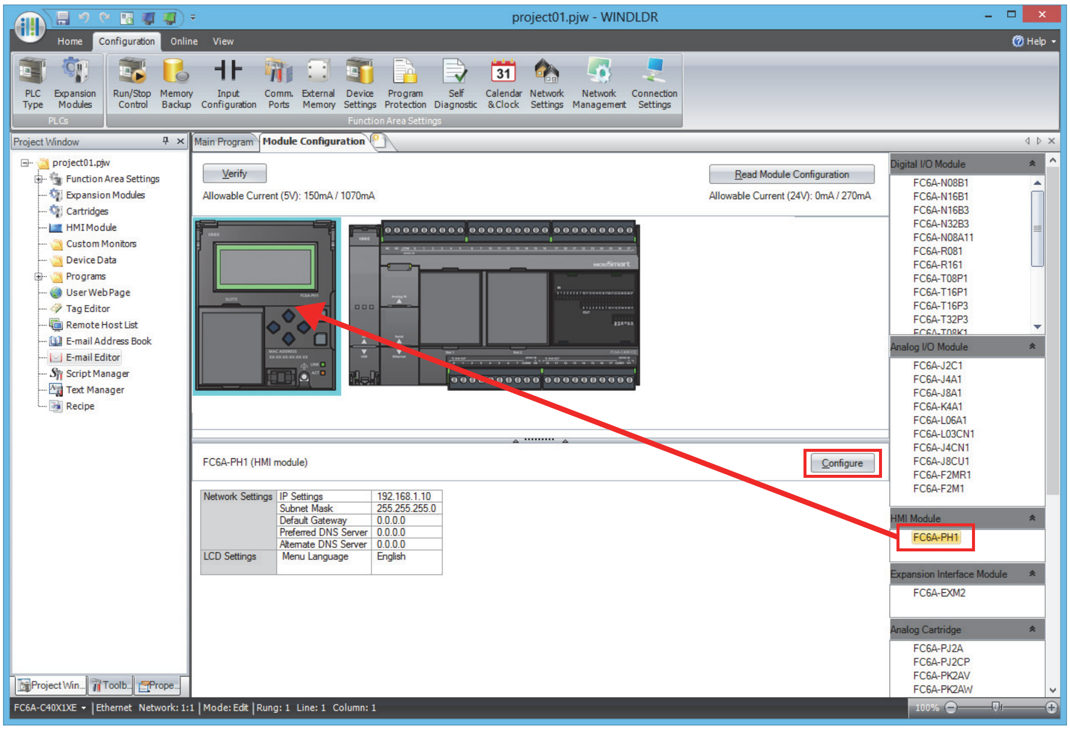
This concludes configuring the web server settings.

HMI Module Web Server Settings

1. Use the Module Configuration Editor to configure the HMI module Web server settings.
On the **Configuration** tab, in the **PLCs** group, select **Expansion Modules**.

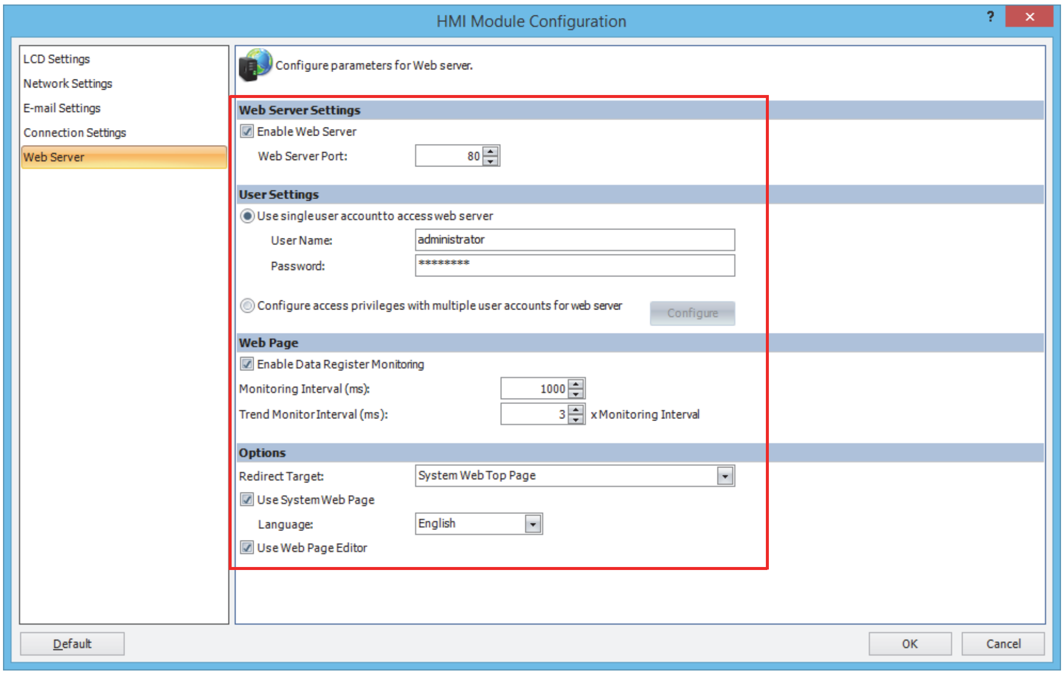
2. Click the inserted HMI module in the module configuration area and click **Configure**.
The **HMI Module Configuration** dialog box is displayed.

Note: You can also display the **HMI Module Configuration** dialog box by double-clicking **HMI Module** in the Project Window.



3. Click **Web Server**.

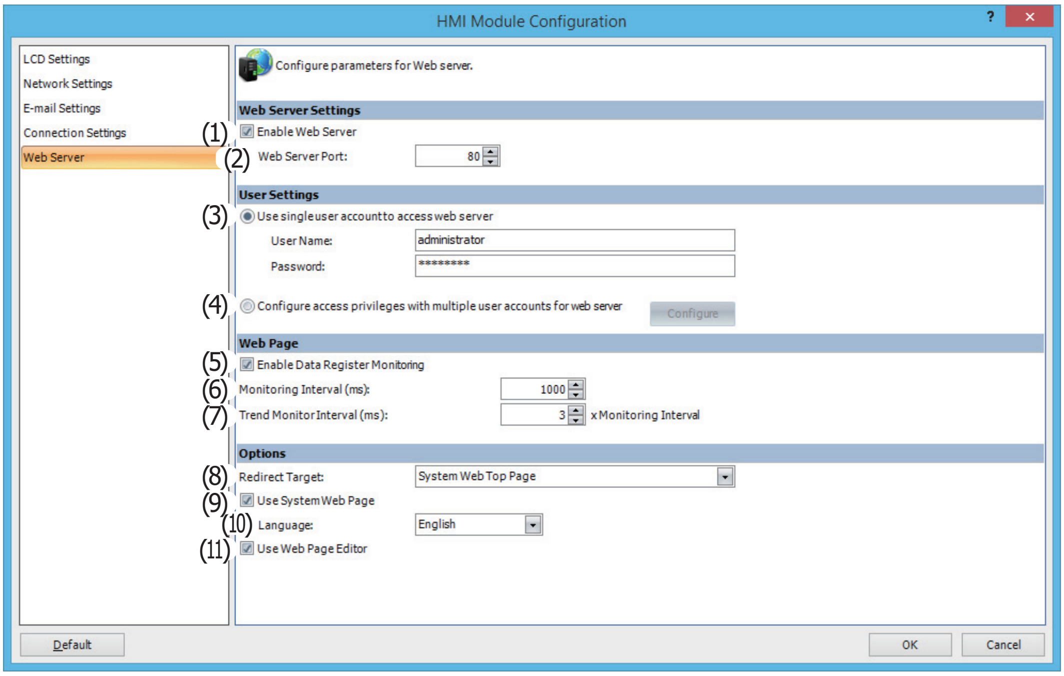
4. Configure the settings in Web Server Settings, Web Page, and Options.



5. Download the user program to the FC6A Series MICROSmart.

This concludes configuring the web server settings.

■ **Web Server Settings**



(1) Enable Web Server

Uncheck to disable Web server functions of the FC6A Series MICROSmart. Password authentication will be required when accessing the web page.

When the Web server is enabled, you will be asked to enter a user name and password. You can access the Web server once you enter the correct user name and the password on the dialog box shown on the browser.

When this is unchecked, the Web server port remains closed, and connection to the Web server using the Web browser is prohibited.

(2) Web Server Port

Specify the port number for the Web server of the FC6A Series MICROSmart. The default port number is 80.

■ **User Settings**

(3) Use single user account to access web server

User Name : Specify the user name used for authentication. The default user name is "administrator."

Password : Specify the password used for authentication. The default password is "password." The password is shown as * in the dialog box.

(4) Configure access privileges with multiple user accounts for web server

Select to use multiple accounts and limit access to the web server. Click **Configure** and create user accounts in the displayed **User Account Configuration** dialog box. For details, see "User Account Settings" on page 13-8.

■ **Web Page**

(5) Enable Data Register Monitoring

Select this check box when using data register monitoring by embedding metacharacters in the html page and when using the graph library. Valid for only files with the extension htm and html.

(6) Monitoring Interval (ms)

When monitoring, set the interval in ms to send requests from the web browser to the FC6A Series MICROSmart.

(7) Trend Monitor Interval (ms)

When performing monitoring that uses the trend graph, set the interval in milliseconds to send requests from the web browser to the FC6A Series MICROSmart as a whole number multiple of the monitoring interval.

■ **Options**

(8) Redirect Target

To redirect the web page to the system web top page after displaying the system top page, select **System Web Top Page**. Select **Disable Redirect** to disable redirect. Redirect may not work depending on each Web browser.

For the Plus CPU module, you can select the save location of the web page specified as the redirect target from **CPU Module** or **SD Memory Card**. This option can be set only when an item other than **System Web Top Page** or **Disable Redirect** is selected.

(9) Use System Web Page
Select this check box to use the system web pages.

(10) Language
Select language for the system web pages.

(11) Use Web Page Editor
Select this check box to use the Web Page Editor. For details, see "Web Page Editor" on page 13-30.

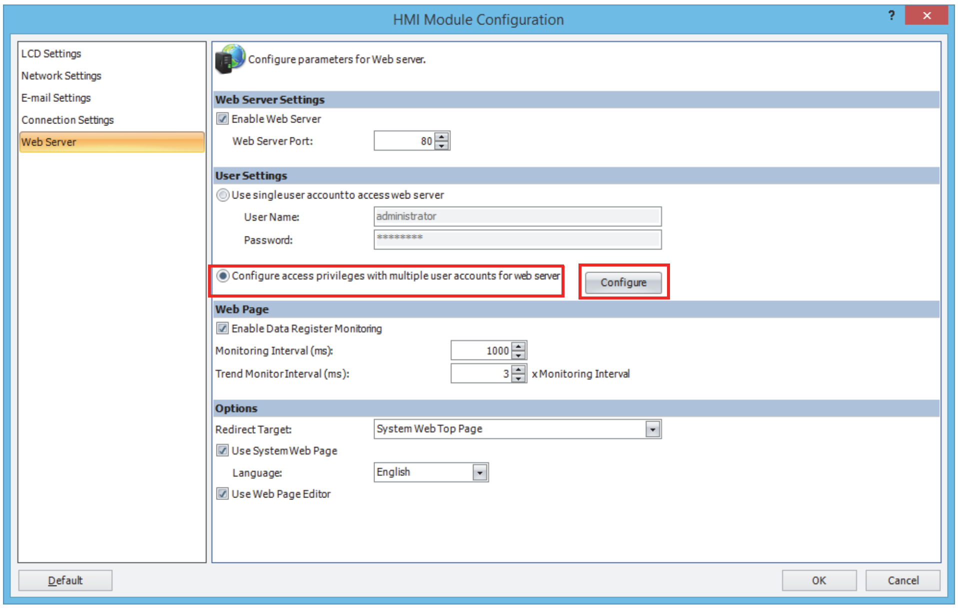
User Account Settings

■ Creating Multiple User Accounts

- **Operation procedure**

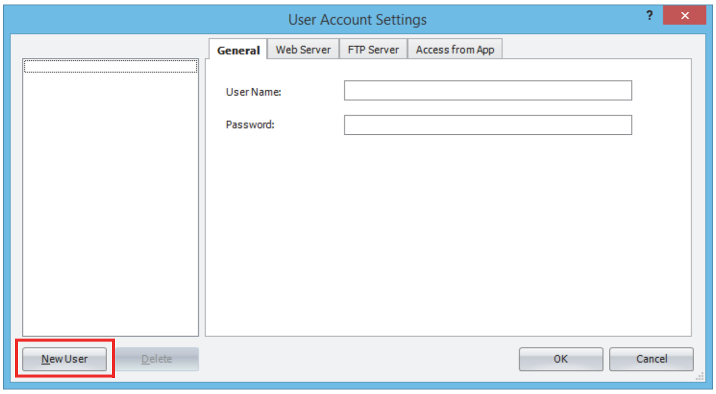
1. In the **Function Area Settings** dialog box, under **Ethernet Port 1** and **Web Server Settings** or in the **HMI Module Configuration** dialog box, on the **Web Server** tab, select **Configure access privileges with multiple user accounts for web server** and click **Configure**.

The **User Account Settings** dialog box is displayed.

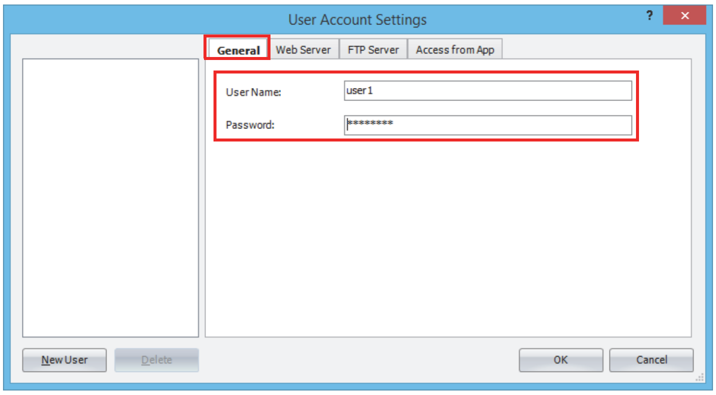


2. Click **New User**.

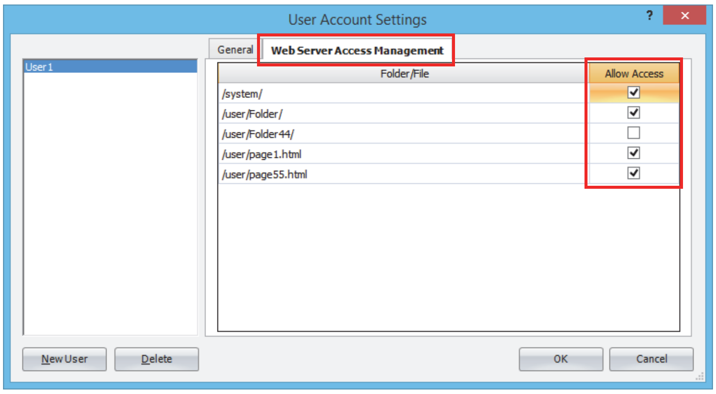
A user account is created in the list.



3. On the **General** tab, set **User Name** and **Password**.



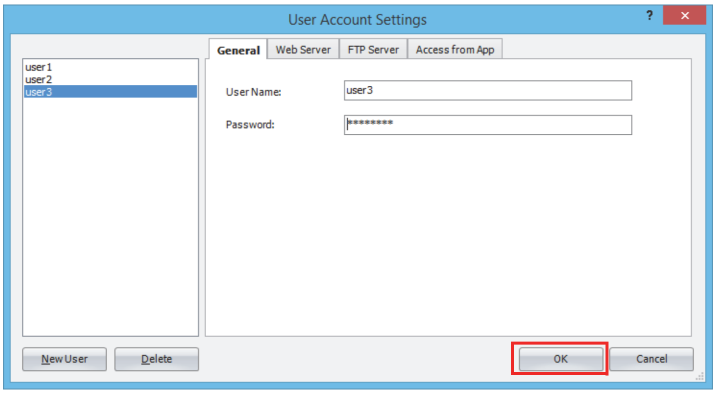
4. Click the **Web Server Access Management** tab and select the check boxes of folders and servers where access will be permitted.



5. Repeat steps 2 to 4 and create the necessary number of user accounts.

Note: If you select a user account on the list, you can edit the settings of the selected account on the **General** and **Web Server Access Management** tabs.

6. Click **OK**.



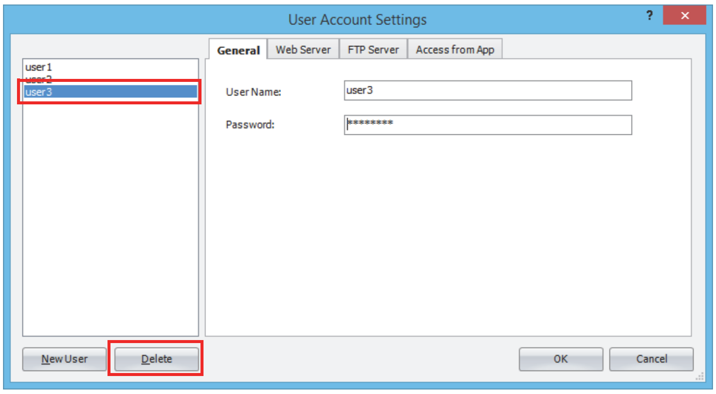
This concludes creating multiple user accounts.

■ Deleting a Created User Account

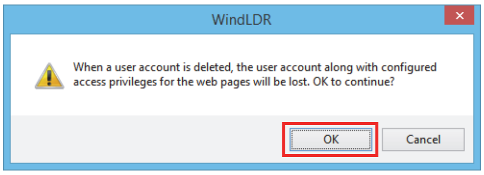
- **Operation procedure**

1. Select the user account on the list and click **Delete**.

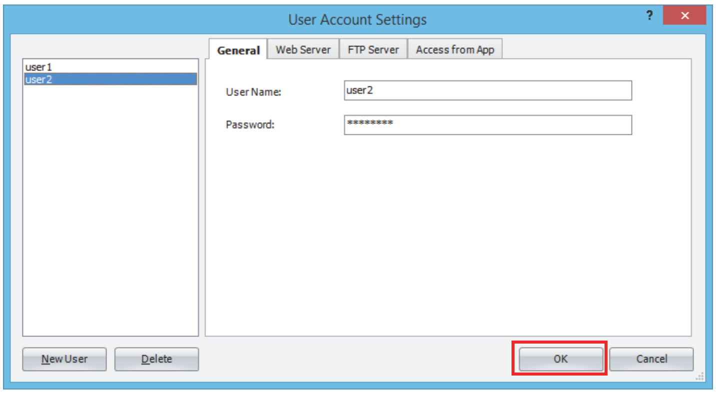
A confirmation message is displayed.



2. Click **OK**.



3. Click **OK**.



This concludes deleting a user account.

System Web Page Overview

The system web page of the FC6A Series MICROSmart consists of the following files.

/system/index.html:	The system web top page that displays the FC6A Series MICROSmart status.
/system/batch_monitor.html:	Batch monitor to monitor a group of data registers.
/system/custom_monitor.html:	Custom monitor to register and monitor a maximum of 30 data registers or internal relays.
/system/lcd_monitor.html:	The system web page with the HMI module LCD monitor function. (HMI-Ethernet port of the HMI module only)
/system/device_read.cgi:	CGI for reading device data used by JavaScript.
/system/device_write.cgi:	CGI for writing device data used by JavaScript.

Separately from these system web pages, the FC6A Series MICROSmart project can contain user web pages with user-defined page data.

The placement of files that are set on the WindLDR Project window is reflected under /user/.

Web Data Type

The following web data types can be specified in batch monitor, custom monitor.

Data Type		Data Size	Valid Range
HEX-W	Hexadecimal	1 word	0000 to FFFF
HEX-D	Hexadecimal	2 words	00000000 to FFFFFFFF
DEC-W	Decimal	1 word	0 to 65535
DEC-I	Decimal	1 word	-32768 to 32767
DEC-D	Decimal	2 words	0 to 4294967296
DEC-L	Decimal	2 words	-2147483648 to 2147483647
DEC-F	Decimal	2 words	IEEE754 format. Digits is 7.
BIN-B	Binary	1 bit	0000 or 0001 (Note)

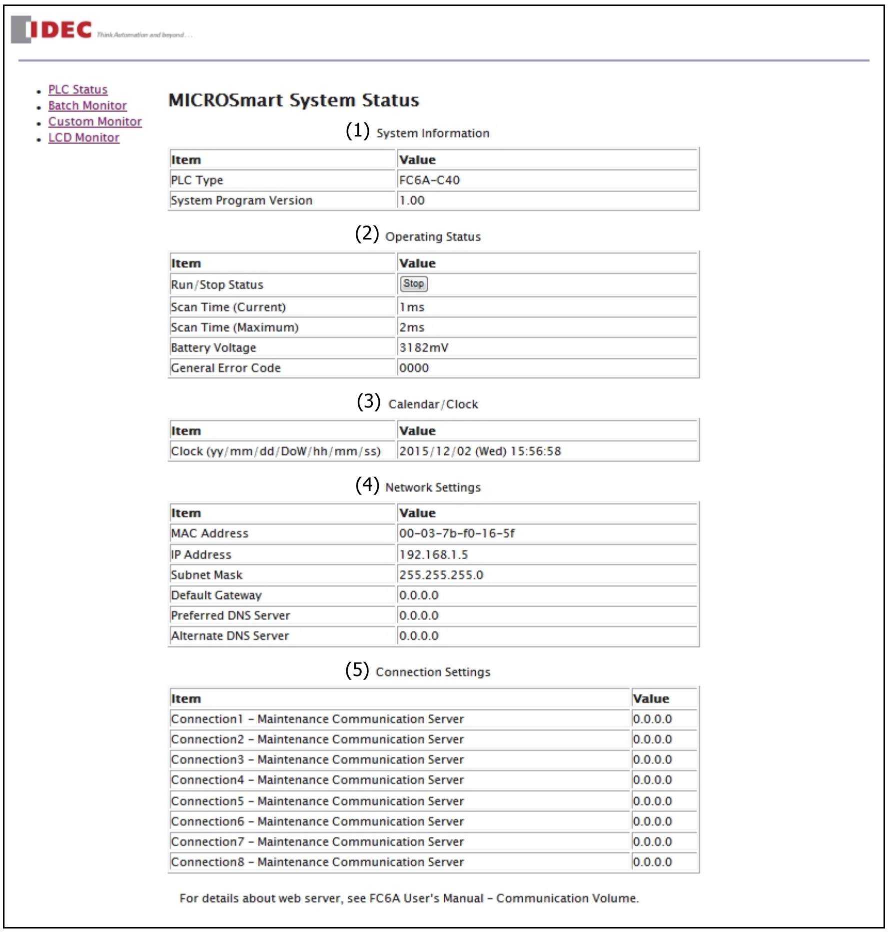
Note: BIN-B is used in the custom monitor when internal relay is monitored. 0000 is OFF and 0001 is ON.

System Web Page

The FC6A Series MICROSmart features the built-in system web page for easily browsing the PLC status and data register values. Since the system web page is embedded in the FC6A Series MICROSmart, you can easily access and utilize the system web page.

■ PLC Status

In the PLC status page, the FC6A Series MICROSmart status, such as the system program version or scan time, can be confirmed. You can also run or stop the FC6A Series MICROSmart using the Run/Stop button.



IDEC

- PLC Status
- Batch Monitor
- Custom Monitor
- LCD Monitor

MICROSmart System Status

(1) System Information

Item	Value
PLC Type	FC6A-C40
System Program Version	1.00

(2) Operating Status

Item	Value
Run/Stop Status	Stop
Scan Time (Current)	1ms
Scan Time (Maximum)	2ms
Battery Voltage	3182mV
General Error Code	0000

(3) Calendar/Clock

Item	Value
Clock (yy/mm/dd/DoW/hh/mm/ss)	2015/12/02 (Wed) 15:56:58

(4) Network Settings

Item	Value
MAC Address	00-03-7b-f0-16-5f
IP Address	192.168.1.5
Subnet Mask	255.255.255.0
Default Gateway	0.0.0.0
Preferred DNS Server	0.0.0.0
Alternate DNS Server	0.0.0.0

(5) Connection Settings

Item	Value
Connection1 - Maintenance Communication Server	0.0.0.0
Connection2 - Maintenance Communication Server	0.0.0.0
Connection3 - Maintenance Communication Server	0.0.0.0
Connection4 - Maintenance Communication Server	0.0.0.0
Connection5 - Maintenance Communication Server	0.0.0.0
Connection6 - Maintenance Communication Server	0.0.0.0
Connection7 - Maintenance Communication Server	0.0.0.0
Connection8 - Maintenance Communication Server	0.0.0.0

For details about web server, see FC6A User's Manual - Communication Volume.

The following data can be confirmed on the PLC status page:

(1) System Information

Type number and system program version of the FC6A Series MICROSmart can be confirmed.

(2) Operating Status

Run/Stop, scan time, and error code can be confirmed. By clicking on the button in the run/stop status, the FC6A Series MICROSmart can be started or stopped.

(3) Calender/Clock

The time information obtained with SNTP can be checked here.

(4) Network Settings

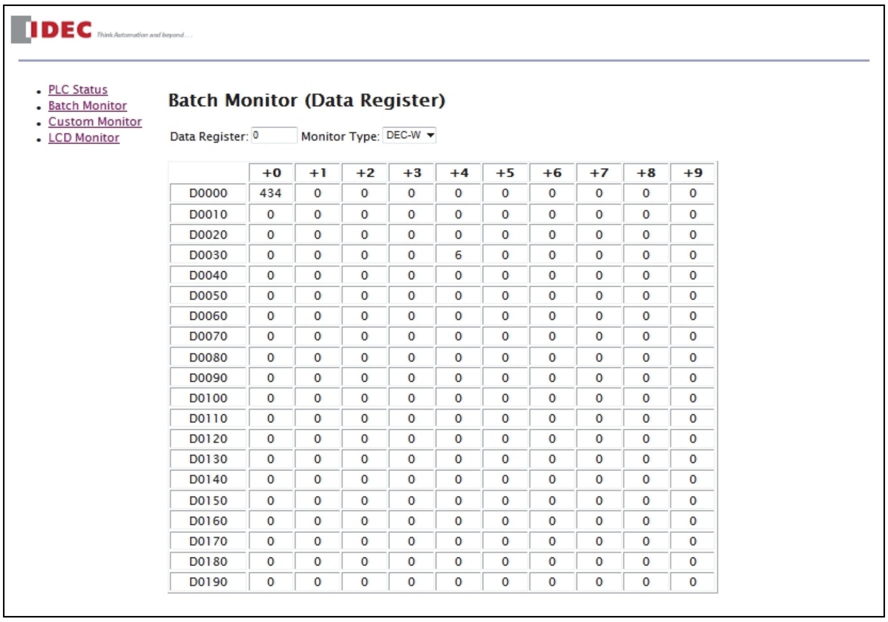
The network settings of the FC6A Series MICROSmart can be confirmed.

(5) Connection Settings

The IP addresses of the devices with which the FC6A Series MICROSmart communicates. The IP addresses of the client devices that are accessing the server ports are shown for the **Maintenance Communication Server** and the **Server Connections**. The IP addresses of the remote hosts to which the FC6A Series MICROSmart communicates are shown for the **Client Connections**. For the Plus CPU module, the IP addresses of the remote hosts on connections 1 to 16 are displayed.

■ **Batch Monitor**

In the batch monitor, 200 consecutive data registers can be monitored and controlled with the specified data type.



- **Operation procedure**

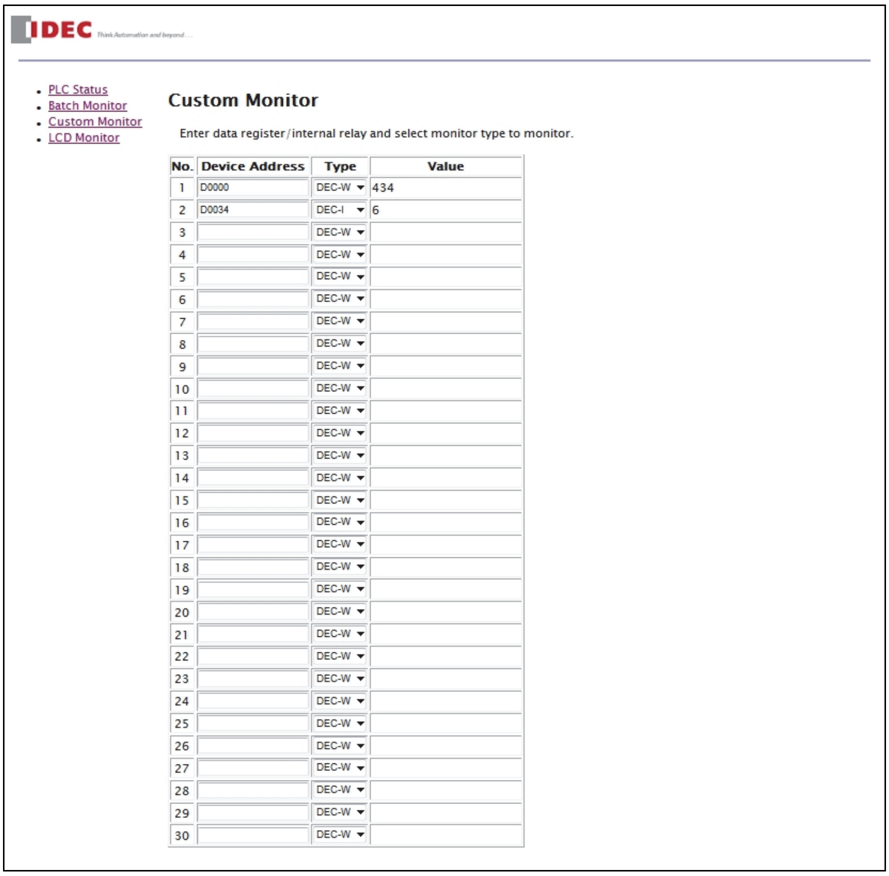
1. In **Device Number**, specify the device number to monitor.
   200 consecutive data registers starting with the specified data register are monitored. The valid device numbers are as follows. If an invalid value is entered or if the last device of the 200 consecutive data registers is out of the range, the value is corrected to a valid value automatically.

CPU Module Type	Device Address Range
All-in-One CPU module	0000 to 7800 8000 to 8300 10000 to 55800
Plus CPU module	0000 to 7800 8000 to 8700 10000 to 61800 70000 to 269800

2. Select the **Monitor Type** to display the data registers in the desired format.
   Data register values are shown in the format specified by the monitor type. See the web data type for the available data types.
3. Click on a cell in which data register value is shown.
   A dialog box to write a data is shown. The entered value is written to the FC6A Series MICROSmart.

■ **Custom Monitor**

In the custom monitor, a maximum of 30 devices and corresponding data types can be specified to monitor and control.



No.	Device Address	Type	Value
1	D0000	DEC-W	434
2	D0034	DEC-I	6
3		DEC-W	
4		DEC-W	
5		DEC-W	
6		DEC-W	
7		DEC-W	
8		DEC-W	
9		DEC-W	
10		DEC-W	
11		DEC-W	
12		DEC-W	
13		DEC-W	
14		DEC-W	
15		DEC-W	
16		DEC-W	
17		DEC-W	
18		DEC-W	
19		DEC-W	
20		DEC-W	
21		DEC-W	
22		DEC-W	
23		DEC-W	
24		DEC-W	
25		DEC-W	
26		DEC-W	
27		DEC-W	
28		DEC-W	
29		DEC-W	
30		DEC-W	

- **Operation procedure**

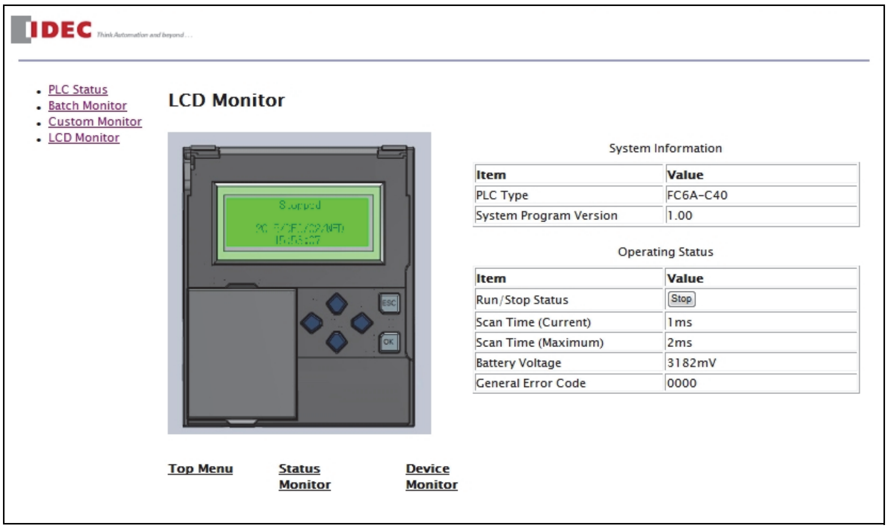
1. Enter the device (type and number) in the **Device Address**.
"D" (data register) or "m" (internal relay) can be specified. Specify data device number in decimal, such as D2058 or m0112.

2. Select the **Type** to display the corresponding device in the desired format.
Data registers or internal relays values are shown in the format specified by the monitor type. See the web data type for the available data types.

3. Click the monitored data that is displayed under **Value**.
A dialog box to write a data is shown. The entered value is written to the FC6A Series MICROSmart.

■ **LCD Monitor**

In the LCD monitor, the HMI module screen can be monitored. The HMI module can also be operated by clicking on the displayed buttons and shortcuts. This page can be displayed only when using the HMI-Ethernet port of the HMI module.



- **Operation procedure**

The monitoring procedure is as follows.

1. The menus and messages displayed on the HMI module LCD are displayed on this page.

2. Click a button to perform the same operation on the HMI module as when the HMI module's own button is pressed.

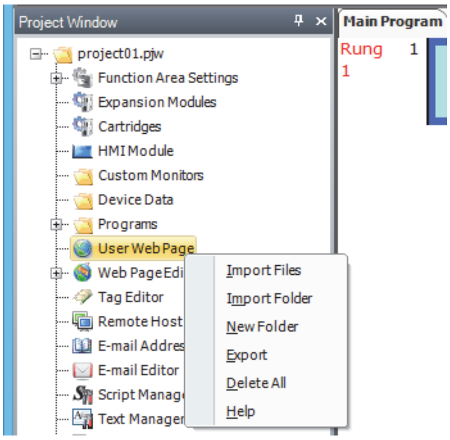
User Web Pages

FC6A Series MICROSmart user web pages can be easily created for any desired purpose or application by using the Web Page Editor tool that has been prepared in WindLDR.

Highly customizable user web pages can also be created by importing web pages that have been created in HTML.

User Web Page Tree Operations

The User Web Page tree can be built by performing the following operations on user web page items on the Project window and files and folders that are registered as web pages.



Creating a Web Page Tree

The web page tree is created on the Project window.

The following procedure describes an example of creating a web page tree with the following structure.

Web page tree structure

page1.html

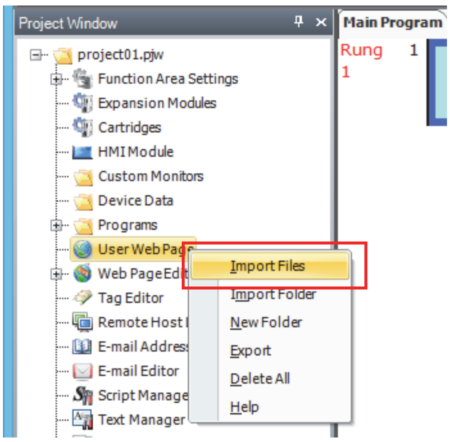
Folder/page2.html

Before this procedure, create "page1.html" and "page2.html" that will be displayed on the FC6A Series MICROSmart.

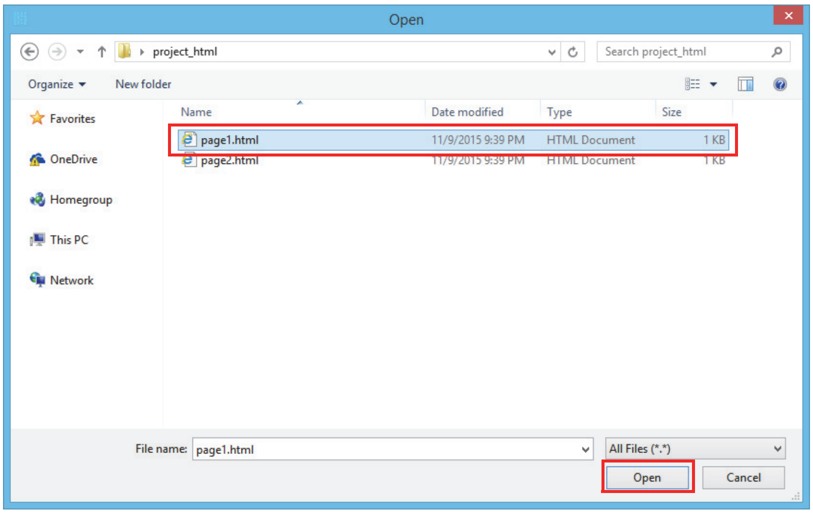
- **Operation procedure**

1. Right-click **User Web Page** and click **Import Files**.

The **Open** dialog box is displayed.

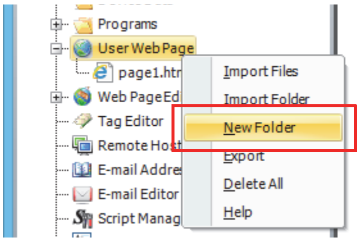


2. Select "page1.html" and click **Open**.



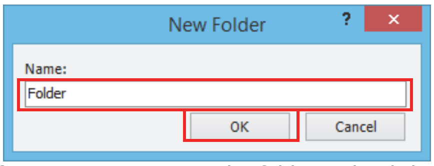
3. Right-click **User Web Page** and click **New Folder**.

The **New Folder** dialog box is displayed.



4. Enter "Folder" as the name of the folder to create and click **OK**.

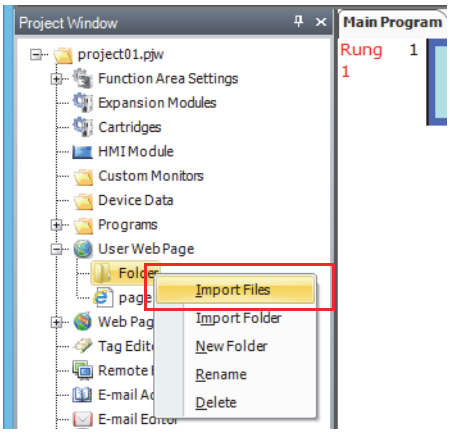
The folder titled "Folder" is created in **User Web Page**.



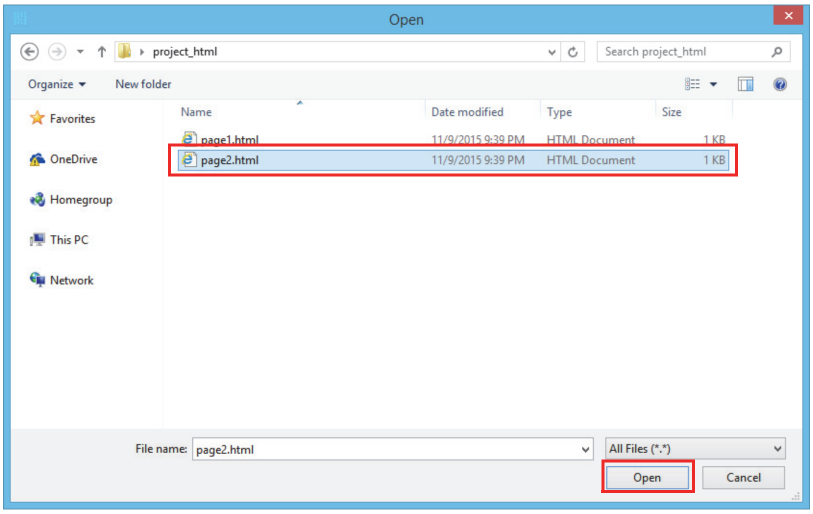
If you want to rename the folder, right click the folder and click **Rename Folder** to change the folder name.

5. Right-click **Folder** created in step 4 and click **Import Files**.

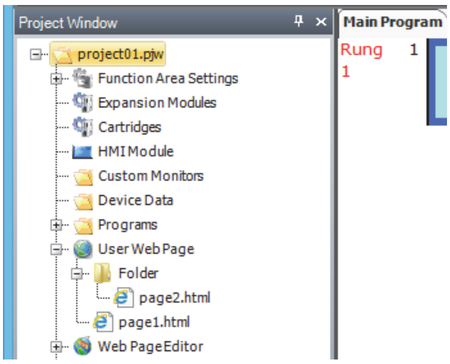
The **Open** dialog box is displayed.



6. Select "page2.html" and click **Open**.



The following is displayed in the Project Window.



Web Page Tree Import

If you already have an HTML file tree to use for building the tree, that entire tree can be imported and added to the user web pages.

In the Project window, right-click **User Web Page** or a folder under **User Web Page** and click **Import Folder**. A dialog box is displayed. Select the folder that contains the web page tree to import, and all of the files under the set folder will be imported.

Web Page Tree Export

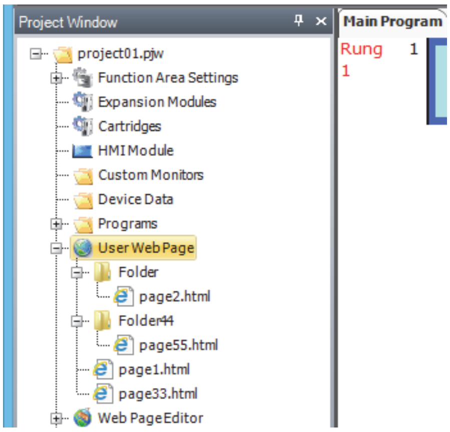
You can rebuild the user web page tree or change files by exporting the user web page data.

In the Project window, right-click **User Web Page** and click **Export**. A dialog box is displayed. Set the folder where the files will be exported, and the user web page data will be exported to that folder.

Deleting User Web Pages

The web page tree is deleted on the Project window.

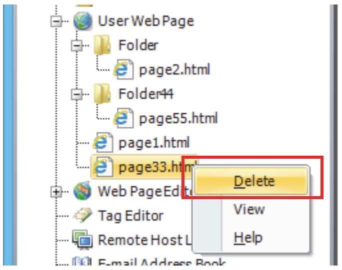
The following procedure describes an example of deleting "page33.html" and "Folder44" from the web page tree.



- **Operation procedure**

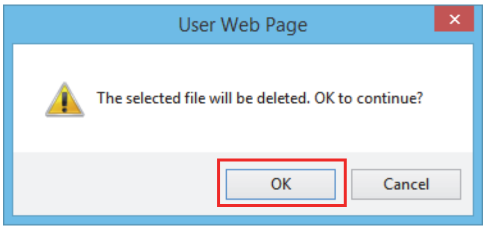
1. Right-click "page33.html" and click **Delete**.

A delete confirmation message is displayed.



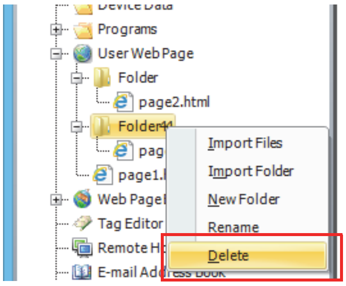
2. Click **OK**.

"Page33.html" is deleted.



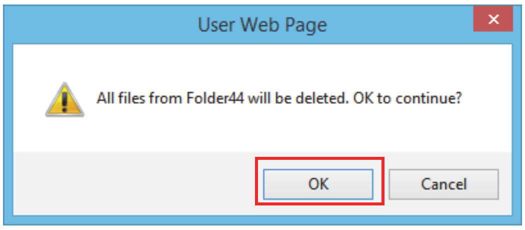
3. Right-click "Folder44" and click **Delete**.

A delete confirmation message is displayed.



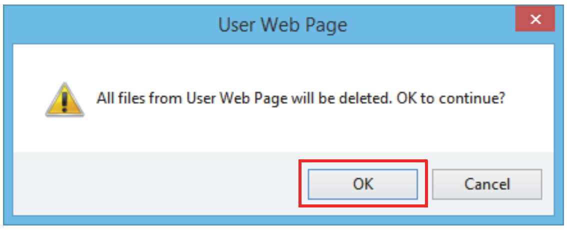
4. Click **OK**.

All of the files and folders under "Folder44" are deleted.



Notes:

- To delete all user web pages, right-click **User Web Page** in the Project window and click **Delete All**.
  The delete confirmation message is displayed. Click **OK** to delete all registered files and folders.



- Use UTF-8 as the encoding for all htm and html files that will be imported as user web pages. Separately back up all files on the PC that will be imported into WindLDR.
  The operation of the web browser may change due to the type and version of the web browser. Fully validate operation using the web browser on the device that will actually be used to access the web server.

Monitor Function

The values of data registers can be monitored without the use of complicated JavaScript or CGI.

In the **Function Area Settings** dialog box, under **Ethernet Port 1** and **Web Server Settings** or in the **HMI Module Configuration** dialog box, on the **Web Server** tab, the **Enable data register monitor** check box must be selected to use this function.

Note: Processing is added to the headers of the imported htm and html files that loads a JavaScript library when the program is converted in WindLDR. The web pages may not operate as expected in the web browser depending on the content of the imported htm and html files.
To stop loading this JavaScript library, in the HMI Module Configuration dialog box, on the Web Server tab, clear the Enable data register monitor check box, and import the files.

Metacharacter Format (Word Device)

When the monitor function is enabled and an html file includes a string with the following format, it is automatically converted to a JavaScript request and replaced with the value of the data register acquired via CGI.

No special knowledge of JavaScript is required.

{{Read/write type, device, device number, driver number, network number, web data type}}

Example: {{R,D,123,0,0,DEC-W}}

Read/write type	: R or W can be set. When R is set, the value is read only. When W is set, a value can also be written to the device.
Device	: Sets the device to monitor. The device that can be set is D (data register).
Device number	: Sets the device number to monitor as a decimal value.
Driver number	: The driver number. Set to 0.
Network number	: The network number. Set to 0.
Web data type	: Sets the web data type when the metacharacters will be replaced.

Notes:
- Strings that do not follow the metacharacter format will not be replaced with numeric values by JavaScript.
- If communication with the FC6A Series MICROSmart fails or the specified device does not exist, the strings are replaced with "----".
- The numeric value that had been replaced with the metacharacters (set to read/write type W) can be used to overwrite the data of the corresponding device in the web browser.

Example

The metacharacters are converted as follows when D2058 is 49910 (C2F6h) and D2059 is 59768 (E978h).

{{W,D,2058,0,0,HEX-W}}	→	C2F6
{{W,D,2058,0,0,HEX-D}}	→	C2F6E978
{{W,D,2058,0,0,DEC-W}}	→	49910
{{W,D,2058,0,0,DEC-I}}	→	-15626
{{W,D,2058,0,0,DEC-L}}	→	-1024005768
{{W,D,2058,0,0,DEC-F}}	→	-123.456

When you click the numeric values in the previous examples, a dialog box will be displayed. A numeric value entered in this dialog box will be reflected in the device.

Metacharacter Format (Bit Device)

When the monitor function is enabled and an html file includes a string with the following format, it is automatically converted to a JavaScript request and replaced with the status of the bit device acquired via CGI.

{{Read/write type, device, device number, driver number, network number, web data type, OFF status display string, ON status display string}}

Example: {{R,m,123,0,0,BIT_FORM,BITOFF,BITON}}

Read/write type	:	R or W can be set. When R is set, the value is read only. When W is set, the value becomes a form button and the status that was read from the device is displayed on the button. Click the button to turn the device on or off.
Device	:	Sets the device to monitor. The device that can be set is m (internal relay).
Device number	:	Sets the device number to monitor as a decimal value.
Driver number	:	The driver number. Set to 0.
Network number	:	The network number. Set to 0.
Web data type	:	Sets the web data type when the metacharacters will be replaced. BIT_FORM is set here.
OFF status display string	:	Sets the string that is displayed when the set bit device is off.
ON status display string	:	Sets the string that is displayed when the set bit device is on.

Metacharacter Format (Bit Device: Image Display)

When the monitor function is enabled and an html file includes a string with the following format, it is automatically converted to a JavaScript request and replaced with the status of the bit device acquired via CGI.

{{Read/write type, device, device number, driver number, network number, web data type, OFF status image file name, ON status image file name}}

Example: {{R,m,123,0,0,BIT,./BIT_IMG_OFF.jpg,./BIT_IMG_ON.jpg}}

Read/write type	:	R or W can be set. When R is set, the value is read only. When W is set, click the image file to turn the device on or off.
Device	:	Sets the device to monitor. The device that can be set is m (internal relay).
Device number	:	Sets the device number to monitor as a decimal value.
Driver number	:	The driver number. Set to 0.
Network number	:	The network number. Set to 0.
Web data type	:	Sets the web data type when the metacharacters will be replaced. BIT is set here.
OFF status image file name	:	With a relative path, sets the file name of the image to display when the set bit device is off.
ON status image file name	:	With a relative path, sets the file name of the image to display when the set bit device is on.

Metacharacter Format (String)

When the monitor function is enabled and an html file includes a string with the following format, it is automatically converted to a JavaScript request and replaced with the string acquired via CGI.

{{Read/write type, device, device number, driver number, network number, format, maximum size}}

Example: {{R,D,123,0,0,STR_ASCII,20}}

Read/write type	:	R or W can be set. When R is set, the value is read only. When W is set, the value can also be written.
Device	:	Sets the device to monitor. The device that can be set is D (data register).
Device number	:	Sets the device number to monitor as a decimal value.
Driver number	:	The driver number. Set to 0.
Network number	:	The network number. Set to 0.
Format	:	Sets type of string that will be replaced. The string type that can be specified is STR_HOSTNAME, STR_EMAILADR, STR_PASSWORD, and STR_ASCII.
Maximum size	:	Sets the maximum size in bytes to read or write. Data that exceeds this size cannot be read or written. The minimum value is 1 and the maximum value is 63.

The string that can be entered and the display in the browser differs by each string type.

String Type	String to Enter and Display
STR_HOSTNAME	Used when reading or writing a host name.
STR_EMAILADR	Used when reading or writing an e-mail address.
STR_PASSWORD	Used when entering a password. The string is masked.
STR_ASCII	Used when reading or writing an ASCII string other than those listed above.

When STR_HOSTNAME and STR_EMAILADR are specified as read/write type W, a check is performed to determine if the entered characters and string are in the appropriate format for a host name or e-mail address, and the data is only written if they are appropriate. The format is not checked for data that was read.

When STR_PASSWORD is set, masked characters determined by the web browser are displayed, not the actual characters. The characters entered in the write dialog box are also masked.

Notes:

- Strings that do not follow the metacharacter format will not be replaced with strings by JavaScript.
- If communication with the FC6A Series MICROSmart fails or the specified device does not exist, the strings are replaced with "-".
- The strings that had been replaced with the metacharacters (set to read/write type W) can be written to corresponding devices with the web browser.
- Strings are set as continuous data registers of 1 word or more. 2 bytes of data are 1 word and they are set from upper byte to lower byte in order. Set 00h at the end of the string.
- When STR_ASCII, STR_HOSTNAME, or STR_EMAILADR is set and the starting byte of the data register that was read is 00h, "blank" is displayed as substitute text.

Notes:

- When reading, the set maximum size of devices is read and the string is replaced from the starting byte up to 00h. The number of devices that are read in words is the maximum size / 2 words (rounded up). If 00h is not included in the data that was read, the string is replaced as if 00h is at maximum size + 1 byte. Note that the maximum size is set in bytes but reading is performed in words.
- When writing, the string entered in the dialog box is written. The data after the end of the entered string is filled with 00h up to the character at maximum size + 1. The number of devices that are written in words is (maximum size + 1) / 2 words (rounded up). If the byte at maximum size + 1 is an odd number, the final word is written as 0000h. If a blank string is set in the dialog, 00h is written to all bytes of data. Note that the maximum size is set in bytes but writing is performed in words.

Example

When D2058 is 'a' 'b' (6162h), D2059 is 'c' 'd' (6364h), and D2060 is 'e' 'f' (6566h) , the metacharacters are replaced as follows.

{{W,D,2058,0,0,STR_ASCII,3}} → abc
{{W,D,2058,0,0,STR_ASCII,4}} → abcd

When you click the strings in the previous examples, a dialog box will be displayed. The string entered in this dialog box will be reflected in the devices.

Example of written data when the maximum number of bytes is odd

For {{W,D,2058,0,0,STR_ASCII,3}}, 2 words of data will be written.

Set Values				
Data Register		D2058	D2059	D2060
Before writing	ASCII	‘a’ ‘b’	‘c’ ‘d’	‘e’ ‘f’
	Value (hexadecimal)	6162h	6364h	6566h
Write A	ASCII	‘A’ ‘\0’	‘\0’ ‘\0’	‘e’ ‘f’
	Value (hexadecimal)	4100h	0000h	6566h
Write AB	ASCII	‘A’ ‘B’	‘\0’ ‘\0’	‘e’ ‘f’
	Value (hexadecimal)	4142h	0000h	6566h
Write ABC	ASCII	‘A’ ‘B’	‘C’ ‘\0’	‘e’ ‘f’
	Value (hexadecimal)	4142h	4300h	6566h

Example of written data when the maximum number of bytes is even

For {{W,D,2058,0,0,STR_ASCII,4}}, 3 words of data will be written.

Set Values				
Data Register		D2058	D2059	D2060
Before writing	ASCII	'a' 'b'	'c' 'd'	'e' 'f'
	Value (hexadecimal)	6162h	6364h	6566h
Write A	ASCII	'A' '\0'	'\0' '\0'	'\0' '\0'
	Value (hexadecimal)	4100h	0000h	0000h
Write AB	ASCII	'A' 'B'	'\0' '\0'	'\0' '\0'
	Value (hexadecimal)	4142h	0000h	0000h
Write ABC	ASCII	'A' 'B'	'C' '\0'	'\0' '\0'
	Value (hexadecimal)	4142h	4300h	0000h
Write ABCD	ASCII	'A' 'B'	'C' 'D'	'\0' '\0'
	Value (hexadecimal)	4142h	4344h	0000h

Note: Metacharacter Format Precautions

The entire string between {{ and }} is recognized as the metacharacters. It cannot contain extra spaces, line breaks, or characters, such as HTML tags.

For example, the metacharacters cannot be coded as follows when attempting to emphasize the numeric value. This violates the metacharacter format, and it will not be replaced with the numeric value.

```
{{<em>W,D,2058,0,0,HEX-D</em>}}
```

To replace the string and emphasize the value, code the metacharacter format as follows.

```
<em>{{W,D,2058,0,0,HEX-D}}</em>
```

Bar Graph (Vertical)

When the monitor function is enabled and the div tag and parameters are coded on the HTML file, it will be automatically converted to a JavaScript request and the values of the data registers acquired via CGI will be displayed in a vertical bar graph.

To draw a vertical bar graph, first set the id attribute to a string that indicates a unique ID for each graph, set the data-graph attribute to "vbar" that indicates the type of data, and code the parameters in the div tag.

The parameters that can be set are as follows. If the default value of the parameter is acceptable, it does not need to be set.

device	:	The device. Set this to "D" (data register).
address	:	The device number. Set this as a numeric value.
driver	:	The driver number. Set this to 0.
net_no	:	The network number. Set this to 0.
format	:	The web data type string.
width	:	The width of the div border in pixels. The default value is 300.
height	:	The height of the div border in pixels. The default value is 300.
line_col	:	The line color of the div border. The default value is "#000000".
barvgutter	:	The vertical space from the div border to the graph in pixels. The default value is 30.
barwidth	:	The width of the graph in pixels. The default value is 20.
gutter	:	The space between graphs in pixels when displaying multiple graphs. The default value is 20.
type	:	The shape of the ends of the graph. Select "square", "round", "sharp", or "soft". The default setting is "square".
scalewidth	:	The width of the scale in pixels. The default value is 5.
labelvgutter	:	The vertical space from the graph where the label will be displayed in pixels. The default value is 20.
labelhgutter	:	The horizontal space from the graph where the label will be displayed in pixels. The default value is 30.
bars[]	:	An array of parameters that configures each graph. Code the parameters for each array in { } and separate each array element with ",".

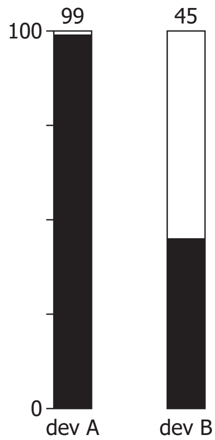
The parameters that can be set as an element of bars[] are as follows. If the default value of the parameter is acceptable, it does not need to be set.

min_val	:	The minimum value of the graph. This parameter is required.
max_val	:	The maximum value of the graph. This parameter is required.
label	:	The label name. The default is a string that combines the device and device number.
back_col	:	The background color of the graph. The default value is "#FFFFFF".
front_col	:	The foreground color of the graph. The default value is a color that differs for each element.
scale_on	:	This parameter sets whether or not the graph has a scale. The default setting is false (no scale).
scale_lbl_on	:	This parameter sets whether or not there is a label for the scale on the graph. The default setting is false (no label).

When the div tag is coded as follows, two vertical bar graphs are displayed using the data in double words D2040 and D2042 (decimal).

```
<div id="div11" data-graph="vbar">

device:"D", address:2040, driver:0, net_no:0, format:"DEC-D",
width:300, height:300, line_col:"#000000",
barvgutter:30, barwidth:20,
gutter:20, type:"square", scalewidth:5,
bars:[
{
  min_val:0, max_val:100, label:"dev A",
  back_col:"#FFFFFF", front_col:"#0000FF",
  scale_on:true, scale_lbl_on:true
},
{
  min_val:0, max_val:100, label:"dev B",
  back_col:"#FFFFFF", front_col:"#FF0000",
  scale_on:false, scale_lbl_on:false
}
]</div>
```



Bar Graph (Horizontal)

When the monitor function is enabled and the div tag and parameters are coded on the HTML file, it will be automatically converted to a JavaScript request and the values of the data registers acquired via CGI will be displayed in a horizontal bar graph.

To draw a horizontal bar graph, first set the id attribute to a string that indicates a unique ID for each graph, set the data-graph attribute to "hbar" that indicates the type of data, and code the parameters in the div tag.

The parameters that can be set are as follows. If the default value of the parameter is acceptable, it does not need to be set.

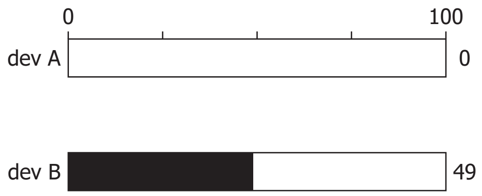
device	:	The device. Set this to "D" (data register).
address	:	The device number. Set this as a numeric value.
driver	:	The driver number. Set this to 0.
net_no	:	The network number. Set this to 0.
format	:	The web data type string.
width	:	The width of the div border in pixels. The default value is 300.
height	:	The height of the div border in pixels. The default value is 300.
line_col	:	The line color of the div border. The default value is "#000000".
barhgutter	:	The horizontal space from the div border to the graph in pixels. The default value is 30.
barwidth	:	The width of the graph in pixels. The default value is 20.
gutter	:	The space between graphs in pixels when displaying multiple graphs. The default value is 20.
type	:	The shape of the ends of the graph. Select "square", "round", "sharp", or "soft". The default setting is "square".
scalewidth	:	The width of the scale in pixels. The default value is 5.
labelvgutter	:	The vertical space from the graph where the label will be displayed in pixels. The default value is 20.
labelhgutter	:	The horizontal space from the graph where the label will be displayed in pixels. The default value is 30.
bars[]	:	An array of parameters that configures each graph. Code the parameters for each array in { } and separate each array element with ",".

The parameters that can be set as an element of bars[] are as follows. If the default value of the parameter is acceptable, it does not need to be set.

min_val	:	The minimum value of the graph. This parameter is required.
max_val	:	The maximum value of the graph. This parameter is required.
label	:	The label name. The default is a string that combines the device and device number.
back_col	:	The background color of the graph. The default value is "#FFFFFF".
front_col	:	The foreground color of the graph. The default value is a color that differs for each element.
scale_on	:	This parameter sets whether or not the graph has a scale. The default setting is false (no scale).
scale_lbl_on	:	This parameter sets whether or not there is a label for the scale on the graph. The default setting is false (no label).

When the div tag is coded as follows, two horizontal bar graphs are displayed using the data in double words D2040 and D2042 (decimal).

```
<div id="div21" data-graph="hbar">
  device:"D", address:2040, driver:0, net_no:0, format:"DEC-D",
  width:300, height:300, line_col:"#000000",
  barhgutter:30, barwidth:20,
  gutter:20, type:"square", scalewidth:5,
  bars:[
    {
      min_val:0, max_val:100, label:"dev A",
      back_col:"#FFFFFF", front_col:"#0000FF",
      scale_on:true, scale_lbl_on:true
    },
    {
      min_val:0, max_val:100, label:"dev B",
      back_col:"#FFFFFF", front_col:"#FF0000",
      scale_on:false, scale_lbl_on:false
    }
  ]
</div>
```



Trend Graph

When the monitor function is enabled and the div tag and parameters are coded on the HTML file, it will be automatically converted to a JavaScript request, and the values of the data registers acquired via CGI will be displayed in a trend graph.

To draw a trend graph, first set the id attribute to a string that indicates a unique ID for each graph, set the data-graph attribute to "trend" that indicates the type of data, and code the parameters in the div tag.

The parameters that can be set are as follows. If the default value of the parameter is acceptable, it does not need to be set.

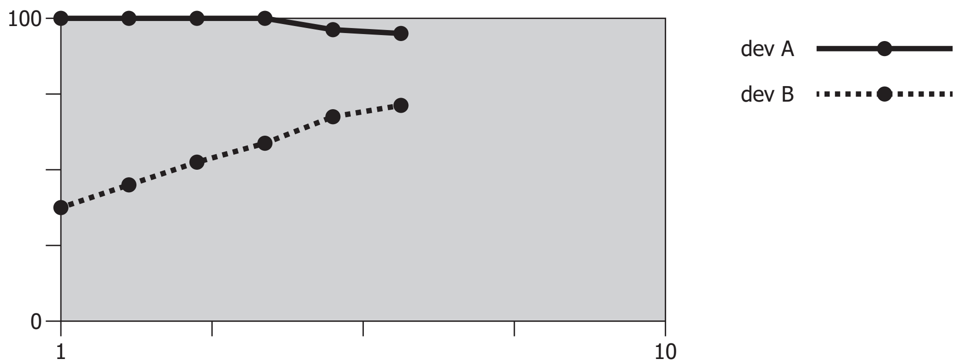
device	:	The device. Set this to "D" (data register).
address	:	The device number. Set this as a numeric value.
driver	:	The driver number. Set this to 0.
net_no	:	The network number. Set this to 0.
format	:	The web data type string.
width	:	The width of the div border in pixels. The default value is 300.
height	:	The height of the div border in pixels. The default value is 300.
line_col	:	The line color of the div border. The default value is "#000000".
min_val	:	The minimum value of the graph. This parameter is required.
max_val	:	The maximum value of the graph. This parameter is required.
plot_num	:	This parameter sets the number of plots of data. This parameter is required.
scale_col	:	The color of the scale. The default value is "#000000".
x_val_col	:	The color of the X-axis label. The default value is "#000000".
y_val_col	:	The color of the Y-axis label. The default value is "#000000".
g_x	:	The position to start drawing the graph border in the horizontal direction in pixels. The default value is 50.
g_y	:	The position to start drawing the graph border in the vertical direction in pixels. The default value is 30.
g_width	:	The horizontal width of the graph border in pixels. The default value is 200.
g_height	:	The vertical of the graph border in pixels. The default value is 100.
g_line_col	:	The color of the graph border. The default value is the same as line_col.
g_line_width	:	The thickness of the graph border in pixels. The default value is 1.
g_back_col	:	The background of the graph border. The default value is "#C0C0C0".
mode	:	This parameter sets the data that will be truncated when the data exceeds the plot area. Select "one" (only one will be truncated), "half" (half of the entire amount will be truncated), or "all" (all will be truncated). The default setting is "all".
legend_gutter	:	The width from the graph border to the legend in pixels. The default value is 30.
legend_margin_x	:	The horizontal margin inside the legend in pixels. The default value is 10.
legend_margin_y	:	The vertical margin inside the legend in pixels. The default value is 10.
legend_line_width	:	The length of the line inside the legend in pixels. The default value is 20.
legend_line_gutter	:	The vertical width in pixels when multiple lines are displayed inside the legend. The default value is 20.
legend_line_col	:	The color of the lines inside the legend. The default value is the same as line_col.
legend_width	:	The horizontal width inside the legend in pixels. The default value is 100.
line_width	:	The initial value for the thickness of lines on the graph in pixels. The default value is 1.
marker_on	:	The initial setting for whether or not to display markers on the graph. The default value is false (no markers).
marker_width	:	The initial value for the size of markers on the graph in pixels. The default value is 3.
lines[]	:	An array of parameters that configures each line. Code the parameters for each array in { } and separate each array element with ",".

The parameters that can be set as an element of lines[] are as follows. If the default value of the parameter is acceptable, it does not need to be set.

label	:	The label name. The default is a string that combines the device type and device number.
front_col	:	The foreground color of the line that can be set for each line. The default value is a color that differs for each element.
marker_col	:	The marker color of the line. The default value is the same as front_col.
line_width	:	The thickness of the line in pixels that can be set for each line. This parameter has priority over the initial value of line_width.
marker_width	:	The size of the marker in pixels that can be set for each line. This parameter has priority over the initial value of marker_width.

When the div tag is coded as follows, the trend graph is displayed with 10 plots and markers using the data in double words D2040 and D2042 (decimal).

```
<div id="div31" data-graph="trend">
  device:"D", address:2040, driver:0, net_no:0, format:"DEC-D",
  width:400, height:300, line_col:"#000000",
  min_val:0, max_val:100, plot_num:10,
  line_width:3, marker_on:true, marker_width:5,
  lines:[
    {
      label:"dev A",
      front_col:"#0000FF"
    },
    {
      label:"dev B",
      front_col:"#FF0000"
    }
  ]
</div>
```



Notes:

Notes on Drawing Graphs

- The string between <div> and </div> that describes the graph is passed as a JavaScript program parameter. Note that when coding the div tag, the content can contain spaces and line breaks for formatting, but it cannot contain comments.
- If a numeric value will not be set as a parameter, enclose it with "".
- All parameters must be separated with ",". Do not put "," after the last parameter. The div tag will not be drawn as a graph if the parameter format is invalid.
- The manner in which the graph is displayed may differ according to the type of web browser.
- There may be a delay in updating the web browser display due to the update frequency and the number of data plots.

JavaScript Functions

When the monitor function is enabled, device data can be read and written using JavaScript functions.
The raw data obtained via CGI can be used and processed in a more complicated manner than that when using metacharacters.
See the appendix for details about the CGI interface.

■ Read Device Data Function

Idec.device_read (device, address, length, driver, net_no)

device	:	Sets the device to read as a string. Currently only D (data register) can be used.
address	:	Sets the device number to read.
length	:	Sets the size of the data from the start of address to read. Set this as a decimal value between 1 and 64.
driver	:	The driver number. Set this to 0.
net_no	:	The network number. Set this to 0.

The return value is a hexadecimal numeric value in a string with the format "XXXX". If length is 2 or larger, "_" is inserted between items of data. The return value is "" if there was no response from the server or if the parameters are incorrect.

Note: The function is as follows when D2058 is 49910 (C2F6h) and D2059 is 59768 (E978h).

var raw_data = idec.device_read (“D”, 2058, 2, 0, 0);

The variable raw_data will be "C2F6_E978" when the function normally terminates.

■ Write Device Data Function

idec.device_write (device, address, length, driver, net_no, data)

device	:	Sets the device to write as a string. Currently only D (data register) can be used.
address	:	Sets the device number to write.
length	:	Sets the size of the data from the start of address to write. Set this as a decimal value between 1 and 64.
driver	:	The driver number. Set this to 0.
net_no	:	The network number. Set this to 0.
data	:	Sets the data to write as a hexadecimal numeric value in a string. If length is 2 or larger, insert "_" between items of data.

The return value is true when the function normally terminates and false in all other cases.

Note: The function is as follows when writing 49910 (C2F6h) to D2058 and 59768 (E978h) to D2059.

var status_write = idec.device_write (“D”, 2058, 2, 0, 0, “C2F6_E978”);

The variable status_write is true when the function normally terminates.

Web Page Editor

Web Page Editor is used to create user web pages that are downloaded to the FC6A Series MICROSmart.

User web pages that have been downloaded to the FC6A Series MICROSmart can be accessed with a web browser from a PC or other device, and the user web pages can be displayed according to FC6A Series MICROSmart device values.

To use Web Page Editor, click the HMI module inserted into the module configuration area, click **Configure**, and in the **HMI Module Configuration** dialog box, select **Use Web Page Editor**.

The behavior of user web pages according to the device values in FC6A Series MICROSmart can be checked without downloading the user web pages to the FC6A Series MICROSmart by linking the WindLDR simulation function to the created user web pages.

Web Page Editor uses the web browser. The recommended web browsers are as follows.

- Google Chrome 47 or later
- Mozilla Firefox 42 or later
- Microsoft Internet Explorer 11

Configure Windows so that these web browsers are the default web browser. This can be set on the **Programs** tab in Internet Options.

Notes:

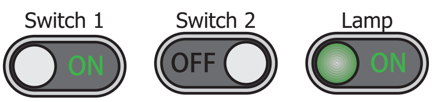
- Web Page Editor can be used with other web browsers that are not recommended, but problems may occur, such as with the display of images.
- The file names of user web pages created with the Web Page Editor cannot use the following characters.
  \+ # %

Creating User Web Pages

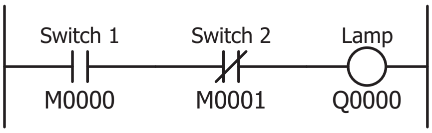
This section describes the procedure to create user web pages in Web Page Editor.

A web page will be created that turns on output Q1 when input M0000 is on and input M0001 is off.

Components to place on the web page



Program to create in WindLDR

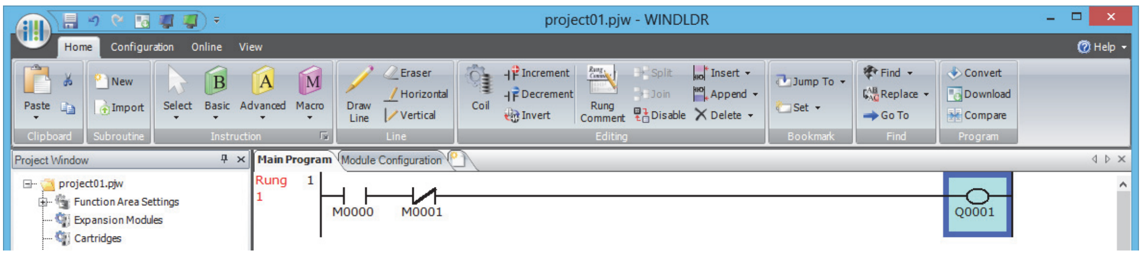


Note:
For details on devices, see "Devices" on page 2-1.
For details on instructions, see the "FC6A Series MICROSmart Ladder Programming Manual".

- **Operation procedure**

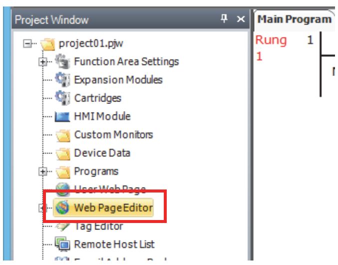
1. Create a program that turns on output Q1 when input M0000 is on and input M0001 is off.

For details, see Chapter 4 "Create Ladder Program" in the "FC6A Series MICROSmart User's Manual".

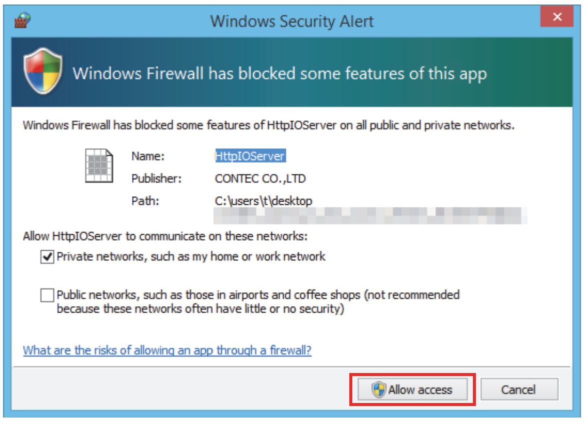


2. Double-click **Web Page Editor** in the Project window.

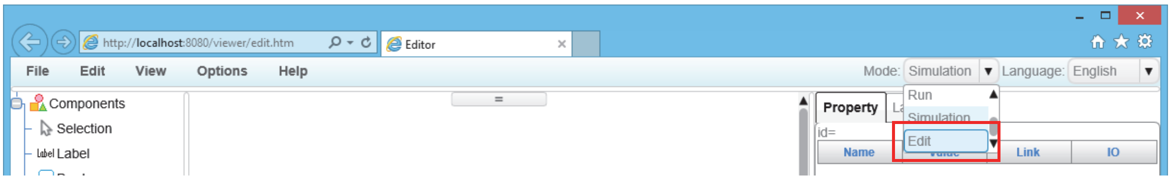
Web Page Editor starts.



Note: If the **Windows Security Alert** dialog box is displayed, click **Allow access**.



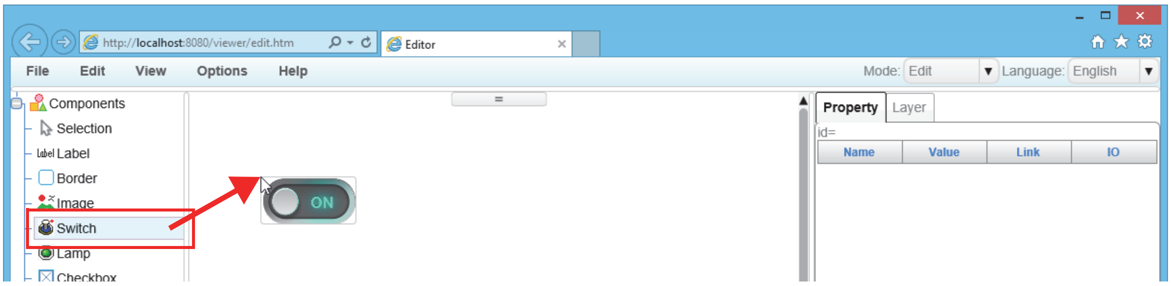
3. For the Web Page Editor **Mode**, click **Edit**.



4. Place Switch 1 on the work area and set it to a normally open contact (M0000).

   Select **Switch** in the **Components** list, and drag and drop it on the work area.

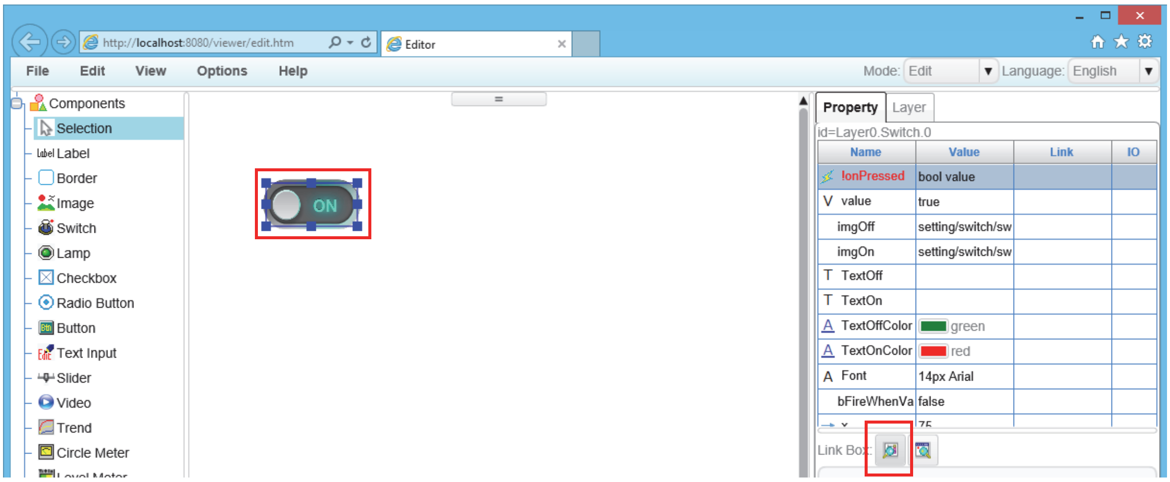
   The switch will be placed on the area with the predefined size.



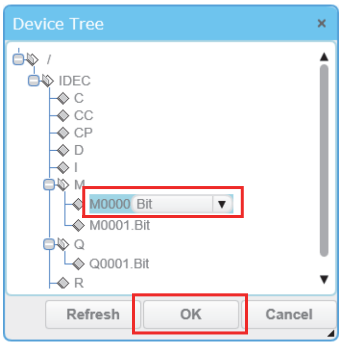
Note: To place a switch on the area with the desired size, click **Switch** in the **Components** list, and then drag the precision select mouse cursor on the work area.

5. Click the switch placed on the work area and on the **Property** tab, for **Link Box**, click .

   The **Device Tree** dialog box is displayed.



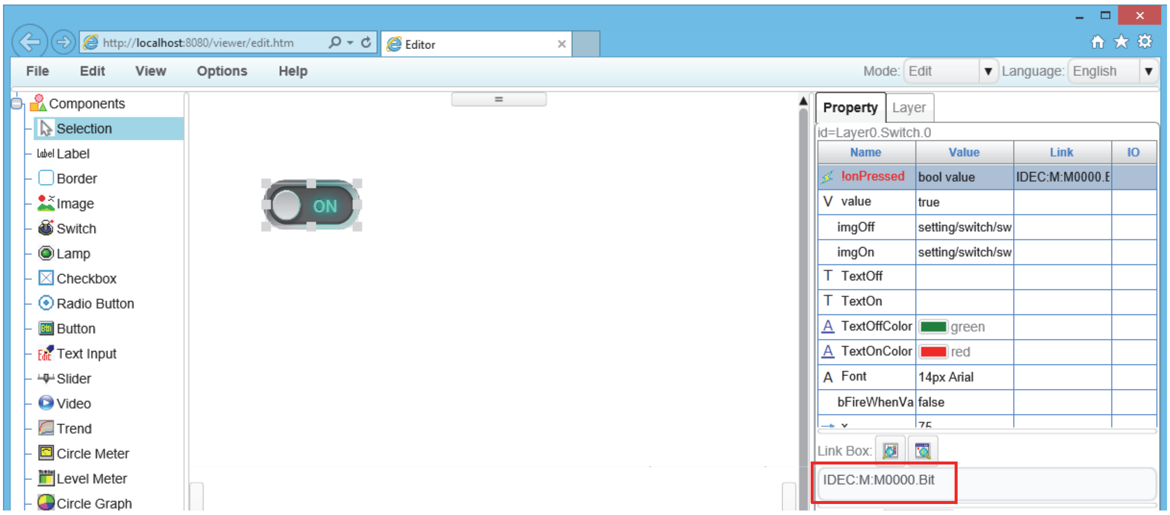
6. Click the device address to set to Switch 1 and click **OK**.



Notes:

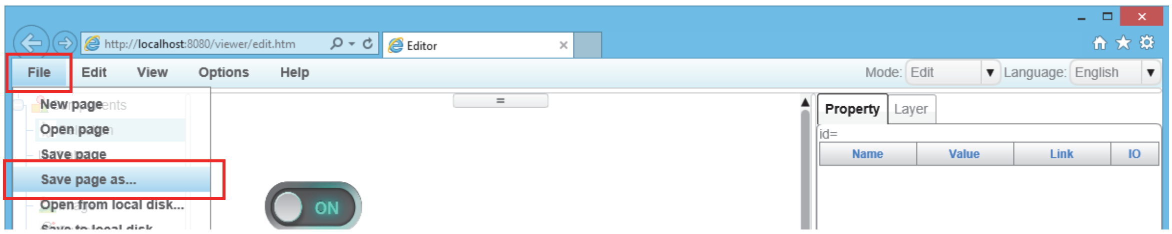
- The devices that are used in WindLDR are displayed in the **Device Tree** dialog box.
- If the device address to set is not in the Device Tree, click **Cancel**, and then directly enter the device address in the text box for **Link Box** on the **Property** tab.

This sets Switch 1 to a normally open contact (M0000).

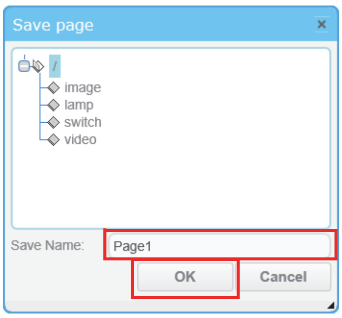


7. Repeat steps 4 through 6 and set Switch 2 and the lamp.

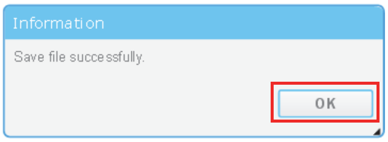
8. On the menu bar, click **File** > **Save page as**.
The **Save Page** dialog box is displayed.



9. Enter the file name in **Save Name**, specify the folder to save to, and click **OK**.
A confirmation message is displayed.



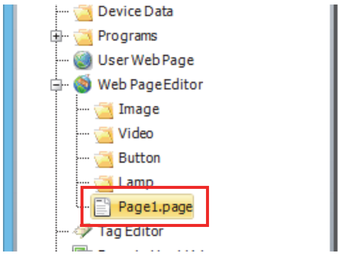
10. Click **OK**.



This concludes creating a user web page.

Notes:

- For details on Web Page Editor, refer to the Web Page Editor help that is displayed by clicking **Help** > **Help** on the menu bar.
- The created user web page will be created in Web Page Editor on the Project window. Image files, video files, image files for switches, and image files for lamps that have been imported with user web pages are displayed in their respective folders on the list.
  Double-clicking a created user web page will start Web Page Editor and open that user web page. The created user web page can then be edited.

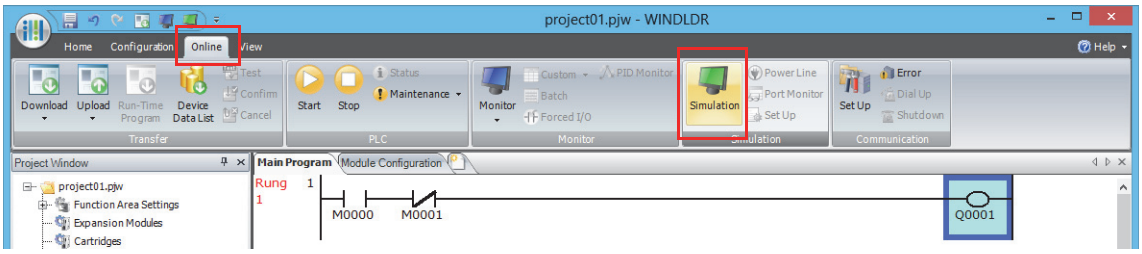


Checking Operation of User Web Pages

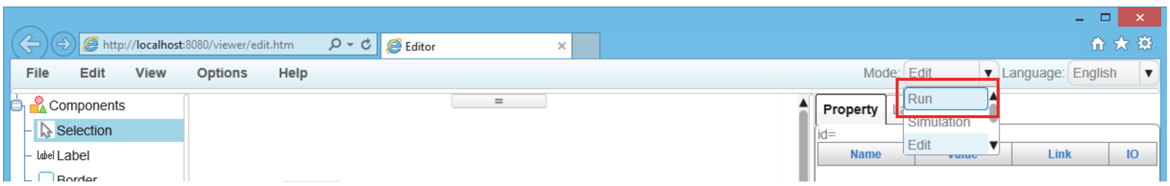
This section describes the procedure to execute operation of the created user web page in Web Page Editor to check its behavior.

- **Operation procedure**

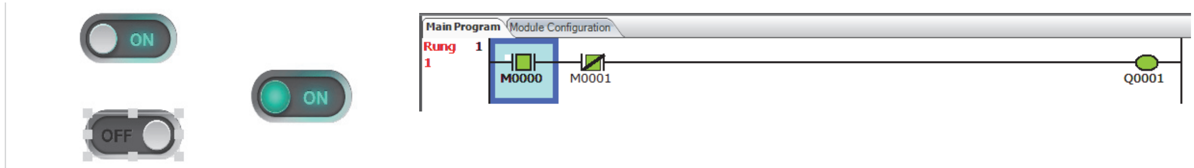
1. On the WindLDR **Online** tab, click **Simulation**.



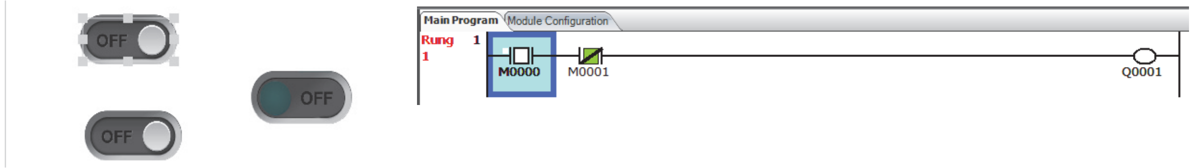
2. For the Web Page Editor **Mode**, click **Run**.



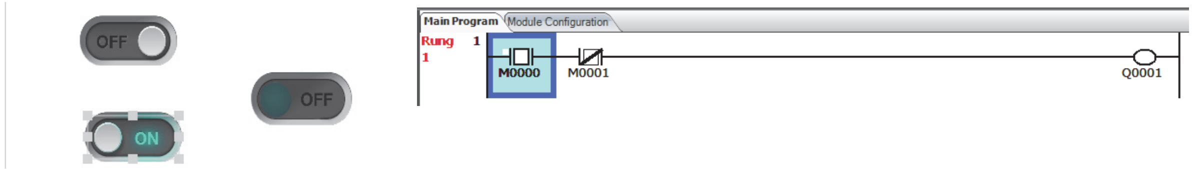
3. Check the following operation while turning Switch 1 and Switch 2 on and off.
   - The lamp turns on when Switch 1 is on and Switch 2 is off



   - The lamp turns off when Switch 1 is off



   - The lamp turns off when Switch 2 is on



Note: The WindLDR program also turns on and off according to the status of Web Page Editor.

This concludes checking operation of the user web page.
You can view the user web page from a web browser if you download the user program in WindLDR to the FC6A Series MICROSmart after saving the created web page.
For detailed configuration methods, see "HMI Module Web Server Settings" on page 13-5.

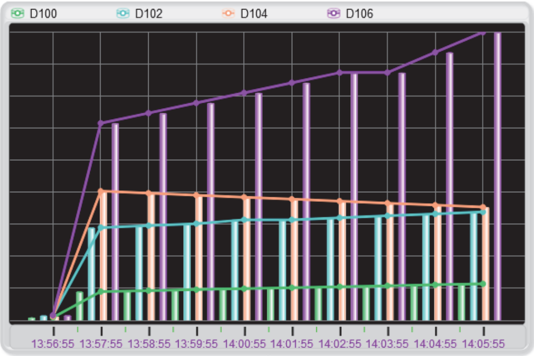
Log Data Integration on User Web Pages

Log data can be integrated on user web pages created with Web Page Editor. The log data that can be integrated is as follows.

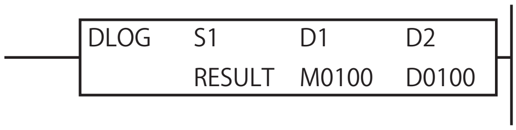
- Log data saved to the SD memory card with the DLOG (Data Log) instruction.

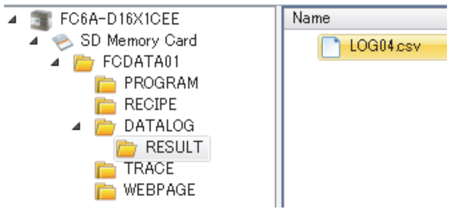
This section describes the operation procedure to reference log data saved to the SD memory card with the DLOG (Data Log) instruction from user web pages created with Web Page Editor.

Component to place on the web page



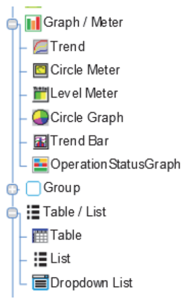
Program to create in WindLDR





Notes:

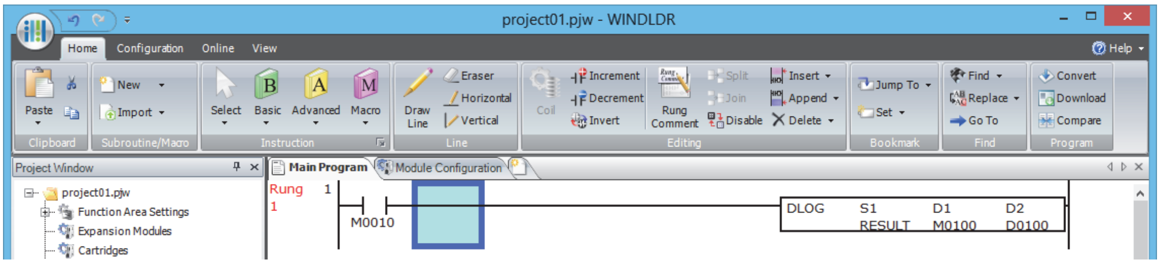
- For details on the DLOG (Data Log) instruction, see Chapter 25 "DLOG (Data Log)" in the "FC6A Series MICROSmart Ladder Programming Manual".
- Log data integration on user web pages can be used only with internal memory on the Plus CPU module and the SD memory card inserted into the Plus CPU module. Log data cannot be integrated on user web pages saved to the internal memory on the HMI module.
- Log data cannot be integrated on user web pages if the separating character for log data in the CSV file format is set to semicolon (;) in the external memory device settings. For details on the external memory device settings, see Chapter 5 "Functions and Settings" in the "FC6A Series MICROSmart User's Manual".
- Web Page Editor has operation status graph, table, list, and drop-down list components that can be used as components for log data integration on user web pages. Log data integration can also be set using these components.



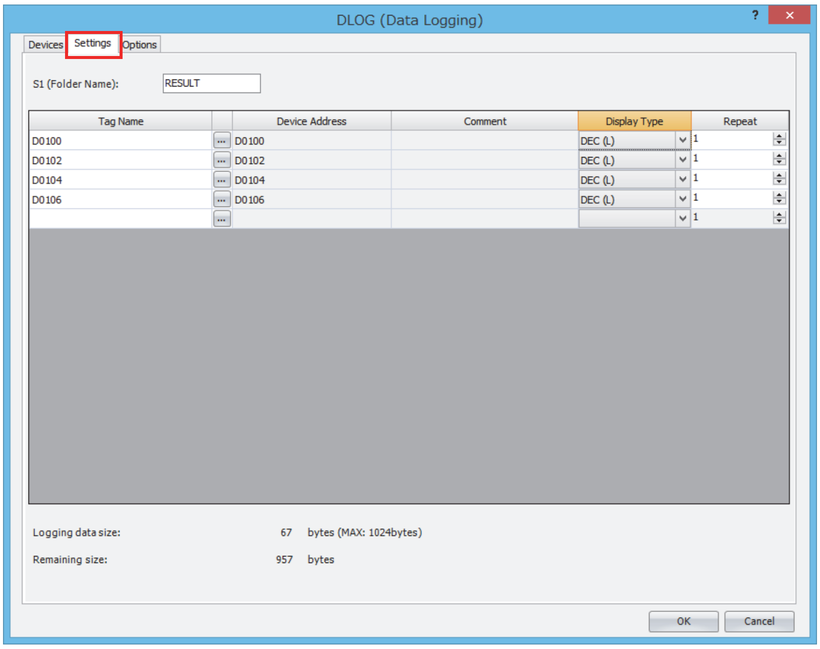
- Loge data cannot be integrated on user web page on Simulation of WindLDR.

- **Operation procedure**

1. Insert the DLOG instruction.

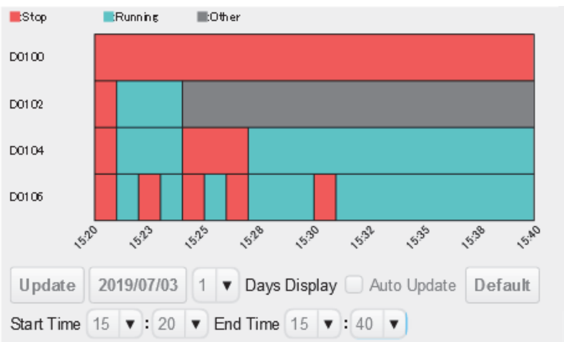


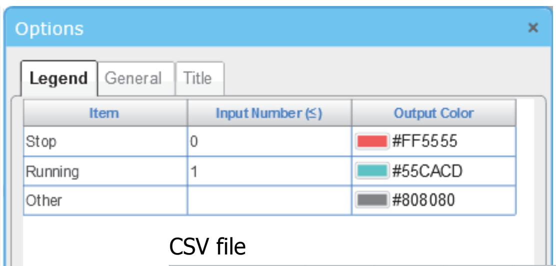
2. On the **Settings** tab, set the device addresses of the data to output to CSV file.
The values of the set device addresses are displayed on the components integrated with the log data on the user web pages.



Note: To use the operation status graph component in the log data integration components on user web pages, create a ladder program so the values of data output to the CSV file with the DLOG instruction match the input values of the operation status graph.
In this example, the value of the data that was output to the CSV file is output using the operation status graph component with the operation status as 0 ("Stop") and 1 ("Running").

Operation status graph

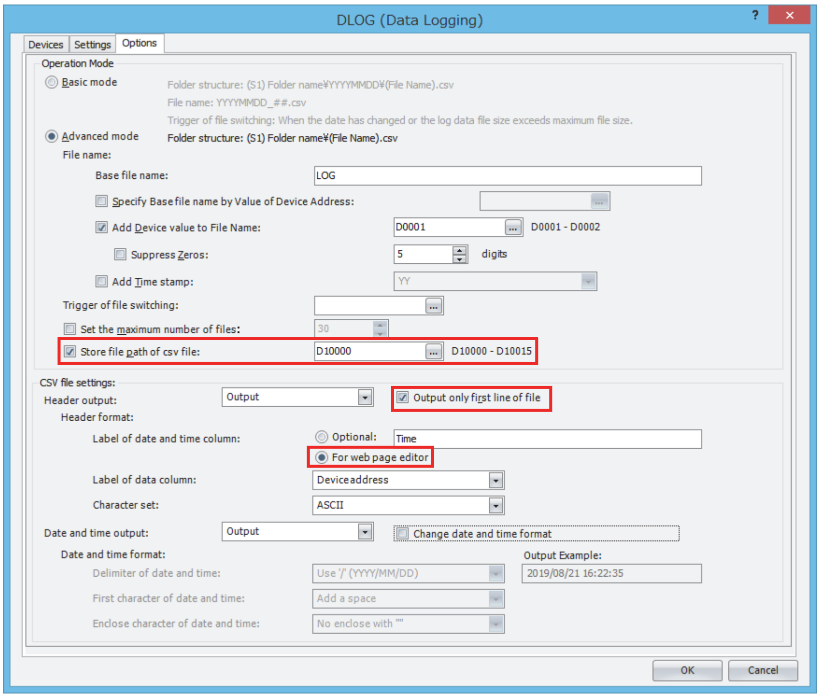




CSV file

#_date_time_	D0100	D0102	D0104	D0106
2019/7/3 15:18:01	0	0	0	0
2019/7/3 15:19:01	0	0	0	1
2019/7/3 15:20:01	0	0	0	0
2019/7/3 15:21:01	0	1	1	1
2019/7/3 15:22:01	0	1	1	0
2019/7/3 15:23:01	0	1	1	1
2019/7/3 15:24:01	0	2	0	0
2019/7/3 15:25:01	0	3	0	1
2019/7/3 15:26:01	0	4	0	0
2019/7/3 15:27:01	0	5	1	1
2019/7/3 15:30:15	0	6	1	0
2019/7/3 15:31:15	0	7	1	1

3. Select **Output only first line of file** in the **Header output**.
   Under **Header format**, select **For web page editor** for **Label of date and time column**.
   Select the **Store file path of csv file** check box, and specify the starting data register for storing the file path.



Store file path of csv file: If this check box is selected, the graph will be displayed simply by setting a device address on **Link Box** of the trend bar component.
If this check box is cleared, the graph will not be displayed when the path to the CSV file is not directly specified in the properties of the trend bar component.

Output only first line of file: If this check box is selected, this header is applied as the header on the X-axis of the trend bar component when the FC6A Series MICROSmart is changed from STOP to RUN.
If this check box is cleared, the graph will not be displayed on the trend bar component even when the FC6A Series MICROSmart is changed from STOP to RUN.

Header format:

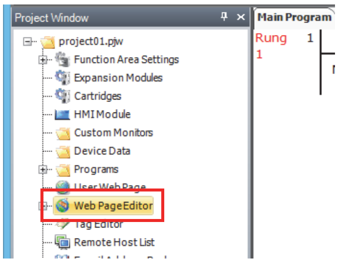
Label of date and time column: If **For web page editor** is selected, this header is applied as the the header on the X-axis of the trend bar component.
If **Optional** is selected, the graph will not be displayed on the trend bar component.
Header output: If set to output the header, this header is applied as the header on the X-axis of the trend bar component.
If set to not output the header, the graph will not be displayed on the trend bar component.

There are no **Options** settings required for the trend bar component except the settings described above. For details on the **Options** settings, see Chapter 25 "DLOG (Data Log)" in the "FC6A Series MICROSmart Ladder Programming Manual".

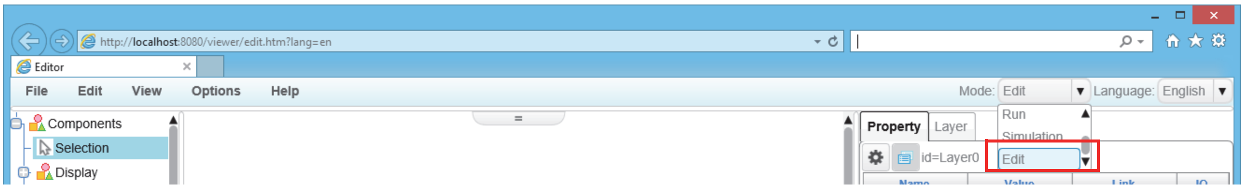
Notes:

- To perform log data integration with Web Page Editor and display a non-ASCII strings, select **Unicode (UTF-8) with BOM** or **Unicode (UTF-8) without BOM** for **Character set**.
  For **Unicode (UTF-8) with BOM**, the BOM values (0xEF 0xBB 0xBF) are output as the first 3 bytes of data.
  For **Unicode (UTF-8) without BOM**, if the file is opened in a specific application, the characters may be unreadable because the application does not open the file as **UTF-8** data.
- To use the table, list, and drop-down list components, the content to set under **Common Settings** on the **Options** tab will vary depending on the properties that can be used with each component. For the properties that can be used with each component, refer to the Web Page Editor help that is displayed by clicking **Help** > **Help** on the menu bar of Web Page Editor.

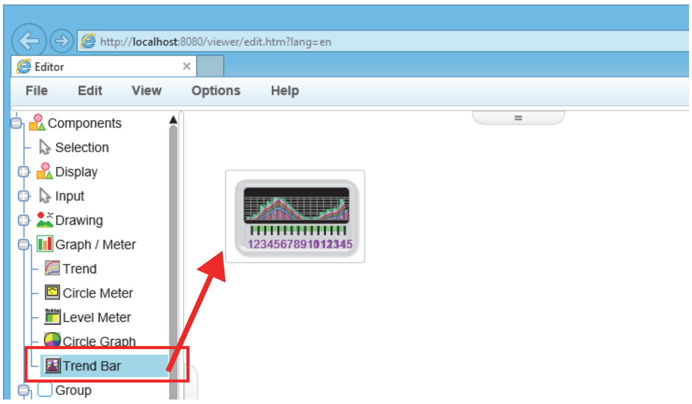
4. Double-click **Web Page Editor** in the Project window.
Web Page Editor starts.



5. For the Web Page Editor **Mode**, click **Edit**.



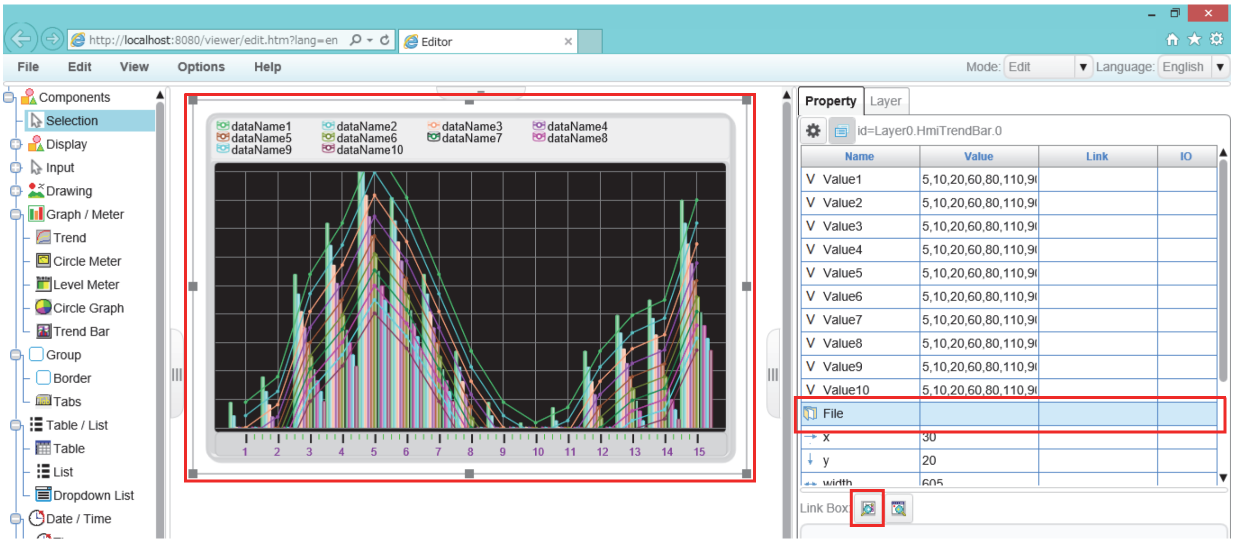
6. Place a trend bar on the work area and set the data register (D0100).
Select **Trend Bar** in the **Components** list, and drag and drop it on the work area.
The trend bar will be placed on the area with the predefined size.



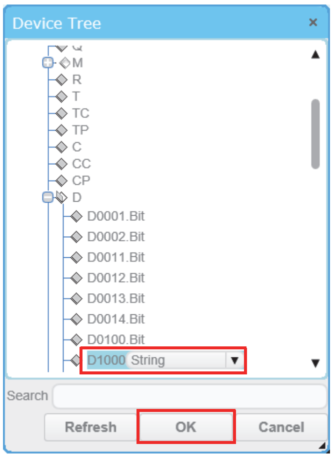
Notes:

- To place a trend bar on the area with the desired size, click **Trend Bar** in the **Components** list, and then drag the precision select mouse cursor on the work area.
- To use the table, list, and drop-down list components, select **Table**, **List**, or **Dropdown List**, and then drag and drop that component to the work area. For details on the table, list, and drop-down list components, refer to the Web Page Editor help that is displayed by clicking **Help** > **Help** on the menu bar.

7. Click the trend bar placed on the work area and on the **Property** tab, click **File**. Next, for **Link Box**, click [icon].
   The **Device Tree** dialog box is displayed.



8. Click the device address to set to the trend bar and click **OK**.

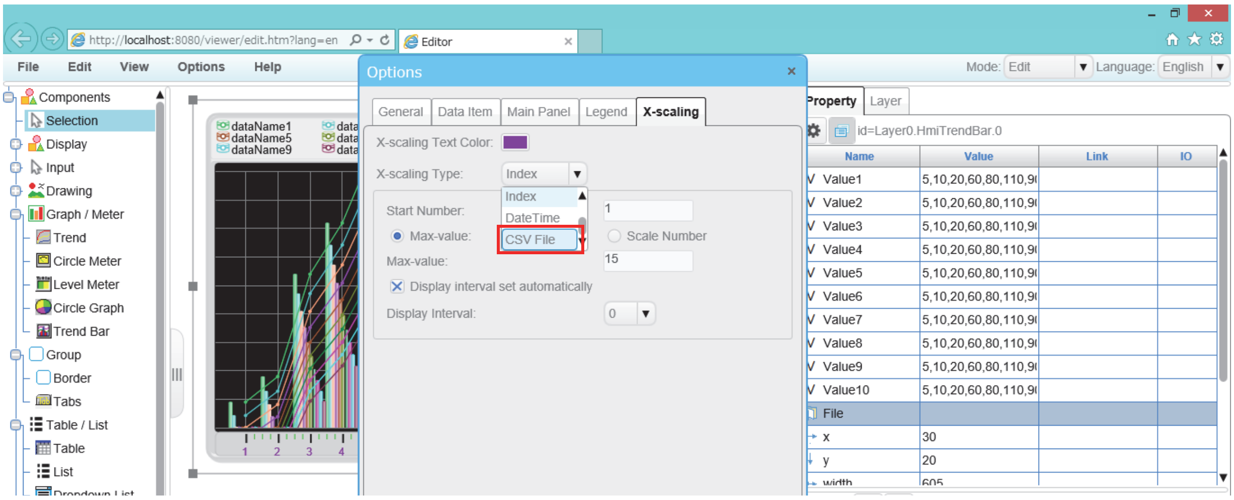


Notes:

- The devices that are used in WindLDR are displayed in the **Device Tree** dialog box.
- If the device address to set is not in the Device Tree, click **Cancel**, and then directly enter the device address in the variable link input area for **Link Box** on the **Property** tab.
- To specify the file path as the file property value, add "<D>\" to the front of the folder name.
  File path format:
  <D>\Folder Name\CSV File Name
  The DLOG instruction stores the path of the file being updated in the data register specified by **Store file path of csv file**.

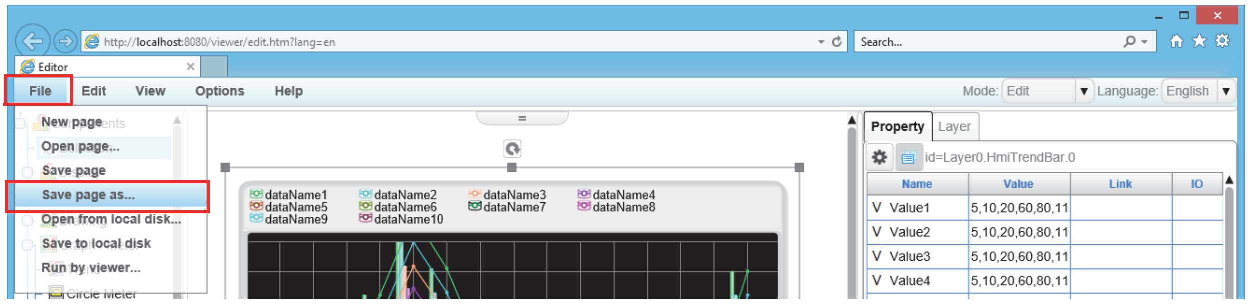
9. Double-click the trend bar that has been placed in the area.
   The trend bar **Options** dialog box is displayed.

10. On the **X-scaling** tab, under **X-scaling Type**, select **CSV File**.

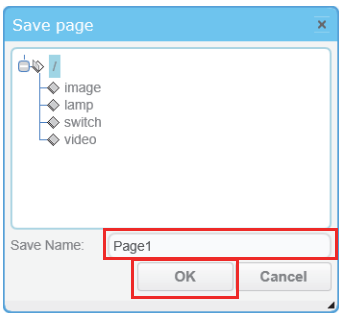


Note: In the **Options** dialog box, on the **General**, **Data Item**, **Main Panel**, and **Legend** tabs, the color, scale, data items, units, tool tips, legend, and other settings of components can be configured.

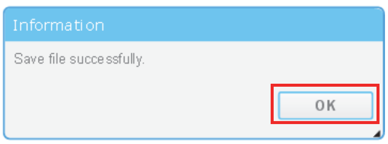
11. On the menu bar, click **File** > **Save page as**.
The **Save page** dialog box is displayed.



12. Enter the file name in **Save Name**, specify the folder to save to, and click **OK**.
A confirmation message is displayed.

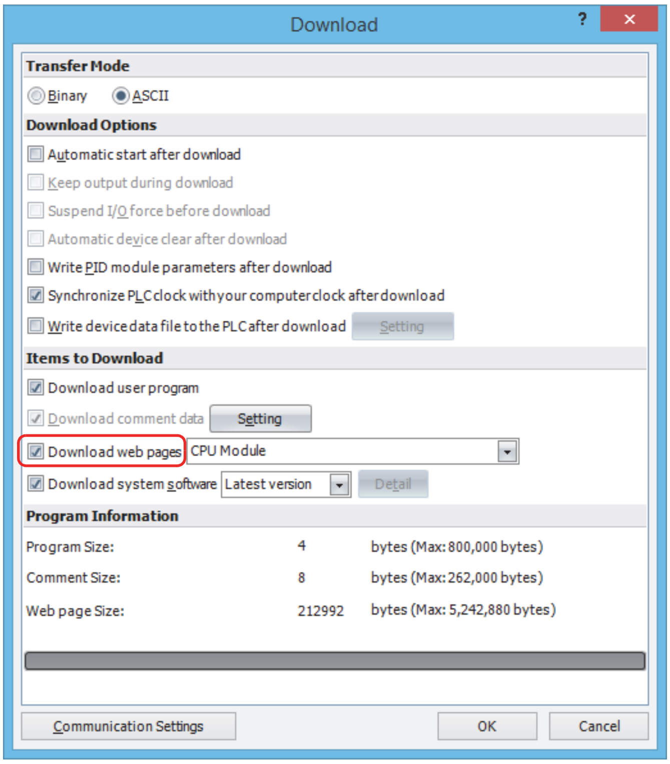


13. Click **OK**.



14. Download the user program to the Plus CPU module.
On the **Online** tab, in the **Transfer** group, click **Download**.
The **Download** dialog box is displayed.

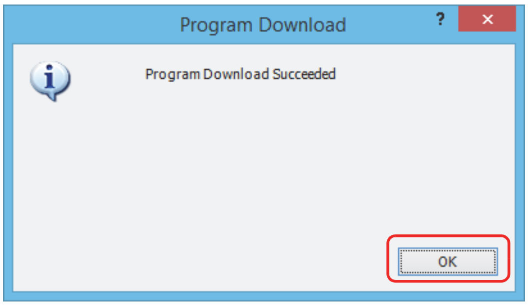
15. Select the **Download web pages** check box.



16. Select **CPU Module** or **CPU Module (SD Memory Card)** as the destination for saving the web pages.

Note: Web pages will not be downloaded when no SD memory card is inserted into the Plus CPU module but **CPU Module (SD Memory Card)** is selected.

17. Click **OK**.
The web pages will be downloaded to the selected save location.
When the following message appears, the download has completed successfully. Click **OK**.



18. When the DLOG (Data Log) instruction is executed, a CSV file is saved to the SD memory card. The saved CSV file can be viewed as a graph from the user web page.

Note: If the size of the CSV file is large, it may take some time until it is displayed on the graph on the web page.
Adjust the maximum size of the CSV file according to the conditions of the communication line.
For the configuration of the maximum log data file size, see Chapter 5 "Functions and Settings" in the "FC6A Series MICROSmart User's Manual".

Supported Web Browsers

The recommend web browsers that should be used to view the web pages downloaded to the FC6A Series MICROSmart are as follows.

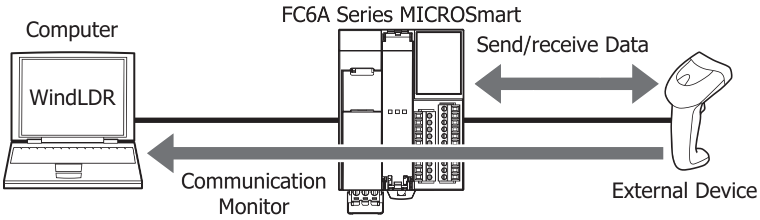
OS	Supported Browsers
Windows 10/8/7	Google Chrome 47 or later Mozilla Firefox 42 or later Microsoft Internet Explorer 11
Android	Google Chrome 47 or later
iOS	Safari 8 or later

14: COMMUNICATION MONITOR

This chapter describes the Communication Monitor.

Overview

The Communication Monitor is a function that monitors the communication data exchanged between the FC6A Series MICROSmart and an external device in the **Communication Monitor** dialog box of WindLDR. The communication protocols that can be monitored are user communication and Modbus communication. The communication monitor can be used with the Plus CPU module only.

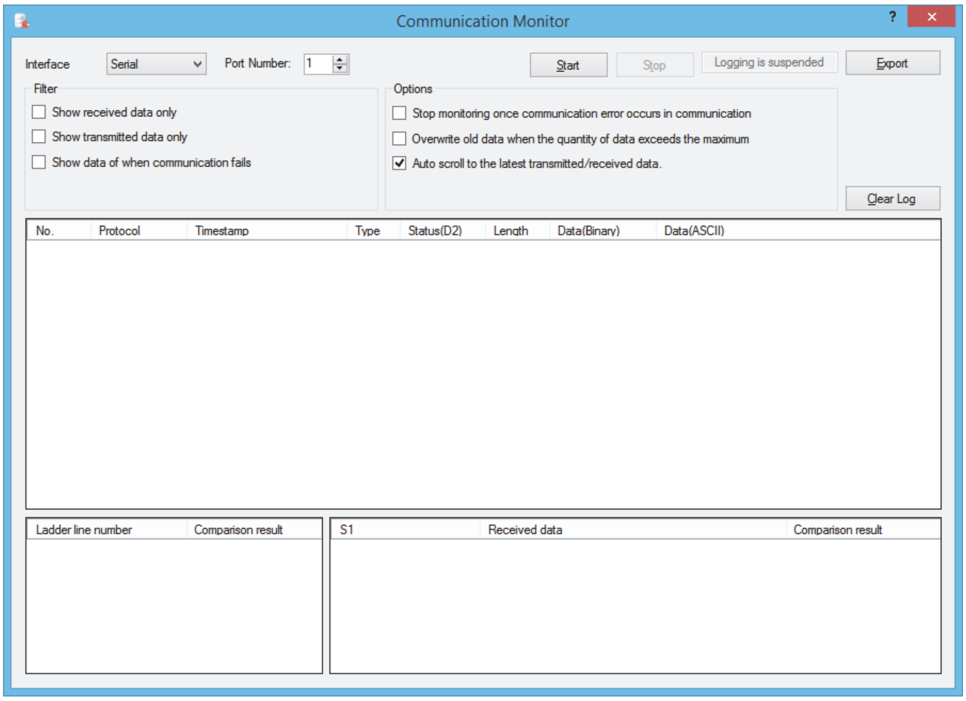


Starting and Stopping the Communication Monitor

This section describes how to start the Communication Monitor and how to start and stop monitoring of an external device.

● **Operation procedure**

1. On the **Online** tab, in the **Monitor** group, click **Communication Monitor**.



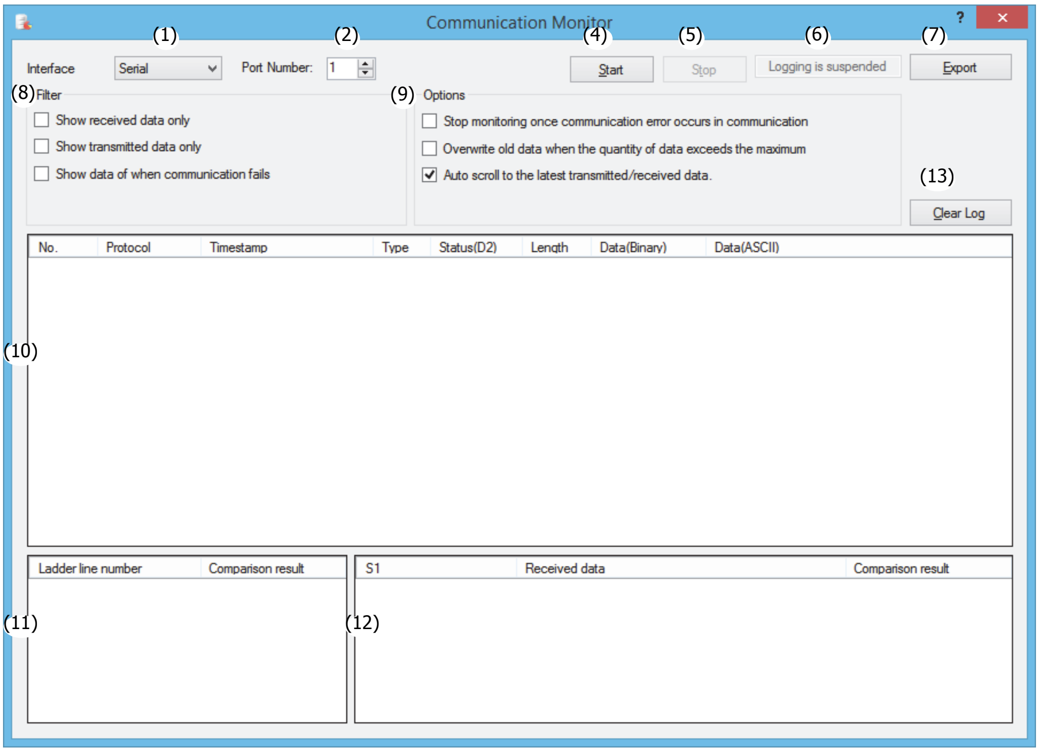
The external device that will be monitored is specified by the communication interface, port number, and connection number. The communication data that will be monitored is that which the FC6A Series MICROSmart saves to internal memory from when **Start** is clicked. Click **Stop** or close the **Communication Monitor** dialog box to stop saving the communication data to internal memory. For this reason, you cannot view send and receive data exchanged while the Communication Monitor is not monitoring.

Up to 65,535 items of data can be monitored. The monitor will automatically stop when the maximum amount of data has been reached. To continue monitoring even when the maximum amount of data has been reached, select the **Overwrite old data when the quantity of data exceeds the maximum** check box. Monitoring will continue and the data will be overwritten from the oldest data.

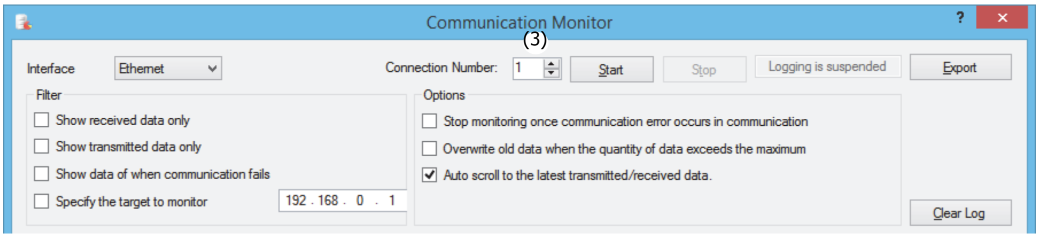
If the communication interval with the monitored external device is faster than the interval used to display and update send and receive data in the Communication Monitor in WindLDR, the FC6A Series MICROSmart may have insufficient internal memory to temporarily save data. In this case, saving data to internal memory will stop.
The display on the Communication Monitor after saving data to internal memory has stopped will change to the monitor stopped status after all send and receive data temporarily saved to internal memory has been displayed.

Communication Monitor

Monitoring Serial Communication



Monitoring Ethernet Communication



(1) Interface

Select the communication interface to monitor from **Serial** or **Ethernet**.

(2) Port Number

Set the port of the communication interface to monitor. This can be set only when Interface (1) is **Serial**.
Port: 1 to 33

(3) Connection Number

Sets the connection number to monitor.
This can be set only when Interface (1) is **Ethernet**.
Connection: 1 to 16

(4) Start button

Click this button to start the Communication Monitor. WindLDR will acquire communication data from the PLC at a regular interval.

(5) Stop button

Click this button to stop the Communication Monitor.
Accumulation the communication data at the PLC stops while the Communication Monitor is stopped.

(6) Status Indicator

Displays the status of the Communication Monitor.

(7) Export button

The communication data displayed in Send and Receive Data Display (10) can be exported.
Click this button, enter the file name in the **Save As** dialog box, and click **Save**.

(8) Filter

The items displayed in the Send and Receive Data Display (10) can be filtered. Select the check boxes for the items to display. Multiple filters can be selected.

When the **Specify Send/Receive Device** check box is selected, enter the IP address.

(9) Options

Select the relevant check boxes to operate the communication monitor with the following processing types:

Stop monitoring once communication error occurs in communication:
Stops monitoring when an error occurs.

Overwrite old data when the quantity of data exceeds the maximum:
When the communication data reaches the maximum amount of data (65,535 items), the monitor will continue by deleting old data instead of stopping the monitor.

Auto scroll to the latest transmitted/received data:
The latest send and receive data will be automatically displayed when send and receive data is added.

(10) Send and Receive Data Display

Displays the communication data exchanged between the FC6A Series MICROSmart and external device. The displayed content is as follows.

- Protocol
- Time stamps (Plus CPU module with system version 1.01 or later)
- Send/receive type
- IP address of target device to send/receive (Ethernet only)
- Status
- Data length
- Data (binary or ASCII)

Note: Time stamps are based on the calendar and clock data of the FC6A Series MICROSmart when the monitor is started, and they are implemented by adding the time difference from the start of the monitor that is appended to the send and receive data. The purpose of time stamps is to indicate the approximate communication interval of communication data. The time of the time stamps may not match the current time of the computer and it may not be adjusted for daylight savings time.

The calendar/clock display format of the computer is used as the display format of the time stamps.

(11) User Communication Instruction Selection

Displays the list of user communication instructions corresponding to the selected send/receive data in Send and Receive Data Display (10). This is not displayed when the protocol is Modbus.

Ladder line number : Displays the line number of the user communication instruction (RXD/ERXD/TXD/ETXD) in WindLDR. The relevant user communication instruction is determined by searching for the instruction in WindLDR based on the device address of D2 obtained from the PLC.

Comparison result : Displays the receive result of the user communication receive instructions. (Not displayed for send instructions.)

(12) Comparison Result Details

Displays details of the comparison result between the actual communication data and the communication format (S1) of the user communication instruction selected in User Communication Instruction Selection (11). In this way, you can check the difference between the format of the user communication receive instruction and the actual received data. However, this is not displayed when the protocol is Modbus.

S1 : Displays the format of the user communication receive instruction selected in User Communication Instruction Selection (11).
- Constant (Character)
- Constant (Hexadecimal)
- Data Register
- BCC
- Skip (receive data only)

Receive Data/Send Data : Displays the content of the communication data.

Comparison Result : Displays the comparison result.
Match: OK
Mismatch: NG

Note: The data received by the FC6A Series MICROSmart is verified against the format specified by the user communication instruction in WindLDR, and the comparison result with the format is associated with the actual communication data and then displayed. For this reason, the user program opened in WindLDR and the user program in the FC6A Series MICROSmart must be the same. If the user programs are different, a warning message will be displayed stating that the comparison results cannot be displayed when the **Communication Monitor** dialog box is displayed, and only Send and Receive Data Display (10) will be displayed.

(13) Clear Log button

Erases the displayed communication data.

15: BACNET/IP

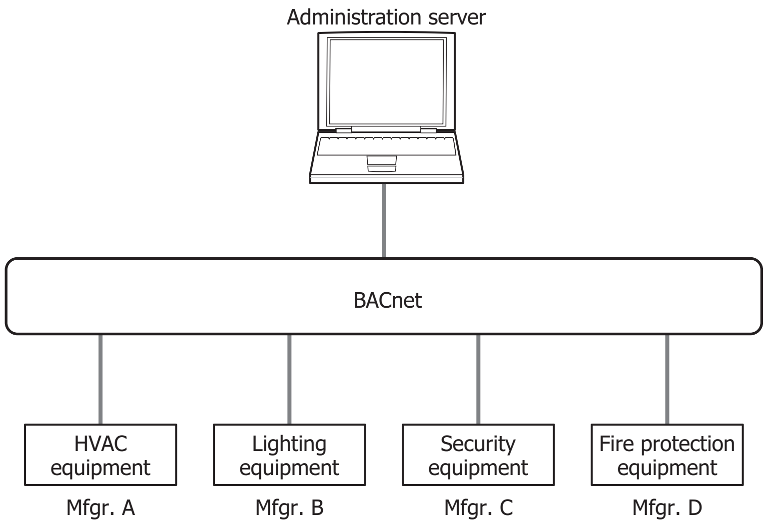
This chapter describes the BACnet communication function.

Overview

The Plus CPU module supports BACnet communication using the Internet protocol (IP), which is called BACnet/IP.

BACnet stands for "Building Automation and Control Networking Protocol". In building equipment, it is an open protocol that has been standardized for interoperability between systems built by different manufacturers, and a communication standard for building networks.

Previous air conditioning, lighting, security, and fire protection systems were connected using methods unique to the building equipment and systems manufacturers. However, conforming to BACnet allows equipment and systems to be connected and monitored using a common method.



Communication Ports Used in BACnet/IP

Supported models and communication ports are as follows.

Communication Port	All-in-One CPU Module			CAN J1939 All-in-One CPU Module	Plus CPU Module	
	16-I/O Type	24-I/O Type	40-I/O Type		Plus 16-I/O Type	Plus 32-I/O Type
Ethernet Port 1	—	—	—	—	Yes	Yes
Ethernet Port 2	—	—	—	—	—	—
HMI-Ethernet Port	—	—	—	—	—	—

BACnet/IP cannot be used with ports 1 to 33.

BACnet Specifications

Model Number		FC6A-D16R1CEE, FC6A-D16R4CEE FC6A-D16K1CEE, FC6A-D16K4CEE FC6A-D16P1CEE, FC6A-D16P4CEE	FC6A-D32K3CEE, FC6A-D32K4CEE FC6A-D32P3CEE, FC6A-D32P4CEE
Supported Port		Ethernet Port 1	
Applicable Standards		ANSI/ASHRAE135-2012	
Standard Specifications	**Protocol**	BACnet/IP	
	Profile	B-ASC	
	Object Type	Device Object, Analog Input Object, Analog Output Object, Analog Value Object, Binary Input Object, Binary Output Object, Binary Value Object	
	Number of Objects	256 maximum*1	
	BIBBs	DS-RP-B, DS-WP-B, DS-RPM-B, DS-WPM-B, DS-COV-B, DS-COVU-B, DM-DDB-B, DM-DOB-B, DM-DCC-B	
	BBMD	None-BBMD Device	
	Virtual Device	No	
	Foreign Device	Yes	
Subscribed COV Function	**Number of Requests That Can Be Accepted**	256 requests maximum	
Unsubscribed COV Function	**Transmission Unit**	Every object	
	Transmission Cycle	1 to 65,535 [ms]*2	
Foreign Device Function	**Registration Method**	Registration as needed by registration trigger device	
	Lifetime	0 to 65,535 [s]	
Device Binding Function		• Synchronization between properties and devices*3 • Data type conversion of Present _Value*4 • Coefficient conversion of Present _Value*4	

*1 Device Object is not included.
*2 The transmission cycle is set for all objects.
*3 The properties of objects created in internal memory are synchronized with specified devices.
*4 Supported objects are Analog Input Object, Analog Output Object, and Analog Value Object.

About BACnet

Applicable Standards

BACnet/IP in the Plus CPU module conforms to the following BACnet standards.

- ANSI/ASHRAE135-2012

Profile

BACnet/IP in the Plus CPU module supports the following profile.

- B-ASC

Objects

Information, such as input and output values handled by devices that support BACnet/IP, are managed in units called objects. Objects are separated into a number of different types, depend on the content of the object, and these are called object types. The object types supported by BACnet/IP in the Plus CPU module are as follows.

Object Type				Standard
Name		Abbreviation	Identifier	ANSI/ASHRAE135-2012
Basic input and output	Analog Input Object Type	AI	0	Yes
	Analog Output Object Type	AO	1	Yes
	Analog Value Object Type	AV	2	Yes
	Binary Input Object Type	BI	3	Yes
	Binary Output Object Type	BO	4	Yes
	Binary Value Object Type	BV	5	Yes
BACnet device attributes	Device Object Type	DV	8	Yes

The Plus CPU module can set a number of objects for each object type. This information can be written to and read from BACnet devices on the same BACnet/IP network.

For details on objects, see "Objects" on page 15-33.

Properties

Properties are the detailed information and attributes of each object. A portion of the properties held by objects can be allocated to Plus CPU module devices, such as data registers, and those devices can be written to and read from a ladder program.

For details on the properties of each object type, see "Objects" on page 15-33.

Services

Services are interfaces used to exchange information between BACnet devices. Services are classified into the client side that initiates services and the server side that executes services.

The Plus CPU module supports the following services.

Services	Initiate*1	Execute*2
ReadProperty	—	Yes
ReadPropertyMultiple	—	Yes
WriteProperty	—	Yes
WritePropertyMultiple	—	Yes
SubscribeCOV	—	Yes
ConfirmedCOVNotification	Yes	—
UnconfirmedCOVNotification	Yes	—
Who-Is	—	Yes
I-Am	Yes	—
Who-Has	—	Yes
I-Have	Yes	—
DeviceCommunicationControl	—	Yes

*1 The Plus CPU module initiates services for other BACnet devices.

*2 The Plus CPU module executes services that are initiated from other BACnet devices.

Note: For details on services, refer to the ANSI/ASHRAE 135-2012 (ISSN 1041-2336) standard or "BACnet Building Automation Data Communication Protocol", a book published by the Institute of Electrical Installation Engineers of Japan.

BIBB

BACnet interoperability building blocks (BIBB) are groupings of multiple services for each function that will be implemented. BIBBs are classified into the client side that uses the function and the server side that provides the function. '-A' is appended to the end of the client-side BIBBs, and '-B' is appended to the end of the server-side BIBBs. BACnet devices define the functions they support using BIBBs.

BACnet/IP in the Plus CPU module supports the following BIBBs.

BIBB Category	BIBB		Services
Data Sharing	DS-RP-B	Data Sharing Read Property B	ReadProperty
	DS-WP-B	Data Sharing Write Property B	ReadPropertyMultiple
	DS-RPM-B	Data Sharing Read Property Multiple B	WriteProperty
	DS-WPM-B	Data Sharing Write Property Multiple B	WritePropertyMultiple
	DS-COV-B	Data Sharing COV B	SubscribeCOV ConfirmedCOVNotification UnconfirmedCOVNotification
	DS-COVU-B	Data Sharing COV Unsubscribed B	UnconfirmedCOVNotification
Device & Network Management	DM-DDB-B	Device Management Dynamic Device Binding B (Who-Is, I-Am)	Who-Is I-Am
	DM-DOB-B	Device Management Dynamic Object Binding B (Who-Has, I-Have)	Who-Has I-Have
	DM-DCC-B	Device Management Device Communication Control B	DeviceCommunicationControl

Note: For details on BIBBs, refer to "BACnet Building Automation Data Communication Protocol".

Function

The Plus CPU module provides the following functions as a single BACnet device.

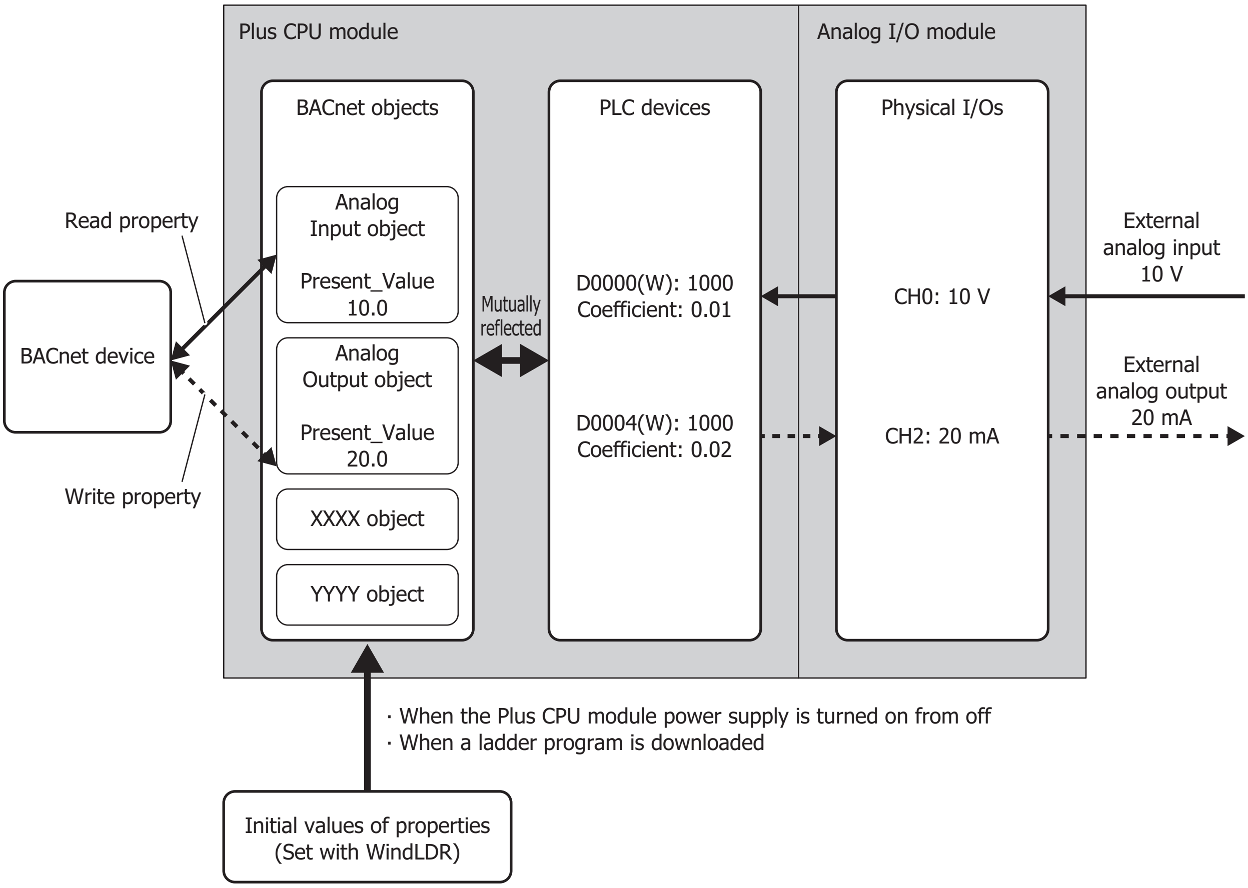
- Object and Device Binding function
- Read Property function
- Write Property function
- Subscribed COV function
- Unsubscribed COV function
- Foreign Device function

Object and Device Binding Function

A portion of the properties held by objects created in the Plus CPU module can be allocated to devices, such as data registers, and those devices can be written to and read from a ladder program.

For example, the analog input value of the Plus CPU modulecan be stored in a data register (D0000) allocated as Present_Value of the Analog Input object, and that analog input value can be read by BACnet devices. The analog output value of the Plus CPU module can also be changed from BACnet devices by allocating the data register (D0004) that stores the analog output value as Present_Value of the Analog Output object.

The types and numbers of objects can be freely set. The initial values of properties are also set with WindLDR.



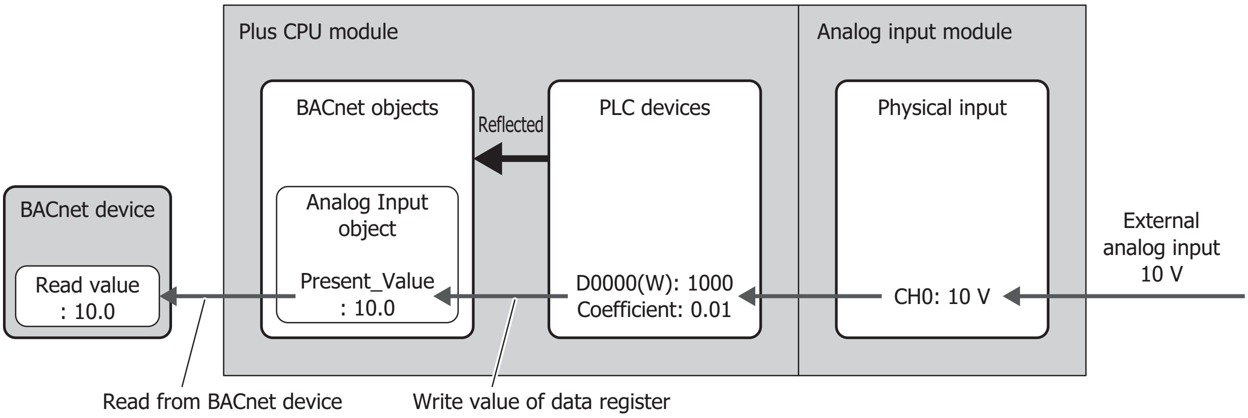
Notes:

- The processing to mutually reflect properties and devices is performed with no relationship to the execution cycle of the ladder program. Because devices allocated to objects will be written and read during execution of the ladder program, create the ladder program so that no problems will occur when devices are referenced or refreshed.
- If a device changes at an interval which is shorter than the cycle at which changes are reflected to the property of an object, those changes may not be reflected to the property. To reflect those changes to the property, keep the value of the device for 1 second.
  For example, the changes may not be reflected to the property when M0000, which has been allocated to Present_Value of a Binary Input object, is turned on for only 10 ms. To reflect those changes to the property, ensure that M0000 stays on for 1 second.

Read Property Function

In the Read Property function, the Plus CPU module returns the value of a property when there is a read property request from a BACnet device. This function is implemented using DS-RP-B (Data Sharing Read Property B) and DS-RPM-B (Data Sharing Read Property Multiple B).

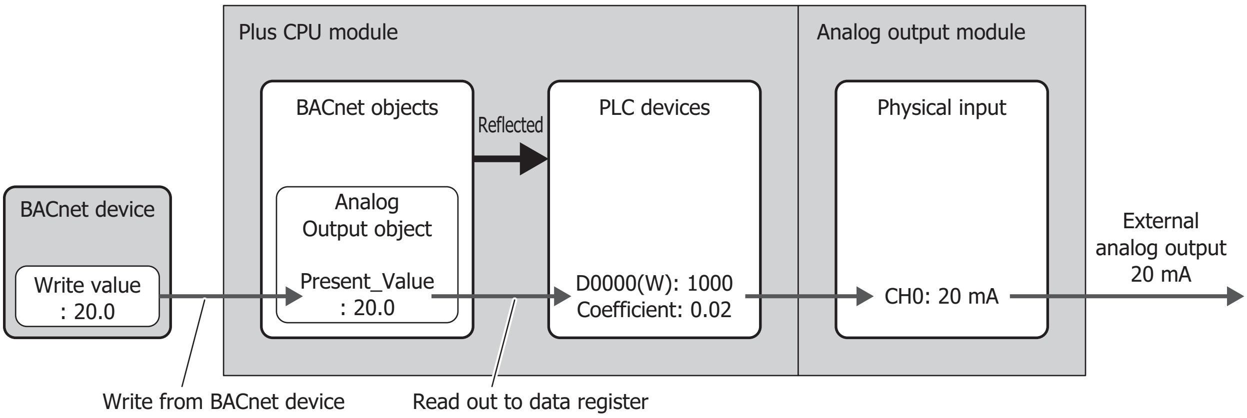
The below diagram shows a BACnet device reading the analog input value of the Plus CPU module that has been associated with the Analog Input object. The BACnet device is reading the analog input value of the Plus CPU module.



Write Property Function

In the Write Property function, the Plus CPU module writes a value to the property of an object when there is a write property request from a BACnet device. This function is implemented using DS-WP-B (Data Sharing Write Property B) and DS-WPM-B (Data Sharing Read Property Multiple B).

The below diagram shows a BACnet device changing the analog output value of the Plus CPU module that has been associated with the Analog Output object.



Subscribed COV (COV) function

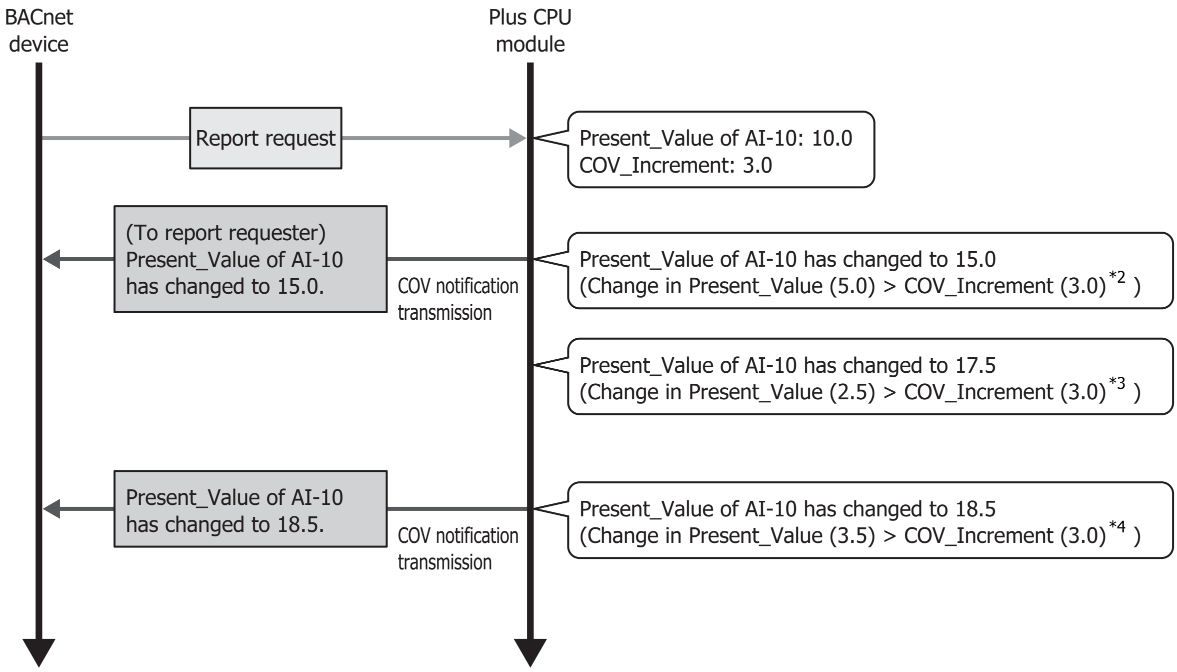
In the COV (Change Of Value) function, the Plus CPU module monitors an object for which a report request has been received from a BACnet device. Then when Present_Value or Status_Flags has changed, the Plus CPU module notifies the BACnet device of the change.

The properties of objects that correspond to the COV function and the timing of when to provide notification of changes are given as follows.

Object	Properties	Timing to Provide Notification of Change
Analog Input Analog Output Analog Value	Present_Value Status_Flags	Notification of change is provided when (1) or (2) as follows. (1) When Present_Value has changed to a value greater than or equal to the value set for COV_Increment (starting point is from when the previous COV notification was sent)*1 (2) When any of the bits in Status_Flags have changed
Binary Input Binary Output Binary Value	Present_Value Status_Flags	Notification of change is provided when (1) or (2) as follows. (1) When Present_Value has changed (2) When any of the bits in Status_Flags have changed

*1 For example, the following diagram shows when a report request is received from a BACnet device and AI-10 (Analog Input object, instance number 10) has the status in the following table.

AI-10 Properties	Value
Present_Value	10.0
COV_Increment	3.0



*2 This change corresponds to timing to provide notification of change (1).
Change in Present_Value = Present value - Value when report request was received = 15.0 - 10.0 = 5.0
Because COV_Increment = 3.0, change in present value ≥ COV_Increment has been satisfied.

*3 Change in Present_Value = Present value - Value when previous COV notification was transmitted = 17.5 - 15.0 = 2.5
Change in Present_Value < COV_Increment, and the condition for timing to provide notification of change (1) is not satisfied.
Therefore, COV notification is not sent.

*4 This change corresponds to timing to provide notification of change (1).
Change in Present_Value = Present value - Value when previous COV notification was transmitted = 18.5 - 15.0 = 3.5
Because COV_Increment = 3.0, Change in Present Value ≥ COV_Increment has been satisfied.

Notes:
- Notification may not be provided for changes in values that are faster than the synchronization cycle of objects and devices.
- The transmit queue for issuing BACnet services is 30 services. Services may not be transmitted if there are simultaneous transmit requests that exceed the transmit queue limit.

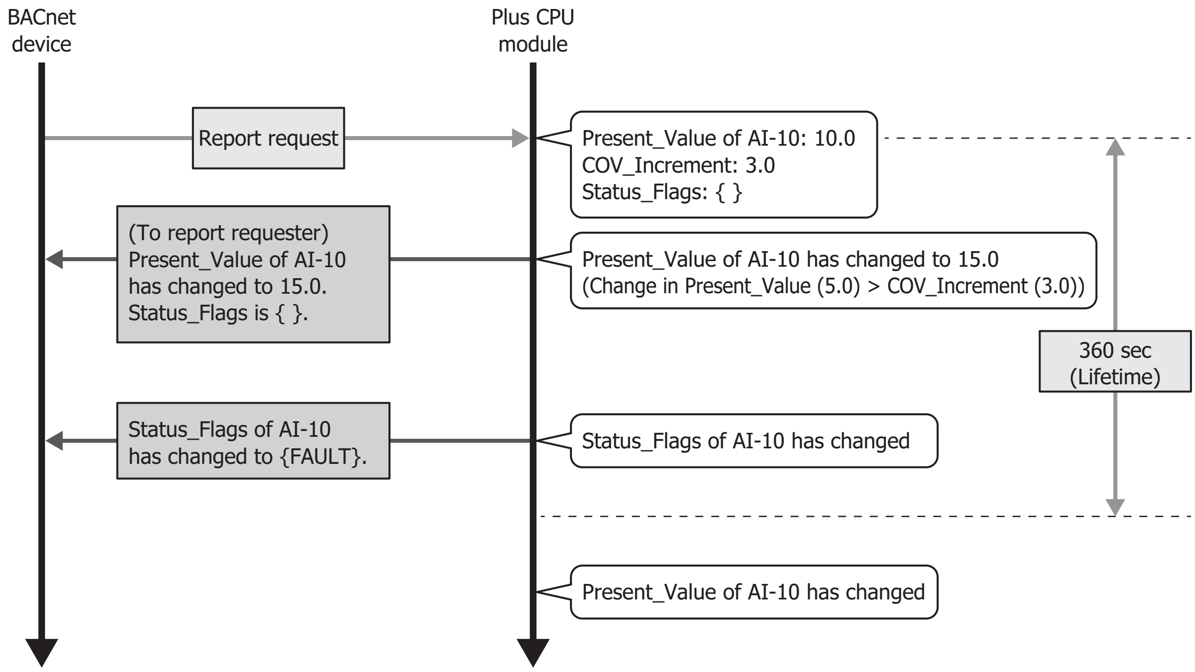
The Plus CPU module implements the COV function by sending the Confirmed COV Notification service/Unconfirmed COV Notification service to the report requesting BACnet device according to the parameters contained in the report request when that report request (Subscribe COV service) was transmitted from the BACnet device.

The key parameters contained in the report request are as follows.

Key Parameters	Description
Monitored Object Identifier	Type and ID of object for which to enable the COV function.
Issue Confirmed Notifications	Selects whether or not to confirm messages that will be sent from the Plus CPU module to the BACnet device. • With confirmation (ConfirmedCOVNotification) • Without confirmation (UnconfirmedCOVNotification)
Lifetime	The time to enable the COV function in 1 sec increments. If 0 or omitted, the COV function is enabled with no time limit.

The following diagram shows when the Plus CPU module has received a report request transmitted from a BACnet device like that in the following table.

Key Report Request Parameters	Description
Monitored Object Identifier	Analog Input object (instant number: 10)
Issue Confirmed Notifications	With confirmation (ConfirmedCOVNotification)
Lifetime	360 sec



Notes:
- The maximum number of COVs that can be registered is 256.
- For details on the parameters in a response request, refer to "BACnet Building Automation Data Communication Protocol".

Unsubscribed COV (COVU) function

In the COVU (Change Of Value Unsubscribed) function, the Plus CPU module spontaneously provides notification that Present_Value or Status_Flags of a specific object has changed to all BACnet devices connected to the same network.

The properties of objects that correspond to the COVU function and the timing of when to provide notification of changes are given as follows.

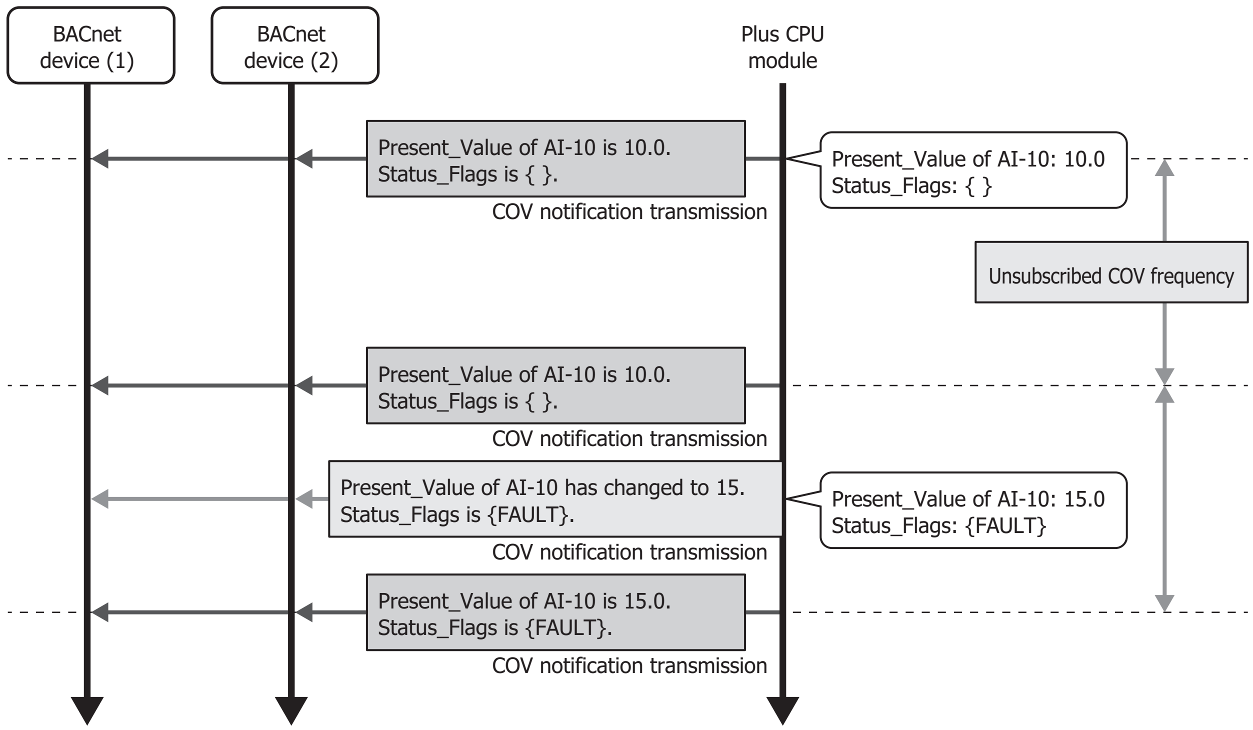
Object	Properties	Timing to Provide Notification of Change
Analog Input Analog Output Analog Value	Present_Value Status_Flags	Notification of the status is provided at the interval of the Unsubscribed COV frequency*1. Notification of change is also provided when (1) or (2) as follows. (1) When Present_Value has changed to a value greater than or equal to the value set for COV_Increment (starting point is from when the previous COV notification was sent) (2) When any of the bits in Status_Flags have changed
Binary Input Binary Output Binary Value	Present_Value Status_Flags	Notification of change is provided when (1) or (2) as follows. (1) When Present_Value has changed (2) When any of the bits in Status_Flags have changed

*1 See "Basic Settings" on page 15-14.

The Plus CPU module implements the COVU function by sending the Unconfirmed COV Notification service to BACnet devices.

For example, when BACnet/IP is started with AI-10 (Analog Input object, instant number 10) having the status in the following table, notification is provided on the status of the properties at the set cycle (Unsubscribed COV frequency).

AI-10 Properties	Value
Present_Value	10.0
COV_Increment	3.0
Status_Flags	{ }

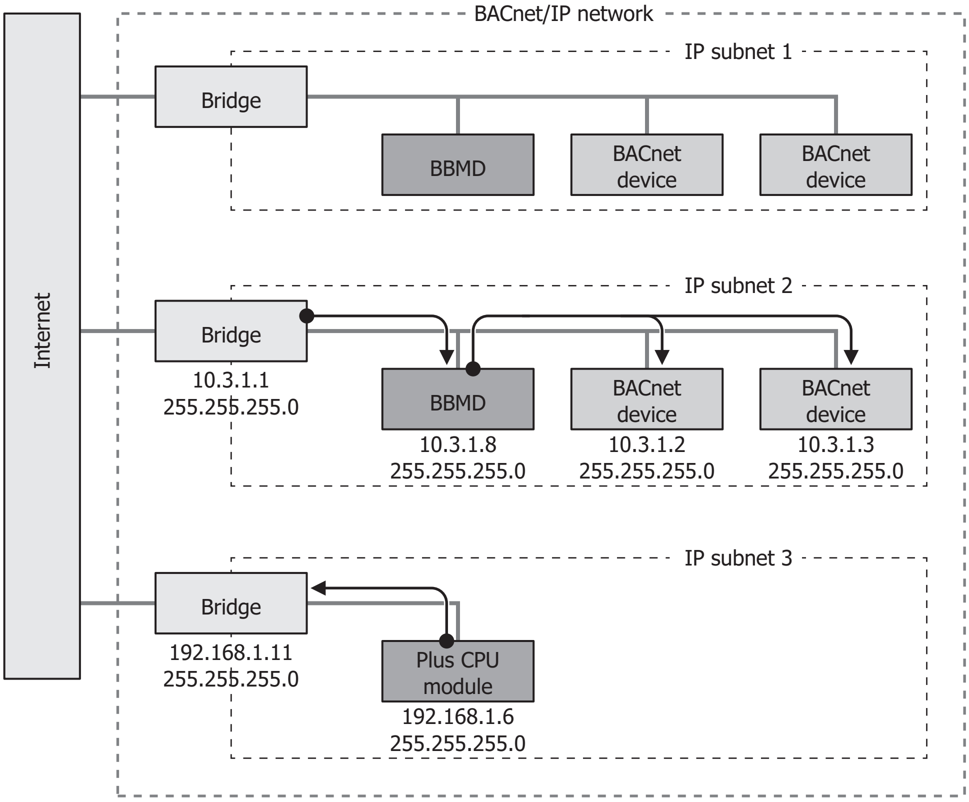


Notes:

- When using the COVU function, the values of properties can be regularly sent to BACnet devices, regardless of any change in the value of the properties.
- The COVU function can be set for each object.
- Only one common Unsubscribed COV Frequency can be set. Different frequencies cannot be set for each object.
- If the Unsubscribed COV Frequency is set to 0, regular notifications will not be provided. Notifications will be provided only when there is a change in the target property.
- When the Unsubscribed COV frequency is changed by a device, the change will be applied after the service is next initiated.
- Notification may not be provided for changes in values that are faster than the synchronization cycle of objects and devices.
- The transmit queue for issuing BACnet services is 30 services. Services may not be transmitted if there are simultaneous transmit requests that exceed the transmit queue limit.

Foreign device function

When the BACnet/IP network is built with multiple IP subnets, install one BBMD (BACnet Broadcast Management Device) on each IP subnet. A BBMD is a device that transfers broadcast communications of BACnet devices to different IP subnets. BACnet devices perform broadcast communications with BACnet devices on different IP subnets via BBMDs. The Foreign Device function is used for performing broadcast communications with BACnet devices on different IP subnets even when there is no BBMD on the IP subnet of the Plus CPU module. Broadcast communications can be performed with a BACnet device on the BACnet/IP network by registering the Plus CPU module as a foreign device for a specified BBMD.



BACnet/IP Operation

Special Devices Used in BACnet/IP

This table shows the special data registers and special internal relays used in BACnet/IP.

Special Data Register	Description	
D8782	BACnet Operation Status	This special data register stores the operation status of BACnet communication. 0: Stopped 1: Ready 2: Operating 3: Stopped by error
D8783	BACnet Error Information	This special data register stores error information that has occurred in BACnet communication. Information for the error that last occurred is stored. 0: Normal 1: Invalid device ID 2: Invalid IP address 3: Invalid BBMD IP address 4: BBDM registration failure

Special Internal Relay	Description	
M8450	BACnet Communication Bit	The special internal relay permits or prohibits BACnet communication. OFF: BACnet communication prohibited ON: BACnet communication permitted

Note: To resume BACnet communication when BACnet Operation Status (D8782) is stopped by an error, first turn off M8450 and wait for BACnet Operation Status (D8782) to become stopped, and then turn on M8450.

Operations during RUN and STOP

BACnet/IP is enabled and disabled according to BACnet Communication Bit (M8450), regardless of the RUN/STOP status.

Properties with allocated devices work by referencing those devices.

When BACnet/IP is enabled and properties or the values of devices that are associated with properties are changed, even during STOP, they operate with the values after the change.

RUN/STOP Status	BACnet/IP Enabled/Disabled	Properties with Allocated Devices and the Values of Those Devices
STOP	Enabled	Bound
	Disabled	Not bound

Note: Output during STOP is performed according to the status of M8025 (Maintain Outputs While CPU Stopped), regardless of whether or not BACnet/IP is enabled or disabled. For details on M8025 (Maintain Outputs While CPU Stopped), see "M8025: Maintain Outputs While Stopped" on page 2-10.

Operation During STOP→RUN

Properties with allocated devices work by referencing those devices.

When the power supply is turned on and off or the Plus CPU module is switched between STOP and RUN and the values of devices associated with properties are changed, such as due to changing device values with the ladder program, the devices operate with the values after the change. Use caution because the Plus CPU module may operate unexpectedly for this reason.

For example, when M0000 is allocated to Out_Of_Service, the value of Out_Of_Service may differ by the memory backup setting of the device.

(1) When internal relays are set to "Clear All" by STOP→RUN

RUN/STOP Status	BACnet/IP Enabled/ Disabled	Out_Of_Service	M000
STOP	Enabled	TRUE	ON
↓	↓	↓	↓
RUN	Enabled	**FALSE**	**OFF**

At the start of RUN, Out_Of_Service=FALSE because the status of internal relays is cleared.

(2) When internal relays are set to "Keep All" by STOP→RUN

RUN/STOP Status	BACnet/IP Enabled/ Disabled	Out_Of_Service	M000
STOP	Enabled	TRUE	ON
↓	↓	↓	↓
RUN	Enabled	**TRUE**	**ON**

Out_Of_Service=TRUE because the status of internal relays is kept.

Notes:

- Set devices that are allocated to properties so that their statuses are kept at STOP→RUN.
- The default settings of memory backup for internal relays and data registers are as follows.

Device	Default Setting
Internal relay	Clear All
Data register	Keep All

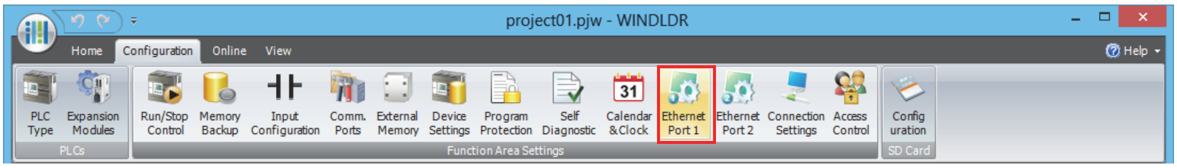
For the memory backup settings of devices, see Chapter 5 "Functions and Settings" in the "FC6A Series MICROSmart User’s Manual".

BACnet/IP Settings

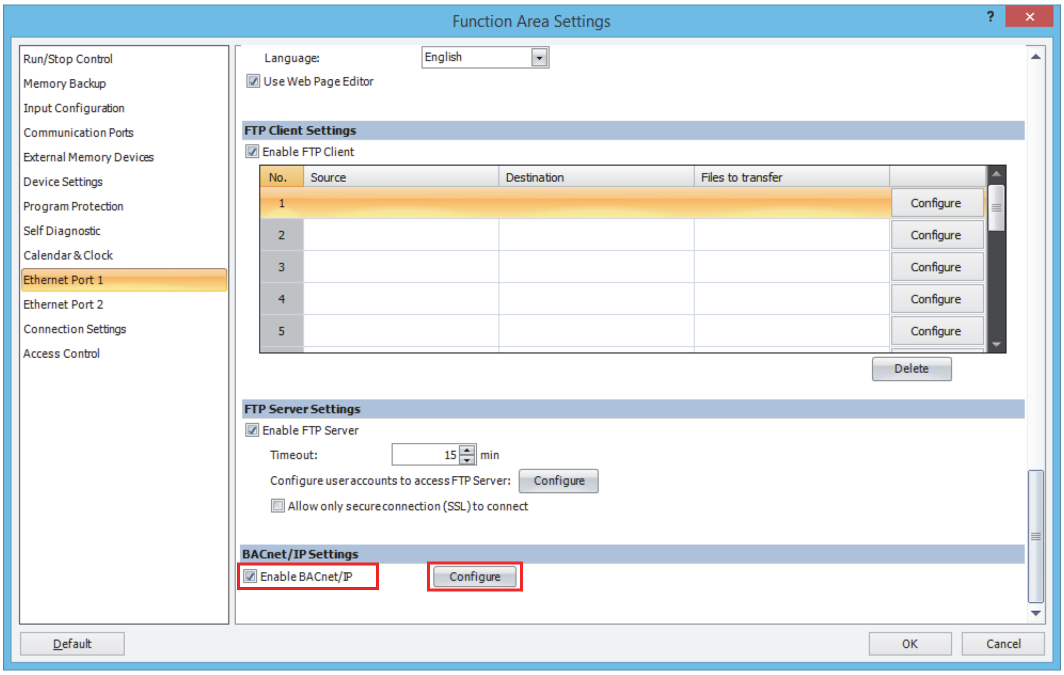
This section describes operation procedures and details about items for performing communication using BACnet/IP.

● Operation procedure

1. On the **Configuration** tab, in the **Function Area Settings** group, click **Ethernet Port 1**.
The **Function Area Settings** dialog box is displayed.



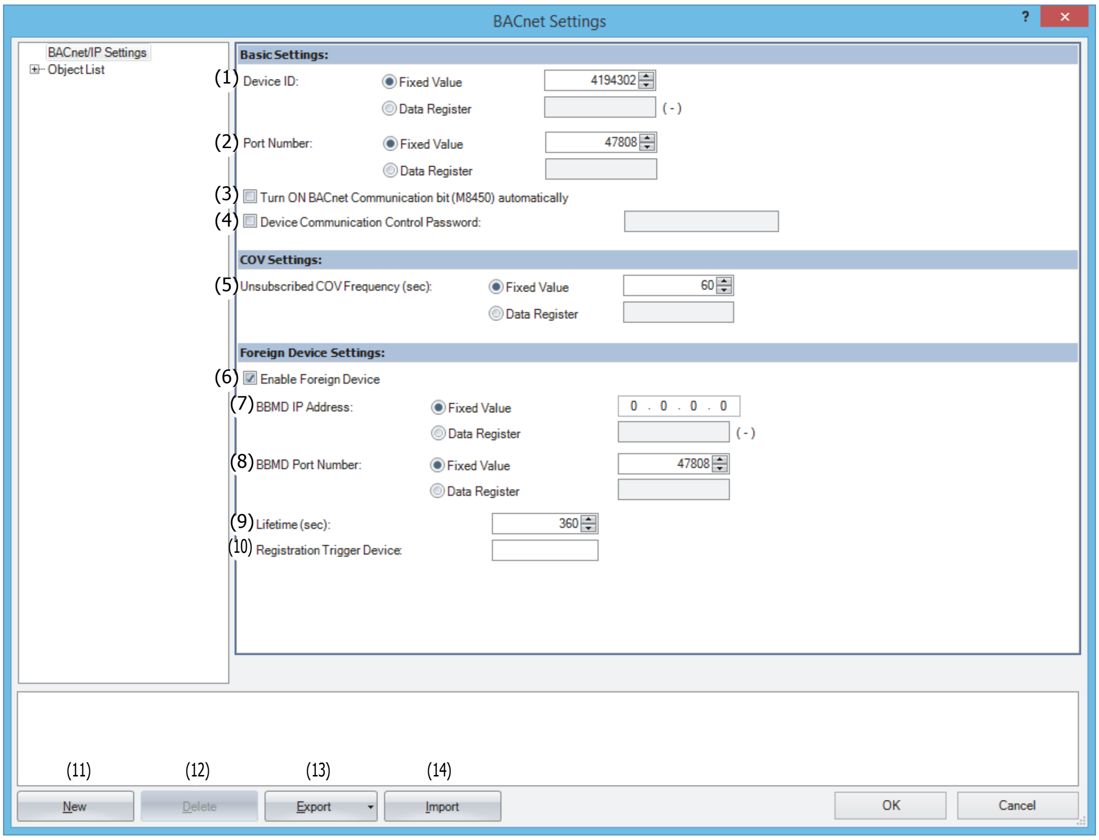
2. Under BACnet/IP settings, select the **Enable BACnet/IP** check box and click **Configure**.
The **BACnet Settings** dialog box is displayed.



Basic Settings

This section describes the basic settings for using BACnet/IP.

This screenshot shows an example of when **BACnet/IP Settings** was selected on the **BACnet Settings** dialog box.



Basic Settings

(1) Device ID

The assigned ID used to identify BACnet devices on the BACnet/IP network.

The Plus CPU module device ID can be set from the following two types.

Setting Method	Description
Fixed Value	Sets a fixed value for the device ID. Set the fixed value between 1 and 4,194,302. The default value is 4,194,302.
Data Register	Sets the device ID with data registers. Starting from the specified data register, 2 continuous words of data registers are used. Specify the first data register so that the device range is not exceeded. Set the device ID between 1 and 4,194,302.

(2) Port Number

Sets the port number for performing BACnet/IP communication.

The port number can be set from the following two types.

Setting Method	Description
Fixed Value	Sets a fixed value for the port number. Set the fixed value between 0 and 65,535. The default value is 47,808 (BAC0h).
Data Register	Sets the port number with a data register. Set the port number between 0 and 65,535.

(3) Turn ON BACnet Communication bit (M8450) automatically

Sets whether or not to turn on BACnet Communication Bit (M8450) in the END processing of the first scan when starting operation (RUN). During Online Edit, BACnet Communication Bit (M8450) is turned on in the END processing of the first scan in which downloading the user program has completed.

The following table shows the statuses.

M8450 Status	RUN/STOP Status	BACnet/IP	Present_Value*1 and Device Value	Overridden Flag in Status_Flags*2
ON	STOP	Enabled	Bound	TRUE
	RUN			FALSE
OFF	STOP	Disabled	Not bound	—
	RUN			

*1 For Present_Value, see "Objects" on page 15-33 and "Present_Value" on page 15-41.
*2 For Status_Flags, see "Objects" on page 15-33 and "Status_Flags" on page 15-54.

Notes:

BACnet Communication Bit (M8450) is turned off in the following cases.

- After the user program has been downloaded, regardless STOP or RUN status
- When the Plus CPU module power is turned off

(4) Device Communication Control Password

Sets the password to request when the Device Communication Control service is received from a BACnet device. Up to 20 single-byte alphanumeric characters can be set. When the Device Communication Control service is received, the Plus CPU module does not initiate a service or provide a response for the specified period of time.

COV Settings

(5) Unsubscribed COV Frequency (sec)

Sets the cycle to regularly provide notifications of property values with the COVU function in 1 sec increments.

The range that can be set is 0 to 65,535. If 0 is set, the COVU function is stopped, and notification is provided only when there is a change in the target property.

Setting Method	Description
Fixed Value	Sets a fixed value for the frequency. The default value is 60 sec.
Data Register	Sets the frequency with a data register. The frequency can be changed during RUN.

Notes:

- Only one Unsubscribed COV frequency can be set for the Plus CPU module, and different frequencies cannot be set per object.
- Whether or not to use the COVU function can be selected per object. The COVU function can also be enabled and disabled by specifying a data register and changing its value.
- When changing the Unsubscribed COV frequency by setting a data register for the Unsubscribed COV frequency and changing the value of the data register, the Unsubscribed COV frequency after it has been changed will be applied after the service is next initiated.

Foreign Device Settings

(6) Enable Foreign Device

Sets whether or not to use the Foreign Device function.

(7) BBMD IP Address

(8) BBMD Port Number

To join a BACnet/IP network when there is no BBMD on the IP subnet of the Plus CPU module, set the IP address and port number of the BBMD on the other IP subnet. The Plus CPU module registers itself as a Foreign Device on the set BBMD.

The IP address of the BBMD can be set from the following two types.

Setting Method	Description
Fixed Value	Sets a fixed value for the IP address of the BBMD.
Data Register	Sets the IP address of the BBMD with data registers. Starting from the specified data register, 4 continuous words of data registers are used. Specify the first data register so that the device range is not exceeded. For example, to set the IP address of the BBMD as 192.168.2.5, write the following values to the data registers. Starting data register+0 = 192 Starting data register+1 = 168 Starting data register+2 = 2 Starting data register+3 = 5

The port number of the BBMD can be set from the following two types.

Setting Method	Description
Fixed Value	Sets a fixed value for the port number of the BBMD. Set the fixed value between 0 and 65,535. The default value is 47,808 (BAC0h).
Data Register	Sets the port number of the BBMD with a data register. Set the port number of the BBMD between 0 and 65,535.

(9) Lifetime (sec)

Sets the interval of time to register the Plus CPU module with the BBMD as a Foreign Device in 1 sec increments.

When (Lifetime + 30) sec has elapsed from registration, the registered content will be deleted from the BBMD.

The range that can be set is 0 to 65,535. The default value is 360 sec.

Setting Method	Description
Fixed Value	Sets a fixed value for the lifetime. The default value is 360 sec.

(10) Registration Trigger Device

The device used to register the Plus CPU module with the BBMD set with (7) above as a Foreign Device. The Plus CPU module is registered with the BBMD when the registration trigger device is turned on from off. To continuously register the Plus CPU module with the BBMD, use the registration trigger device to re-register the Plus CPU module before (Lifetime + 30) sec elapses.

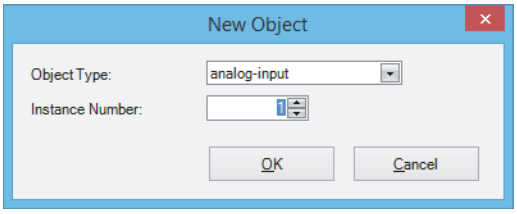
Set an internal relay for the registration trigger device.

Adding and Deleting Objects

(11) New button

Adds a new object.

Click **New** to display the **New Object** dialog box. Use this dialog box to set the object type and instance number. Set the instance number between 0 and 4,194,302, and ensure that objects of the same object type do not have the same instance number.



(12) Delete button

Deletes the object under the node selected in the object list.

(13) Export button

Exports the settings of the following objects as a CSV file.

Analog Input Object, Analog Output Object, Analog Value Object, Binary Input Object, Binary Output Object, Binary Value Object

(14) Import button

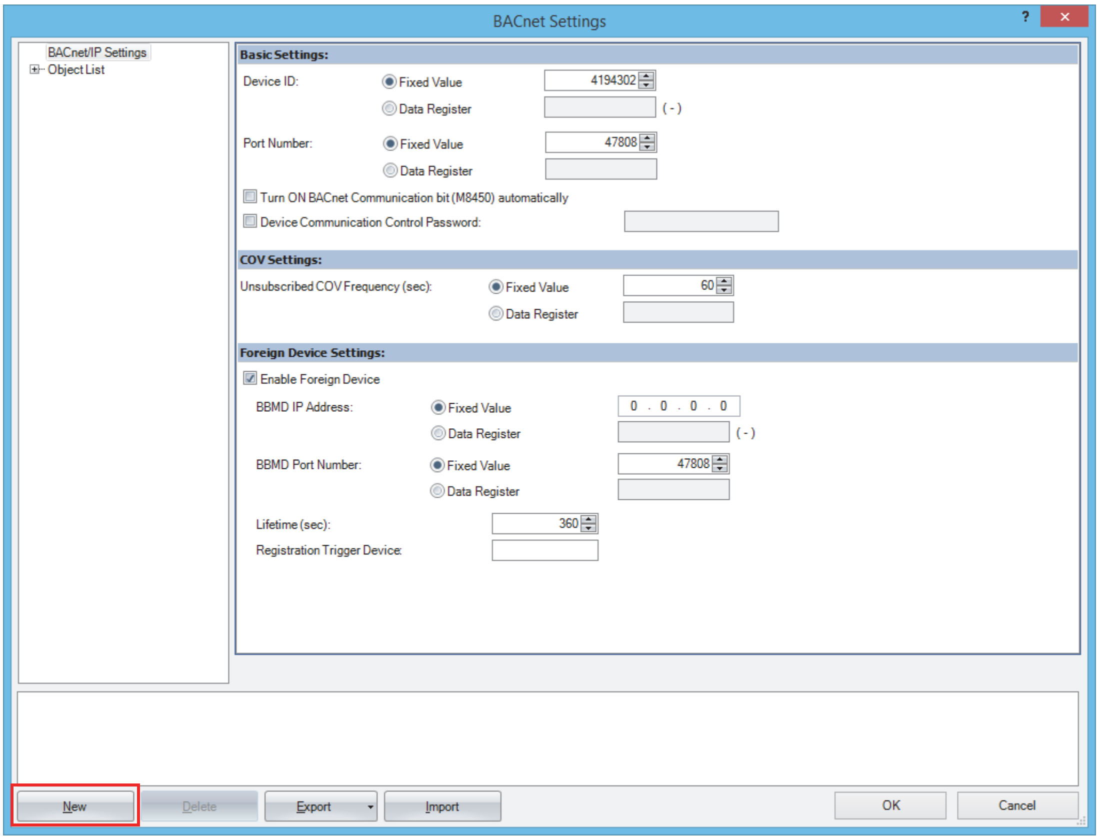
Imports a CSV file that was export and automatically creates objects.

Objects cannot be imported if the format of the CSV file is invalid or if the maximum number of objects has been reached.

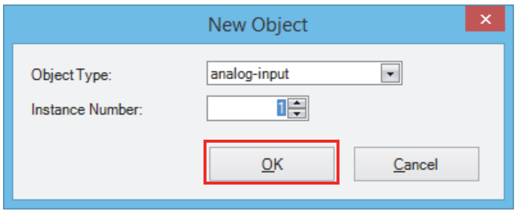
Adding a New Object

This section describes the procedure to add a new object.

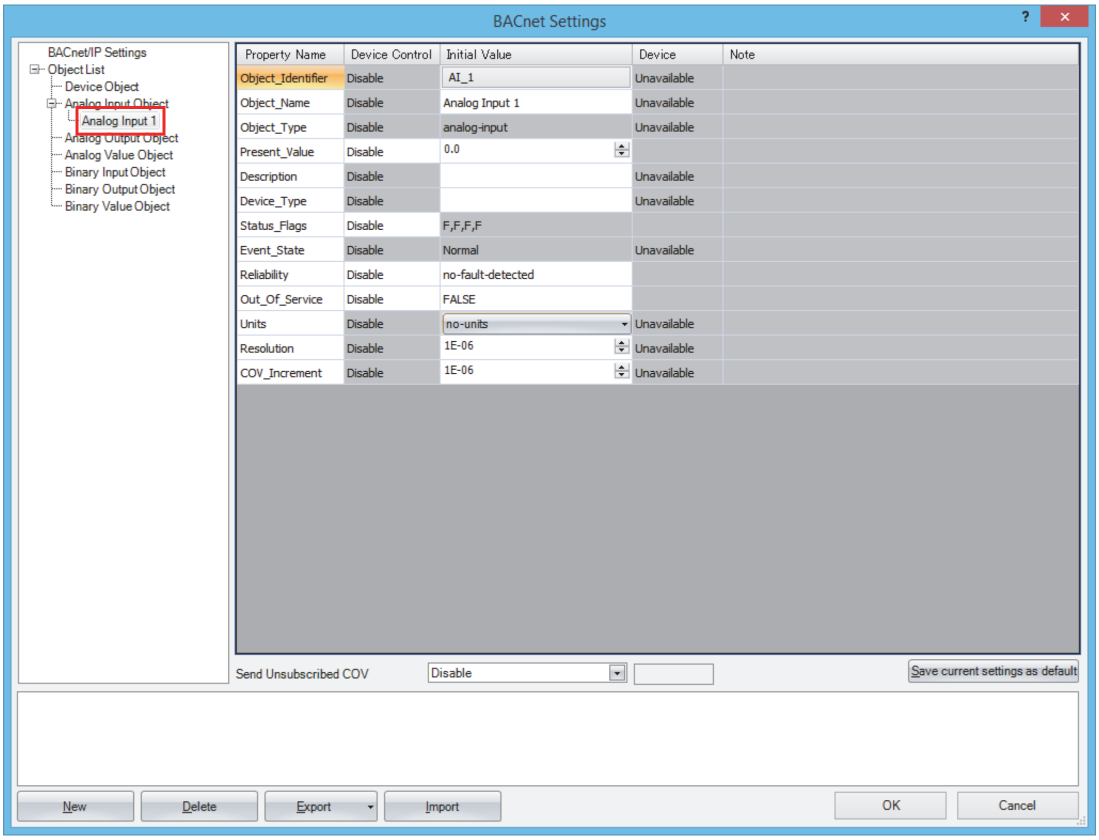
1. Click **New**.
The **New Object** dialog box is displayed.



2. Select the object type of the object to register, and assign it an instance number.
Click **OK**.



3. Click the registered object ID in the object list.
   The properties are displayed.



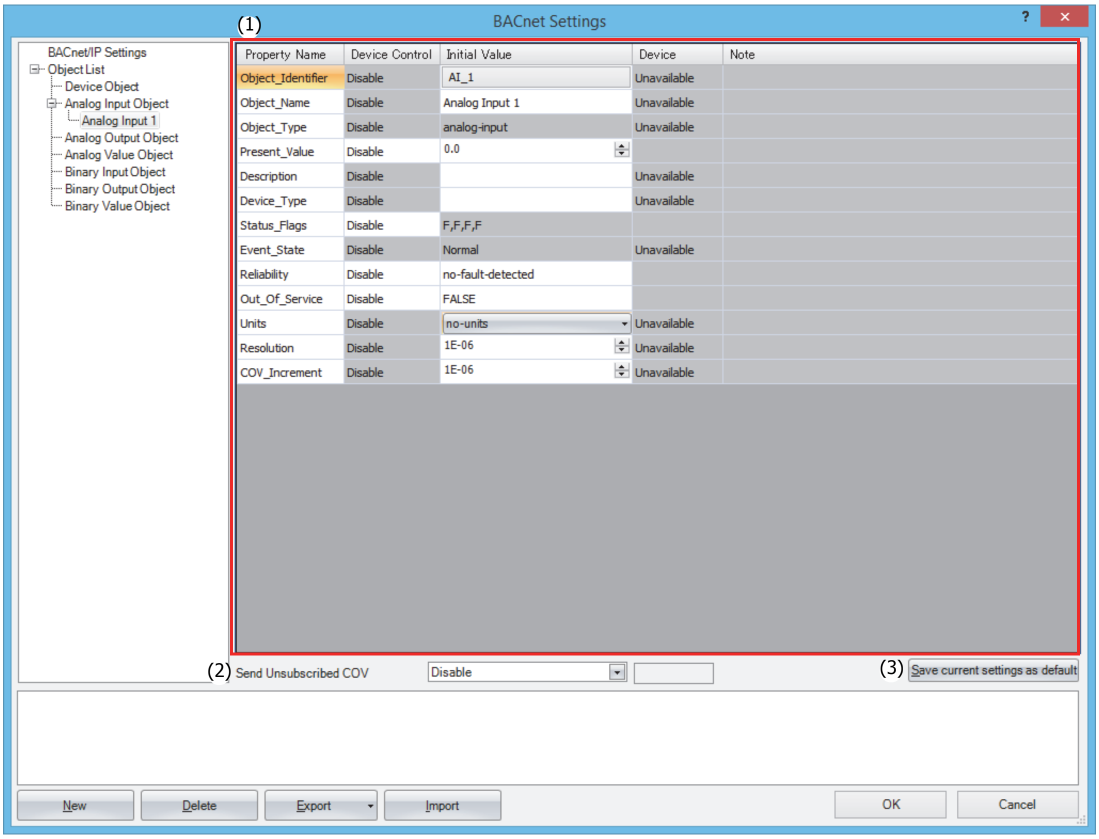
4. Configure the properties and click **OK**.
   For details, see "Object Settings" on page 15-20.

Note: The maximum number of objects that can be registered is 256.

Object Settings

This section describes the settings of properties held by objects.

This screenshot shows an example of when a registered object ID was selected on the **BACnet Settings** dialog box.



(1) Property settings

This area displays the properties of the object ID selected in the object list.

A portion of the properties can be edited.

(2) Send Unsubscribed COV

Sets whether or not to send Unsubscribed COV.

This property can be set from the following three types.

- Enable
- Disable
- Control by device

Setting	Description
Enable	When BACnet Communication Bit (M8450) is on, the COVU function is always enabled and the Unconfirmed COV Notification service is sent.
Disable	The COVU function is always disabled and the Unconfirmed COV Notification service is not sent.
Control by device	The function can be enabled and disabled by setting an internal relay and turning it on and off. A special internal relay cannot be set.

(3) **Save current settings as default** button

Saves the settings of the properties as the default values of the object.

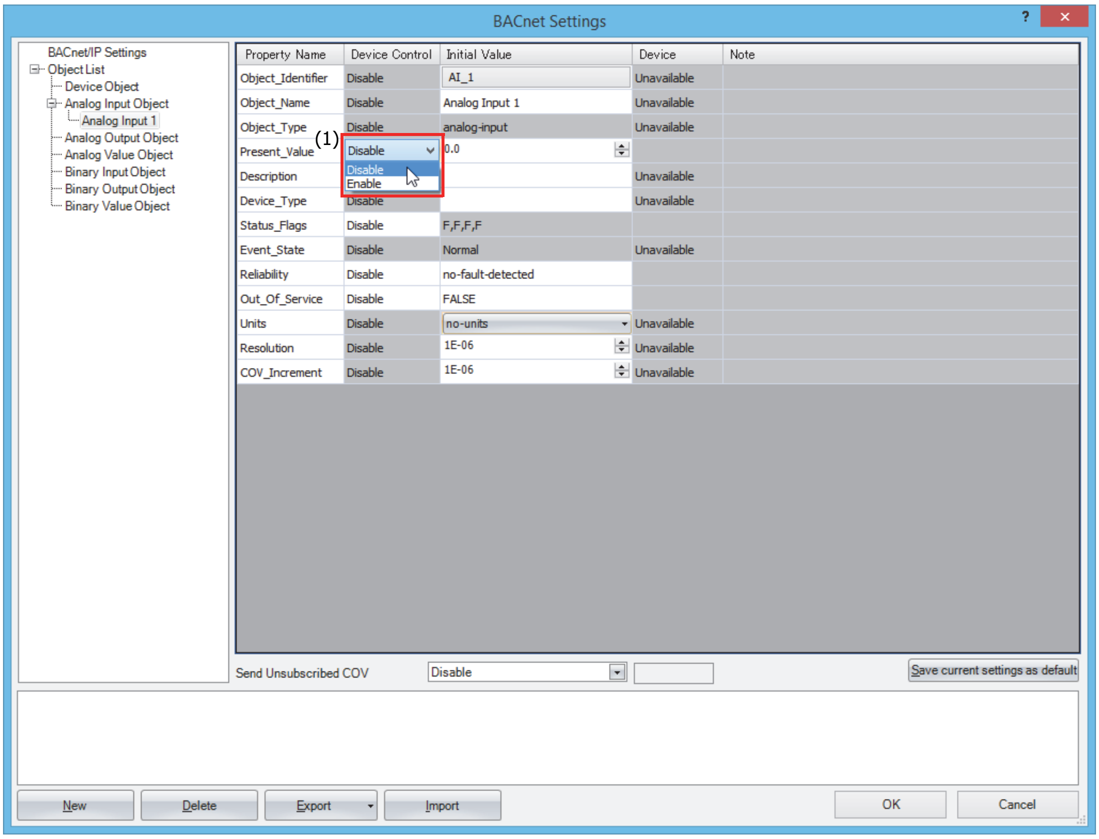
Present_Value Settings

This section describes how to configure Present_Value.

Analog Input Object

Use these settings to configure fixed values and devices for Present_Value of the Analog Input object.

This screenshot shows an example of when an Analog Input object ID is selected in the object list.



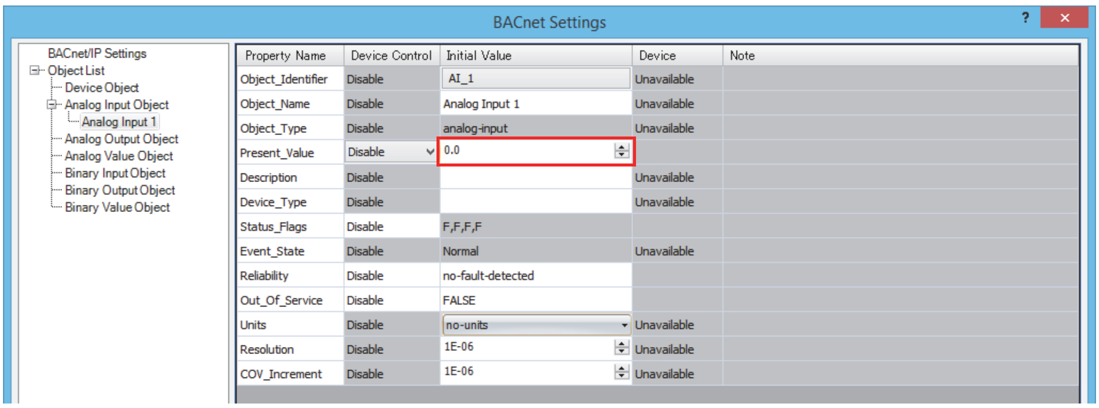
(1) Device Control

Selects whether to set a fixed value for Present_Value or to allocate data registers.

Device Control	Description
Disable	Set a fixed value for Present_Value.
Enable	Allocate data registers for Present_Value.

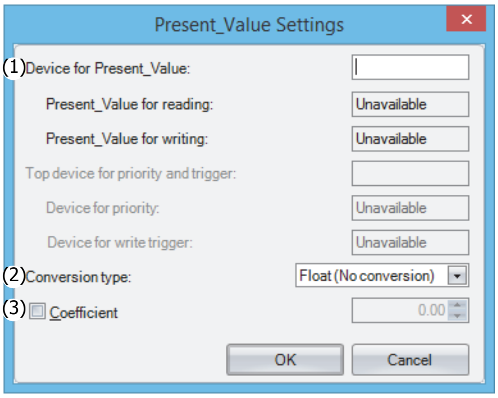
■Setting a Fixed Value for Present_Value

Set device control of Present_Value (1) to **Disable**, and enter a fixed value for the initial value.



■Allocating Devices to Present_Value

Set device control of Present_Value (1) to **Enable**, and configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Sets the device to store Present_Value.

Data registers are the only device that can be allocated to Present_Value.

Properties	I	Q	M	R	T	C	D	Constant
Present_Value	—	—	—	—	—	—	Yes*1	—

*1 Special data registers cannot be used.

According to **Conversion type** (2), 1 or 2 continuous words of data registers are used starting from the specified data register. Specify the first data register so that the device range is not exceeded.

Device for Present_Value	Storage Destination	
	Word Integer	Double Word Long Float (No Conversion)
Present_Value for writing	Starting number+0	Starting number+0, starting number+1

(2) Conversion type

Sets the data type of the device to which Present_Value (float) is allocated. For details, see Present_Value in "Analog Input Object" on page 15-45.

(3) Coefficient

Sets Present_Value as the product of the value stored in the data registers multiplied by the coefficient.

Present_Value = Value stored in the data registers × Coefficient

For example, Present_Value is calculated as shown in the following table.

Plus CPU module			Coefficient	BACnet Settings Device
Data Register	Data Type	Value		Present_Value
D0000	Word	1,000	0.01	10.0
D0000, D0001	Float	2.5	0.5	1.25

Notes:

The coefficient is multiplied as a float-type value. The conversion order is as follows.

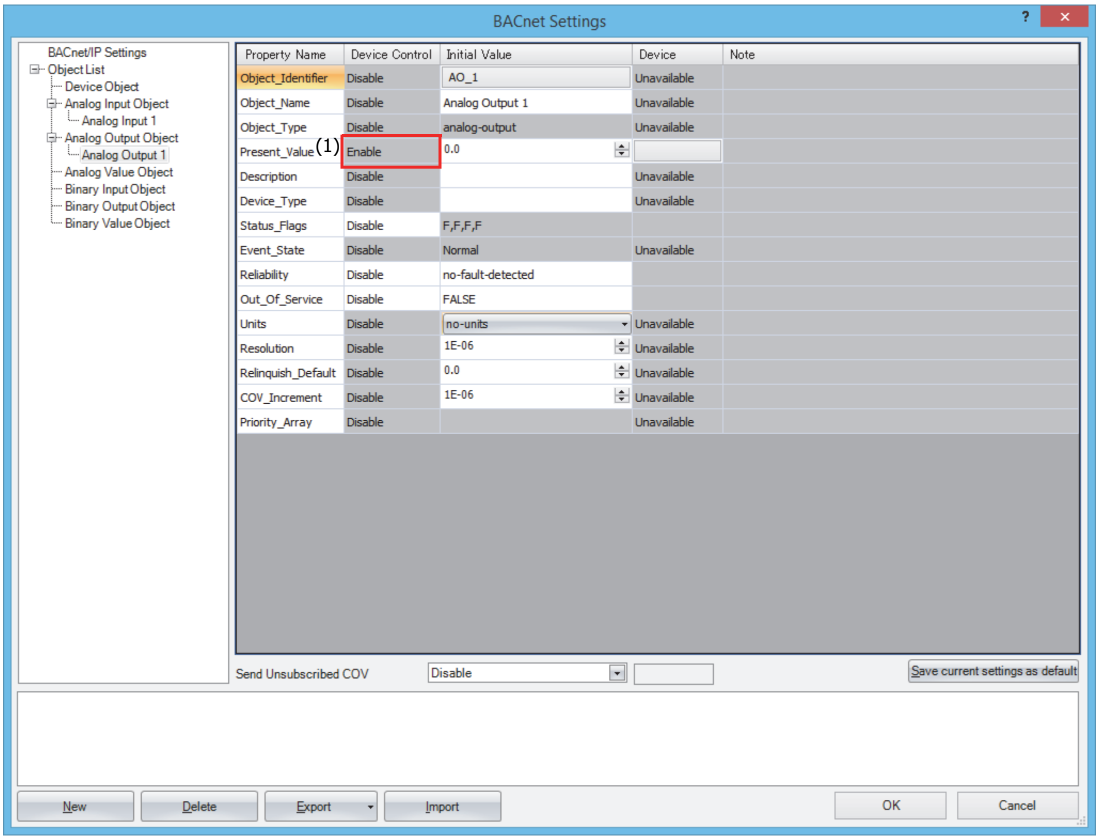
(1) The value of the data registers is converted to a float-type value.

(2) The coefficient is multiplied by the conversion result in (1).

Analog Output Object

Use these settings to configure devices for Present_Value of the Analog Output object.

This screenshot shows an example of when an Analog Output object ID is selected in the object list.



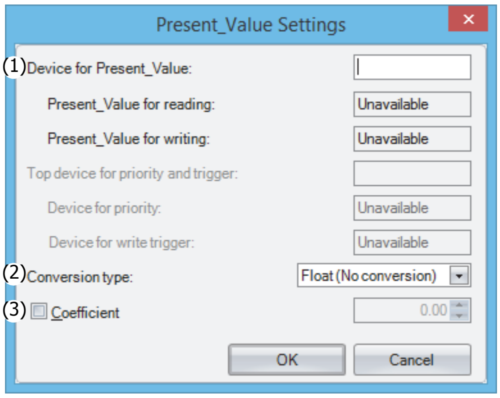
(1) Device Control

Device control for Present_Value is fixed to **Enable**.

Note: A fixed value cannot be set for Present_Value of the Analog Output object.

■Allocating Devices to Present_Value

Configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Sets the device to store Present_Value.

Data registers are the only device that can be allocated to Present_Value.

Properties	I	Q	M	R	T	C	D	Constant
Present_Value	—	—	—	—	—	—	Yes*1	—

*1 Special data registers cannot be used.

According to **Conversion type** (2), 1 or 2 continuous words of data registers are used starting from the specified data register. Specify the first data register so that the device range is not exceeded.

Device for Present_Value	Storage Destination	
	Word Integer	Double Word Long Float (No Conversion)
Present_Value for reading	Starting number+0	Starting number+0, starting number+1

(2) Conversion type

Sets the data type of the device to which Present_Value (float) is allocated. For details, see Present_Value in "Analog Output Object" on page 15-46.

(3) Coefficient

The product of Present_Value multiplied by 1/coefficient multiplied is stored in the data registers.

Value of data registers = Present_Value × (1/coefficient)

For example, when the coefficient and data type are set as follows, Present_Value is stored in the data registers as follows.

BACnet Device	Coefficient	Plus CPU module		
Present_Value		Data Register	Data Type	Value
10.0	0.01	D0000	Word	1,000
1.25	0.5	D0000, D0001	Float	2.5

Notes:

The coefficient is multiplied as a float-type value. The conversion order is as follows.

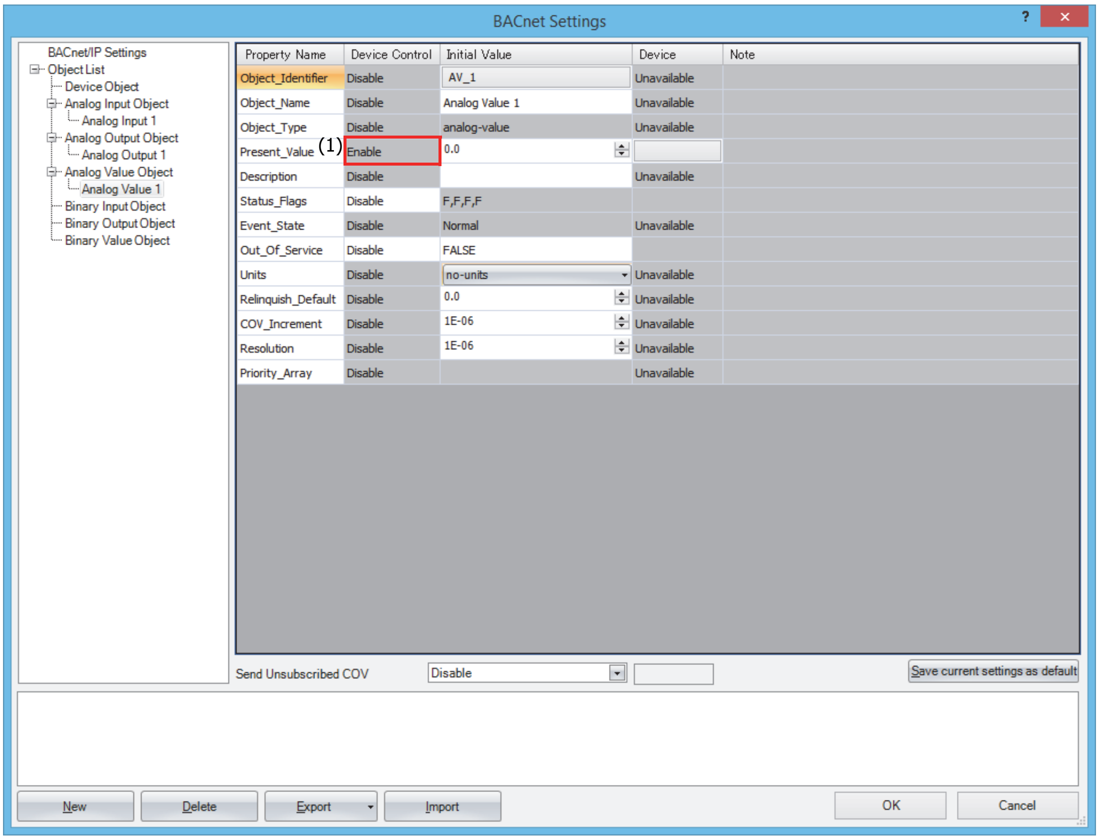
(1) Present_Value is multiplied by (1/coefficient).

(2) The data type of the result of (1) (float) is converted.

Analog Value Object

Use these settings to configure devices for Present_Value of the Analog Value object.

This screenshot shows an example of when an Analog Value object ID is selected in the object list.



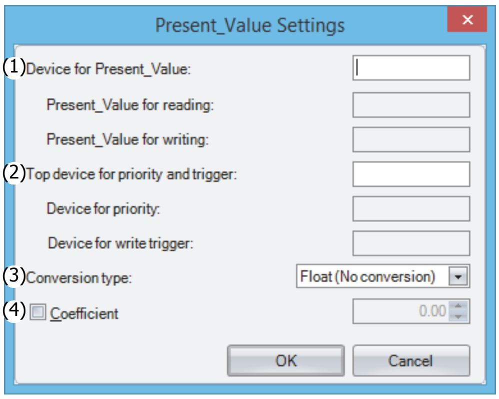
(1) Device Control

Device control for Present_Value is fixed to **Enable**.

Note: A fixed value cannot be set for Present_Value of the Analog Value object.

■Allocating Devices to Present_Value

Configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Sets the device for reading Present_Value and the device for writing Present_Value.
Data registers are the only device that can be allocated to Present_Value.

Properties	I	Q	M	R	T	C	D	Constant
Present_Value	—	—	—	—	—	—	Yes*1	—

*1 Special data registers cannot be used.

The devices for reading Present_Value and writing Present_Value are automatically assigned according to the set device and **Conversion type** (3). Starting from the specified data register, 2 or 4 continuous words of data registers are used. Specify the first data register so that the device range is not exceeded.

Device for Present_Value	Storage Destination	
	Word Integer	Double Word Long Float (No Conversion)
Present_Value for reading	Starting number+0	Starting number+0, starting number+1
Present_Value for writing	Starting number+1	Starting number+2, starting number+3

(2) Top device for priority and trigger

Use this device when writing the value of the devices to Present_Value. For details, see Present_Value in "Analog Value Object" on page 15-47.
Device for priority and **Device for write trigger** are automatically assigned when the data register is set. Starting from the specified data register, 2 continuous words of data registers are used. Specify the first data register so that the device range is not exceeded.

<table>
<tr><th>Storage Destination</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">Starting number+0</td><td colspan="2">Device for priority</td></tr>
<tr><td>Bit 15</td><td>Write NULL flag*1
OFF: Writes the value of the Present_Value for writing device.
ON: Writes NULL.</td></tr>
<tr><td>Bits 14 to 5</td><td>Disabled</td></tr>
<tr><td>Bits 4 to 0</td><td>Priority*2</td></tr>
<tr><td>Starting number+1</td><td colspan="2">Device for write trigger</td></tr>
</table>

*1 When the Write NULL flag is on and the device for write trigger is turned on from off, NULL will be written to Priority_Array of the index number indicated by the priority.
*2 Specify the priority between 1 and 16. If the priority is out of range, the value will not be written, even if the device for write trigger is used.

(3) Conversion type

Sets the data type of the device to which Present_Value (float) is allocated. For details, see Present_Value in "Analog Value Object" on page 15-47.

(4) Coefficient

The product of Present_Value multiplied by 1/coefficient is stored in the data registers allocated as Present_Value for reading.
Present_Value for reading = Present_Value × (1/coefficient)

When the device for write trigger is turned on from off, the product of multiplying the value of the data registers allocated as Present_Value for writing by the coefficient is set as Present_Value.
Present_Value = Present_Value for writing × Coefficient

Present_Value	Coefficient	Conversion Type	Data Registers Allocated as Present_Value for Reading/Writing	
			Address	Value
10.0	0.01	Word	D0000	1,000
1.25	0.5	Float	D0002, D0003	2.5

The coefficient is multiplied as a float-type value. The conversion order is as follows.

Data registers→Present_Value

(1) The value of the data registers is converted to a float-type value.
(2) The coefficient is multiplied by the conversion result in (1).

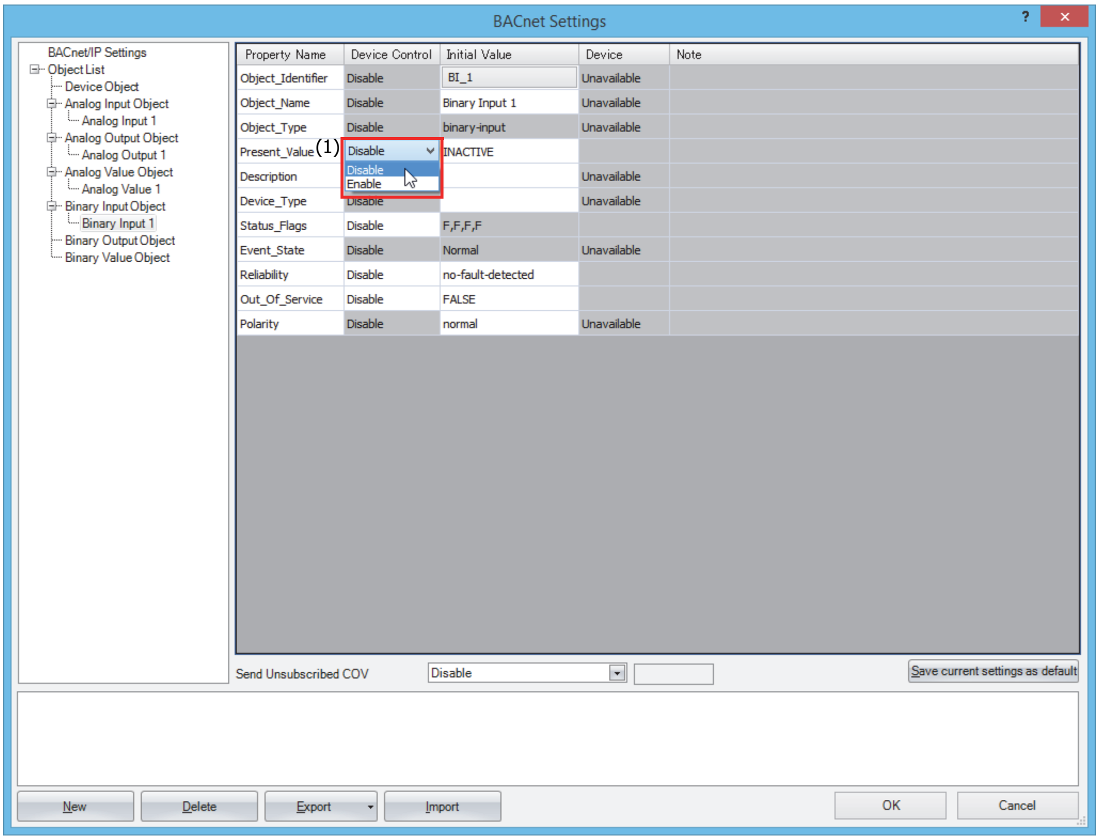
Present_Value→Data registers

(1) Present_Value is multiplied by (1/coefficient).
(2) The data type of the result of (1) (float) is converted.

Binary Input Object

Use these settings to configure fixed values and devices for Present_Value of the Binary Input object.

This screenshot shows an example of when a Binary Input object ID is selected in the object list.



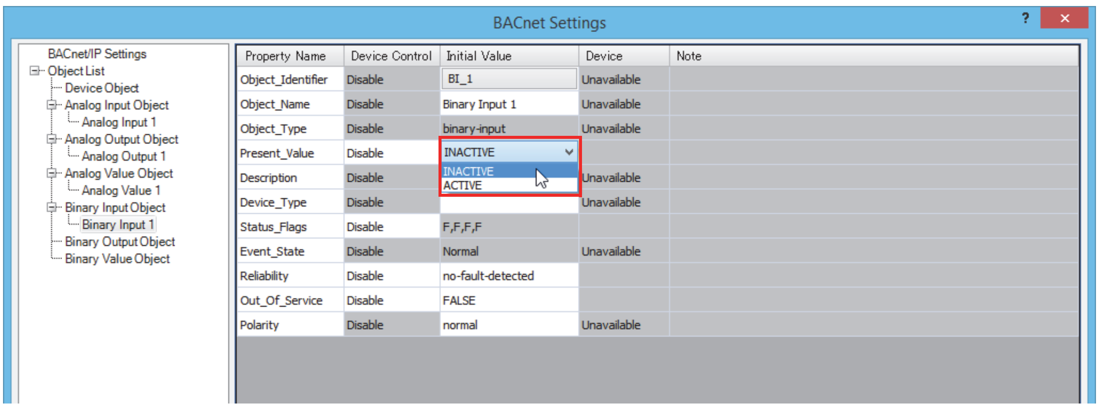
(1) Device Control

Selects whether to set a fixed value for Present_Value or to allocate data registers.

Device Control	Description
Disable	Set a fixed value for Present_Value.
Enable	Allocate data registers for Present_Value.

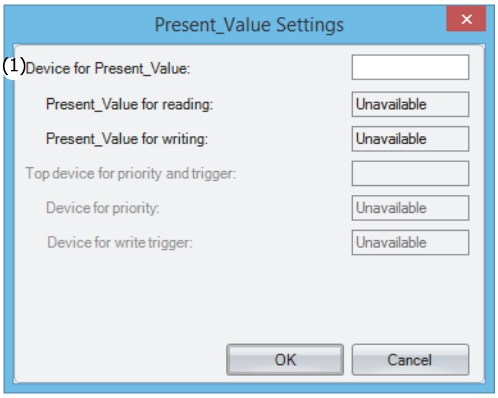
■Setting a Fixed Value for Present_Value

Set device control of Present_Value (1) to **Disable**, and set a fixed value for the initial value.



■Allocating Devices to Present_Value

Set device control of Present_Value (1) to **Enable**, and configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Sets the device to store Present_Value.

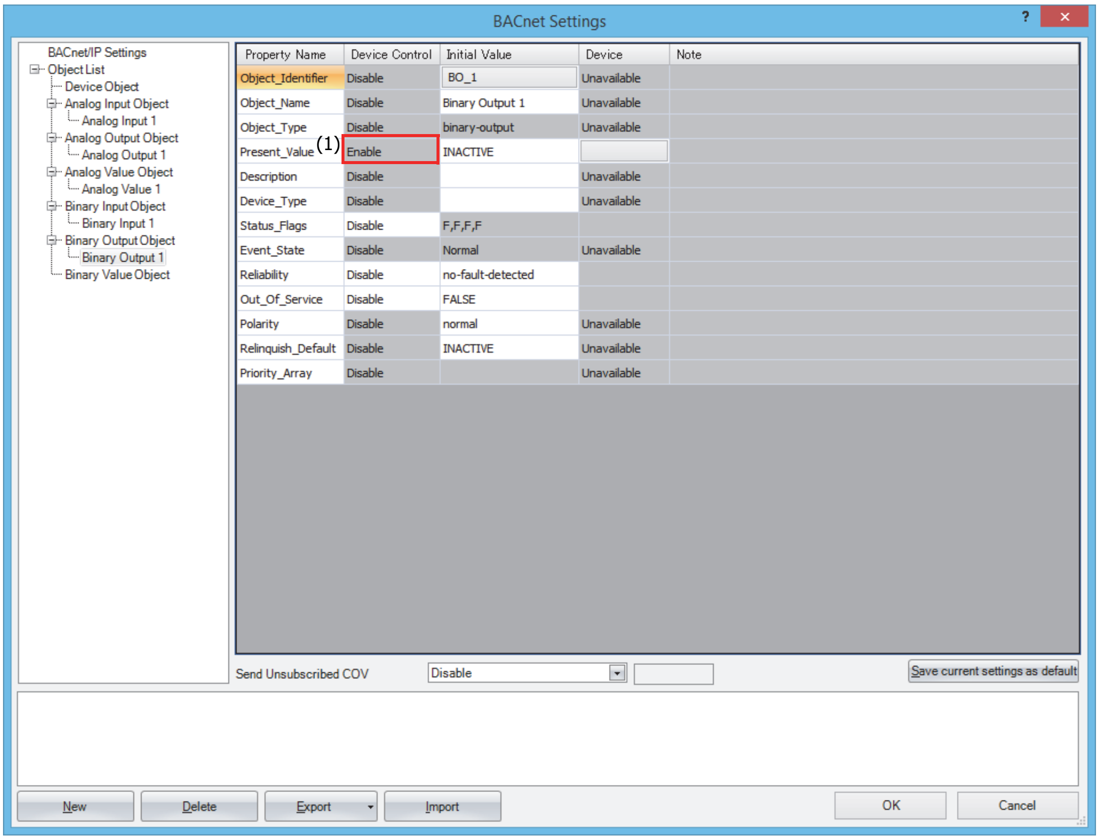
Properties	I	Q	M	R	T	C	D	Constant
Present_Value	Yes	—	Yes*1	—	—	—	—	—

*1 Special internal relays cannot be used.

Binary Output Object

Use these settings to configure devices for Present_Value of the Binary Value object.

This screenshot shows an example of when a Binary Output object ID is selected in the object list.



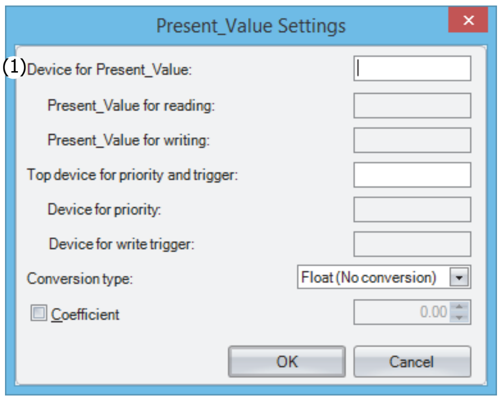
(1) Device Control

Device control for Present_Value is fixed to **Enable**.

Note: A fixed value cannot be set for Present_Value of the Binary Output object.

■Allocating Devices to Present_Value

Configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Sets the device to store Present_Value.

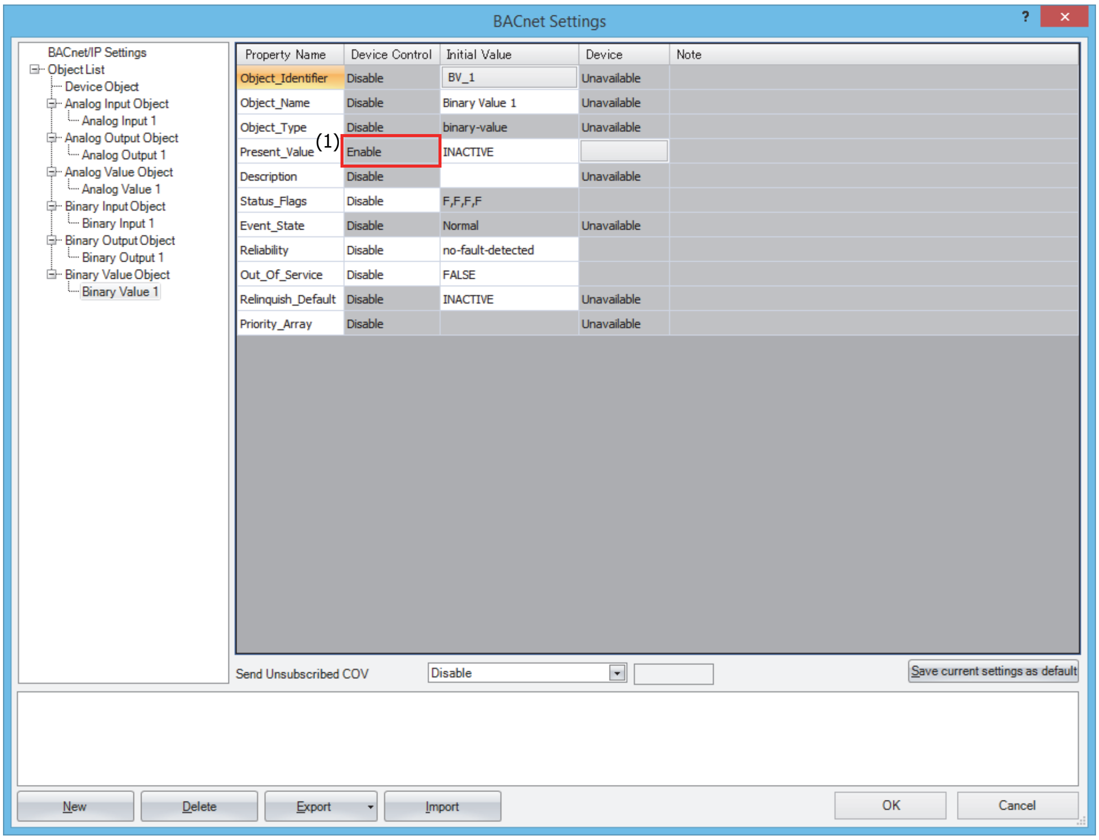
Properties	I	Q	M	R	T	C	D	Constant
Present_Value	—	Yes	Yes*1	—	—	—	—	—

*1 Special internal relays cannot be used.

Binary Value Object

Use these settings to configure devices for Present_Value of the Binary Value object.

This screenshot shows an example of when a Binary Value object ID is selected in the object list.



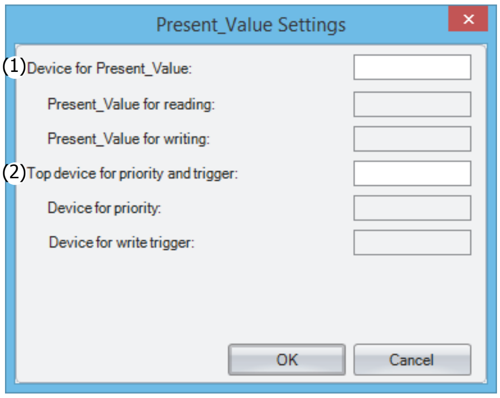
(1) Device Control

Device control for Present_Value is fixed to **Enable**.

Note: A fixed value cannot be set for Present_Value of the Binary Value object.

■Allocating Devices to Present_Value

Configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Sets the device for reading Present_Value and the device for writing Present_Value.

Properties	I	Q	M	R	T	C	D	Constant
Present_Value	—	—	Yes*1	—	—	—	—	—

*1 Special internal relays cannot be used.

The devices for reading Present_Value and writing Present_Value are automatically assigned according to the set device. Starting from the specified data register, 2 bits of the bit device are used. Specify the first bit device so that the device range is not exceeded.

Device for Present_Value	Storage Destination
Present_Value for reading	Starting number+0
Present_Value for writing	Starting number+1

(2) Top device for priority and trigger

Use this device when writing the value of the devices to Present_Value. For details, see Present_Value in "Binary Value Object" on page 15-52.

Device for priority and **Device for write trigger** are automatically assigned when the data register is set. Starting from the specified data register, 2 continuous words of data registers are used. Specify the first data register so that the device range is not exceeded.

<table>
<tr><th>Storage Destination</th><th colspan="2">Description</th></tr>
<tr><td rowspan="4">Starting number+0</td><td colspan="2">Device for priority</td></tr>
<tr><td>Bit 15</td><td>Write NULL flag*1
OFF: Writes the value of the Present_Value for writing device.
ON: Writes NULL.</td></tr>
<tr><td>Bits 14 to 5</td><td>Disabled</td></tr>
<tr><td>Bits 4 to 0</td><td>Priority*2</td></tr>
<tr><td>Starting number+1</td><td colspan="2">Device for write trigger</td></tr>
</table>

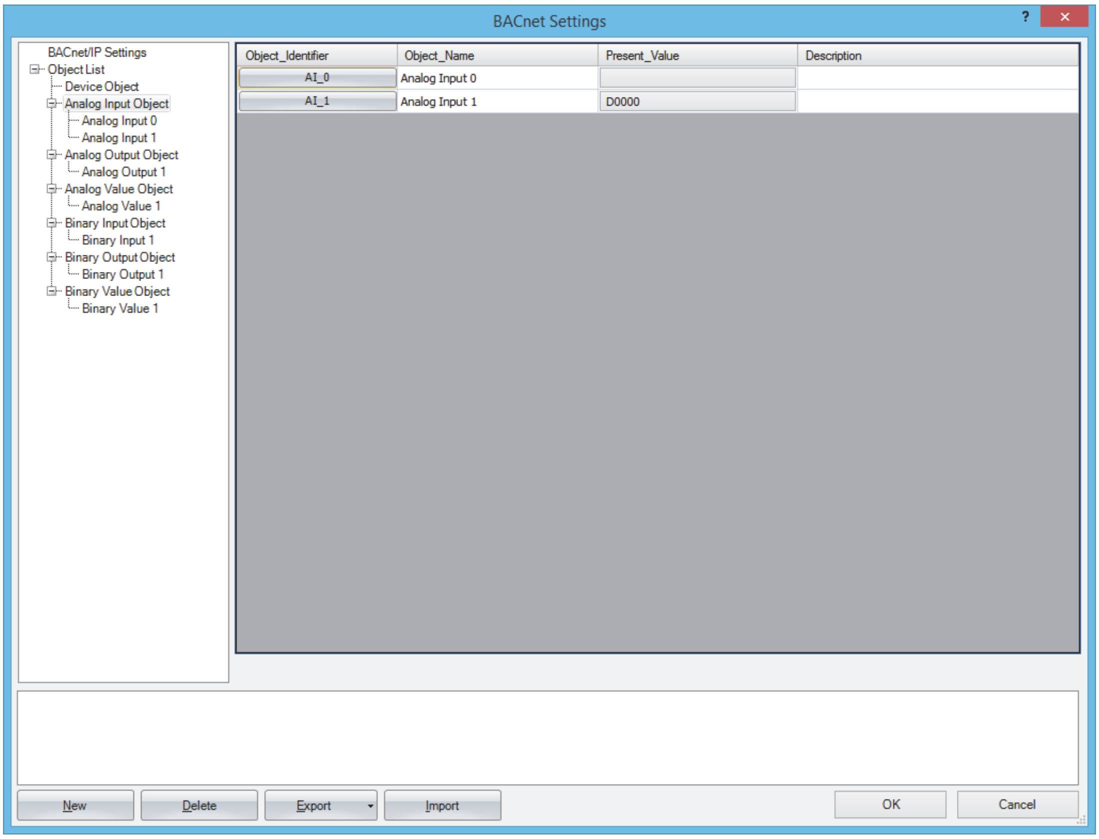
*1 When the Write NULL flag is on and the device for write trigger is turned on from off, NULL will be written to Priority_Array of the index number indicated by the priority.

*2 Specify the priority between 1 and 16. If the priority is out of range, the value will not be written, even if the device for write trigger is used.

Object List

Select an object node to display the list of registered objects.

For example, select Analog Input to display the list of Analog Input objects, and select the object list node to display the list of all objects. The properties can also be changed when the list is displayed.



Objects

The Plus CPU module holds objects registered with WindLDR in internal memory. A portion of the properties held by objects can be allocated to devices, and those devices can be written to and read from a ladder program. BACnet devices can read and write the properties of objects on the Plus CPU module using services.

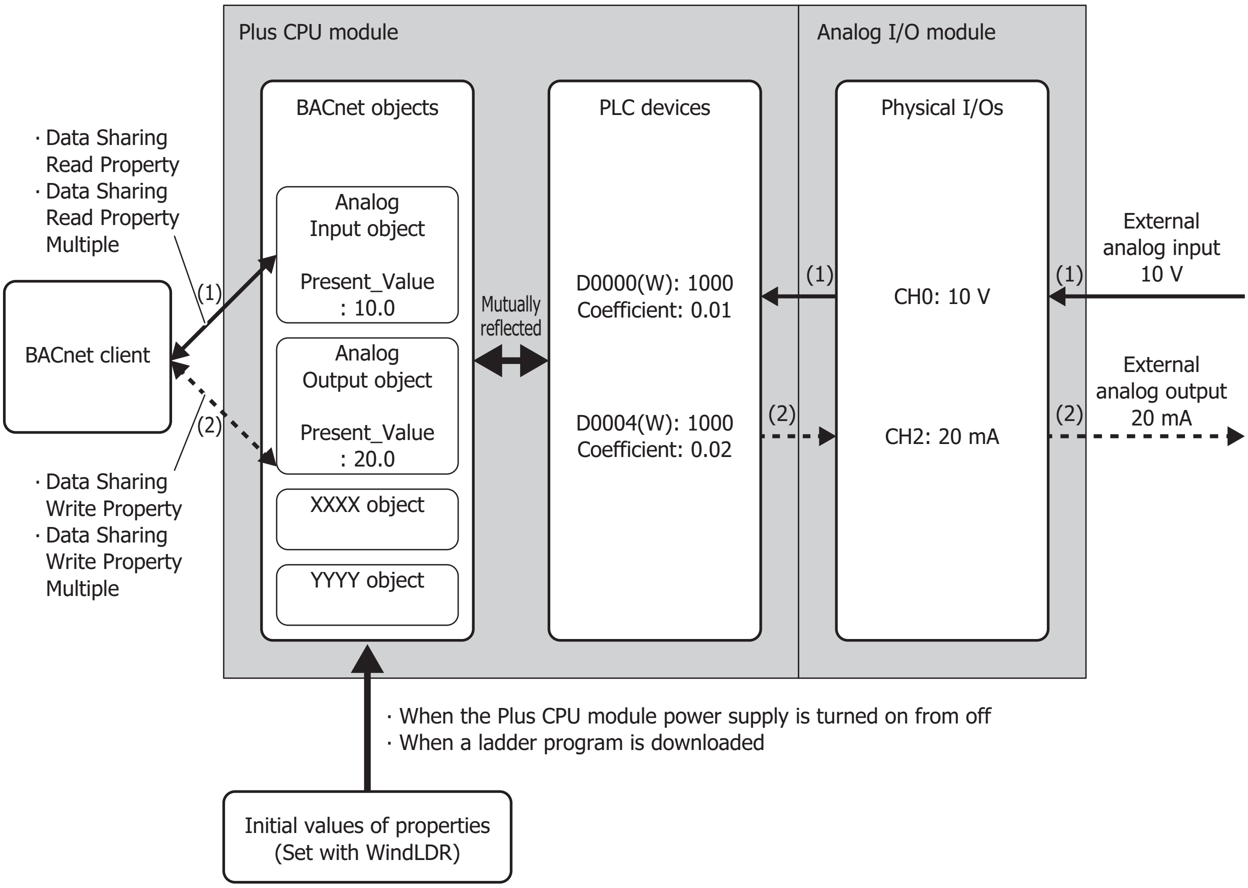
Properties of objects on the Plus CPU module and values of devices allocated to properties are continually synchronized.

For how to register objects using WindLDR, see "Adding a New Object" on page 15-18. The maximum number of objects that can be registered is 256.

The following diagram shows two concepts.

(1) The BACnet device reads the analog input value of the Analog Input object.

(2) The BACnet device writes the analog output value of the Analog Output object.



For details on properties held by objects, refer to "BACnet Building Automation Data Communication Protocol".

Analog Input Object

This object manages a float-type numeric value. The Plus CPU module can make arbitrary numeric values available to BACnet devices. For example, use this object when handling analog values input with the analog input module, the measured room temperature, and other analog values.

Properties List

This list shows the properties supported by the Analog Input object.

(1): Read/write from BACnet devices

(2): Read/write from ladder programs when properties are allocated to devices

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR.
Present_Value	Real number	R	W	See "Present_Value" on page 15-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR.
Device_Type	Character string*1	R	-	Set a fixed value with WindLDR.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 15-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not the value of Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 15-55.
Units	BACnetEngineeringUnits	R/W	-	Set the initial value with WindLDR.
Resolution	Real number	R	-	Set a fixed value with WindLDR.
COV_Increment	Real number	R/W	-	See "COV_Increment" on page 15-54.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	Hidden	-	Hidden in WindLDR.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Analog Output Object

This object manages a float-type numeric value. The Plus CPU module can receive arbitrary numeric values from BACnet devices. For example, this object is used when receiving analog values such as those values that are output with the analog output module and temperature set points used as operating parameters from BACnet devices.

Properties List

This list shows the properties supported by the Analog Output object.

(1): Read/write from BACnet devices

(2): Read/write from ladder programs when properties are allocated to devices

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR.
Present_Value	Real number	R	R	See "Present_Value" on page 15-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR.
Device_Type	Character string*1	R	-	Set a fixed value with WindLDR.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 15-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not the value of Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 15-55.
Units	BACnetEngineeringUnits	R/W	-	Set the initial value with WindLDR.
Resolution	Real number	R	-	Set a fixed value with WindLDR.
Priority_Array	BACnetPriority_Array	R/W	-	See "Priority_Array" on page 15-54.
Relinquish_Default	Real number	R/W	-	See "Relinquish_Default" on page 15-54.
COV_Increment	Real number	R/W	-	See "COV_Increment" on page 15-54.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	Hidden	-	Hidden in WindLDR.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Analog Value Object

This object can be used in the same manner as the Analog Input object and Analog Output object.

Properties List

This list shows the properties supported by the Analog Value object.

(1): Read/write from BACnet devices

(2): Read/write from ladder programs when properties are allocated to devices

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR.
Present_Value	Real number	R	R	See "Present_Value" on page 15-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 15-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 15-55.
Units	BACnetEngineeringUnits	R/W	-	Set the initial value with WindLDR.
Priority_Array	BACnetPriority_Array	R/W	R/W	See "Priority_Array" on page 15-54.
Relinquish_Default	Real number	R/W	-	See "Relinquish_Default" on page 15-54.
COV_Increment	Real number	R/W	-	See "COV_Increment" on page 15-54.
Resolution	Real number	R	-	Set a fixed value with WindLDR.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	Hidden	-	Hidden in WindLDR.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Binary Input Object

This object manages a binary value (on/off). Use this object when the Plus CPU module makes binary values available to BACnet devices.

Properties List

This list shows the properties supported by the Binary Input object.

(1): Read/write from BACnet devices

(2): Read/write from ladder programs when properties are allocated to devices

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR.
Present_Value	BACnetBinaryPV	R	W	See "Present_Value" on page 15-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR.
Device_Type	Character string*1	R	-	Set a fixed value with WindLDR.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 15-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not the value of Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 15-55.
Polarity	BACnetPolarity	R/W	-	See "Polarity" on page 15-54.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	Hidden	-	Hidden in WindLDR.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Binary Output Object

This object manages a binary value (on/off). Use this object when the Plus CPU module receives binary values from BACnet devices.

Properties List

This list shows the properties supported by the Binary Output object.

(1): Read/write from BACnet devices

(2): Read/write from ladder programs when properties are allocated to devices

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR.
Present_Value	BACnetBinaryPV	R	R	See "Present_Value" on page 15-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR.
Device_Type	Character string*1	R	-	Set a fixed value with WindLDR.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 15-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not the value of Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 15-55.
Polarity	BACnetPolarity	R/W	-	See "Polarity" on page 15-54.
Priority_Array	BACnetPriority_Array	R/W	-	See "Priority_Array" on page 15-54.
Relinquish_Default	BACnetBinaryPV	R/W	-	See "Relinquish_Default" on page 15-54.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	Hidden	-	Hidden in WindLDR.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Binary Value Object

This object can be used in the same manner as the Binary Input object and Binary Output object.

Properties List

This list shows the properties supported by the Binary Value object.

(1): Read/write from BACnet devices

(2): Read/write from ladder programs when properties are allocated to devices

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR.
Present_Value	BACnetBinaryPV	R	R	See "Present_Value" on page 15-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 15-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not the value of Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 15-55.
Priority_Array	BACnetPriority_Array	R/W	R/W	See "Priority_Array" on page 15-54.
Relinquish_Default	BACnetBinaryPV	R/W	-	See "Relinquish_Default" on page 15-54.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	Hidden	-	Hidden in WindLDR.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Device Object

Use this object when the Plus CPU module makes basic information available to BACnet devices.

Properties List

This list shows the properties supported by the Device object.

(1): Read/write from BACnet devices

(2): Read/write from ladder programs when properties are allocated to devices

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Cannot be edited with WindLDR.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR.
Object_Type	BACnetObjectType	R	-	Cannot be edited with WindLDR.
System_Status	BACnetDeviceStatus	R	-	Cannot be edited with WindLDR.
Vendor_Name	Character string*1	R	-	Cannot be edited with WindLDR.
Vendor_Identifier	16-bit unsigned integer	R	-	Cannot be edited with WindLDR.
Model_Name	Character string*1	R	-	Set a fixed value with WindLDR.
Firmware_Revision	Character string*1	R	-	Cannot be edited with WindLDR.
Application_Software_Version	Character string*1	R	-	Set a fixed value with WindLDR.
Location	Character string*1	R/W	-	Set the initial value with WindLDR.
Description	Character string*1	R/W	-	Set the initial value with WindLDR.
Protocol_Version	Unsigned integer	R	-	Cannot be edited with WindLDR.
Protocol_Revision	Unsigned integer	R	-	Cannot be edited with WindLDR.
Protocol_Services_Supported	BACnetServicesSupported	R	-	Cannot be edited with WindLDR.
Protocol_Object_Types_Supported	BACnetObjectType Supported	R	-	Cannot be edited with WindLDR.
Object_List	BACnetARRAY[N] of type BACnetObjectIdentifier	R	-	Cannot be edited with WindLDR.
Max_APDU_Length_Accepted	Unsigned integer	R	-	Cannot be edited with WindLDR.
Segmentation_Supported	BACnetSegmentation	R	-	Cannot be edited with WindLDR.
Local_Time	Time	R	-	Cannot be edited with WindLDR.
Local_Date	Date	R	-	Cannot be edited with WindLDR.
APDU_Timeout	Unsigned integer	R	-	Cannot be edited with WindLDR.
Number_of_APDU_Retries	Unsigned integer	R	-	Cannot be edited with WindLDR.
Device_Address_Binding	BACnetLIST of type BACnetAddressBiding	R	-	Cannot be edited with WindLDR.
Database_Revision	Unsigned integer	R	-	Cannot be edited with WindLDR.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	Hidden	-	Hidden in WindLDR.
Profile_Name	Character string*1	R	-	Set a fixed value with WindLDR.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Key Properties

Present_Value

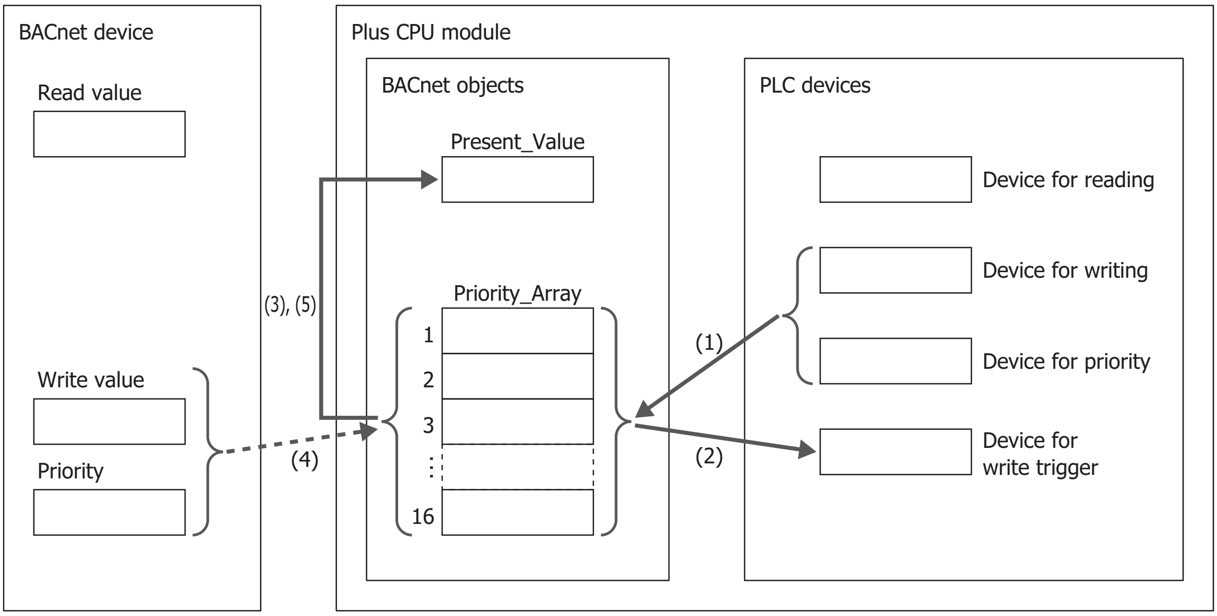
This property represents the current value. This property handles input and output values associated with objects.

Priority Mechanism

Objects[*1] with Present_Value that can be written from BACnet devices use a priority mechanism with Priority_Array in which write instructions are ranked and the value of Present_Value is determined. In this case, values cannot be directly written to Present_Value. To write a value to Present_Value, the value is first stored in Priority_Array ("Present_Value Settings" on page 15-21) at the index number indicated by **Priority** ((4) "Priority_Array" on page 15-54). Then the value stored in Priority_Array with the smallest index number out of all non-NULL values is used as the value of Present_Value.

*1 Analog Output/Analog Value/Binary Output/Binary Value objects

■Writing the Value of a Data Register as Present_Value from a Ladder Program or Writing Present_Value from a BACnet Device



The items used in the above diagram are described in the following table.

Item	Description
Device for reading	This device stores Present_Value read from the BACnet device.
Device for writing	This device stores the value to be written to the BACnet device as Present_Value.
Device for priority	This device stores the index number in the array (Priority_Array) where the priority value is stored.
Device for write trigger	When this device is 1, the value stored in the device for writing is stored in Priority_Array at the index number stored in device for priority.
Present_Value	Prevent_Value held by the BACnet object in the Plus CPU module.
Priority_Array	See "Priority_Array" on page 15-54.
Read value	Current Present_Value.
Write value	Present_Value to be written from the BACnet device.

Writing from a Ladder Program

(1) When bit 15 of **Device for priority** is 0 and **Device for write trigger** is 1, the value of **Device for writing** is written to Priority_Array at the index number stored in **Device for priority**.

(2) **Device for write trigger** is automatically reset to 0.

(3) The value stored in Priority_Array with the smallest index number out of all non-NULL values is used as the value of Present_Value. (The value of Priority_Array at the index number is used as Present_Value until it is reset to NULL.)

Note: When bit 15 of **Device for priority** is 1 and **Device for write trigger** is 1, NULL is written to Priority_Array at the index number stored in **Device for priority**.
For details on the bit assignments of the device for priority, see "Analog Value Object" on page 15-25 in "(2) Top device for priority and trigger" on page 15-26.

Writing from the BACnet Device

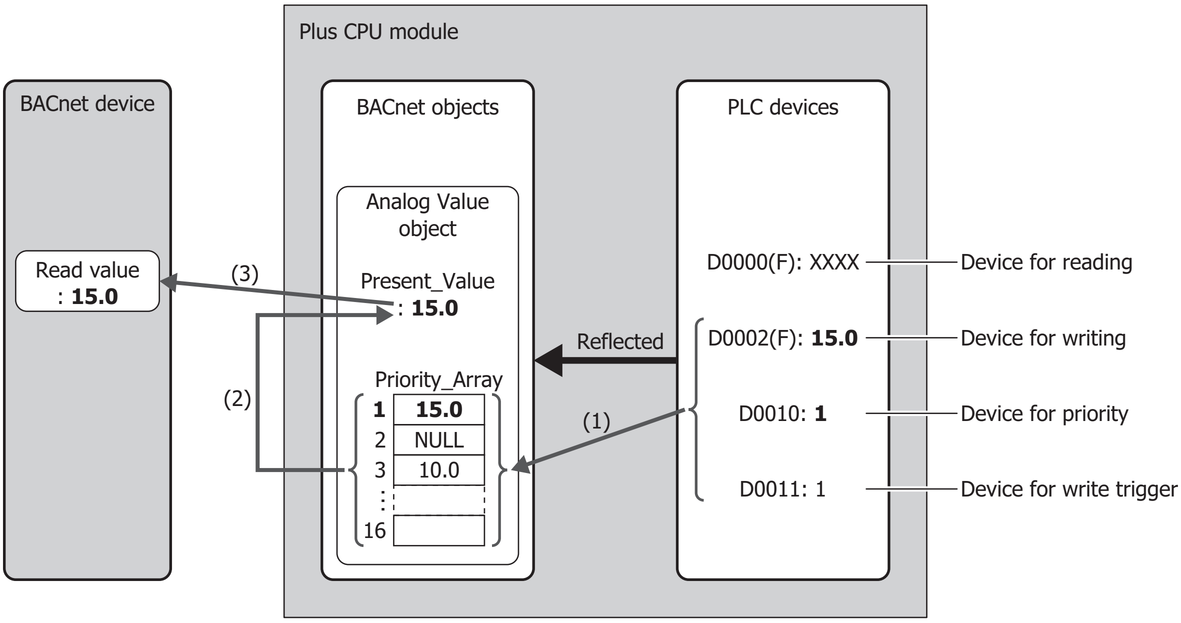
(4) The BACnet device writes the value of Present_Value to Priority_Array at the index number indicated by the specified priority.

(5) The value stored in the Priority_Array property with the smallest index number out of all non-NULL values is used as the value of Present_Value. (The value of Priority_Array at the index number is used as Present_Value until it is reset to NULL.)

Note: If all values stored in Priority_Array are NULL, "Relinquish_Default" on page 15-54 is used as Present_Value.

Writing the Value of a Data Register to Present_Value from a Ladder Program

For example, when D0000 is allocated to the Present_Value device, D0010 is allocated to the top device for priority and trigger, and float (no conversion) is allocated to the conversion type, the devices are allocated as follows: D0000 is the device for reading, D0002 is the device for writing, D0010 is the device for priority, and D0011 is the device for write trigger.



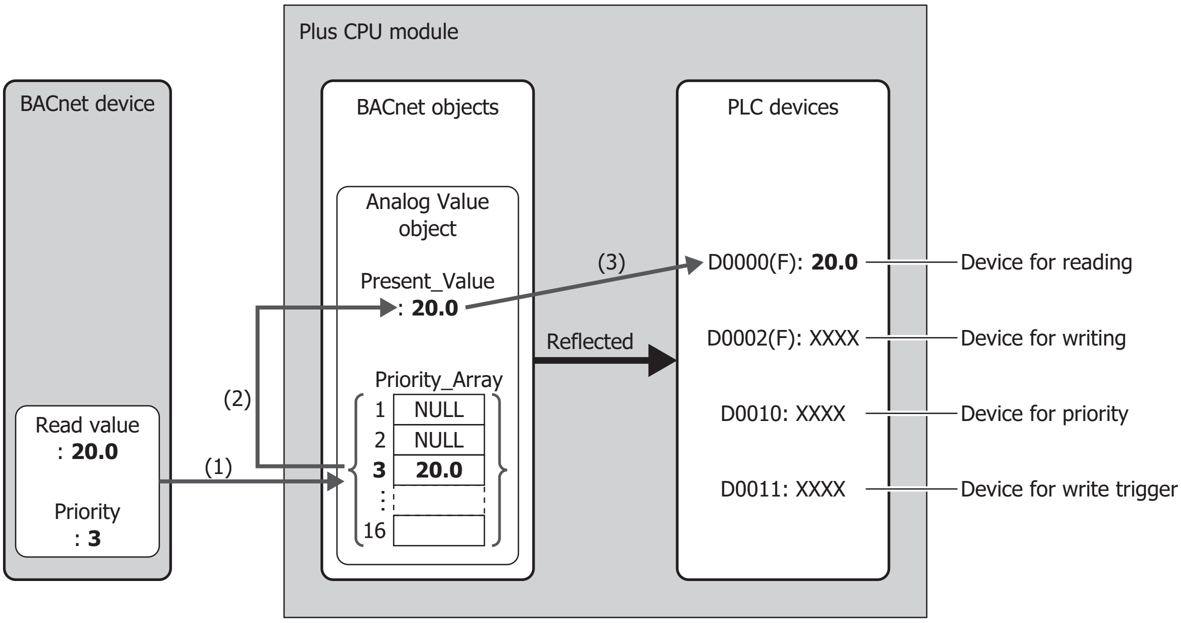
(1) When the device for write trigger (D0011) is 1, the device for priority (D0010) is 1, so the value (15.0) of the device for writing (D0002) is written to element 1 of Priority_Array. After the write has completed, the device for write trigger (D0011) is reset to 0.

(2) Priority_Array with the smallest index number out of all non-NULL values is element 1 (15.0), so 15.0 is used as the value of Present_Value.

(3) Present_Value (15.0) is read from the BACnet device.

Writing a Value to Present_Value from the BACnet Device

For example, when D0000 is allocated to the Present_Value device, D0010 is allocated to the top device for priority and trigger, and float (no conversion) is allocated to the conversion type, the devices are allocated as follows: D0000 is the device for reading, D0002 is the device for writing, D0010 is the device for priority, and D0011 is the device for write trigger.



(1) The BACnet device writes the write value (20.0) to Priority_Array at the index number indicated by the specified priority (3).

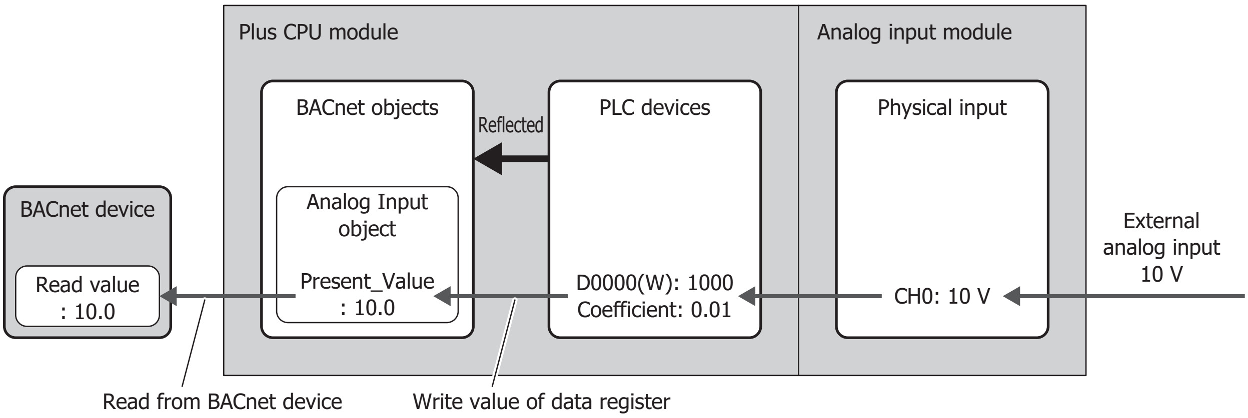
(2) Priority_Array with the smallest index number out of all non-NULL values is element 3 (20.0), so 20.0 is used as the value of Present_Value.

(3) Present_Value (20.0) is written to the device for reading (D0000).

Analog Input Object

Present_Value of the Analog Input object can be set to a fixed value or allocated data registers and set to the value of those data register. Present_Value is a float-type numeric value. When data registers are allocated to Present_Value, Present_Value is set to the product of those data registers multiplied by the coefficient.

The following diagram illustrates the concept when data registers are allocated to Present_Value of the Analog Input object and the BACnet device reads out the analog input value.



Data registers are the only device that can be allocated to Present_Value of the Analog Input object.

Properties	I	Q	M	R	T	C	D	Constant
Present_Value	—	—	—	—	—	—	Yes*1	—

*1 Special data registers cannot be used.

When writing the value of data registers as Present_Value, Present_Value changes in the following manner.

Data Type	Value of Data Registers	Present_Value of BACnet Device
Word	Value within range of each data type	Value of Data Registers
Integer		
Double Word		
Long		
Float (No Conversion)	±0	±0.0
	Denormalized number	Value of Data Registers
	Normalized number	
	±∞ (±infinity)	Present_Value does not change
	Non-number	

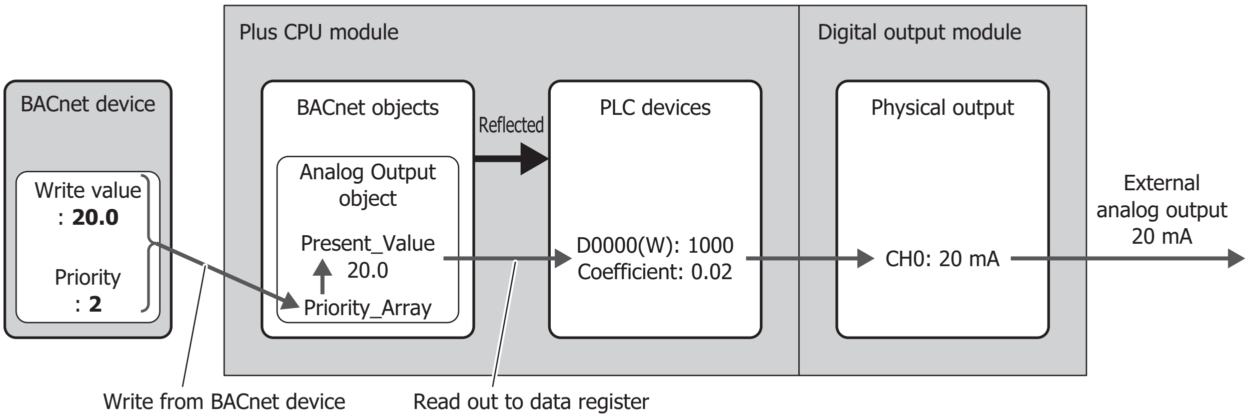
The number of data registers required for writing Present_Value of the Analog Input object handled by the Plus CPU module depends on the data type.

Data Type of Present_Value Handled by the Plus CPU Module	Number of Data Registers
Word	1
Integer	
Double Word	2
Long	
Float (No Conversion)	

Analog Output Object

Present_Value of the Analog Output object can be allocated data registers and set to the value of those data registers. Present_Value is a float-type numeric value. When data registers are allocated to Present_Value, the product of Present_Value multiplied by 1/coefficient is stored in the data registers.

The following diagram illustrates the concept when data registers are allocated to Present_Value of the Analog Output object and the BACnet device writes the analog output value.



Note: Present_Value of the Analog Output object cannot be changed from a ladder program.

Data registers are the only device that can be allocated to Present_Value of the Analog Output object.

Properties	I	Q	M	R	T	C	D	Constant
Present_Value	—	—	—	—	—	—	Yes*1	—

*1 Special data registers cannot be used.

When reading out Present_Value to data registers, you must be aware of the data type. Depending on the data type of the data registers, the value is stored in the data registers as follows. Set the data type according to the value of Present_Value. The data type of Present_Value is set in "(2) Conversion type" on page 15-24 of the **Present_Value Settings** dialog box (page 15-24).

Data Type	Present_Value of BACnet Device	Value of Data Registers
Word	Value within range of 0 to 65,535	Present_Value
	Value outside range of 0 to 65,535	0
Integer	Value within range of -32,768 to 32,767	Present_Value
	Value outside range of -32,768 to 32,767	0
Double Word	Value within range of 0 to 4,294,967,295	Present_Value
	Value outside range of 0 to 4,294,967,295	0
Long	Value within range of -2,147,483,648 to 2,147,483,647	Present_Value
	Value outside range of -2,147,483,648 to 2,147,483,647	0
Float (No Conversion)	—	Present_Value

The number of data registers required for reading Present_Value of the Analog Output object handled by the Plus CPU module depends on the data type.

Data Type of Present_Value Handled by the Plus CPU Module	Number of Data Registers
Word	1
Integer	
Double Word	2
Long	
Float (No Conversion)	

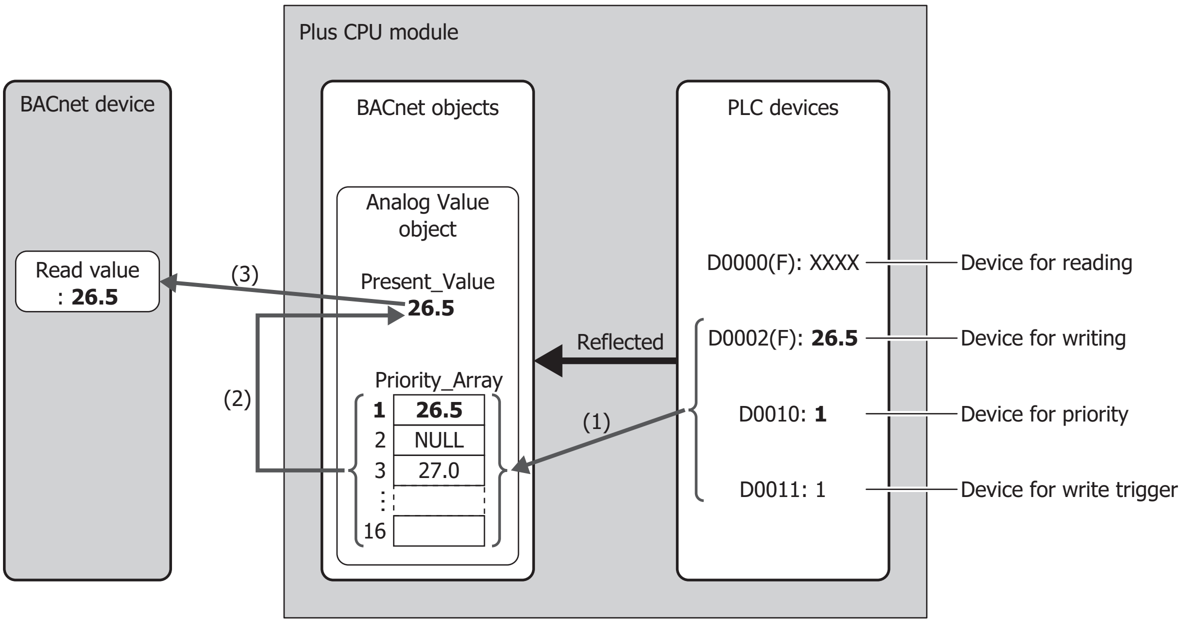
Analog Value Object

Present_Value of the Analog Value object can be used as Present_Value of both the Analog Input and Analog Output objects. Present_Value is a float-type numeric value.

The following diagram illustrates the concept when data registers are allocated to Present_Value of the Analog Value object and the temperature of an office air conditioner is temporarily lowered from the base temperature (27.0°C) to 26.5°C.

For example, when D0000 is allocated to the Present_Value device, D0010 is allocated to the top device for priority and trigger, and float (no conversion) is allocated to the conversion type, the devices are allocated as follows: D0000 is the device for reading, D0002 is the device for writing, D0010 is the device for priority, and D0011 is the device for write trigger.

Writing the Value of a Data Register to Present_Value from a Ladder Program

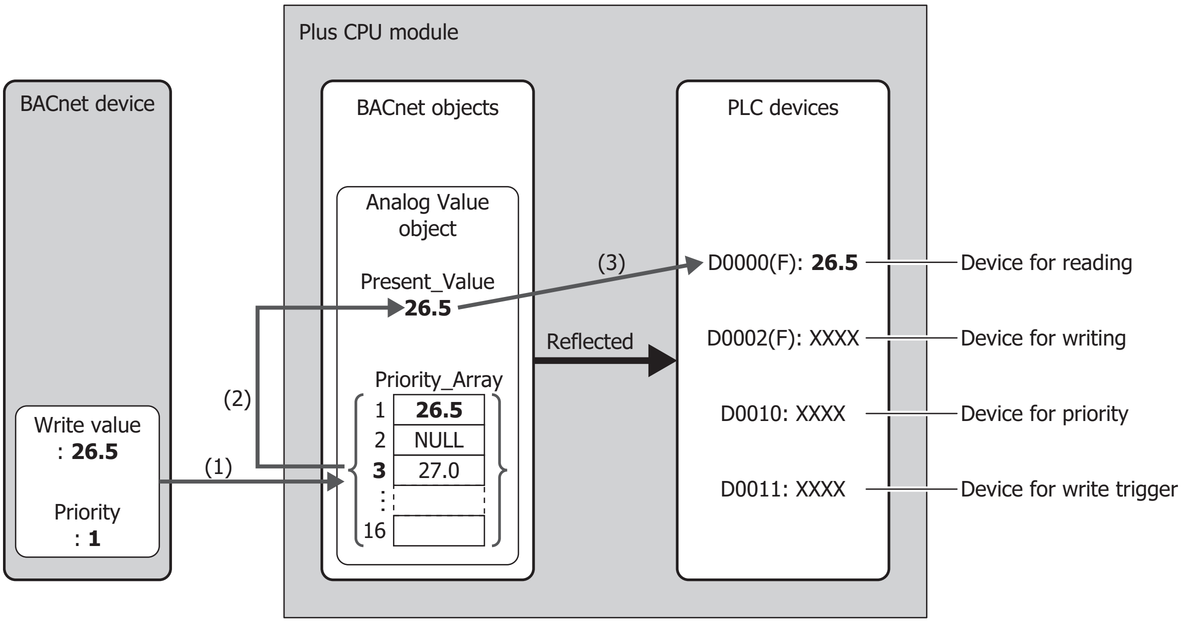


(1) When the device for write trigger (D0011) is 1, the value of the device for writing (D0002) is written to Priority_Array at the index number stored in the device for priority (D0010).

(2) The device for write trigger (D0011) is automatically reset to 0.

(3) Priority_Array with the smallest index number out of all non-NULL values is element 1 (26.5), so 26.5 is used as the value of Present_Value.

Writing a Value to Present_Value from the BACnet Device



(1) The BACnet device writes the write value (26.5) to Priority_Array at the index number indicated by the specified priority (1).

(2) Priority_Array with the smallest index number out of all non-NULL values is element 1 (26.5), so 26.5 is used as the value of Present_Value.

(3) Present_Value (26.5) is written to the device for reading (D0000).

Note: Element 1 (26.5) of Priority_Array is used as the value of Present_Value until it is reset to NULL. When element 1 is reset to NULL, element 3 (27.0) is used as the value of Present_Value.
For how to write NULL, see "Analog Value Object" on page 15-25 in "(2) Top device for priority and trigger" on page 15-26.

Data registers are the only device that can be allocated to Present_Value of the Analog Value object.

Properties	I	Q	M	R	T	C	D	Constant
Present_Value	—	—	—	—	—	—	Yes*1	—

*1 Special data registers cannot be used.

The number of data registers required for reading and writing depends on the data type of Present_Value of the Analog Value object handled by the Plus CPU module. Starting from the specified data register, 2 or 4 data registers are used.

Data Type of Present_Value Handled by the Plus CPU Module	Number of Data Registers		
	Read	Write	Total
Word	1	1	2
Integer			
Double Word	2	2	4
Long			
Float (No Conversion)			

Depending on the data type of the allocated data registers, Present_Value is stored in the data registers as follows. Set the data type according to the value of Present_Value. The data type of Present_Value is set in "(3) Conversion type" on page 15-26 of the **Present_Value Settings** dialog box (page 15-25).

Data Type	Present_Value of BACnet Device × (1/Coefficient)	Value of Data Registers
Word	Value within range of 0 to 65,535	Present_Value
	Value outside range of 0 to 65,535	0
Integer	Value within range of -32,768 to 32,767	Present_Value
	Value outside range of -32768 to 32,767	0
Double Word	Value within range of 0 to 4,294,967,295	Present_Value
	Value outside range of 0 to 4,294,967,295	0
Long	Value within range of -2,147,483,648 to 2,147,483,647	Present_Value
	Value outside range of -2,147,483,648 to 2,147,483,647	0
Float (No Conversion)	—	Present_Value

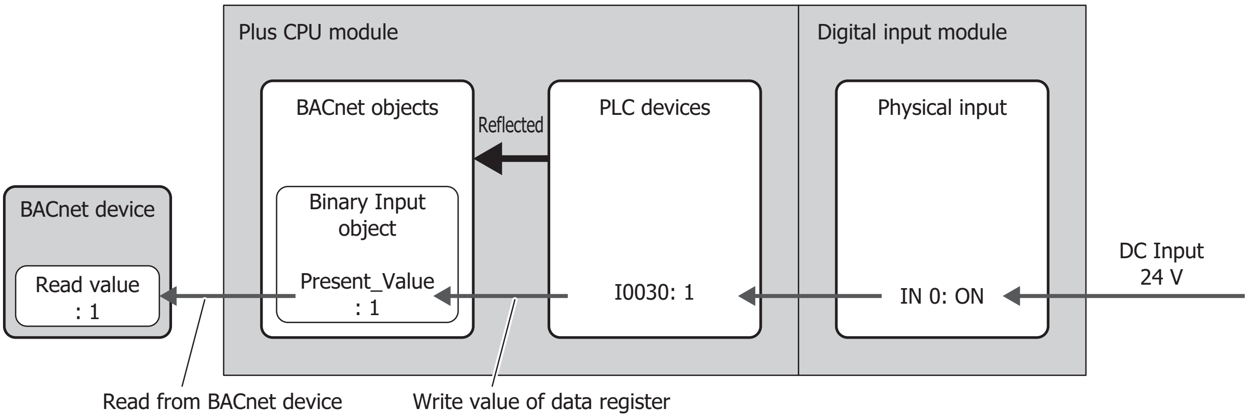
Additionally, when writing the value of data registers to Present_Value, Present_Value changes in the following manner.

Data Type	Value of Data Registers	Present_Value of BACnet Device
Word	Value within range of each data type	Value of data registers × Coefficient
Integer		
Double Word		
Long		
Float (No Conversion)	±0	±0.0
	Denormalized number	Value of data registers × Coefficient
	Normalized number	
	±∞ (±infinity)	Present_Value does not change
	Non-number	

Binary Input Object

Present_Value of the Binary Input object can be set to a fixed value or allocated to a bit device and set to the value of that bit device. Present_Value is a binary value.

The following diagram illustrates the concept when an external input is allocated to Present_Value of the Binary Input object and the BACnet device reads out the state of the external input.



The devices that can be allocated to Present_Value of the Binary Input object are as follows.

Properties	I	Q	M	R	T	C	D	Constant
Present_Value	Yes	—	Yes*1	—	—	—	—	—

*1 Special internal relays cannot be used.

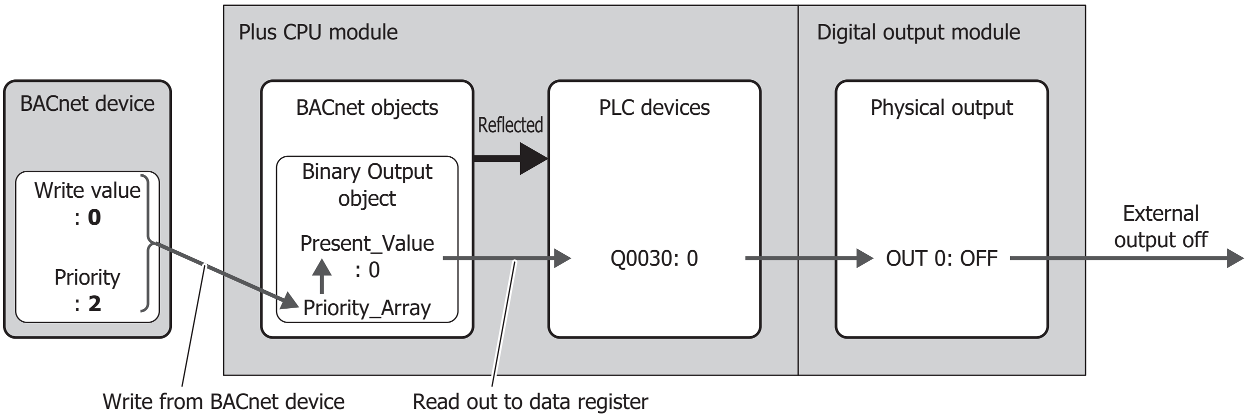
The following table shows Present_Value, Polarity, and the physical state of the input.

Present_Value	Polarity	Physical State of Input
INACTIVE	NORMAL	OFF or INACTIVE
ACTIVE	NORMAL	ON or ACTIVE
INACTIVE	REVERSE	ON or ACTIVE
ACTIVE	REVERSE	OFF or INACTIVE

Binary Output Object

Present_Value of the Binary Output object can be allocated to a bit device and set to the value of that bit device. Present_Value is a binary value.

The following diagram illustrates the concept when a data register is allocated to Present_Value of the Binary Output object and the BACnet device writes the state of the external output.



The devices that can be allocated to Present_Value of the Binary Output object are as follows.

Properties	I	Q	M	R	T	C	D	Constant
Present_Value	—	Yes	Yes*1	—	—	—	—	—

*1 Special internal relays cannot be used.

The following table shows Present_Value, Polarity, and the physical state of the output.

Present_Value	Polarity	Physical State of Output
INACTIVE	NORMAL	OFF or INACTIVE
ACTIVE	NORMAL	ON or ACTIVE
INACTIVE	REVERSE	ON or ACTIVE
ACTIVE	REVERSE	OFF or INACTIVE

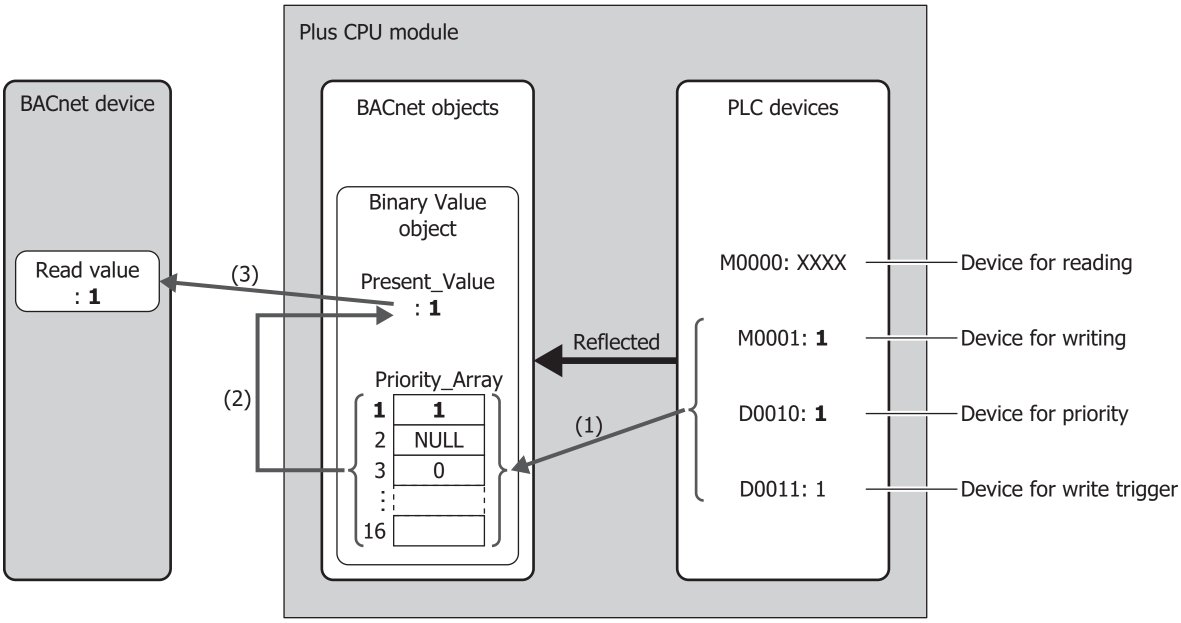
Binary Value Object

Present_Value of the Binary Value object can be used as Present_Value of both the Binary Input and Binary Output objects. Present_Value is a binary value.

The following diagram illustrates the concept when an internal relay is allocated to Present_Value of the Binary Value object and the office lights are temporarily turned on from off.

For example, when M0000 is allocated to the Present_Value device and D0010 is allocated to the top device for priority and trigger, the devices are allocated as follows: M0000 is the device for reading, M0001 is the device for writing, D0010 is the device for priority, and D0011 is the device for write trigger.

Writing the Value of an Internal Relay to Present_Value from a Ladder Program

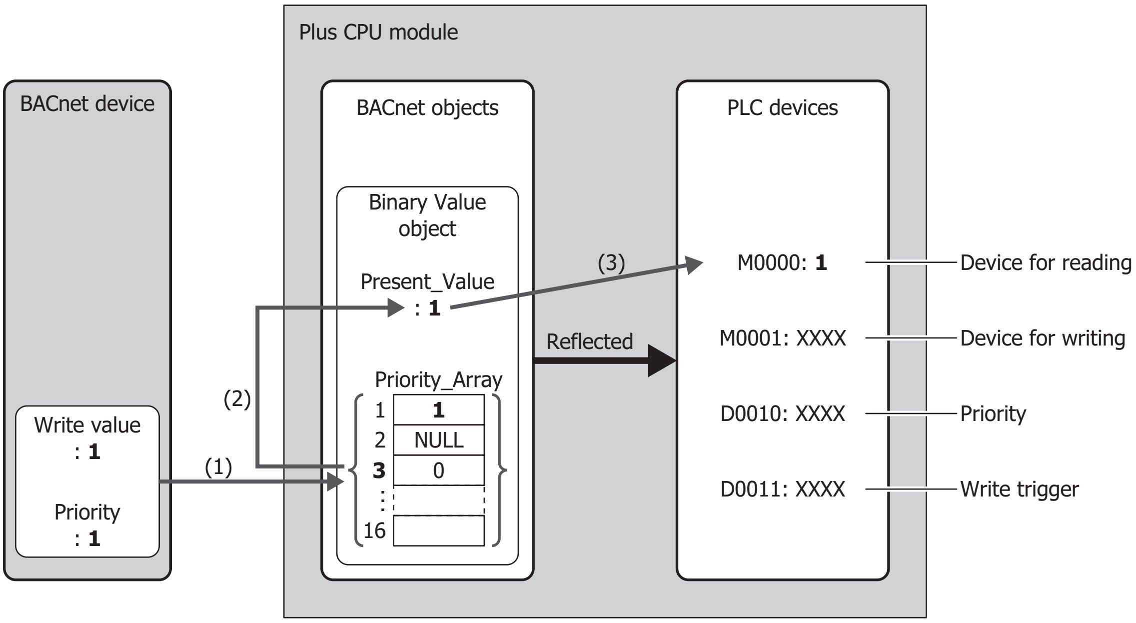


(1) When the device for write trigger (D0011) is 1, the value of the device for writing (M0001) is written to Priority_Array at the index number stored in the device for priority (D0010).

(2) The device for write trigger (D0011) is automatically reset to 0.

(3) Priority_Array with the smallest index number out of all non-NULL values is element 1 (1), so 1 is used as the value of Present_Value.

Writing a Value to Present_Value from the BACnet Device



(1) The BACnet device writes the write value (1) to Priority_Array at the index number indicated by the specified priority (1).

(2) Priority_Array with the smallest index number out of all non-NULL values is element 1 (1), so 1 is used as the value of Present_Value.

(3) Present_Value (1) is written to the device for reading (M0000).

Note: Element 1 (1) of Priority_Array is used as the value of Present_Value until it is reset to NULL. When element 1 is reset to NULL, element 3 (0) is used as the value of Present_Value.
For how to write NULL, see "Binary Value Object" on page 15-30 in "(2) Top device for priority and trigger" on page 15-31.

The devices that can be allocated to Present_Value of the Binary Value object are as follows.

Properties	I	Q	M	R	T	C	D	Constant
Present_Value	—	—	Yes*1	—	—	—	—	—

*1 Special internal relays cannot be used.

Internal relays are required for both reading and writing for Present_Value of the Binary Value object handled by the Plus CPU module. Starting from the specified internal relay, 2 internal relays are used.

Device	Read	Write	Total
Internal relay	1	1	2

Status_Flags

This property represents the current status of the object (in alarm, fault, out of service, etc.).

Status_Flags	Value	Logical Value	Condition
IN_ALARM*1	0	FALSE	When obtaining a value where Event State is Normal.
	1	TRUE	Other than above
FAULT	0	FALSE	Other than below
	1	TRUE	When Reliability is present and the no-fault-detected value is not obtained
OVERRIDDEN*1	0	FALSE	Other than below
	1	TRUE	Present_Value and Reliability did not follow the change in physical input
OUT_OF_SERVICE	0	FALSE	When Out_Of_Service is FALSE
	1	TRUE	When Out_Of_Service is TRUE

*1 Always FALSE on the Plus CPU module.

Status_Flags can be read out to a data register.

Properties	I	Q	M	R	T	C	D	Constant
Status_Flags	—	—	—	—	—	—	Yes*1	—

*1 Special data registers cannot be used.

The allocation of flags in the data register (bit assignment) is as follows.

Bit	Flag	Value
Bits 15 to 8	Reserved	Undefined
Bit 7	IN_ALARM	0 (fixed)
Bit 6	FAULT	0 / 1
Bit 5	OVERRIDDEN	0 (fixed)
Bit 4	OUT_OF_SERVICE	0 / 1
Bits 3 to 0	Reserved	Undefined

COV_Increment

This property represents the minimum amount of change in Present_Value.

COV notifications are sent when the value of Present_Value for the COV notification that was last sent changes to a value that is greater than or equal to the value set with COV_Increment. COV_Increment cannot be read out to device. Set the initial value with WindLDR.

Priority_Array

Priority_Array is a read-only property representing the array that stores the priority values.

Of the 16 elements in the array (element 1 to element 16), the value stored in Priority_Array with the smallest index number out of all non-NULL values is used as the value of Present_Value. If all values stored in Priority_Array are NULL, Relinquish_Default is used as Present_Value.

Relinquish_Default

Relinquish_Default is the default value used as Present_Value when all values stored in Priority_Array are NULL.

Polarity

This property represents the relationship between the physical state of the input/output and the logical state indicated by Present_Value in the Binary Input and Binary Output objects.

Polarity	Physical State of Input/Output	Present_Value	Physical State of Device
NORMAL	OFF/ INACTIVE	INACTIVE	Not running
NORMAL	ON/ ACTIVE	ACTIVE	Running
REVERSE	ON/ ACTIVE	INACTIVE	Not running
REVERSE	OFF/ INACTIVE	ACTIVE	Running

Out_Of_Service

Out_Of_Service is the property that represents whether or not Present_Value and the physical input/output have been unbound.

If Out_Of_Service is TRUE, the object is out of service.

Properties	Value	Logical Value	Condition
Out_Of_Service	0	FALSE	In service (Present_Value and the physical input/output are bound.)
	1	TRUE	Out of service (Present_Value and the physical input/output are unbound.)

Out_Of_Service can be read out to an internal relay, and the status of an internal relay can be written as Out_Of_Service.

Properties	I	Q	M	R	T	C	D	Constant
Out_Of_Service	—	—	Yes*1	—	—	—	—	—

*1 Special internal relays cannot be used.

Note: Out_Of_Service = TRUE is used for simulations.

Reliability

This property represents the reliability of the object property.

The following table shows the definition of Reliability for each object type.
Yes: Valid, No: Invalid

Definition	Value	Analog Input	Analog Output	Analog Value	Binary Input	Binary Output	Binary Value
no-fault-detected	0	Yes	Yes	Yes	Yes	Yes	Yes
no-sensor	1	Yes	No	No	Yes	No	No
over-range	2	Yes	No	Yes	No	No	No
under-range	3	Yes	No	Yes	No	No	No
open-loop	4	Yes	Yes	No	Yes	Yes	No
shorted-loop	5	Yes	Yes	No	Yes	Yes	No
no-output	6	No	Yes	No	No	Yes	No
unreliable-other	7	Yes	Yes	Yes	Yes	Yes	Yes
process-error	8	No	No	No	No	No	No
multi-state-fault	9	No	No	No	No	No	No
configuration-error	10	No	No	No	No	No	No
-- enumeration value 11 is reserved for a future addendum	11	Yes	Yes	No	Yes	Yes	Yes
communication-failure	12	Yes	Yes	Yes	Yes	Yes	Yes
member-fault	13	No	No	No	No	No	No
monitored-object-fault	14	No	No	No	No	No	No
tripped	15	No	No	No	No	No	No

Reliability can be read out to a data register, and the value of a data register can be written as Reliability.

Properties	I	Q	M	R	T	C	D	Constant
Reliability	—	—	—	—	—	—	Yes*1	—

*1 Special data registers cannot be used.

System_Status

Indicates the physical status and logical status of the Plus CPU module.

Parameter	Value
OPERATIONAL	0
OPERATIONAL_READ_ONLY	1
DOWNLOAD_REQUIRED	2
DOWNLOAD_IN_PROGRESS	3
NON_OPERATIONAL	4
BACKUP_IN_PROGRESS	5

System_Status of the Plus CPU module is fixed as OPERATIONAL.

Firmware_Revision

This property is set with the system firmware version of the Plus CPU module.

Application_Software_Version

Set information about the application, such as the modification date of the created ladder program, as a fixed string with WindLDR.

Protocol_Services_Supported

This property represents the types of services that are supported by the Plus CPU module.

Protocol_Object_Types_Supported

This property represents the types of objects that are supported by the Plus CPU module.

Object_List

This property represents the list of created objects.

16: EtherNet/IP Communication

This chapter describes the EtherNet/IP communication function.

Overview

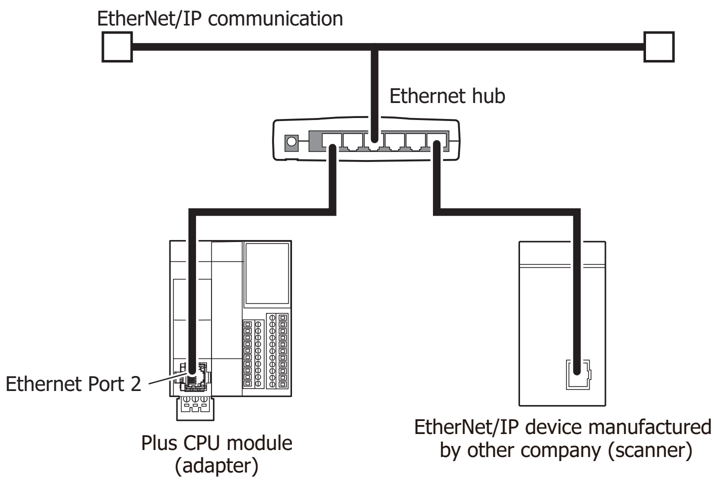
The Plus CPU module supports EtherNet/IP (scanner*1) communication.
EtherNet/IP stands for "Ethernet Industrial Protocol", which is an industrial multi-vendor network that uses Ethernet.
The Plus CPU module works as a scanner in EtherNet/IP communication and communication can be performed with devices that are compatible with EtherNet/IP communication.
EtherNet/IP communication also uses standard Ethernet technologies, which allows networks to be built that include various Ethernet-compatible devices. The Plus CPU module has an adapter*1 function in addition to a scanner function which also allows it to work as an adapter in EtherNet/IP communication.

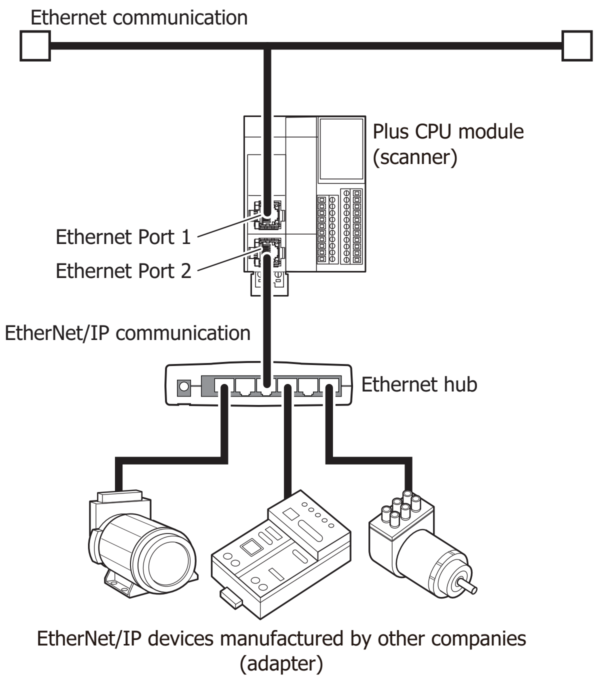
*1 In EtherNet/IP communication, the master side is called the scanner and the slave side is called the adapter. In general, an EtherNet/IP device that has a function to be given a request to open a CIP connection is called an adapter, and an EtherNet/IP device that also combines a function to give a request to open a CIP connection is called a scanner.

*2 In this manual, a device that is compatible with EtherNet/IP communication is subsequently called an EtherNet/IP device.

Multi-Vendor Network between EtherNet/IP Devices

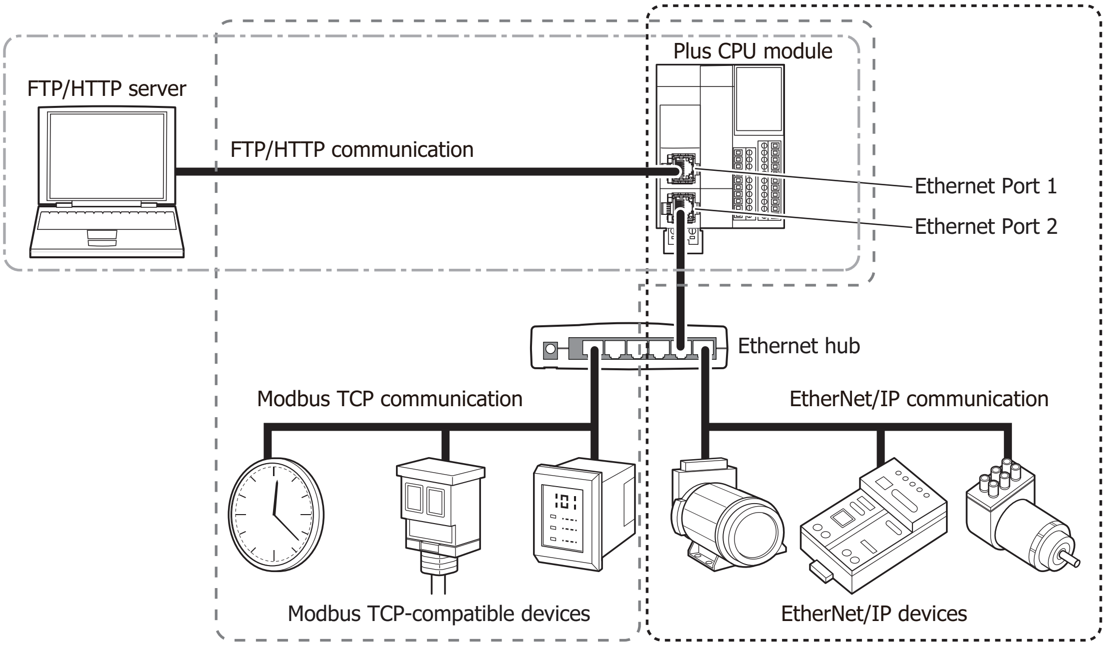
You can build a multi-vendor network between EtherNet/IP devices.





Network That Mixes General-Purpose Ethernet Devices

You can build a network that mixes EtherNet/IP devices and general-purpose Ethernet devices.



Note: When EtherNet/IP communication and other Ethernet communications are performed simultaneously on the same network, data may no longer be transmitted or received at the intended timing, depending on the load status. In this situation, build the network using an Ethernet switch that supports the QoS function and adjust the priority of data that flows on the network.

Communication Ports Used in EtherNet/IP

Supported models and communication ports are as follows.

Communication port	All-in-One CPU module			CAN J1939 All-in-One CPU module	Plus CPU module	
	16-I/O type	24-I/O Type	40-I/O Type		Plus 16-I/O type	Plus 32-I/O Type
Ethernet Port 1	—	—	—	—	—	—
Ethernet Port 2	—	—	—	—	Yes	Yes
HMI-Ethernet Port	—	—	—	—	—	—

EtherNet/IP Specifications

<table>
<tr><th colspan="3">Module</th><th>Plus CPU module</th></tr>
<tr><td colspan="3">Supported Port</td><td>Ethernet Port 2</td></tr>
<tr><td rowspan="10">I/O Message Communication Function</td><td colspan="2">Number of CIP connections</td><td>32*1</td></tr>
<tr><td rowspan="2">CIP connection points</td><td>Number of settings</td><td>8</td></tr>
<tr><td>Predefined</td><td>Instance ID 198 (for Input Only output)
Instance ID 199 (for Listen Only output)</td></tr>
<tr><td colspan="2">RPI (communication cycle)</td><td>10 to 10,000 ms (in 1 ms increments)</td></tr>
<tr><td colspan="2">CIP connection type</td><td>Input Only/Exclusive Owner/Listen Only</td></tr>
<tr><td colspan="2">Trigger of send</td><td>Cyclic/COS (change of state)*2</td></tr>
<tr><td colspan="2">Maximum data size per CIP connection</td><td>504 bytes or 1,444 bytes*3, *4</td></tr>
<tr><td rowspan="2">I/O message communication allowable bandwidth</td><td>504 bytes/packet</td><td>600pps</td></tr>
<tr><td>1,444 bytes/packet</td><td>200pps</td></tr>
<tr><td colspan="2">Multicast filter function</td><td>None</td></tr>
<tr><td rowspan="4">Explicit Message Communication Function</td><td rowspan="2">Class 3 (connection type)</td><td>Server</td><td>Number of CIP connections: 32*5</td></tr>
<tr><td>Client</td><td>Not supported</td></tr>
<tr><td rowspan="2">UCMM</td><td>Server</td><td>Number of simultaneous executions: 32</td></tr>
<tr><td>Client</td><td>Not supported</td></tr>
</table>

*1 The maximum number of connections is 32 in total with the CIP connections used in the class 3 (connection type) explicit message communication function.
*2 Communication can be performed with EtherNet/IP devices that output data with the Change Of State (transmit data when there is a change of state) method.
*3 For 1,444 bytes, the EtherNet/IP device must support Large Forward Open (CIP option specification).
*4 Make the total data size within 5,776 bytes for one CIP connection.
*5 The maximum number of connections is 32 in total with the CIP connections used in I/O message communication function.

EtherNet/IP

Supported Standards

EtherNet/IP communication in the Plus CPU module supports the following standards.

- EtherNet/IP

Vendor ID

The vendor ID is 159.

Device Profile

EtherNet/IP communication in the Plus CPU module supports the scanner.

Objects

EtherNet/IP communication in the Plus CPU module supports the following objects.

Object Name	Class Code
Identity	01H
Message Router	02H
Assembly	04H
Connection Manager	06H
TCP/IP Interface	F5H
Ethernet Link	F6H

For details on objects, see "Objects" on page 16-37.

Function

The Plus CPU module provides the following functions as an EtherNet/IP device scanner.

- I/O message communication (originator/target) function (page 16-5)
- Explicit message communication (server) function (page 16-10)
- Data and device binding function (page 16-11)

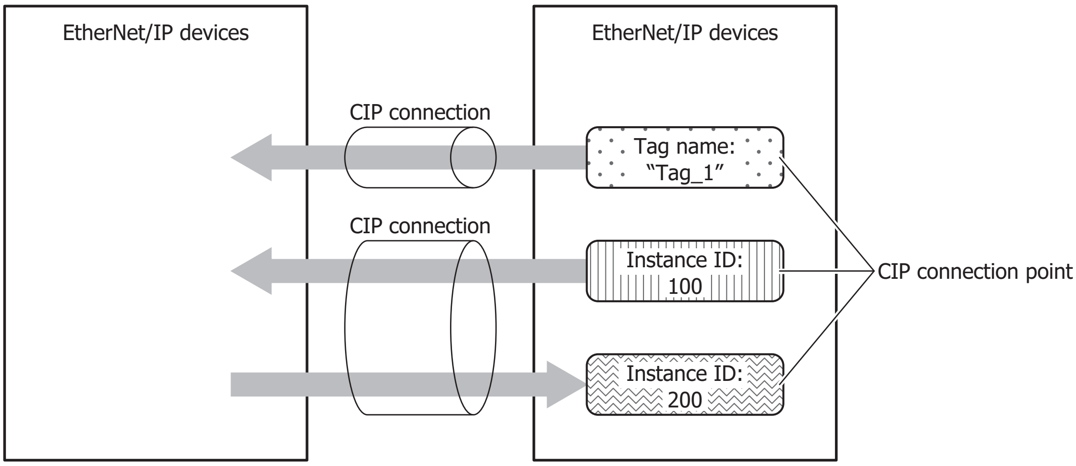
I/O Message Communication Function

This section describes the I/O message communication function and how to use it.

■ What Is I/O Message Communication?

I/O message communication is a function that allows EtherNet/IP devices to cyclically read and write data with each other. The target unit for reading/writing data is called a CIP connection point. CIP tag names and instance IDs are assigned to CIP connection points. In I/O message communication, an EtherNet/IP device reads/writes data by specifying the CIP tag name or instance ID of a CIP connection point.

Conceptual Diagram of I/O Message Communication



■ What Is a CIP Connection?

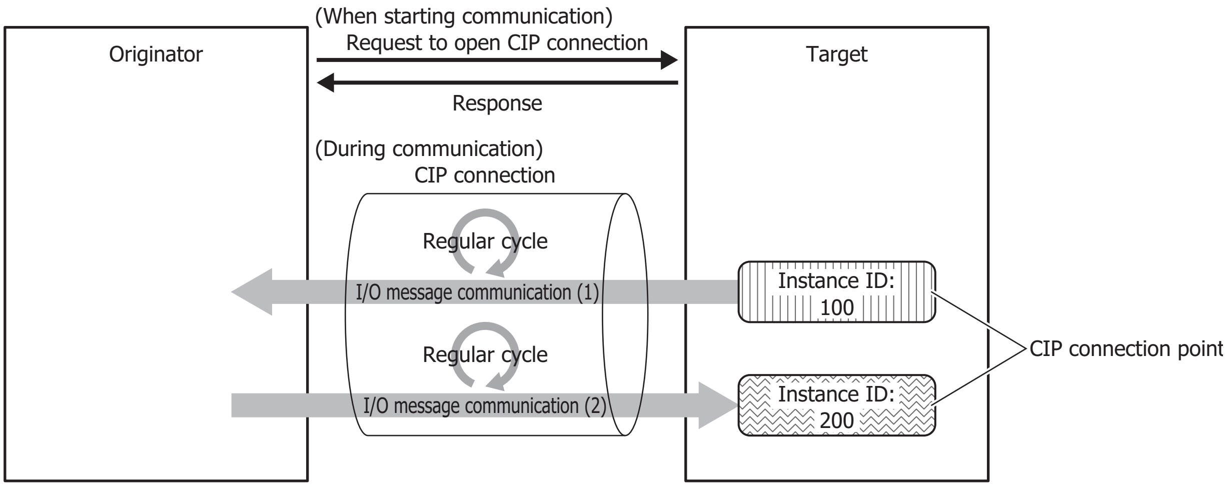
A CIP connection is a virtual communication line between EtherNet/IP devices. In I/O message communication, one device sends a request to open a CIP connection to the CIP connection point of the other device, and when successful, data is read and written. The side that sends the request to open the CIP connection is called the originator. The side that receives the request is called the target. The Plus CPU module can be the originator and the target. The Plus CPU module can open 32 CIP connections at the same time.

■ **Types of CIP Connections**

There are the following three types of CIP connections. Exclusive Owner can transmit and receive data. Input Only and Listen Only can only receive data.

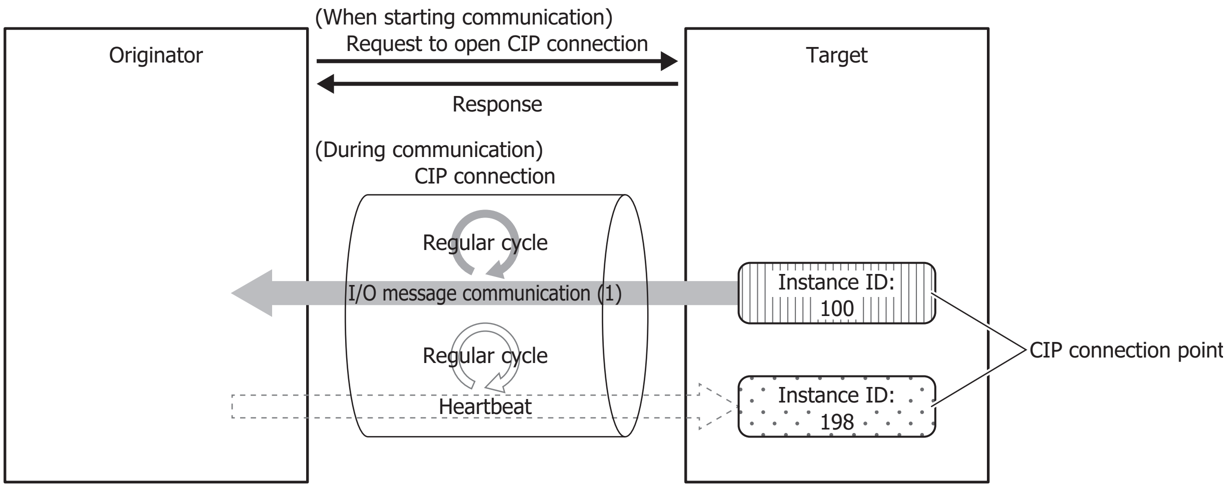
(1) Exclusive Owner

This CIP connection cyclically transmits data from the target to the originator (1) and cyclically transmits data from the originator to the target (2). The CIP connection point is specified as a CIP tag name or an instance ID. Communication is performed at the cycle (RPI) specified when the CIP connection is opened. Data can be transmitted and received when the originator opens the connection.



(2) Input Only

This CIP connection cyclically transmits data from the target to the originator (1). The CIP connection point is specified as a CIP tag name or an instance ID. Communication is performed at the cycle (RPI) specified when the CIP connection is opened. In order for the target to perform alive monitoring of the originator, a heartbeat that does not include data is cyclically transmitted from the originator to the target. The originator specifies the Input Only instance ID (198) on the target as the destination for the heartbeat. The heartbeat is transmitted at the specified cycle (RPI).

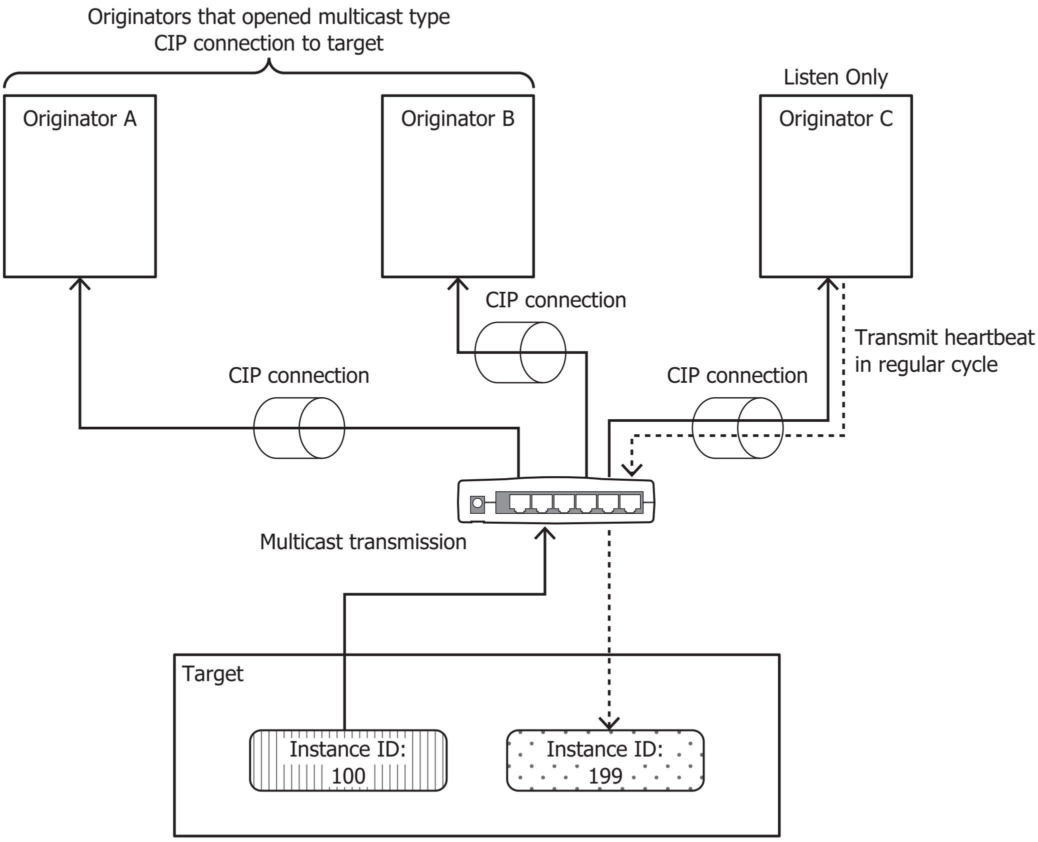


Notes:

- For Input Only, when setting the CIP connection point of the input data to receive from the target as an instance ID, set the CIP connection point of the heartbeat to transmit to the target as an instance ID too.
- For Input Only, when setting the CIP connection point of the input data to receive from the target as a CIP tag name, a CIP tag name of the CIP connection point of the output data to transmit to the target does not exist and is not set.
- The heartbeat is always performed in Input Only.

(3) Listen Only

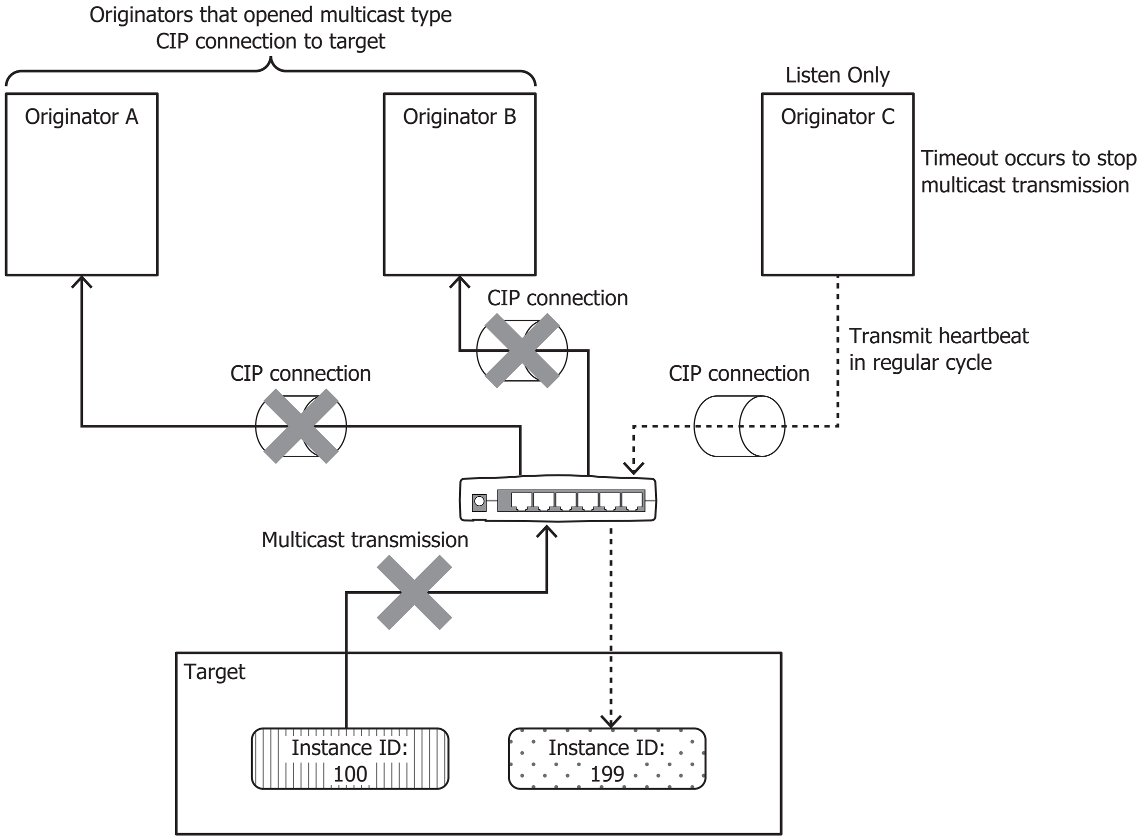
This CIP connection cyclically receives data for originators from a target that already has a multicast type CIP connection using Exclusive Owner or Input Only opened. Set this connection type to simultaneously receive data when the target transmits that data by multicast to other originators. The CIP connection point is specified as an instance ID. Communication is performed at the cycle (RPI) specified when the CIP connection is opened. In order for the target to perform alive monitoring of the originator, a heartbeat that does not include data is cyclically transmitted from the originator to the target. The originator specifies the Listen Only instance ID (199) on the target as the destination for the heartbeat. The heartbeat is transmitted at the specified cycle (RPI).



Notes:

- For Listen Only, when setting the CIP connection point of the input data to receive from the target as an instance ID, set the CIP connection point of the output data to transmit to the target as an instance ID too.
- The heartbeat is always performed in Listen Only.

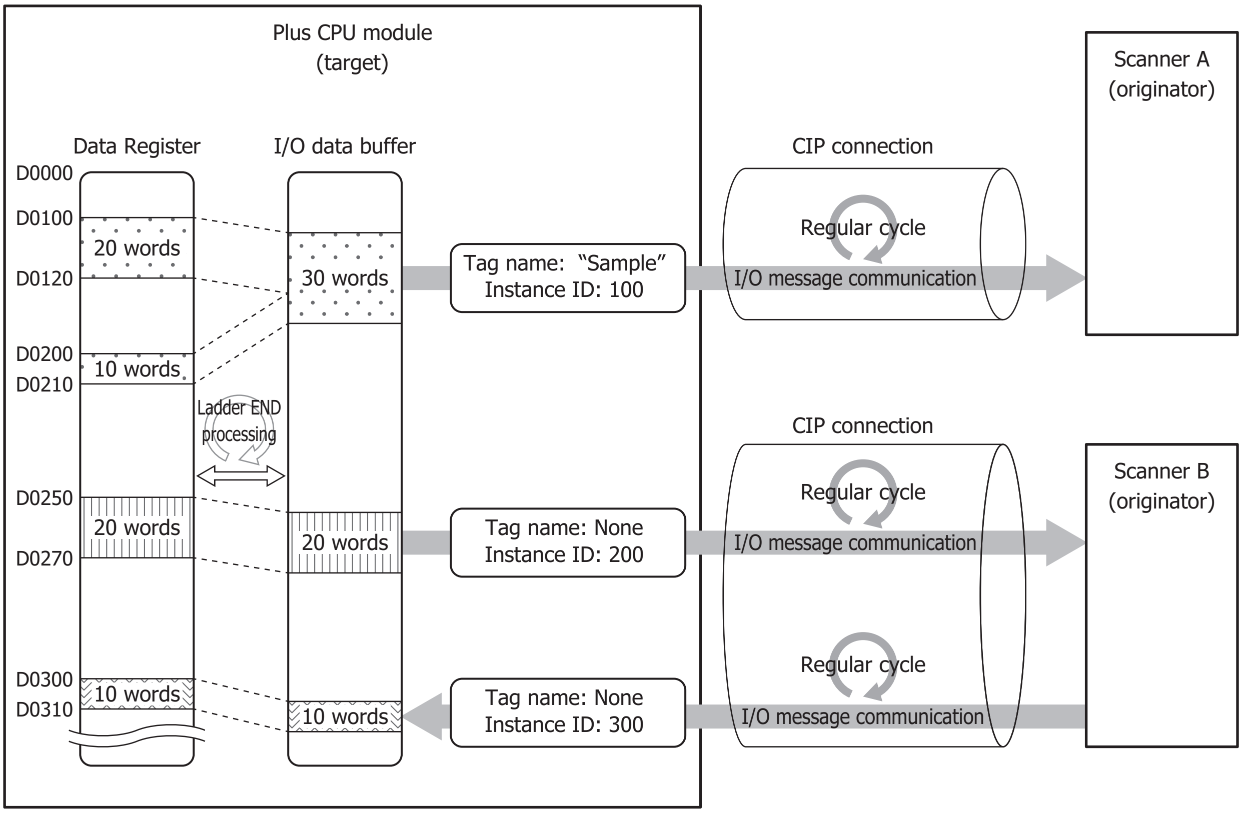
A Listen Only CIP connection can be opened only when a multicast type CIP connection is opened. When the CIP connection between the target and originator that opened a multicast type CIP connection using Exclusive Owner or Input Only is disconnected, the multicast transmissions from the target to originators that opened Listen Only CIP connections are stopped.



I/O Message Communication (Target) Function

A function where the Plus CPU module is the target with CIP connection points and originators cyclically read and write data. A maximum of 8 CIP connection points can be defined.

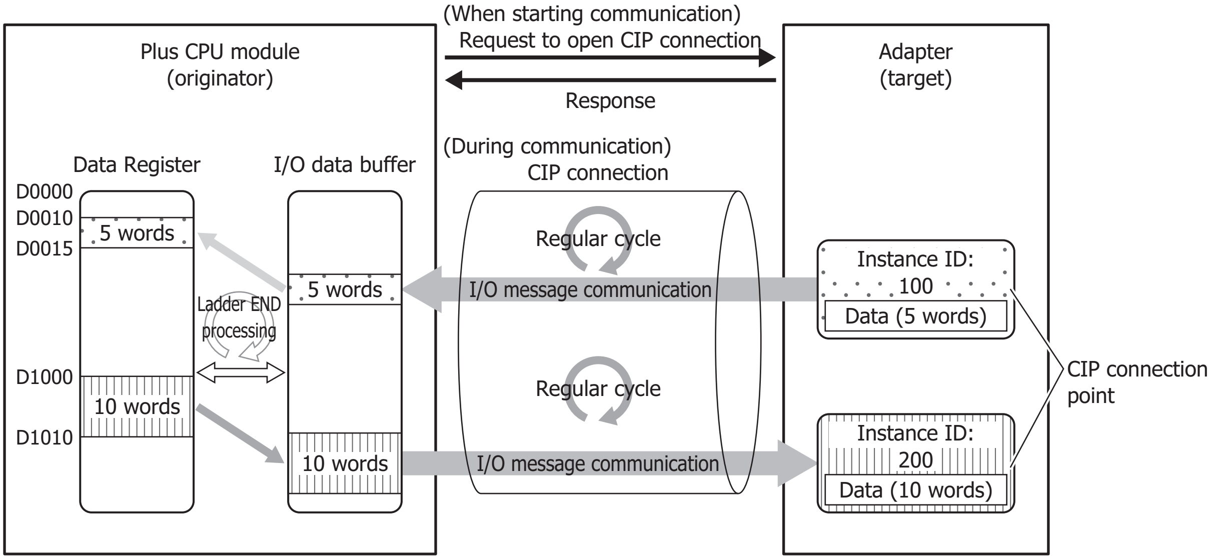
The originator sends a request to open a CIP connection to the CIP connection point of the Plus CPU module, and when successful, the CIP connection point data is read and written. The Plus CPU module first writes the values of data registers to the I/O data buffer*1 and then transmits the data to the originator. Data received from the originator is also first written to the I/O data buffer and then stored in data registers.



*1 The I/O data buffer is a buffer inside the Plus CPU module that stores the data to transmit and receive in I/O message communication. For details on the I/O data buffer and data register binding, see "Data and Device Binding Function" on page 16-11.

I/O Message Communication (Originator) Function

A function where the Plus CPU module cyclically reads and writes data of the CIP connection points of targets. When the EtherNet/IP Communication Bit (M8460) is turned on, the Plus CPU module sends a request to open the CIP connections to the CIP connection point of targets registered to the scan list*1. When successful*2, the CIP connection point data is read and written. The Plus CPU module first writes the data received from the target to the I/O data buffer*3 inside the Plus CPU module and then stores the data in data registers. The values of data registers are also first written to the I/O data buffer and then transmitted to the target.



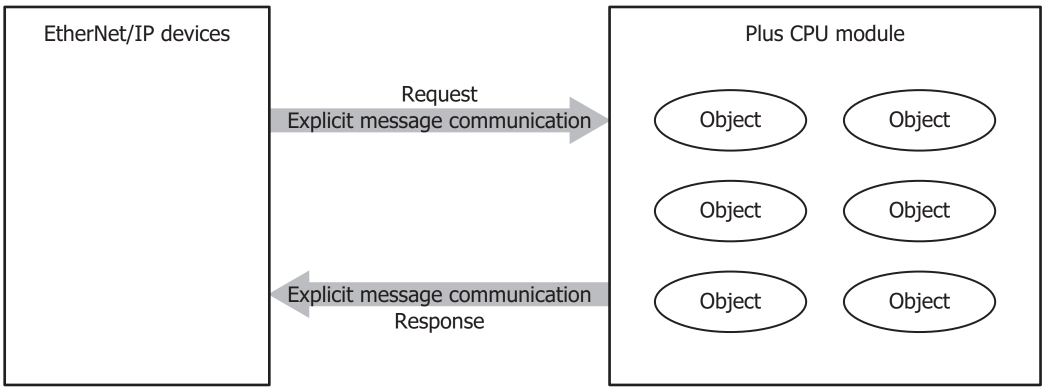
*1 The scan list is a list of CIP connections the Plus CPU module opens as the originator. The list of configured targets or CIP connections is displayed in the EtherNet/IP tree area on the **EtherNet/IP Settings** dialog box. For details, see "EtherNet/IP Tree Area" on page 16-20.

*2 If opening a CIP connection fails, opening the CIP connection continues being retried until successful. During I/O message communication, if I/O messages from the target cannot be received even after the specified timeout elapses, a new CIP connection is automatically reopened. If opening the new CIP connection fails, opening the CIP connection continues being retried until successful.

*3 The I/O data buffer is a buffer inside the Plus CPU module that stores the data to transmit and receive in I/O message communication. For details on the I/O data buffer and data register binding, see "Data and Device Binding Function" on page 16-11.

Explicit Message Communication (Server) Function

This function allows the services that correspond to Plus CPU module objects*1 to be executed from other EtherNet/IP devices.

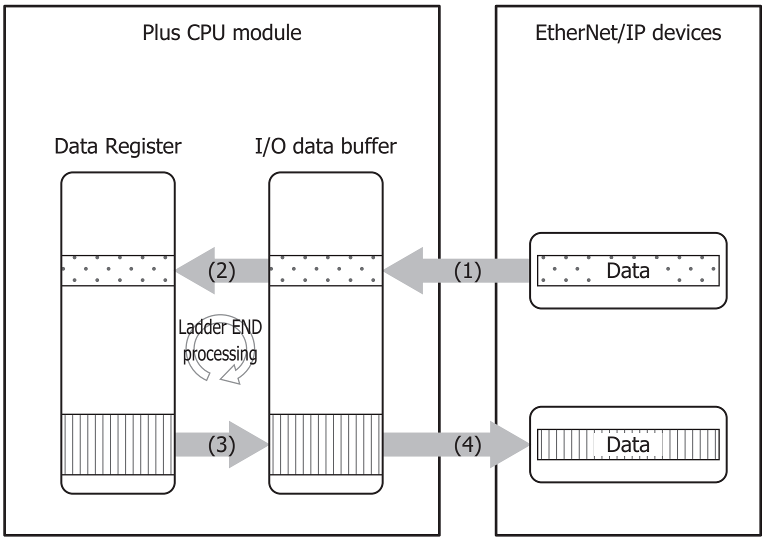


*1 Objects are the abstraction of functions and data of EtherNet/IP devices.

Data and Device Binding Function

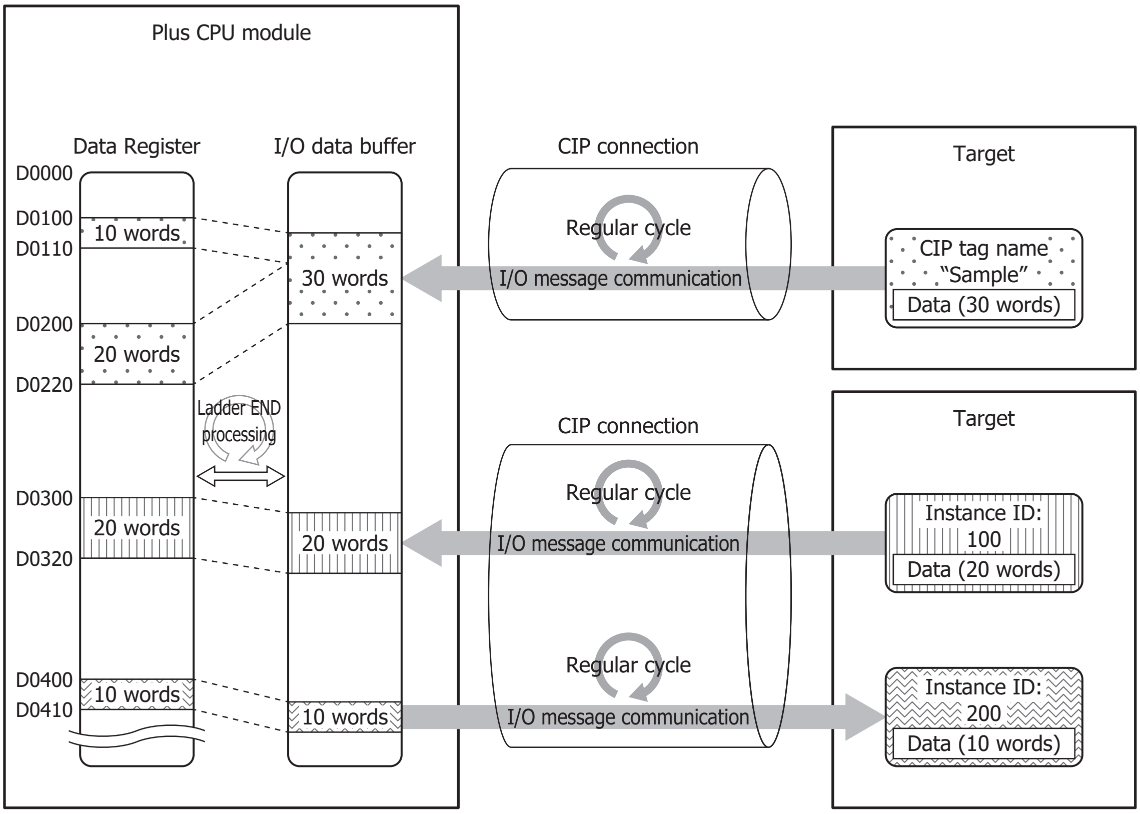
The Plus CPU module assigns IN data[*1] and OUT data[*2] set on the scan list and in CIP connection points to the I/O data buffer in the Plus CPU module.

The Plus CPU module first writes the data received from the other device in I/O message communications to the I/O data buffer (1) and then stores the data in data registers (2). The values stored in data registers are also first written to the I/O data buffer (3) and then transmitted to the other device (4). The processing to reflect data between the I/O data buffer and data registers is performed in the END processing of the ladder program.



*1 IN data is received by the originator from the target in I/O message communication.
*2 OUT data is transmitted to the target from the originator in I/O message communication.

For example, to have the Plus CPU module (originator) cyclically read and write the CIP connection point data of the target, the data received from the target is first written to the I/O data buffer and then stored in the data registers. The values of data registers are also first written to the I/O data buffer and then transmitted to the target.



■ **Refresh Limit Count**

The size of data that can be reflected between the I/O data buffer and data registers in END processing one time is a maximum of 400 words. This size can be set as the refresh limit count between 1 and 400 words. For example, if the refresh limit count is set to 400 words, a maximum of 256 words of data are reflected from the I/O data buffer to data registers and a maximum of 256 words of data are reflected from data registers to the I/O data buffer in END processing one time. When IN data and OUT data are set that exceeded the refresh limit count, the processing is split into multiple portions of END processing. When the refresh limit count is increased, the amount of data that is reflected in the END processing increases, so the scan time of the ladder program becomes longer. Adjust the value according to the target system.

Note: Normally, I/O message communication is performed more frequently. If the values received from an EtherNet/IP device change in an amount of time shorter than the scan time, those changes may not be reflected in the data registers.

EtherNet/IP Communication Operations

Special Devices Used in EtherNet/IP Communication

These tables show the special data registers and special internal relay used in EtherNet/IP communication.

Special Data Register	Description		R/W
D8790	EtherNet/IP Operation Status	This special data register stores the operating status of EtherNet/IP communication. 0000h: Stopped 0100h: Preparing to go online 0200h: Online 0300h: Restarting 0F00h: Stopped by error	R
D8791	EtherNet/IP Error Information	This special data register stores error information that has occurred in EtherNet/IP communication. 0000h: Normal 0800h: Timeout waiting for link 8000h: Timeout waiting for communication to start 8100h: Timeout waiting for communication to stop This special data register is reset to 0 when EtherNet/IP communication is preparing to go online ((D8790)=0100h). If multiple errors occur in EtherNet/IP communication, the error information for the error that last occurred is stored.	R

Special internal relay	Description		R/W
M8460	EtherNet/IP Communication Bit	This special internal relay permits or prohibits EtherNet/IP communication. OFF: Prohibit EtherNet/IP communication ON: Permit EtherNet/IP communication If M8460 is changed from off to on during the RUN status, the Plus CPU module establishes CIP connections with the other devices and starts EtherNet/IP communication. If M8460 is turned off, the CIP connections are disconnected and EtherNet/IP communication is stopped. When changing from run to stop, M8460 is automatically turned off.	R/W

Notes:

- To resume EtherNet/IP communication when EtherNet/IP Operation Status (D8790) is stopped by an error, first turn off M8460 and wait for EtherNet/IP Operation Status (D8790) to become stopped, and then turn on M8460.
- When the Plus CPU module power is on and when starting to run, the EtherNet/IP Communication Bit (M8460) is off. To start EtherNet/IP communication, turn on M8460 in the ladder program or download and run a user program in which the **Turn ON EtherNet/IP Communication bit automatically** check box has been selected on the **EtherNet/IP Settings** dialog box.

Basic Operation

This chapter describes the basic operation of EtherNet/IP communication.

■Starting and Stopping EtherNet/IP Communication

When the EtherNet/IP Communication Bit (M8460) is changed from off to on during the RUN status, the Plus CPU module establishes CIP connections to the targets or CIP connection points registered to the scan list and starts EtherNet/IP communication (starts cyclic data exchange by I/O message communication).

When the EtherNet/IP Communication Bit (M8460) is turned off, EtherNet/IP communication stops (I/O message communication stops and all CIP connections are disconnected). When changing from run to stop, EtherNet/IP Communication Bit (M8460) is automatically turned off.

■Operation during EtherNet/IP Communication

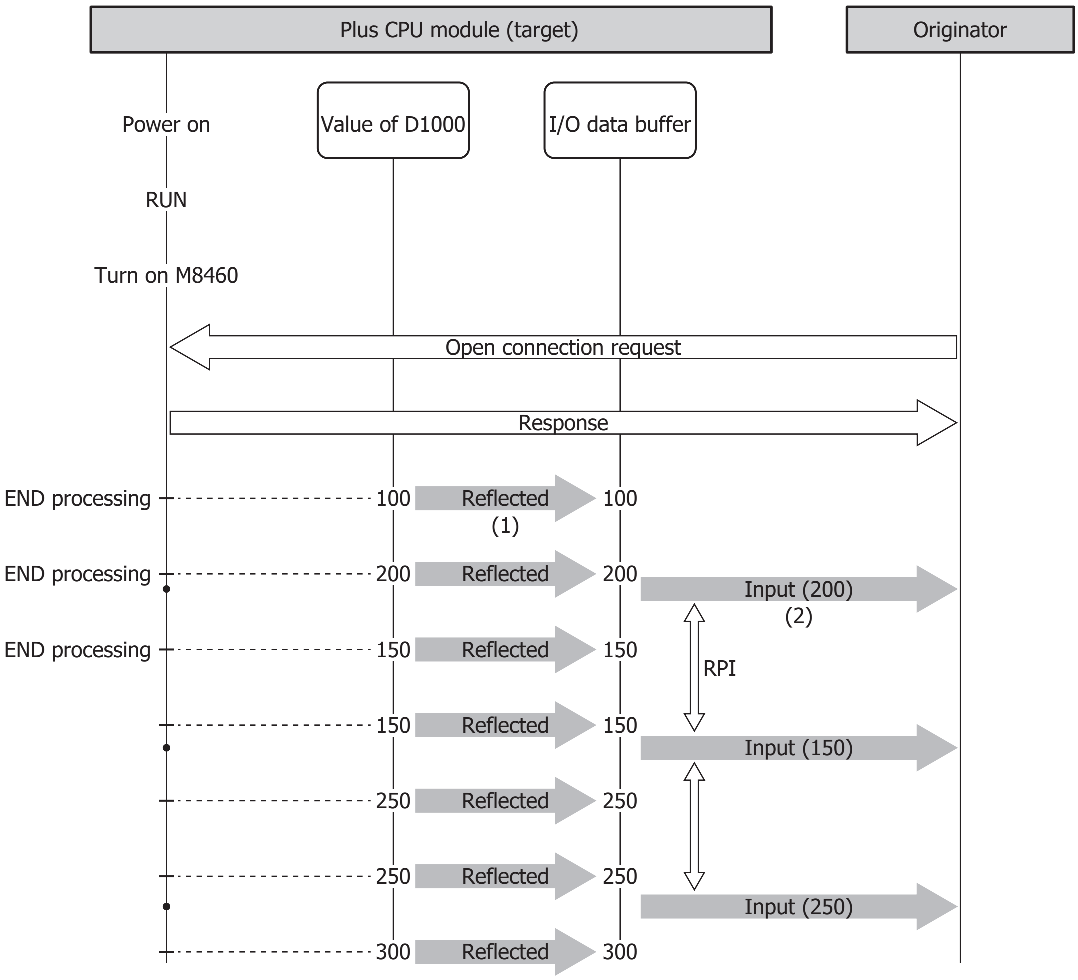
The I/O data buffer held in the Plus CPU module corresponds with data registers and data is reflected between the two in the END processing of the ladder program.

The processing to reflect data between the I/O data buffer and data registers and I/O message communication between the Plus CPU module and EtherNet/IP device is not synchronized.

Note: Normally, I/O message communication is performed more frequently, so if the values received from an EtherNet/IP device change in an amount of time shorter than the scan time, those changes may not be reflected in the data registers.

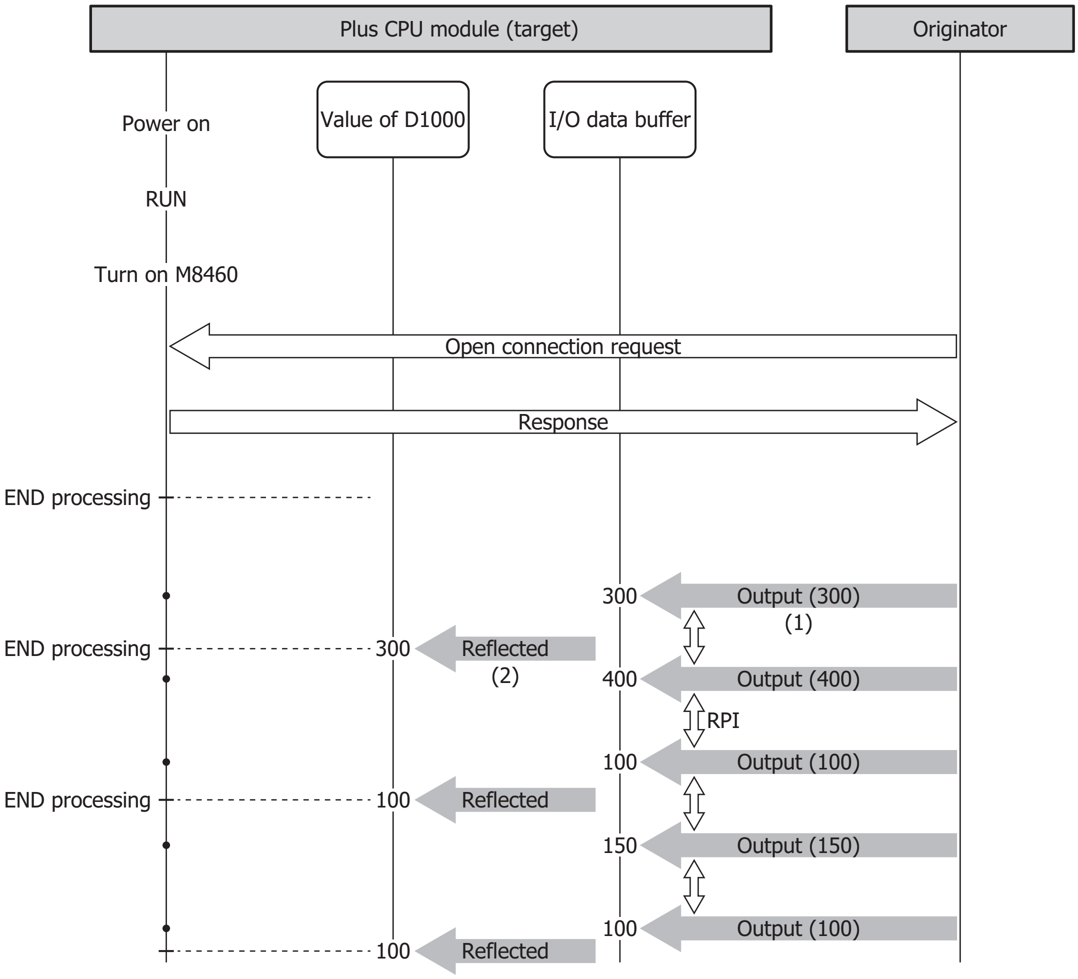
When the Plus CPU module (Target) Transmits Data to the Originator (Input)

For example, when the value stored in D1000 is transmitted to the originator, the value stored in D1000 is reflected to the I/O data buffer in the END processing of the ladder program immediately before the data is transmitted (1). Then the data in the I/O data buffer is transmitted to the target in I/O message communication immediately afterward (2).



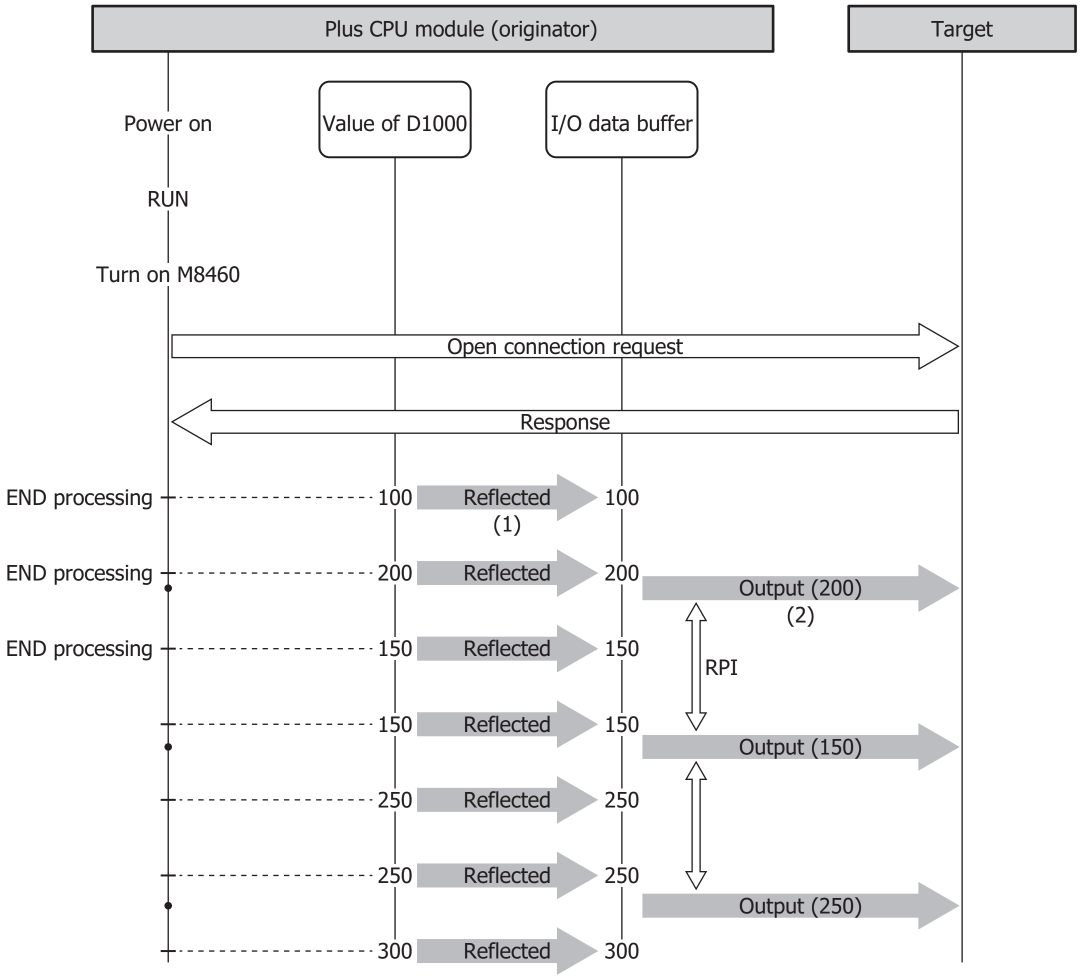
When the Plus CPU module (Target) Receives Data from the Originator (Output)

For example, when the data received from the originator is stored in D1000, the Plus CPU module writes the data received from the originator in I/O message communication to the I/O data buffer (1). Then the data in the I/O data buffer is reflected to the data registers in the END processing of the ladder program immediately after the data was received (2).



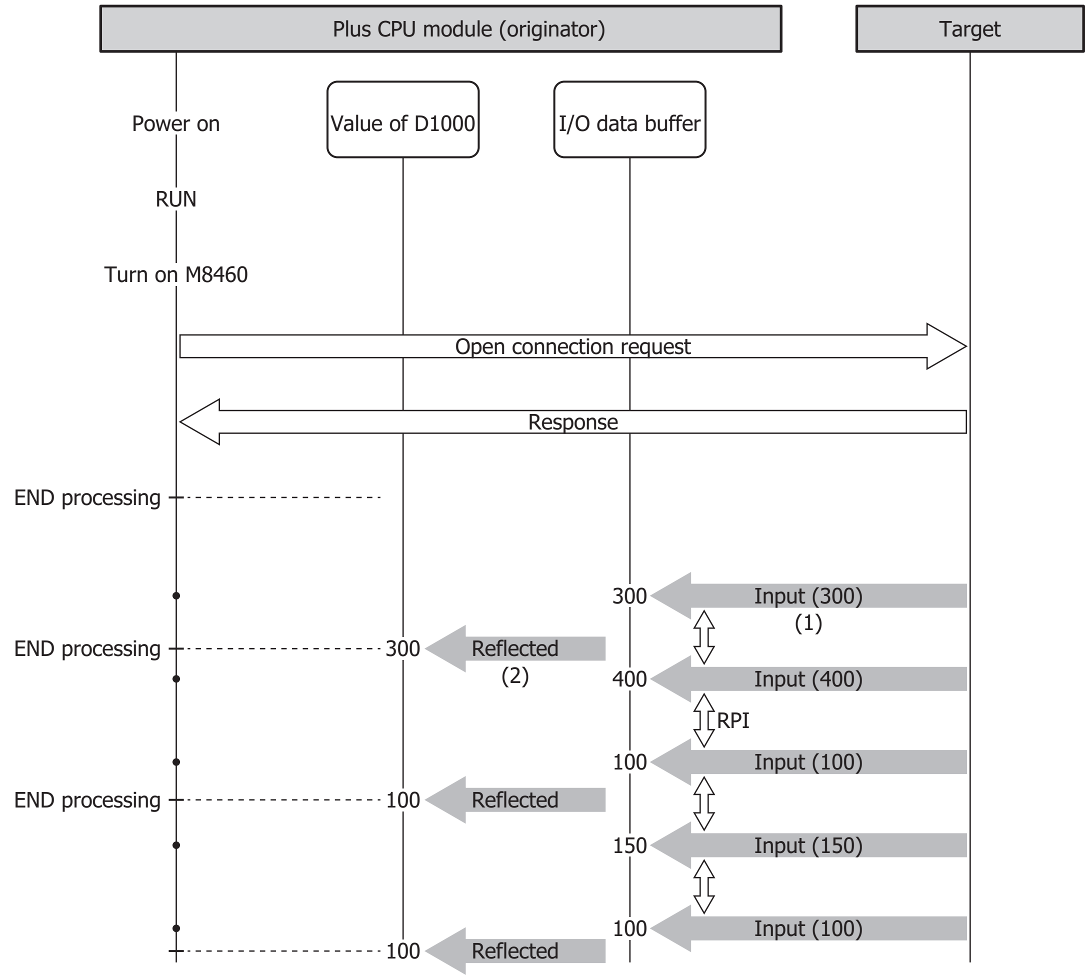
When the Plus CPU module (Originator) Transmits Data from the Target (Output)

For example, when the value stored in D1000 is transmitted to the target, the value stored in D1000 is reflected to the I/O data buffer in the END processing of the ladder program immediately before the data is transmitted (1). Then the data in the I/O data buffer is transmitted to the target in I/O message communication immediately afterward (2).



When the Plus CPU module (Originator) Receives Data from the Target (Input)

For example, when the data received from the target is stored in D1000, the Plus CPU module writes the data received from the target in I/O message communication to the I/O data buffer (1). Then the data in the I/O data buffer is reflected to the data registers in the END processing of the ladder program immediately after the data was received (2).



Operations during RUN and STOP

This table shows the status of EtherNet/IP communication during RUN and STOP.

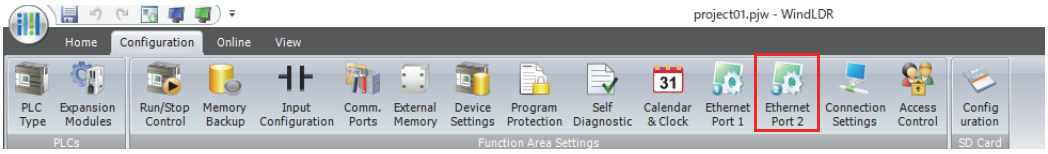
RUN/STOP Status	M8460 Status	EtherNet/IP Communication
STOP	OFF	Stopped
	ON	
RUN	OFF	
	OFF -> ON	Start
	ON	Communicating
	ON -> OFF	End

EtherNet/IP Communication Settings

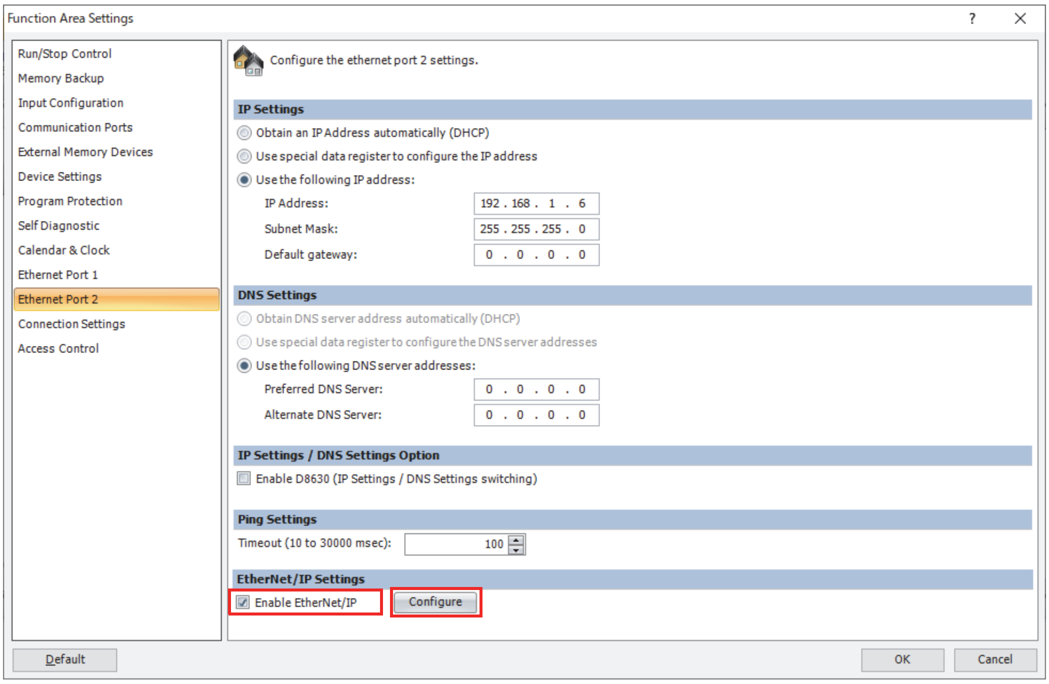
The EtherNet/IP communication settings are configured on the **EtherNet/IP Settings** dialog box. This section describes the **EtherNet/IP Settings** tab and the settings according to usage method.

● Procedure to display the EtherNet/IP Settings dialog box

1. On the **Configuration** tab, in the **Function Area Settings** group, click **Ethernet Port 2**.
The **Function Area Settings** dialog box is displayed.



2. Under EtherNet/IP settings, select the **Enable EtherNet/IP** check box and click **Configure**.
The **EtherNet/IP Settings** dialog box is displayed.



Overview of EtherNet/IP Settings

The settings required to perform EtherNet/IP communication and the related settings dialog boxes in WindLDR are as follows.

Description	Required Settings	WindLDR Settings Dialog Box	Available Functions
Perform EtherNet/IP communication	Basic settings for EtherNet/IP communication	EtherNet/IP settings	Common to EtherNet/IP communication
Using the Plus CPU module as a target	CIP connection point settings • CIP tag name or instance ID settings • Device allocations	CIP connection point settings	I/O message communication (target) function
Using the Plus CPU module as an originator	Create scan list • Target settings • CIP connection settings	Target settings CIP connection settings	I/O message communication (originator) function

EtherNet/IP Settings Dialog Box

The **EtherNet/IP Settings** dialog box is composed of the following three areas.

(1) EtherNet/IP Tree Area (page 16-20)

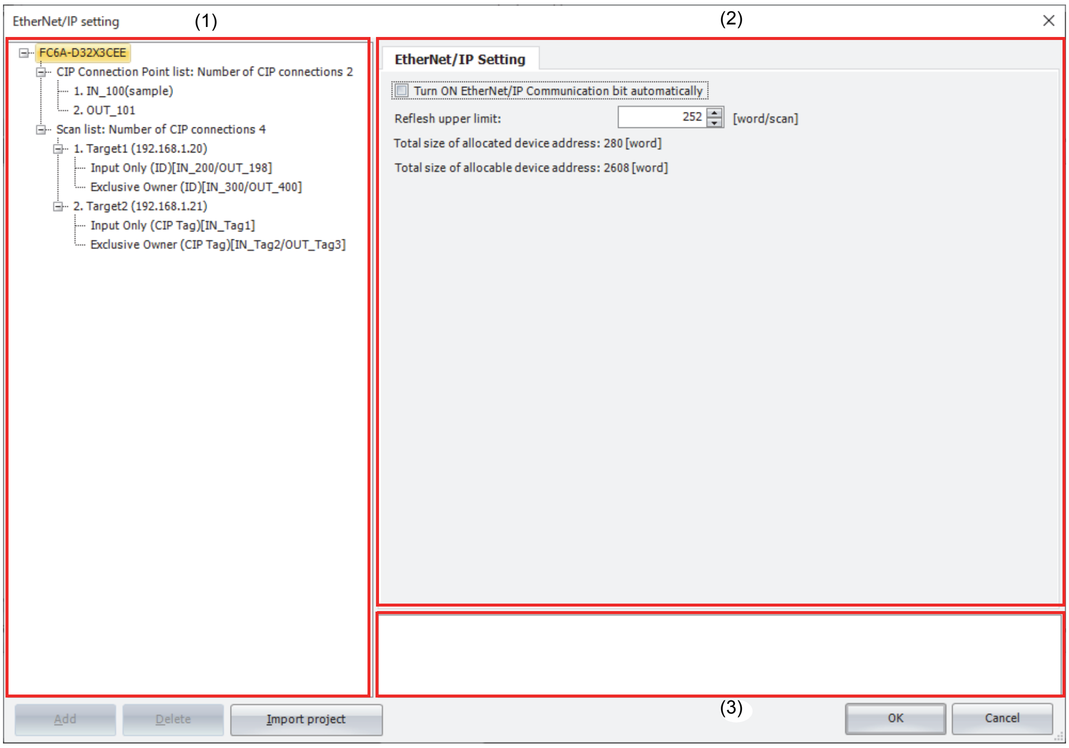
This area displays the list of configured CIP connection points, targets, and CIP connections.

(2) Parameter Setting Area (page 16-21)

This area displays details on the node selected in the EtherNet/IP tree area.

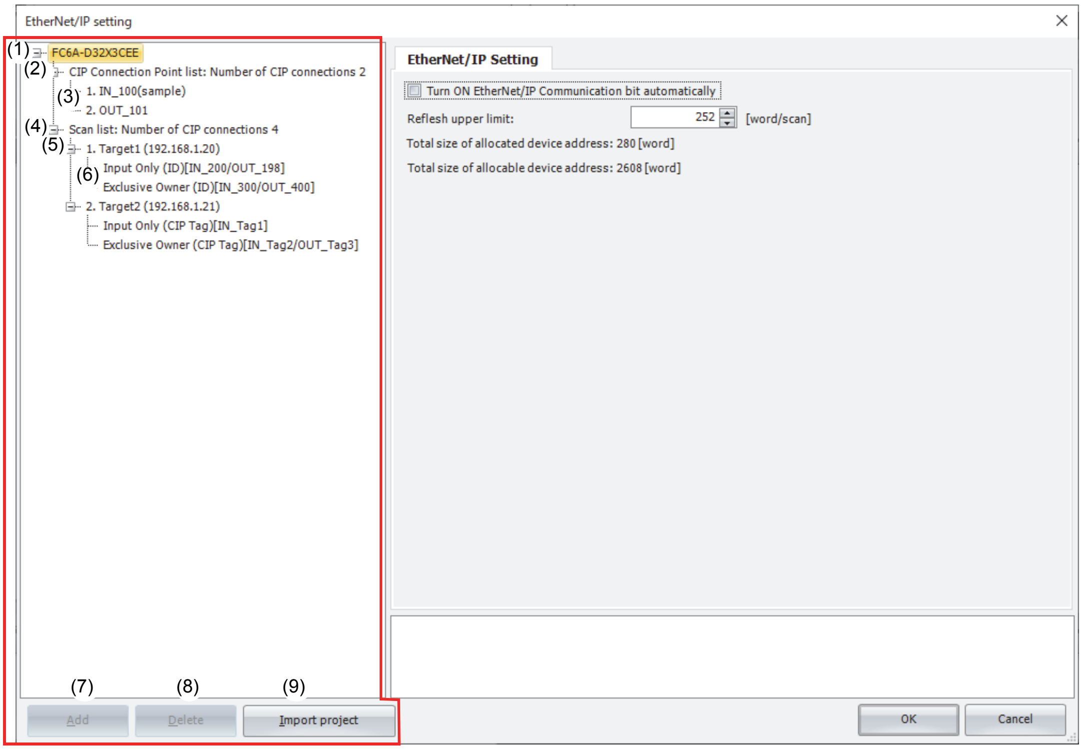
(3) Information View Area (page 16-29)

This area displays the details of errors if there are errors in content set on the **EtherNet/IP Settings** dialog box.



EtherNet/IP Tree Area

This area displays the list of configured CIP connection points, targets, and CIP connections.



(1) PLC Type node

Displays the PLC type name set on the **PLC Selection** dialog box. When the **PLC Type** node is selected and expanded, the list of configured CIP connection points and the scan list are displayed.

(2) CIP Connection Point List node

Displays the number of CIP connection points. When the **CIP Connection Point List** node is expanded, the list of configured CIP connection points is displayed.

(3) CIP Connection Point node

Displays the CIP tag name and instance ID of a configured CIP connection point.

(4) Scan List node

Displays the number of CIP connections. When the **Scan List** node is expanded, the list of configured targets is displayed.

(5) Target node

Displays the node name and IP address of a configured target. When a **Target** node is selected and expanded, the list of configured CIP connections is displayed.

(6) CIP Connection node

Displays the CIP connection name and connection point of a configured CIP connection.

(7) Add button

Adds a new node according to the node selected in the EtherNet/IP tree area. You can also add a node from the menu displayed when right-clicking the node.

(8) Delete button

Deletes the node selected in the EtherNet/IP tree area. You can also delete a node from the menu displayed when right-clicking the node.

(9) Import project button

Imports only the EtherNet/IP settings from a WindLDR project file.

Parameter Setting Area

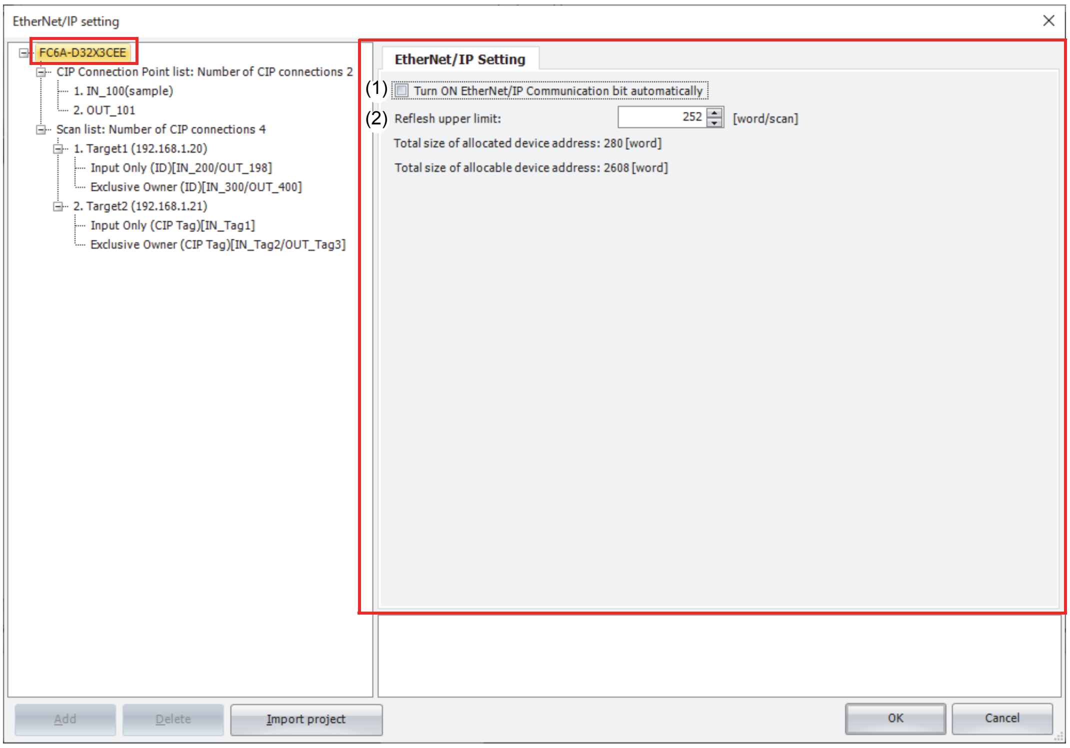
This area displays details on the node selected in the EtherNet/IP tree area.

When the PLC Type node is selected

Select the **PLC Type** node in the EtherNet/IP tree area. The **EtherNet/IP Setting** tab is displayed in the parameter settings area.

■EtherNet/IP Setting tab

Configure the basic settings for performing EtherNet/IP communication.



(1) Turn ON EtherNet/IP Communication bit automatically

Sets whether to automatically turn on the EtherNet/IP Communication Bit (M8460) when changing from stop to run. The check box is cleared by default. When the check box is cleared, the EtherNet/IP Communication Bit does not turn on automatically.

Setting Method	Description
Selected	The EtherNet/IP Communication Bit (M8460) turns on automatically when changing from stop to run. When EtherNet/IP Communication Bit (M8460) is turned on, the Plus CPU module performs EtherNet/IP communication with the other devices according to the content registered on the scan list.
Cleared	The EtherNet/IP Communication Bit (M8460) does not turn on automatically.

(2) Refresh upper limit

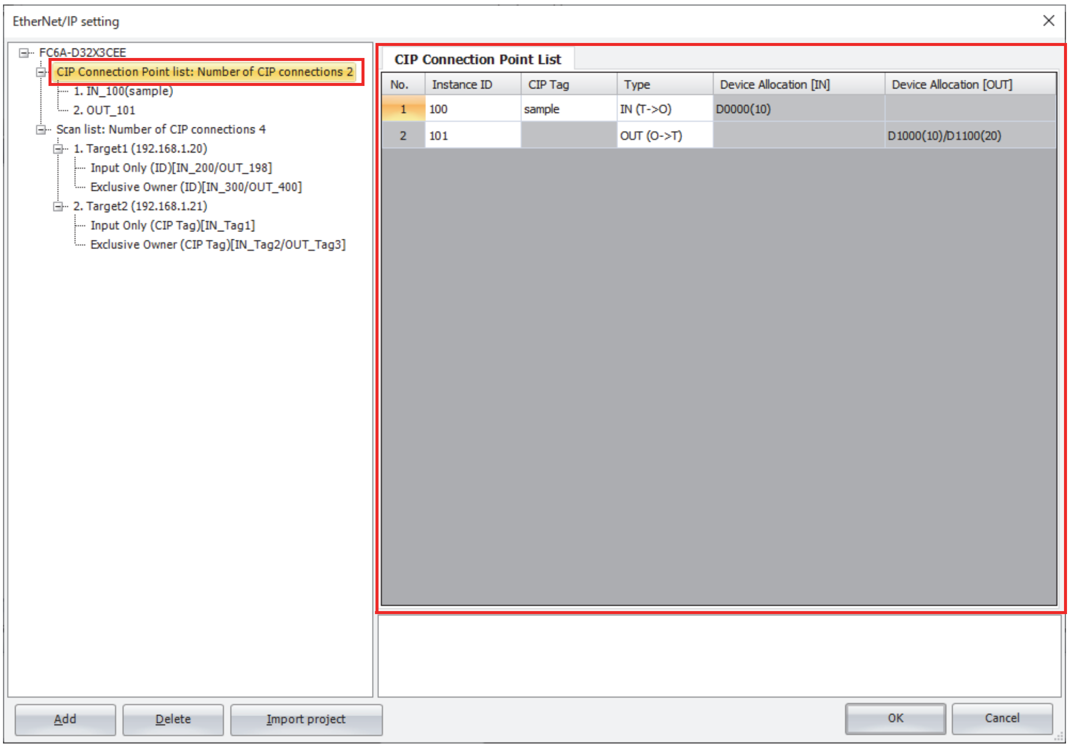
Sets the maximum size of data that can be reflected between the I/O data buffer and data registers in END processing one time. The refresh upper limit is set between 1 and 400 words per scan. The default value is 252 words per scan. The maximum size of data to write from the I/O data buffer to the data registers and the maximum size of data to write from the data registers to the I/O data buffer in END processing one time are half of the refresh upper limit.

When the CIP Connection Point List node is selected

Select the **CIP Connection Point List** node in the EtherNet/IP tree area. The **CIP Connection Point List** tab is displayed in the parameter settings area.

■CIP Connection Point List tab

This tab displays the list of configured CIP connection points. The CIP tag, instance ID, and type can be changed. Double-click a device allocation cell. The **CIP Connection Point List** tab is displayed and you can move to the relevant cell for device allocation.

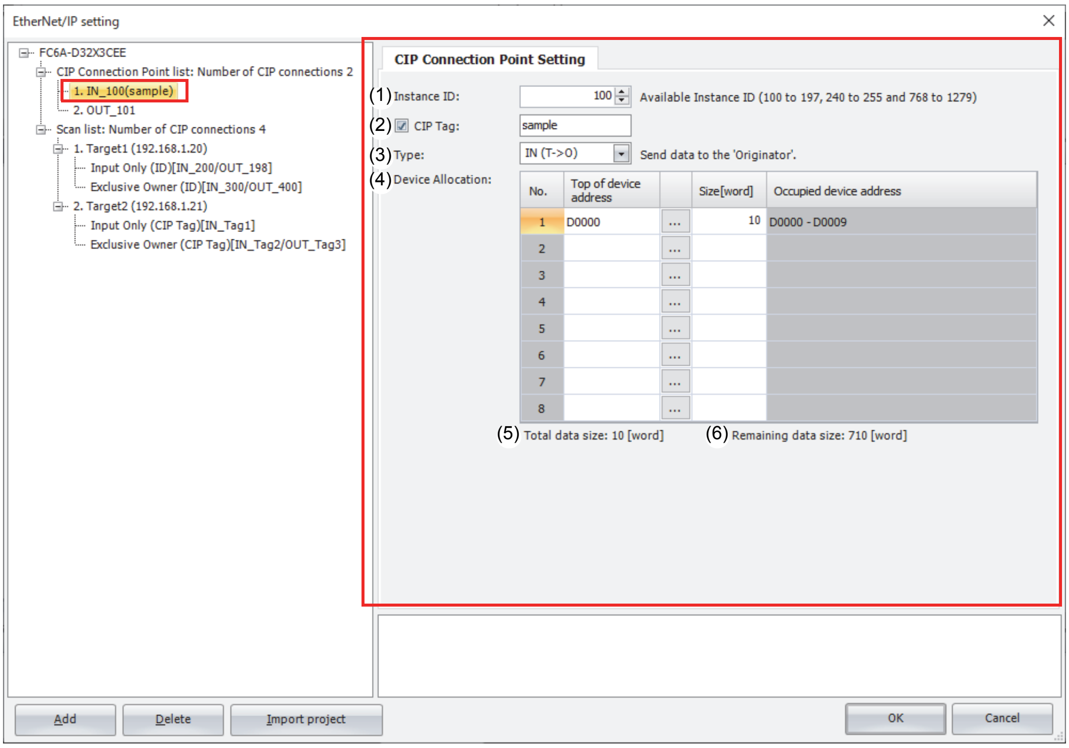


When the CIP Connection Point node is selected

Select a **CIP Connection Point** node in the EtherNet/IP tree area. The **CIP Connection Point Setting** tab is displayed in the parameter settings area.

■CIP Connection Point Setting tab

Configure the CIP connection point on this tab.



(1) Instance ID

Sets the instance ID. When the Plus CPU module (target) performs I/O message communication with another scanner device (originator) that cannot handle CIP tag names, the originator sets the CIP connection for this instance ID. Set the instance ID between 100 and 1,279. The default value is 100. Set this value so that it is not duplicated by other CIP connections.

Note: Certain instance IDs cannot be used between 100 and 1,279. Set the value to one of the available values displayed in WindLDR.

(2) CIP Tag

Sets the CIP tag name. Up to 64 bytes can be set. The other scanner device (originator) sets the CIP connection for this tag name and performs I/O message communication with the Plus CPU module (target). Select this check box and a CIP tag can be set.

(3) Type

Sets the direction of the data to handle in I/O message communication. This item can be set from the following two types. The default value is "IN (T->O)".

Setting	Description
IN (T->O)	The target (Plus CPU module) transmits data to the originator.
OUT (O->T)	The target (Plus CPU module) receives data from the originator.

(4) Device Allocation

Sets the data registers that store the data to receive or transmit in I/O message communication. Starting from the set data register, data registers are used in only the amount set by the data size. Set the data size between 1 and 720. The default value is 1. Set the first data register so that the device range is not exceeded.

(5) Total data size

Displays the total data size of data registers used in I/O message communication.

(6) Remaining data size

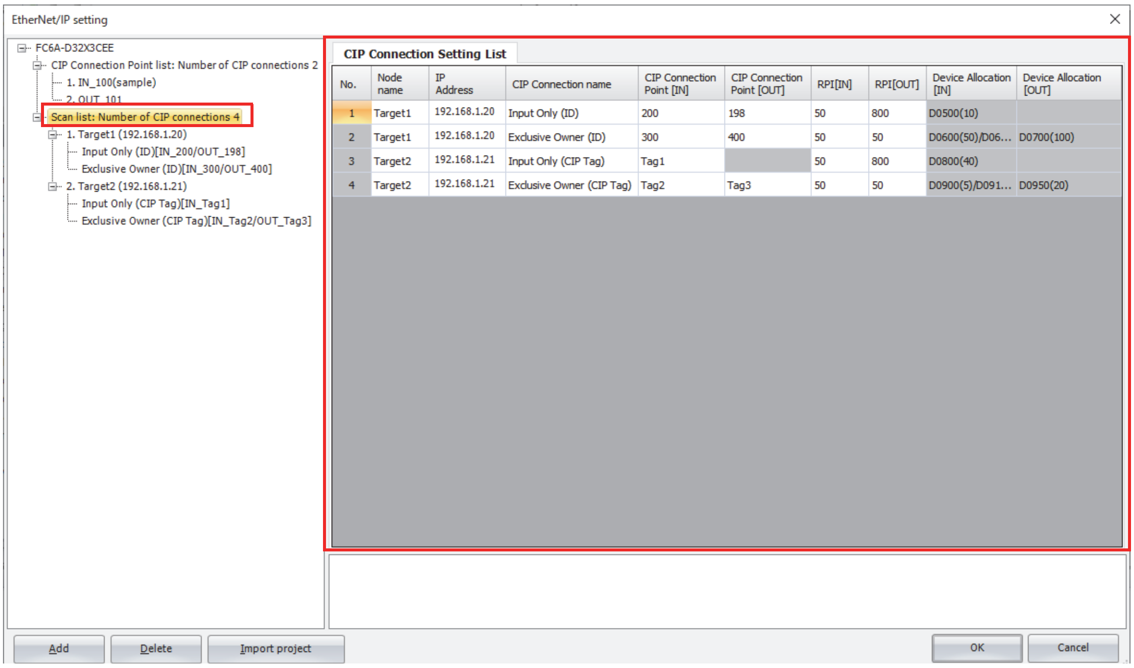
Displays the remaining data size of data that can be used (720 bytes minus the total value of data registers that are used).

When the Scan List node is selected

Select the **Scan List** node in the EtherNet/IP tree area. The **CIP Connection Setting List** tab is displayed in the parameter settings area.

■CIP Connection Setting List tab

This tab displays the list of configured CIP connections. The node name, IP address, CIP connection name, CIP connection point, and RPI can be changed. Double-click a device allocation cell. The **CIP Connection Setting** tab is displayed and you can move to the relevant cell for device allocation.

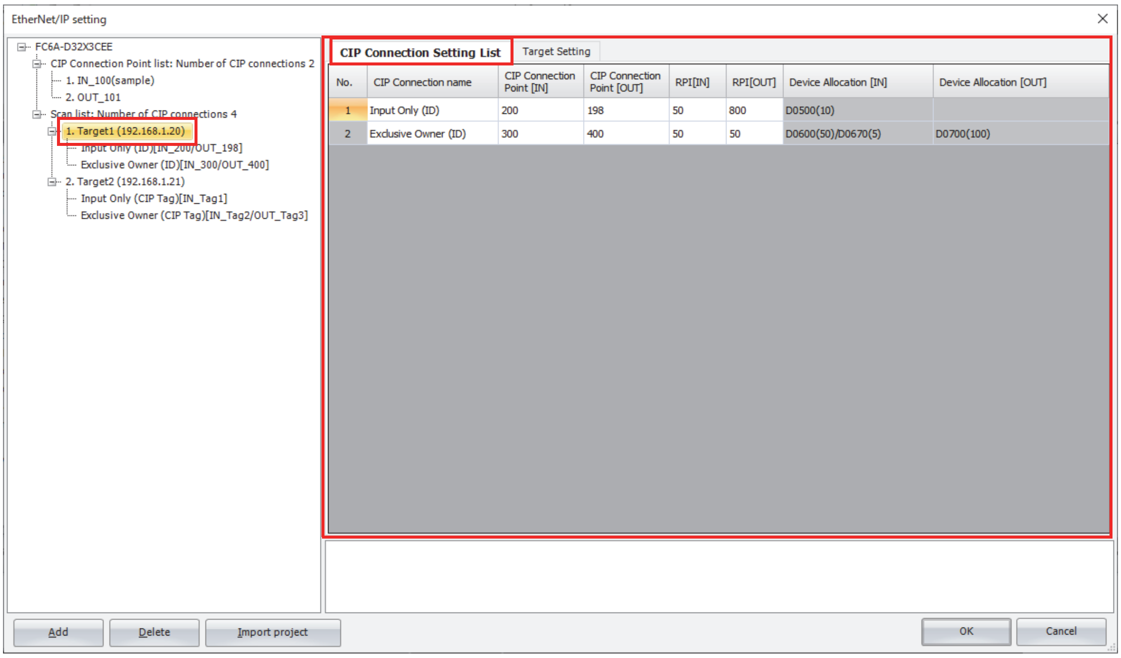


When the Target node is selected

Select a **Target** node in the EtherNet/IP tree area. The **CIP Connection Setting List** and **Target Setting** tabs are displayed in the parameter settings area.

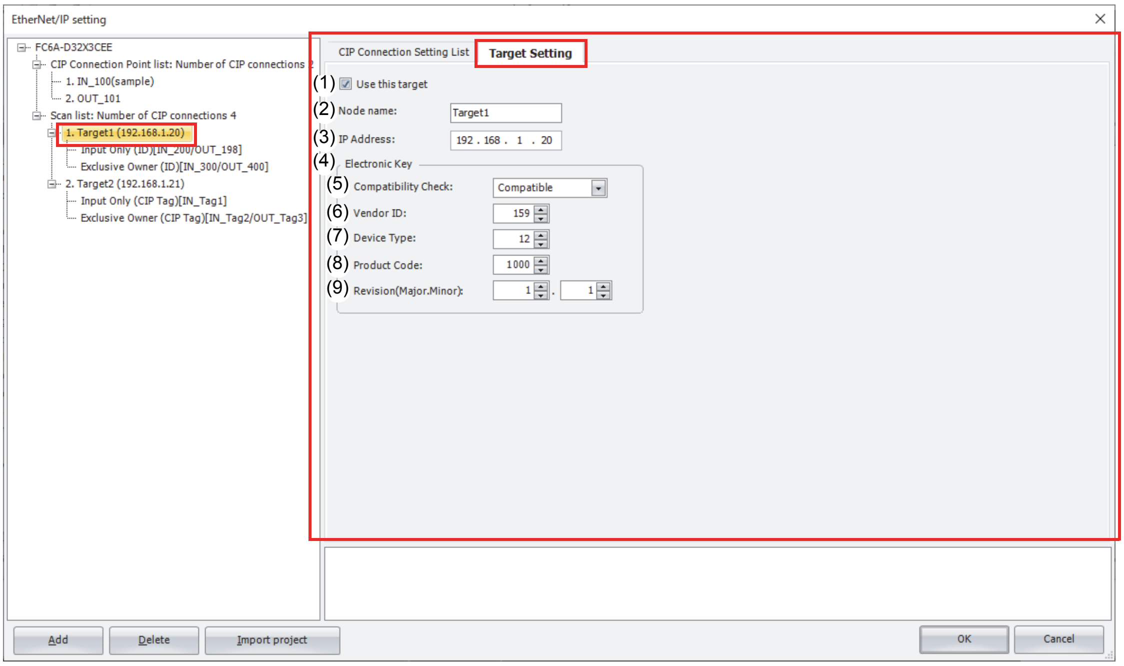
■CIP Connection Setting List tab

This tab displays the list of CIP connections of the configured target. The CIP connection name, CIP connection point, and RPI (communication cycle) can be changed on this tab.



■Target Setting tab

Configure the target device settings on this tab.



(1) Use this target

Enables or disables the CIP connections in the scan list by target. If the CIP connection is enabled, the Plus CPU module performs I/O message communication with the target when the EtherNet/IP Communication Bit (M8460) is turned on. If the CIP connection is disabled, the Plus CPU module does not perform I/O message communication. Select this check box to enable the CIP connection.

(2) Node name

Sets the node name of the target. The set node name is displayed on the scan list. Up to 30 bytes can be set.

(3) IP Address

Sets the IP address of the target.

(4) Electronic Key

Checks if the target registered to the scan list matches the EtherNet/IP device that is actually connected according to the compatibility check settings when the CIP connection is opened. I/O message communication is allowed with only a verified and matched EtherNet/IP device. If a device does not match, the Plus CPU module does not perform I/O message communication with the EtherNet/IP device.

(5) Compatibility Check

Sets the method for the compatibility check from the following three types.

Compatibility Check	Description
Disabled	Electronic key verification is not performed.
Compatibility	I/O message communication is allowed with only a target that satisfies all of the following conditions. • The settings and the actual vendor ID, device type, and product code match • The actual major revision and minor revision are larger than the settings
Complete Match	I/O message communication is performed with only a target that matches all of the set electronic key.

(6) Vendor ID

Sets the vendor ID of the EtherNet/IP device. Set this item between 0 and 65,535.

(7) Device Type

Sets the device type of the EtherNet/IP device. Set this item between 0 and 65,535.

(8) Product Code

Sets the product code of the EtherNet/IP device. Set this item between 0 and 65,535.

(9) Revision (Major, Minor)

Sets the major revision and minor revision of the EtherNet/IP device.

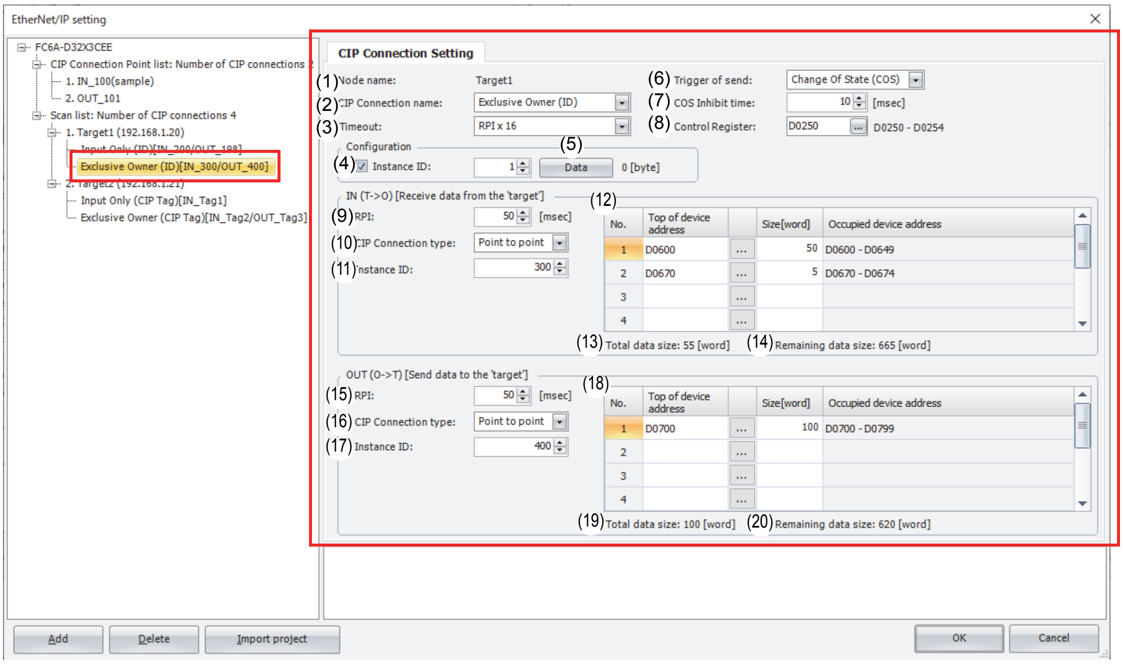
Revision	Range
Major revision	0 to 127
Minor revision	0 to 255

When the CIP Connection node is selected

Select a **CIP Connection** node in the EtherNet/IP tree area. The **CIP Connection Setting** tab is displayed in the parameter settings area.

■CIP Connection Setting tab

Configure the CIP connection of the target on this tab.



(1) Node name

Displays the node name that was set on the **Target Setting** tab. Configure the CIP connection of the target with this node name.

(2) CIP Connection name

Sets the CIP connection name of the target with the node name (1). This item can be set from the following five types.

Setting	Description
Exclusive Owner (ID)	Set this name when the Plus CPU module transmits and receives data with the target. For Exclusive Owner (ID), set the CIP connection point of the target as an instance ID. For Exclusive Owner (CIP Tag), set the CIP connection point of the target as a CIP tag name. The Plus CPU module specifies the CIP connection point of the target as a CIP tag name or an instance ID and performs I/O message communication with the target.
Exclusive Owner (CIP Tag)	
Input Only (ID)	Set this name when the Plus CPU module receives data from the target only. For Input Only (ID), set the CIP connection point of the target as an instance ID. For Input Only (CIP Tag), set the CIP connection point of the target as a CIP tag name. The Plus CPU module specifies the CIP connection point of the target as a CIP tag name or an instance ID and performs I/O message communication with the target.
Input Only (CIP Tag)	
Listen Only (ID)	When other originators and a target have opened a CIP connection as Exclusive Owner or Input Only and that target is transmitting data by multicast, set this name to have the Plus CPU module receive that data. For Listen Only (ID), the CIP connection point of the target is specified as an instance ID.

Note: The Plus CPU module cannot specify a Listen Only CIP connection as a CIP tag name.

(3) Timeout

Sets the timeout for I/O message communication. The Plus CPU module and target both monitor the time interval to receive data. The setting range is an integer multiple (x4, x8, x16, x32, x64, x128, x256, or x512) of the RPI ((9), (15)). The default value is RPI x 16.

(4) Instance ID

Sets the instance ID of the target Configuration. Select this check box and the instance ID can be set. Set the instance ID between 1 and 65,535. The default value is 1.

(5) Data

Sets the data for the target Configuration. When the Plus CPU module opens a CIP connection, it also transmits the Configuration instance ID to the target. A 0 to 400 byte parameter can be set in hexadecimal depending on the specification of the target.

(6) Trigger of send

Sets the method for the Plus CPU module to transmit the data to the target. This item can be set from the following two types. The default value is "Cyclic".

Setting	Description
Cyclic	The Plus CPU module and target transmit data at the set RPI (communication cycle) (9).
Change Of State (COS)	The Plus CPU module and target transmit data at the set RPI (communication cycle) (9) or when a value changes.

(7) COS Inhibit time

Sets the time from when a value changes to when the transmission occurs when Change Of State (COS) is selected for the trigger of send (6). The Plus CPU module does not transmit the data until the set time has elapsed from when the change in value was detected.

Note: A change in a value means a change in the data in the I/O data buffer held in the Plus CPU module. Data is reflected from the data registers to the I/O data buffer in the END processing.

(8) Control Register

Sets the data registers that will be used by the CIP connection.

Device Address	Item		Description
Starting number+0	Bit 0	Connection Status	ON: Connected and communicating status OFF: Not connected status (during retries, etc.)
	Bits 1 to 15	Reserved	
Starting number+1	Reserved		
Starting number+2			
Starting number+3			
Starting number+4			

Configure the parameters related to the input direction (target -> Plus CPU module (originator)) of I/O message communication in (9) to (14).

(9) RPI

Sets the communication cycle of the input direction (target -> originator) when the Plus CPU module (originator) is performing I/O message communication with the target. The range that can be set is 10 to 10,000 ms. The default value is 50 ms.

(10) CIP Connection type

Sets the method for the Plus CPU module (originator) to receive data transmitted from the target. This item can be set from the following two types. The default value is "Point to point".

Setting	Description
Point to point	Set this type when the Plus CPU module receives data from the target on a one-to-one basis.
Multicast	Set this type when multiple originators including the Plus CPU module receive data from one target.

(11) Instance ID/CIP Tag

Sets the CIP connection point of the target. Set the instance ID or CIP tag name of the target according to the CIP connection name. This item can be set from the following five types.

CIP Connection Name	CIP Connection Point	Range
Exclusive Owner (ID)	Instance ID	1 to 65,535
Exclusive Owner (CIP Tag)	CIP tag name	UTF-8, 64 bytes maximum (including string terminator)
Input Only (ID)	Instance ID	1 to 65,535
Input Only (CIP Tag)	CIP tag name	UTF-8, 64 bytes maximum (including string terminator)
Listen Only (ID)	Instance ID	1 to 65,535

(12) Device Allocation (IN (T->O))

Sets the data registers in which the Plus CPU module (originator) stores data that is received from the target. The data can be allocated to multiple data registers in word units. A maximum of 8 areas can be set.

The total size of the areas to allocate is 720 words maximum. Starting from the set data register, data registers are used in only the amount set by the data size. Set the data size between 1 and 720. The default value is 1. Set the first data register so that the device range is not exceeded.

(13) Total data size (IN (T->O))

Displays the total data size of data registers used in the device allocations (IN (T->O)) (12).

(14) Remaining data size (IN (T->O))

Displays the remaining data size of data that can be used (720 bytes minus the total value of data registers that are used).

Configure the parameters related to the output direction (Plus CPU module (originator) -> target) of I/O message communication in (15) to (20).

(15) RPI

Sets the communication cycle of the output direction (originator -> target) when the Plus CPU module (originator) is performing I/O message communication with the target. The range that can be set is 10 to 10,000 ms. The default value is 50 ms.

This RPI is the heartbeat cycle when Input Only or Listen Only is set for the CIP connection name. The value of the RPI for IN (T->O) data multiplied by 16 is automatically set, but it can be changed.

(16) CIP Connection type

Sets the method for the Plus CPU module (originator) to transmit data to the target. Only "Point to point" can be set.

Setting	Description
Point to point	Set this type when the Plus CPU module transmits data from the target on a one-to-one basis.

(17) Instance ID/CIP Tag

Sets the CIP connection point of the target. Set the instance ID or CIP tag name of the target according to the CIP connection name. This item can be set from the following five types.

CIP Connection Name	CIP Connection Point	Range
Exclusive Owner (ID)	Instance ID	1 to 65,535
Exclusive Owner (CIP Tag)	CIP tag name	UTF-8, 64 bytes maximum (including string terminator)
Input Only (ID)	Instance ID	1 to 65,535*1
Input Only (CIP Tag)	CIP tag name	Setting not required
Listen Only (ID)	Instance ID	1 to 65,535*2

*1 Set the instance ID for an Input Only target.
*2 Set the instance ID for a Listen Only target.

(18) Device Allocation (OUT (O->T))

Sets the data registers in which the Plus CPU module (originator) stores data to transmit to the target. The data can be allocated to multiple data registers in word units. A maximum of 8 areas can be set.

The total size of the areas to allocate is 720 words maximum. Starting from the set data register, data registers are used in only the amount set by the data size. Set the data size between 1 and 720. The default value is 1. Set the first data register so that the device range is not exceeded.

(19) Total data size (OUT (O->T))

Displays the total data size of data registers used in the device allocations (OUT (O->T)) (18).

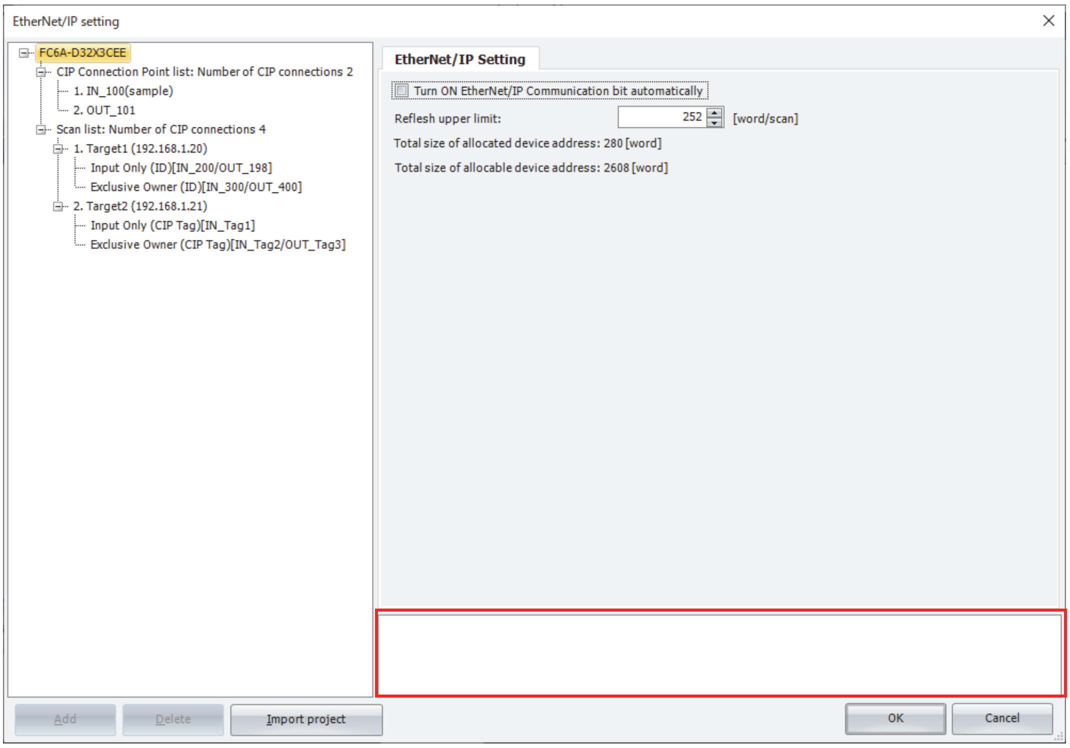
(20) Remaining data size (OUT (O->T))

Displays the remaining data size of data that can be used (720 bytes minus the total value of data registers that are used).

Information View Area

This area displays the details of errors if there are errors in content set on the **EtherNet/IP Settings** dialog box.

Click on the details of an error to display the location with the error.



Flow of EtherNet/IP Communication Settings

Basic Settings for EtherNet/IP Communication

Select the **PLC Type** node*1 in the EtherNet/IP tree area. The **EtherNet/IP Setting** tab is displayed in the parameter settings area. Configure the basic settings for performing EtherNet/IP communication on this tab. For details, see "When the PLC Type node is selected" on page 16-21.

*1 For details on the **PLC Type** node, see "EtherNet/IP Tree Area" on page 16-20.

Using the Plus CPU module as a Target

This section describes the settings required to use the Plus CPU module as a target.

A CIP connection point must be configured when the Plus CPU module (target) transmits and receives data with another scanner device (originator).

CIP Connection Point Settings

In the settings for the Plus CPU module to transmit and receive data with the originator, configure the data registers for storing communication data, data size, input and output direction of data, and other settings.

Select a **CIP Connection Point** node*1 in the EtherNet/IP tree area. The **CIP Connection Point Setting** tab is displayed in the parameter settings area. Configure the CIP connection point on this tab. For details, see "When the CIP Connection Point node is selected" on page 16-23.

*1 For details on the **CIP Connection Point** node, see "EtherNet/IP Tree Area" on page 16-20.

Using the Plus CPU module as an Originator

This section describes the settings required to use the Plus CPU module as an originator.

You must configure the settings for the target device that performs I/O message communication with the Plus CPU module and the CIP connection settings.

Target Settings

Configure the target device settings, such as the IP address of the target that performs I/O message communication with the Plus CPU module.

Select a **Target** node*1 in the EtherNet/IP tree area. The **Target Setting** tab is displayed in the parameter settings area. Configure the target device settings on this tab. For details, see "When the Target node is selected" on page 16-24.

*1 For details on the **Target** node, see "EtherNet/IP Tree Area" on page 16-20.

CIP Connection Settings

In the settings for the Plus CPU module to transmit and receive data with the target, configure the CIP connection name, CIP connection type, data registers for storing communication data, data size, RPI (communication cycle) for I/O message communication, and the other settings for each target.

Select a **CIP Connection** node*1 in the EtherNet/IP tree area. The **CIP Connection Setting** tab is displayed in the parameter settings area. Configure the CIP connection settings on this tab. For details, see "When the CIP Connection node is selected" on page 16-26.

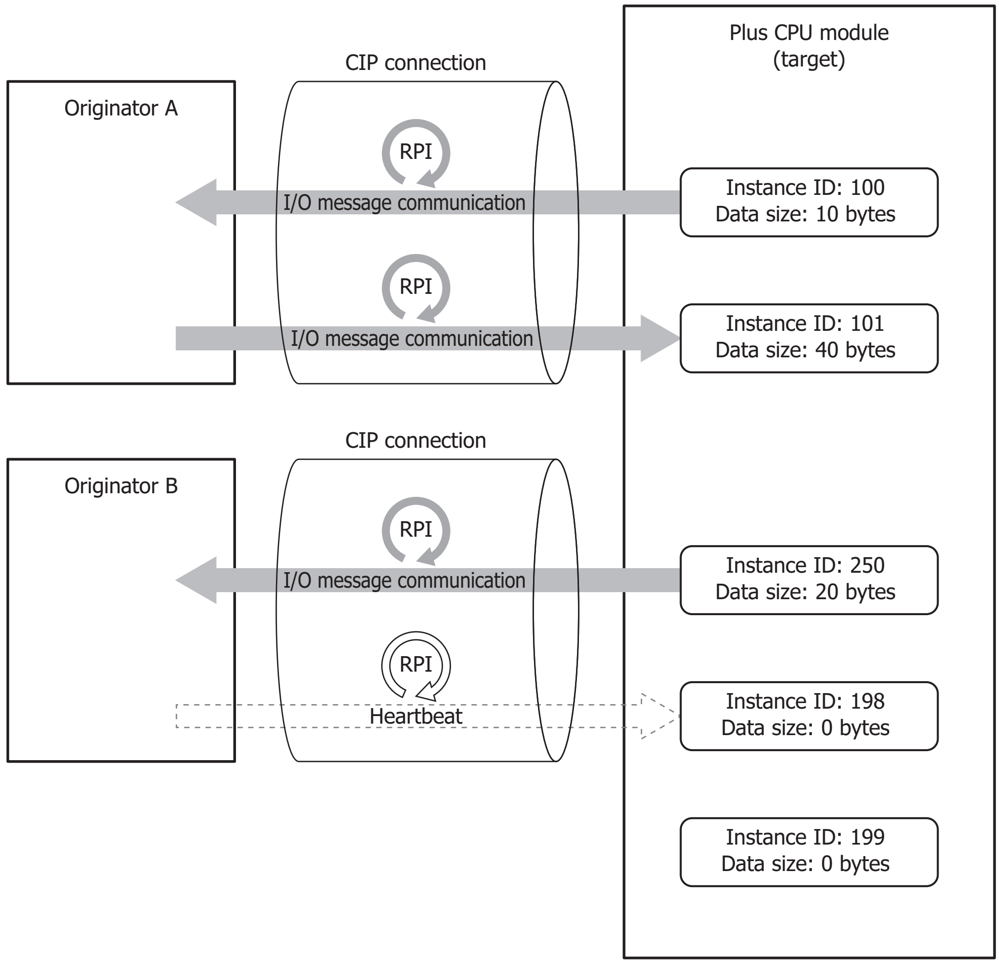
*1 For details on the **CIP Connection** node, see "EtherNet/IP Tree Area" on page 16-20.

Examples of EtherNet/IP Settings

Using the Plus CPU module as a Target

This section describes an example of the settings for connecting to an originator with the Plus CPU module as the target.

System configuration diagram



Specifications to set for the Plus CPU module

Configure a CIP connection point to read and write from Originator A and B using the Plus CPU module as a target.

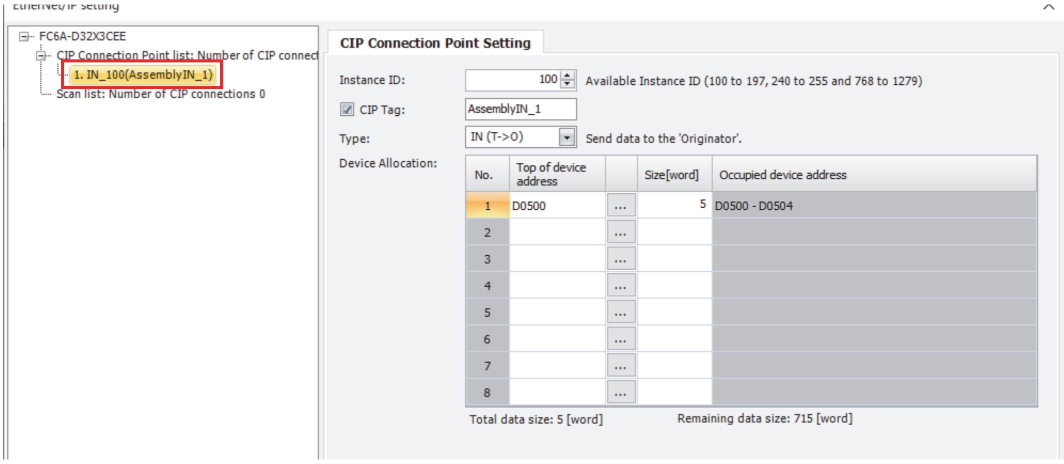
CIP Connection Point			Data Register for Storing Data
Input & Output Type	Instance ID	Data Size	
Input	100	10 bytes (5 words)	D0500
Output	101	40 bytes (20 words)	D0600
Input	250	20 bytes (10 words)	D0700
Output (for Input Only)	198 (fixed by system)	0 bytes	—
Output (for Listen Only)	199 (fixed by system)	0 bytes	—

Settings

Configure the CIP connection point settings for the Plus CPU module.

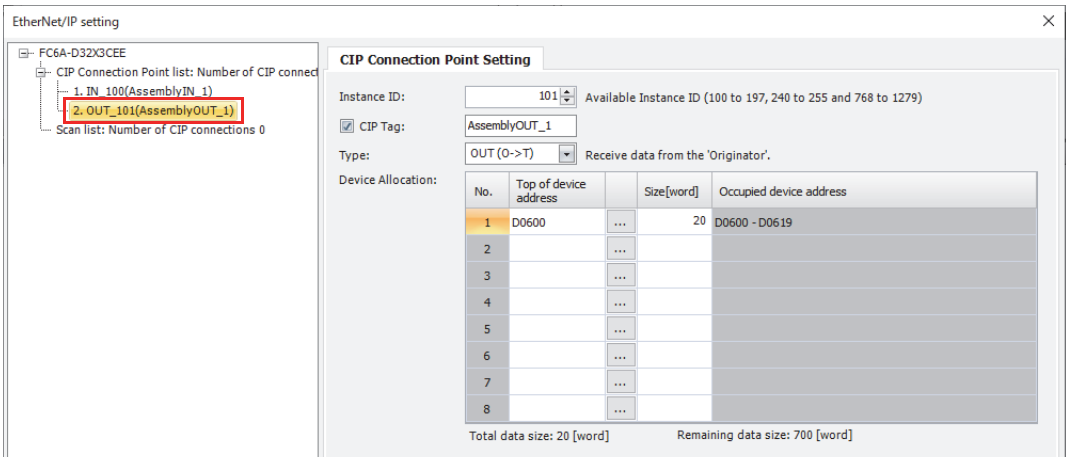
■Plus CPU module CIP Connection Point Settings

The Plus CPU module (target) transmits the CIP connection point (instance ID: 100) data stored in data registers to the originator.



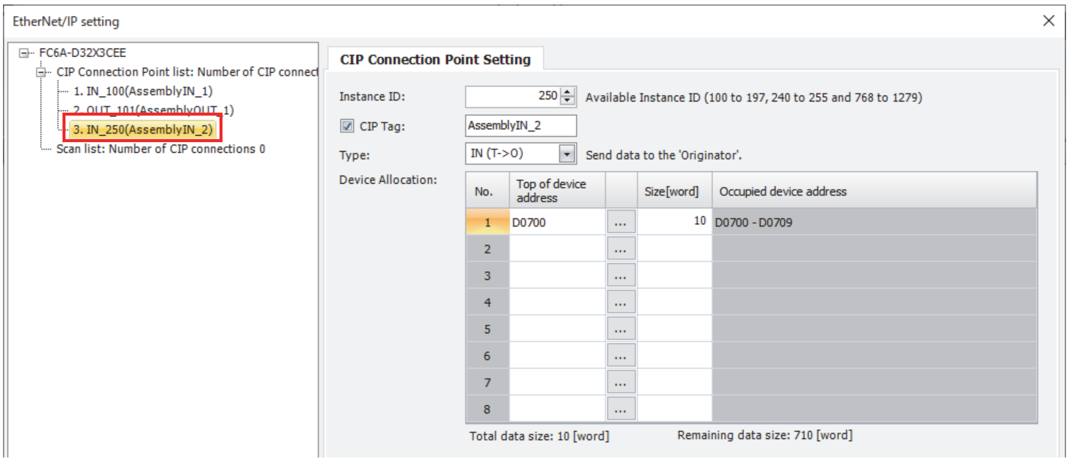
Settings		Description
1.	CIP tag	"AssemblyIN_1"
	Instance ID	100
	Type	IN (T->O)
	Device allocations	D0500 to D0504 (5 words)

The Plus CPU module (target) receives the CIP connection point (instance ID: 101) data from the originator and stores it in data registers.



Settings		Description
2.	CIP tag	"AssemblyOUT_1"
	Instance ID	101
	Type	OUT (O->T)
	Device allocations	D0600 to D0619 (20 words)

The Plus CPU module (target) transmits the CIP connection point (instance ID: 200) data stored in data registers to the originator.



Settings		Description
3.	CIP tag	"AssemblyIN_2"
	Instance ID	250
	Type	IN (T->O)
	Device allocations	D0700 to D0709 (10 words)

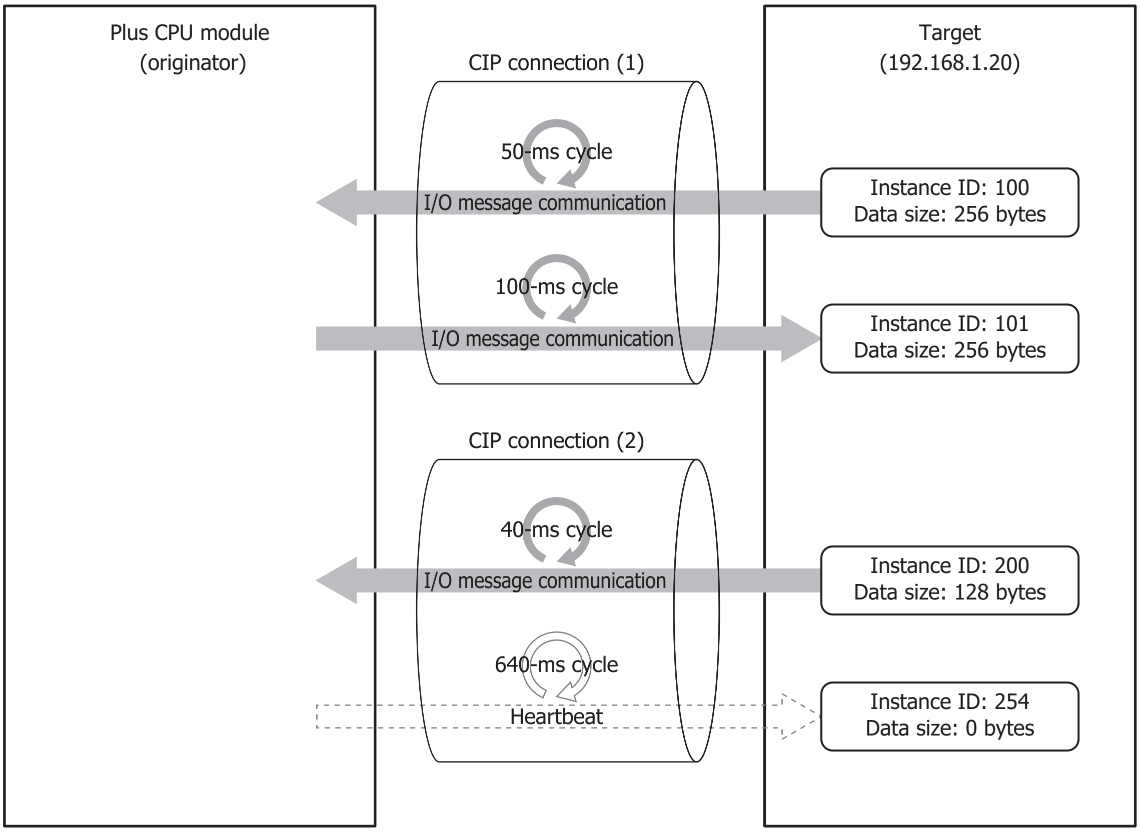
Note: The RPI and CIP connection type, such as Exclusive Owner, are specified by the originator when it opens the CIP connection.

This concludes configuring the settings.

Using the Plus CPU module as an Originator

This section describes an example of the settings for connecting to a target with the Plus CPU module as the originator.

System configuration diagram



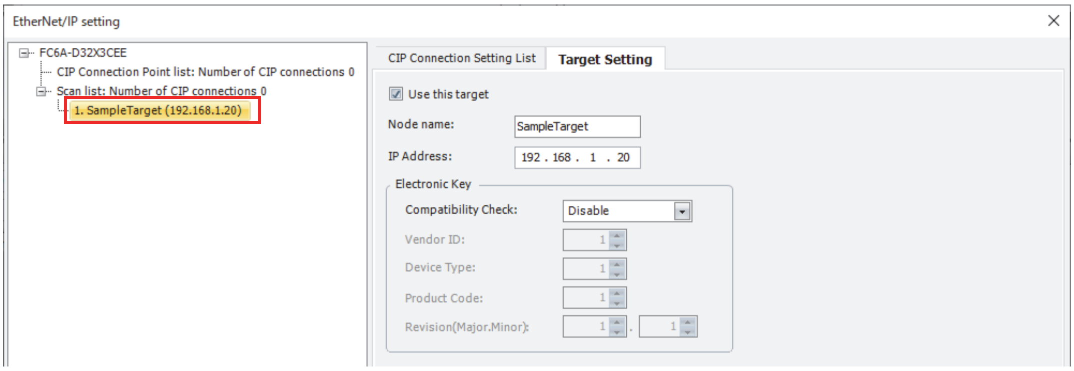
Specifications of destination target

IP Address	192.168.1.20			
CIP Connection Point	The following table shows the CIP connection point of the target that Plus CPU module connects to as the originator.			
	CIP connection	Type	Instance ID	Data Size
	CIP connection (1)	Configuration	1	0 bytes
		Input	100	256 bytes (128 words)
		Output	101	256 bytes (128 words)
	CIP connection (2)	Configuration	1	0 bytes
		Input	200	128 bytes (64 words)
		Output (Heartbeat)	254	0 bytes

Settings

Configure the target device and CIP connection settings.

■Plus CPU module Scan List Settings (Target Device Settings)

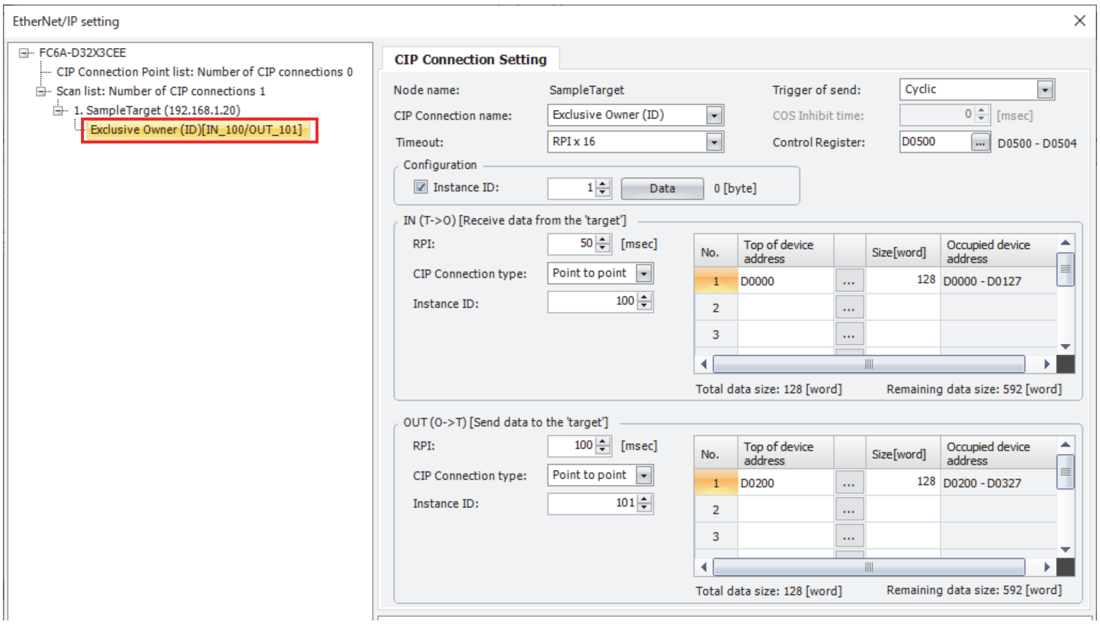


Settings		Description
Use this target		Enable
Node name		SampleTarget
IP Address		192.168.1.20
Electronic Key	Compatibility Check	Disable

■Plus CPU module Scan List Settings (CIP Connection Settings)

CIP connection (1)

The Plus CPU module (originator) sends a request to open the CIP connection to the CIP connection point (instance ID: 100 and 101), and when successful, receives data from the target every 50 ms and transmits data to the target every 100 ms.

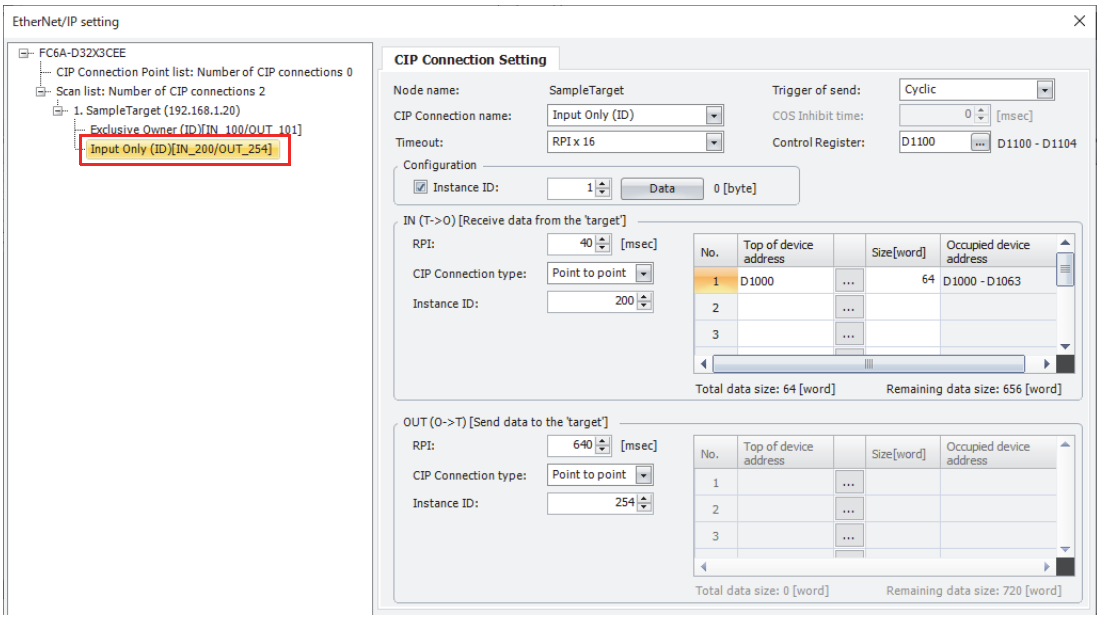


Settings		Description
CIP Connection Name		Exclusive Owner (ID)
Trigger of send		Cyclic
Timeout		RPI x 16
Control registers		D0500
Configuration	Instance ID	1
	Data	0 bytes

Settings		Description
IN (T->O)	RPI	50 ms
	CIP connection type	Point to point
	Instance ID	100
	Device allocations	D0000 to D0127 (128 words)
OUT (O->T)	RPI	100 ms
	CIP connection type	Point to point
	Instance ID	101
	Device allocations	D0200 to D0327 (128 words)

CIP connection (2)

Set Input Only for the CIP connection name when data is not transmitted to the target. The Plus CPU module (originator) sends a request to open the CIP connection to the CIP connection point (instance ID: 200 and 254), and when successful, receives data from the target every 40 ms. The heartbeat, not data, is sent to the target every 640 ms.



Settings		Description
CIP Connection Name		Input Only (ID)
Trigger of send		Cyclic
Timeout		RPI x 16
Control registers		D1100
Configuration	Instance ID	1
	Data	0 bytes
IN (T->O)	RPI	40 ms
	CIP connection type	Point to point
	Instance ID	200
	Device allocations	D1000 to D1063 (64 words)
OUT (O->T)	RPI	640msec
	CIP connection type	Point to point
	Instance ID	254

This concludes configuring the settings.

Objects

Identity Object

Class

■Class Services

Supports Get_Attribute_Single (0EH) and Get_Attributes_All (01H).

ID	Services
0EH	Get_Attribute_Single
01H	Get_Attributes_All

■Class Attributes (Instance ID: 0)

ID	Access	Name	Data Type
1	Get	Revision	UINT
2	Get	Max Instance	UINT
3	Get	Number of Instances	UINT
6	Get	Maximum ID Number Class Attributes	UINT
7	Get	Maximum ID Number Instance Attributes	UINT

Instance

■Instance Services

Supports Get_Attribute_Single (0EH), Get_Attributes_All (01H), and Reset service (05H).

ID	Services
0EH	Get_Attribute_Single
01H	Get_Attributes_All
05H	Reset

■Instance Attributes (Instance ID: 1)

ID	Access	Name	Data Type	Description	Attribute Value
1	Get	Vendor ID	UINT	Vendor identification number	159
2	Get	Device Type	UINT	General device type	14 (Programmable Logic Controller)
3	Get	Product Code	UINT	Product identification code	2000
4	Get	Revision	STRUCT	Revision of Identity object	System software version
		Major Revision	USINT	Major revision	
		Minor Revision	USINT	Minor revision	
5	Get	Status	WORD	Current status of device	
6	Get	Serial Number	UDINT	Serial number	
7	Get	Product Name	SHORT-STRING	Product name	"FC6A"

Message Router Object

Class

■Class Services

ID	Services
0EH	Get_Attribute_Single

■Class Attributes (Instance ID: 0)

ID	Access	Name	Data Type
1	Get	Revision	UINT
2	Get	Max Instance	UINT
3	Get	Number of Instances	UINT
6	Get	Maximum ID Number Class Attributes	UINT
7	Get	Maximum ID Number Instance Attributes	UINT

Instance

■Instance Services

ID	Services
0EH	Get_Attribute_Single

■Instance Attributes (Instance ID: 1)

ID	Access	Name	Data Type
1	Get	Object_list	STRUCT
	Get	Number	UINT
	Get	Classes	Array of UINT
2	Get	Number Available	UINT

Assembly Object

Class

■**Class Services**

ID	Services
0EH	Get_Attribute_Single

■**Class Attributes (Instance ID: 0)**

ID	Access	Name	Data Type
1	Get	Revision	UINT

Instance

■**Instance Services**

ID	Services
0EH	Get_Attribute_Single
10H	Set_Attribute_Single

■**Instance Attributes**

(1) Originator -> Target

ID	Access	Name	Data Type
3	Get/Set	Data	Array of BYTE
4	Get	Size	UINT

(2) Target -> Originator

ID	Access	Name	Data Type
3	Get	Data	Array of BYTE
4	Get	Size	UINT

Connection Manager Object

Class

■**Class Services**
There are no class services.

■**Class Attributes**
There are no class attributes.

Instance

■**Instance Services**

ID	Services
54H	Forward_Open
5BH	Large_Forward_Open
4EH	Forward_Close

■**Instance Attributes**
There are no instance attributes.

TCP/IP Interface Object

Class

■Class Services

ID	Services
0EH	Get_Attribute_Single

■Class Attributes

ID	Access	Name	Data Type
1	Get	Revision	UINT
2	Get	Max Instance	UINT
3	Get	Num Instances	UINT
6	Get	Maximum ID Number Class Attributes	UINT
7	Get	Maximum ID Number Instance Attributes	UINT

Instance

■Instance Services

ID	Services
0EH	Get_Attribute_Single
10H	Set_Attribute_Single

■Instance Attributes

ID	Access	Name	Data Type
1	Get	Status	DWORD
2	Get	Configuration Capability	DWORD
3	Get	Configuration Control	DWORD
4	Get	Physical Link Object	STRUCT
		Path size	UINT
		Path	EPATH
5	Get	Interface Configuration	STRUCT
		IP Address	UDINT
		Network Mask	UDINT
		Gateway Address	UDINT
		Name Server	UDINT
		Name Server 2	UDINT
		Domain Name	STRING
6	Get	Host Name	STRING
13	Get/Set	Encapsulation Inactivity Timeout	UINT

Ethernet Link Object

Configures and reads Ethernet communication settings. Also reads the status of Ethernet communication.

Class

■Class Services

ID	Services
0EH	Get_Attribute_Single

■Class Attributes

ID	Access	Name	Data Type
1	Get	Revision	UINT
2	Get	Max Instance	UINT
3	Get	Num Instances	UINT

Instance

■Instance Services

Supports Get_Attribute_Single (0EH) and Get_Attributes_All (01H).

ID	Services
0EH	Get_Attribute_Single
01H	Get_Attributes_All

■Instance Attributes

ID	Access	Name	Data Type
1	Get	Interface Speed	UDINT
2	Get	Interface Flags	DWORD
3	Get	Physical Address	Array of 6 USINTs
11	Get	Interface Capability	STRUCT
		Capability Bits	DWORD
		Speed/Duplex Options	STRUCT
		Speed/Duplex Array Count	USINT
		Speed/Duplex Array	Array of STRUCT
		Interface Speed	UINT
		Interface Duplex Mode	USINT

INDEX

IDEC CORPORATION

Head Office
6-64, Nishi-Miyahara-2-Chome, Yodogawa-ku, Osaka 532-0004, Japan

www.idec.com

USA	IDEC Corporation	Tel: +1-408-747-0550	opencontact@idec.com	**Hong Kong**	IDEC Izumi (H.K.) Co., Ltd.	Tel: +852-2803-8989	info@hk.idec.com
Germany	APEM GmbH	Tel: +49-40-25 30 54 - 0	service@eu.idec.com	**China/Shanghai**	IDEC (Shanghai) Corporation	Tel: +86-21-6135-1515	idec@cn.idec.com
Singapore	IDEC Izumi Asia Pte. Ltd.	Tel: +65-6746-1155	info@sg.idec.com	**China/Shenzhen**	IDEC (Shenzhen) Corporation	Tel: +86-755-8356-2977	idec@cn.idec.com
Thailand	IDEC Asia (Thailand) Co., Ltd	Tel: +66-2-392-9765	sales@th.idec.com	**China/Beijing**	IDEC (Beijing) Corporation	Tel: +86-10-6581-6131	idec@cn.idec.com
Australia	IDEC Australia Pty. Ltd.	Tel: +61-3-8523-5900	sales@au.idec.com	**Japan**	IDEC Corporation	Tel: +81-6-6398-2527	marketing@idec.co.jp
Taiwan	IDEC Taiwan Corporation	Tel: +886-2-2698-3929	service@tw.idec.com				

B-1730(11) JULY 2020