FC6A SERIES

MICROSmart

All-in-One Type
Communication Manual

Automation Organizer
WindLDR

SAFETY PRECAUTIONS

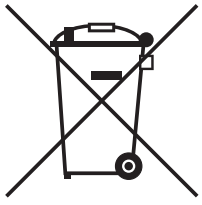
- Read the "FC6A Series MicroSmart All-in-One Type Communication Manual" to ensure correct operation before starting installation, wiring, operation, maintenance, and inspection of the FC6A Series MicroSmart.
- All FC6A Series MicroSmart modules are manufactured under IDEC's rigorous quality control 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.
- 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 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 All-in-One Type 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 All-in-One Type 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

This manual describes functions, specifications, installation, and operation basics of the FC6A Series MicroSmart. Also included is information on the powerful communications tools of the FC6A Series MicroSmart, as well as troubleshooting procedures.

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: 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.

Chanpter 10: Send E-mail Function
Descriptions of the EMAIL instruction that sends preregistered e-mails.

Chanpter 11: Web Server
Description of the Web server 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

Trademarks

FC6A Series MicroSmart is a trademark of IDEC Corporation.

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
- EMC Directive

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

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
- CSA C22.2 No.142
- ANSI/ISA 12,12,01*1
- CAN/CSA C22.2 No.213*1

*1 Certain FC6A Series MicroSmart models are not compatible. For details, please contact IDEC Corporation.
For details on compatible 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.

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 All-in-One Type User's Manual	Describes product specifications, installation and wiring instructions, instructions for basic programming operations and special functions, device and instruction lists, communication functions, and troubleshooting procedures for the FC6A Series MicroSmart series.
FC9Y-B1726	FC6A Series MicroSmart LAD 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 All-in-One Type 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 series.

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 Type		FC6A-C16R1AE, FC6A-C16R1CE, FC6A-C16K1CE, FC6A-C16P1CE, FC6A-C24R1AE, FC6A-C24R1CE, FC6A-C24K1CE, FC6A-C24P1CE, FC6A-C40R1AE, FC6A-C40R1CE, FC6A-C40K1CE, FC6A-C40P1CE, FC6A-C40R1DE, FC6A-C40R1DE, FC6A-C40K1DE, FC6A-C40P1DE
	CAN J1939 All-in-One Type		FC6A-C40R1AEJ, FC6A-C40R1CEJ, FC6A-C40K1CEJ, FC6A-C40P1CEJ, FC6A-C40R1DEJ, FC6A-C40K1DEJ, FC6A-C40P1DEJ
	16-I/O type		The general term for the model with 16 I/O points (FC6A-C16R1AE, FC6A-C16R1CE, FC6A-C16K1CE, FC6A-C16P1CE)
	24-I/O type		The general term for the model with 24 I/O points (FC6A-C24R1AE, FC6A-C24R1CE, FC6A-C24K1CE, FC6A-C24P1CE)
	40-I/O type		The general term for the model with 40 I/O points (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)
	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
		12V DC power type	FC6A-C40R1DE, FC6A-C40K1DE, FC6A-C40P1DE, FC6A-C40R1DEJ, FC6A-C40K1DEJ, FC6A-C40P1DEJ
	Relay output type		FC6A-C16R1AE, FC6A-C16R1CE, FC6A-C24R1AE, FC6A-C24R1CE, FC6A-C40R1AE, FC6A-C40R1CE, FC6A-C40R1DE, FC6A-C40R1AEJ, FC6A-C40R1CEJ, FC6A-C40R1DEJ
	Transistor output type	Transistor sink output type	FC6A-C16K1CE, FC6A-C24K1CE, FC6A-C40K1CE, FC6A-C40K1DE, FC6A-C40K1CEJ, FC6A-C40K1DEJ
		Transistor protection source output type	FC6A-C16P1CE, FC6A-C24P1CE, FC6A-C40P1CE, FC6A-C40P1DE, FC6A-C40P1CEJ, FC6A-C40P1DEJ
Expansion module			Expansion I/O module, functional module
Expansion I/O module			Input module, output module, mixed I/O module
Functional module			Analog module, PID module
Analog module			Analog input module, analog output module, mixed analog I/O module
Option module			HMI module, expansion interface module, analog cartridge, communication cartridge
Cartridge			Analog cartridge, 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 RS232C and RS485 communication using serial port 1. The communication ports can be expanded by using communication cartridges to allow for multiple instances of RS232C and RS485 communication. The Ethernet port is standard equipment to enable communication using Ethernet. Also, the Ethernet port can be expanded by using an HMI module. The CAN J1939 All-in-One Type is equipped with a CAN port to enable J1939 communication.

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.

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, data link communication
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

Ethernet Port 1

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 of these connections can simultaneously use a different communication protocol. Each connection can be configured for maintenance communication (server), user communication (server/client), or Modbus TCP communication (server/client).

Communication Type	IEEE 802.3 compliant
Communication Speed	10BASE-T, 100BASE-TX
Number of Connections	8 maximum
Communication Functions	Maintenance communication, user communication, Modbus TCP server/client
Connector	RJ45
Cable	CAT 5. STP
Maximum Cable Length	100 m
Isolation between Internal Circuit	Pulse transformer isolated

Communication Cartridge

The communication cartridges can only be used when connected to cartridge slots 1 and 2 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-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 communication, data link communication	
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

CAN Port

The CAN J1939 All-in-One Type 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	System Web Page	Used	Used	Not used	Not used
	Web Page Editor	Used	Not used	Used	Not used
	Area Available to User	2.5 MB	4.5 MB	3.0 MB	5.0 MB
Send E-mail Function		Yes			
Connector		RJ45			
Cable		CAT 5. STP or higher			
Maximum Cable Length		100 m			
Isolation between Internal Circuit		Pulse transformer isolated			

⚠ 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 All-in-One Type User's Manual".

<table>
<tr><th>Type No.</th><th>USB Port</th><th>Serial Port 1</th><th>Ethernet Port 1</th><th>Communication Cartridge*1</th><th>CAN Port</th><th>HMI-Ethernet Port*2</th></tr>
<tr><td>FC6A-C16R1AE</td><td rowspan="24">1</td><td rowspan="15">1</td><td rowspan="24">1</td><td rowspan="5">1 maximum</td><td rowspan="15">—</td><td rowspan="24">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-C24R1AE</td></tr>
<tr><td>FC6A-C24R1CE</td><td rowspan="19">2 maximum</td></tr>
<tr><td>FC6A-C24K1CE</td></tr>
<tr><td>FC6A-C24P1CE</td></tr>
<tr><td>FC6A-C40R1AE</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="9">—</td><td rowspan="9">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>
</table>

*1 The communication cartridges can only be used when connected to cartridge slots 1 and 2 of the CPU module.
*2 The HMI-Ethernet port can only be used when a CPU module and an HMI module are connected.

Communication Ports, Serial Port 1, Cartridge Slot 1 and 2 Corresponding Table

The communication ports that are used in serial communication support the following communication interfaces.

<table>
<tr><th rowspan="2">Type No.</th><th colspan="3">Serial Port</th></tr>
<tr><th>Port 1</th><th>Port 2</th><th>Port 3</th></tr>
<tr><td>FC6A-C16R1AE</td><td rowspan="15">Serial Port 1*1</td><td rowspan="23">Cartridge Slot 1
communication cartridge*2 *4</td><td rowspan="8">No supported communication interface</td></tr>
<tr><td>FC6A-C16R1CE</td></tr>
<tr><td>FC6A-C16K1CE</td></tr>
<tr><td>FC6A-C16P1CE</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="15">Cartridge Slot 2
communication cartridge*3 *4</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="8">No supported communication interface</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>
</table>

*1 Can be used as port 1 for RS232C communication or RS485 communication.
To use, configure the interface under Communication Port in Function Area Settings.

*2 Can be used as port 2 by installing the RS232C communication cartridge (FC6A-PC1) or the RS485 communication cartridge (FC6A-PC3).

*3 Can be used as port 3 by installing the RS232C communication cartridge (FC6A-PC1) or the RS485 communication cartridge (FC6A-PC3).

*4 Cannot be set to "Data Bits: 7 bits" and "Parity: None".

Notes:

- For the locations of serial port 1, cartridge slot 1, and cartridge slot 2, see Chapter 2 "Parts Description" in the "FC6A Series MicroSmart All-in-One Type User's Manual".
- For serial port 1 wiring, see Chapter 2 "Other Inputs and Ports" in the "FC6A Series MicroSmart All-in-One Type User's Manual". For communication cartridge wiring, see Chapter 2 "Terminal Arrangement and Wiring Examples" in the "FC6A Series MicroSmart All-in-One Type User's Manual".

Communication Functions Overview

The FC6A Series MicroSmart supports maintenance communication, user communication, Modbus communication, data link communication, and J1939 communication functions.

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]

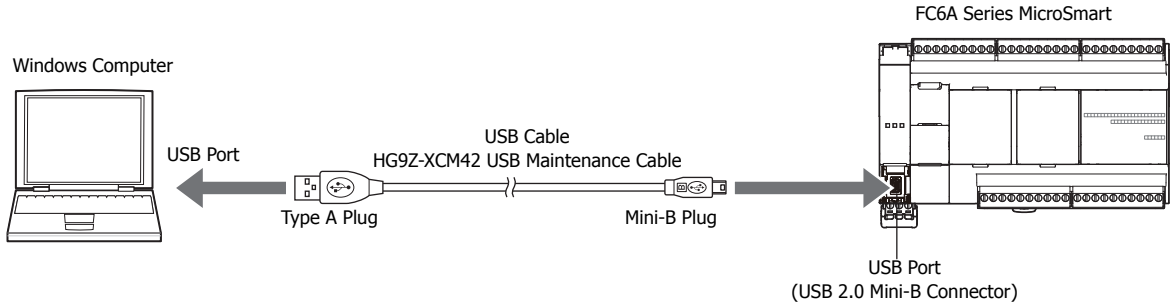
USB Port	Serial Port 1	Ethernet Port 1	Communication Cartridge	CAN Port	HMI-Ethernet Port
Yes	Yes	Yes	Yes	No	Yes

*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.

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 All-in-One Type User's Manual".

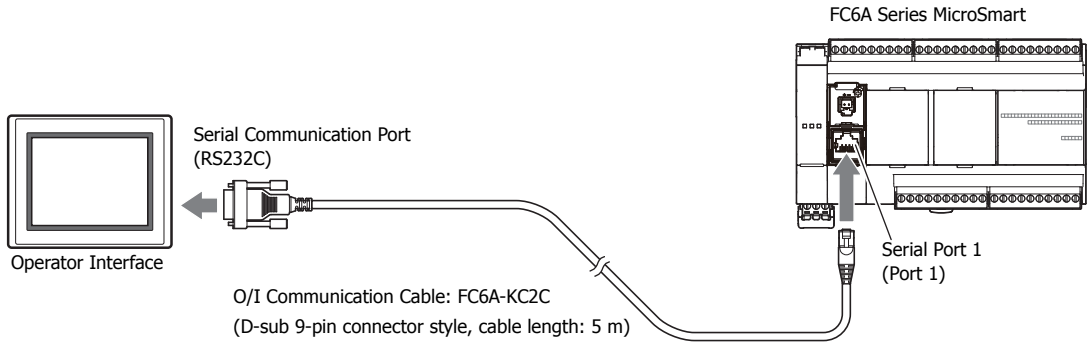
■ 1:1 Maintenance Communication System

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

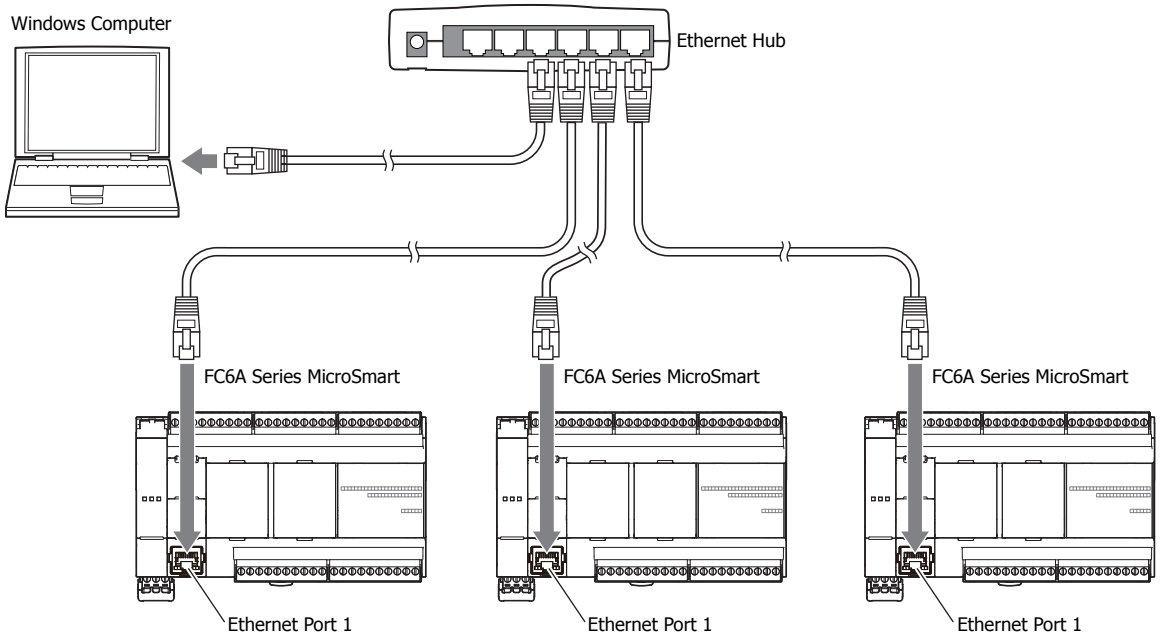
This example shows maintenance communication between the FC6A Series MicroSmart and an operator interface, as well as monitoring and changing FC6A Series MicroSmart device values using the operator interface. An IDEC operator interface is connected to serial port 1 of the FC6A Series MicroSmart.



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

■ **1:N Maintenance Communication System**

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

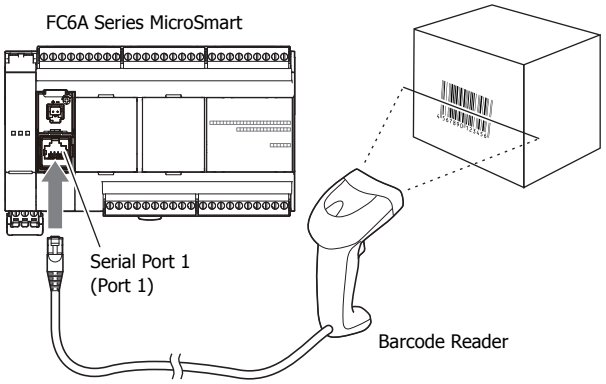
The user communication of the FC6A Series MicroSmart enables you to control external devices such as computers, printers, and barcode readers. For details on user communication, see "User Communication Instructions" on page 5-1.

Supported ports

USB Port	Serial Port 1	Ethernet Port 1	Communication Cartridge	CAN Port	HMI-Ethernet Port
No	Yes	Yes	Yes	No	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 is compliant with 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.

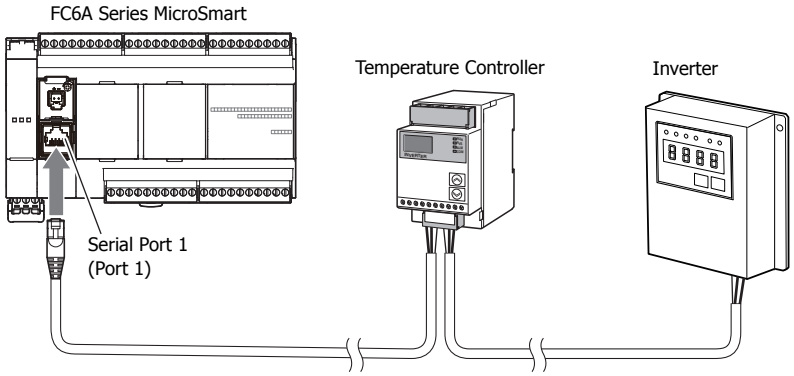
For details on Modbus communication, see "Modbus Communication" on page 6-1.

Supported ports

USB Port	Serial Port 1	Ethernet Port 1	Communication Cartridge	CAN Port	HMI-Ethernet Port
No	Yes	Yes	Yes	No	No

■ 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 System

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.

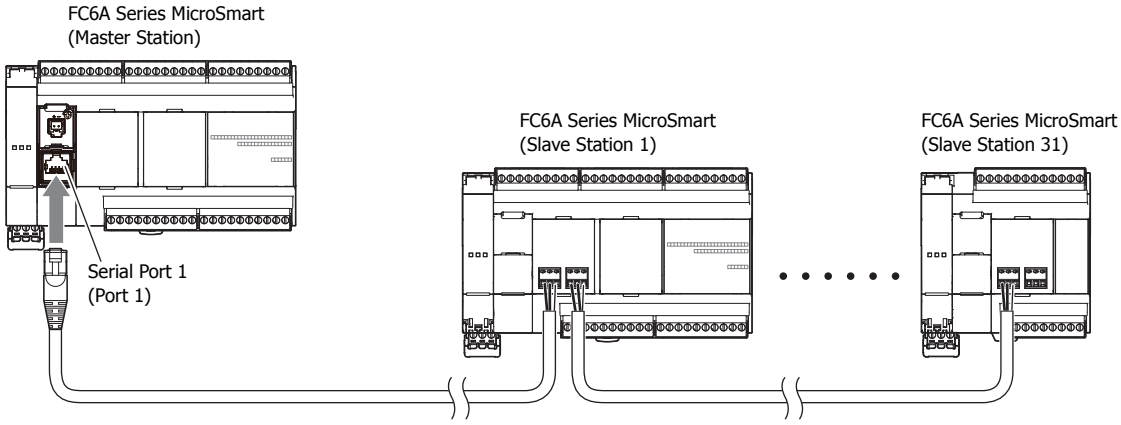
For details about the data link communication, see "Data Link Communication" on page 7-1.

Supported ports

USB Port	Serial Port 1	Ethernet Port 1	Communication Cartridge	CAN Port	HMI-Ethernet Port
No	Yes	No	Yes	No	No

■ 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.



Using J1939 Communication

The CAN J1939 All-in-One Type 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.

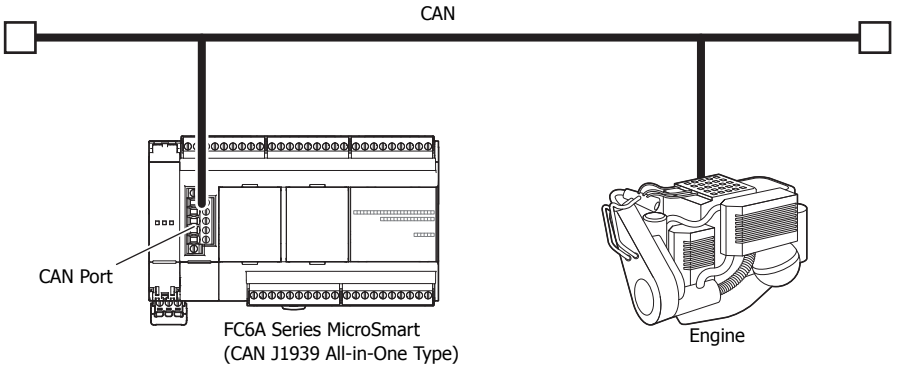
For details on J1939 communication, see "J1939 Communication" on page 8-1.

Supported ports

USB Port	Serial Port 1	Ethernet Port 1	Communication Cartridge	CAN Port	HMI-Ethernet Port
No	No	No	No	Yes	No

■ CAN Port Usage Example

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



2: DEVICE

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 when entering and editing devices in the user program.

Note: The entry 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

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 Input Relays*1	I	Bit	I30 - I187 (128 points) I190 - I507*2 (256 points)	I30 - I307 (224 points) I310 - I627*3 (256 points)	
Output*1	Q	Bit	Q0 - Q6 (7 points)	Q0 - Q11 (10 points)	Q0 - Q17 (16 points)
Expansion Outputs*1*3	Q	Bit	Q30 - Q187 (128 points) I190 - I507*2 (256 points)	Q30 - Q307 (224 points) Q310 - Q627*3 (256 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 an expansion module (expansion interface side) is connected using the expansion interface module.
*3 I310 to I627 and Q310 to Q627 are devices that can only be used when an expansion module (expansion interface side) is connected using the expansion interface module.

■ **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 LAD 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 LAD 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 M17497) and special internal relays (M8000 to M8317) 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 D55999) and special data registers (D8000 to D8499) 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	**— Reserved —**		—	—	—
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	Cleared	Cleared	R/W
M8044		Comparison Output Reset	Maintained	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 to M8147	**— Reserved —**		—	—	—
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	Cleared	R
M8170 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 to M8183	**— Reserved —**		—	—	—
M8184	Change HMI Module Network Settings Trigger		Operating	Cleared	R/W
M8185	In Daylight Saving Time Period		Operating	Cleared	R
M8186	Executing Auto Ping		Operating	Cleared	R
M8187	Auto Ping Stop Flag		Operating	Cleared	R/W
M8190	Change CPU Module 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	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 Completed 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 M8317	**— Reserved —**		—	—	—

Supplementary Descriptions of the Special Internal Relays Related to the Communication Functions

■ **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.

■ **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.

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

■ **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.

■ **M8184: Change HMI Module Network Settings Trigger**
When M8184 is turned on, the values written to 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 3-5.

■ **M8186: Executing Auto Ping**
M8186 is on when auto ping is operating. M8186 is off when auto ping is stopped. For details on auto ping, see "Auto Ping Function" on page 3-18.

■ **M8187: Auto Ping Stop Flag**
While M8187 is on, auto ping stops. 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 Network Settings Trigger**
When M8190 is turned on, the values written to D8304 to D8323 are set as the CPU module IP address.
The IP address is not set just by changing the values of D8304 to D8323. For details on changing the CPU module network settings, 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.

■ **M8200 to M8207: User Communication Receive Instruction Cancel Flag**
When M8200 to M8207 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

■ **M8211: Send E-mail Server Settings Initialization**
When M8211 is turned on, the send E-mail server settings are initialized.

■ **M8212 to M8221: 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

■ **M8222 to M8231: 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
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.

■ **M8300 to M8304: J1939 Communication**
Special data registers used in J1939 communication. For details, see "Special Internal Relay Allocations" on page 8-4.

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 Adapter Information		Power-up	R
D8031	Optional Cartridge 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 to D8051	**— Reserved —**		—	—
D8052	J1939 Communication Error Code		Every scan	R/W
D8053 to D8055	**— Reserved —**		—	—
D8056	Battery Voltage		—	R

Device Address	Description		Update Timing	R/W
D8057	Analog Potentiometer (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 3)		Every scan	R
D8105	RS232C DSR Input Control Signal Option (Port 1 to 3)		When sending/ receiving data	R
D8106	RS232C DTR Output Control Signal Option (Port 1 to 3)		When sending/ receiving data	R
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

Device Address	Description			Update Timing	R/W
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 Cartridge I/O AI2/AQ2			Every scan	R
D8171	Analog Cartridge I/O AI3/AQ3			Every scan	R
D8172	Analog Cartridge Status AI2/AQ2			Every scan	R
D8173	Analog Cartridge Status AI3/AQ3			Every scan	R
D8174	Analog Cartridge I/O AI4/AQ4			Every scan	R
D8175	Analog Cartridge I/O AI5/AQ5			Every scan	R
D8176	Analog Cartridge Status AI4/AQ4			Every scan	R
D8177	Analog Cartridge Status AI5/AQ5			Every scan	R
D8178	Analog Cartridge I/O AI6/AQ6			Every scan	R
D8179	Analog Cartridge I/O AI7/AQ7			Every scan	R
D8180	Analog Cartridge Status AI6/AQ6			Every scan	R
D8181	Analog 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	Preset 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	Preset Value	—	R/W
D8203		Low Word		—	R/W
D8204 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	Preset 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
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

Device Address	Description			Update Timing	R/W
D8230 D8231	**— Reserved —**			—	—
D8232	High-speed Counter (Group 5/I6)	High Word	Preset Value	—	R/W
D8233		Low Word		—	R/W
D8234	High-speed Counter (Group 3/I3)	High Word	Preset Value	—	R/W
D8235		Low Word		—	R/W
D8236	High-speed Counter (Group 4/I4)	High Word	Preset 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 D8359	**— 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 to 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 D8303	**— Reserved —**			—	—

Device Address	Description	Update Timing	R/W
D8304	CPU Module IP Address (Write-only)	—	W
D8305		—	W
D8306		—	W
D8307		—	W
D8308	CPU Module Subnet Mask (Write-only)	—	W
D8309		—	W
D8310		—	W
D8311		—	W
D8312	CPU Module Default Gateway (Write-only)	—	W
D8313		—	W
D8314		—	W
D8315		—	W
D8316	CPU Module Preferred DNS Server (Write-only)	—	W
D8317		—	W
D8318		—	W
D8319		—	W
D8320	CPU Module Alternate DNS Server (Write-only)	—	W
D8321		—	W
D8322		—	W
D8323		—	W
D8324	CPU Module 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 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 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 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
D8342	CPU Module 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 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

Device Address	Description	Update Timing	R/W
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
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

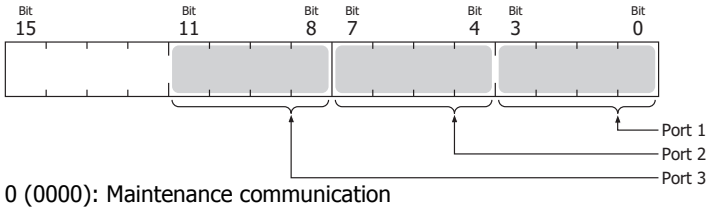
Device Address	Description		Update Timing	R/W
D8416 to D8428	**— Reserved —**		**—**	**—**
D8429	HMI Module Connection Information Reference	Connection No.	—	R/W
D8430		Connected IP Address	Every 1 s	R
D8431			Every 1 s	R
D8432			Every 1 s	R
D8433			Every 1 s	R
D8434		Connected Port No.	Every 1 s	R
D8435 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		—	R
D8458 to D8469	**— Reserved —**		**—**	**—**
D8470	Expansion Module Slot 1 Information	Type ID/Status	—	R
D8471		System Software Version	—	R
D8472	Expansion Module Slot 2 Information	Type ID/Status	—	R
D8473		System Software Version	—	R
D8474	Expansion Module Slot 3 Information	Type ID/Status	—	R
D8475		System Software Version	—	R
D8476	Expansion Module Slot 4 Information	Type ID/Status	—	R
D8477		System Software Version	—	R
D8478	Expansion Module Slot 5 Information	Type ID/Status	—	R
D8479		System Software Version	—	R
D8480	Expansion Module Slot 6 Information	Type ID/Status	—	R
D8481		System Software Version	—	R
D8482	Expansion Module Slot 7 Information	Type ID/Status	—	R
D8483		System Software Version	—	R
D8484	Expansion Module Slot 8 Information	Type ID/Status	—	R
D8485		System Software Version	—	R
D8486	Expansion Module Slot 9 Information	Type ID/Status	—	R
D8487		System Software Version	—	R
D8488	Expansion Module Slot 10 Information	Type ID/Status	—	R
D8489		System Software Version	—	R
D8490	Expansion Module Slot 11 Information	Type ID/Status	—	R
D8491		System Software Version	—	R

Device Address	Description		Update Timing	R/W
D8492	Expansion Module Slot 12 Information	Type ID/Status	—	R
D8493		System Software Version	—	R
D8494	Expansion Module Slot 13 Information	Type ID/Status	—	R
D8495		System Software Version	—	R
D8496	Expansion Module Slot 14 Information	Type ID/Status	—	R
D8497		System Software Version	—	R
D8498	Expansion Module Slot 15 Information	Type ID/Status	—	R
D8499		System Software Version	—	R

Supplementary Descriptions of the Special Data Registers Related to the Communication Functions

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

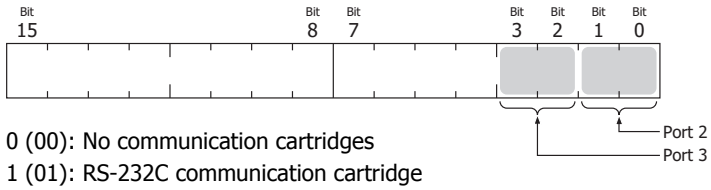
This register indicates communication mode information for Port 1 to Port 3.
The allocation of communication ports in the device (bit assignment) is as follows.



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

■ **D8030: Communication Adapter 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): Free

■ **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.

■ **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.

■ **D8100, D8102, D8103: Slave Number (Port 1 to 3)**

The slave number is written to these registers when the Port 1 to 3 communication mode is maintenance communication, Modbus RTU slave, or data link communication. The slave number can be changed by changing the value of D8100, D8102, and D8103 in Function Area Settings

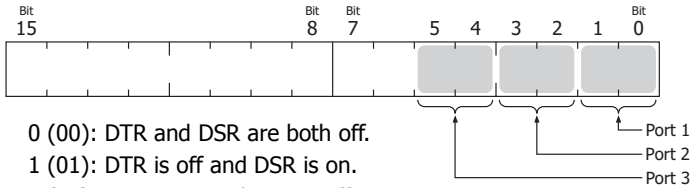
D8100 = Port 1 Slave No.
D8102 = Port 2 Slave No.
D8103 = Port 3 Slave No.

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: Control Signal Status (Port 1 to 3)**

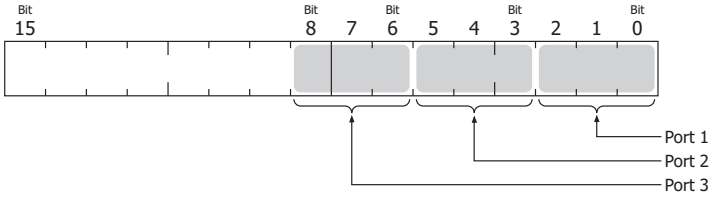
The signal statuses of the DSR and DTR controls lines are written to this register. This register is updated in END processing when stopped and while running. The allocation of communication ports in the device (bit assignment) is as follows.



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.

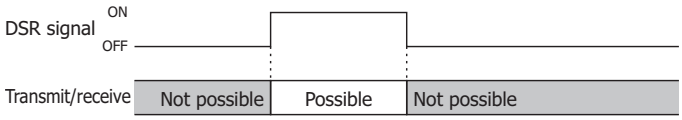
■ **D8105: RS232C DSR Input Control Signal Option (Port 1 to 3)**

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.

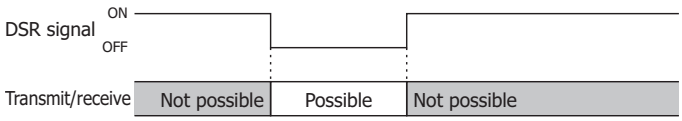


0 (000): The DSR signal status is not used for FC6A Series MicroSmart transmission control. Use this status when DSR signal control is not required.

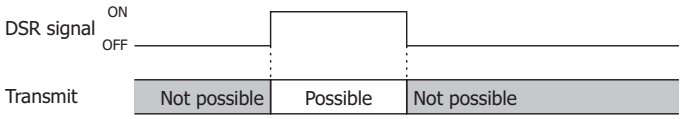
1 (001): When the DSR signal is on, the FC6A Series MicroSmart can transmit and receive.



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



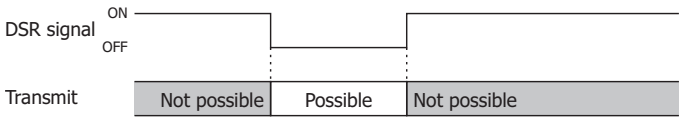
3 (011): When the DSR signal is on, transmission is possible (reception is always possible).



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.

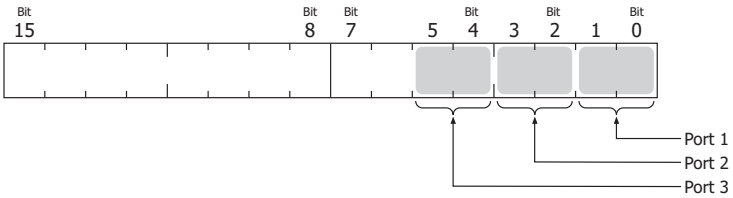


5 or higher: The operation is the same as the setting "000".

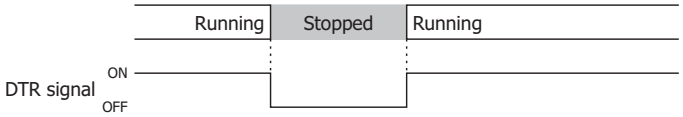
■ **D8106: RS232C DTR Output Control Signal Option (Port 1 to 3)**

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.

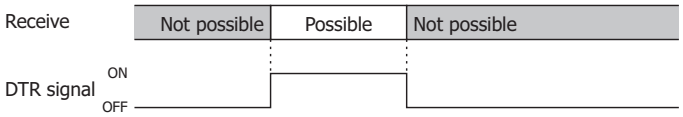


0 (00): The signal is on when the FC6A Series MicroSmart is set to run and off when stopped.
While running, the signal is always on regardless of transmitting or receiving data. Set this value when it is necessary to indicate the run status.



1 (01): Always off.

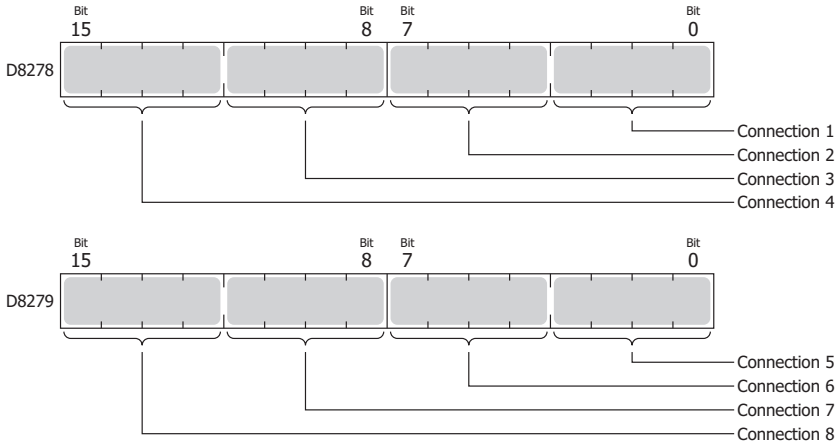
2 (10): Set this value when performing flow control for received data. The signal is on when data from the connected device can be received.
The signal is off when data cannot be received.



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

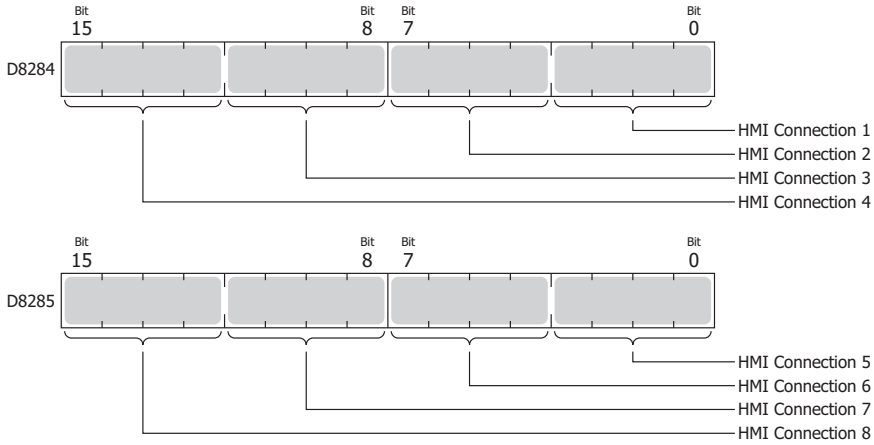
■ **D8278, D8279: Communication Mode Information (Client Connection) (Connection 1 to 8)**

D8278 = Indicates the communication mode of connections 1 through 4.
D8279 = Indicates the communication mode of connections 5 through 8.
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
- 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

■ **D8304 to D8307: CPU Module 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 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 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 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 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 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 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 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 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 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 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: 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

■ **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 3-8.

■ **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 3-8.

■ **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 3-8.

■ **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**
Detailed error information for the EMAIL instruction is written to this register.
For details, see "Send E-mail Function" on page 10-1.

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.	3-8	
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.	3-11	
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.	3-16	Remote host list
Auto Ping Function	Configures the auto ping function to monitor the network connection status of remote hosts.	3-18	

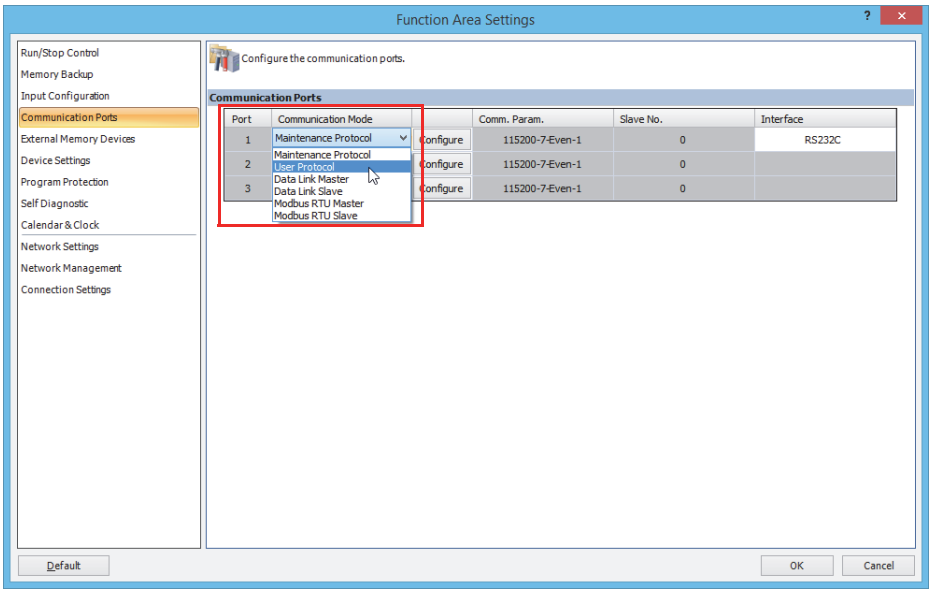
Communication Port Settings

This section describes how to configure the communication ports when using serial port 1 (port 1) or a communication cartridge (port 2 or port 3) connected to cartridge slot 1 or cartridge slot 2 to communicate with the communication device.

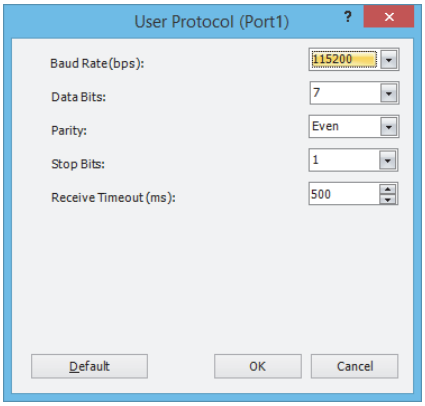
Programming WindLDR

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 or the HMI-Ethernet port to connect the FC6A Series MicroSmart to a network.

Description

In order for the FC6A Series MicroSmart to connect to an Ethernet network, the FC6A Series MicroSmart network settings (FC6A Series MicroSmart IP address, subnet mask, default gateway, and DNS server addresses) must be acquired or configured according to the specification method.

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 in special data registers D8304 to D8315. These values are set on the FC6A Series MicroSmart when the FC6A Series MicroSmart power is turned on and when special internal relay M8190 (CPU module network settings changed trigger) 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 All-in-One Type 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 in special data registers D8316 to D8323. These values are set on the FC6A Series MicroSmart when the FC6A Series MicroSmart power is turned on and when special internal relay M8190 (CPU module network settings changed trigger) 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 All-in-One Type User's Manual".

Network settings by special data registers

When **IP Settings** is set to "Use special data registers to configure the IP address" and **DNS Settings** is set to "Use special data registers to configure the DNS server addresses" in the function area settings, the values that are set on the FC6A Series MicroSmart when the FC6A Series MicroSmart power is turned on and when special internal relay M8190 (CPU module network settings changed trigger) is turned on are stored in the following special data registers.

Special Data Register	Description	Read/Write
D8304-D8307	The FC6A Series MicroSmart IP address setting value. Example: When the IP address is 192.168.0.1 D8304=192, D8305=168, D8306=0, D8307=1	R/W
D8308-D8311	The FC6A Series MicroSmart subnet mask setting value. Example: When the subnet mask is 255.255.255.0 D8308=255, D8309=255, D8310=255, D8311=0	R/W
D8312-D8315	The FC6A Series MicroSmart default gateway setting value. Example: When the default gateway is 192.168.0.24 D8312=192, D8313=168, D8314=0, D8315=24	R/W
D8316-D8319	The setting value of the preferred DNS server address used by the FC6A Series MicroSmart. Example: When the preferred DNS address is 192.168.0.100 D8316=192, D8317=168, D8318=0, D8319=100	R/W
D8320-D8323	The setting value of the alternate DNS server address used by the FC6A Series MicroSmart. Example: When the alternate DNS address is 192.168.0.101 D8320=192, D8321=168, D8322=0, D8323=101	R/W

Checking the network settings

The current FC6A Series MicroSmart IP address, subnet mask, default gateway, and DNS server addresses can be checked with special data registers.

Special Data Register	Description	Read/Write
D8330-D8333	The FC6A Series MicroSmart IP address current value. Example: When the IP address is 192.168.0.1 D8330=192, D8331=168, D8332=0, D8333=1	R
D8334-D8337	The FC6A Series MicroSmart subnet mask current value. Example: When the subnet mask is 255.255.255.0 D8334=255, D8335=255, D8336=255, D8337=0	R
D8338-D8341	The FC6A Series MicroSmart default gateway current value. Example: When the default gateway is 192.168.0.24 D8338=192, D8339=168, D8340=0, D8341=24	R
D8342-D8345	The current value of the preferred DNS server address used by the FC6A Series MicroSmart. Example: When the preferred DNS address is 192.168.0.100 D8342=192, D8343=168, D8344=0, D8345=100	R
D8346-D8349	The current value of the alternate DNS server address used by the FC6A Series MicroSmart. Example: When the alternate DNS address is 192.168.0.101 D8346=192, D8347=168, D8348=0, D8349=101	R

Network settings by HMI module special data registers

When **IP Settings** is set to "Use special data register to configure the IP address" and **DNS Settings** is set to "Use special data registers to configure the DNS server addresses" in the function area settings, the values that are set on the HMI module when the FC6A Series MicroSmart power is turned on and when special internal relay M8184 (HMI module network settings changed trigger) is turned on are stored in the following special data registers.

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

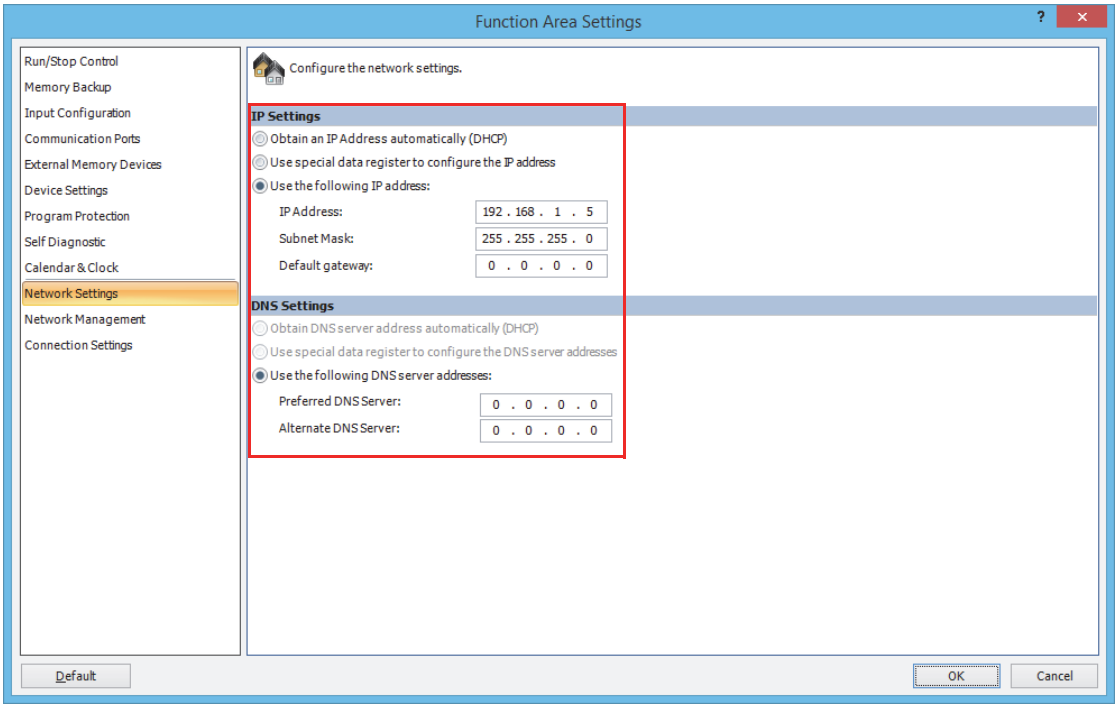
Checking the HMI module network settings

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

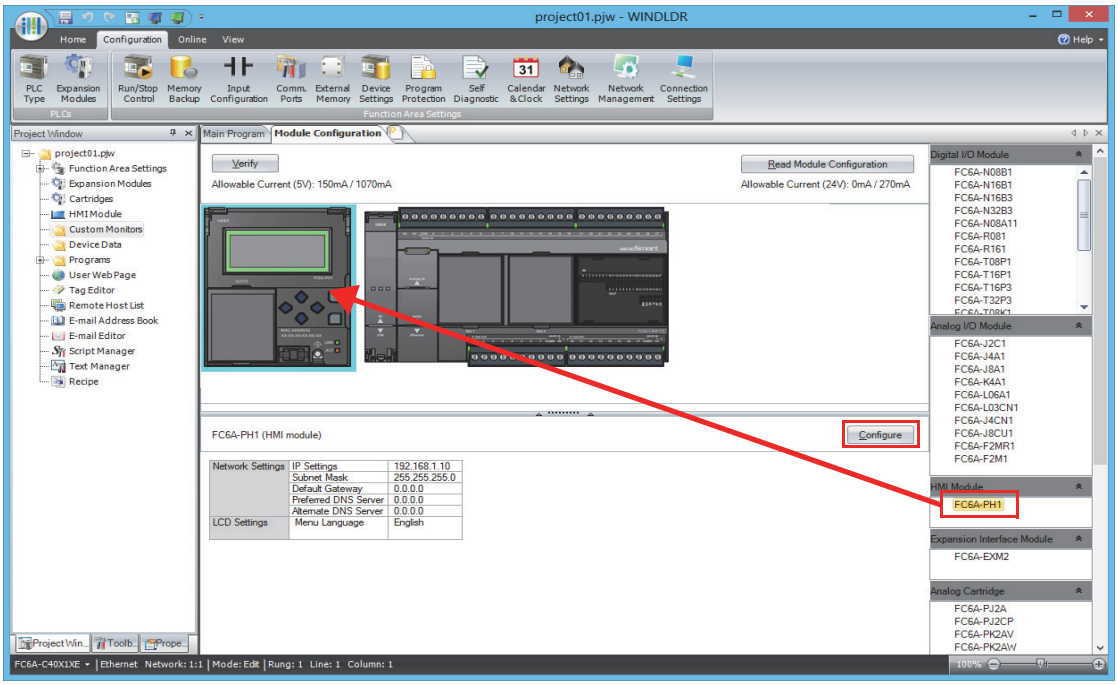
Programming WindLDR

1. Select **Configuration** from the WindLDR menu bar, and then click **Network Settings**.
   The **Function Area Settings** dialog box appears.
2. Configure **IP Settings** and **DNS Settings**.

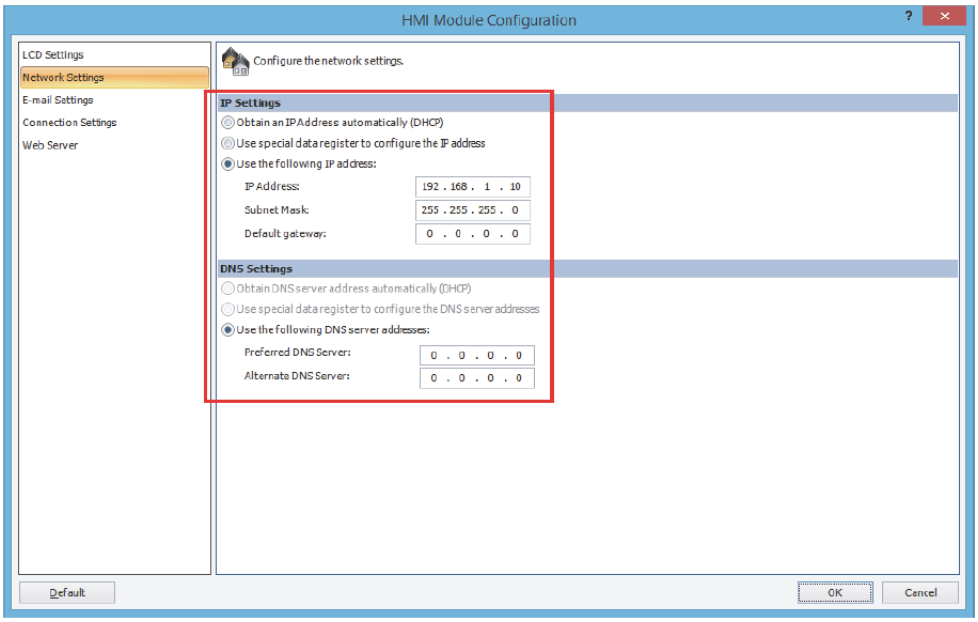


3. Click **OK**.
4. Use the Module Configuration Editor to configure the HMI module network settings.
   On the **Configuration** tab, in the **PLCs** group, select **Expansion Modules**.
5. Click the inserted HMI module in the module configuration area and click **Configure**.
   The **HMI Module Configuration**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

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.

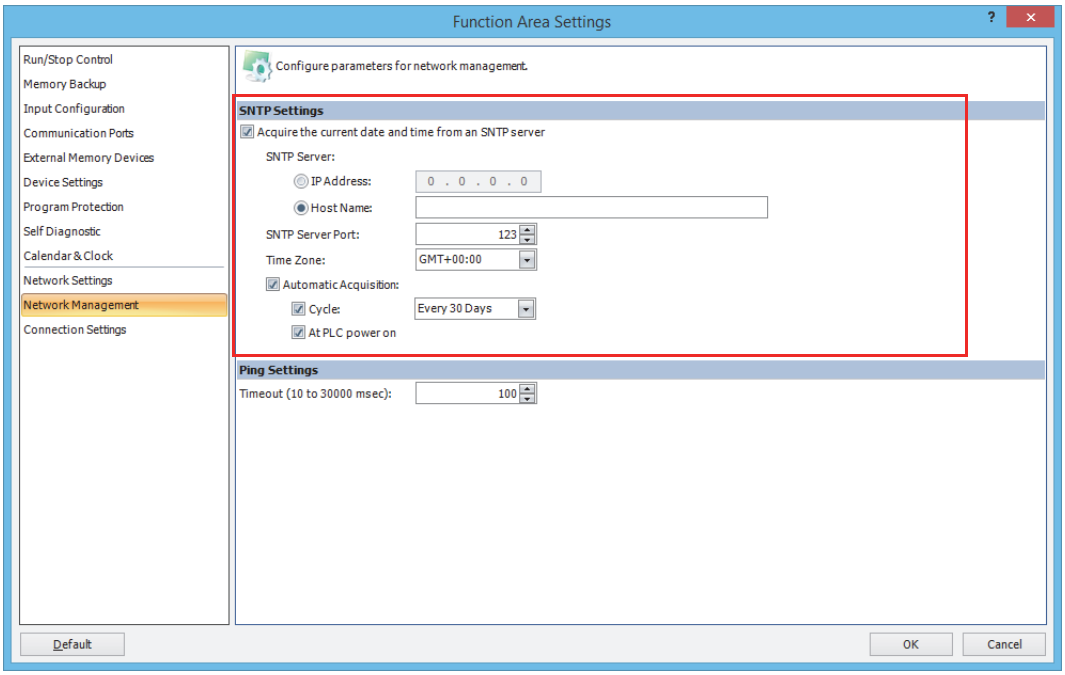
Description

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 **Network Management**.
   The **Function Area Settings** dialog box is displayed.
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**.

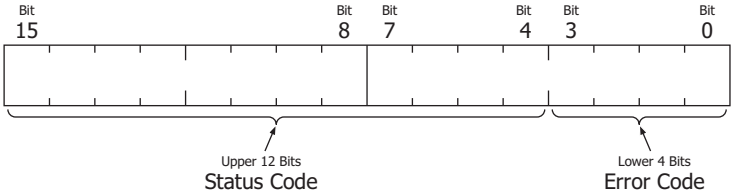
This concludes configuring the settings.

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

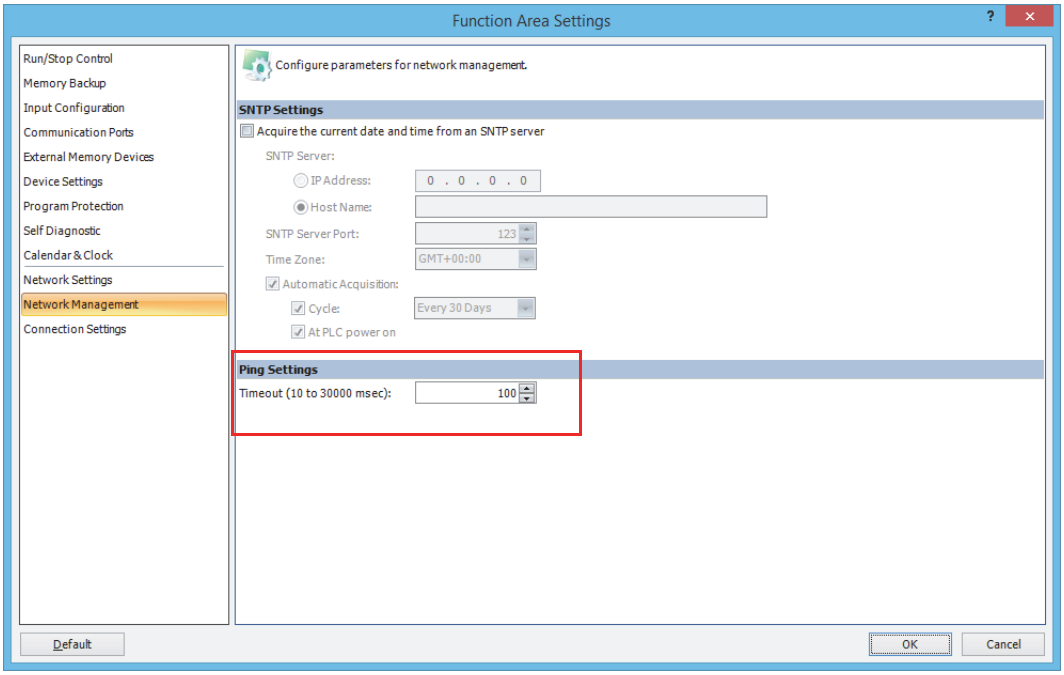
This section describes the ping timeout setting for the PING instruction and when the auto ping function is being executed.

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 **Network Management**.
   The **Function Area Settings** dialog box is displayed.
2. With **Timeout (10 to 30000 msec)**, set the PING instruction timeout between 10 and 30,000 ms.
   The default is 10 ms.



3. Click **OK**.

This concludes configuring the settings.

Connection Settings

This section describes the settings for client/server communication during TCP/IP communication using Ethernet port 1 or the HMI-Ethernet port of the FC6A Series MicroSmart.

Applications

The FC6A Series MicroSmart is capable of maintenance communication, user communication, and Modbus TCP communication using a maximum of eight connections over Ethernet port 1.

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

Each connection can be selected as maintenance communication server, user communication server, user communication client, Modbus TCP server, Modbus TCP client, or unused, and a maximum of eight connections can be configured.

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 Mode	Overview	Reference
Maintenance communication server (Default)	This mode allows monitoring and changing devices and user program uploading and downloading from WindLDR.	4-6
User communication server	This mode enables communication with client devices according to the ETXD (Ethernet user communication transmit) instruction and the ERXD (Ethernet user communication receive) instruction.	5-40
User communication client	This mode enables communication with server devices according to the ETXD (Ethernet user communication transmit) instruction and the ERXD (Ethernet user communication receive) instruction.	5-37
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.	6-24
Modbus TCP client	Modbus TCP server-compatible device data monitoring and modifications can be executed from Modbus TCP clients.	6-19
Unused	The connection is not used.	—

Connection Status and Connected IP Address

The connection status of connections with remote hosts can be confirmed with special internal relays M8212 to M8221. 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		
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 and Modbus TCP 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		

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		

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

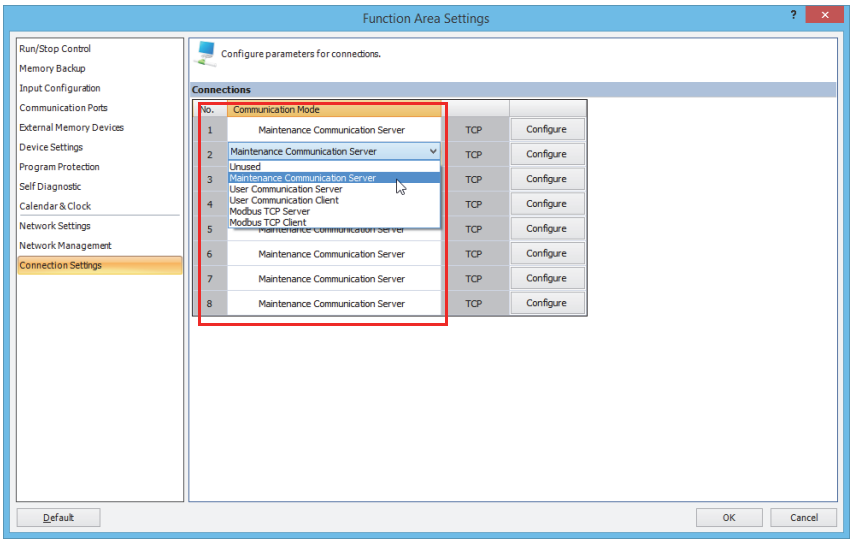
Connected IP Address

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, D8350=aaa, D8351=bbb, D8352=ccc, D8353=ddd	R

Programming WindLDR

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

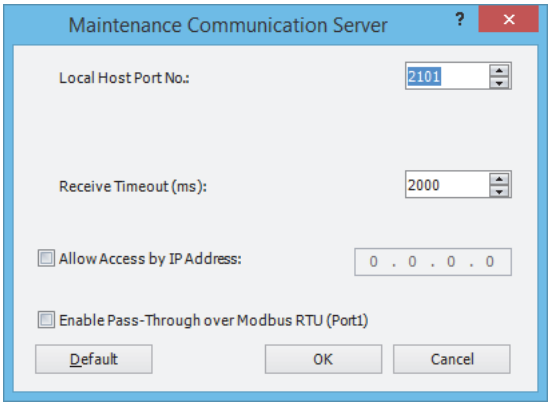
2. Click **Communication Mode** for the port to be used and select communication mode of the connection to use.



Configure the communication mode allocated to a maximum of eight connections 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.

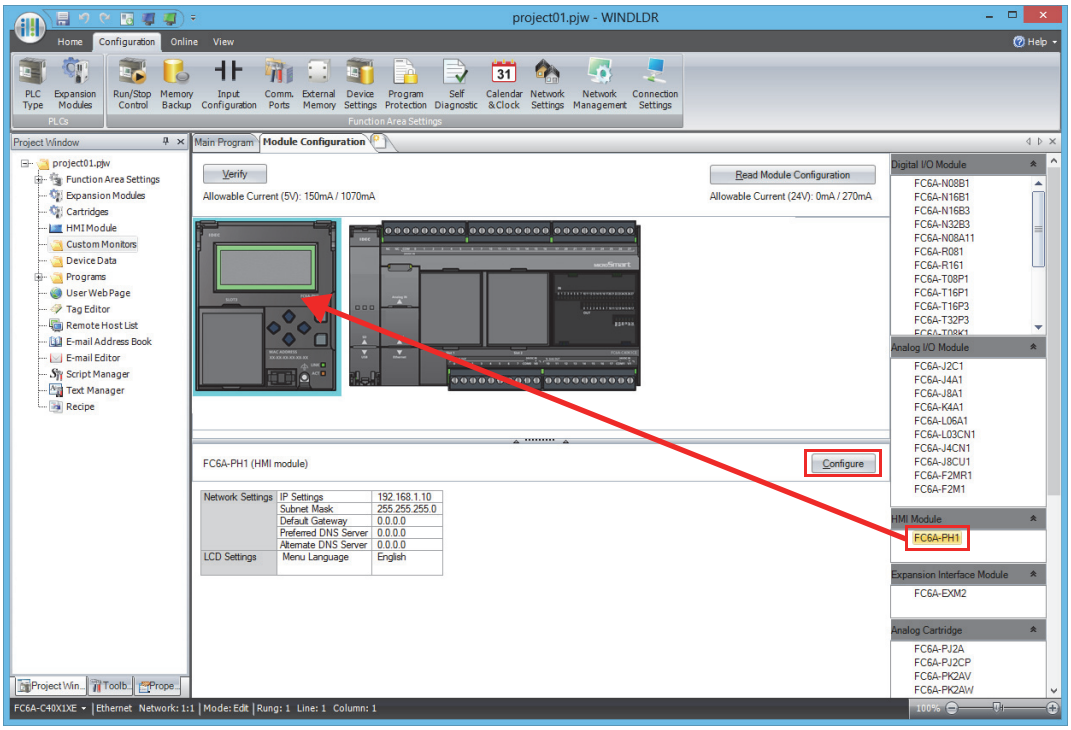


4. Click **OK**.

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

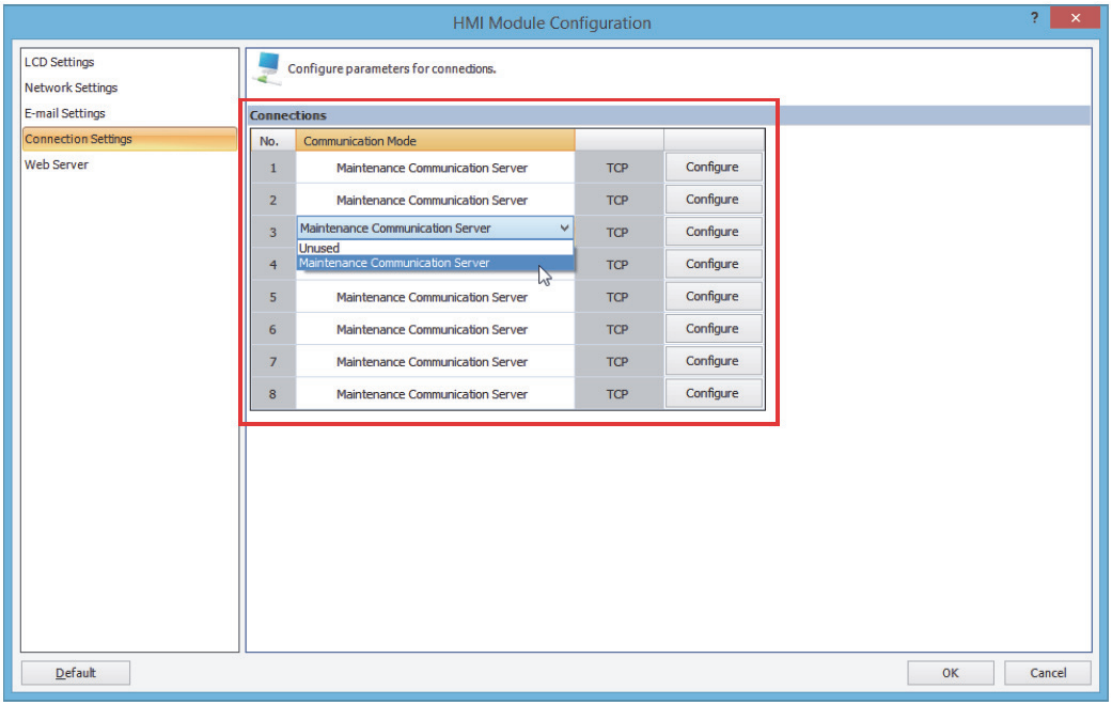
6. Click the inserted HMI module in the module configuration area and click **Configure**.
   The **HMI Module Configuration** dialog box is displayed.

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

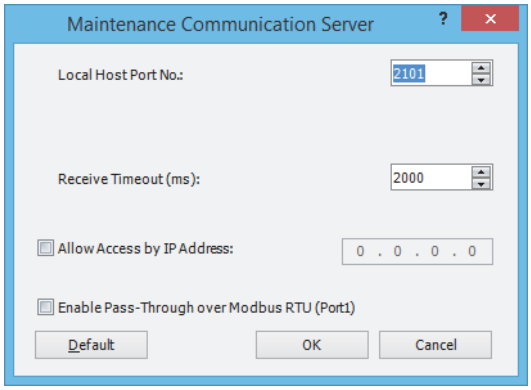


7. Click **Connection Settings**.

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**.

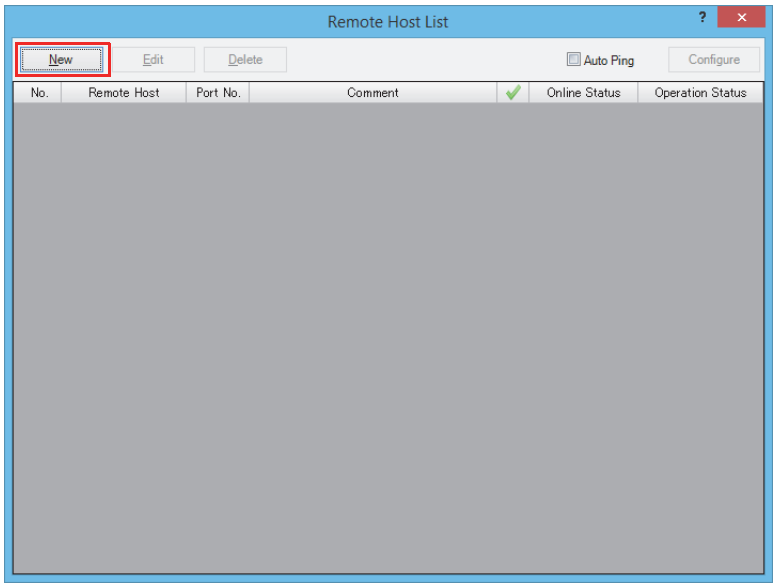
When a remote host is specified with an IP address, and the FC6A Series MicroSmart establishes connection with the remote host that has the specified IP address and the corresponding port number, then communication is started.

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.

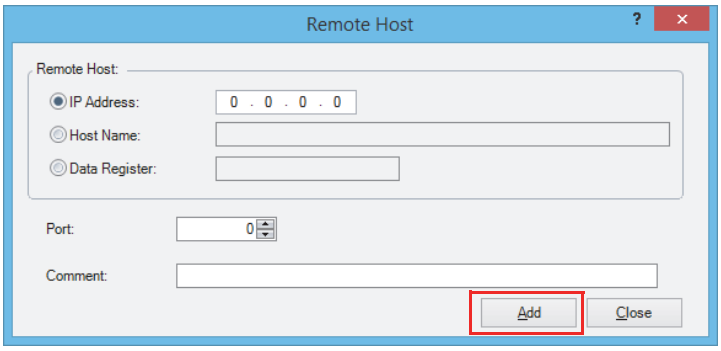
For details about DNS server settings, see Network Settings in the **Function Area Settings**.

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.



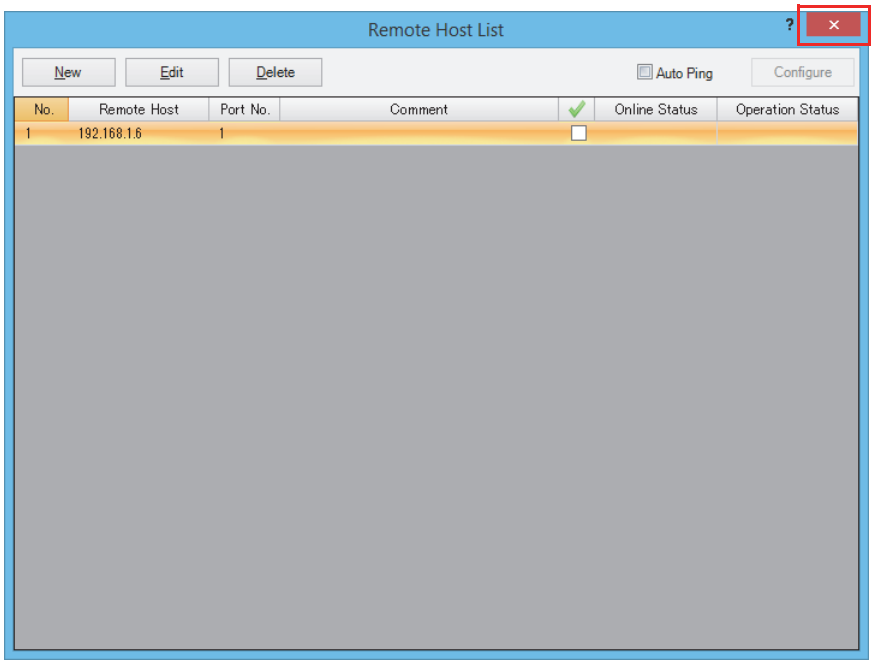
The remote host is composed of the following items.

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 Number	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.

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 3-18.

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.

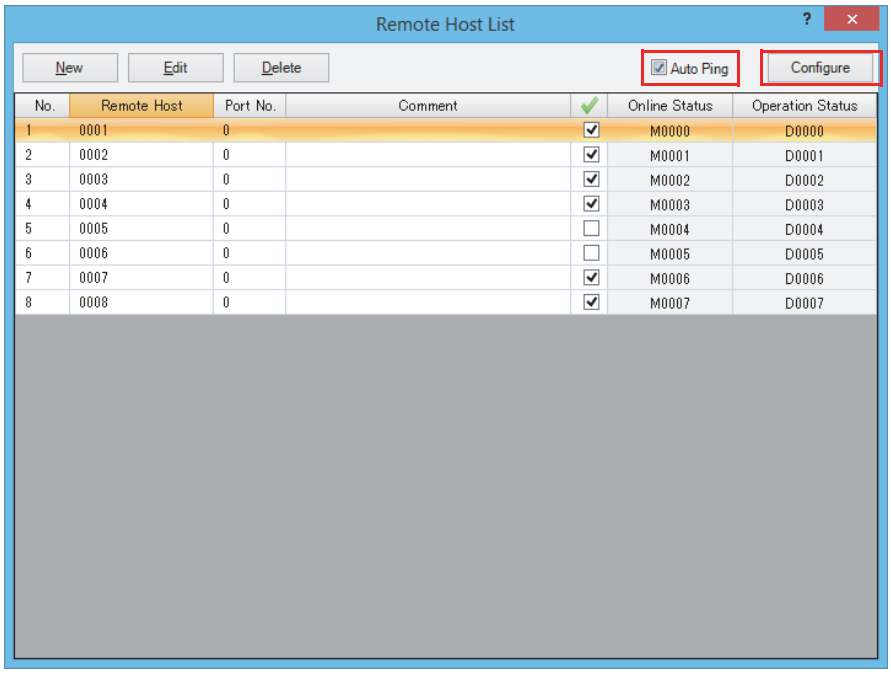
The remote host is specified with the remote host list.

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. Auto ping is stopped while M8187 (auto ping stop flag) is on. When M8187 is turned off, auto ping execution starts.

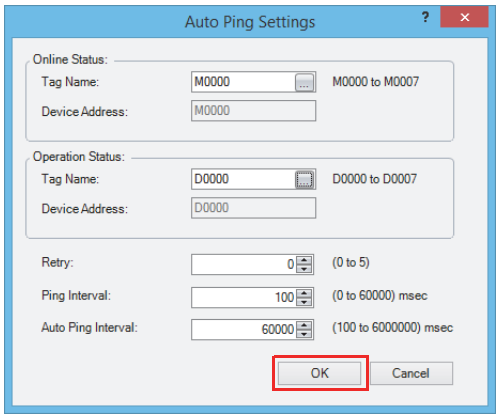
The results of the auto ping function can be linked with Modbus TCP client request processing. For linking with Modbus TCP clients, see "Modbus TCP Client" on page 6-19.

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.



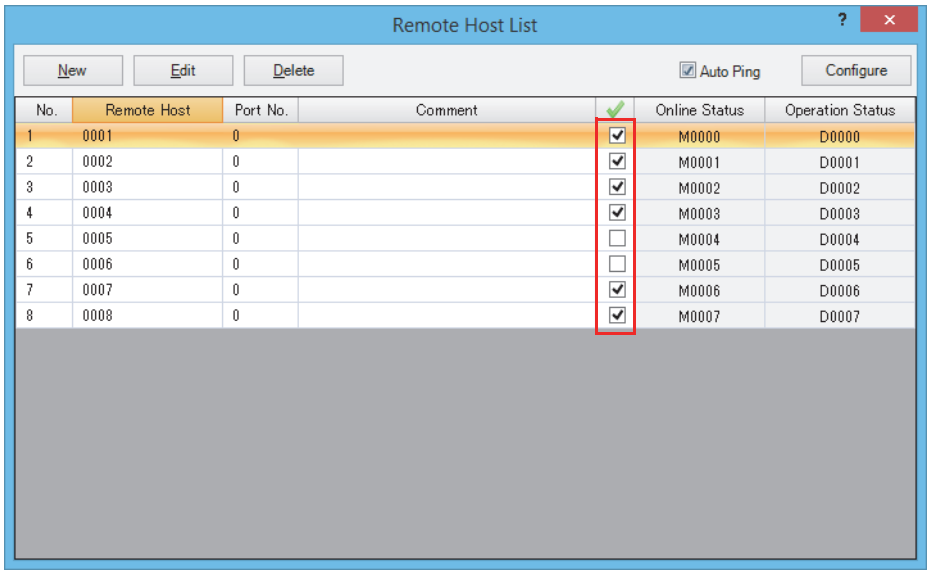
2. Specify the auto ping function items and click **OK**.



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. When M8187 (auto ping stop flag) is turned from off to on, or when the user program is changed and the auto ping function is set from "Used" to "Not used", the internal relay on/off status immediately before that is kept. When M8187 (auto ping stop flag) is turned from on to off, or when the user program is changed and the 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 3-20. 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.

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



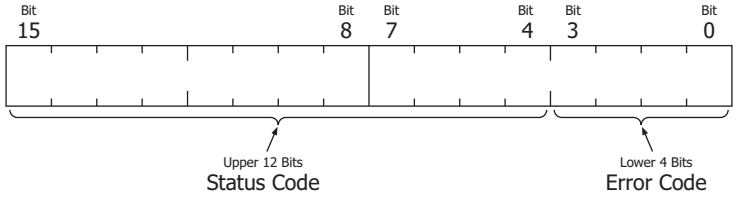
4. Click **Close**.

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.



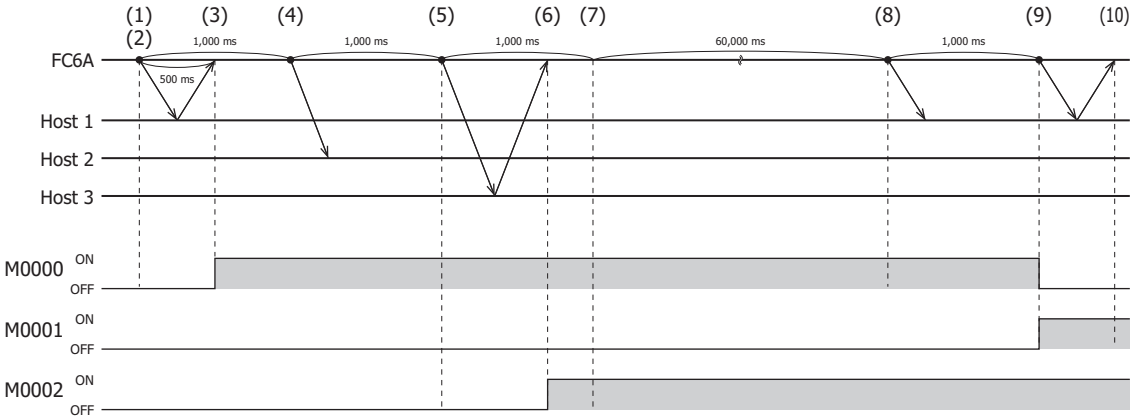
Status Code	Operation Status
16 (000000010000)	Status after ping executed and before the packet is sent
32 (000000100000)	Status after packet send processing has completed and waiting for a response from the host
64 (000001000000)	Status where receiving the response for the packet completed normally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

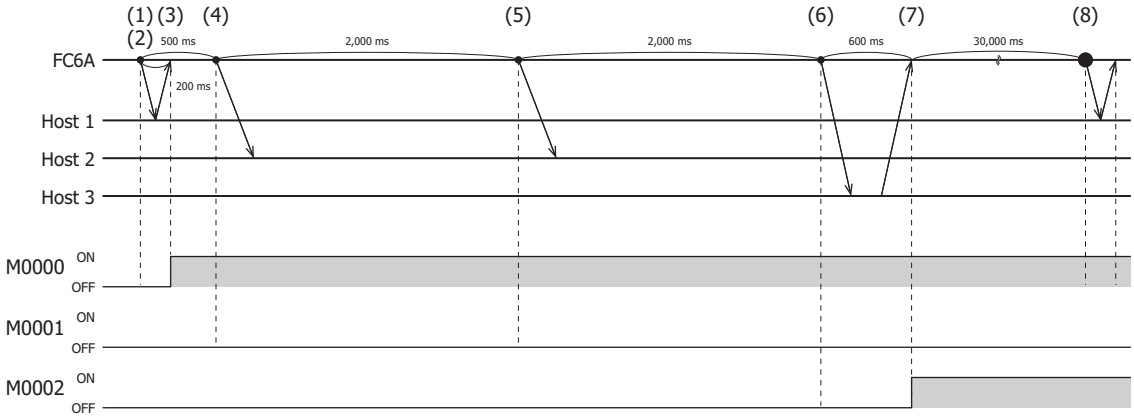


(1) : Auto ping operation starts
(2) to (3) : Response from Host 1 within the timeout
(4) : No response from Host 2 within the timeout
(5) to (6) : Response from Host 3 within the timeout
(7) to (8) : Auto Ping Interval
(8) : No response from Host 1 within the timeout
(9) to (10) : Response from Host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

Maintenance communication of FC6A Series MicroSmart is available on USB port, port1, Ethernet port1, port 2 and port 3, and HMI-Ethernet 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 All-in-One Type 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 All-in-One Type 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 All-in-One Type User's Manual".

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

Communication Ports Used For Maintenance Communication

Supported models, ports, and slots are as follows.

Port	All-in-One Type			CAN J1939 All-in-One Type
	16-I/O Type	24-I/O Type	40-I/O Type	
USB Port	Yes	Yes	Yes	Yes
Port 1	Yes	Yes	Yes	—
Ethernet Port 1	Yes	Yes	Yes	Yes
Port 2	Yes	Yes	Yes	Yes
Port 3	—	—	Yes	Yes
HMI-Ethernet Port	Yes	Yes	Yes	Yes

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

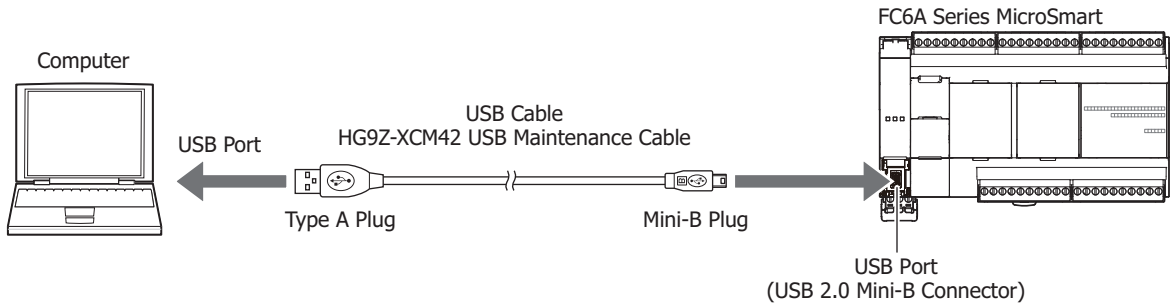
Communication Method		USB Port	Port 1	Ethernet Port 1	Port 2 and Port 3	HMI-Ethernet Port
Maintenance Communication	Downloading system software	Yes	Yes	Yes	No	Yes
	Download/upload user programs	Yes	Yes	Yes	No	Yes
	Run-time download	Yes	Yes	Yes	No	Yes
	Monitoring/changing devices	Yes	Yes	Yes	Yes	Yes

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

Communication Port	Standard/Option	No. of Ports	Communication Settings
USB Port	Standard	1	None
Port 1	All-in-One Type: Standard equipment CAN J1939 All-in-One Type: —	1	Function area settings See "Maintenance Communication via Port 1" on page 4-4.
Ethernet Port 1	Standard	1	Function area settings See "Maintenance Communication via Ethernet Port 1" on page 4-6.
Port 2 and Port 3	Options FC6A-PC1: EIA RS-232C (terminal block) FC6A-PC3: EIA RS-485 (terminal block)	16-I/O type: 1 24-I/O type: 1 40-I/O type: 2	Function area settings See "Maintenance Communication via a Communication Cartridge (Port 2, Port 3)" on page 4-11.
HMI-Ethernet Port	Options FC6A-PH1 (HMI module)	1	Function area settings See "Maintenance Communication via HMI-Ethernet port" on page 4-13

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).



Maintenance Communication Specifications for the USB Port

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

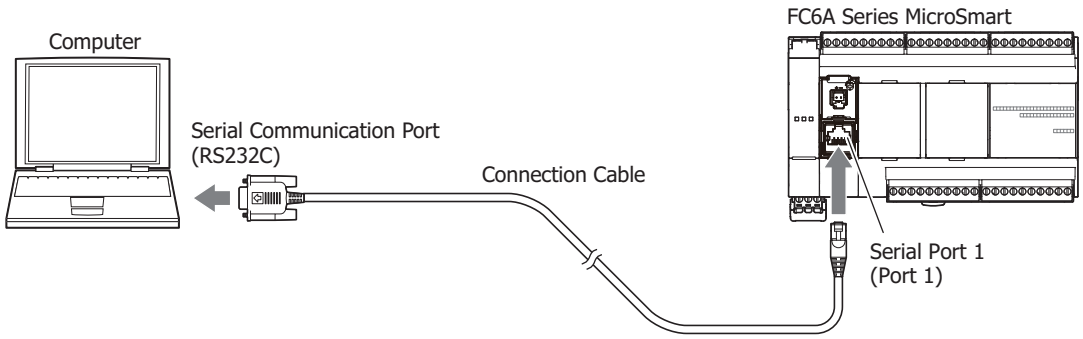
For details on the operation of maintenance communication functions, see the following in the "FC6A Series MicroSmart All-in-One Type User's Manual".

- 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"

Maintenance Communication via Port 1

Using Serial Port 1 on the FC6A Series MicroSmart, it is possible to connect to a computer on which WindLDR is installed or an operator interface, and monitor and change device values, download and upload user programs, and download system software.

For details on communication cables, see Appendix "Cables" in the "FC6A Series MicroSmart All-in-One Type User's Manual".



Port 1 maintenance communication specification

Item	Specifications/Functions
Cable	FC6A-KC2C: O/I communication cable (D-sub 9-pin connector style, cable length: 5 m)
Maintenance communication functions	Monitoring/changing device values Downloading/uploading user programs Downloading system software Run-time download

For details on the operation of maintenance communication functions, see the following in the "FC6A Series MicroSmart All-in-One Type User's Manual".

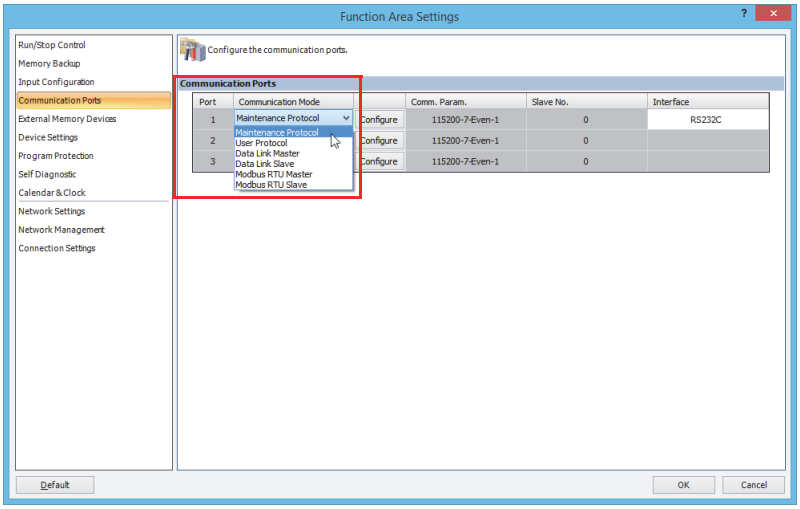
- Monitor/change device values: Chapter 4 "Monitor Operation"
- Downloading/uploading user programs: Chapter 4 "Download Program"
- Downloading system software: Appendix "Upgrade FC6A Series MicroSmart System Software"

Programming WindLDR

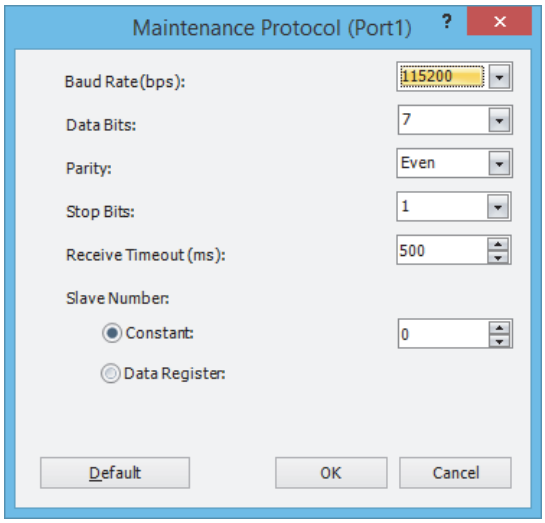
Configure the settings for maintenance communication.

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.

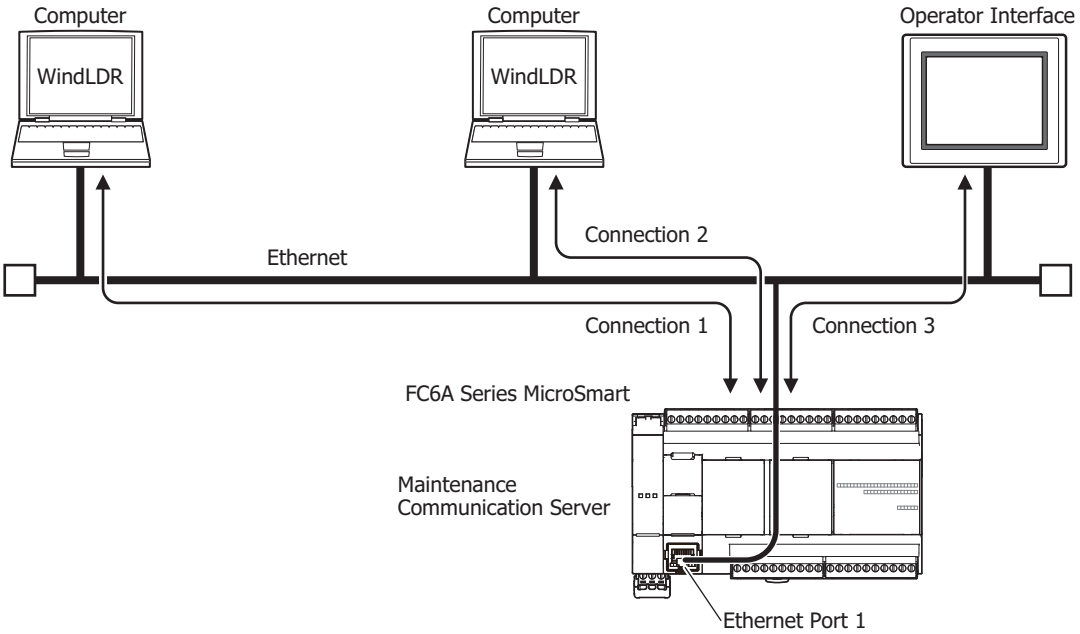
Type	Details
Constant	Set within the range of 0 to 31
Data register	Store the slave numbers 0 to 31 in the following special data registers Port 1: D8100 Port 2: D8102 Port 3: D8103

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

Maintenance Communication via Ethernet Port 1

Network devices such as computers or IDEC operator interfaces can communicate with FC6A Series MicroSmart via Ethernet port 1. External devices on the network can monitor or change the device values and download or upload user programs.

It is possible to use the maintenance communication server and other communications simultaneously by assigning a separate communication function such as a maintenance communication server or Modbus TCP communication to each of the three connections on the FC6A Series MicroSmart.



Maintenance Communication Specifications for the Ethernet Port 1

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

For details on the operation of maintenance communication functions, see the following in the "FC6A Series MicroSmart All-in-One Type User's Manual".

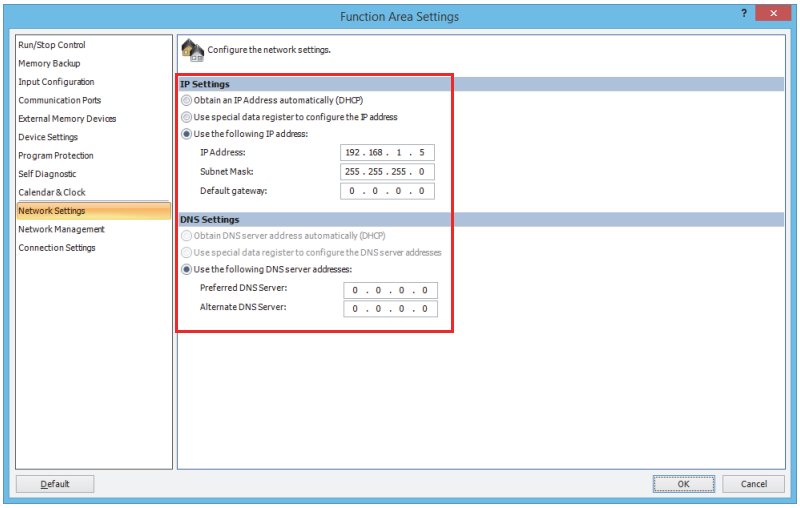
- Monitor/change device values: Chapter 4 "Monitor Operation"
- Download/upload user programs: Chapter 4 "Download Program"
- Downloading system software: Appendix "Upgrade FC6A Series MicroSmart System Software"

Programming WindLDR

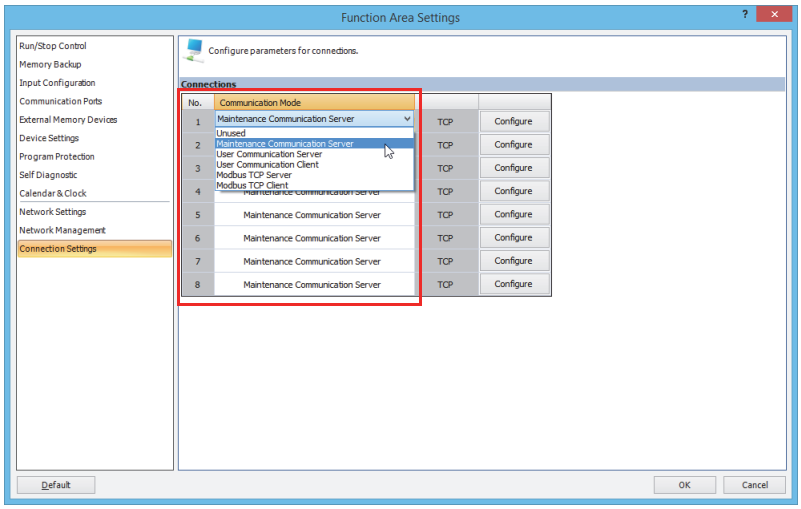
This section describes the procedures to configure the maintenance communication server for the Ethernet port 1 and communicate with the FC6A Series MicroSmart via Ethernet.

Configure Maintenance Communication Server

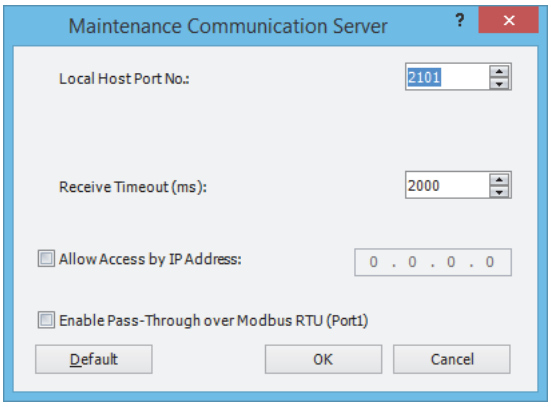
1. From the WindLDR menu bar, select **Configuration** > **Network Settings**.
   The **Function Area Settings** dialog box appears.
2. Enter the IP address, subnet mask, and default gateway.



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.)

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.

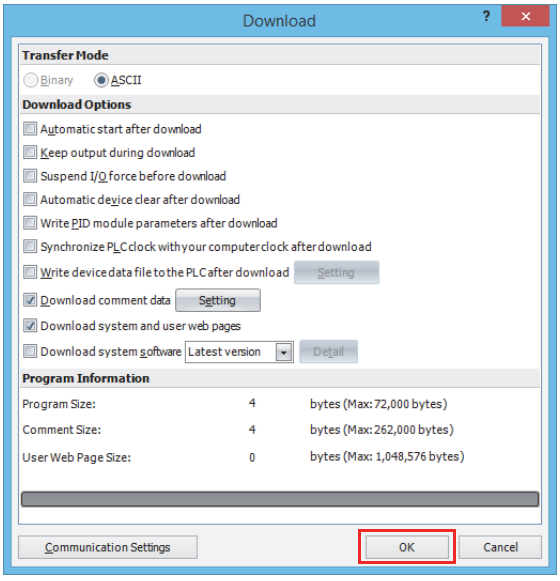
Download User Program and Confirm IP Addresses via USB Port

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.

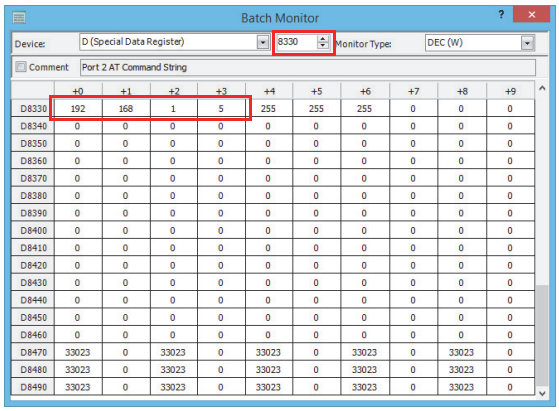
9. Click **OK**.
The user program is downloaded to the FC6A Series MicroSmart.



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.



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D8330	192	168	1	5	255	255	255	0	0	0
D8340	0	0	0	0	0	0	0	0	0	0
D8350	0	0	0	0	0	0	0	0	0	0
D8360	0	0	0	0	0	0	0	0	0	0
D8370	0	0	0	0	0	0	0	0	0	0
D8380	0	0	0	0	0	0	0	0	0	0
D8390	0	0	0	0	0	0	0	0	0	0
D8400	0	0	0	0	0	0	0	0	0	0
D8410	0	0	0	0	0	0	0	0	0	0
D8420	0	0	0	0	0	0	0	0	0	0
D8430	0	0	0	0	0	0	0	0	0	0
D8440	0	0	0	0	0	0	0	0	0	0
D8450	0	0	0	0	0	0	0	0	0	0
D8460	0	0	0	0	0	0	0	0	0	0
D8470	33023	0	33023	0	33023	0	33023	0	33023	0
D8480	33023	0	33023	0	33023	0	33023	0	33023	0
D8490	33023	0	33023	0	33023	0	33023	0	33023	0

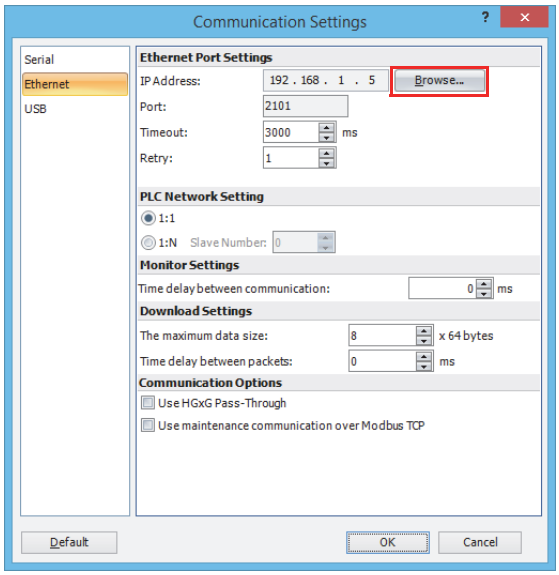
This concludes downloading the user program via the USB port and checking the IP address.

Monitor FC6A Series MicroSmart via the Ethernet Port 1

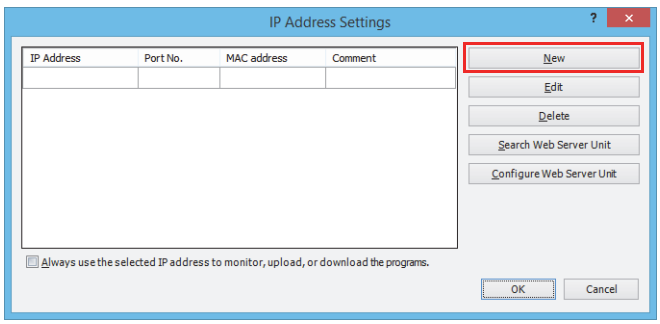
Monitor the FC6A Series MicroSmart via Ethernet using WindLDR.

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

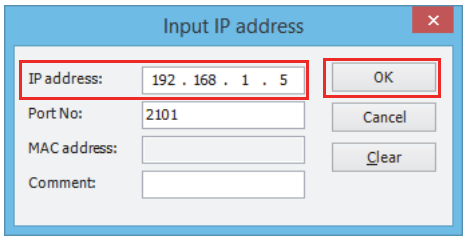
14. Select the **Ethernet** tab and click **Browse**.
The **IP Address Settings** dialog box appears.



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

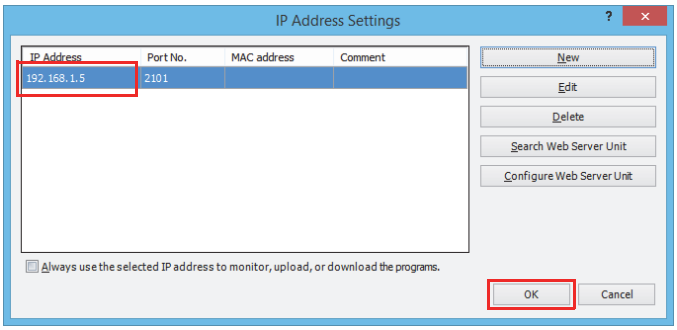


16. Enter the IP address entered in step 2 and click **OK**.



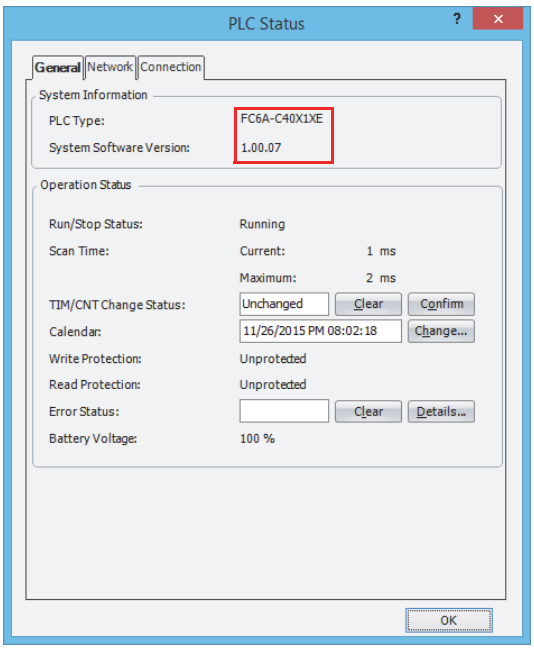
17. From the WindLDR menu bar, select **Online** > **Monitor** > **Monitor**.
The **IP Address Settings** dialog box appears.

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.

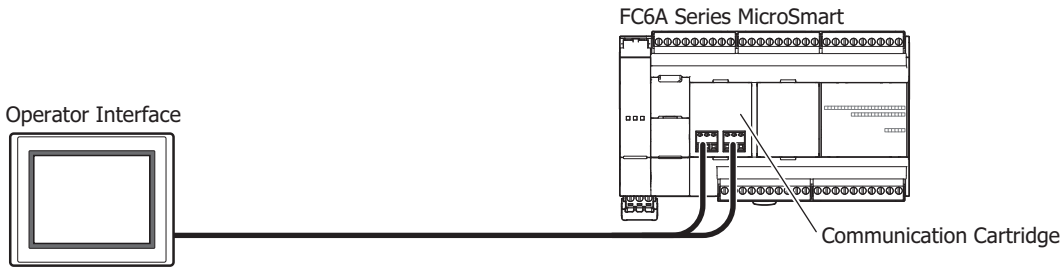


Configuring the initial Ethernet setup for the maintenance communication server is now complete. You can download and upload user programs and monitor and change device values via Ethernet.

Maintenance Communication via a Communication Cartridge (Port 2, Port 3)

By attaching a communication cartridge to a cartridge slot on the FC6A Series MicroSmart, it is possible to connect to a computer or an operator interface with a RS-232C or RS-485 port and to monitor and change device values.

For the communication cable, see Appendix "Upgrade FC6A Series MicroSmart System Software" in the "FC6A Series MicroSmart All-in-One Type User's Manual".



Communication cartridge maintenance communication specifications

Item	Specifications/Functions
Communication Cartridge	FC6A-PC1: RS232C communication cartridge FC6A-PC3: RS485 communication cartridge
Cable	FC6A-PC1: RS232C: Shielded multicore FC6A-PC3: Shielded twisted-pair
Maintenance Communication Function	Monitoring/changing device values

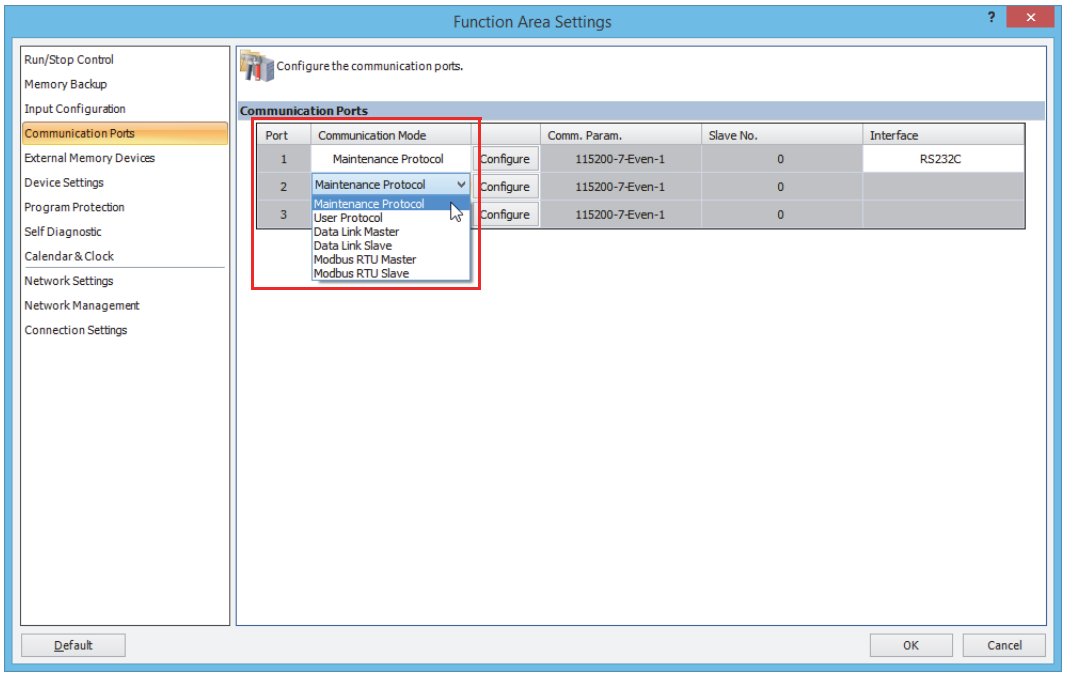
For details on the operation of maintenance communication functions, see the following in the "FC6A Series MicroSmart All-in-One Type User's Manual".

- Monitoring/changing device values: Chapter 4 "Monitor Operation"

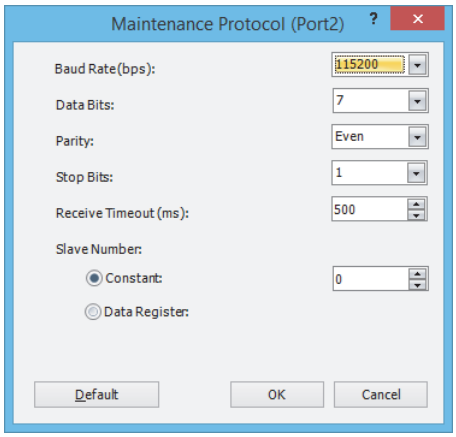
Programming WindLDR

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 or port 3 and select **Maintenance Protocol**.
   The **Maintenance Protocol (Port 2)** or **Maintenance Protocol (Port 3)** 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

* Values not in parentheses are the default settings.

4. Click **OK**.

This concludes configuring the maintenance communications settings.

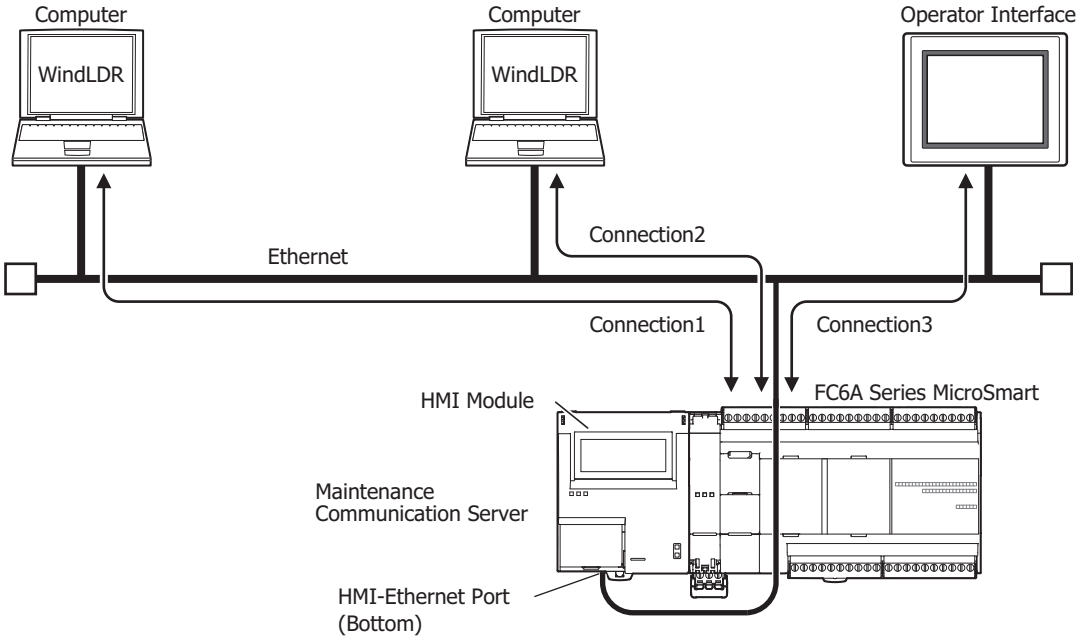
Maintenance Communication via HMI-Ethernet port

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 (up to eight connections), which is equipped as standard on the CPU module of the FC6A Series MicroSmart.

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



HMI-Ethernet Port 1 maintenance communication specification

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

For details on the operation of maintenance communication functions, see the following in the "FC6A Series MicroSmart All-in-One Type User's Manual".

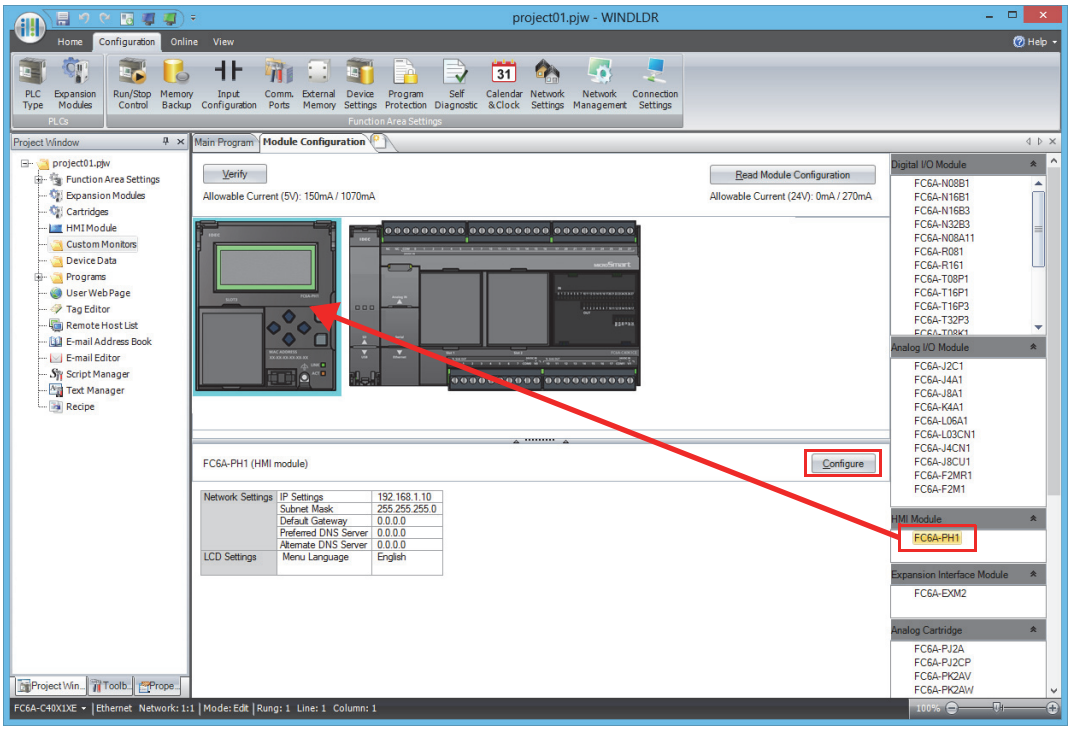
- Monitor/change device values: Chapter 4 "Monitor Operation"
- Download/upload user programs: Chapter 4 "Download Program"
- Downloading system software: Appendix "Upgrade FC6A Series MicroSmart System Software"

Programming WindLDR

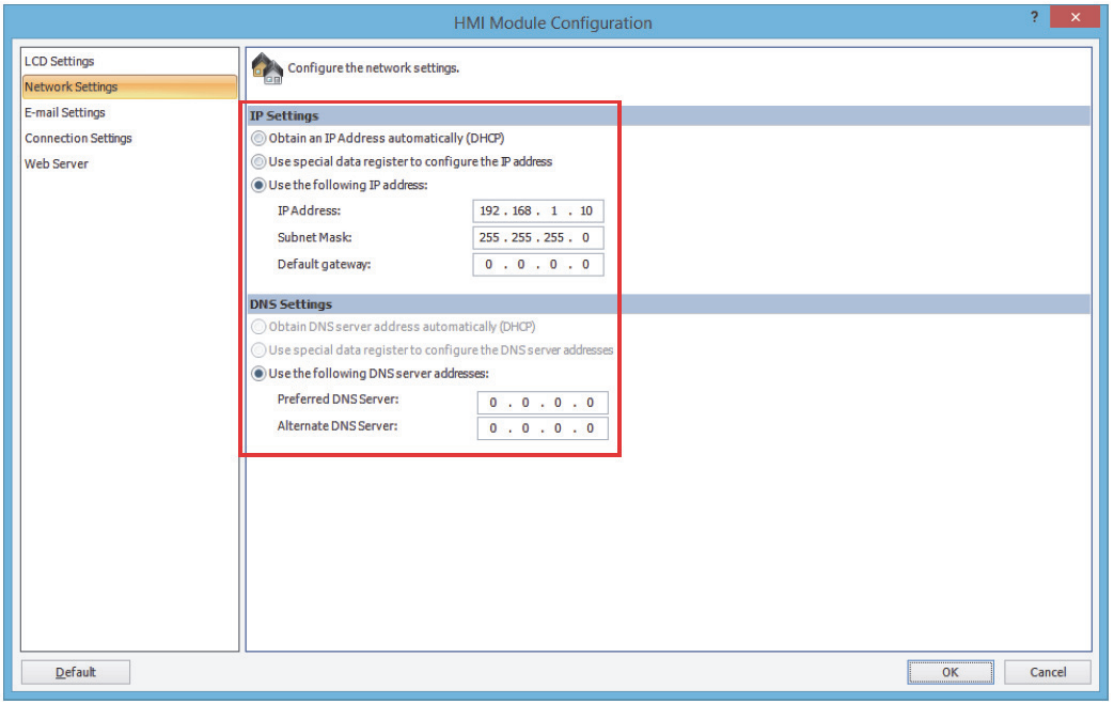
Configure the settings for maintenance communication.

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.

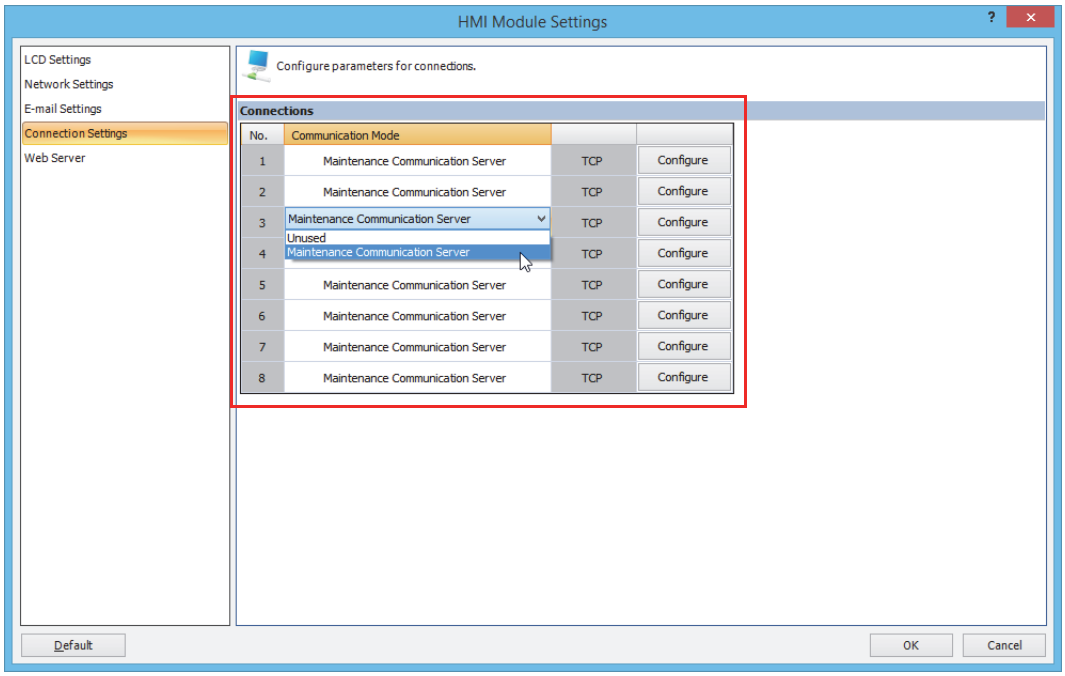
Note: You can also display the **HMI Module Configuration** dialog box by double-clicking **HMI Module** in the Project Window.



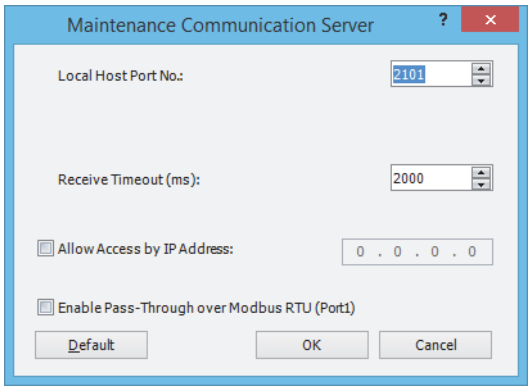
3. Click the **Network Settings** tab, and then configure the **IP Settings** and **DNS Settings**.



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.

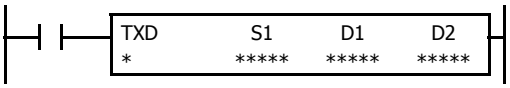
The user communication instructions differ according to the communication interface that will be used.

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

User communication allows the following two types of communication methods:

- Serial communication with an external device connected to port 1 to port 3 (RS232C/RS485)
- Ethernet communication with an external device connected by the Ethernet port 1

TXD (Transmit)



The transmit data is converted to the set data type and transmitted to the external device using port 1 to 3.

Valid Devices

Device	Function	I	Q	M	R	T	C	D	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 Chapter 2 "Device Addresses" in the "FC6A Series MicroSmart LAD Programming Manual".

*1 Internal relays M0 through M7997 or M10000 through M17497 can be designated as D1. 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.

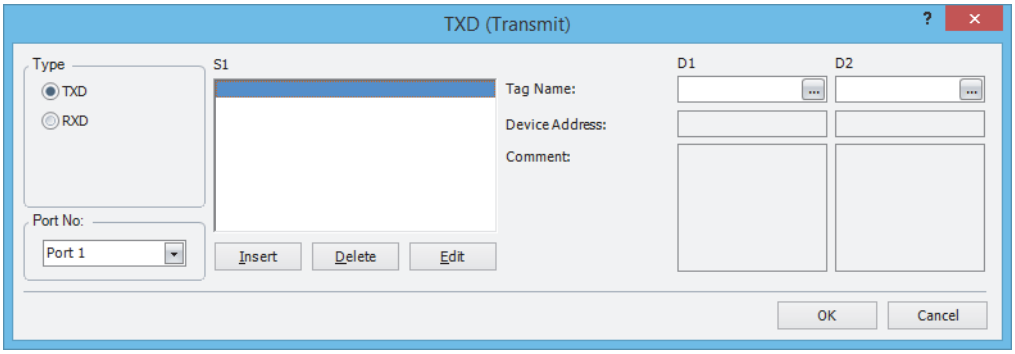
Destination 2 occupies two consecutive data registers starting with the device designated by D2. The transmit status data register, D0000-D7998 and D10000-D55998, stores the status of transmission and error code. The next data register stores the byte count of transmitted data. The same data registers cannot be used as transmit status registers for TXD1 through TXD3 instructions and receive status registers for RXD1 through RXD3 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 LAD Programming Manual".

Precautions for Programming TXD Instruction

- The FC6A Series MicroSmart has five formatting areas each for executing TXD1 through TXD3 instructions, so five instructions each of TXD1 through TXD3 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 LAD 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 3**	Transmit user communication from port 1 (TXD1) through port 3 (TXD3)
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 D0000-D7998 or D10000-D55998. 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 1,536 bytes of data.

S1 (Source 1)

Transmit Data	Device	Conversion Type	Transmit Digits (Bytes)	Repeat	BCC Calculation	Calculation Start Position
Constant (Character) **Constant (Hexadecimal)**	0-255	No conversion	1	—	—	—
Data Register	D0000-D7999 D10000-D55999	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 3 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.

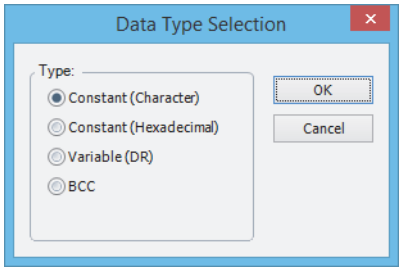
Constant (Hexadecimal)

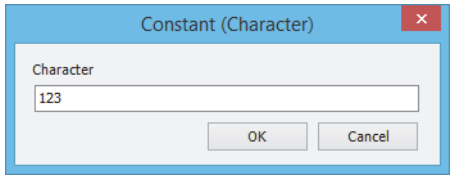
Use this option to enter the hexadecimal code of any ASCII character. ASCII control codes NUL (00h) through US (1Fh) can also be entered using this option.

Example:

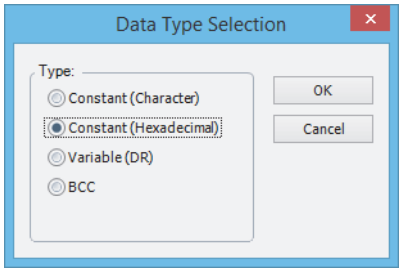
The following example shows two methods to enter 3-byte ASCII data "1" (31h), "2" (32h), "3" (33h).

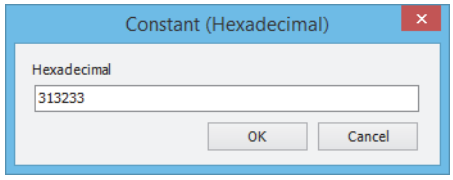
(1) Constant (Character)





(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.

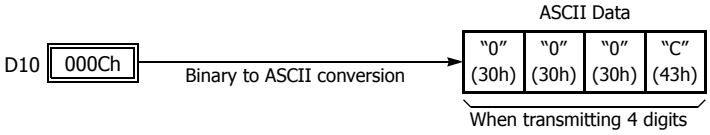
When repeat is designated, data of data registers as many as the repeat cycles are transmitted, starting with the designated data register. Repeat cycles can be up to 99.

Conversion Type

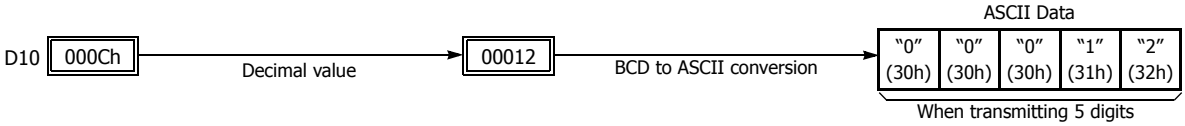
The transmit data is converted according to the designated conversion type as described below:

Example: D10 stores 000Ch (12)

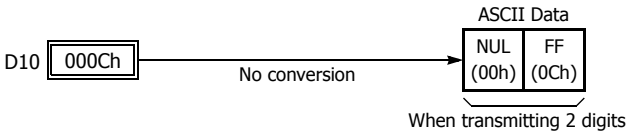
(1) Binary to ASCII conversion



(2) BCD to ASCII conversion



(3) No conversion

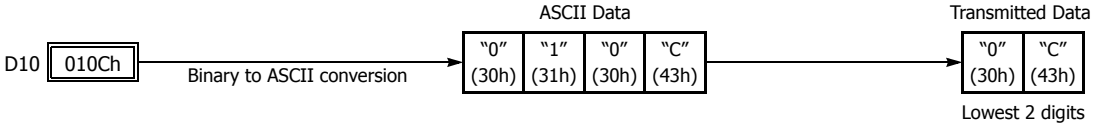


Transmit Digits (Bytes)

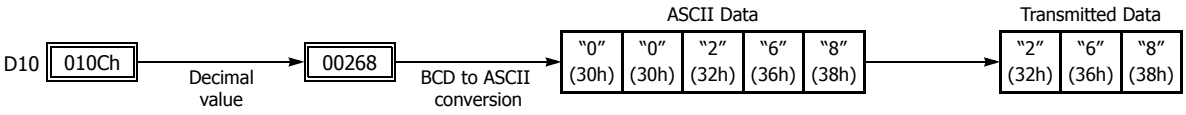
After conversion, the transmit data is taken out in specified digits. Possible digits depend on the selected conversion type.

Example: D10 stores 010Ch (268)

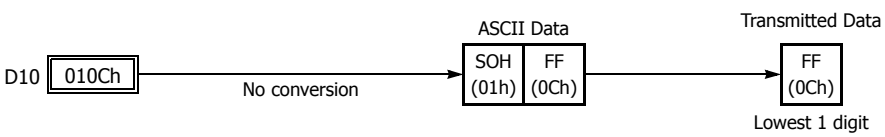
(1) Binary to ASCII conversion, Transmit digits = 2



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



(3) No conversion, Transmit digits = 1



Repeat Cycles

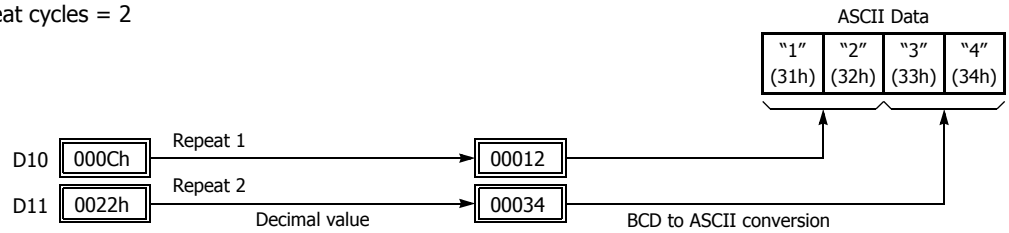
When a data register is assigned to repeat, as many consecutive data registers, as the repeat cycles, are used to transmit data in the same conversion type and transmit digits.

Example:

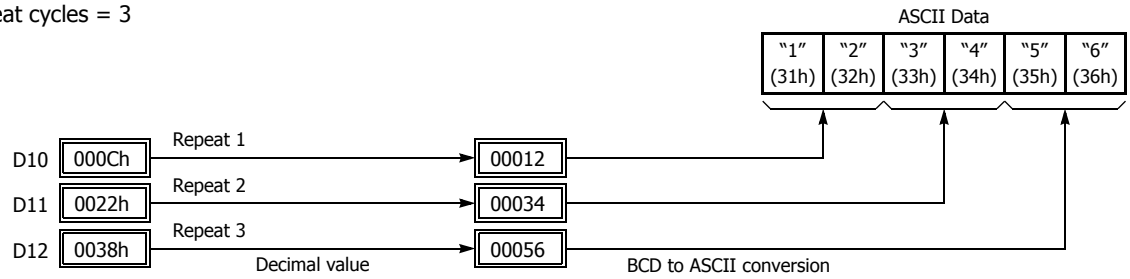
Data register No.: D10
Transmit digits: 2
Conversion type: BCD to ASCII

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

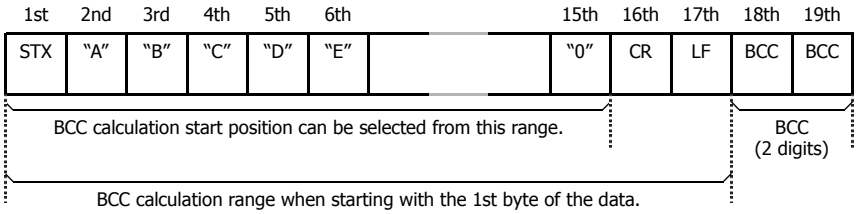


(2) Repeat cycles = 3



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.

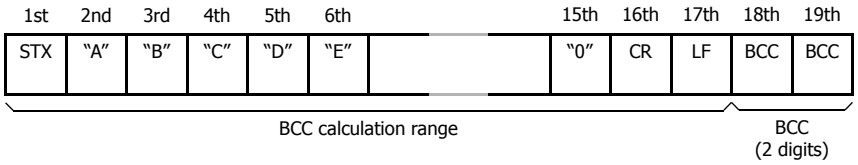


BCC Calculation Start Position

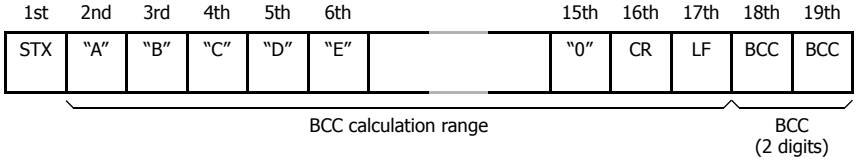
The start position for the BCC calculation can be specified from the first byte through the 15th byte. The BCC is calculated for the range starting at the designated position up to the byte immediately before the BCC of the transmit data.

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

(1) Calculation start position = 1



(2) Calculation start position = 2



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

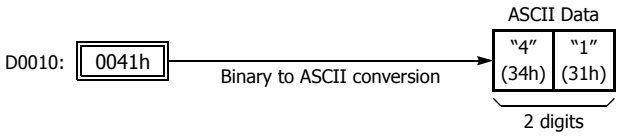
Calculation result = 85h 0Fh (binary)

Conversion Type

The BCC calculation result is converted or not according to the designated conversion type as described below:

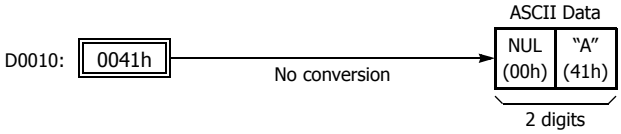
Example: BCC calculation result is 0041h.

(1) Binary to ASCII conversion



Note: On WindLDR, Modbus ASCII is defaulted to binary to ASCII conversion.

(2) No conversion

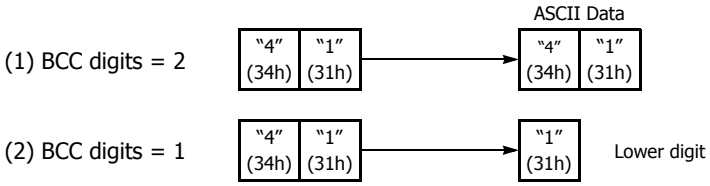


Note: On WindLDR, Modbus RTU is defaulted to no conversion.

BCC Digits (Bytes)

The quantity of digits (bytes) of the BCC code can be selected from 1 or 2.

Example:



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.

Data registers D0000 through D7998 and D10000 through D55998 can be set.

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

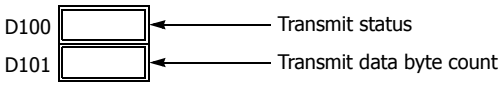
If the transmit status code is other than shown above, a transmit instruction error is suspected. See "User Communication Error" on page 5-43.

D2+1 (Transmit Digits (Bytes))

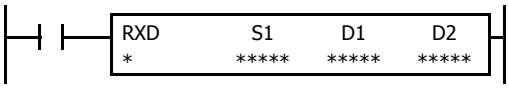
The transmit data byte count is stored in the data register after that set with D2 (D2+1). BCC data is also included in the transmit data byte count.

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 3 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	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 Chapter 2 "Device Addresses" in the "FC6A Series MicroSmart LAD Programming Manual".

*1 Internal relays M0 through M7997 or M10000 through M17497 can be designated as D1. 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.

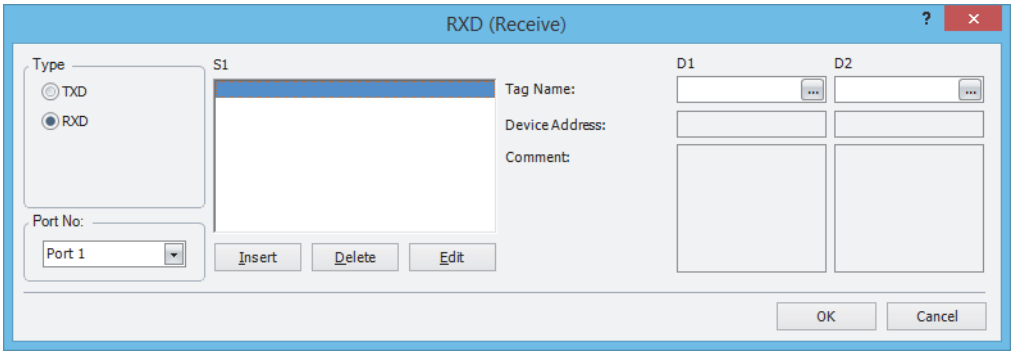
Destination 2 occupies two consecutive data registers starting with the device designated by D2. The receive status data register, D0000-D7998 or D10000-D55998, stores the status of data receive and error code. The next data register stores the byte count of received data. The same data registers can not be used as transmit status registers for TXD1 through TXD3 instructions and receive status registers for RXD1 through RXD3 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 LAD Programming Manual".

Precautions for Programming the RXD Instruction

- The FC6A Series MicroSmart can execute a maximum of five instructions each of RXD1 through RXD3 that have a start delimiter at the same time. If a start delimiter is not programmed in RXD1 through RXD3 instructions, the FC6A Series MicroSmart can execute only one instruction each of RXD1 through RXD3 at a time. If the start input for a RXD1 through RXD3 instruction is turned on while another RXD1 through RXD3 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 3**	Receive user communication to port 1 (RXD1) through port 3 (RXD3)
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 D0000-D7998 or D10000-D55998. 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 200 bytes of data at the maximum.

S1 (Source 1)

Receive Format	Device	Receive Digits (Bytes)	Conversion Type	Repeat	BCC Calculation	Calculation Start Position	Skip Bytes	Delimiter
Data Register	D0000-D7999 D10000-D55999	1-4 1-5 1-2	A:ASCII to Binary B:ASCII to BCD –:No conversion	1-99	—	—	—	Hex ASCII
Start Delimiter	00h-FFh 1 to 5 bytes	—	No conversion	—	—	—	—	
End Delimiter	00h-FFh	—	No conversion	—	—	—	—	
Constant for Verification	00h-FFh	—	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	

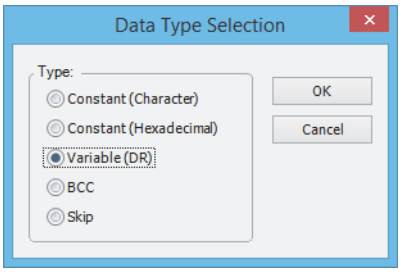
Note: Total byte count of the receive format is shown in S1 of RXD instruction on the ladder diagram of WindLDR.

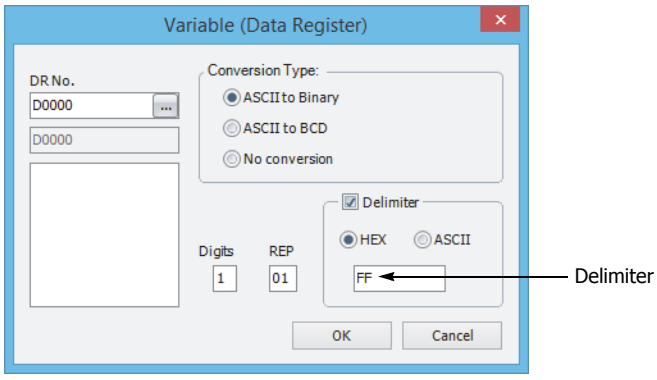
Designating Data Register as S1

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.



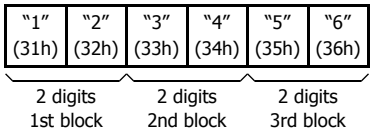


Receive Digits

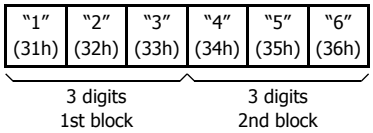
The received data is divided into blocks of specified receive digits before conversion as described below:

Example: Received data of 6 bytes are divided in different receive digits. (Repeat is also designated.)

(1) Receive digits = 2



(2) Receive digits = 3

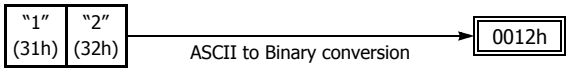


Conversion Type

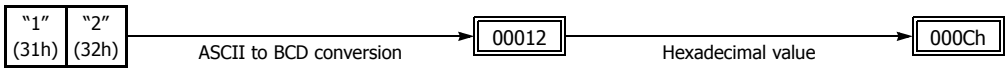
The data block of the specified receive digits is then converted according to the designated conversion type as described below:

Example: Received data has been divided into a 2-digit block.

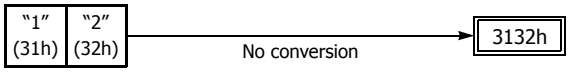
(1) ASCII to Binary conversion



(2) ASCII to BCD conversion



(3) No conversion

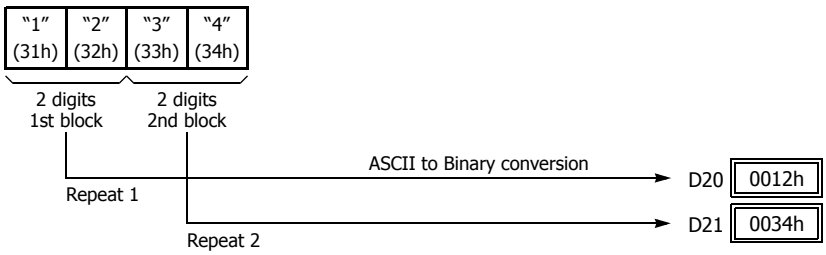


Repeat Cycles

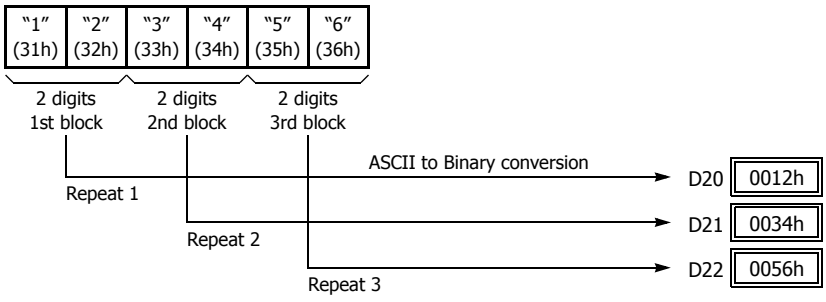
When a data register is assigned to repeat, the received data is divided and converted in the same way as specified, and the converted data is stored to as many consecutive data registers as repeat cycles.

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.

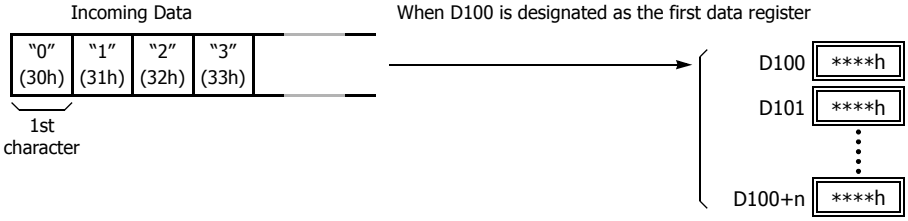
When a constant value is designated at the first byte of source device S1, the one-byte data serves as a start delimiter to start the processing of the received data.

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.

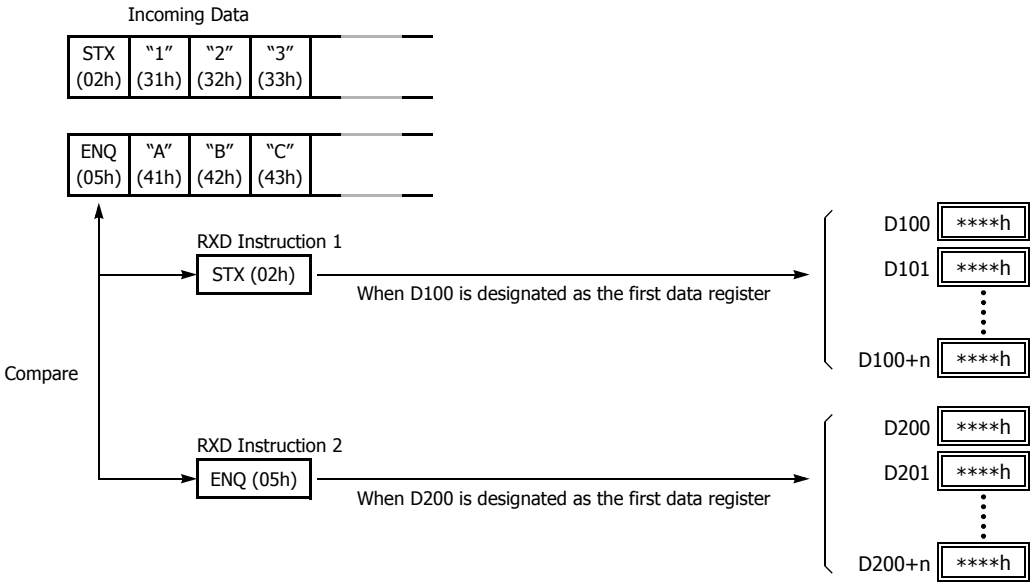
Example:

(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.

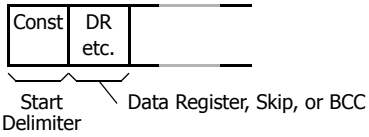
When the first one byte is received, a timer is started to monitor the interval between incoming data even when a multi-byte start delimiter is assigned. If data is not received in the period specified for the receive timeout value after receiving one byte of data, a receive timeout error occurs, and user communication error code 11 is stored in the status data register.

Examples: Multi-byte Start Delimiter

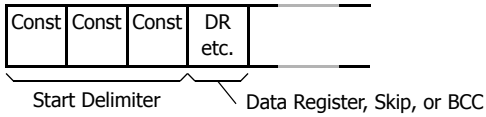
Multi-byte start delimiter is determined in the structure of the Receive Format. The following examples show how multi-byte start delimiter is determined.

- **Constants are followed by data register, skip, or BCC**

Receive Format



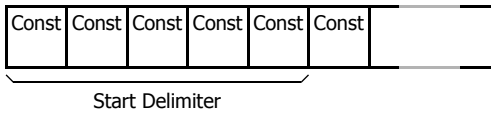
Receive Format



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**

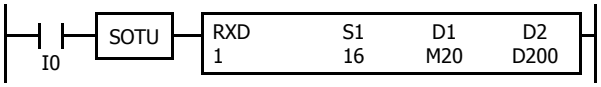
Receive Format



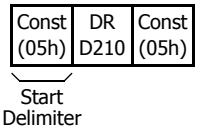
Note: Constants that are neither start delimiters or end delimiters are considered constants for verification. See "Constant for Verification" on page 5-17.

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.

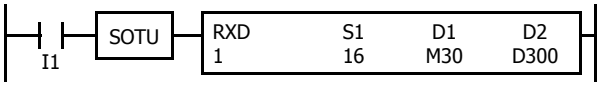


Receive Format S1

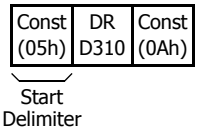


Communication port:	Port 1
Receive completion output:	M20
Receive status register:	D200
Receive data byte count:	D201

D200	32
M20	OFF



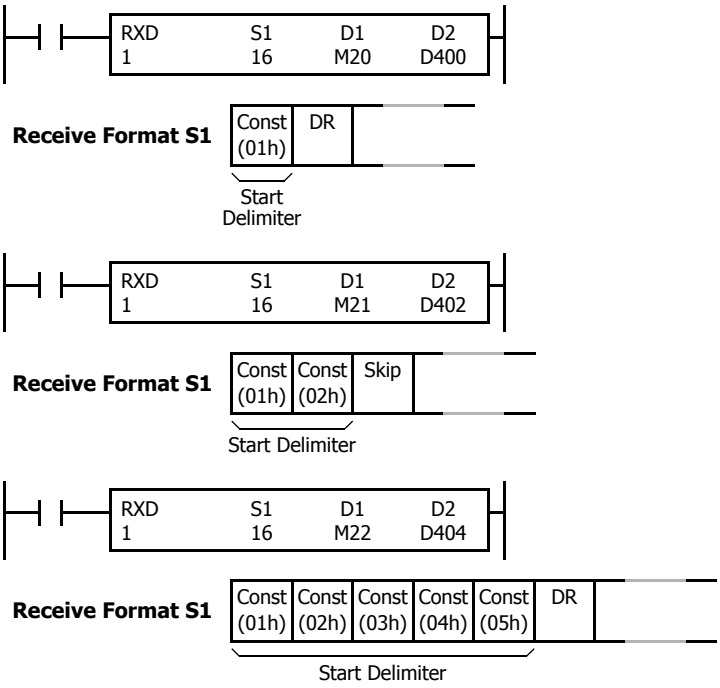
Receive Format S1



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.

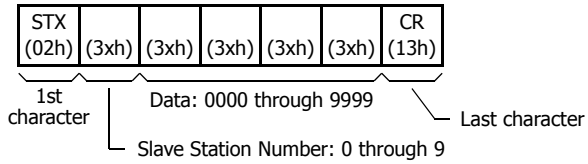


Example: Using Multi-byte Start Delimiter

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.

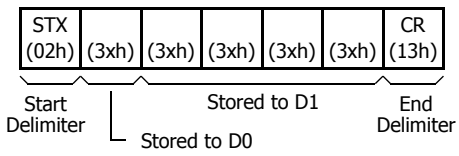
- **Incoming data**

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.



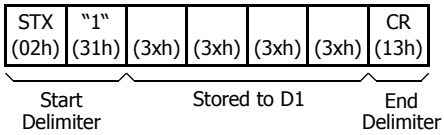
- **Single-byte start delimiter**

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.

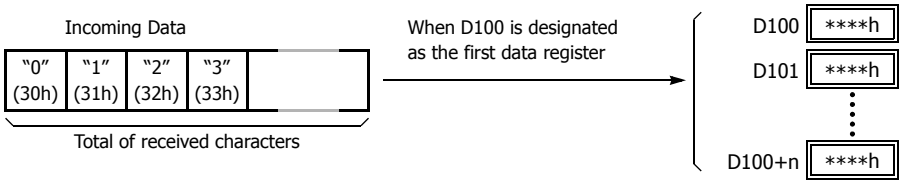
If a character in incoming data matches the end delimiter, the RXD instruction ends receiving data at this point and starts subsequent receive processing as specified. Even if a character matches the end delimiter at a position earlier than expected, the RXD instruction ends receiving data there.

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.

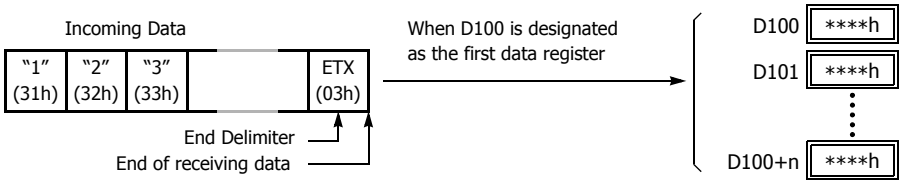
Example:

(1) When a RXD instruction without an end delimiter is executed



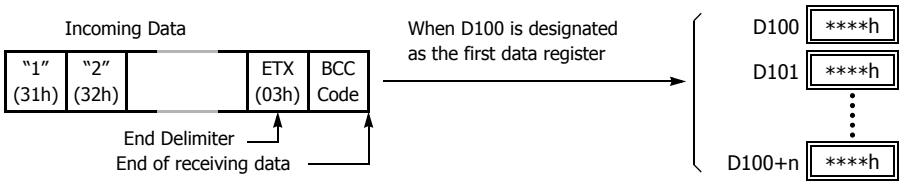
The incoming data is divided, converted, and stored to data registers according to the receive format.
Receive operation is completed when the total characters programmed in RXD are received.

(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

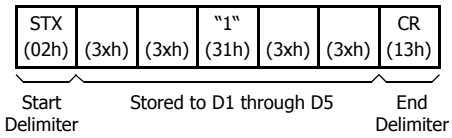
Constants excluding start and end delimiters can be configured in the receive format to verify the incoming data with the constants, which are either characters or hexadecimal values. As many constants can be configured for the verification as required. The verification result is stored in the receive 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.

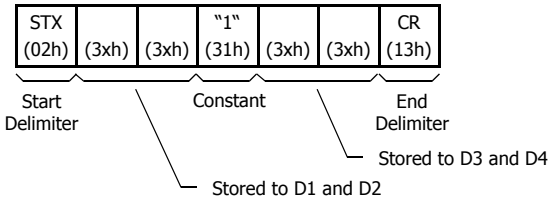
- **Using Data Register**

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: Receive completion output:	7 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: Receive completion output:	74 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

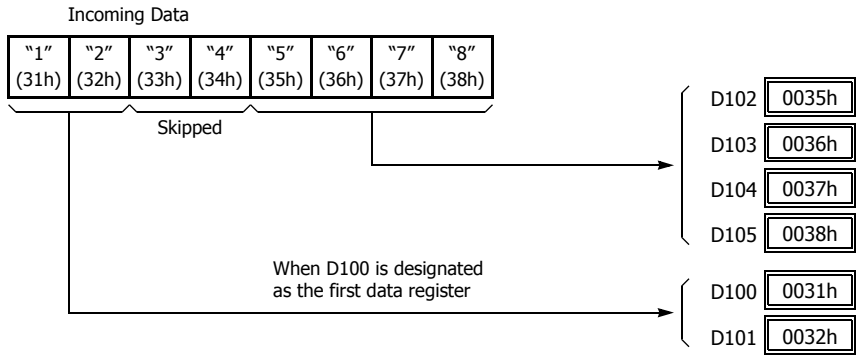
Data register, skip, or BCC

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.

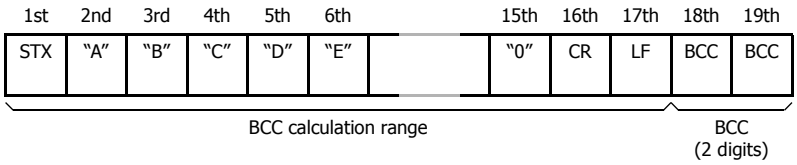
When an end delimiter is not used in the RXD instruction, the BCC code must be positioned at the end of the receive format designated in Source 1 device. When an end delimiter is used, the BCC code must be immediately before or after the end delimiter. The FC6A Series MicroSmart reads a specified number 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

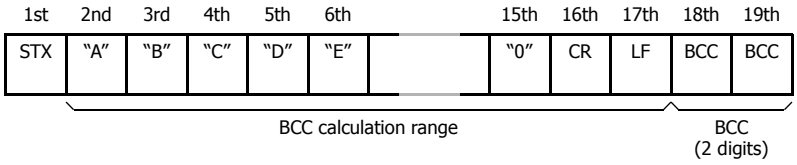
The start position for the BCC calculation can be specified from the first byte through the 15th byte. The BCC is calculated for the range starting at the designated position up to the byte immediately before the BCC of the receive data.

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

(1) Calculation start position = 1



(2) Calculation start position = 2



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

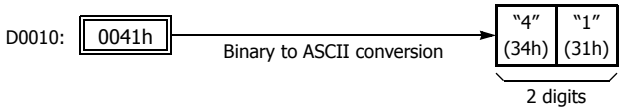
Calculation result = 85h 0Fh (binary)

Conversion Type

The BCC calculation result can be converted or not according to the assigned conversion type as described below:

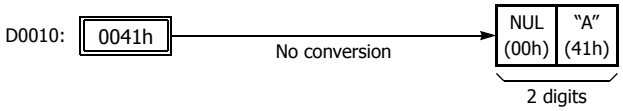
Example: BCC calculation result is 0041h.

(1) Binary to ASCII conversion



Note: On WindLDR, Modbus ASCII defaults to binary to ASCII conversion.

(2) No conversion

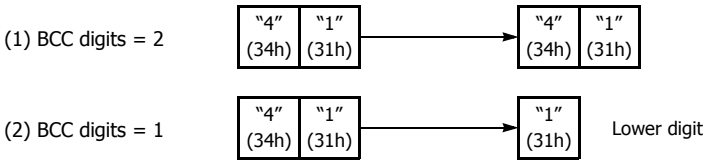


Note: On WindLDR, Modbus RTU defaults to no conversion.

BCC Digits (Bytes)

The quantity of digits (bytes) of the BCC code can be selected from 1 or 2.

Example:

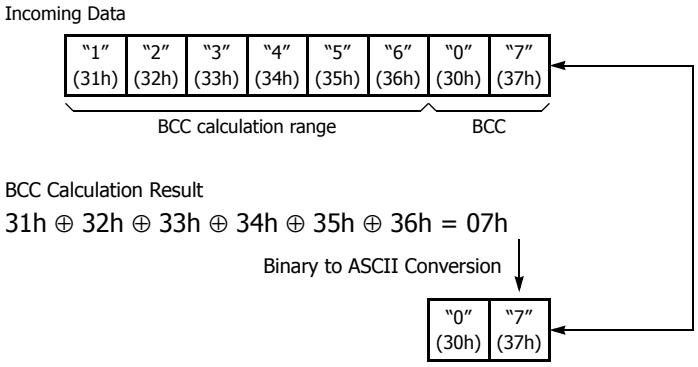


Note: On WindLDR, Modbus ASCII and Modbus RTU default is set to 2 digits.

Comparing BCC Codes

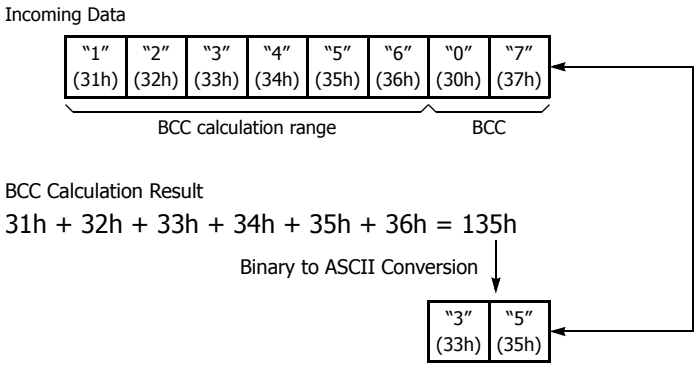
The FC6A Series MicroSmart compares the BCC calculation result with the BCC code in the received incoming data to check for any error in the incoming communication due to external noises or other causes. If a disparity is found in the comparison, an error code is stored in the data register designated as receive status in the RXD instruction. For user communication error code, see "User Communication Error" on page 5-43.

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)

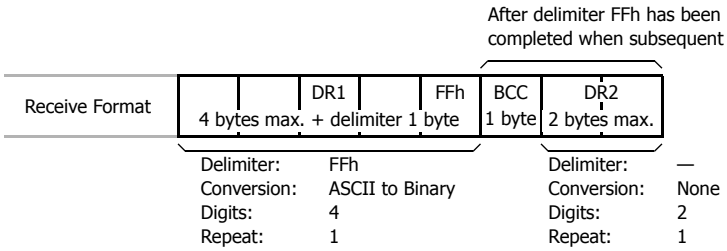
Set an internal relay or an output for the receive completion output.
When the start input for a RXD instruction is turned on, preparation for receiving data is initiated, followed by data conversion and storage. When the data receive operation sequence is complete, the designated output or internal 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.
Data registers D0000 through D7998 and D10000 through D55998 can be set.

D2+0 (Receive Status)

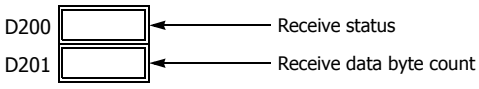
The receive status is stored in the data register set with D2. The receive status includes the reception operation status and the error information.

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 or M8026

If the receive status code is other than shown above, a receive instruction error is suspected. See "User Communication Error" on page 5-43.

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.



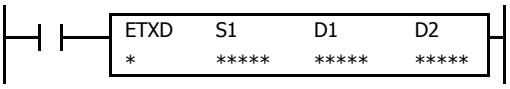
User Communication Receive Instruction Cancel Flag

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	W
M8023	User Communication Receive Instruction Cancel Flag (Port 2)	Cleared	Cleared	W
M8026	User Communication Receive Instruction Cancel Flag (Port 3)	Cleared	Cleared	W

"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.

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.

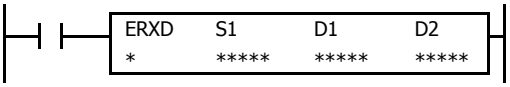
Apart from the connection settings, the settings of ETXD and TXD instructions are the same. For details on TXD instruction, see "TXD (Transmit)" on page 5-2.

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 LAD 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 5-35.

ERXD (User Communication Receive over Ethernet)



The ERXD instruction receives data from an external device connected over Ethernet, 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) 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 LAD 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 5-35.

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 3, such as a printer or barcode reader, by using the TXD (user communication transmit) and RXD (user communication receive) instructions.

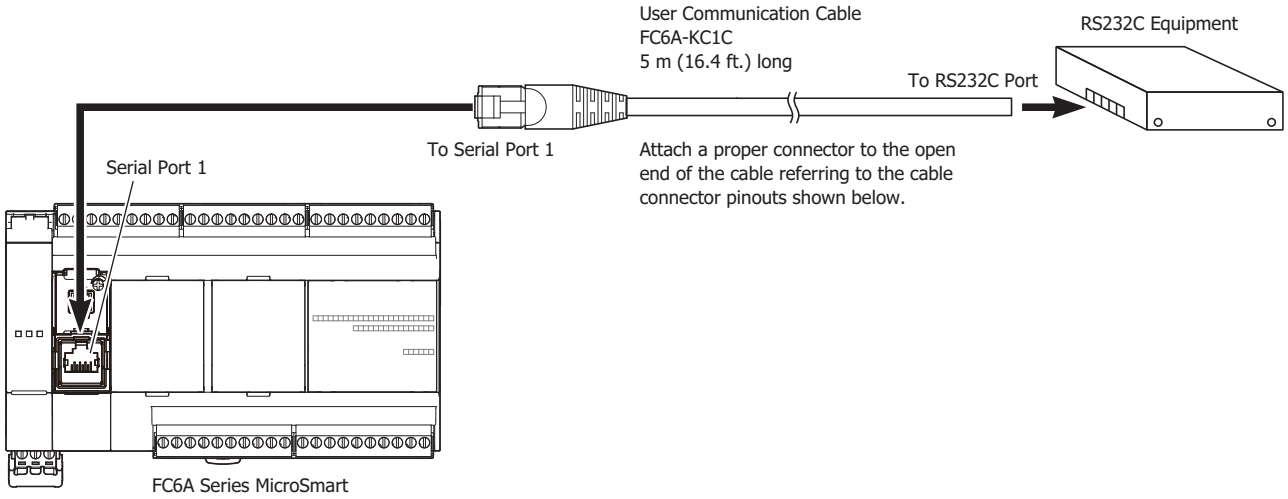
User Communication Overview

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.

A barcode reader is connected to port 1 of the FC6A Series MicroSmart.



User Communication Mode Specifications

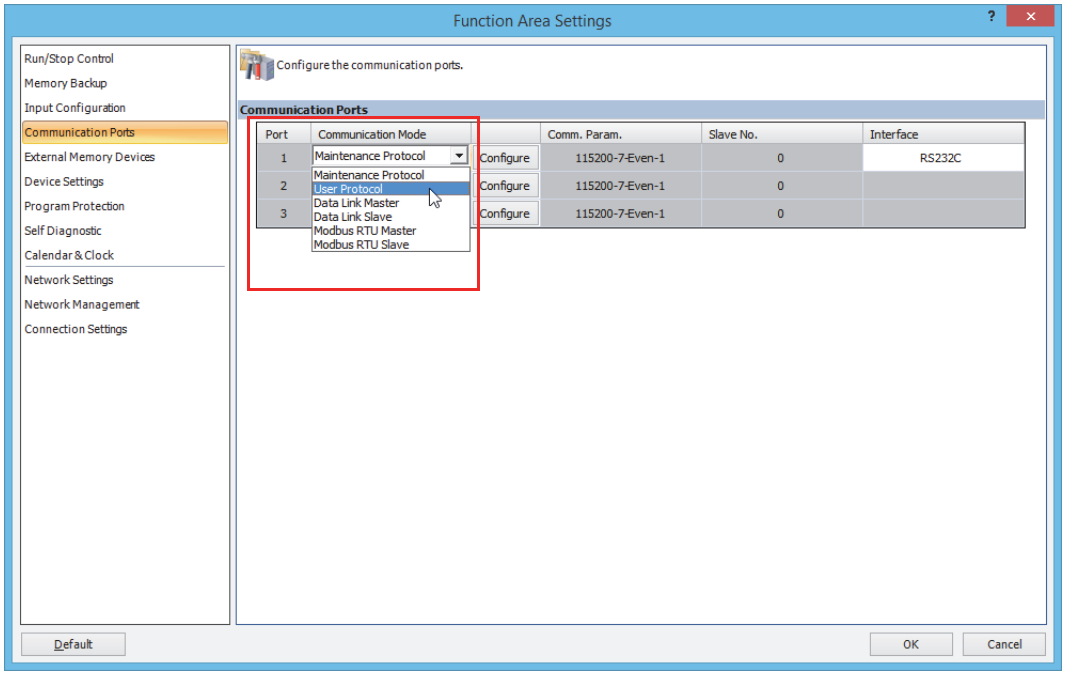
Type	RS232C User Communication	RS485 User Communication
Communication Port	Port 1 to port 3	Port 1 to port 3
Maximum Nodes	1 per port	31 maximum
Standards	EIA RS232C	EIA RS485
Baud Rate	1,200, 2,400, 4,800, 9,600, 19,200, 38,400, 57,600, 115,200 bps (Default: 115,200)	
Data Bits	7 or 8 bits (Default: 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	5 m	200 m
Maximum Transmit Data	1,536 bytes	
Maximum Receive Data	1,536 bytes	
BCC Calculation	XOR, ADD, ADD-2comp *, Modbus ASCII *, Modbus RTU * (* For calculation examples, see "BCC Calculation Examples" on page 5-49.)	

Programming WindLDR

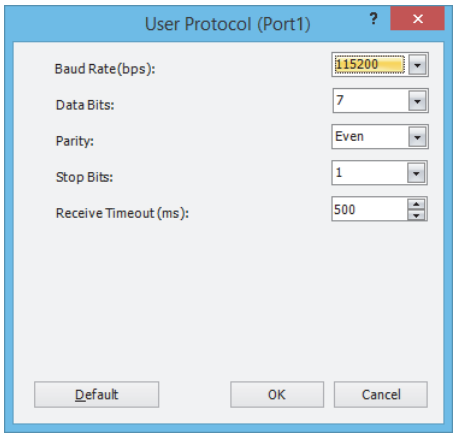
When using the user communication function to communicate with an external RS232C or RS485 device, set the communication parameters for the FC6A Series MicroSmart to match those of the external device.

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**, **Port 2**, and **Port 3**, select **User Protocol**. (Click the **Configure** button when changing previous settings.)
   The **User Protocol** dialog box appears.



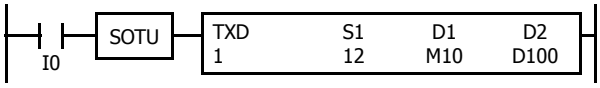
When **2550 ms** is selected in the **Receive Timeout** box, the receive timeout function is disabled.

3. Select communication parameters to the same values for the device to communicate with.
4. Click the **OK** button.

Programming TXD Instruction Using WindLDR

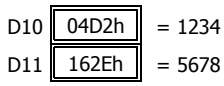
The following example demonstrates how to program a TXD instruction including a start delimiter, BCC, and end delimiter using WindLDR.

TXD sample program:

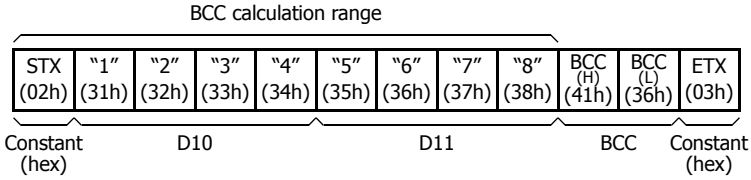


Communication port:	Port 1
Transmit completion output:	M10
Transmit status register:	D100
Transmit data byte count:	D101

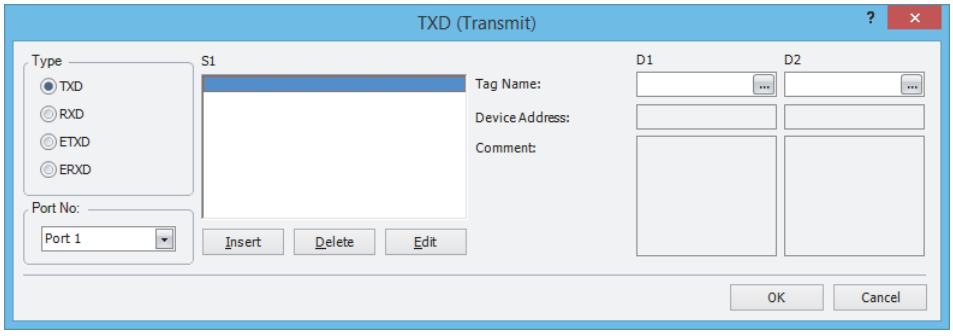
Data register contents:



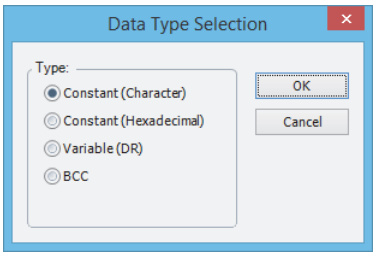
Transmit data example:

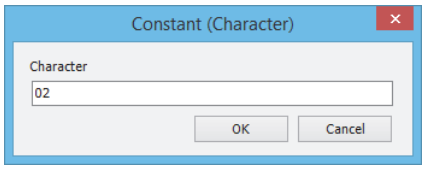


1. Start to program a TXD instruction. Move the cursor where you want to insert the TXD instruction, and type **TXD**. You can also insert the TXD instruction by clicking the User Communication icon in the menu bar and clicking where you want to insert the TXD instruction in the program edit area.
   The **TXD (Transmit)** dialog box appears.

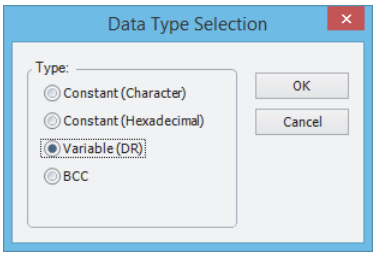


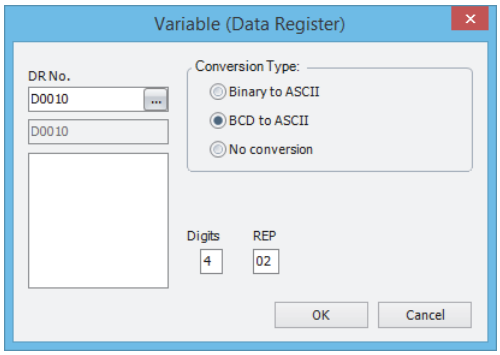
2. Check that **TXD** is selected in the **Type** box and select **Port 1** in the **Port No.** box. Then, click **Insert**.
   The **Data Type Selection** dialog box appears. You will program source device S1 using this dialog box.
3. Click **Constant (Hexadecimal)** in the **Type** box and click **OK**. Next, in the **Constant (Hexadecimal)** dialog box, type **02** to program the start delimiter STX (02h). When finished, click **OK**.



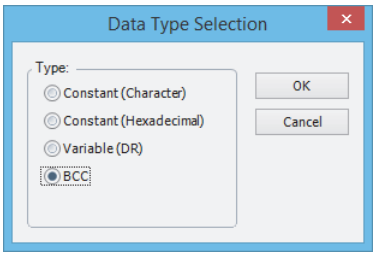


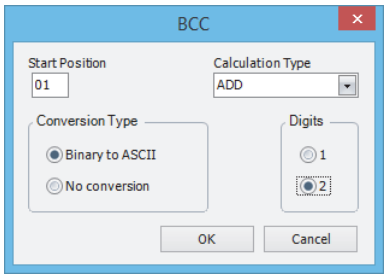
4. Since the **TXD (Transmit)** dialog box reappears, repeat the above procedure. In the **Data Type Selection** dialog box, click **Variable (DR)** and click **OK**. Next, in the **Variable (Data Register)** dialog box, type **D0010** in the **DR No.** box and click **BCD to ASCII** to select the BCD to ASCII conversion. Enter **4** in the **Digits** box (4 digits) and **2** in the **REP** box (2 repeat cycles). When finished, click **OK**.



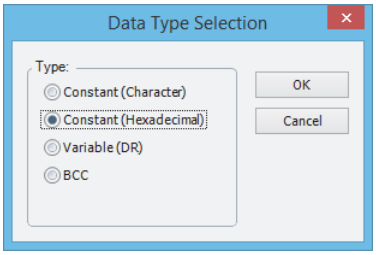


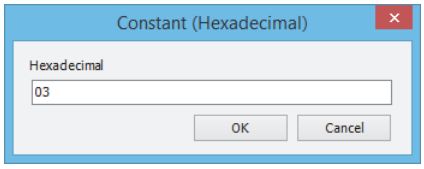
5. Again in the **Data Type Selection** dialog box, click **BCC** and click **OK**. Next, in the **BCC** dialog box, enter **1** in the **Calculation Start Position** box, select **ADD** for the **Calculate Type**, click **Binary to ASCII** for the **Conversion Type**, and click **2** for the **Digits**. When finished, click **OK**.



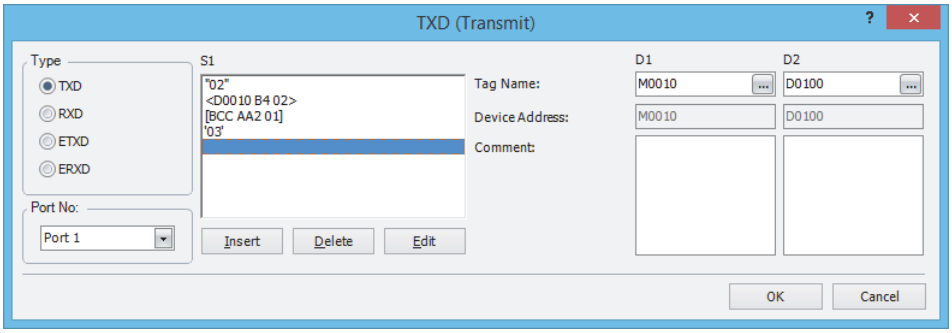


6. Once again in the **Data Type Selection** dialog box, click **Constant (Hexadecimal)** and click **OK**. Next, in the Constant (Hexadecimal) dialog box, type **03** to program the end delimiter ETX (03h). When finished, click **OK**.

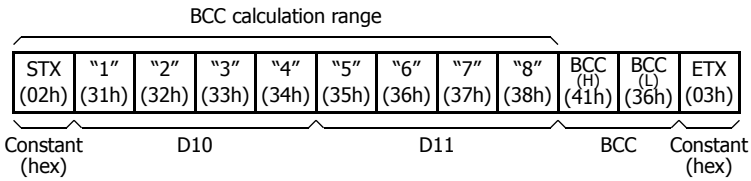




7. In the **TXD (Transmit)** dialog box, type **M0010** in the destination D1 box and type **D0100** in the destination D2 box. When finished, click **OK**.



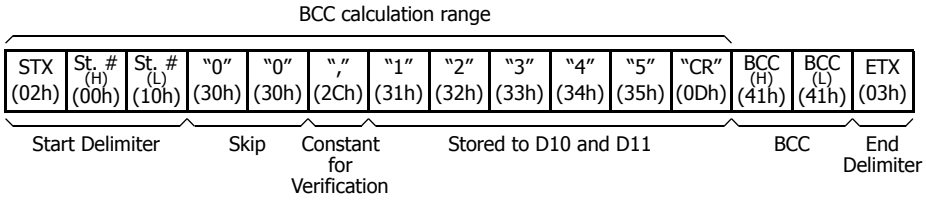
Programming of the TXD1 instruction is complete and the transmit data is specified as follows:



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.

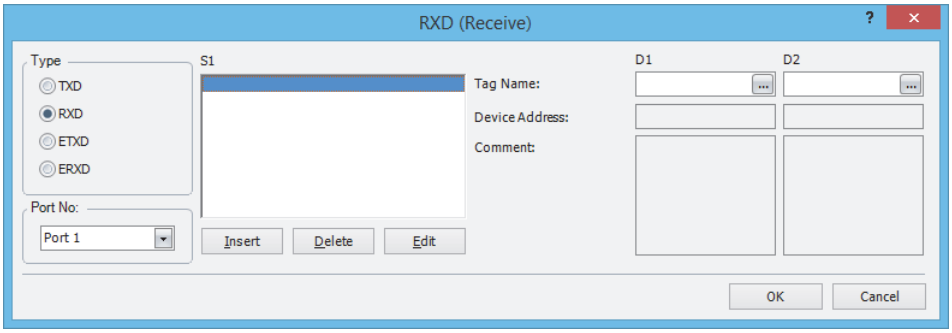
Receive data example:



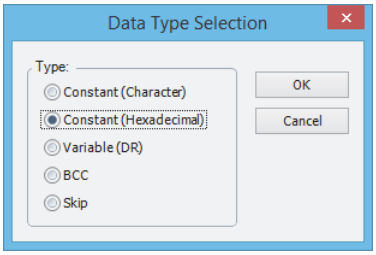
RXD sample program:

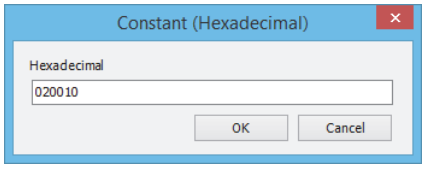
Communication port:	Port 1
Receive completion output:	M100
Receive status register:	D1000
Receive data byte count:	D1001

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.

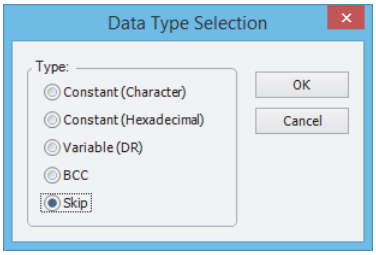


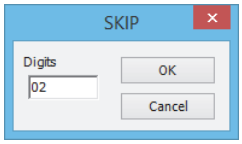
2. Check that **RXD** is selected in the Type box and select **Port 1** in the Port box. Then, click **Insert**.
   The **Data Type Selection** dialog box appears. You will program source device S1 using this dialog box.
3. Click **Constant (Hexadecimal)** in the **Type** box and click **OK**. Next, 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**.



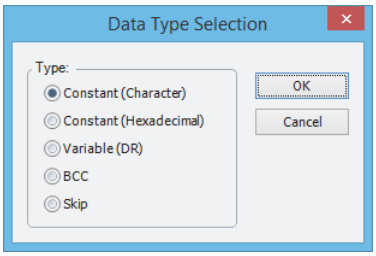


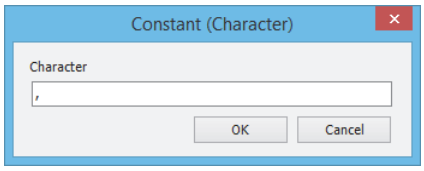
4. Since the **RXD (Receive)** dialog box reappears, repeat the above procedure. In the **Data Type Selection** dialog box, click **Skip** and click **OK**. Next, in the **Skip** dialog box, type **02** in the **Digits** box and click **OK**.



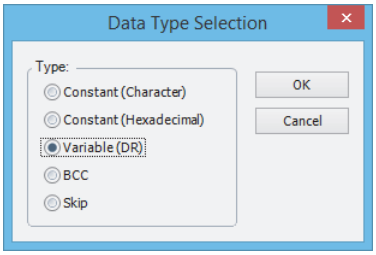


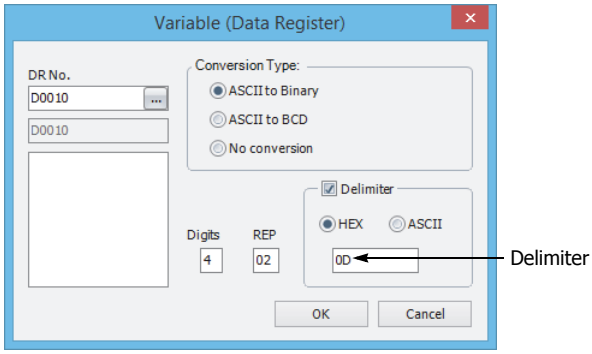
5. Again in the **Data Type Selection** dialog box, click **Constant (Character)** and click **OK**. Next, in the **Constant (Character)** dialog box, type **, (2Ch)** in the **Character** box to program a comma as a constant to verify. When finished, click **OK**.



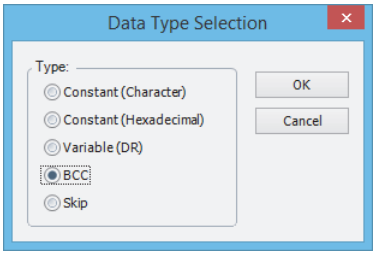


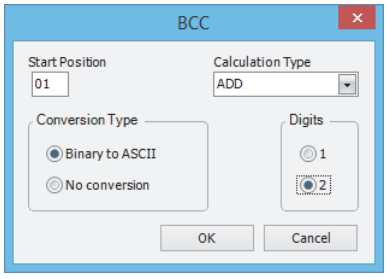
6. Again in the **Data Type Selection** dialog box, click **Variable (DR)** and click **OK**. Next, in the **Variable (Data Register)** dialog box, type **D10** in the **DR No.** box and click **ASCII to Binary** to select ASCII to binary conversion. Enter **4** in the **Digits** box (4 digits) and **2** in the **REP** box (2 repeat cycles). Click **Variable**, select **HEX**, and type **0D** to designate a delimiter. When finished, click **OK**.





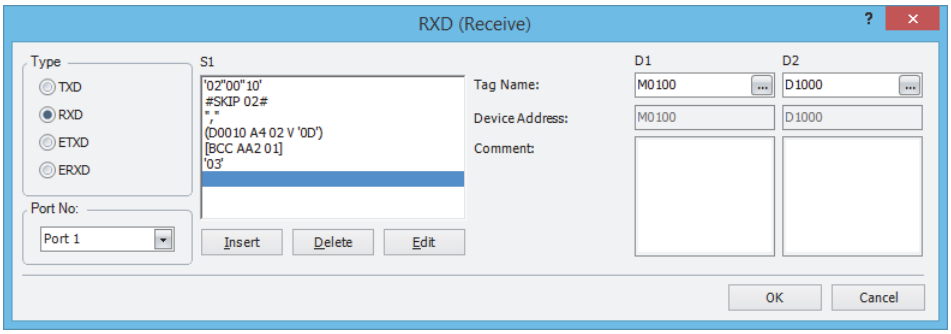
7. Again in the **Data Type Selection** dialog box, click **BCC** and click **OK**. Next, in the **BCC** dialog box, enter **01** in the **Calculation Start Position** box, select **ADD** for the **Calculation Type**, click **Binary to ASCII** for the **Conversion Type**, and click **2** for the **Digits**. 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 3.

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 Port 1 to Port 3 RS232C Line Control Signals

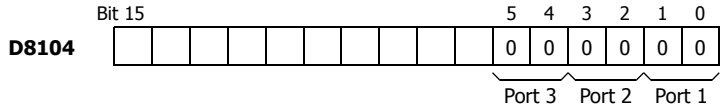
Special data registers D8104 through D8106 are allocated for RS232C line control signals.

Communication Port	DR No.	Data Register Function	Data Register Value Updated	R/W
Port 1 to Port 3	**D8104**	Control signal status	Every scan	R
	D8105	DSR input control signal option	When sending/receiving data	R/W
	D8106	DTR output control signal option	When sending/receiving data	R/W

Control Signal Status D8104

Special data register D8104 stores a value to show that DSR and DTR are on or off at port 1 through port 3.

The data of D8104 is updated at every END processing.



D8104 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	D8105 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

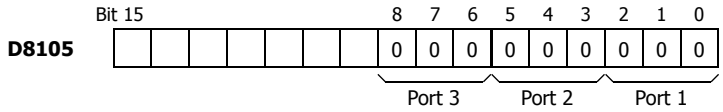
Communication Mode	D8106 2-bit Binary Value	DTR (Output) Status	
		RUN Mode	STOP Mode
User Communication Mode	**00 (default)**	ON	OFF
	01	OFF	OFF
	10	RXD enabled: ON RXD disabled: OFF	OFF
	11	ON	OFF
Maintenance Mode	—	ON	ON

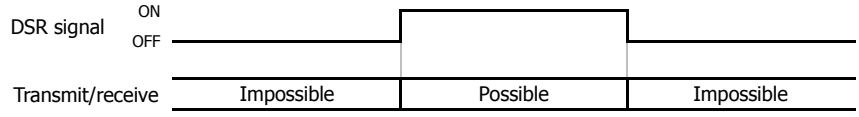
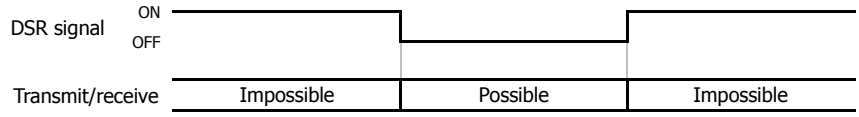
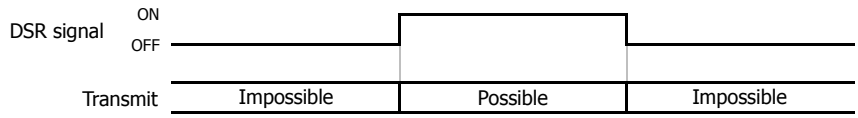
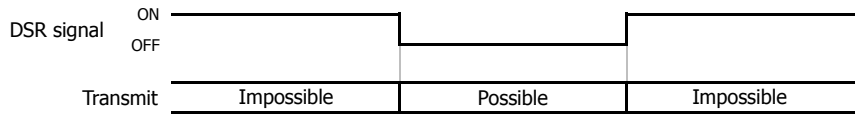
DSR Input Control Signal Option D8105

Special data register D8105 is used to control data flow between the FC6A Series MicroSmart RS232C port 1 through port 3 and the remote terminal depending on the DSR (data set ready) signal sent from the remote terminal. The DSR signal is an input to the FC6A Series MicroSmart to determine the status of the remote terminal. The remote terminal informs the FC6A Series MicroSmart using DSR whether the remote terminal is ready for receiving data or is sending valid data.

The DSR control signal option can be used only for the user communication through the RS232C port 1 to port 3.

The control status of each port is allocated as shown below:



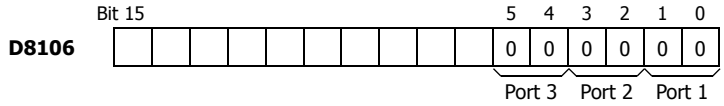
D8105 3-bit Binary Value	Description
000	DSR is not used for data flow control. When DSR control is not needed, set 0 to D8105.
001	When DSR is on, the FC6A Series MicroSmart can transmit and receive data. 
010	When DSR is off, the FC6A Series MicroSmart can transmit and receive data. 
011	When DSR is 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**	Same as D8105 = 000. DSR is not used for data flow control.

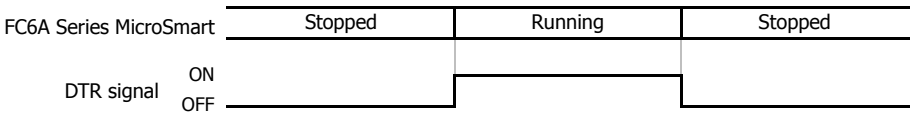
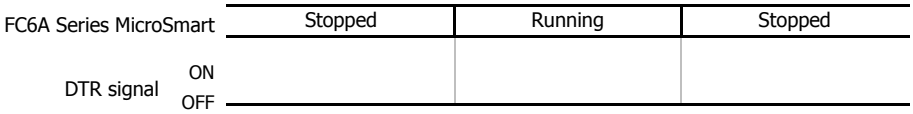
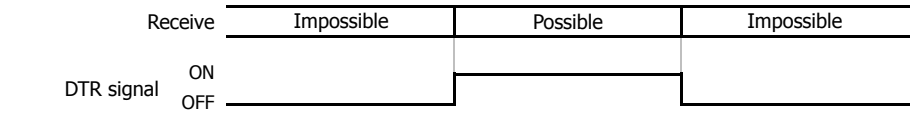
DTR Output Control Signal Option D8106

Special data register D8106 is used to control the DTR (data terminal ready) signal to indicate the FC6A Series MicroSmart operating status or transmitting/receiving status.

The DTR control signal option can be used only for the user communication through the RS232C port 1 to port 3.

The control status of each port is allocated as shown below:



D8106 2-bit Binary Value	Description
00	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. 
01	Whether the FC6A Series MicroSmart is running or stopped, DTR remains off. 
10	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. 
11	Same as D8106 = 00.

User Communication via Ethernet Communication

This section describes the Ethernet user communication. Ethernet user communication works on TCP/IP 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 client/server. 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 client connections of the FC6A Series MicroSmart can be allocated to user communication.

User communication client functions and configuration are described in "To use the FC6A Series MicroSmart as a user communication client" on page 5-36. User communication server functions and configuration are described in "User Communication Server" on page 5-40.

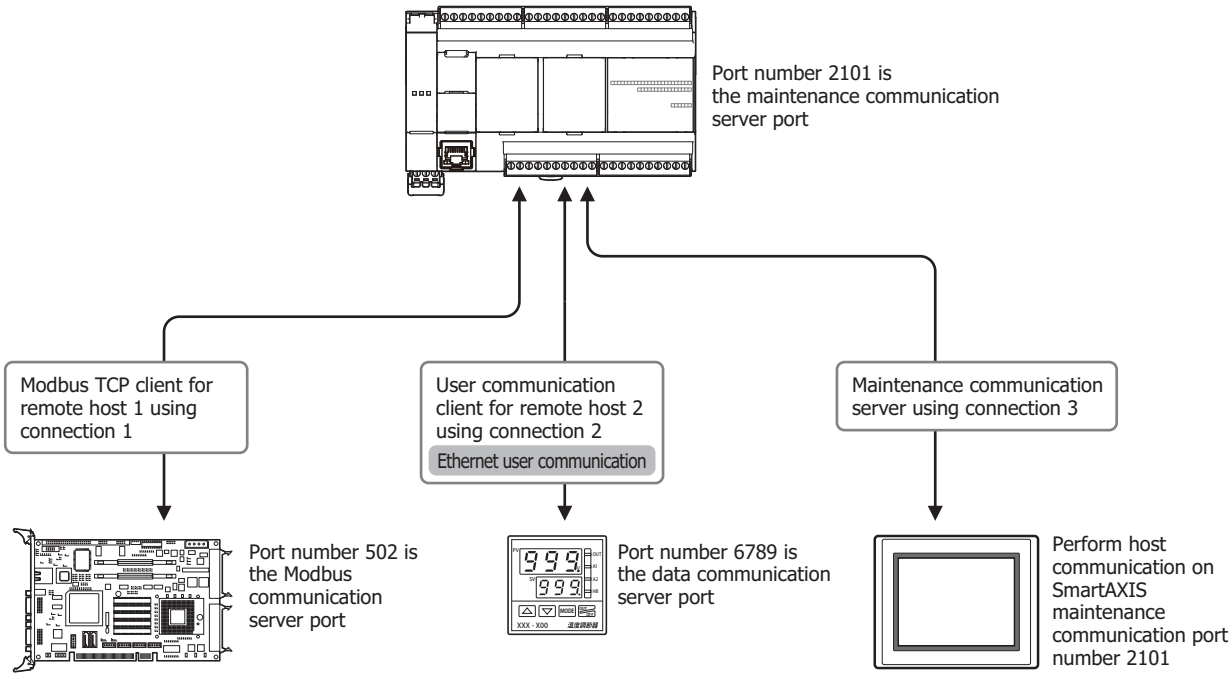
The FC6A Series MicroSmart supports the TCP/IP protocol.

The FC6A Series MicroSmart can send data to and receive data from devices on a network by using the ETXD (Ethernet user communication transmit) instruction and the ERXD (Ethernet user communication receive) 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

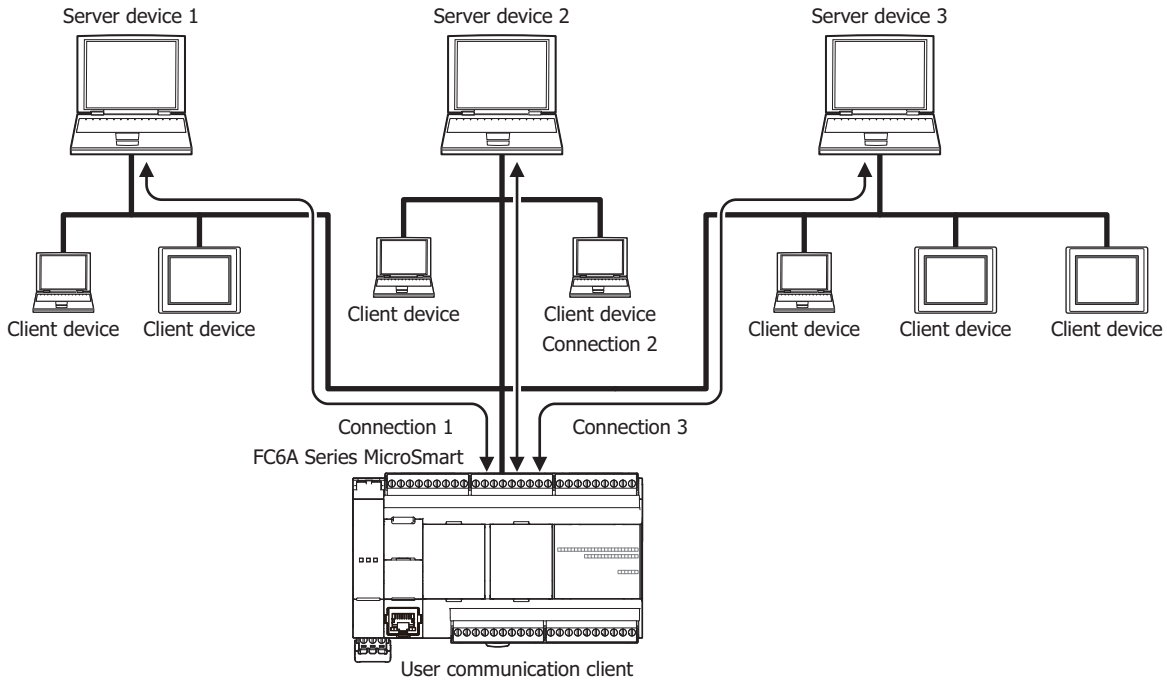
Remote Host Number	IP Address	Port Number
1	192.168.0.12	502
2	192.168.0.13	6789

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 can be allocated to user communication clients. The FC6A Series MicroSmart can simultaneously connect to and communicate with eight different server devices.

When three connections are allocated to user communication client

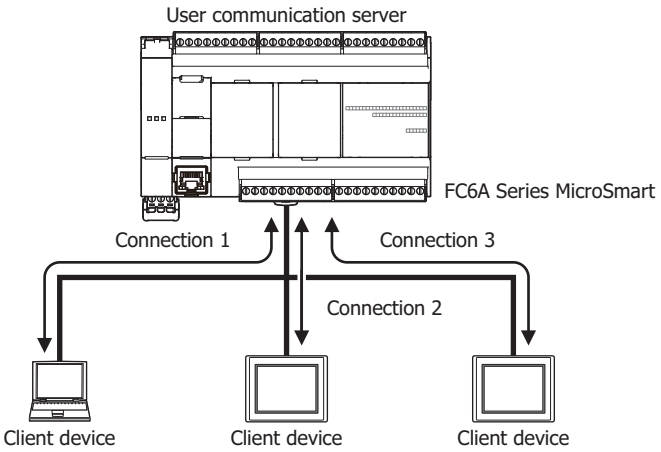


To use the FC6A Series MicroSmart as a user communication server

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 can be allocated to user communication servers. A maximum of eight client devices can simultaneously connect to and communicate with the FC6A Series MicroSmart.

When three connections are allocated to user communication server



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
Establish Connection	• When ETXD/ERXD Instructions are executed • When the FC6A Series MicroSmart starts to run (Note 1)
Disconnect Connection	• When the FC6A Series MicroSmart is stopped • When special internal relays (M8130 to M8132) are turned on
Number of Remote Hosts that the FC6A Series MicroSmart Can Communicate Simultaneously	One remote host per a user communication client (Note 2)
Receive Timeout Time	100 to 25,500 ms (100 ms increments); Default: 1,000 ms

Note 1: Can be enabled or disabled in **Function Area Settings**, **Connection Settings**.

Note 2: The settings can be configured in the Connection Settings tab in the Function Area Settings dialog box.

Establishing/Disconnecting User Communication Client Connections

When user communication clients are configured, connections are established on TCP/IP protocol. The connections are established when ETXD/ERXD instructions are executed or when the FC6A Series MicroSmart has started to run (See Note1 above). After a connection has been established, the connection will be kept open until either the FC6A Series MicroSmart is stopped or a special internal relay allocated to the connection is turned on.

Device Address	Description	Operation
M8222	Connection 1 Disconnect	When the reply 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	

Note 1: Can be enabled or disabled in **Function Area Settings**, **Connection Settings**.

User Communication Client Receive Instruction (ERXD) Cancel Flag

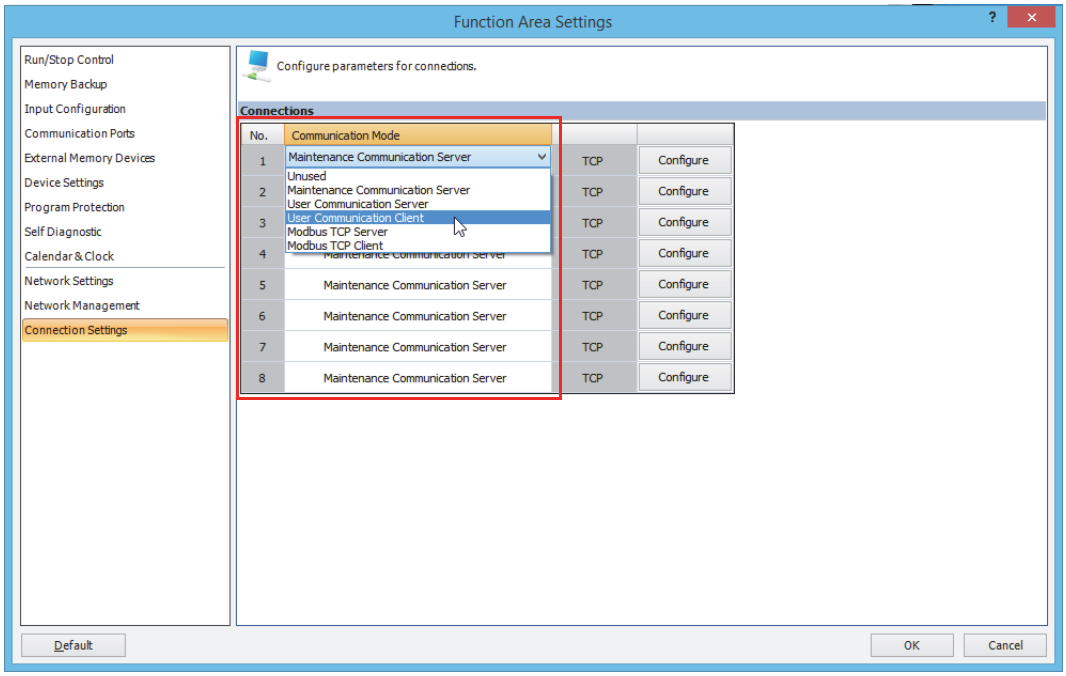
The allocation of the user communication receive instruction cancel flags for each client connection is shown in the table below. For details about the user communication receive instruction cancel flag, see "User Communication Receive Instruction Cancel Flag" on page 5-22.

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)

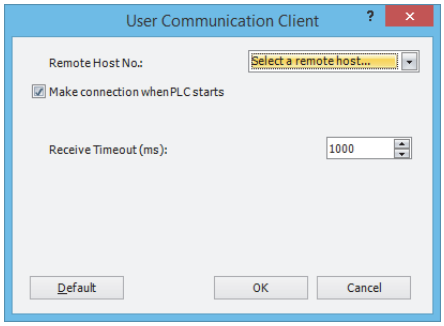
Programming WindLDR (User Communication Client)

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.

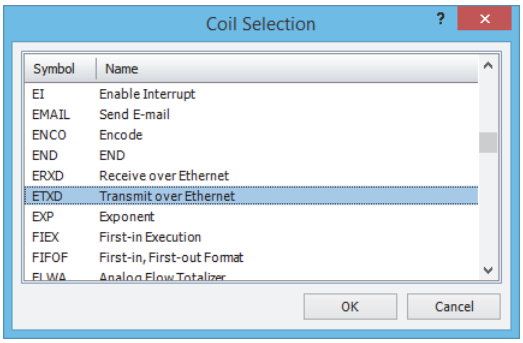


The **User Communication Client** dialog box appears.

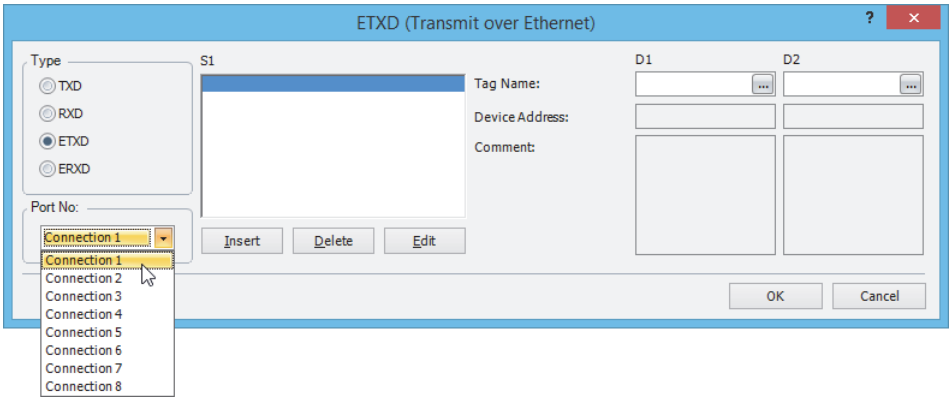


Configure the remote host number and receive timeout. If you want the FC6A Series MicroSmart to establish the connection when it starts to run, select "Make Connection when PLC starts." Click **OK** button to close the dialog box.

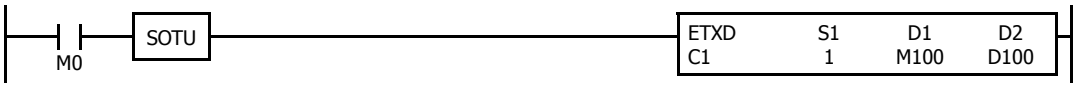
3. Edit the user program.
To insert Ethernet user communication instructions to the ladder editor, select the Ethernet user communication instructions (ETXD or ERXD instructions) in the **Coil Selection** dialog box.



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 2109 (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 (Note 1)
Receive Timeout Time	100 to 25,500 ms (100 ms increments)

Note 1: A maximum of eight connections can be allocated to user communication servers, and a maximum of eight client devices can simultaneously connect to and communicate with the FC6A Series MicroSmart.

User Communication Server Receive Instruction (ERXD) Cancel Flag

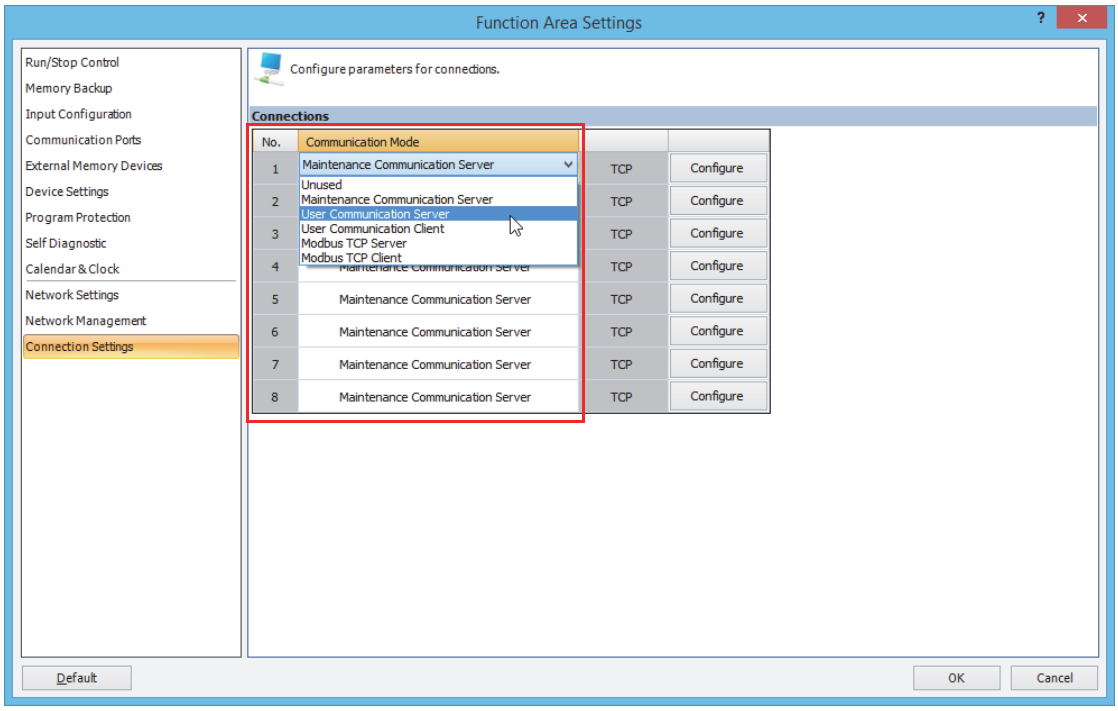
The allocation of the user communication receive instruction cancel flags for each server connection is shown in the table below. For details about the user communication receive instruction cancel flag, see "User Communication Receive Instruction Cancel Flag" on page 5-22.

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)

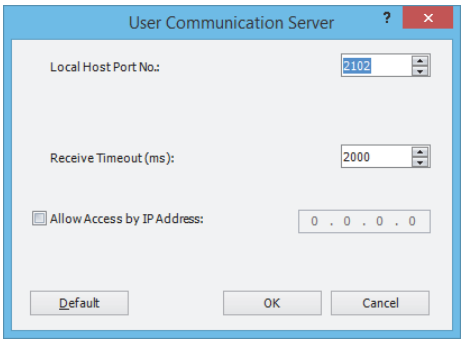
Programming WindLDR (User Communication Server)

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.

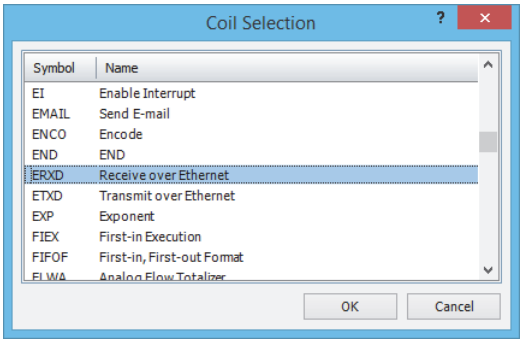


The **User Communication Server** dialog box appears.

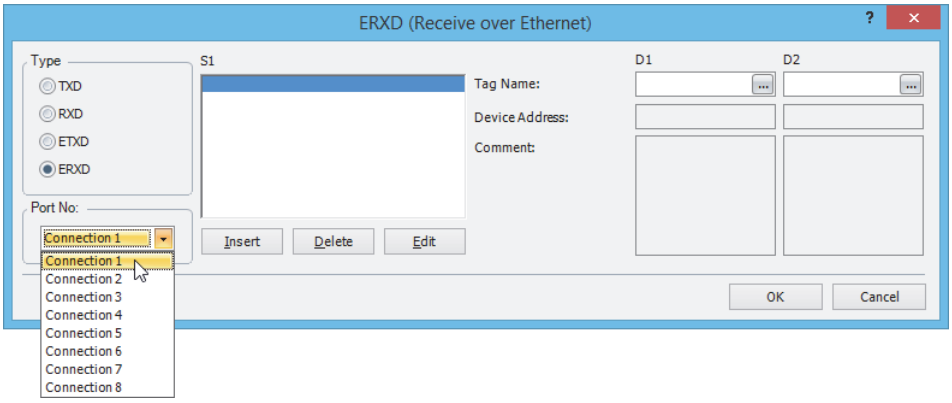


Configure the local host port number and receive timeout. If you want to restrict the access using IP address, configure the allowed IP address.

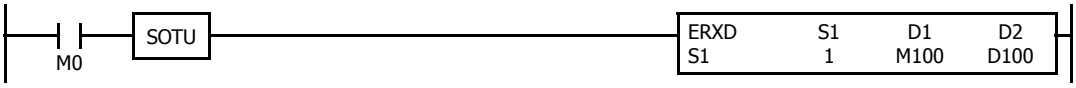
3. Edit the user program.
To insert Ethernet user communication instructions to the ladder editor, select the Ethernet user communication instructions (ETXD or ERXD instructions) in the **Coil Selection** dialog box.



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 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 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 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 and "D2 (Destination 2)" on page 5-21.

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.

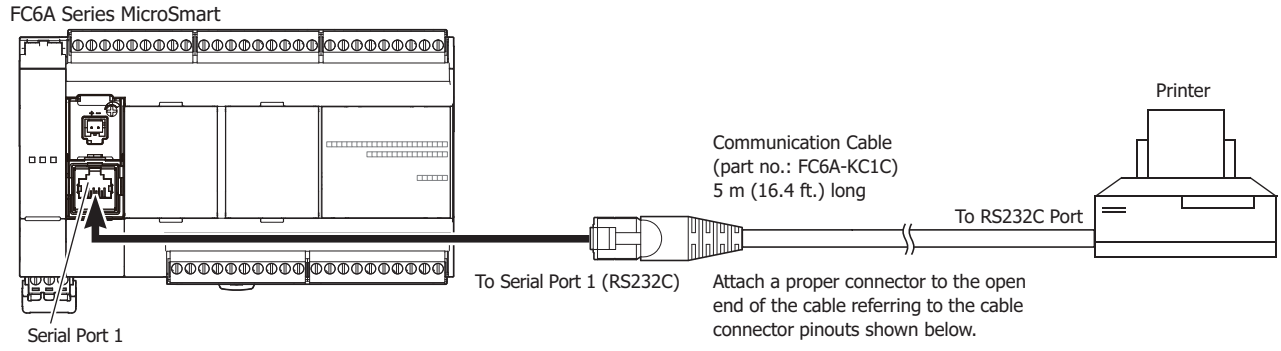
ASCII Character Code Table

Upper Bit / Lower Bit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	$^{N}U_{L}$	$^{D}L_{E}$	SP	0	@	P	`	p								
Decimal	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1	$^{S}O_{H}$	$^{D}C_{1}$	!	1	A	Q	a	q								
Decimal	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
2	$^{S}T_{X}$	$^{D}C_{2}$	″	2	B	R	b	r								
Decimal	2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
3	$^{E}T_{X}$	$^{D}C_{3}$	#	3	C	S	c	s								
Decimal	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
4	$^{E}O_{T}$	$^{D}C_{4}$	$	4	D	T	d	t								
Decimal	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
5	$^{E}N_{Q}$	$^{N}A_{K}$	%	5	E	U	e	u								
Decimal	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
6	$^{A}C_{K}$	$^{S}Y_{N}$	&	6	F	V	f	v								
Decimal	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
7	$^{B}E_{L}$	$^{E}T_{B}$	′	7	G	W	g	w								
Decimal	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
8	BS	$^{C}A_{N}$	(	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	$^{S}U_{B}$	*	:	J	Z	j	z								
Decimal	10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
B	VT	$^{E}S_{C}$	+	;	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

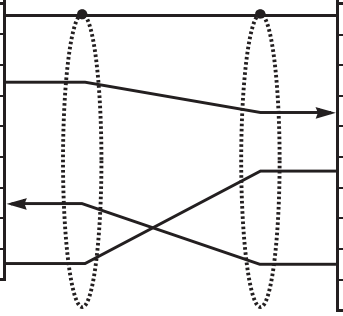
This example demonstrates a program to send data to a printer using the user communication TXD1 (transmit) instruction.

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.

Printout Example

```
--- PRINT TEST ---

11H 00M

CNT2...0050
D030...3854

--- PRINT TEST ---

11H 01M

CNT2...0110
D030...2124
```

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 5-43.

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 D8105" on page 5-33.

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:

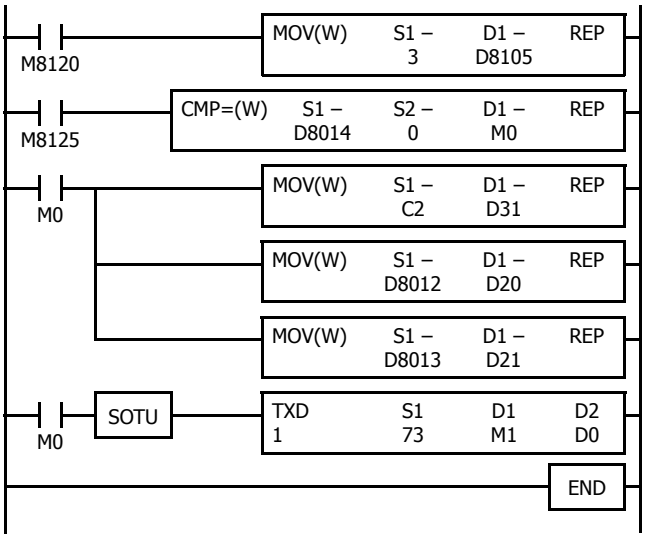
Communication Parameters:

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.

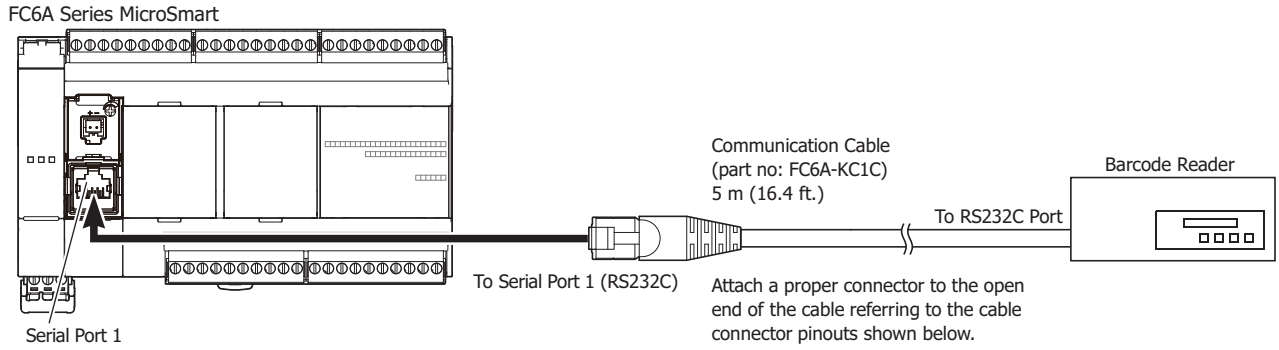
Details of S1 Settings in the Transmit Instruction

SP SP SP – – – SP P R I N T SP T 20h 20h 20h 2Dh 2Dh 2Dh 20h 50h 52h 49h 4Eh 54h 20h 54h	
E S T SP – – – CR LF CR LF SP SP SP 45h 53h 54h 20h 2Dh 2Dh 2Dh 0Dh 0Ah 0Dh 0Ah 20h 20h 20h	D20 hour data is converted from BCD to ASCII, and 2 digits are sent.
D20 Conversion: BCD→ASCII Digits: 2 REP: 01	
H SP 48h 20h	D21 minute data is converted from BCD to ASCII, and 2 digits are sent
D21 Conversion: BCD→ASCII Digits: 2 REP: 01	
M CR LF CR LF 4Dh 0Dh 0Ah 0Dh 0Ah	D31 counter C2 data is converted from BCD to ASCII, and 4 digits are sent.
SP SP SP C N T 2 . . . 20h 20h 20h 43h 4Eh 54h 32h 2Eh 2Eh 2Eh	
D31 Conversion: BCD→ASCII Digits: 4 REP: 01	D30 data is converted from BCD to ASCII, and 4 digits are sent.
CR LF SP SP SP D 0 3 0 . . . 0Dh 0Ah 20h 20h 20h 44h 30h 33h 30h 2Eh 2Eh 2Eh	
D30 Conversion: BCD→ASCII Digits: 4 REP: 01	
CR LF CR LF 0Dh 0Ah 0Dh 0Ah	

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.

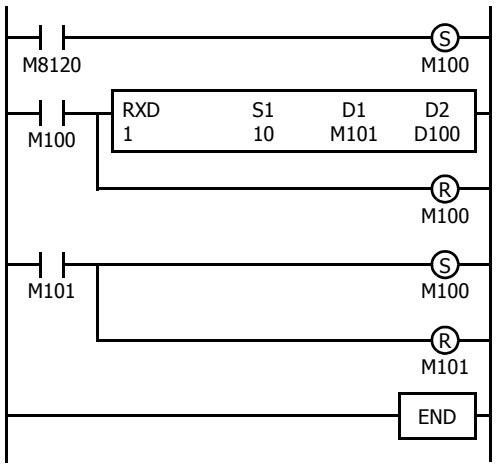
Synchronization Mode	Auto			
Read Mode	Single read or multiple read			
Communication Parameter	Baud rate:	9,600 bps	Data bits:	7
	Parity check:	Even	Stop bit:	1
Other Communication Settings	Header:	02h	Terminator:	03h
	Data echo back:	No	BCR data output:	Yes
	Output timing:	Output priority 1	Character suppress:	No
	Data output filter:	No	Main serial input:	No
	Sub serial:	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.



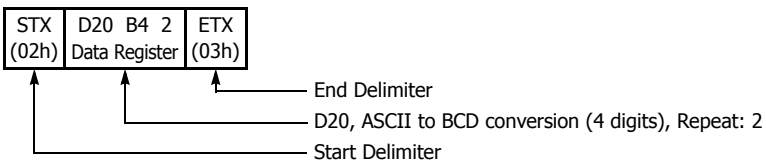
M8120 is the initialize pulse special internal relay used to set M100.

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.

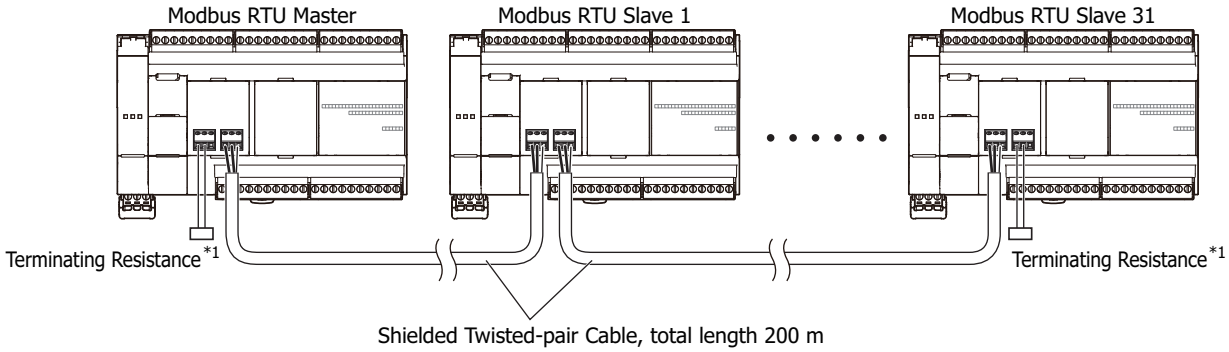
Modbus communication allows the following two types of communication methods.

- Serial communication with an external device connected to port 1 to port 3
  "Modbus RTU Communication via RS232C/RS485" on page 6-1
- Ethernet communication with an external device connected to Ethernet port 1
  "Modbus TCP Communication via Ethernet Communication" on page 6-18

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 6-2. For the Modbus RTU slave function and how to configure it, see "Modbus RTU Slave Communication" on page 6-8.



*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.

Note: The 16- and 24-I/O types have one cartridge slot and the 40-I/O type has two.

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	RS-232C: 5 m RS-485: 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	1 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:

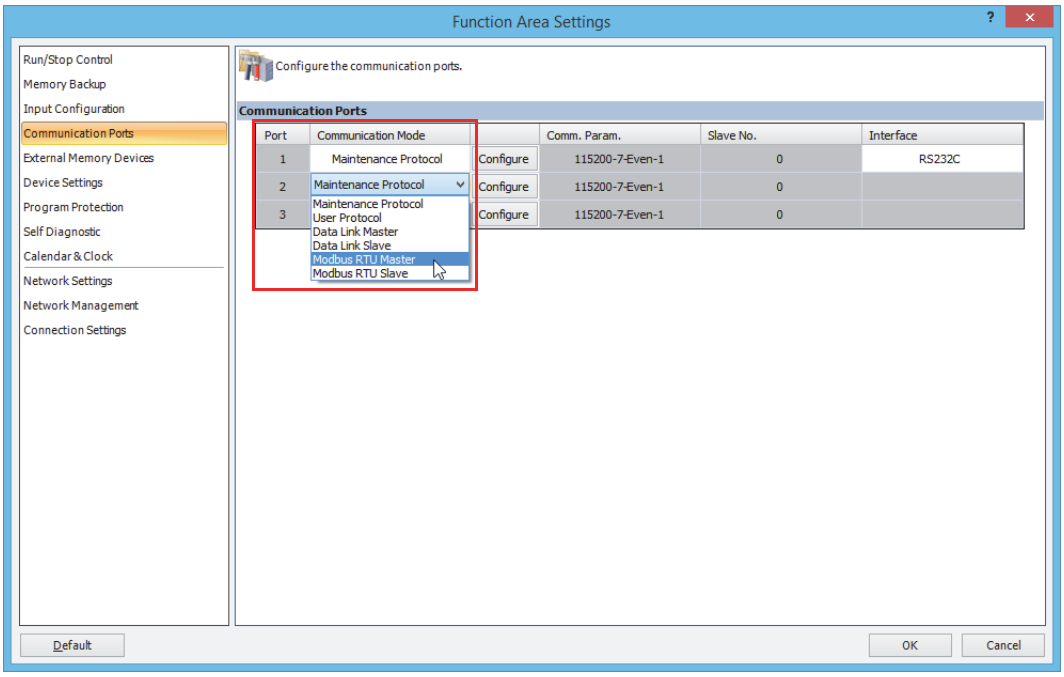
Port	Port 1, Port 2 and Port 3
No. of Requests	255

Note: 8 bytes of the user program area are needed per each request.

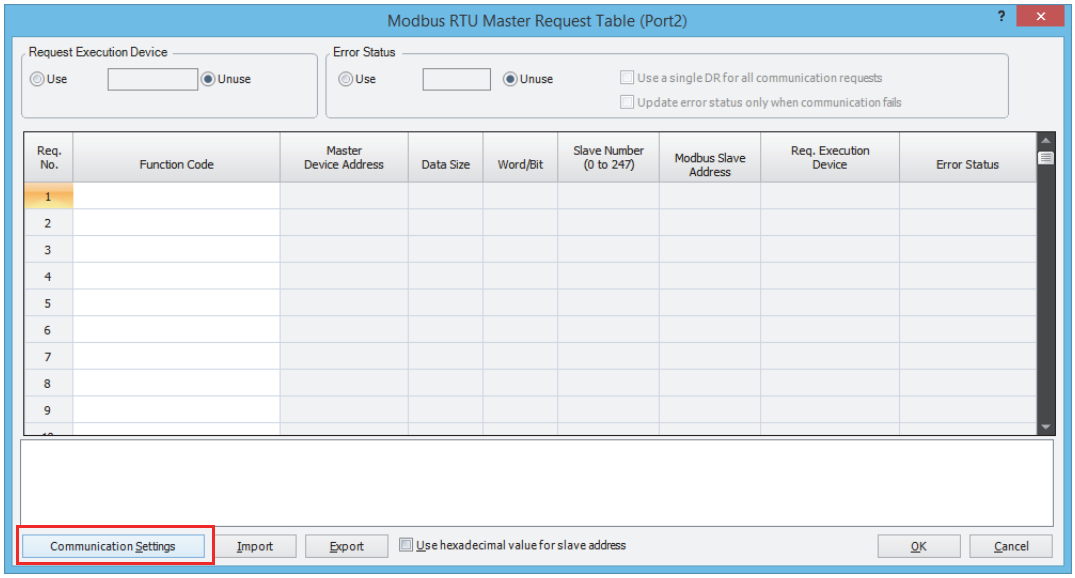
Programming Modbus RTU Master Using WindLDR

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.

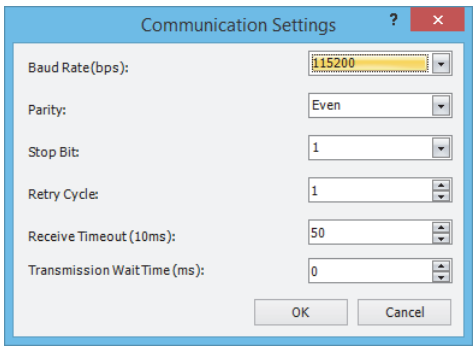
1. From the WindLDR menu bar, select **Configuration** > **Communication Ports**.
   The **Function Area Settings** dialog box for Communication Ports appears.
2. Click **Communication Mode** for the port to use and select **Modbus RTU Master**.



3. Click the **Configure** button for Port 2. The **Modbus RTU Master Request Table** appears.



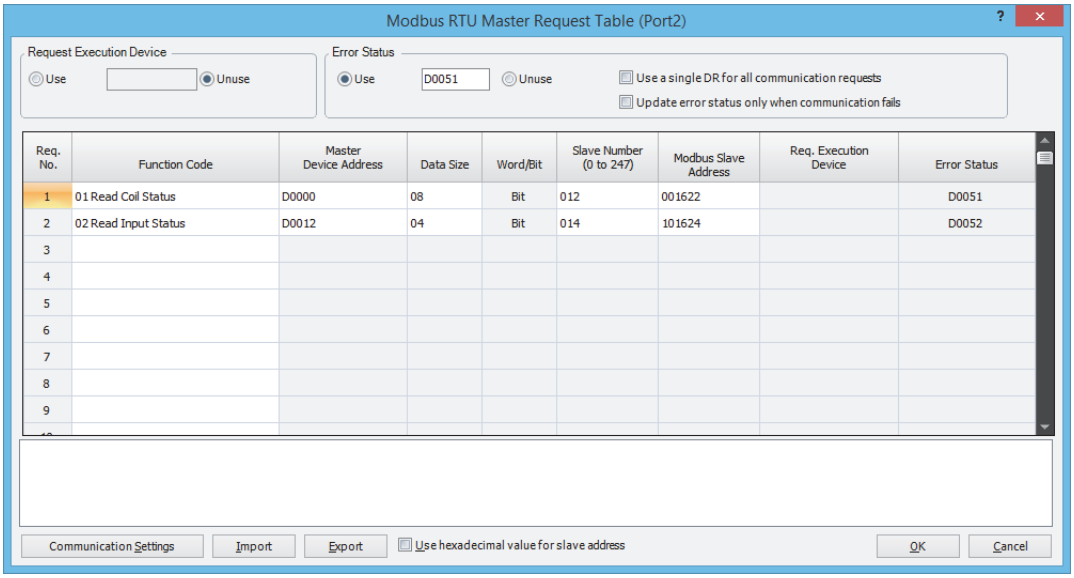
4. Click the **Communication Settings** button. The **Communication Settings** dialog box appears. Change settings, if required.



Baud Rate (bps)	9600, 19200, 38400, 57600, 115200
Parity	Even, Odd, None
Stop Bits	1 or 2
Retry Cycle	1 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 128 bits	000001 - 065535	Reads bit device statuses of Q (output), R (shift register), or M (internal relay).
02 Read Input Status	1 to 128 bits	100001 - 165535	Reads bit device statuses of I (input), T (timer contact), or C (counter contact).
03 Read Holding Registers	1 to 64 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 64 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 128 bits	000001 - 065535	Changes multiple bit device statuses of Q (output), R (shift register), or M (internal relay).
16 Preset Multiple Registers	1 to 64 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

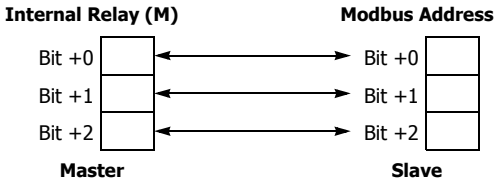
To use error status data registers, click the radio button for "Use" and designate the first data register in the Modbus RTU Master Request Table. Data registers used for storing error statuses are automatically listed in the table. When Use a single DR for all communication requests is selected, the first data register is shared by all requests.

Processing Requests

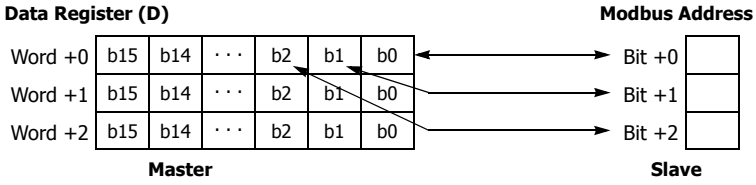
The data for Modbus communication are processed between the master and slaves as shown below.

Bit Data at Slaves (Function Codes 01, 02, 05, and 15)

- **Master Device Address: Internal Relay**

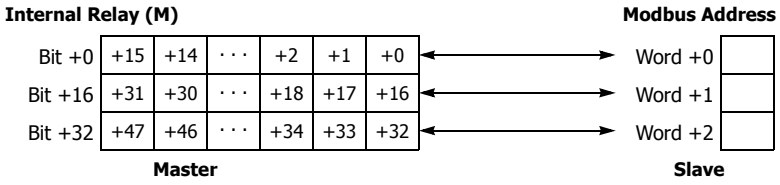


- **Master Device Address: Data Register**

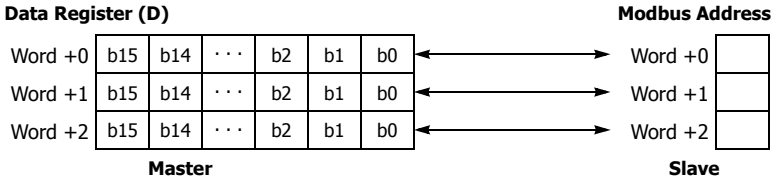


Word Data at Slaves (Function Codes 03, 04, 06, and 16)

- **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
Timeout between Characters*1	1.5 characters minimum*2	
Timeout between Frames *1	3.5 characters minimum*3	

*1 When timeout occurs, the FC6A Series MicroSmart discards the received data and waits for the first frame of the next valid communication.
*2 For communication at 19,200 bps or higher, the timeout between characters needs to be a minimum of 0.75 ms.
*3 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 - 000504	0000 - 01F7	Q0 - Q627	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	
	011001 - 017000	2AF8 - 4267	M10000 - M17497	
Input Relay (100000 and above)	100001 - 100504	0000 - 01F7	I0 - I627	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)	
	104001 - 104256	0FA0 - 109F	C256 - C511 (counter contact)	
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)	
	304001 - 304256	0AF0 - 10A0	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	
	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
	462001 - 462768	F230 - F52F	T256 - T1023 (timer preset value)	3
	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 6-9 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

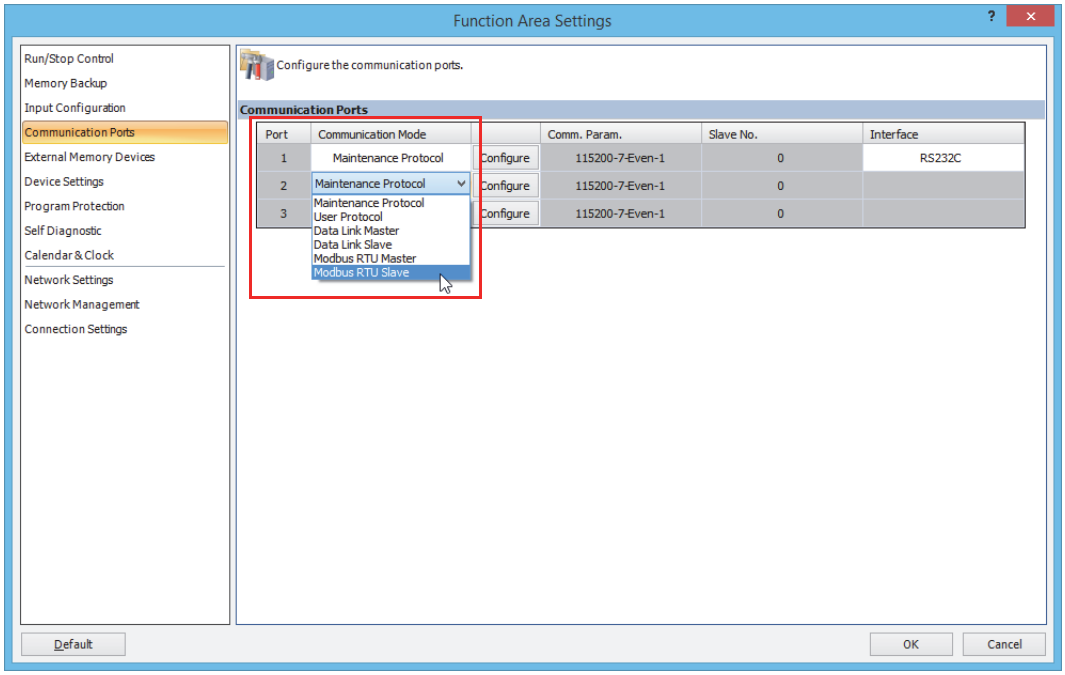
FC6A Series MicroSmart Device		Calculating Modbus Address	Calculation Example
I, Q, M	M XXX X (2): Octal (1): Decimal	((1) − (4)) × 8 + (2) + (5) Minimum address Offset	Example: M325 (32 − 0) × 8 + 5 + 1001 = 1262 Modbus address: 1262 1262 − 1 = 1261 = 04ED Slave addresses in communication: 04ED
R, T, 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 - Q627	0	1
	R000 - R255	0	701
	M0000 - M2557	0	1001
	M2560 - M7997	2560	3049
	M8000 - M8317	8000	9001
	M10000 - M17497	10000	11001
Input Relay	I0 - I627	0	100001
	T000 - T255 (timer contact)	0	101001
	C000 - C255 (counter contact)	0	101501
	T256 - T1023 (timer contact)	256	102001
	C256 - C511 (counter contact)	256	104001
Input Register	T000 - T255 (timer current value)	0	300001
	C000 - C255 (counter current value)	0	300501
	T256 - T1023 (timer current value)	256	302001
	C256 - C511 (counter current value)	256	304001
Holding Register	D0000 - D7999	0	400001
	D8000 - D8499	8000	408001
	T000 - T255 (timer preset value)	0	409001
	C000 - C255 (counter preset value)	0	409501
	D10000 - D55999	10000	410001
	T256 - T1023 (timer preset value)	256	462001
	C256 - C511 (counter preset value)	256	464001

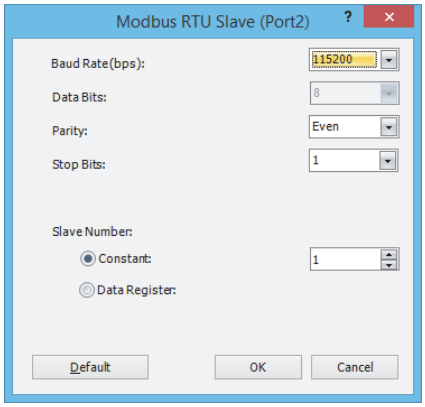
Programming Modbus Slave Using WindLDR

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), or D8103 (port 3) is used.

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 128 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 128 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 $\rightarrow$ (1 – 0) $\times$ 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 64 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 64 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 128 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 64 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)

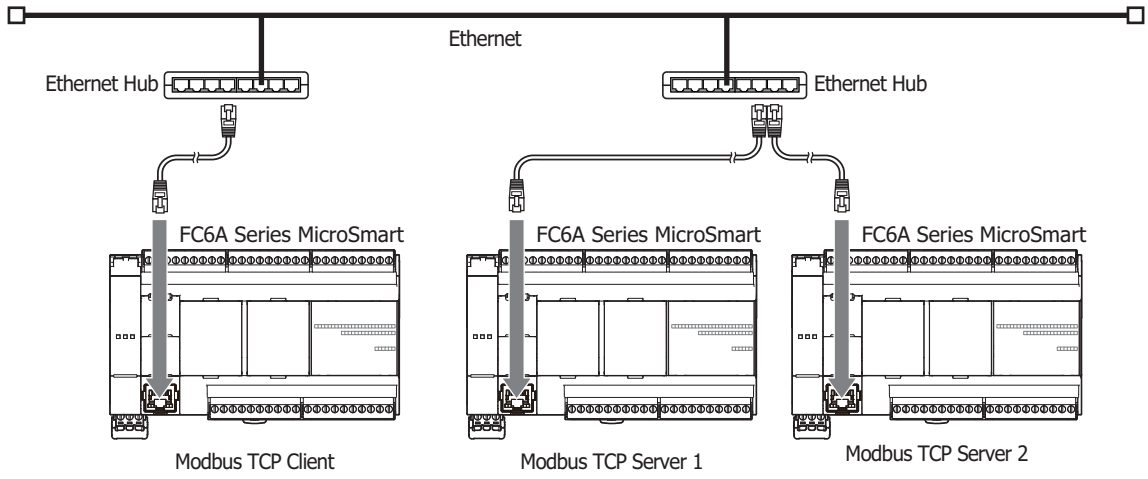
Modbus TCP Communication via Ethernet Communication

The FC6A Series MicroSmart supports Modbus TCP clients and Modbus TCP servers. Connect the Ethernet port 1 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 can be allocated to Modbus TCP clients. Each connection can communicate with one Modbus TCP server device.

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. The FC6A Series MicroSmart can allocate a maximum of eight connections for Modbus TCP servers.

For the Modbus TCP client function and how to configure it, see "Modbus TCP Client" on page 6-19. For the Modbus TCP server function and how to configure it, see "Modbus TCP Server" on page 6-24.



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.

Modbus TCP Client Specifications

Parameter	Modbus TCP Client
Slave Number	1 to 247
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 3-18.

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.

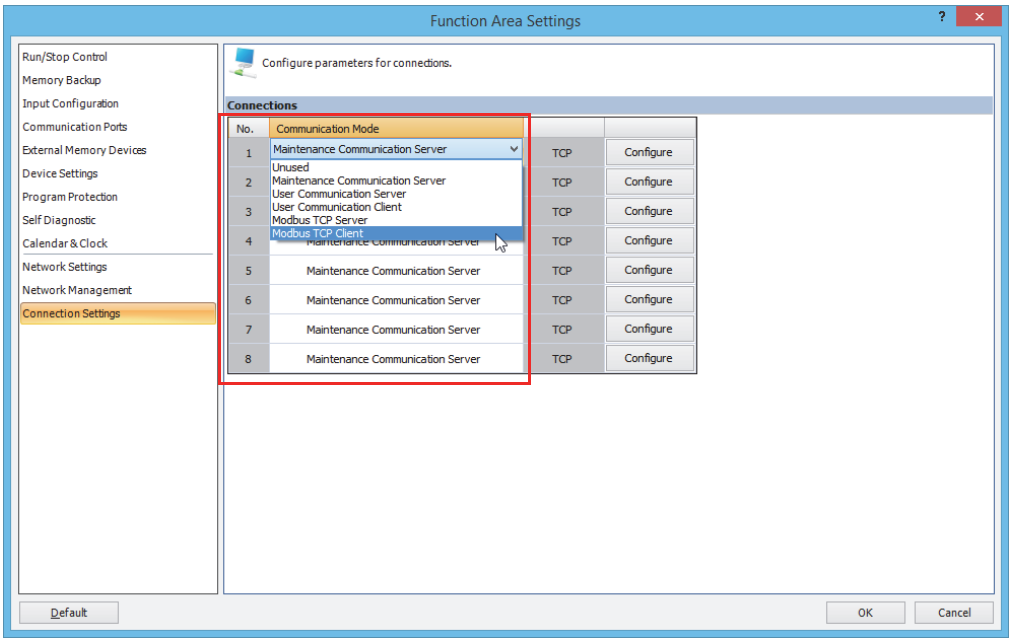
Notes:

- 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.

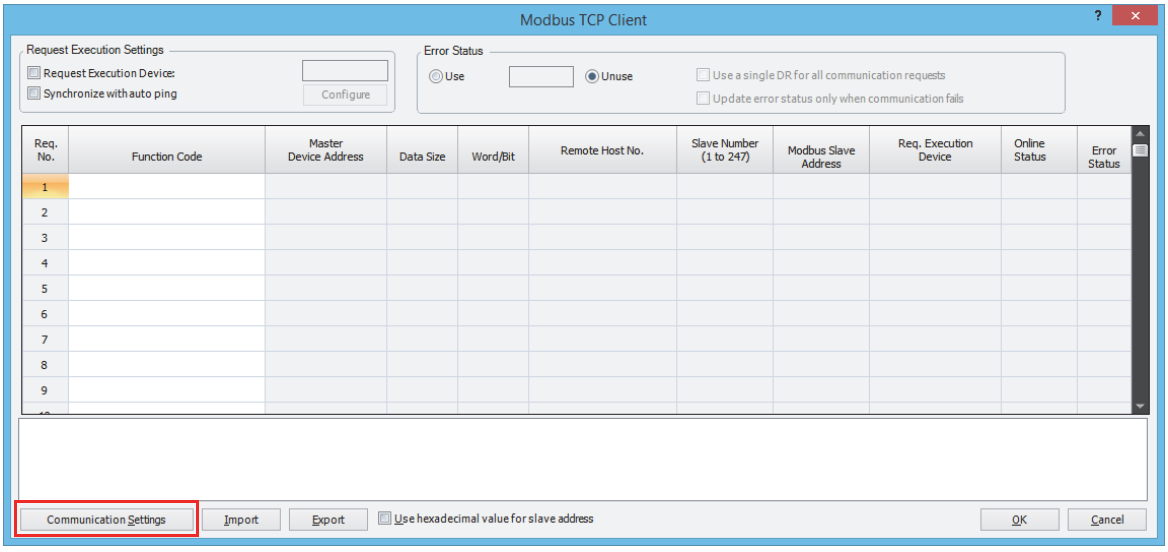
Programming WindLDR (Modbus TCP Client)

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.

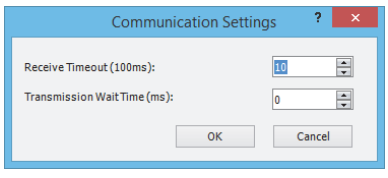
1. From the WindLDR menu bar, select **Configuration** > **Connection Settings**.
   The **Function Area Settings** dialog box appears.
2. Select **Modbus TCP Client** as the communication mode for the connection 1.



The **Modbus TCP Client** dialog box appears.



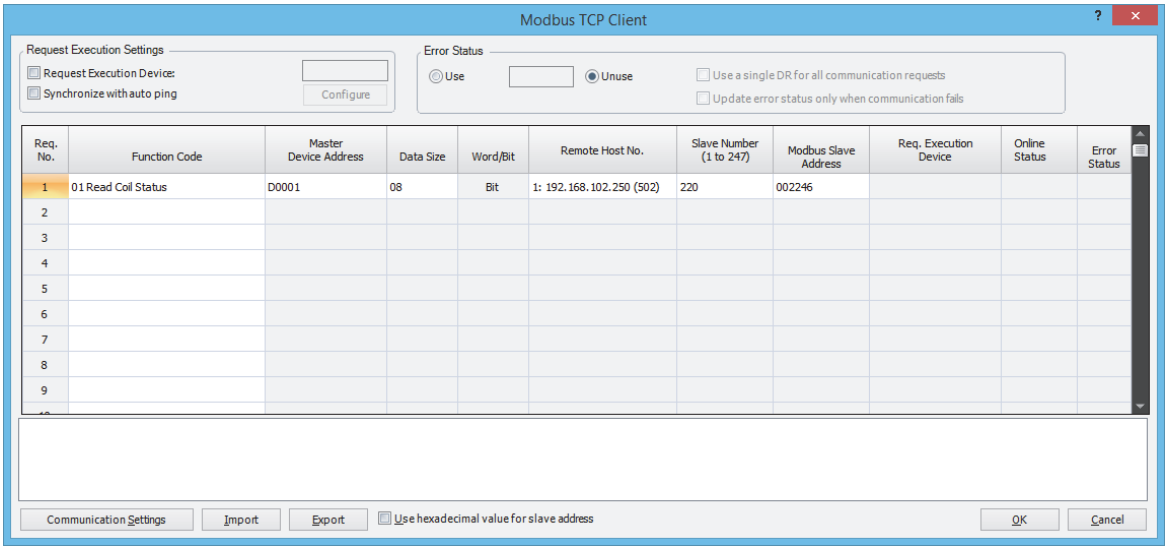
3. Click on the **Communication Settings** button.
   The **Communication Settings** dialog box appears. Configure the receive timeout and the transmission wait time.
   Click **OK**.



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 **Req. Execution Devices** and **Error Status** data registers if necessary. When using **Req. 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 128 bits	000001 - 065535	Reads bit device statuses of Q (output), R (shift register), or M (internal relay).
02 Read Input Status	1 to 128 bits	100001 - 165535	Reads bit device statuses of I (input), T (timer contact), or C (counter contact).
03 Read Holding Registers	1 to 64 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 64 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 128 bits	000001 - 065535	Changes multiple bit device statuses of Q (output), R (shift register), or M (internal relay).
16 Preset Multiple Registers	1 to 64 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 6-22.

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 1 through 247. The same slave number can be designated repeatedly for different request numbers which can be 1 through 255. The slave number is usually not referred by Modbus TCP server. Configure the slave number if Modbus TCP server requires.

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	8 (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 when all eight connections 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 - 000504	0000 - 01F7	Q0 - Q627	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	
	011001 - 017000	2AF8 - 4267	M10000 - M17497	
Input Relay (100000 and above)	100001 - 100504	0000 - 01F7	I0 - I627	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)	
	104001 - 104256	0FA0 - 109F	C256 - C511 (counter contact)	
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)	
	304001 - 304256	0AF0 - 10A0	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	
	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
	462001 - 462768	F230 - F52F	T256 - T1023 (timer preset value)	3
	464001 - 464256	FA00 - FAFF	C256 - C511 (counter preset value)	

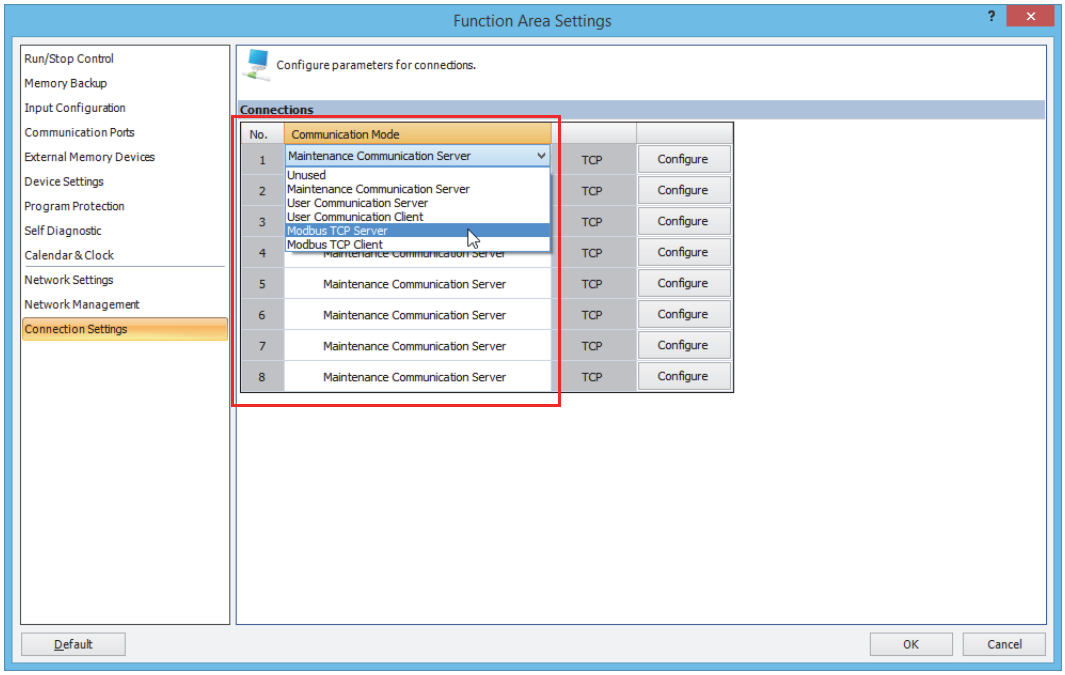
*1 Slave addresses in communication are 4-digit values used in the address portion of the communication frame. Subtract 1 from the lower 5 digits of the slave address and store that value in hexadecimal. For details, see "Modbus RTU Slave Communication" on page 6-8.

*2 Access within the device range for the FC6A Series MicroSmart type used.

Programming WindLDR (Modbus TCP Server)

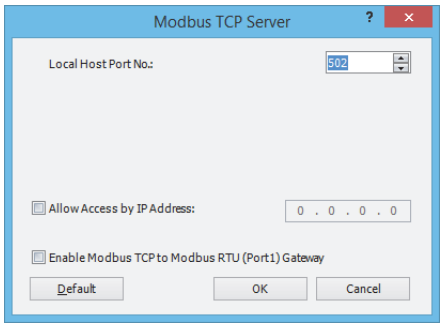
In order to use Modbus TCP server, configure the Modbus TCP server in the **Function Area Settings** dialog box and then download the user program to the FC6A Series MicroSmart.

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

RTU Mode Communication Format	Idle 3.5 characters	Slave No.	Function Code	Data	CRC	Idle 3.5 characters
		1 byte	1 byte	N bytes	2 bytes	

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

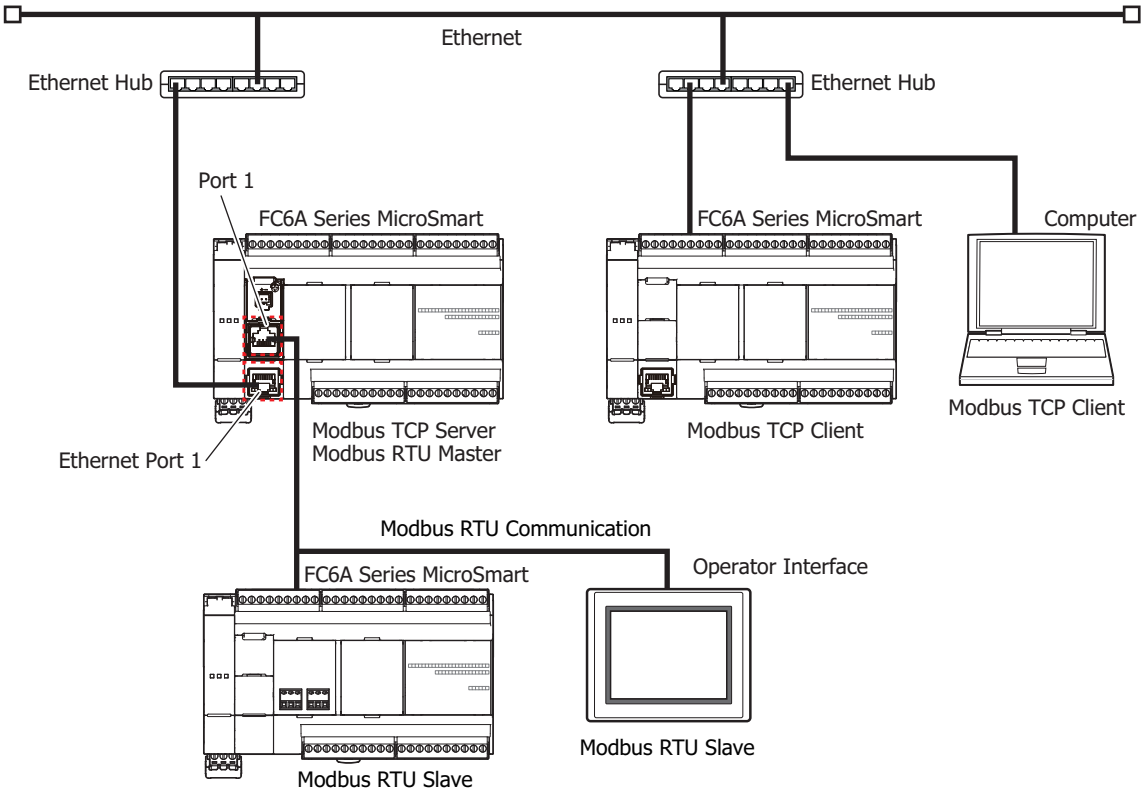
Designate required data for each function.

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.

This function can read and write devices on a Modbus RTU slave device connected to port 1 of the FC6A Series MicroSmart from the Modbus TCP client device connected to Ethernet port 1 of the FC6A Series MicroSmart.

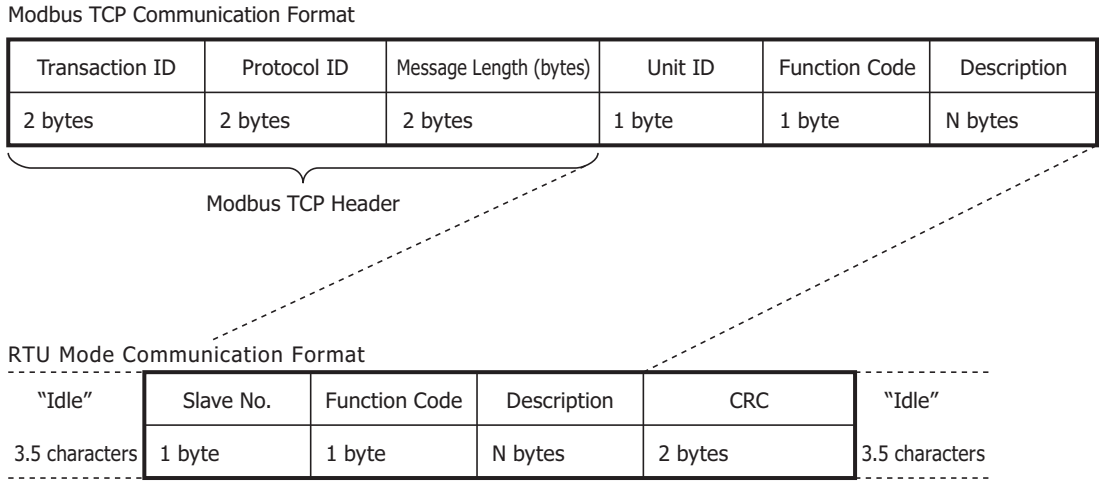
The Modbus RTU pass-through function is only supported between Ethernet port 1 and port 1 on the FC6A Series MicroSmart.



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.



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.

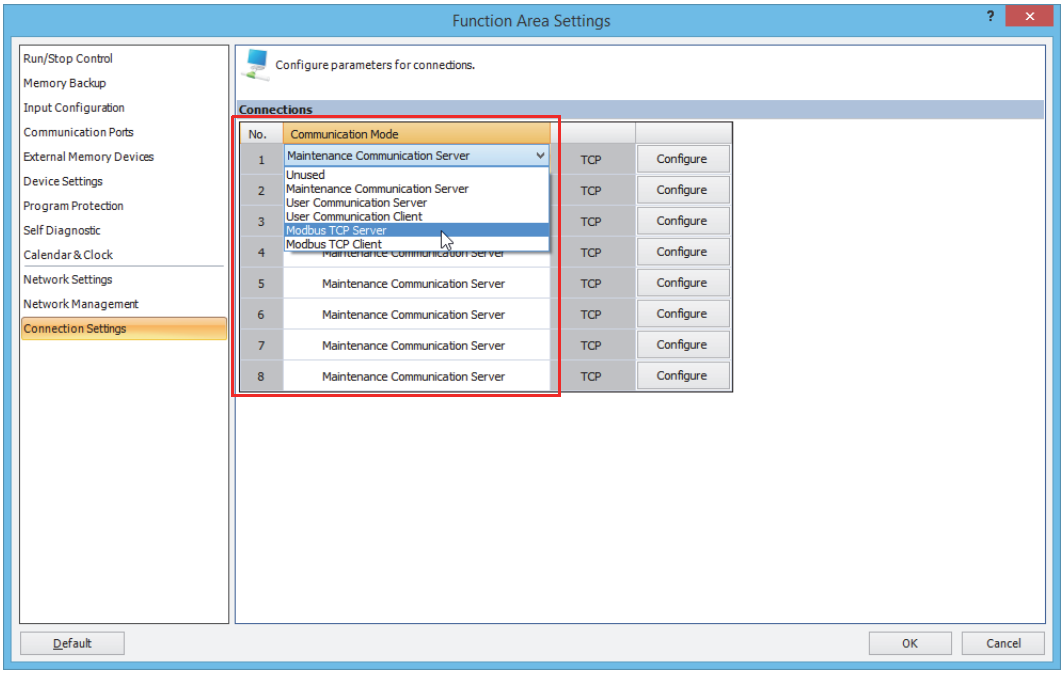
Modbus RTU pass-through function specifications

Supported Port	Ethernet Port 1	Port 1
Communication Mode	Modbus TCP server (slave)	Modbus RTU master
Enable/disable Settings	Configure in **Function Area Settings**.	None
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.	

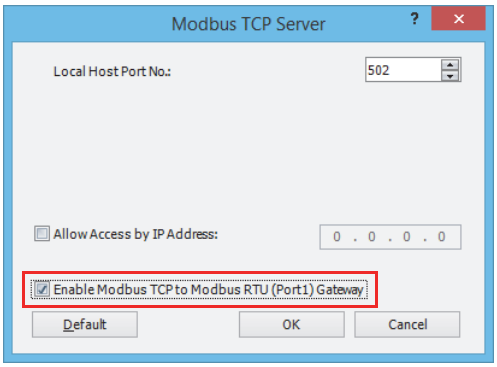
Programming WindLDR

To use the Modbus RTU pass-through function, download a user program to the FC6A Series MicroSmart configured in **Function Area Settings**.

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.

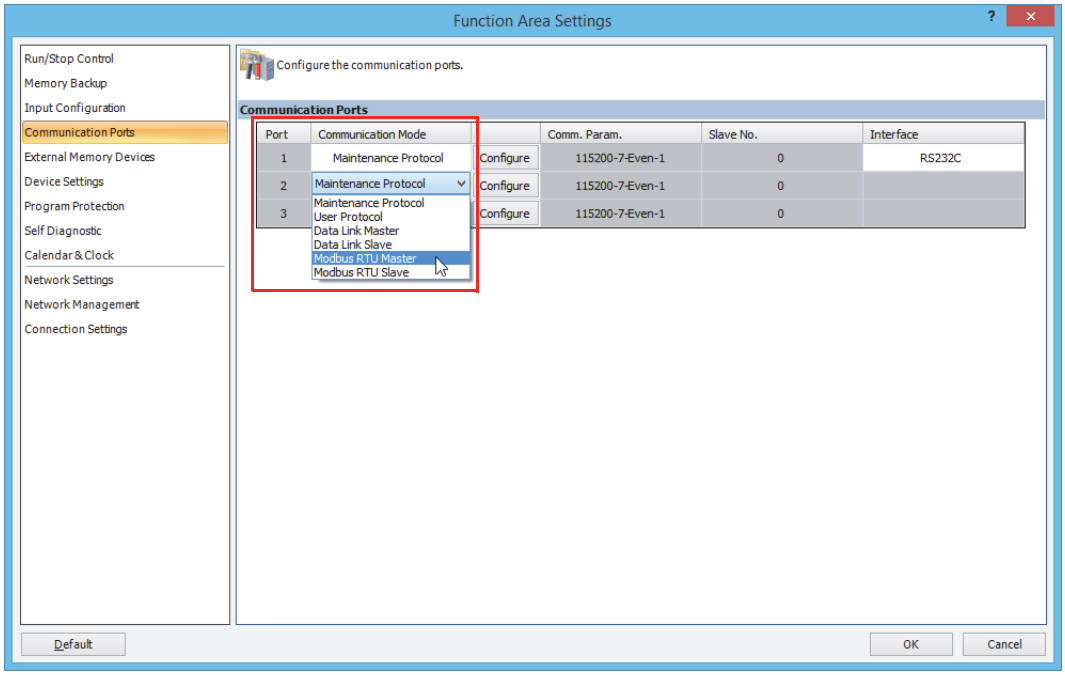


3. Select the **Enable Modbus TCP to Modbus RTU (Port1) Gateway** check box and configure the local host port number and the allow access by IP address.
   Specify a number that is different from the other server connections for the local host port number of the port that will use the pass-through function.



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 6-2.

7. 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.

Data link communication can be used by setting port 1 of the FC6A Series MicroSmart to RS485 or by using an RS485 communication cartridge in port 2 or port 3.

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.

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)

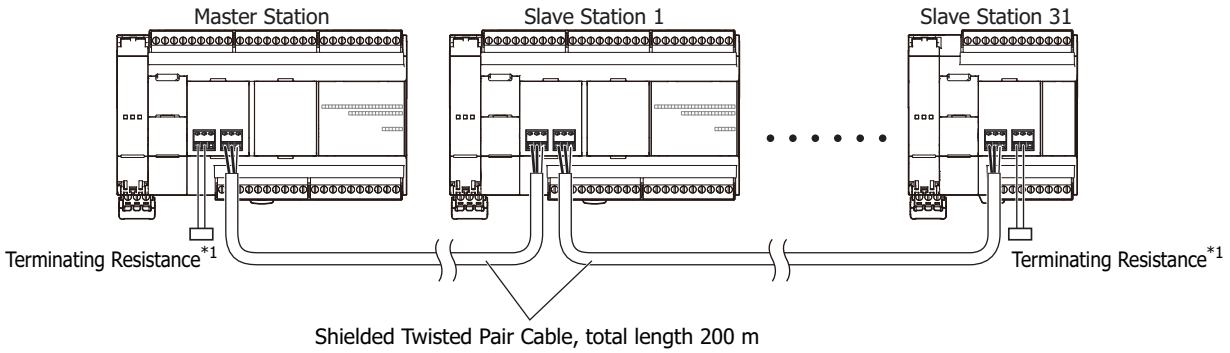
The master station sends the appropriate amount of data (in units of words) to the slave stations. The slave stations that received this transmission send an amount of data matching the number of words received to the master station. In this way data is exchanged. The data refresh is performed on the 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

D8069	Communication Error
D900 - D901	Transmit Data
D906 - D907	Receive Data
D8070	Communication Error
D912 - D913	Transmit Data
D918 - D919	Receive Data
D8071	Communication Error
D924 - D925	Transmit Data
D930 - D931	Receive Data
D8072	Communication Error
D936 - D937	Transmit Data
D942 - D943	Receive Data
⋮	⋮
D8098	Communication Error
D1248 - D1249	Transmit Data
D1254 - D1255	Receive Data
D8099	Communication Error
D1260 - D1261	Transmit Data
D1266 - D1267	Receive Data

Slave Stations

D8069	Communication Error	Slave Station 1
D900 - D901	Transmit Data	
D906 - D907	Receive Data	
D8069	Communication Error	Slave Station 2
D900 - D901	Transmit Data	
D906 - D907	Receive Data	
D8069	Communication Error	Slave Station 3
D900 - D901	Transmit Data	
D906 - D907	Receive Data	
D8069	Communication Error	Slave Station 4
D900 - D901	Transmit Data	
D906 - D907	Receive Data	
⋮	⋮	⋮
D8069	Communication Error	Slave Station 30
D900 - D901	Transmit Data	
D906 - D907	Receive Data	
D8069	Communication Error	Slave Station 31
D900 - D901	Transmit Data	
D906 - D907	Receive Data	

Example 2: Transmit Data 6 Words and Receive Data 6 Words

Master Station

D8069	Communication Error
D900 - D905	Transmit Data
D906 - D911	Receive Data
D8070	Communication Error
D912 - D917	Transmit Data
D918 - D923	Receive Data
D8071	Communication Error
D924 - D929	Transmit Data
D930 - D935	Receive Data
D8072	Communication Error
D936 - D941	Transmit Data
D942 - D947	Receive Data
⋮	⋮
D8098	Communication Error
D1248 - D1253	Transmit Data
D1254 - D1259	Receive Data
D8099	Communication Error
D1260 - D1265	Transmit Data
D1266 - D1271	Receive Data

Slave Stations

D8069	Communication Error	Slave Station 1
D900 - D905	Transmit Data	
D906 - D911	Receive Data	
D8069	Communication Error	Slave Station 2
D900 - D905	Transmit Data	
D906 - D911	Receive Data	
D8069	Communication Error	Slave Station 3
D900 - D905	Transmit Data	
D906 - D911	Receive Data	
D8069	Communication Error	Slave Station 4
D900 - D905	Transmit Data	
D906 - D911	Receive Data	
⋮	⋮	⋮
D8069	Communication Error	Slave Station 30
D900 - D905	Transmit Data	
D906 - D911	Receive Data	
D8069	Communication Error	Slave Station 31
D900 - D905	Transmit 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 7-4.

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.

M8007Data 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.

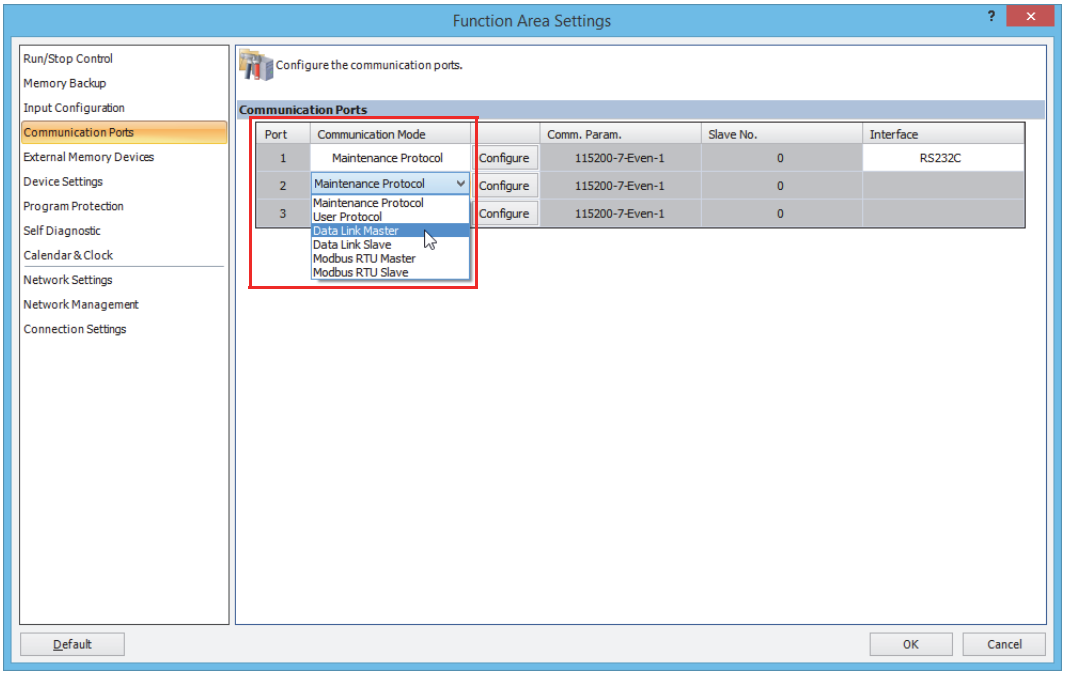
Programming WindLDR

The Communication page in the **Function Area Settings** is used to program the data link master and slave stations.

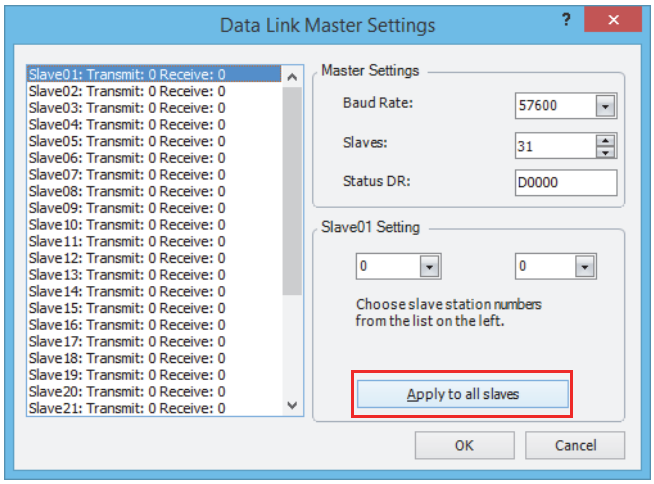
Since these settings relate to the user program, the user program must be downloaded to the CPU module after changing any of these settings.

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.

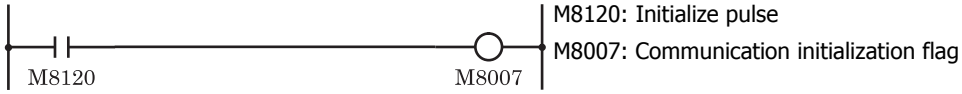
This concludes configuring the settings.

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.

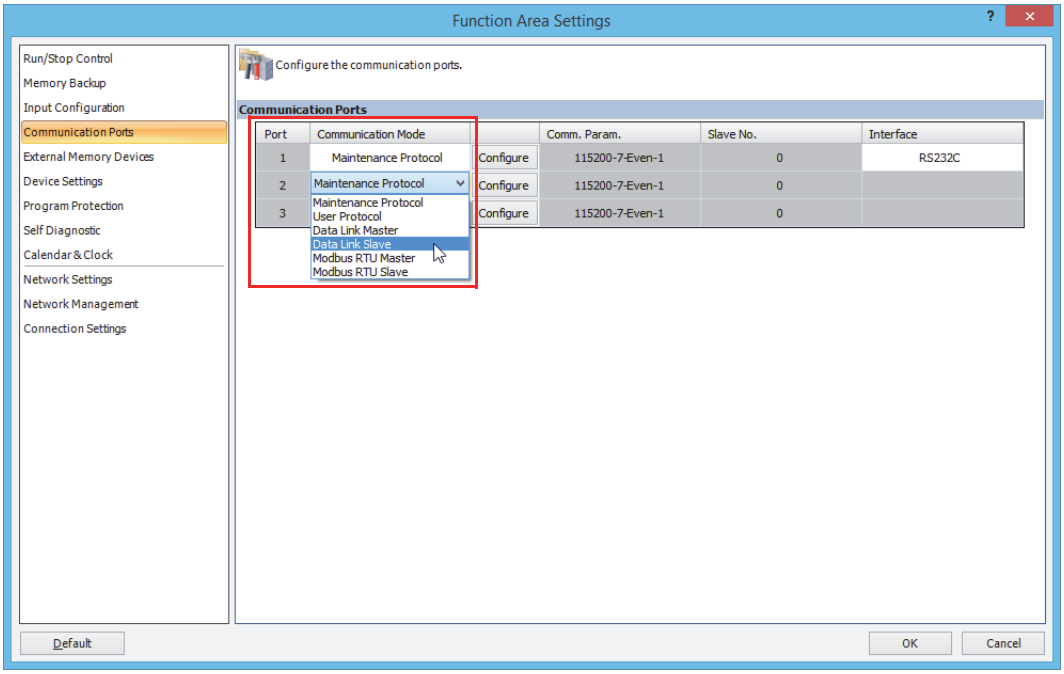
Stop and then run the master station to have it recognize the slave stations.



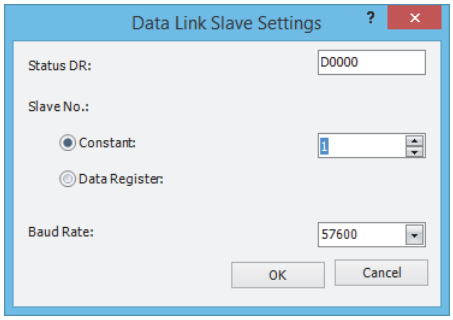
When operation (RUN) starts, M8007 is turned on for a period of one scan, and the data link is initialized.

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 Data Link Station Number and the Baud Rate.
   Select **Constant** or **Data Register** under **Data Link Station Number**. If you select **Constant**, set the data link station number. If you select **Data Register**, write the data link station number 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 **Data Link Station Number** in the **Data Link Slave Settings** dialog box, you can change the data link station number of the slave station without downloading the user program by writing the data link station number into the following special data registers.

Communication Port	Special Data Register
Port 1	D8100
Port 2	D8102
Port 3	D8103

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 data link station numbers 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 = $\sum$ Trf = $\sum$ {4.2 ms + 2.4 ms $\times$ (Transmit Words + Receive Words) + 1 scan time}

[Baud Rate 38,400 bps] Trfn = $\sum$ Trf = $\sum$ {2.2 ms + 1.3 ms $\times$ (Transmit Words + Receive Words) + 1 scan time}

[Baud Rate 57,600 bps] Trfn = $\sum$ Trf = $\sum$ {1.6 ms + 0.9 ms $\times$ (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 $\times$ (6 + 6) + 20 ms} $\times$ 8 = 424.0 ms

[Baud Rate 38,400 bps] Trf8 = {2.2 ms + 1.3 ms $\times$ (6 + 6) + 20 ms} $\times$ 8 = 302.4 ms

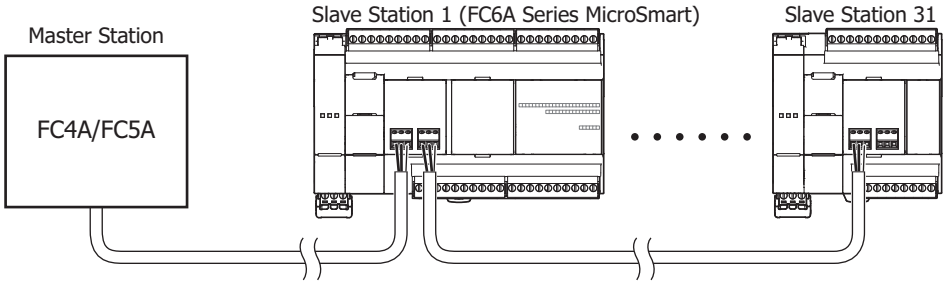
[Baud Rate 57,600 bps] Trf8 = {1.6 ms + 0.9 ms $\times$ (6 + 6) + 20 ms} $\times$ 8 = 259.2 ms

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 Type.

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

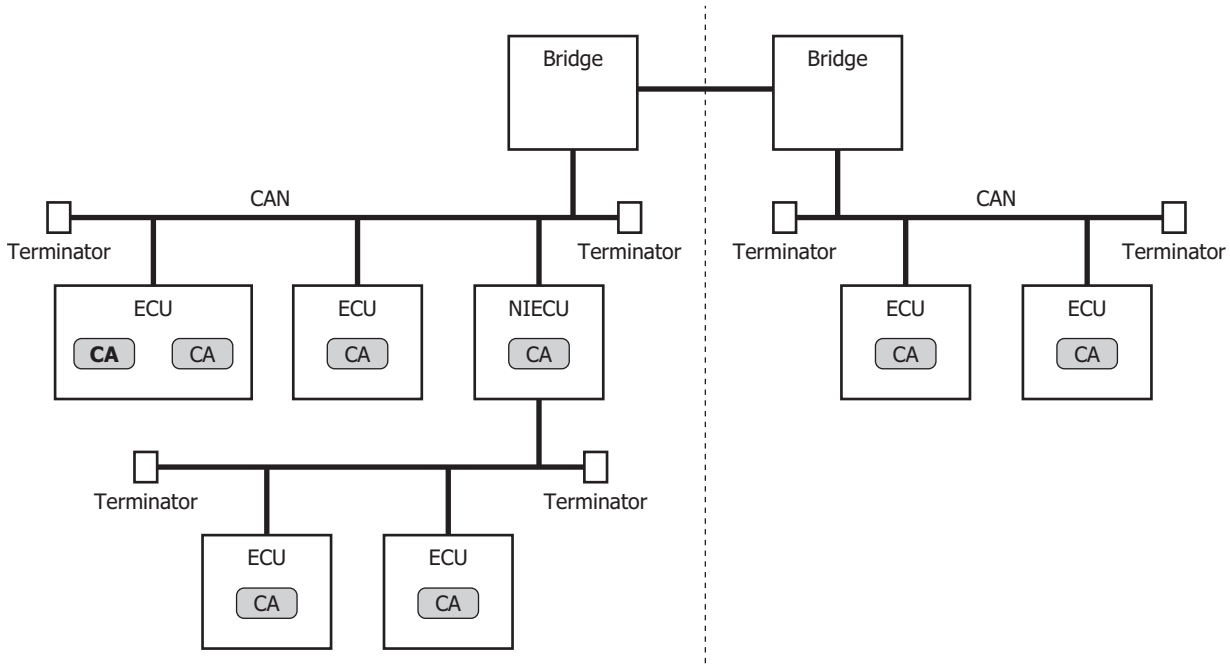
Overview of J1939 Communication over CAN

The CAN J1939 All-in-One Type is compatible with CAN communication based on SAE J1939. The CAN J1939 All-in-One Type 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.



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.

The CAN J1939 All-in-One Type 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 Type 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 Type can exchange messages with the set local address if its priority is high.
If the CAN J1939 All-in-One Type priority is low, the local address is 254 (null address).

If the Arbitrary Address Capable bit is 1

The CAN J1939 All-in-One Type 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 Type 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.

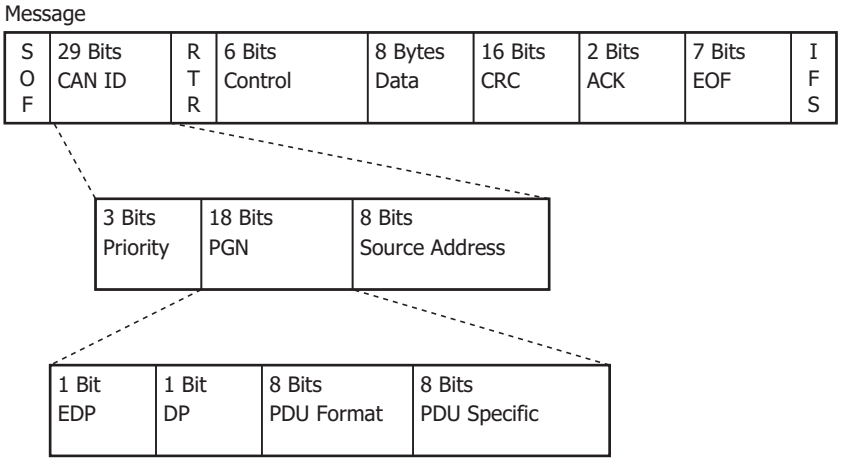
Example: SPNs that make up PGN 1792 General Purpose Valve Pressure*1

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).



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.

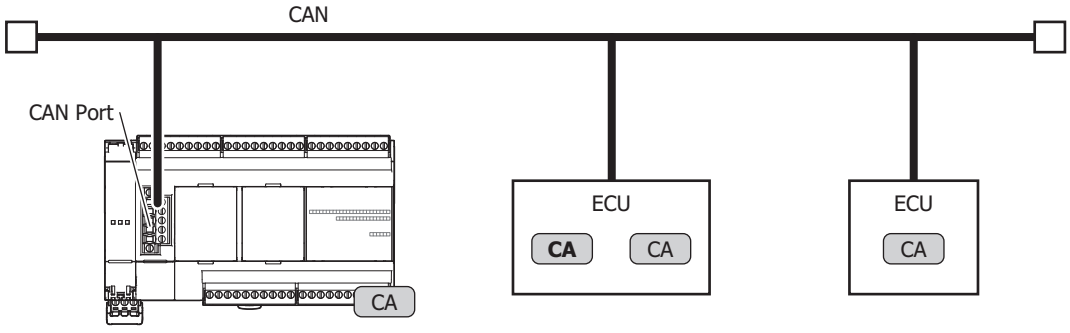
PDU 2 format

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 CAN J1939 All-in-One Type

The CAN J1939 All-in-One Type 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 Type) (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 Type

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.

0: Communication prohibited (default)

1: Communication permitted

J1939 Online Status (M8301)
Indicates the J1939 communication online status. While offline, messages cannot be exchanged because the CAN J1939 All-in-One Type is not connected to a J1939 communication network.

0: Offline

1: 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.

0: Local address unconfirmed

1: 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).

0: No communication error

1: 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 Type 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 Type 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.

0: Bus off has not occurred

1: 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

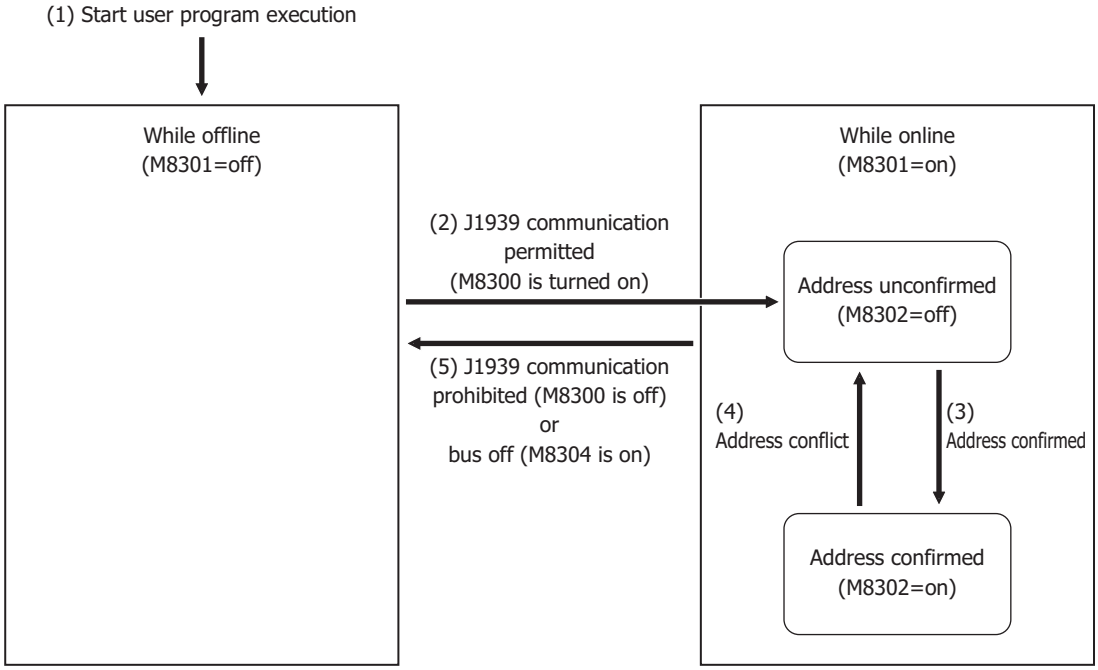
J1939 Communication Error Code (D8052)
This register stores the J1939 communication error code. To initialize this register, write "0".

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	CAN communication error	An error/warning interrupt occurred on the CAN.
2003	CAN communication error	An error has occurred on the CAN controller.
2004	CAN communication 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.
Before acquiring 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.

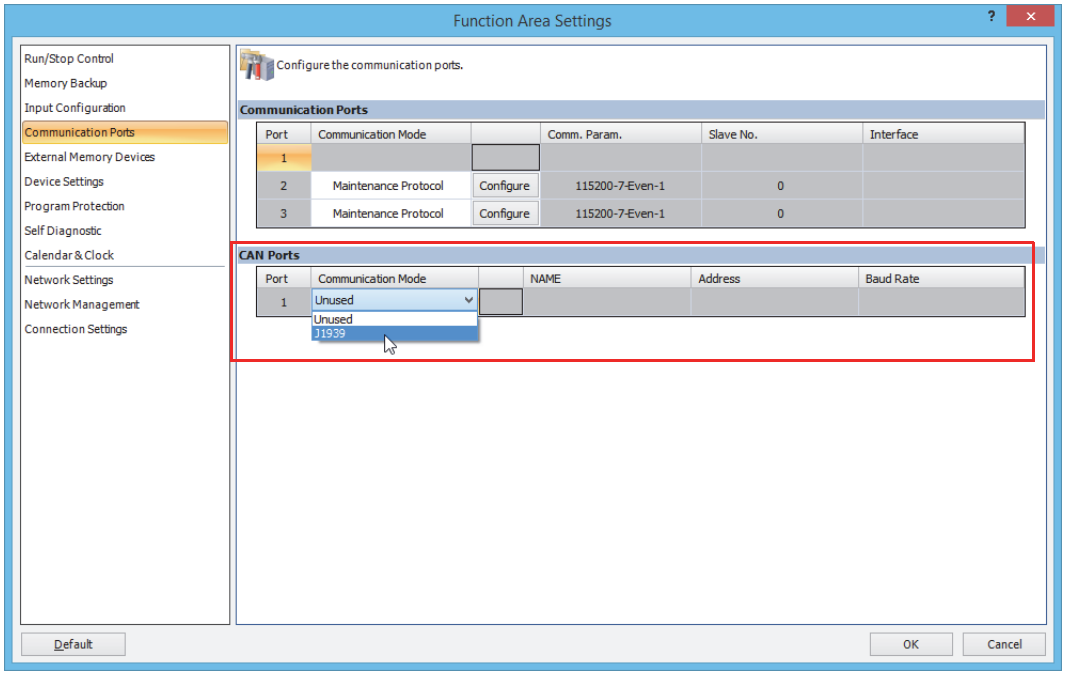
J1939 Communication Settings

This section describes the operation procedure and details about the items related to J1939 communication.

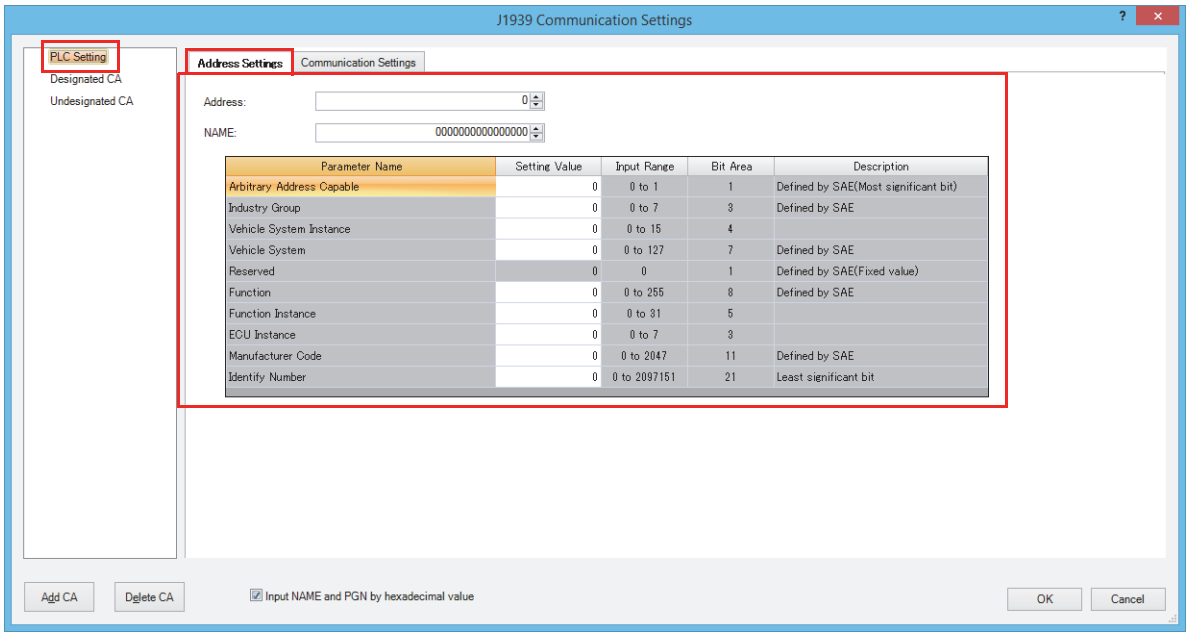
Programming WindLDR

Configure J1939 communication according to the usage environment.

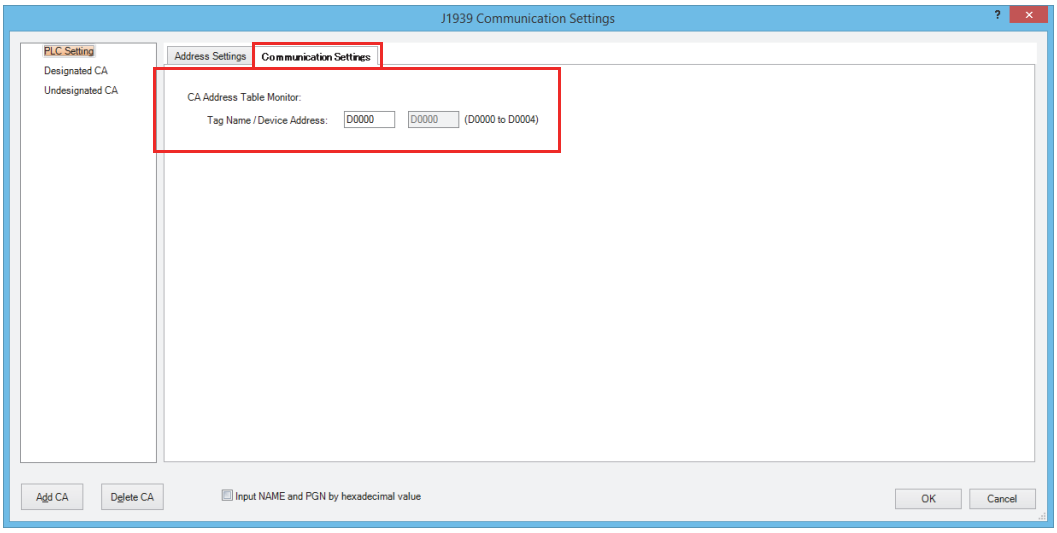
1. On the **Configuration** tab, in the **Function Area Settings** group, click **Comm. Port.**
   The **Function Area Settings** dialog box is displayed.
2. Select **J1939** from the CAN port group communication mode.
   The **J1939 Communication Settings** dialog box is displayed.



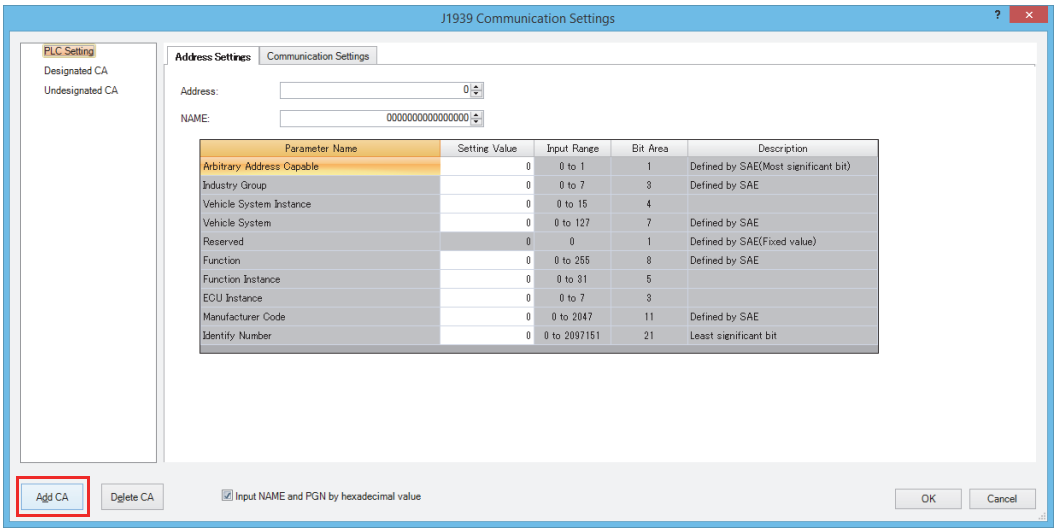
3. Under **PLC Setting**, on the **Address Settings** tab, set the local address, device name (NAME), and CA address table.
   Set the local address and device name (NAME).



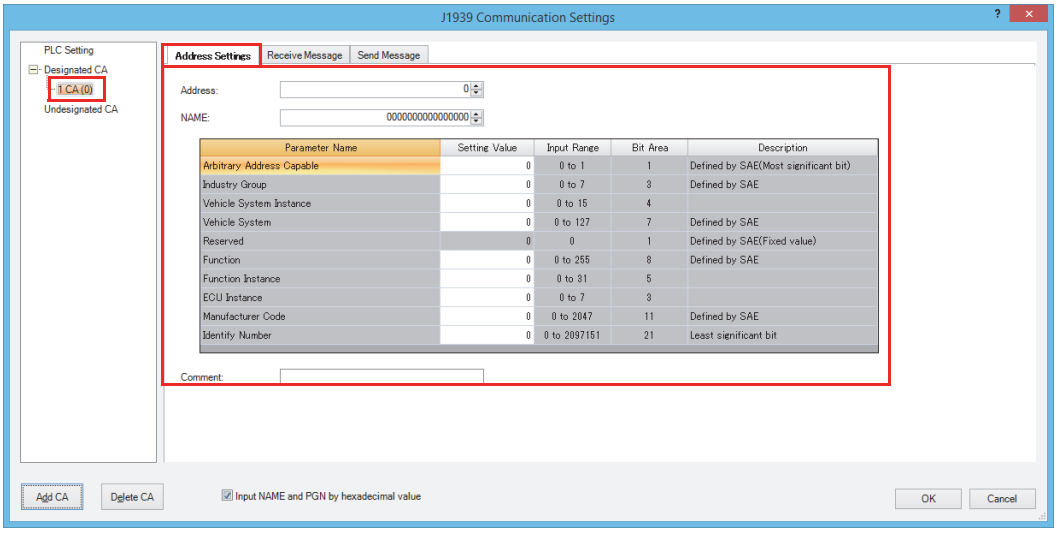
4. Click the **Communication Settings** tab and set the device address to store the CA address table.



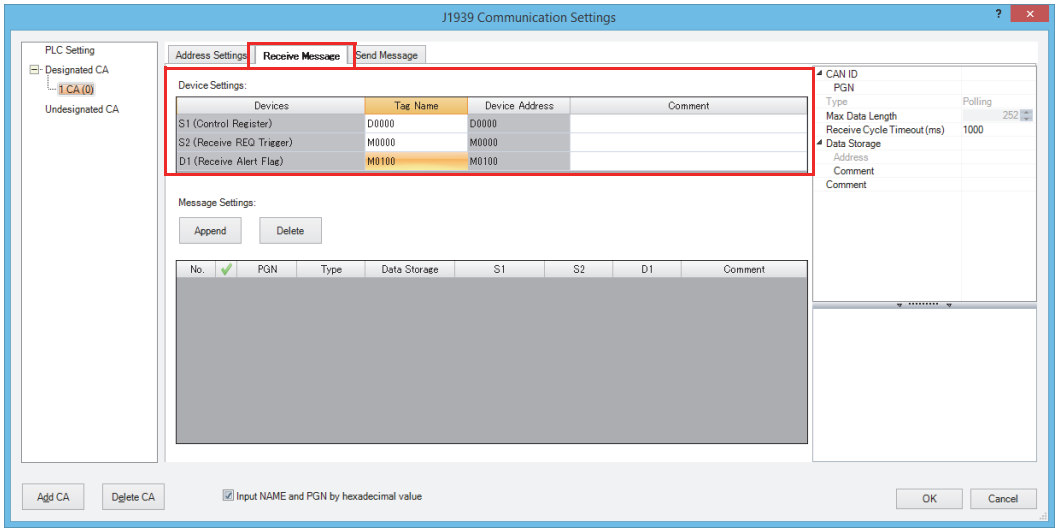
5. Click **Add CA** to add a node to configure a CA.



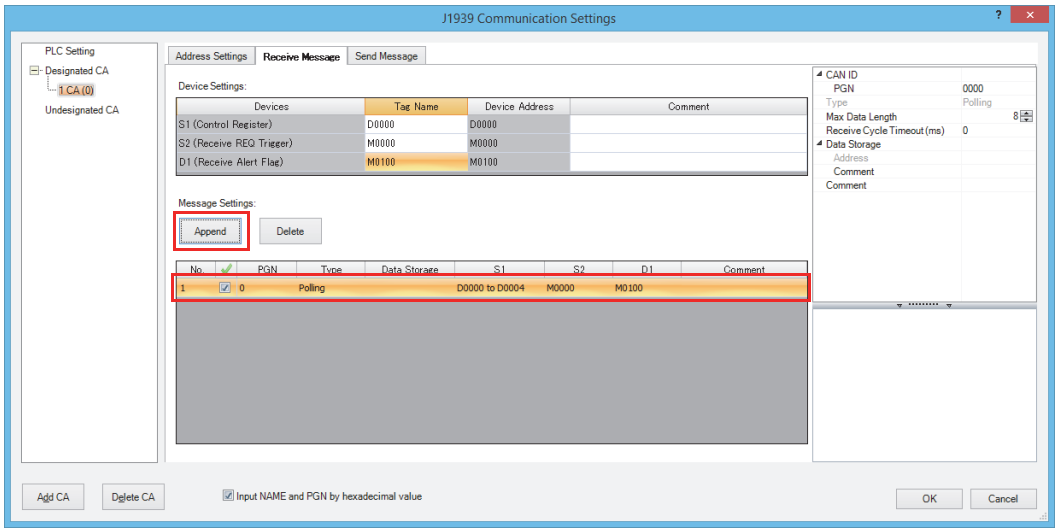
6. Click the node that corresponds to the CA to configure and click the **Address Settings** tab. Configure the address and the device name for the CA.



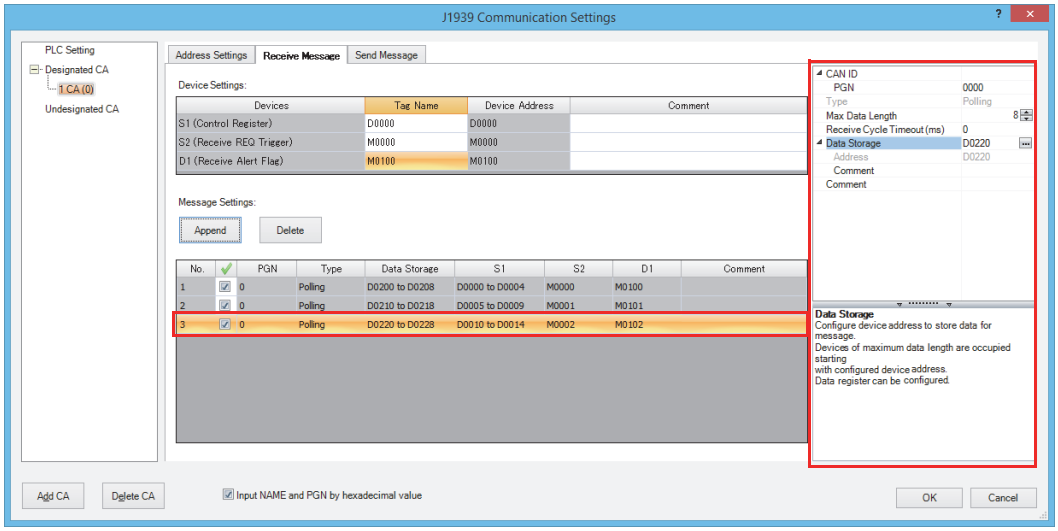
7. Click the **Receive Message** tab and first configure the devices that will used with received messages.



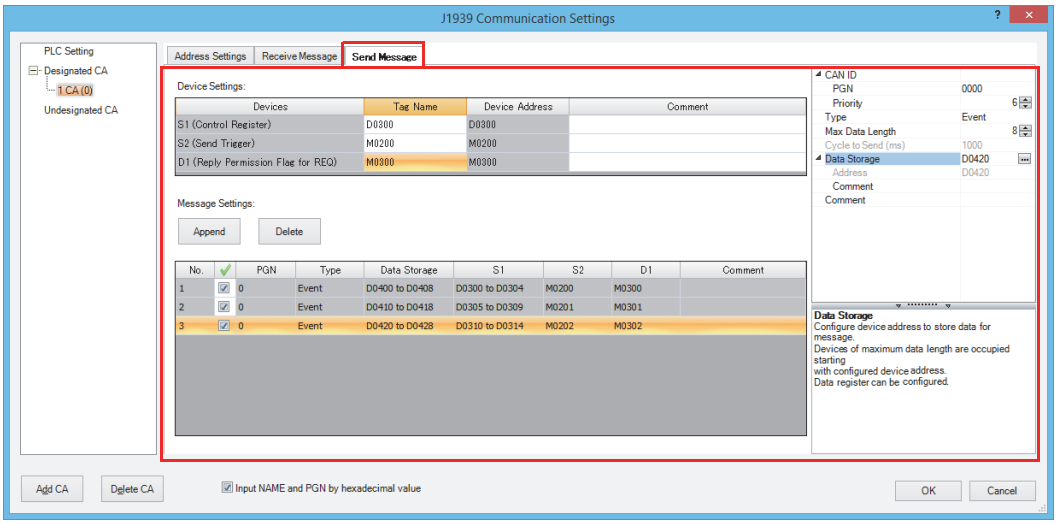
8. Click **Append** to add a received message.



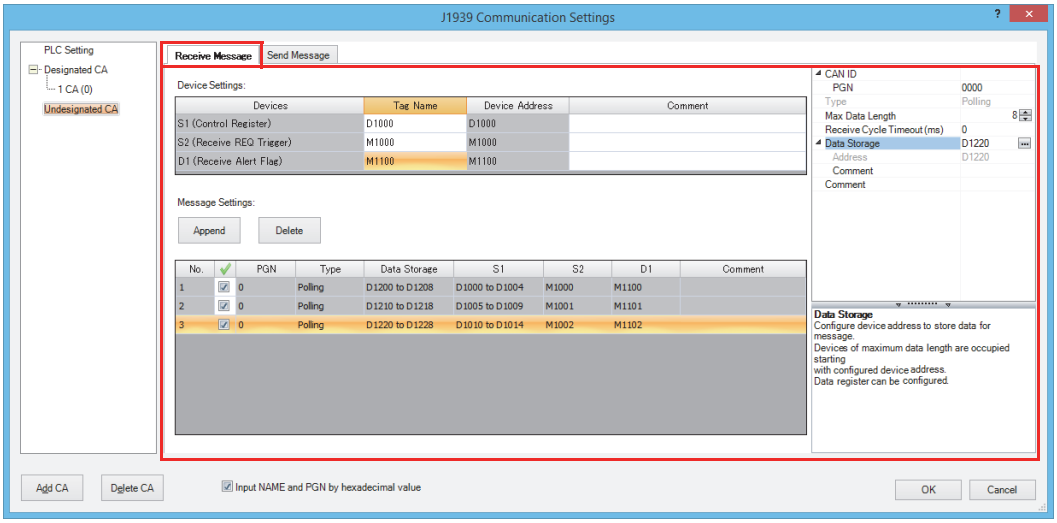
9. Select any of the cells for the message settings to display the parameters for the messages on the right side. Configure the items.

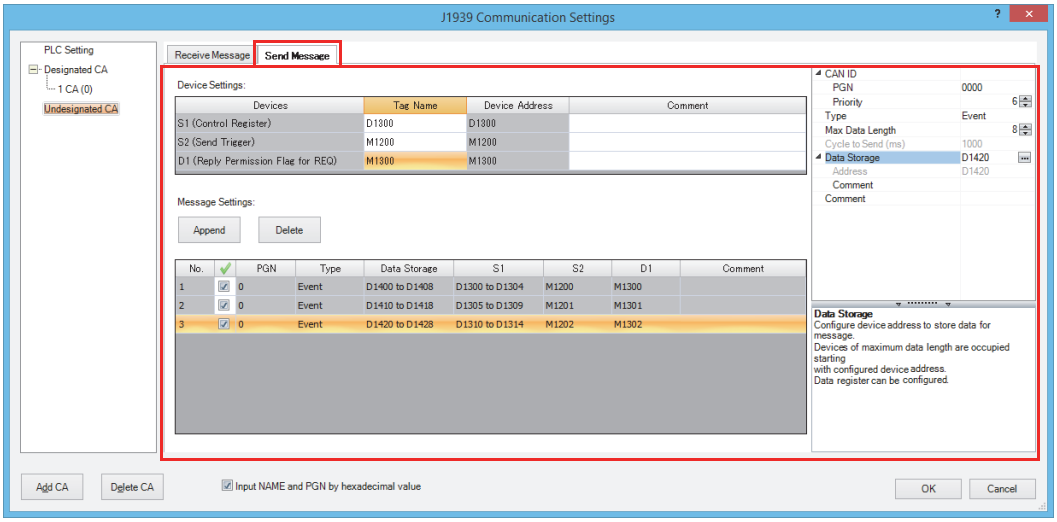


10. Click the **Send Message** tab and configure it in the same manner as the **Receive Message** tab.



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

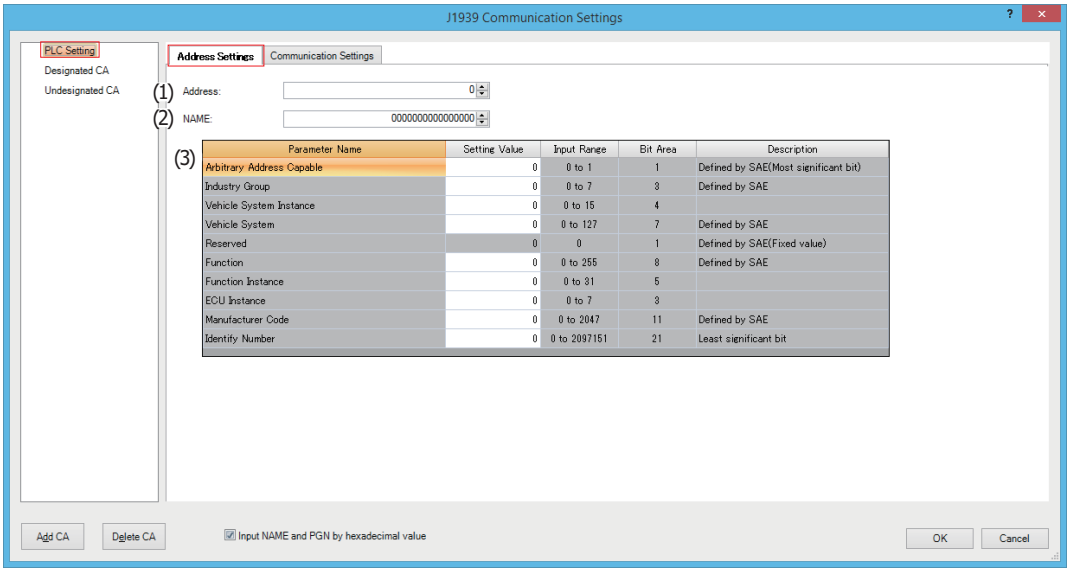
This section describes the **J1939 Settings** dialog box parameter settings required to use J1939 communication.

PLC Setting

These settings configure the CAN J1939 All-in-One Type local address, local NAME, and CA address table.

■ **Address Settings tab**

This tab configures the CAN J1939 All-in-One Type local address and local NAME.



(1) Address

Sets the initial value of the CAN J1939 All-in-One Type 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 Type 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.

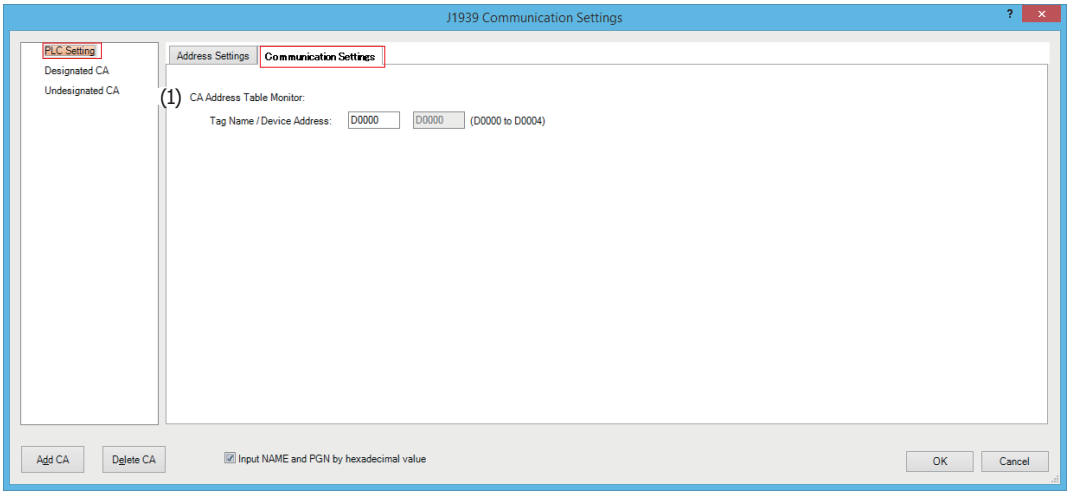
The device name (NAME) is composed of the set values in the device name field list (3). When the device name (NAME) is changed, those changes are reflected under **Preset Value** in the device name field list.

(3) Device name field list

Displays each field of the device name (NAME) (2). When a **Preset 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 Type (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 Type 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 Type exchanges messages based on the addresses in the CA address table.

CA Address Table Monitor Definition

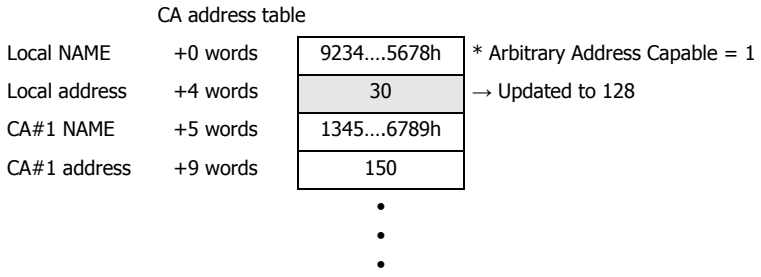
CA	Storage Destination	Function	Setting Details
Local	Starting number+0	Local NAME	Stores the CAN J1939 All-in-One Type NAME.
	Starting number+1		
	Starting number+2		
	Starting number+3		
	Starting number+4	Local address*1	Stores the CAN J1939 All-in-One Type address. If the address changes while online, 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+5×N	CA#N NAME*2	Stores the CA#N NAME.
	Starting number+5×N+1		
	Starting number+5×N+2		
	Starting number+5×N+3		
	Starting number+5×N+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.

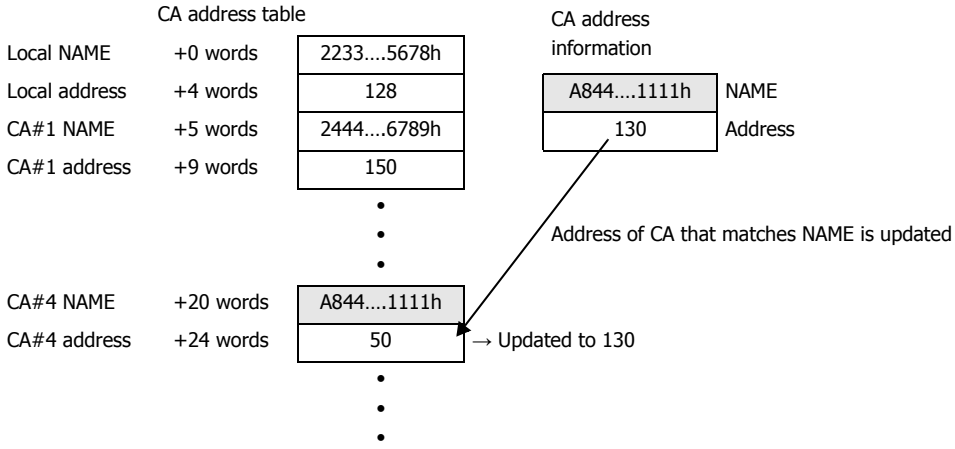
*2 If the address is outside the range of 0 to 253, messages cannot be exchanged with the corresponding CA.

*3 N is 1 to 253.

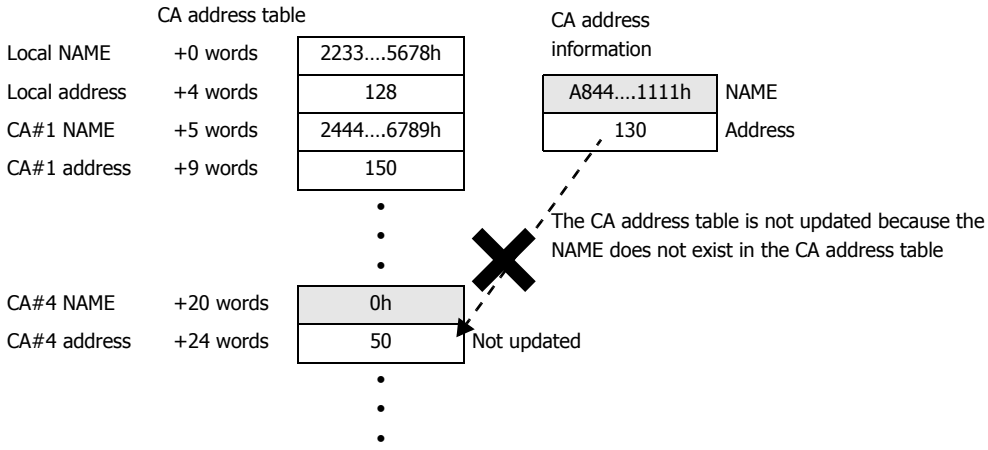
Example1: When updated by local address conflict



Example 2: When updated by receiving CA address information



Example3: When CA#4 NAME is set to 0 in Example 2

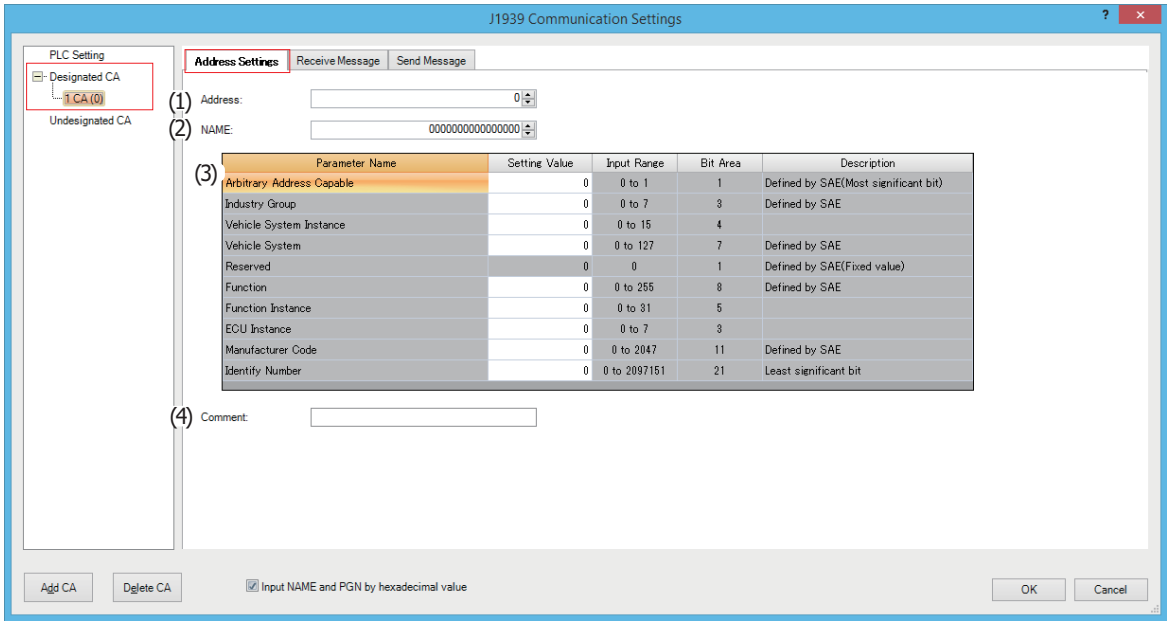


Designated CA

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 **Preset Value** in the device name field list.

(3) Device name field list

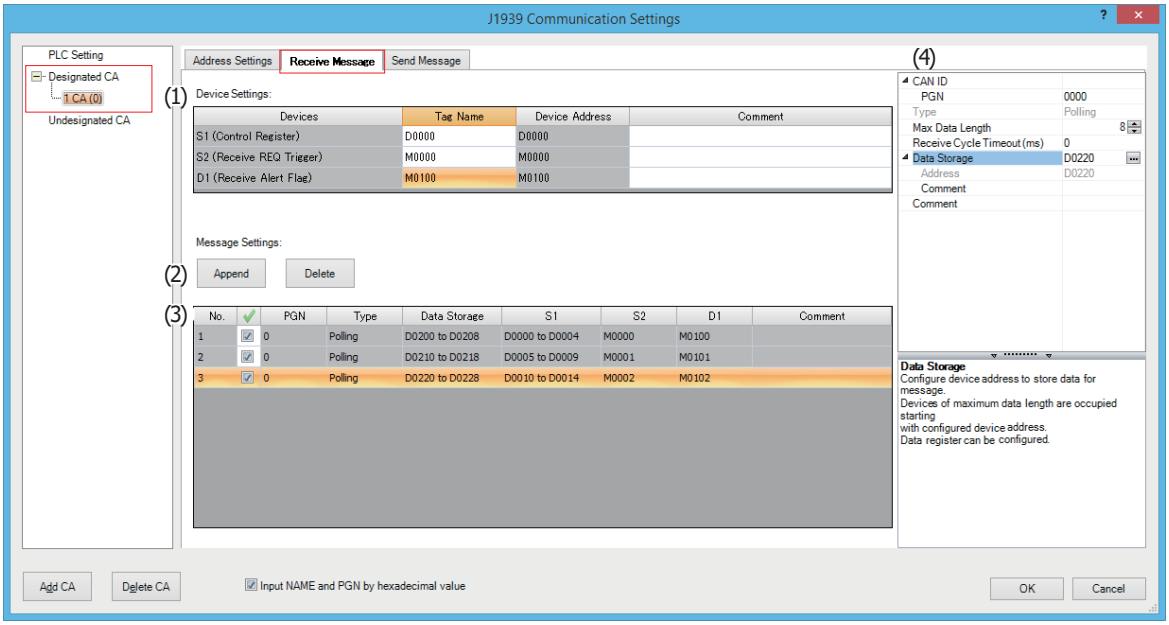
Displays each field of the device name (NAME) (2). When a **Preset Value** is changed, the result is reflected in the device name (NAME).

(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 Type 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.

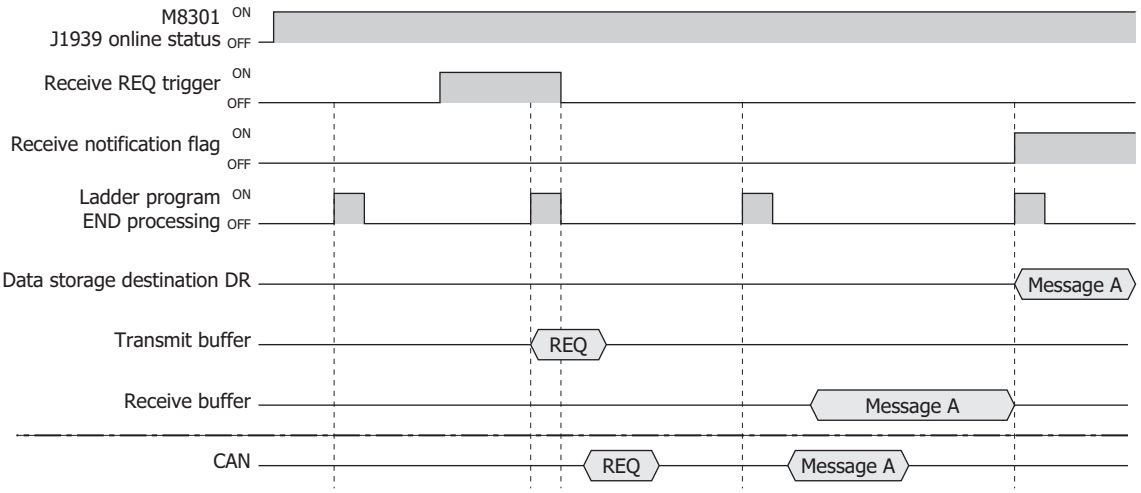
Storage Destination	Item		Description
Starting number+0	Received data length		This register stores the length of the received message data. If the message that will be received exceeds 8 bytes, a multi-packet message split into 8-byte packets will be received. After the multi-packet message has finished 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 Notification Flag)
Sets the flag for notification of a received message. This register is on for one scan when a message is received.

(2) Append button/Delete button
Click **Append** 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.

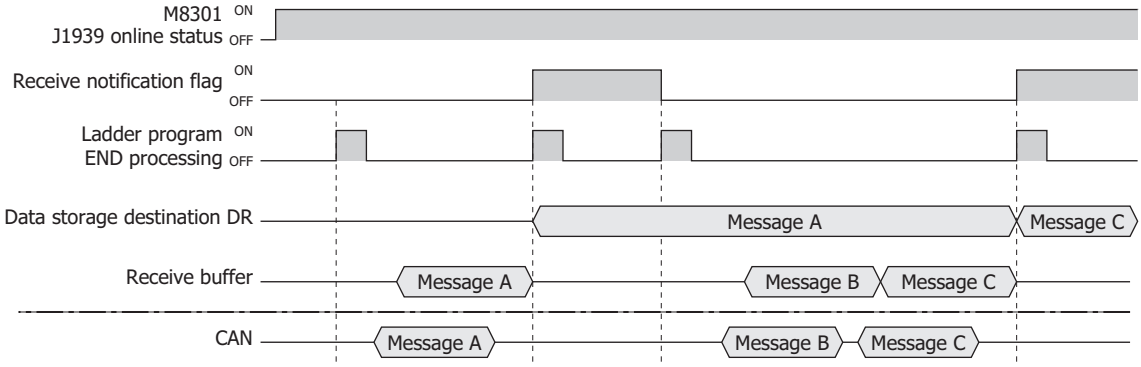
CAN ID

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



Maximum 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 destination
Sets the data registers that will store the received message. The maximum data length is used starting from the set device.

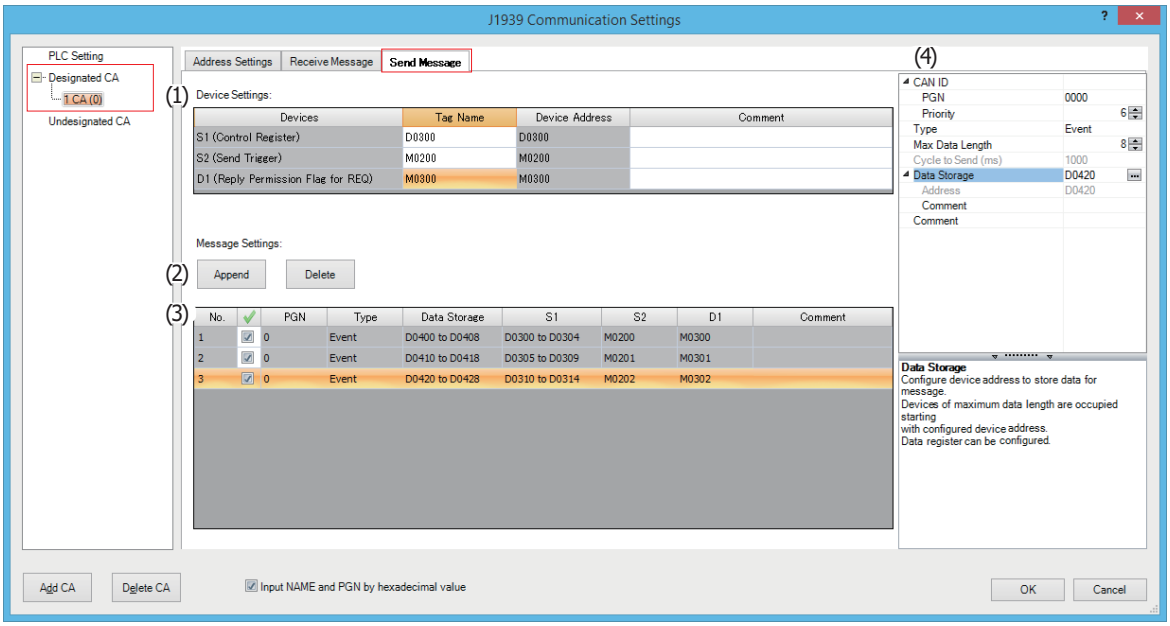
Address: Shows the device address that corresponds to the tag name of the device or the device address.

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 Type 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**		

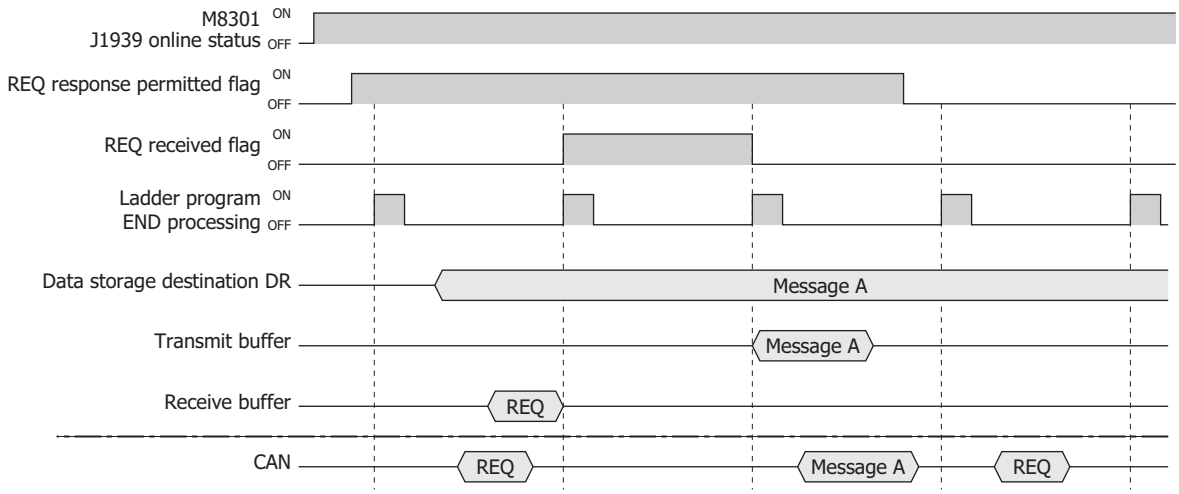
Transmit results status

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 (Transmit 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 (REQ Response Permitted Flag)
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) Append button/Delete button
Click **Append** 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.

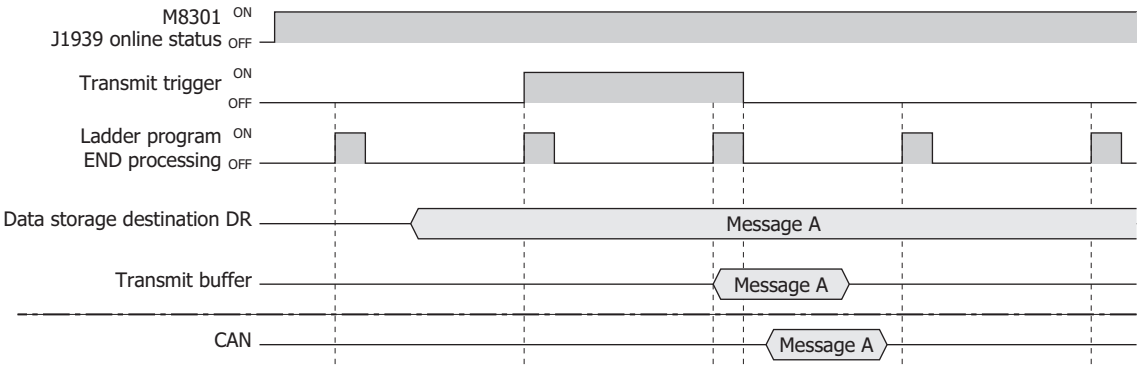
Priority: Sets the priority for the message to transmit between 0 and 7.

Type

Sets the transmit method for the message. Select as "Event" or "Cycle".

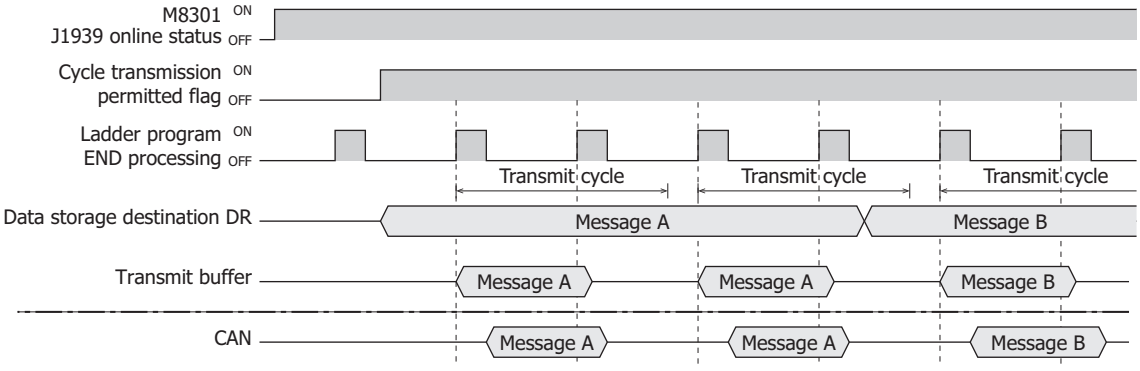
Event: When the transmit trigger is turned on, the transmit data stored in the data storage destination is transmitted in END Processing.

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



Maximum data length

Sets the maximum data length of the PGN to transmit. The maximum value that can be set is 252 bytes.

Transmit cycle (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 destination
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.

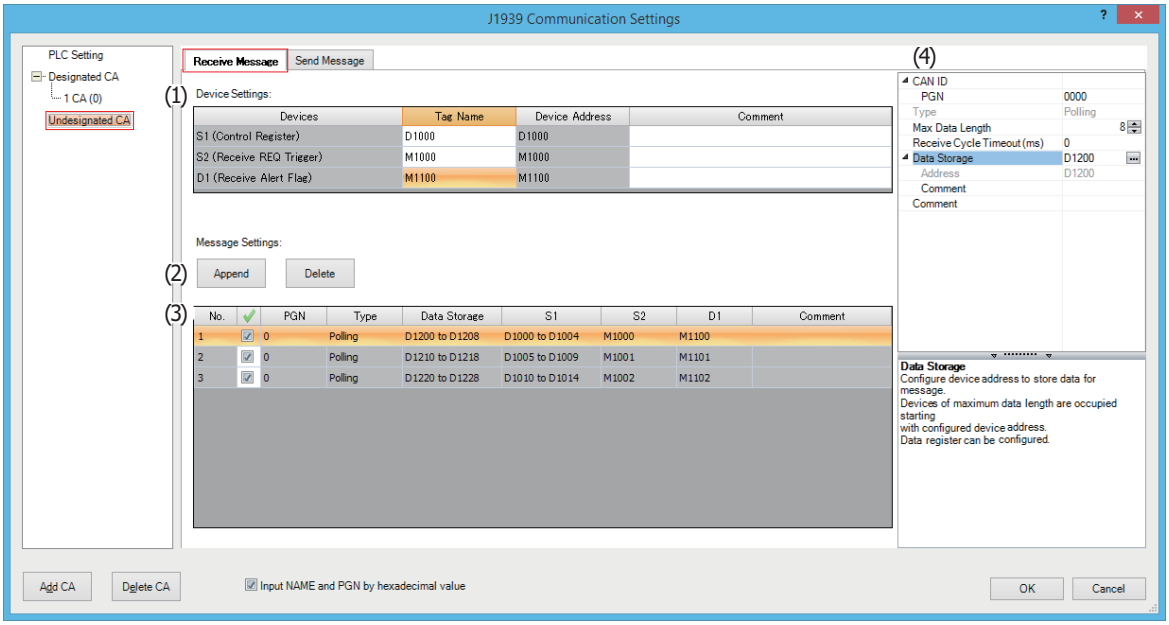
Comment
Sets the comment for the transmitted message. The comment that can be set is a maximum of 127 bytes.

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.

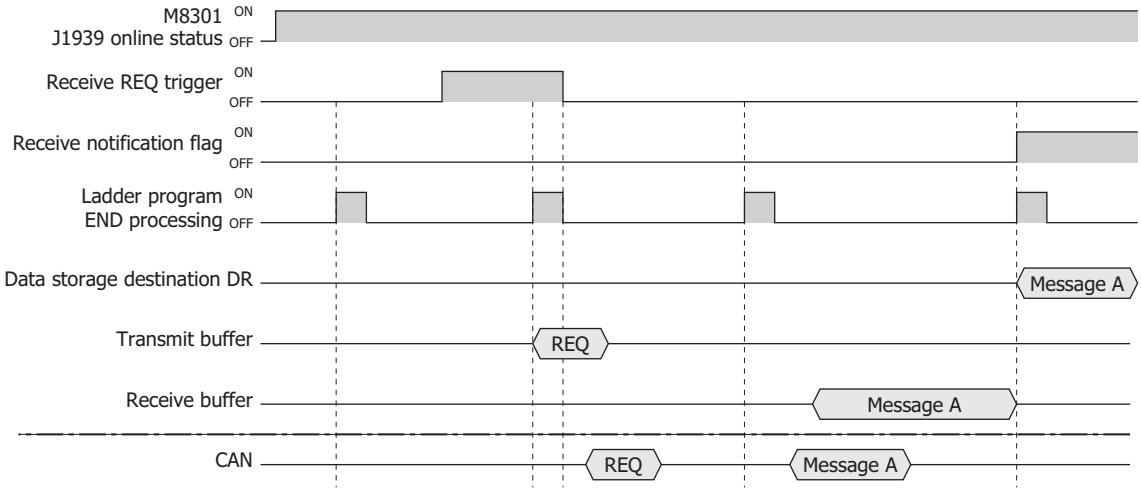
Storage Destination	Item		Description
Starting number+0	Received data length		This register stores the length of the received message data. If the message that will be received exceeds 8 bytes, a multi-packet message split into 8-byte packets will be received. After the multi-packet message has finished 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 8-16.
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 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 Notification Flag)
Sets the flag for notification of a received message. This register is on for one scan when a message is received.

(2) Append button/Delete button
Click **Append** 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

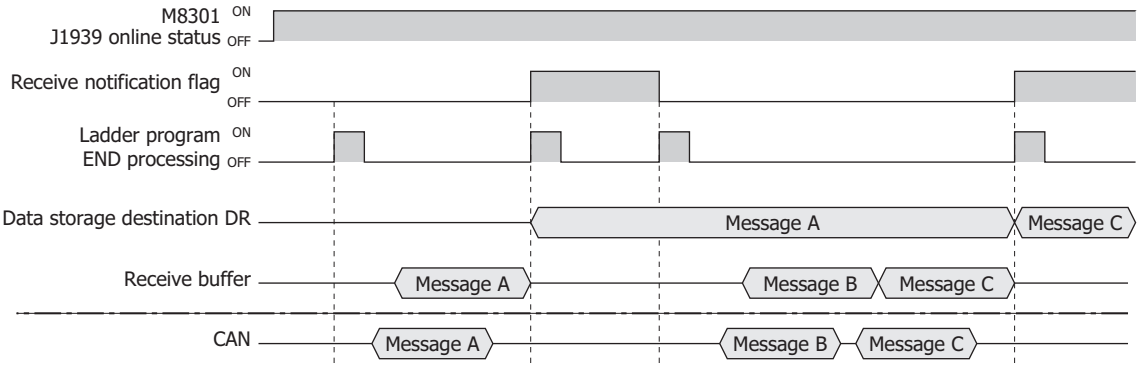
PGN: Sets the PGN of the message to receive in 2 bytes. The range of the PGN that can be set is 0 to 61,184 (0000h to EF00h) (PDU1 format) and 61,440 to 65,535 (F000h to FFFFh) (PDU2 format). Set the low order 1 byte (PDU Specific) that indicates the destination address in PDU1 format to 00h.

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



Maximum 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 destination
Sets the data registers that will store the received message. The maximum data length is used starting from the set device.

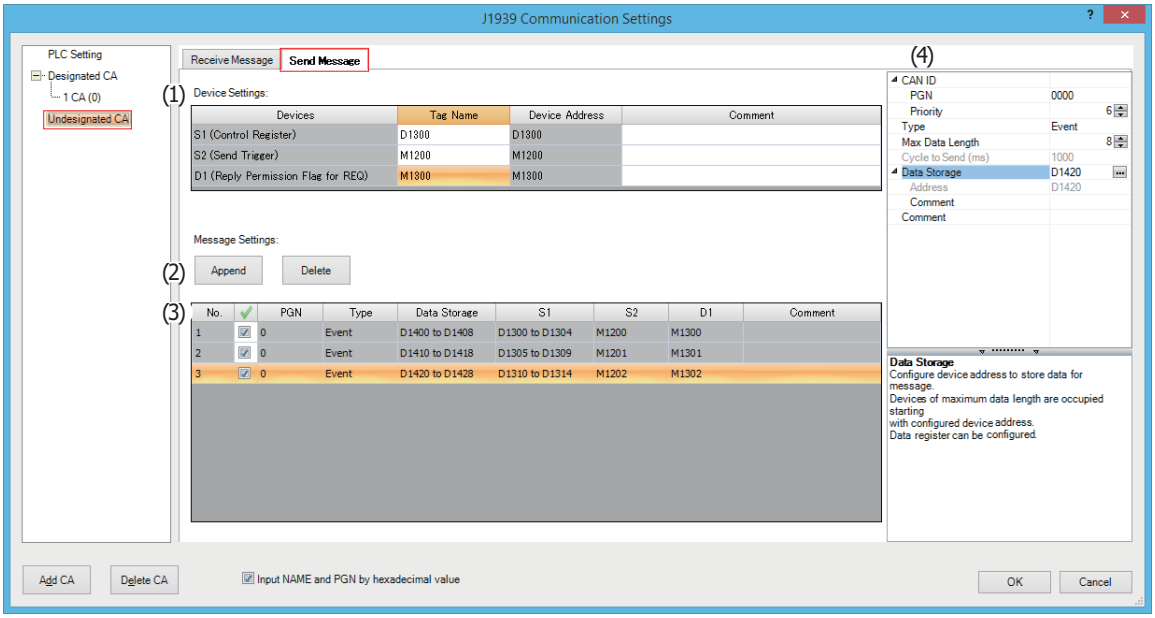
Address: Shows the device address that corresponds to the tag name of the device or the device address.

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 Type 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 8-19.
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**		

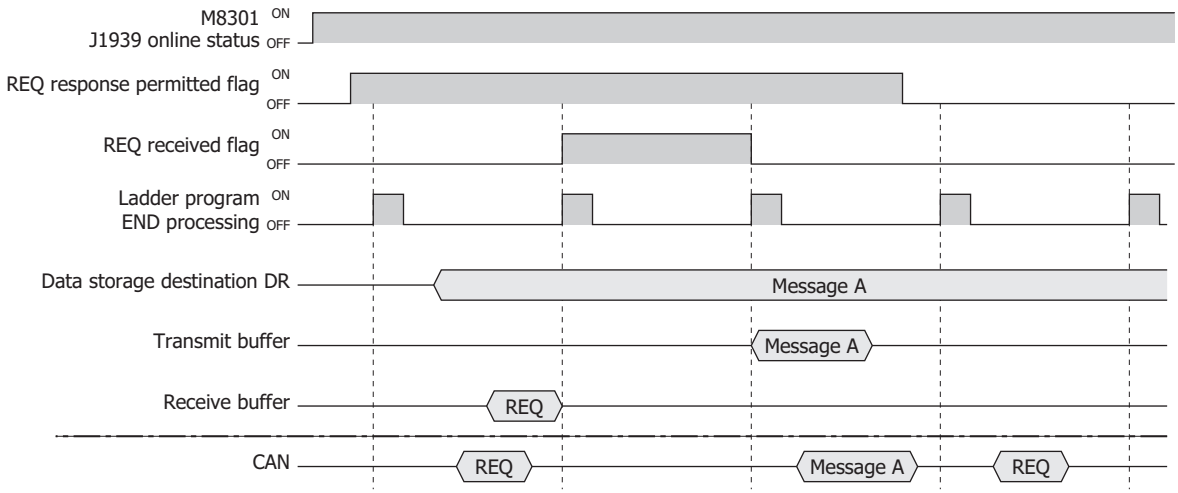
Transmit results status

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 (Transmit 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 (REQ Response Permitted Flag)
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) Append button/Delete button
Click **Append** 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.

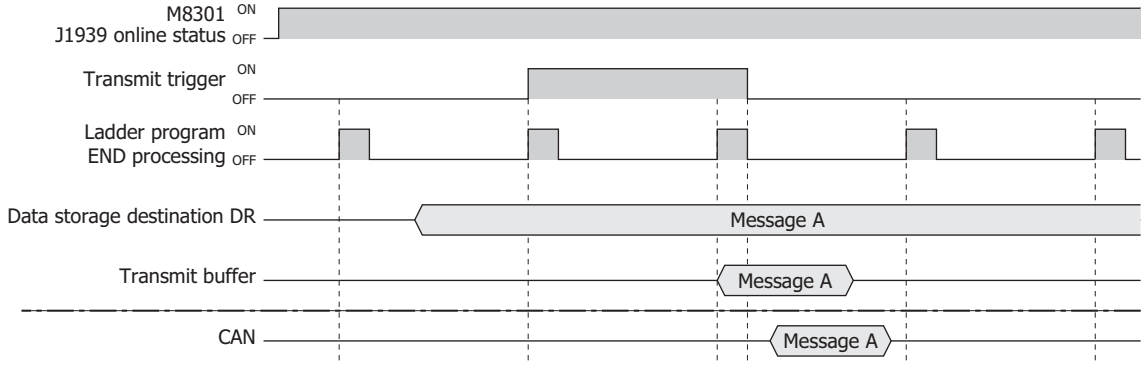
Priority: Sets the priority for the message to transmit between 0 and 7.

Type

Sets the transmit method for the message. Select as "Event" or "Cycle".

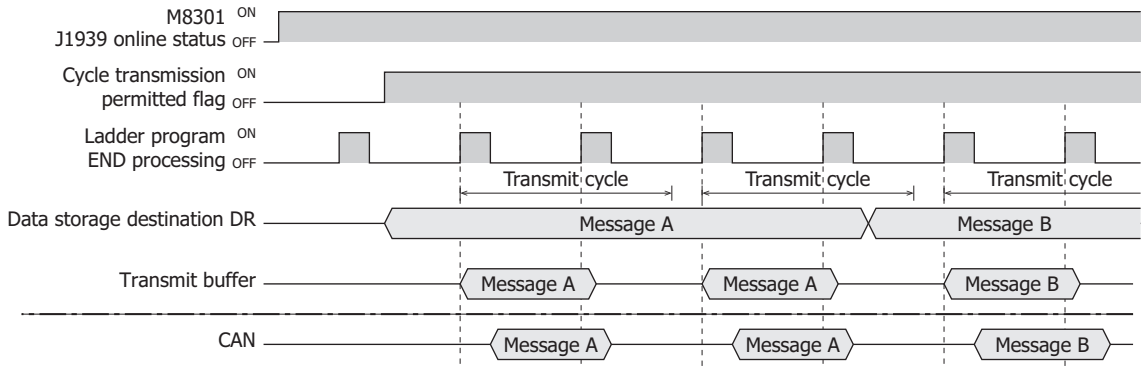
Event: When the transmit trigger is turned on, the transmit data stored in the data storage destination is transmitted in END Processing.

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



Maximum data length

Sets the maximum data length of the PGN to transmit. The maximum value that can be set is 252 bytes.

Transmit cycle (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 destination
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.

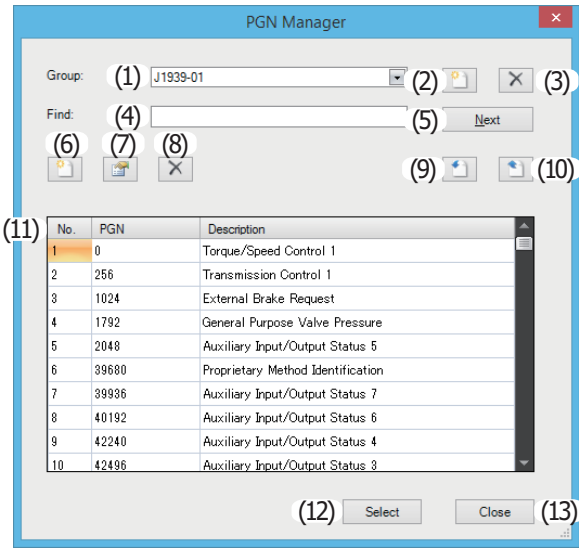
Address: Shows the device address that corresponds to the tag name of the device or the device address.

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.

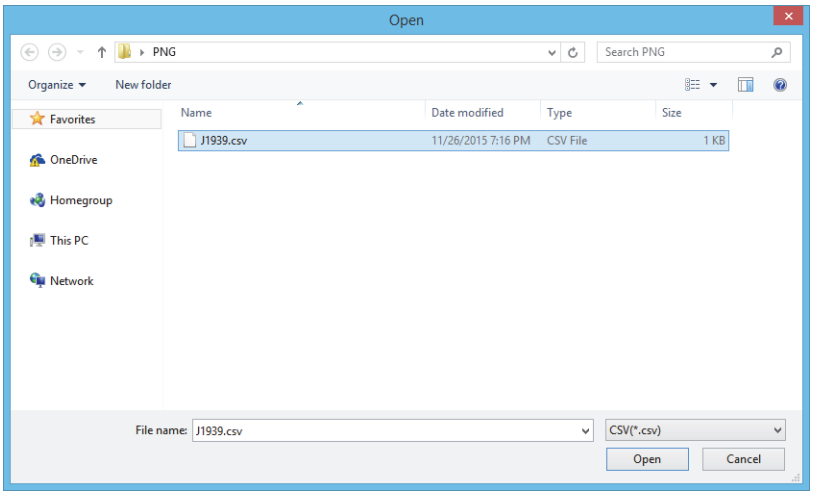
Select a registered group in the group list that contains the PGN to delete, and then select the PGN to delete in the PGN list. Click this button to delete the PGN.

(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.

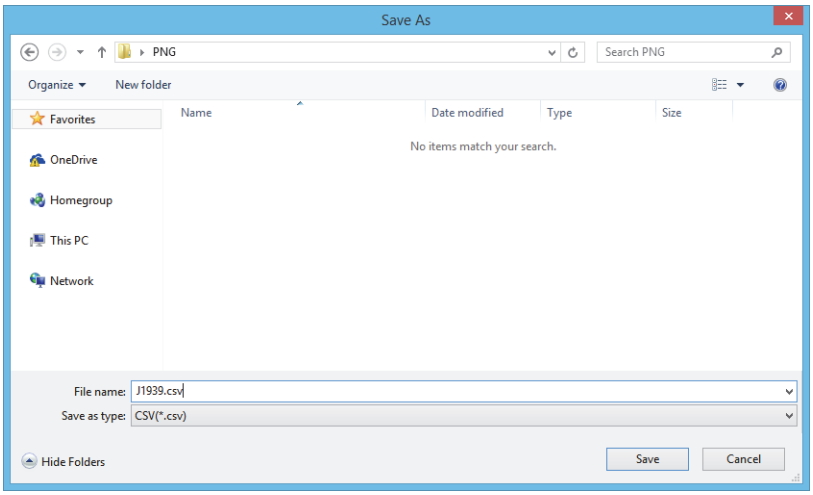


Select the CSV file where the PGNs are saved and click **Open** to import the PGNs in that file to the PGN list for the group selected in the group list.

(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

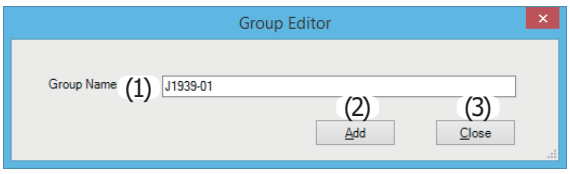
Closes PGN Manager and sets **CAN ID** in the calling message parameter to the PGN selected in the PGN list.

(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

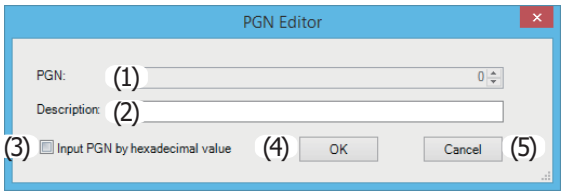
Closes PGN Group Editor and adds the group to the PGN Manager group list.

(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: 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.

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 LAD 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 3-16.

Valid Devices

	Device	Function	I	Q	M	R	T	C	D	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 3-10.

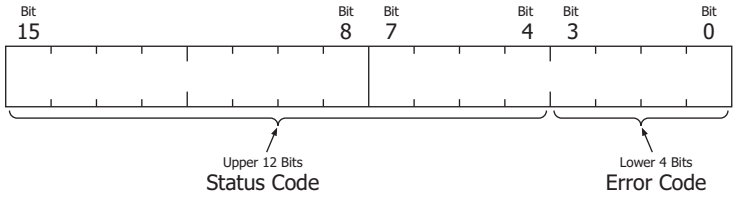
D2 (Destination 2)

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 D0 to D1998, D2000 to D7998, and D10000 to D49998 can be specified.

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.



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

10: SEND E-MAIL FUNCTION

Introduction

This chapter describes the EMAIL instruction that sends preregistered e-mails.

Overview

The HMI-Ethernet port of the HMI module supports 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 to the EMAIL instruction is on, the EMAIL instruction sends an e-mail specified by S1. 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 LAD Programming Manual".

In order to use the EMAIL instruction, you must configure the settings such as e-mail server name on **E-mail Settings** in **HMI Module Configuration** and create the FC6A Series MicroSmart e-mails in advance.

Valid Devices

	Device	Function	I	Q	M	R	T	C	D	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

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.

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.

Status Code (Decimal)	Operation Transition State	Description
16	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	Waiting for response	From sending the e-mail to the sending e-mail server, until the response is received.
64	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	Normal	ON (The status code is 64)
1	HMI module connection error • The HMI module is not connected • The HMI module is not operating correctly	ON
2	Timeout error • Port number may be incorrect.	ON
3	• 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	The specified e-mail is not configured.	ON
5	Multiple EMAIL instructions are executed simultaneously.	ON
6	Parameter error	ON
7	Other errors *1	ON
8	Converting a data register value failed *2	ON

*1 The detail of the error code 7 is stored in special data register D8457.

*2 When converting a data register value fails, ---- is inserted in the e-mail body instead of data register value.

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

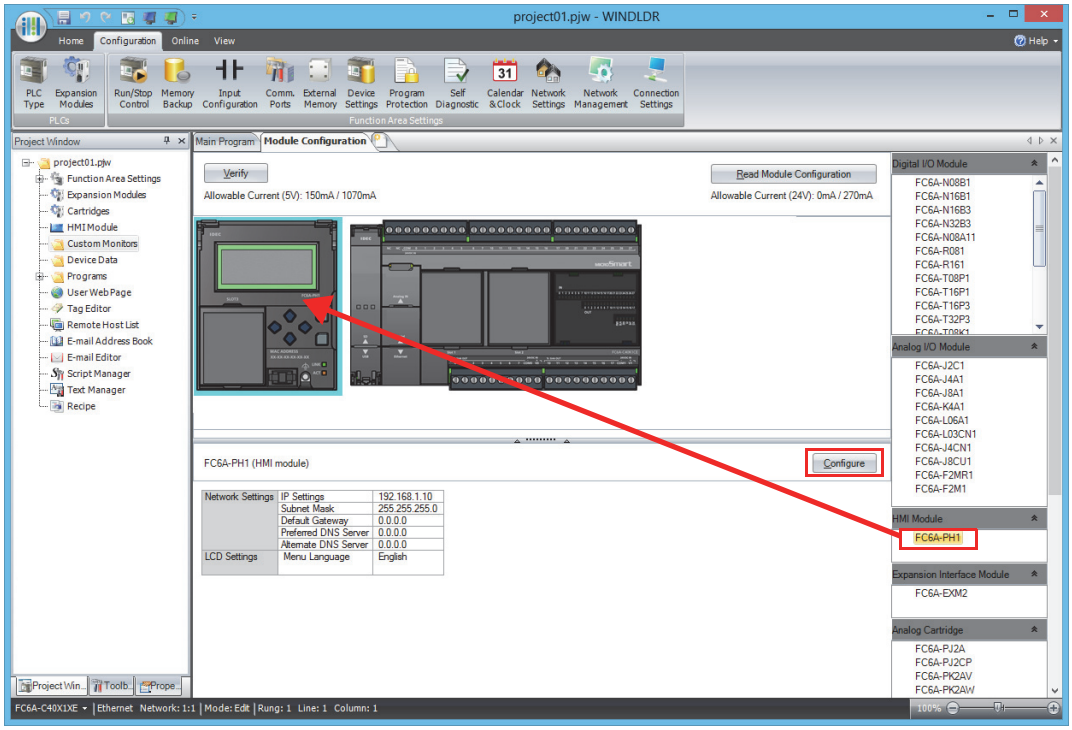
When the error code stored in D2 of EMAIL instruction is 7, the response from sending e-mail server can be confirmed with special data register D8457. The error code that the sending e-mail servers return could vary with each sending e-mail server. For details about the error code, contact the administrator of the sending e-mail server.

Error Code	Description	Possible Cause
451	The requested action is canceled	Sending e-mail server is not working normally
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

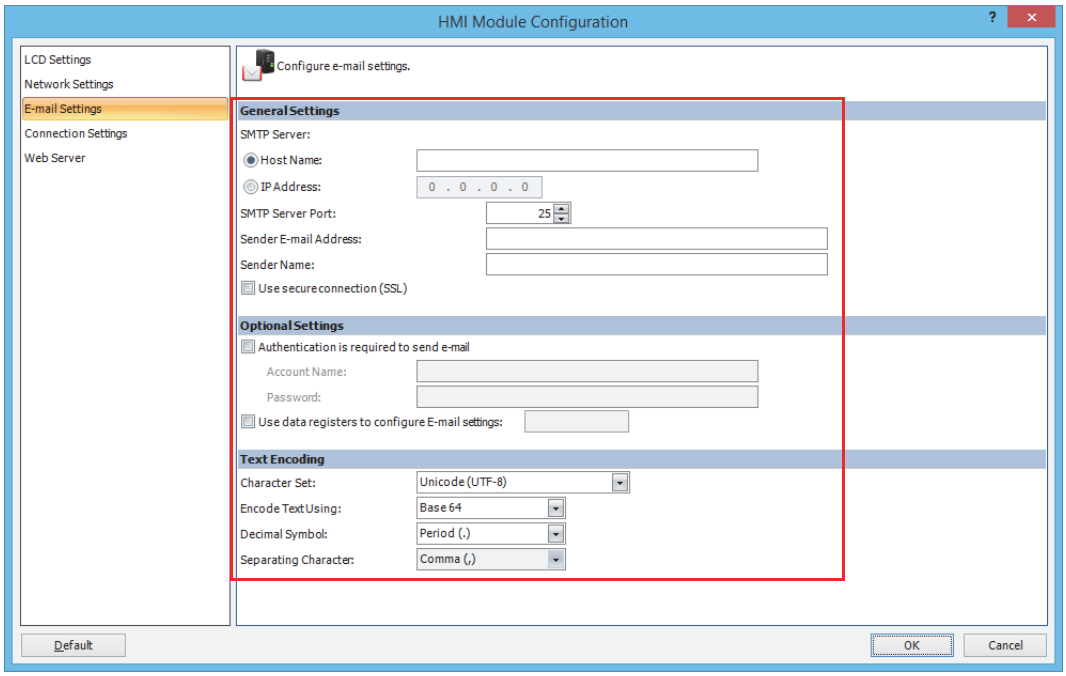
Programming WindLDR

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.



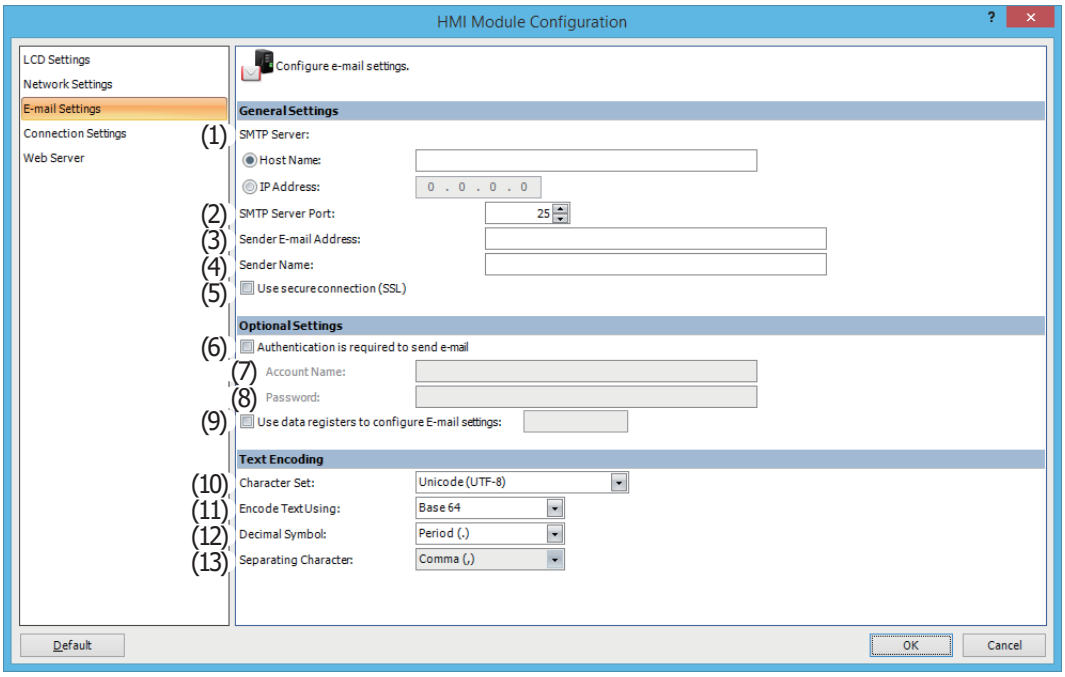
3. Click **E-mail**.
4. Configure the settings in General Settings, Optional Settings, and Text Encoding.



5. Download the user program to the FC6A Series MicroSmart.

This concludes configuring the e-mail settings.

■ 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)
When SSL 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)	Decimal value	1	Starting data register+107		When SSL communication is required with the sending e-mail server that is used, specify 1. When this 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) : 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)	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 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.

Base64: 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**.

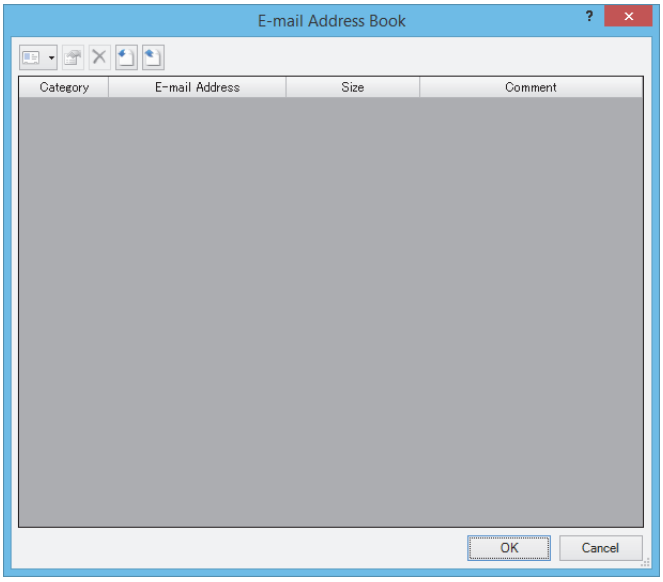
Decimal Symbol	Separator Character
Period '.' (2Eh)	Comma ',' (2Ch)
Comma ',' (2Ch)	Semicolon ';' (3Eh)

E-mail Address Book

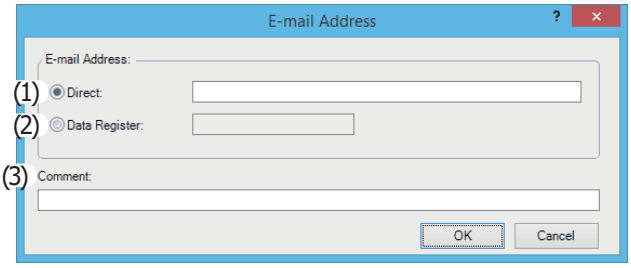
The e-mail addresses and e-mail address groups can be configured in E-mail Address Book dialog box. E-mail addresses can be grouped into an e-mail address group. The e-mail address group can be used to specify a group of e-mail addresses in each e-mail.

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

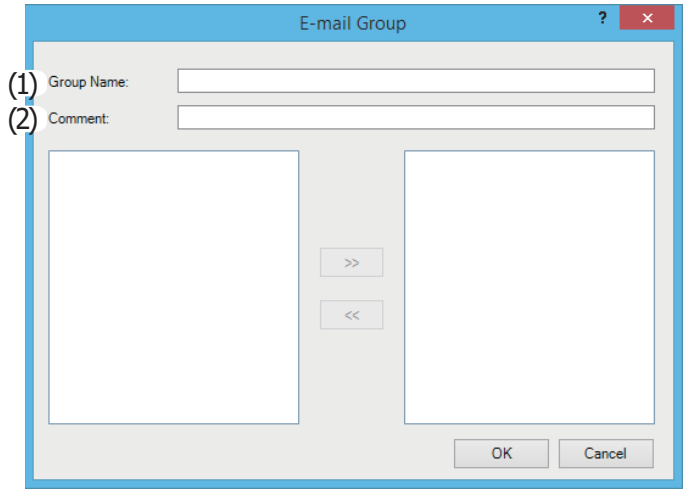
Notes:

- 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.

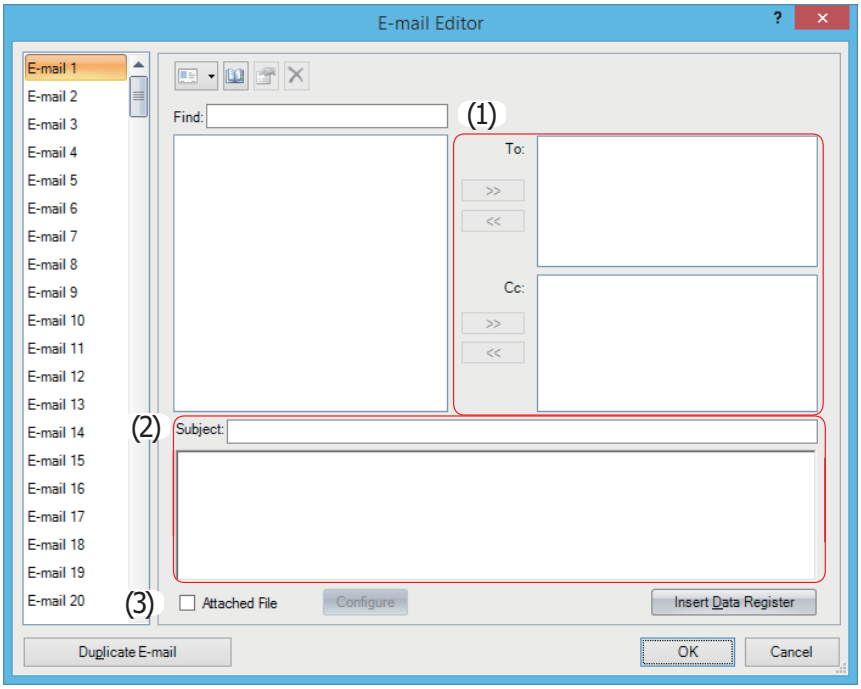
The e-mail addresses that have not been added to the e-mail address group are shown in the list box on the left. The e-mails addresses are listed in the order that they were registered in the E-mail Address Book dialog box. In order to add an e-mail address to the e-mail address group, select the e-mail 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 and Cc)

E-mail address or e-mail group can be specified as the recipients. The maximum size of texts for **To** or **Cc** 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** and **CC** 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.

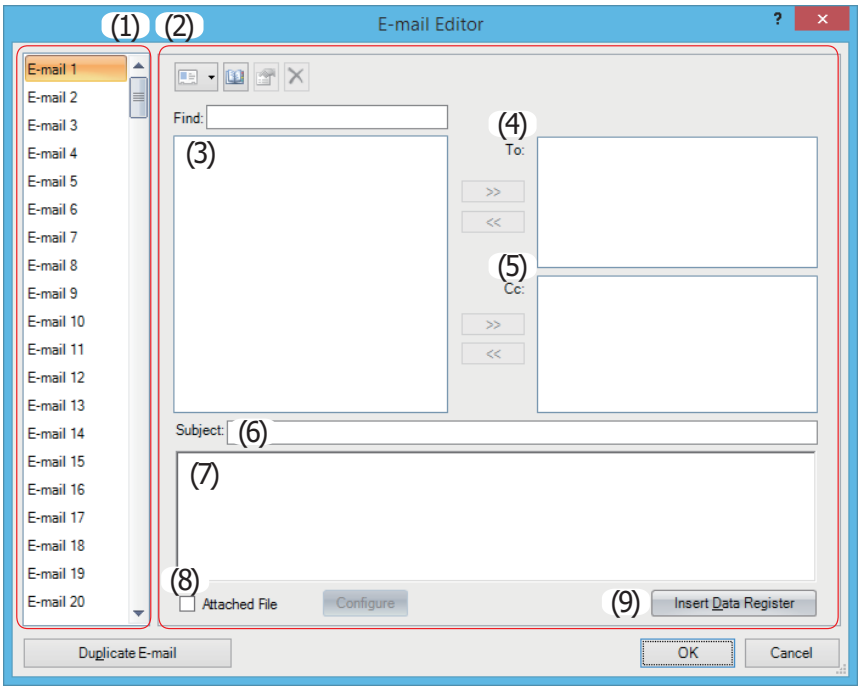
- Strings that use **character set** under **E-mail settings** in the **Function Area settings**
- Numeric value strings of the values of data registers that have been embedded in the body by inserting data registers
- Spaces, separator characters, and newlines

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.

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 five elements: **To**, **Cc**, **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** or **Cc** and click **>>** on the appropriate side to add that e-mail address to **To** or **Cc**.
Of the e-mail addresses in the **To** or **Cc** boxes, select the e-mail address to delete and click **<<** on the appropriate side to delete that e-mail address from **To** or **Cc**.

(4) To/(5) Cc
Specifies the addresses of the e-mail to send.

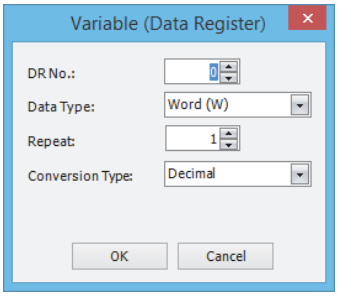
(6) Subject
Specifies the subject of the e-mail to send.

(7) Body
Specifies the e-mail body.
The e-mail body can be written in multiple lines.

(8) Attached File
Select this check box to attach a file to the e-mail. The items in the attached file (CSV file) and its layout can be freely configured in a range that fits into the upper limit of the file size. The CSV file content can include text and data register values.
For editing the attached file, see "Attached File Editor" on page 10-13.

(9) 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.

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

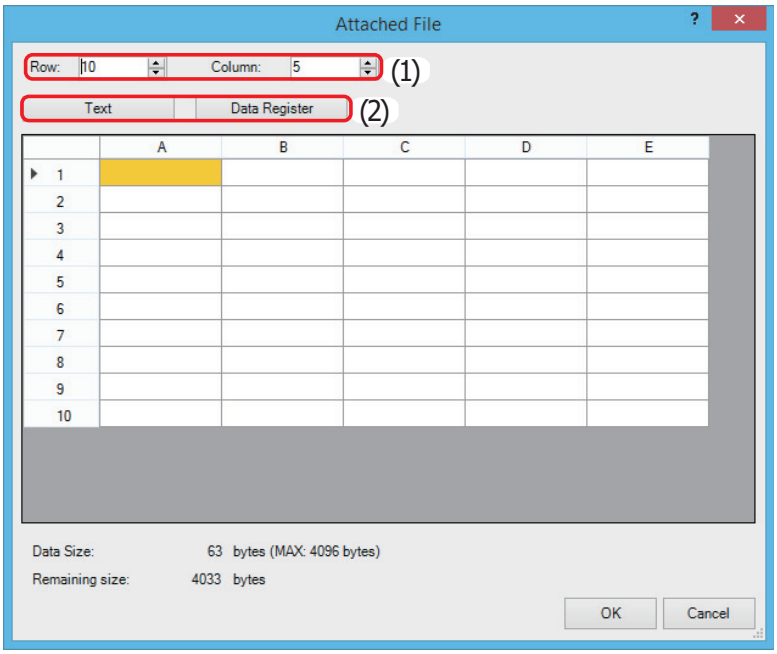
The text created in the E-mail Editor is encoded and sent according to the **Encoding method** settings specified on **E-mail Settings** in **Function Area Settings**.

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.

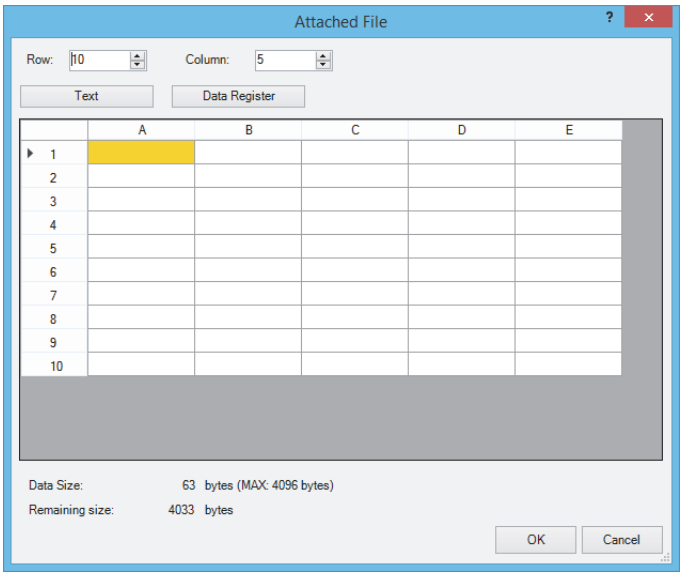
(2) Text, Data Register

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 10-3 in this chapter.

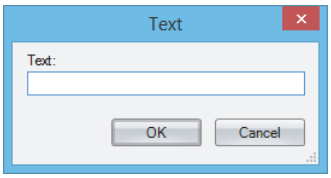
Programming WindLDR

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.



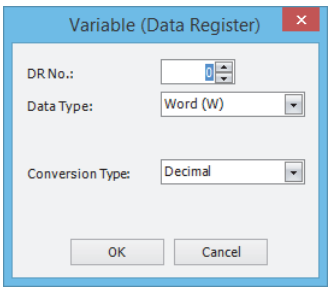
The content of the CSV file can be modified at the upper part of the dialog. The current file size and the remaining size that can be used for editing the content are displayed at the bottom of the dialog. You can expand the grid area by changing the size of the dialog. The file size includes separators 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**.

11: WEB SERVER

Introduction

This chapter describes the Web server functions in the FC6A Series MicroSmart.

Overview

The HMI-Ethernet port of the HMI module supports 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.

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.

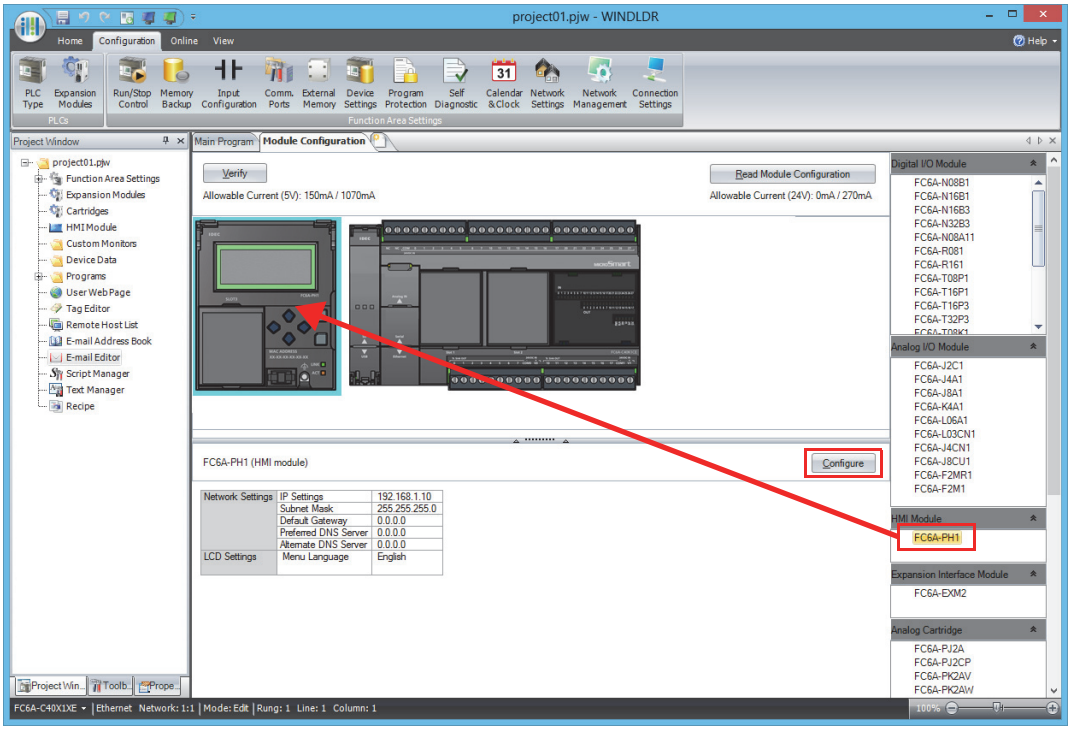
http://192.168.1.5/system/

Programming WindLDR

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**.

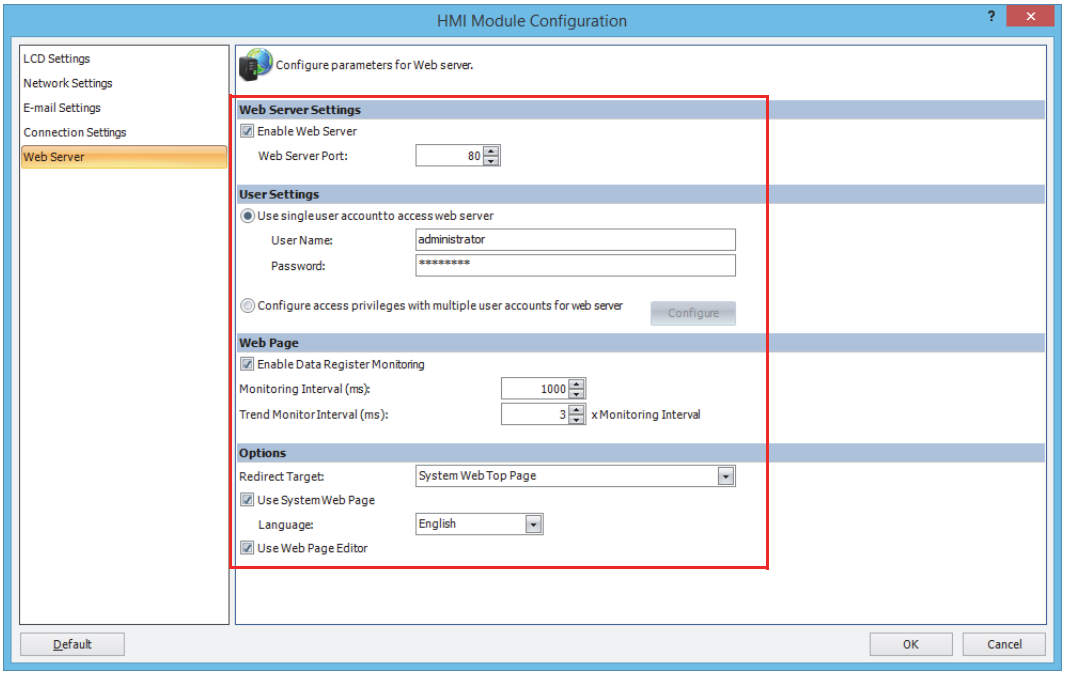
2. Click the inserted HMI module in the module configuration area and click **Configure**.
   The **HMI Module Configuration** dialog box is displayed.

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



3. Click **Web Server**.

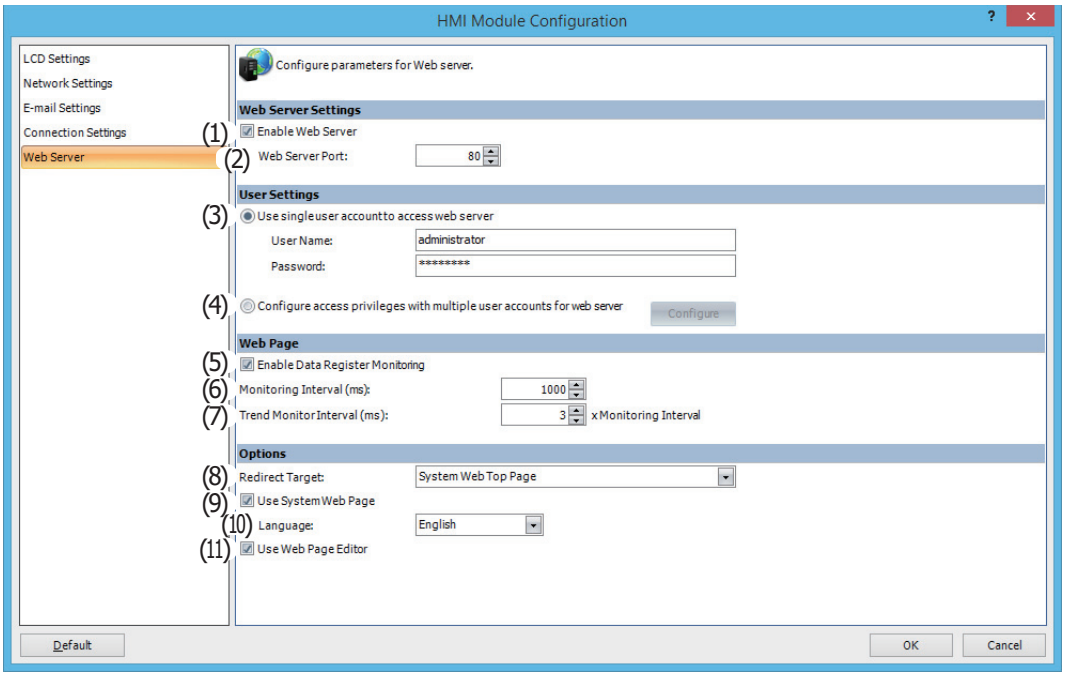
4. Configure the settings in Web Server Settings, Web Page, and Options.



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 11-4.

■ 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.

(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 11-26.

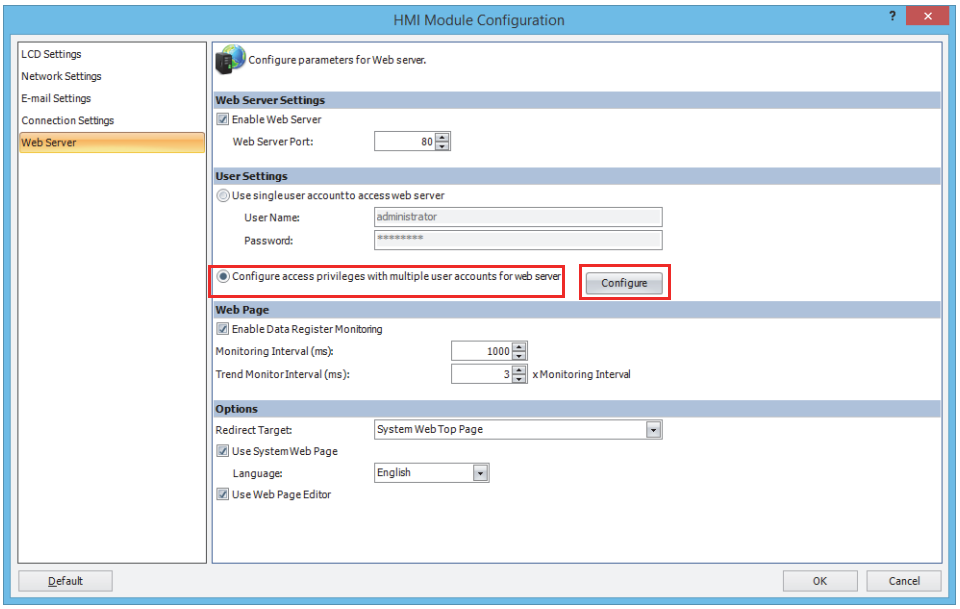
User Account Settings

■ Creating Multiple User Accounts

- **Operation procedure**

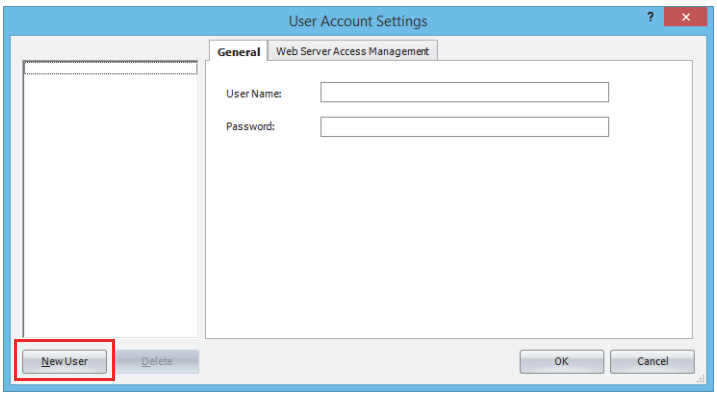
1. In the **HMI Module Configuration** dialog box, on the **Web Server** tab, select **Configure access privileges with multiple user accounts for web server** and click **Configure**.

   The **User Account Settings** dialog box is displayed.

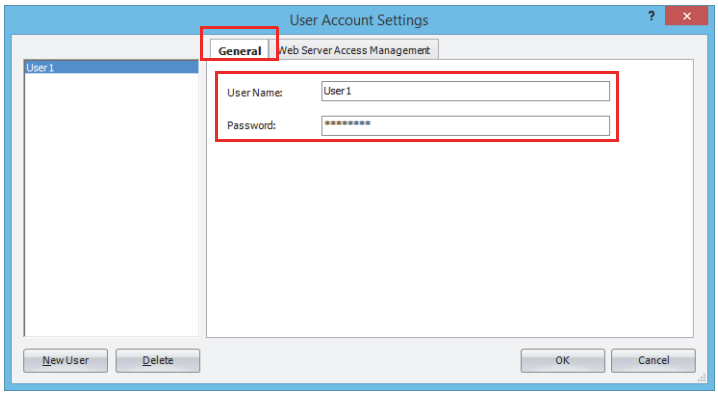


2. Click **New User**.

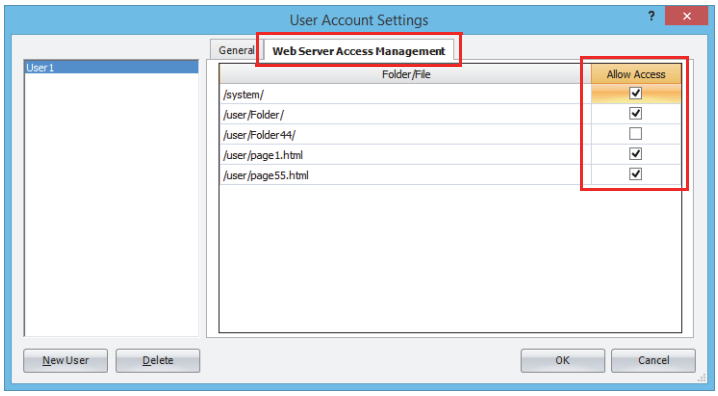
   A user account is created in the list.



3. On the **General** tab, set **User Name** and **Password**.



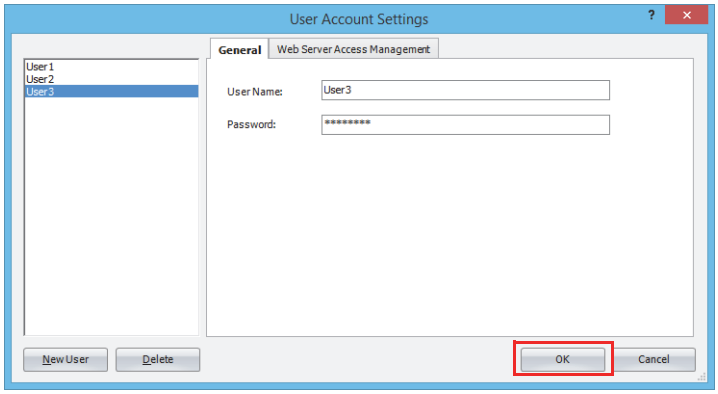
4. Click the **Web Server Access Management** tab and select the check boxes of folders and servers where access will be permitted.



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**.



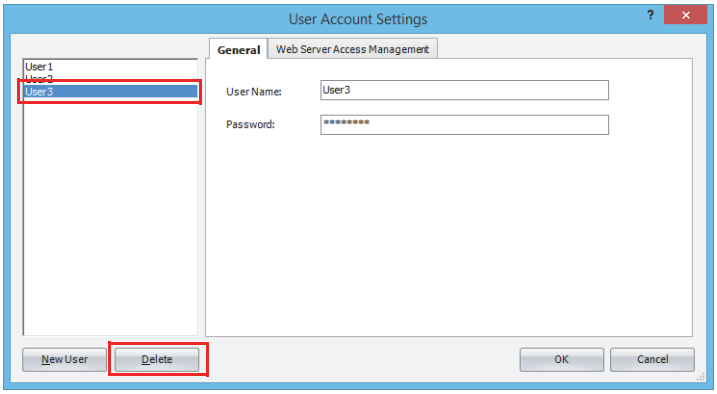
This concludes creating multiple user accounts.

■ Deleting a Created User Account

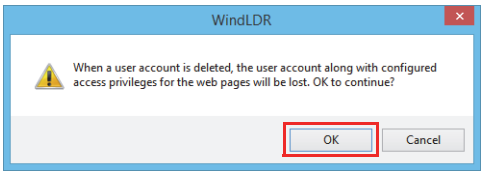
- **Operation procedure**

1. Select the user account on the list and click **Delete**.

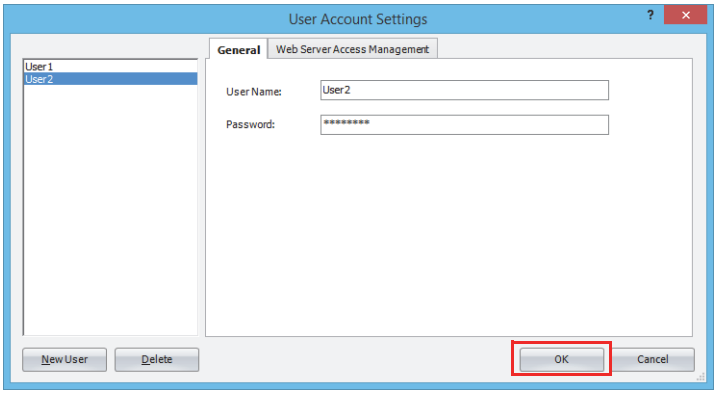
   A confirmation message is displayed.



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.
/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)

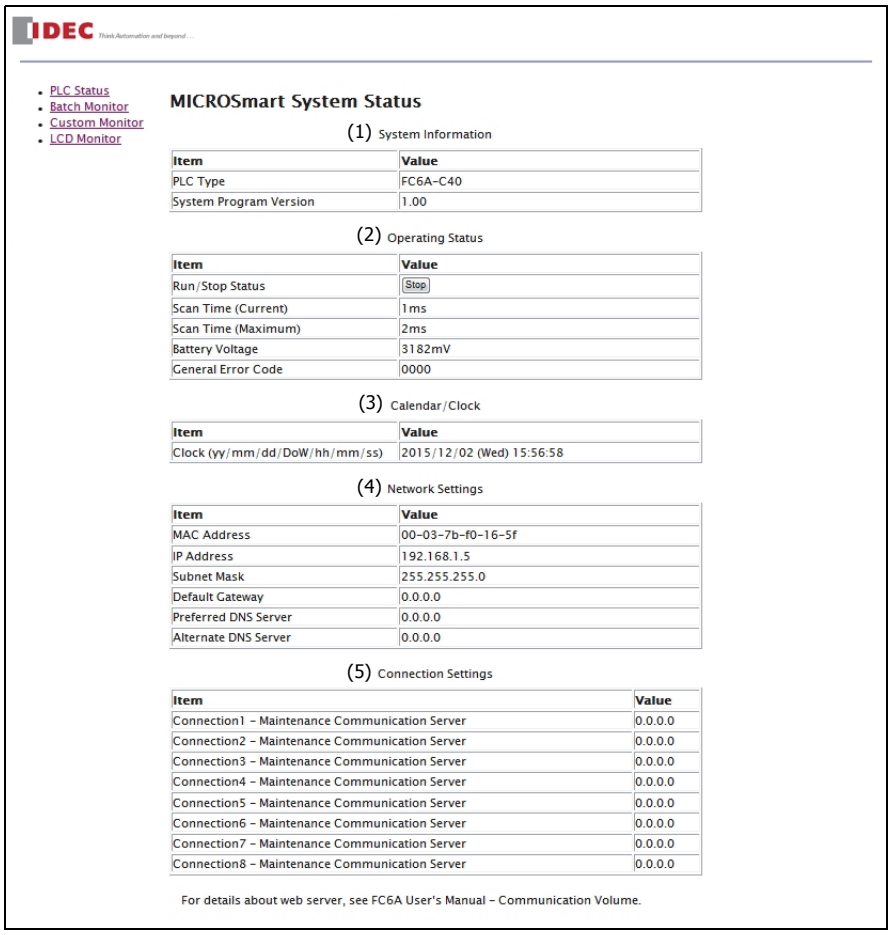
Note: BIN-B is used in the custom monitor when internal relay is monitored. 0000 is OFF and 0001 is ON.

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.



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

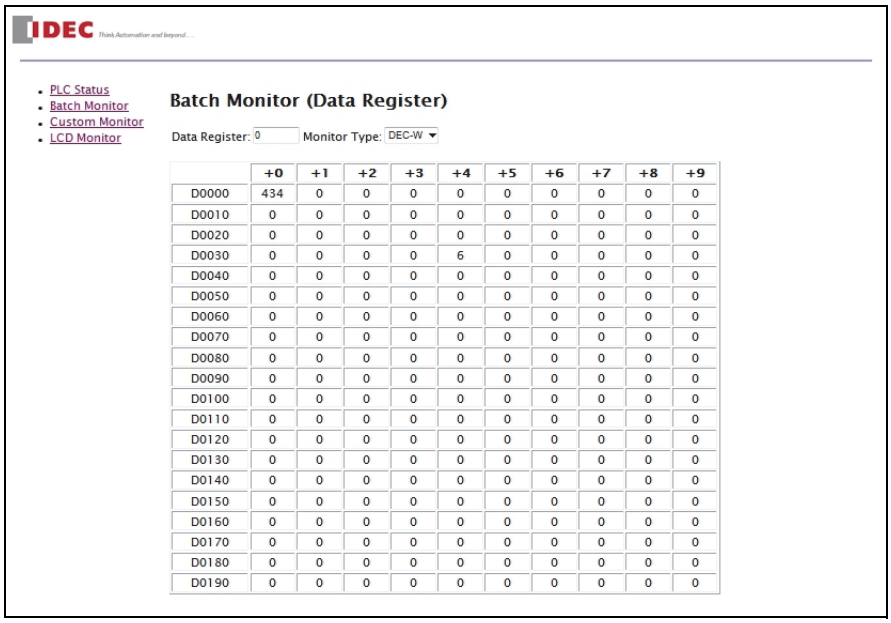
The network settings of the FC6A Series MicroSmart can be confirmed.

(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**.

■ Batch Monitor

In the batch monitor, 200 consecutive data registers can be monitored and controlled with the specified data type.



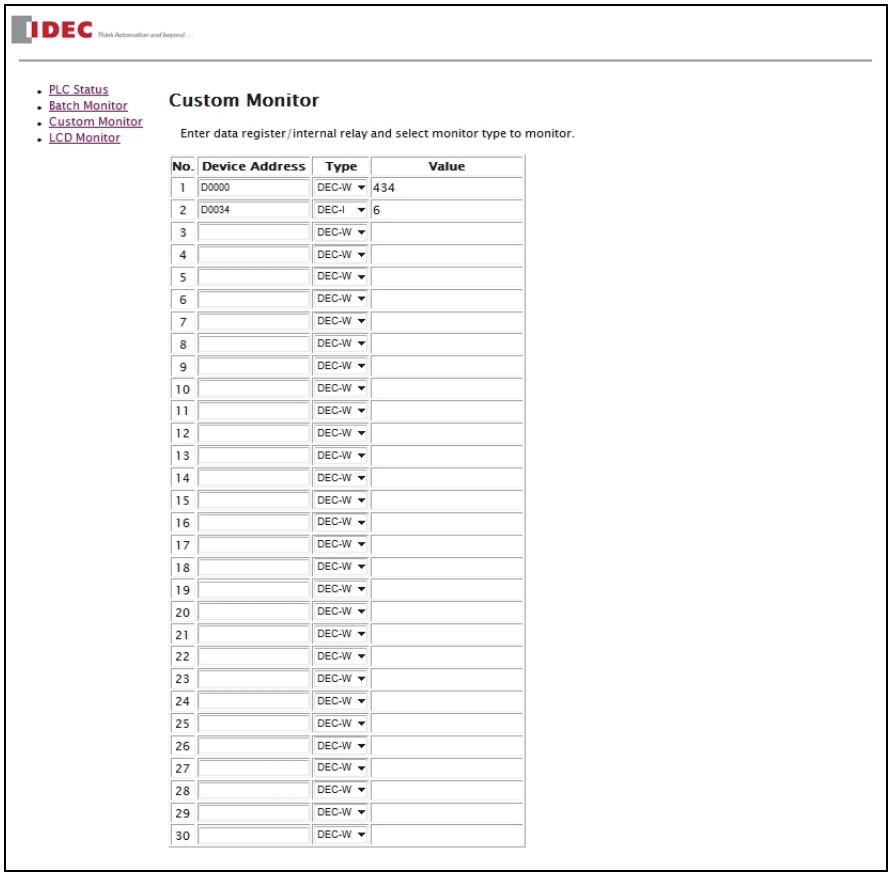
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D0000	434	0	0	0	0	0	0	0	0	0
D0010	0	0	0	0	0	0	0	0	0	0
D0020	0	0	0	0	0	0	0	0	0	0
D0030	0	0	0	0	6	0	0	0	0	0
D0040	0	0	0	0	0	0	0	0	0	0
D0050	0	0	0	0	0	0	0	0	0	0
D0060	0	0	0	0	0	0	0	0	0	0
D0070	0	0	0	0	0	0	0	0	0	0
D0080	0	0	0	0	0	0	0	0	0	0
D0090	0	0	0	0	0	0	0	0	0	0
D0100	0	0	0	0	0	0	0	0	0	0
D0110	0	0	0	0	0	0	0	0	0	0
D0120	0	0	0	0	0	0	0	0	0	0
D0130	0	0	0	0	0	0	0	0	0	0
D0140	0	0	0	0	0	0	0	0	0	0
D0150	0	0	0	0	0	0	0	0	0	0
D0160	0	0	0	0	0	0	0	0	0	0
D0170	0	0	0	0	0	0	0	0	0	0
D0180	0	0	0	0	0	0	0	0	0	0
D0190	0	0	0	0	0	0	0	0	0	0

- **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 0 to 7800, 8000 to 8300, and 10000 to 49800. If an invalid value is entered, the value is corrected automatically.
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.

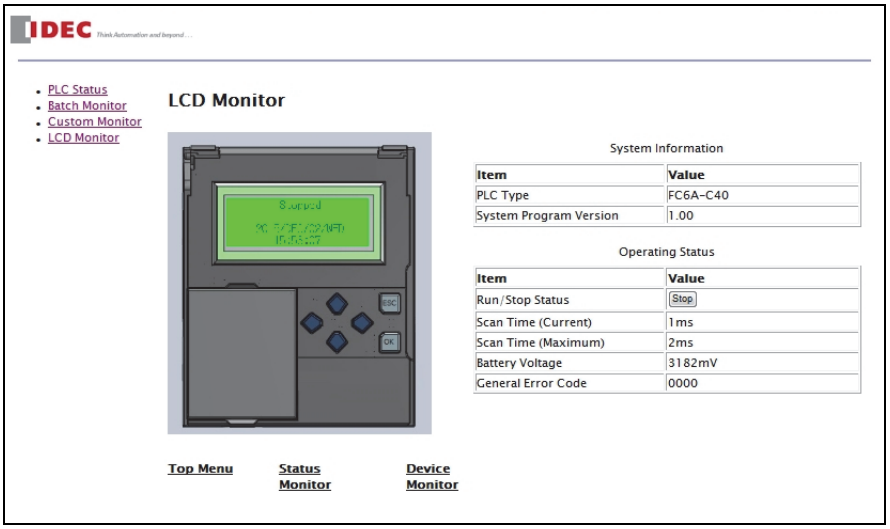


- **Operation procedure**

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.



- **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.

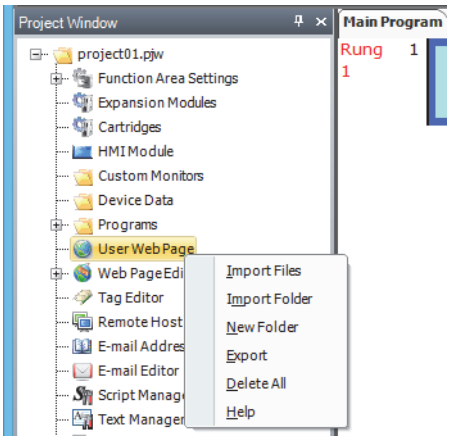
User Web Pages

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

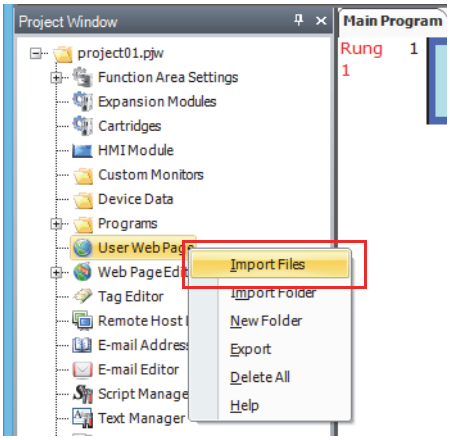
page1.html
Folder/page2.html

Before this procedure, create "page1.html" and "page2.html" that will be displayed on the FC6A Series MicroSmart.

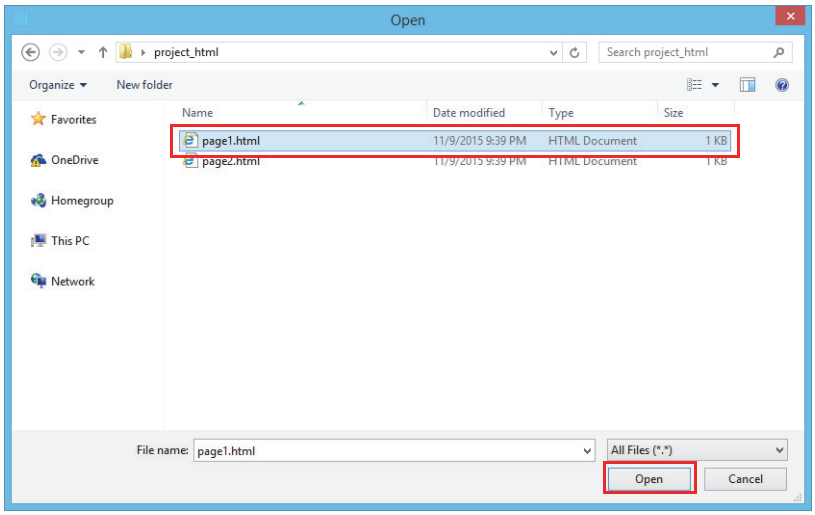
- **Operation procedure**

1. Right-click **User Web Page** and click **Import Files**.

   The **Open** dialog box is displayed.

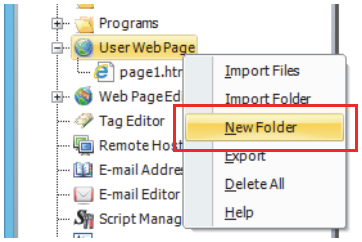


2. Select "page1.html" and click **Open**.



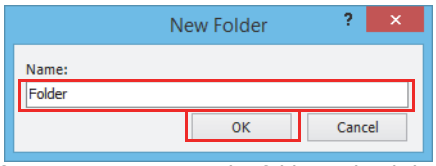
3. Right-click **User Web Page** and click **New Folder**.

The **New Folder** dialog box is displayed.



4. Enter "Folder" as the name of the folder to create and click **OK**.

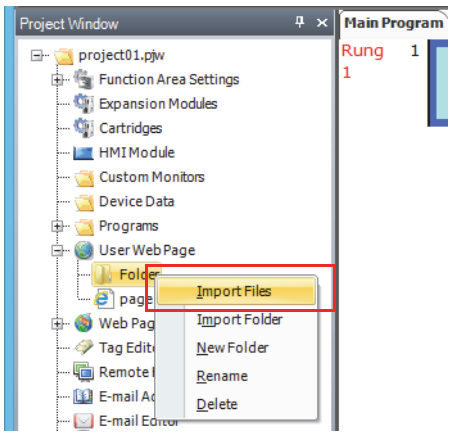
The folder titled "Folder" is created in **User Web Page**.



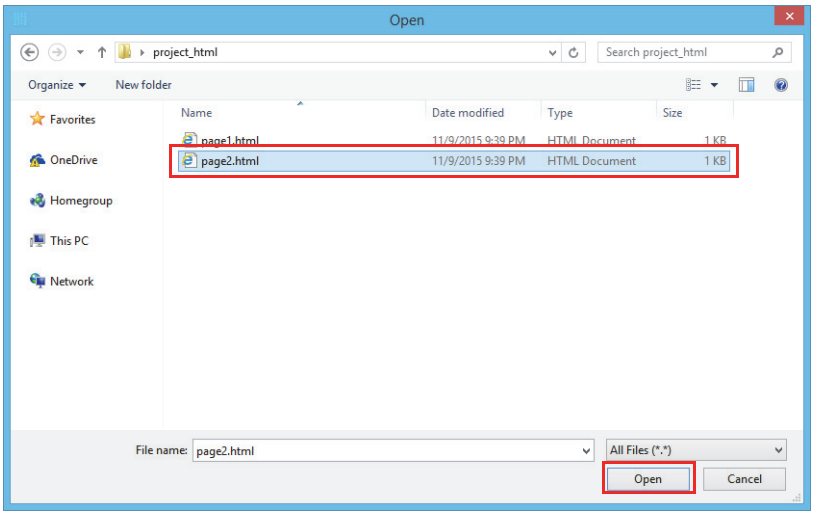
If you want to rename the folder, right click the folder and click **Rename Folder** to change the folder name.

5. Right-click **Folder** created in step 4 and click **Import Files**.

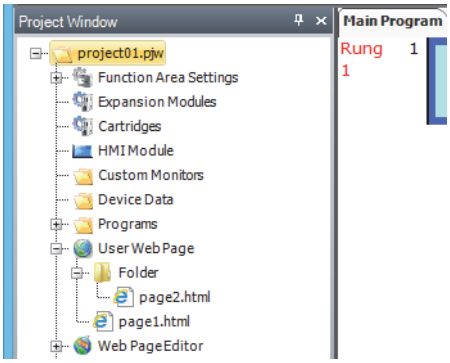
The **Open** dialog box is displayed.



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

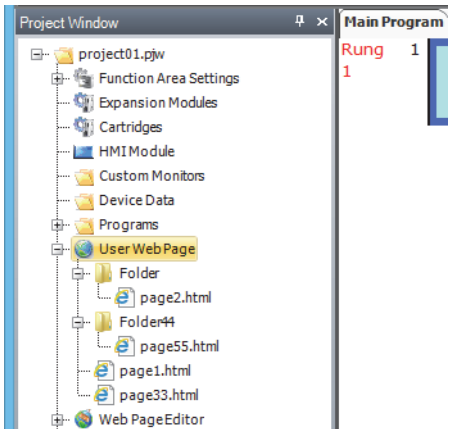
You can rebuild the user web page tree or change files by exporting the user web page data.

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.

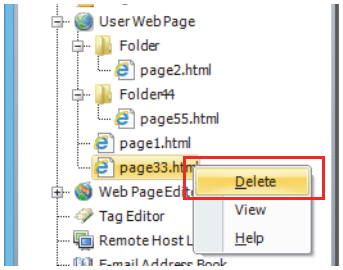
The following procedure describes an example of deleting "page33.html" and "Folder44" from the web page tree.



- **Operation procedure**

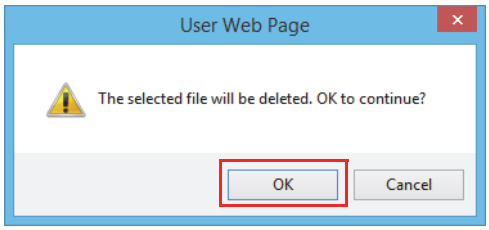
1. Right-click "page33.html" and click **Delete**.

A delete confirmation message is displayed.



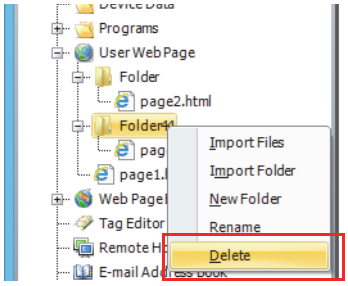
2. Click **OK**.

"Page33.html" is deleted.



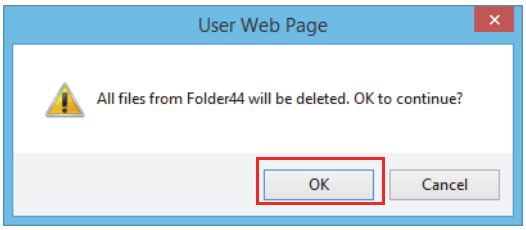
3. Right-click "Folder44" and click **Delete**.

A delete confirmation message is displayed.



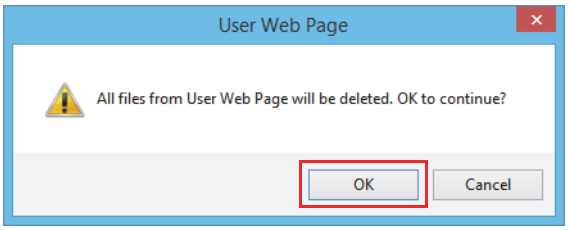
4. Click **OK**.

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 **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_EMAILADDR 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.

{{W,D,2058,0,0,HEX-D}}

To replace the string and emphasize the value, code the metacharacter format as follows.

{{W,D,2058,0,0,HEX-D}}

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 ",".

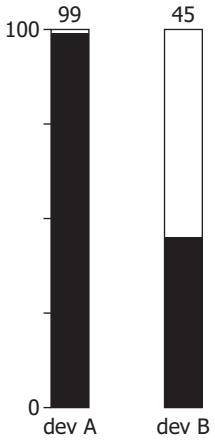
The parameters that can be set as an element of bars[] are as follows. If the default value of the parameter is acceptable, it does not need to be set.

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.

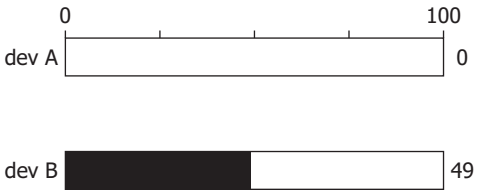
device	:	The device. Set this to "D" (data register).
address	:	The device number. Set this as a numeric value.
driver	:	The driver number. Set this to 0.
net_no	:	The network number. Set this to 0.
format	:	The web data type string.
width	:	The width of the div border in pixels. The default value is 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.

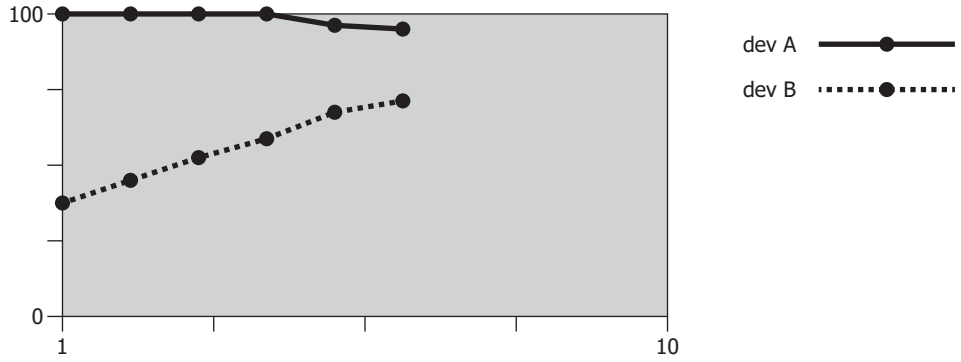
device	:	The device. Set this to "D" (data register).
address	:	The device number. Set this as a numeric value.
driver	:	The driver number. Set this to 0.
net_no	:	The network number. Set this to 0.
format	:	The web data type string.
width	:	The width of the div border in pixels. The default value is 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 and 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.

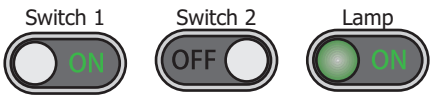
Note: Web Page Editor can be used with other web browsers that are not recommended, but problems may occur, such as with the display of images.

Creating User Web Pages

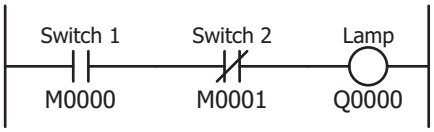
This section describes the procedure to create user web pages in Web Page Editor.

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:

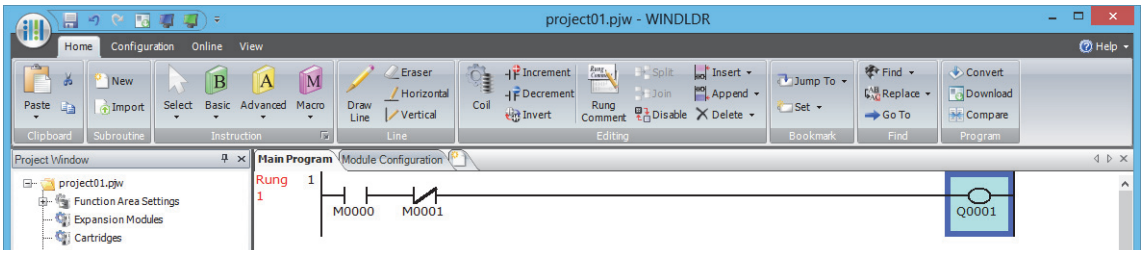
For details on devices, see "Device" on page 2-1.

For details on instructions, see the "FC6A Series MicroSmart LAD Programming Manual".

- **Operation procedure**

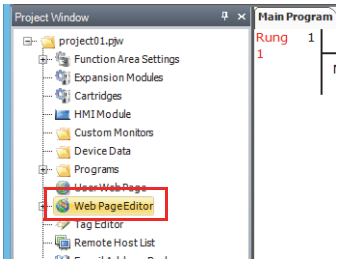
1. Create a program that turns on output Q1 when input M0000 is on and input M0001 is off.

For details, see Chapter 4 "Create Ladder Program" in the "FC6A Series MicroSmart All-in-One Type User's Manual".

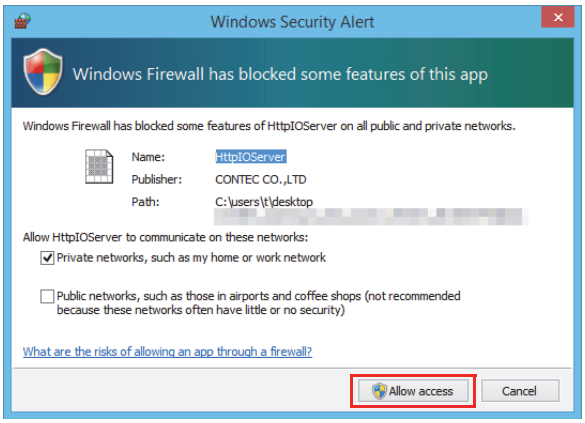


2. Double-click **Web Page Editor** in the Project window.

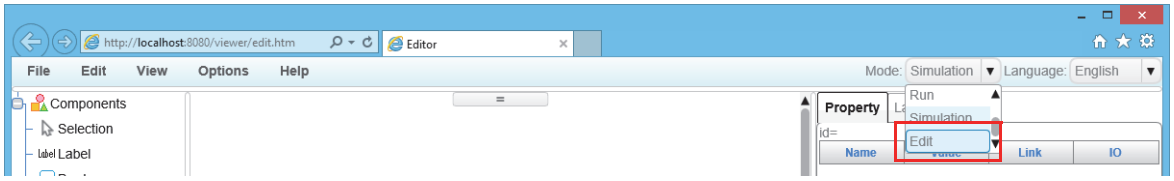
Web Page Editor starts.



Note: If the **Windows Security Alert** dialog box is displayed, click **Allow access**.



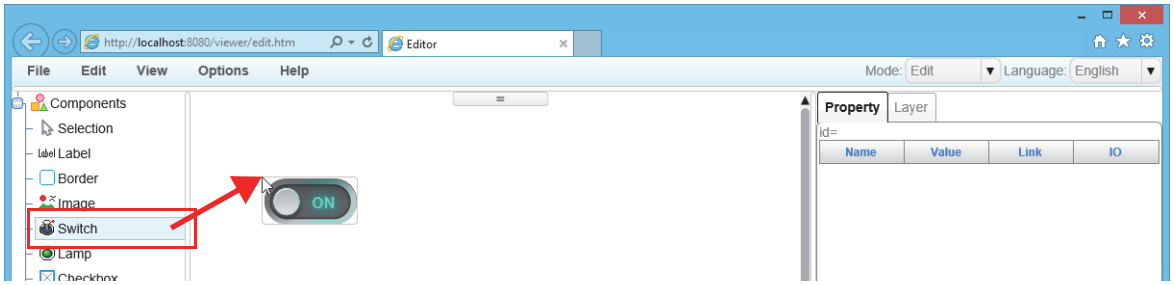
3. For the Web Page Editor **Mode**, click **Edit**.



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.

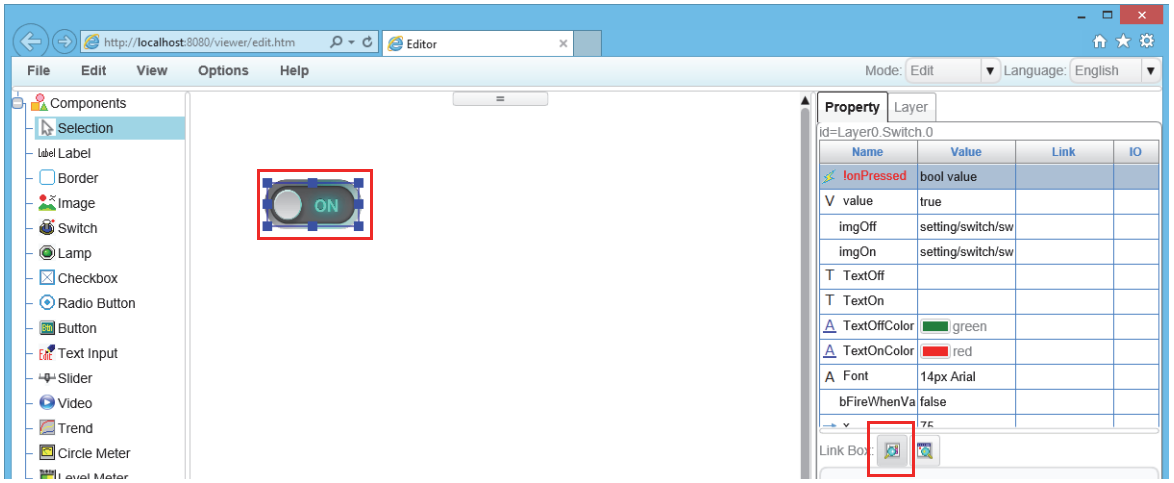
   The switch will be placed on the area with the predefined size.



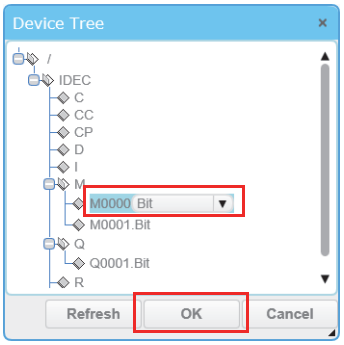
Note: To place a switch on the area with the desired size, click **Switch** in the **Components** list, and then drag the precision select mouse cursor on the work area.

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.



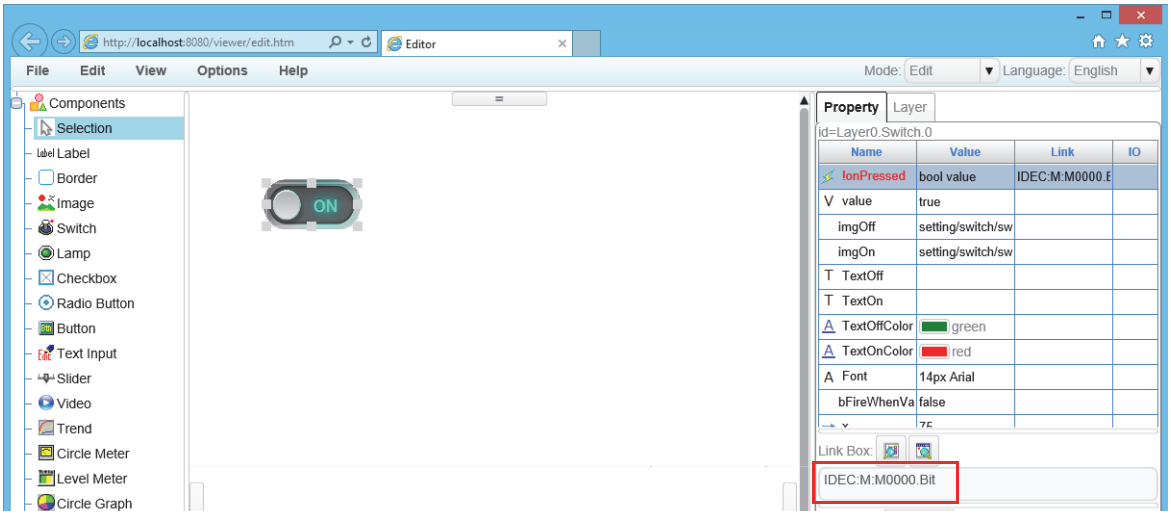
6. Click the device address to set to Switch 1 and click **OK**.



Notes:

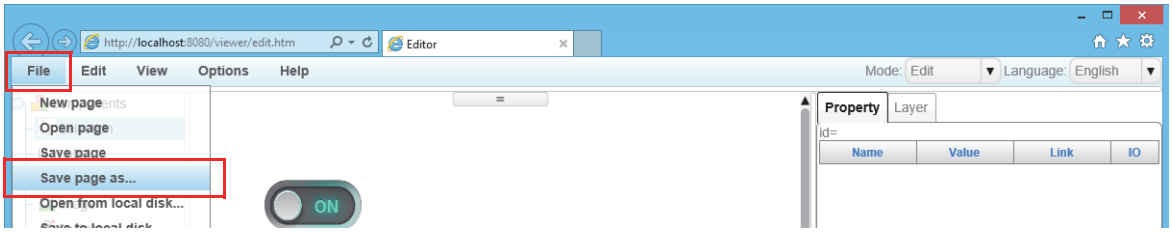
- The devices that are used in WindLDR are displayed in the **Device Tree** dialog box.
- If the device address to set is not in the Device Tree, click **Cancel**, and then directly enter the device address in the text box for **Link Box** on the **Property** tab.

This sets Switch 1 to a normally open contact (M0000).

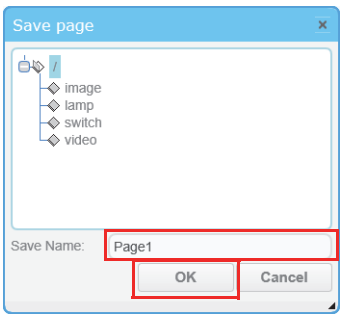


7. Repeat steps 4 through 6 and set Switch 2 and the lamp.

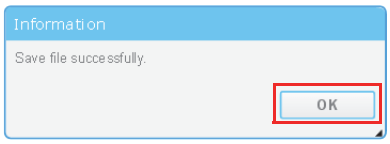
8. On the menu bar, click **File** > **Save page as**.
The **Save Page** dialog box is displayed.



9. Enter the file name in **Save Name**, specify the folder to save to, and click **OK**.
A confirmation message is displayed.



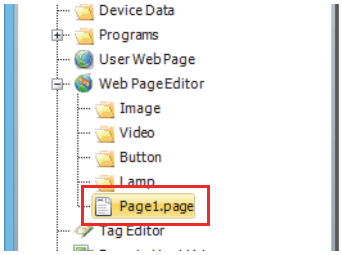
10. Click **OK**.



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.

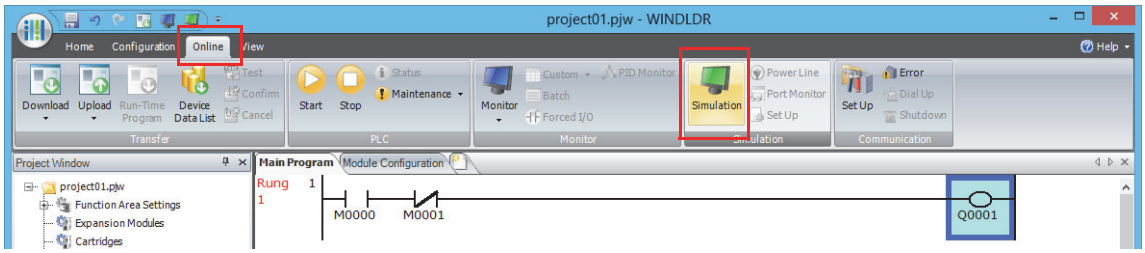


Checking Operation of User Web Pages

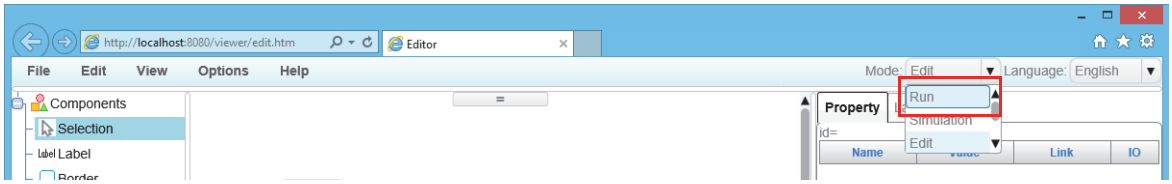
This section describes the procedure to execute operation of the created user web page in Web Page Editor to check its behavior.

- **Operation procedure**

1. On the WindLDR **Online** tab, click **Simulation**.

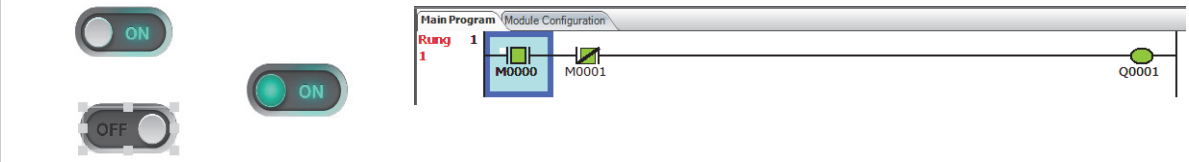


2. For the Web Page Editor **Mode**, click **Run**.

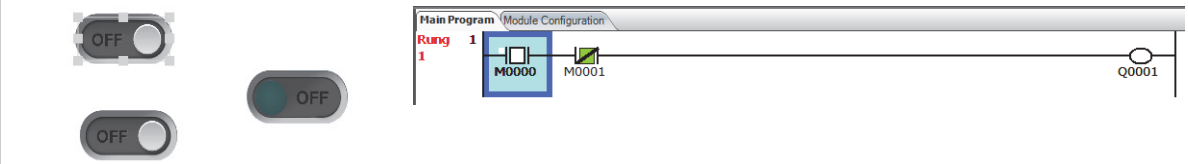


3. Check the following operation while turning Switch 1 and Switch 2 on and off.

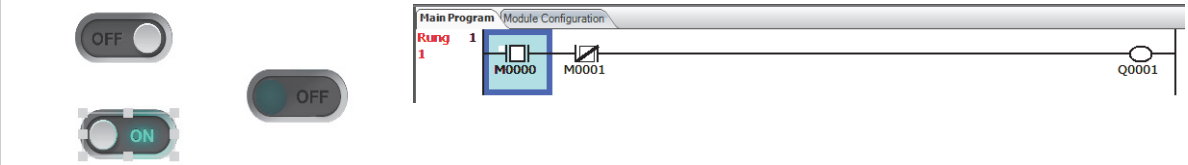
- The lamp turns on when Switch 1 is on and Switch 2 is off



- 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 "Programming WindLDR" on page 11-2.

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 8/7/Vista/XP	Google Chrome 47 or later Mozilla Firefox 42 or later Microsoft Internet Explorer 11
Android	Google Chrome 47 or later
iOS	Safari

INDEX