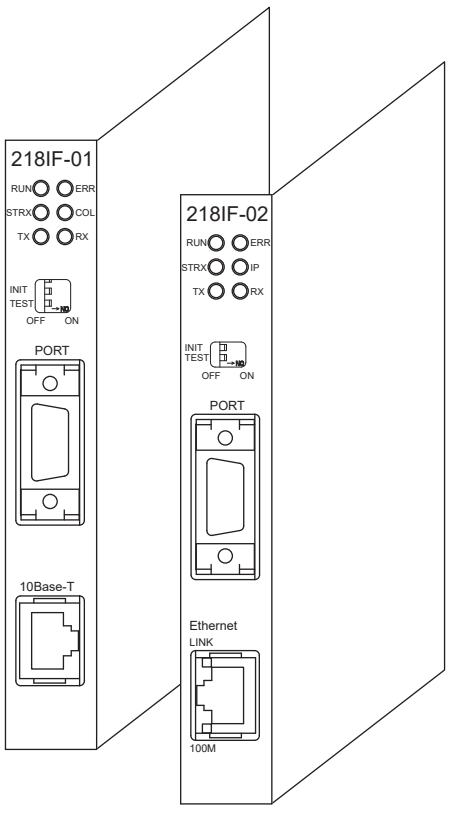
YASKAWA

Machine Controller MP2000 Series

Communication Module
USER'S MANUAL

Model: JAPMC-CM23□□



Starting Communication Modules 1
Communication Modes and Protocols 2
Serial Communication 3
218IF-01 Module 4
218IF-02 Module 5
217IF-01 Module 6
260IF-01 Module 7
261IF-01 Module 8
215AIF-01 Module 9
Appendices App

MANUAL NO. SIEP C880700 04Q

Copyright © 2003 YASKAWA ELECTRIC CORPORATION

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of Yaskawa. No patent liability is assumed with respect to the use of the information contained herein. Moreover, because Yaskawa is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, Yaskawa assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Using this Manual

Please read this manual to construct the communication system and network with the MP2000 Series Machine Controller. Keep this manual in a safe place for future reference.

■ Basic Terms

Unless otherwise specified, the following definitions are used:

- MP2000 Series Machine Controller: MP2100M, MP2200, MP2300, MP2300S, MP2310, and MP2400 Machine Controllers
- PLC: Programmable Logic Controller
- PP: Programming Panel
- MPE720: The Programming Device Software or a personal computer running the Programming Device Software

■ Manual Configuration

This manual consists of the chapters listed in the following table. Read the chapters of this manual as required by the purpose.

Purpose / Chapter	Selecting Models and Peripheral Devices	Studying Specifications and Ratings	Designing the System	Panel Installation and Wiring	Trial Operation	Maintenance and Inspection
Chapter 1 Starting Communication Modules	✓	–	✓	✓	✓	✓
Chapter 2 Communication Modes and Protocols	–	✓	✓	–	✓	–
Chapter 3 Serial Communication	–	✓	✓	–	✓	✓
Chapter 4 218IF-01 Module	✓	✓	✓	✓	✓	✓
Chapter 5 218IF-02 Module	✓	✓	✓	✓	✓	✓
Chapter 6 217IF-01 Module	✓	✓	✓	✓	✓	✓
Chapter 7 260IF-01 Module	✓	✓	✓	✓	✓	✓
Chapter 8 261IF-01 Module	✓	✓	✓	✓	✓	✓
Chapter 9 215AIF-01 Module	✓	✓	✓	✓	✓	✓

■ Graphic Symbols Used in this Manual

The graphic symbols used in this manual indicate the following type of information.

- This symbol is used to indicate important information that should be memorized or minor precautions, such as precautions that will result in alarms if not heeded.

■ MPE720 Engineering Tool Version Number

In this manual, the operation of MPE720 is described using screen captures of MPE720 version 7.
For this reason, the screen captures and some descriptions may differ for MPE720 version 5 or version 6.

■ Related Manuals

The following table lists the manuals relating to the MP2000 Series Machine Controller Communication Modules.

Refer to these manuals as required.

Manual Name	Manual Number	Contents
Machine Controller MP2100/MP2100M User's Manual Design and Maintenance	SIEP C880700 01	Describes how to use the MP2100 and MP2100M Machine Controllers.
Machine Controller MP2200 User's Manual	SIEP C880700 14	Describes how to use the MP2200 Machine Controller and the modules that can be connected.
Machine Controller MP2300 Basic Module User's Manual	SIEP C880700 03	Describes how to use the MP2300 Basic Module and the modules that can be connected.
Machine Controller MP2300S Basic Module User's Manual	SIEP C880732 00	Describes how to use the MP2300S Basic Module and the modules that can be connected.
Machine Controller MP2310 Basic Module User's Manual	SIEP C880732 01	Describes how to use the MP2310 Basic Module and the modules that can be connected.
Machine Controller MP2400 User's Manual	SIEP C880742 00	Describes how to use the MP2400 Machine Controller and the modules that can be connected.
Machine Controller MP2000 Series User's Manual Ladder Programming	SIE-C887-1.2	Describes the instructions used in MP2000 ladder programming.
Machine Controller MP2000 Series User's Manual for Motion Programming	SIEP C880700 38	Describes the instructions used in MP2000 motion programming.

■ Copyrights

- DeviceNet is a registered trademark of the ODVA (Open DeviceNet Venders Association).
- Ethernet is a registered trademark of the Xerox Corporation.
- PROFIBUS is a trademark of the PROFIBUS User Organization.
- MPLINK is a trademark of the Yaskawa Electric Corporation.
- MECHATROLINK is a trademark of the MECHATROLINK Members Association.
- Other product names and company names are the trademarks or registered trademarks of the respective company. "TM" and the ® mark do not appear with product or company names in this manual.

Safety Information

The following conventions are used to indicate precautions in this manual. Information marked as shown below is important for the safety of the user. Always read this information and heed the precautions that are provided. The conventions are as follows:

⚠ WARNING — Indicates precautions that, if not heeded, could possibly result in loss of life or serious injury.

⚠ CAUTION — Indicates precautions that, if not heeded, could result in relatively serious or minor injury, or property damage.

If not heeded, even precautions classified under ⚠ CAUTION can lead to serious results depending on circumstances.

PROHIBITED — Indicates prohibited actions. Specific prohibitions are indicated inside ⃠.

For example, indicates no fire or open flame.

MANDATORY — Indicates mandatory actions. Specific actions are indicated inside ●.

For example, indicates that grounding is required.

Safety Precautions

The following precautions are for checking products on delivery, storage, transportation, installation, wiring, operation, application, inspection, and disposal. These precautions are important and must be observed.

■ General Precautions

⚠ WARNING

- Before starting operation while connected to the machine, ensure that an emergency stop procedure has been provided and is working correctly.
  There is a risk of injury.
- Do not touch anything inside the product.
  There is a risk of electrical shock.
- Always keep the front cover attached when power is being supplied.
  There is a risk of electrical shock.
- Observe all procedures and precautions given in this manual for trial operation.
  Operating mistakes while the Servomotor and machine are connected can cause damage to the machine or even accidents resulting in injury or death.
- Do not remove the front cover, cables, connector, or options while power is being supplied.
  There is a risk of electrical shock.
- Do not damage, pull on, apply excessive force to, place heavy objects on, or pinch cables.
  There is a risk of electrical shock, operational failure of the product, or burning.
- Do not attempt to modify the product in any way.
  There is a risk of injury or device damage.
- Do not approach the machine when there is a momentary interruption to the power supply. When power is restored, the MP2000 Series Machine Controller or machine connected to it may start operation suddenly. Provide suitable safety measures to protect people when operation restarts.
  There is a risk of injury.
- Do not allow installation, disassembly, or repairs to be performed by anyone other than specified personnel.
  There is a risk of electrical shock or injury.

■ Storage and Transportation

⚠ CAUTION

- Do not store or install the product in locations subject to the following. There is a risk of fire, electric shock, and machine product damage.
  - Direct sunlight
  - Ambient temperatures exceeding the storage or operating conditions
  - Ambient humidity exceeding the storage or operating conditions
  - Extreme changes in temperature that would result in condensation
  - Corrosive or flammable gas
  - Excessive dust, dirt, salt, or metallic powder
  - Water, oil, or chemicals
  - Vibration or shock
- Do not overload the product during transportation.
  There is a risk of injury or an accident.
- Never subject the product to an atmosphere containing halogen (fluorine, chlorine, bromine, or iodine) during transportation or installation.
  There is a risk of device damage or an accident.
- If disinfectants or insecticides must be used to treat packing materials such as wooden frames, pallets, or plywood, the packing materials must be treated before the product is packaged, and methods other than fumigation must be used.
  Example: Heat treatment, where materials are kiln-dried to a core temperature of 56°C for 30 minutes or more.
  If the electronic products, which include stand-alone products and products installed in machines, are packed with fumigated wooden materials, the electrical components may be greatly damaged by the gases or fumes resulting from the fumigation process. In particular, disinfectants containing halogen, which includes chlorine, fluorine, bromine, or iodine can contribute to the erosion of the capacitors.

■ Installation

⚠ CAUTION

- Never use the product in locations subject to water, corrosive atmospheres, or flammable gas, or near burnable objects.
  There is a risk of electrical shock or fire.
- Do not step on the product or place heavy objects on the product.
  There is a risk of injury.
- Do not block the air exhaust port on the product. Do not allow foreign objects to enter the product.
  There is a risk of element deterioration inside, an accident, or fire.
- Always mount the product in the specified orientation.
  There is a risk of an accident.
- Do not subject the product to strong shock.
  There is a risk of an accident.

■ Wiring

⚠ CAUTION

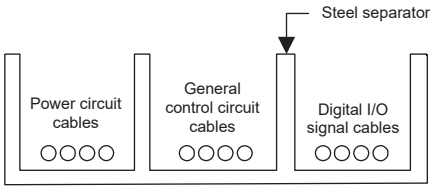
- Check the wiring to be sure it has been performed correctly.
  There is a risk of motor run-away, injury, or an accident.
- Always use a power supply of the specified voltage.
  There is a risk of burning.
- In places with poor power supply conditions, take all steps necessary to ensure that the input power is supplied within the specified voltage range.
  There is a risk of device damage.
- Install breakers and other safety measures to provide protection against shorts in external wiring.
  There is a risk of fire.
- Provide sufficient shielding when using the product in the locations subject to the following.
  There is a risk of device damage.
  - Noise, such as from static electricity
  - Strong electromagnetic or magnetic fields
  - Radiation
  - Near power lines
- When connecting the battery, connect the polarity correctly.
  There is a risk of battery damage or explosion.
- Allow only worker trained in safety measure to replace the battery.
  Unsuitable battery replacement may result in electric shock. It may also cause machine malfunction, resulting in worker injury or machine damage.
- Do not touch the electrodes when replacing the battery.
  There is a risk of electrostatic discharge failure.

■ Selecting, Separating, and Laying External Cables

⚠ CAUTION

- Consider the following items when selecting the I/O signal lines (external cables) to connect the product to external devices.
  - Mechanical strength
  - Noise interference
  - Wiring distance
  - Signal voltage, etc.
- Separate the I/O signal lines from the power lines both inside and outside the control box to reduce the influence of noise from the power lines.
  If the I/O signal lines and power lines are not separated properly, malfunctioning may result.

Example of Separated External Cables



■ Maintenance and Inspection

⚠ CAUTION
• Do not attempt to disassemble the product. There is a risk of electrical shock or injury. • Do not change wiring while power is being supplied. There is a risk of electrical shock or injury. • When replacing the MP2000 Series Machine Controller, restart operation only after transferring the programs and parameters from the old Controller to the new Controller. Machine damage may result if the new Controller is operated without this data.

■ Disposal

⚠ CAUTION
• Dispose of the product as general industrial waste.

■ General Precautions

Observe the following general precautions to ensure safe application.
• The products shown in illustrations in this manual are sometimes shown without covers or protective guards. Always replace the cover or protective guard as specified first, and then operate the products in accordance with the manual. • The drawings presented in this manual are typical examples and may not match the product you received. • If the manual must be ordered due to loss or damage, inform your nearest Yaskawa representative or one of the offices listed on the back of this manual.

Warranty

(1) Details of Warranty

■ Warranty Period

The warranty period for a product that was purchased (hereinafter called “delivered product”) is one year from the time of delivery to the location specified by the customer or 18 months from the time of shipment from the Yaskawa factory, whichever is sooner.

■ Warranty Scope

Yaskawa shall replace or repair a defective product free of charge if a defect attributable to Yaskawa occurs during the warranty period above. This warranty does not cover defects caused by the delivered product reaching the end of its service life and replacement of parts that require replacement or that have a limited service life.
This warranty does not cover failures that result from any of the following causes.

1. Improper handling, abuse, or use in unsuitable conditions or in environments not described in product catalogs or manuals, or in any separately agreed-upon specifications
2. Causes not attributable to the delivered product itself
3. Modifications or repairs not performed by Yaskawa
4. Abuse of the delivered product in a manner in which it was not originally intended
5. Causes that were not foreseeable with the scientific and technological understanding at the time of shipment from Yaskawa
6. Events for which Yaskawa is not responsible, such as natural or human-made disasters

(2) Limitations of Liability

1. Yaskawa shall in no event be responsible for any damage or loss of opportunity to the customer that arises due to failure of the delivered product.
2. Yaskawa shall not be responsible for any programs (including parameter settings) or the results of program execution of the programs provided by the user or by a third party for use with programmable Yaskawa products.
3. The information described in product catalogs or manuals is provided for the purpose of the customer purchasing the appropriate product for the intended application. The use thereof does not guarantee that there are no infringements of intellectual property rights or other proprietary rights of Yaskawa or third parties, nor does it construe a license.
4. Yaskawa shall not be responsible for any damage arising from infringements of intellectual property rights or other proprietary rights of third parties as a result of using the information described in catalogs or manuals.

(3) Suitability for Use

1. It is the customer's responsibility to confirm conformity with any standards, codes, or regulations that apply if the Yaskawa product is used in combination with any other products.
2. The customer must confirm that the Yaskawa product is suitable for the systems, machines, and equipment used by the customer.
3. Consult with Yaskawa to determine whether use in the following applications is acceptable. If use in the application is acceptable, use the product with extra allowance in ratings and specifications, and provide safety measures to minimize hazards in the event of failure.
    - Outdoor use, use involving potential chemical contamination or electrical interference, or use in conditions or environments not described in product catalogs or manuals
    - Nuclear energy control systems, combustion systems, railroad systems, aviation systems, vehicle systems, medical equipment, amusement machines, and installations subject to separate industry or government regulations
    - Systems, machines, and equipment that may present a risk to life or property
    - Systems that require a high degree of reliability, such as systems that supply gas, water, or electricity, or systems that operate continuously 24 hours a day
    - Other systems that require a similar high degree of safety
4. Never use the product for an application involving serious risk to life or property without first ensuring that the system is designed to secure the required level of safety with risk warnings and redundancy, and that the Yaskawa product is properly rated and installed.
5. The circuit examples and other application examples described in product catalogs and manuals are for reference. Check the functionality and safety of the actual devices and equipment to be used before using the product.
6. Read and understand all use prohibitions and precautions, and operate the Yaskawa product correctly to prevent accidental harm to third parties.

(4) Specifications Change

The names, specifications, appearance, and accessories of products in product catalogs and manuals may be changed at any time based on improvements and other reasons. The next editions of the revised catalogs or manuals will be published with updated code numbers. Consult with your Yaskawa representative to confirm the actual specifications before purchasing a product.

Contents

Using this Manual ---- iii
Safety Information ---- v
Safety Precautions ---- v
Warranty ---- ix

1 Starting Communication Modules ---- 1-1

1.1 Starting Modules ---- 1-2
1.1.1 Overall Flow of Starting Modules ---- 1-2

1.2 Preparing the Equipment ---- 1-3
1.2.1 Checking the Equipment ---- 1-3
1.2.2 Applicable Machine Controllers ---- 1-3
1.2.3 Applicable Versions of CPU Module ---- 1-5
1.2.4 Software Versions of Supported Communications Module ---- 1-6
1.2.5 Applicable Versions of MPE720 ---- 1-8

1.3 Mounting and Removing a Module on Machine Controller ---- 1-12
1.3.1 Mounting a Communication Module ---- 1-12
1.3.2 Removing a Communication Module ---- 1-14

1.4 Setting the Communication Platform ---- 1-17
1.4.1 Opening the Communication Platform ---- 1-17
1.4.2 Setting the Communication Port ---- 1-18

1.5 Executing Self-configuration ---- 1-26

1.6 Starting the MPE720, and Setting Communication or Network Parameters ---- 1-28
1.6.1 Starting MPE720 Ver. 7 and Setting Communication ---- 1-28

1.7 Communication Module Transmission Definitions ---- 1-29
1.7.1 Displaying the Module Configuration Window ---- 1-29
1.7.2 Module Configuration Window ---- 1-30
1.7.3 Transmission Definitions ---- 1-30

2 Communication Modes and Protocols ---- 2-1

2.1 Communication Modes ---- 2-2
2.1.1 Overview ---- 2-2
2.1.2 Message Communication ---- 2-3
2.1.3 Engineering Communication ---- 2-5
2.1.4 Link Communication ---- 2-5

2.2 Message Communication Protocols ---- 2-6
2.2.1 Types of Communication Protocol ---- 2-6
2.2.2 MEMOBUS Protocol ---- 2-6
2.2.3 Extended MEMOBUS Protocol ---- 2-7
2.2.4 MELSEC Protocol ---- 2-8
2.2.5 OMRON Protocol ---- 2-9
2.2.6 MODBUS/TCP Protocol ---- 2-9
2.2.7 Non-Procedure Protocol ---- 2-10

3 Serial Communication - 3-1

3.1 Serial Communications - 3-3

3.2 Message Communication Protocols - 3-4

3.2.1 MEMOBUS Protocol - 3-4
3.2.2 MELSEC Protocol - 3-5
3.2.3 OMRON Protocol - 3-8
3.2.4 Non-procedure Protocol - 3-10
3.2.5 Non-procedure FD Protocol - 3-11

3.3 217IF Transmission Definitions - 3-14

3.3.1 217IF Transmission Configuration Window - 3-14
3.3.2 Setting the Parameters - 3-16
3.3.3 Saving and Deleting 217IF Transmission Definitions - 3-19

3.4 Time Required for Transmission- 3-20

3.4.1 Overview - 3-20
3.4.2 Estimating the Time Required for Each Step - 3-20

3.5 Serial Communication Connection Example - 3-22

3.5.1 Connecting to the 217IF-01 - 3-22
3.5.2 Connecting to an Operator Interface Using MEMOBUS Communication - - - - - - - - - - - - - 3-27
3.5.3 Connection to MELSEC - 3-32
3.5.4 Connection to an OMRON PLC - 3-37
3.5.5 Connection to a Temperature Controller - 3-42
3.5.6 Connection to an Inverter - 3-48
3.5.7 Connection to Programming Device - 3-51

3.6 Message Send Function - 3-53

3.6.1 Basic Specifications - 3-53
3.6.2 Inputs and Outputs for the Message Send Function - 3-54
3.6.3 Parameter List for Serial Communication (217IF)- 3-58
3.6.4 Function Setting Example and Parameter Details for the MEMOBUS Protocol - - - - - - - - - - 3-59
3.6.5 Function Setting Example and Parameter Details for the MELSEC Protocol- - - - - - - - - - - 3-67
3.6.6 Function Setting Example and Parameter Details for the OMRON Protocol - - - - - - - - - - - 3-71
3.6.7 Function Setting Example and Parameter Details for the Non-procedure Protocol - - - - - - - 3-75

3.7 Message Receive Function - 3-79

3.7.1 Basic Specifications - 3-79
3.7.2 Inputs and Outputs for the Message Receive Function- 3-80
3.7.3 Parameter List for Serial Communication (217IF)- 3-84
3.7.4 Function Setting Example and Parameter Details for the MEMOBUS Protocol - - - - - - - - - - 3-85
3.7.5 Function Setting Example and Parameter Details for the MELSEC Protocol- - - - - - - - - - - 3-91
3.7.6 Function Setting Example and Parameter Details for the OMRON Protocol - - - - - - - - - - - 3-94
3.7.7 Function Setting Example and Parameter Details for the Non-procedure Protocol - - - - - - - 3-97

4 218IF-01 Module- 4-1

4.1 Overview - 4-2

4.1.1 Appearance and Connectors - 4-2
4.1.2 System Configuration Example - 4-3
4.1.3 Module Specifications - 4-4
4.1.4 Indicators- 4-6
4.1.5 Switch Settings- 4-7
4.1.6 Default Parameters at Initial Startup - 4-8

4.2 Connection Specifications - 4-9

4.2.1 Connector Specifications- 4-9
4.2.2 Cable Specifications - 4-9

4.3 RS-232C Communication - - - - 4-10

4.4 218IF Transmission Definitions - - - - 4-11
4.4.1 218IF Transmission Configuration Window - - - - 4-11
4.4.2 Transmission Parameters Tab Page - - - - 4-13
4.4.3 Status Tab Page - - - - 4-20
4.4.4 Saving and Deleting 218IF Transmission Definitions - - - - 4-23

4.5 Ethernet Communication - - - - 4-24
4.5.1 Overview - - - - 4-24
4.5.2 Ethernet Communication Protocol - - - - 4-24
4.5.3 MEMOBUS Protocol - - - - 4-25
4.5.4 Extended MEMOBUS Protocol - - - - 4-26
4.5.5 MELSEC Protocol - - - - 4-27
4.5.6 MODBUS/TCP Protocol - - - - 4-31
4.5.7 Non-Procedure Protocol - - - - 4-32

4.6 Connection Examples for Ethernet Communication - - - - 4-34
4.6.1 Connecting 218IF-01 Modules - - - - 4-34
4.6.2 Connection to MELSEC - - - - 4-39
4.6.3 Connection to a Personal Computer - - - - 4-43

4.7 Message Send Function and Message Receive Function - - - - 4-47

5 218IF-02 Module - - - - 5-1

5.1 Overview - - - - 5-3
5.1.1 Appearance and Connectors - - - - 5-3
5.1.2 System Configuration Examples - - - - 5-4
5.1.3 Module Specifications - - - - 5-6
5.1.4 Indicators - - - - 5-8
5.1.5 Switch Settings - - - - 5-9
5.1.6 Default Parameters at Initial Startup - - - - 5-10

5.2 Connection Specifications - - - - 5-11
5.2.1 Connector Specifications - - - - 5-11
5.2.2 Cable Specifications - - - - 5-11

5.3 RS-232C Communication - - - - 5-12

5.4 218IFB Transmission Definitions - - - - 5-13
5.4.1 218IFB Transmission Configuration Window - - - - 5-13
5.4.2 Transmission Parameters Tab Page - - - - 5-15
5.4.3 Status Tab Page - - - - 5-20
5.4.4 Saving and Deleting 218IFB Transmission Definitions - - - - 5-23

5.5 Ethernet Communication - - - - 5-24
5.5.1 Overview - - - - 5-24
5.5.2 Ethernet Communication Protocols - - - - 5-24
5.5.3 MEMOBUS Protocol - - - - 5-25
5.5.4 Extended MEMOBUS Protocol - - - - 5-26
5.5.5 MELSEC Protocol - - - - 5-27
5.5.6 MODBUS/TCP Protocol - - - - 5-32
5.5.7 Non-Procedure Protocol - - - - 5-33

5.6 Connection Examples for Ethernet Communication - - - - 5-35
5.6.1 Connecting 218IF-02 Modules - - - - 5-35
5.6.2 Connection to MELSEC - - - - 5-40
5.6.3 Connection to a Personal Computer - - - - 5-44

5.7 Message Send Function - - - 5-48
5.7.1 Basic Specifications - - - 5-48
5.7.2 Inputs and Outputs for the Message Send Function - - - 5-49
5.7.3 Parameter List for Ethernet Communication (218IF or 218IFB) - - - 5-53
5.7.4 Function Setting Example and Parameter Details for the MEMOBUS and Extended MEMOBUS Protocol - - - 5-54
5.7.5 Function Setting Example and Parameter Details for the MELSEC Protocol - - - 5-62
5.7.6 Function Setting Example and Parameter Details for the MODBUS/TCP Protocol - - - 5-67
5.7.7 Function Setting Example and Parameter Details for the Non-procedure Protocol - - - 5-73

5.8 Message Receive Function - - - 5-78
5.8.1 Basic Specifications - - - 5-78
5.8.2 Inputs and Outputs for the Message Receive Function - - - 5-79
5.8.3 Parameter List for Ethernet Communication (218IF or 218IFB) - - - 5-83
5.8.4 Function Setting Example and Parameter Details for the MEMOBUS and Extended MEMOBUS Protocol - - - 5-84
5.8.5 Function Setting Example and Parameter Details for the MELSEC Protocol - - - 5-91
5.8.6 Function Setting Example and Parameter Details for the MODBUS/TCP Protocol - - - 5-95
5.8.7 Function Setting Example and Parameter Details for the Non-procedure Protocol - - - 5-100

6 217IF-01 Module - - - 6-1

6.1 Overview - - - 6-2
6.1.1 Appearance and Connectors - - - 6-2
6.1.2 System Configuration Examples - - - 6-3
6.1.3 Module Specifications - - - 6-4
6.1.4 Indicators - - - 6-6
6.1.5 Switch Settings - - - 6-7
6.1.6 Default Parameters at Initial Startup - - - 6-7
6.1.7 Online Switching of Transmission Parameter Definitions by Setting INIT Switch (Software Version 1.10 or Later) - - - 6-8

6.2 Connection Specifications - - - 6-9
6.2.1 Connector Specifications - - - 6-9
6.2.2 Cable Specifications - - - 6-10
6.2.3 Module Connection Examples - - - 6-11

6.3 Serial Communication - - - 6-12

7 260IF-01 Module - - - 7-1

7.1 Overview - - - 7-2
7.1.1 Appearance and Connectors - - - 7-2
7.1.2 Network Configuration - - - 7-3
7.1.3 System Configuration Examples - - - 7-8
7.1.4 Module Specifications - - - 7-9
7.1.5 Indicators - - - 7-11
7.1.6 Switch Settings - - - 7-12
7.1.7 Default Parameters at Initial Startup - - - 7-12
7.1.8 Operation for Self-configuration - - - 7-13

7.2 Connection Specifications - - - 7-14
7.2.1 Connector Specifications - - - 7-14
7.2.2 Cable Specifications - - - 7-15
7.2.3 DeviceNet System Configuration Example - - - 7-15

7.3 RS-232C Communication - - - 7-16

7.4 260IF Transmission Definitions - - - 7-17
7.4.1 260IF Transmission Configuration Window - - - 7-17
7.4.2 Transmission Parameters Tab Page - - - 7-19
7.4.3 I/O Status Tab Page - - - 7-21
7.4.4 Status Tab Page - - - 7-22
7.4.5 Saving and Deleting 260IF Transmission Definitions - - - 7-23
7.4.6 Parameter Setting Examples - - - 7-24
7.4.7 Communication Cycle Time - - - 7-26
7.4.8 Troubleshooting - - - 7-28

7.5 Connection Examples for DeviceNet Communication - - - 7-32
7.5.1 I/O Communication Example - - - 7-32
7.5.2 Setting Example for Explicit Message Communication - - - 7-35

7.6 Message Send Function - - - 7-40
7.6.1 Basic Specifications - - - 7-40
7.6.2 Inputs and Outputs for the Message Send Function - - - 7-41
7.6.3 Parameter List for DeviceNet Communication (260IF) - - - 7-45
7.6.4 Function Setting Example and Parameter Details for Explicit Messages - - - 7-45

8 261IF-01 Module - - - 8-1

8.1 Overview - - - 8-2
8.1.1 Appearance and Connectors - - - 8-2
8.1.2 System Configuration Example - - - 8-3
8.1.3 Module Specifications - - - 8-4
8.1.4 Indicators - - - 8-6
8.1.5 Switch Settings - - - 8-7
8.1.6 Default Parameters at Initial Startup - - - 8-7

8.2 Connection Specifications - - - 8-8
8.2.1 Connector Specifications - - - 8-8
8.2.2 Cable Specifications - - - 8-9

8.3 RS-232C Communication - - - 8-9

8.4 261IF Transmission Definitions - - - 8-10
8.4.1 261IF Transmission Configuration Window - - - 8-10
8.4.2 I/O Assignment Tab Page - - - 8-12
8.4.3 I/O Map Tab Page - - - 8-14
8.4.4 I/O Status Tab Page - - - 8-15
8.4.5 Status Tab Page - - - 8-16
8.4.6 Saving and Deleting 261IF Transmission Definitions - - - 8-17

9 215AIF-01 Module - - - 9-1

9.1 Overview - - - 9-3
9.1.1 Appearance and Connectors - - - 9-3
9.1.2 System Configuration Examples - - - 9-4
9.1.3 Module Specifications - - - 9-6
9.1.4 Indicators - - - 9-8
9.1.5 Switch Settings - - - 9-9
9.1.6 Default Parameters at Initial Startup - - - 9-10

9.2 Connection Specifications - - - 9-11
9.2.1 Connector Specifications - - - 9-11
9.2.2 Cable Specifications - - - 9-13

9.3 RS-232C Communication - - - 9-15

9.4 MPLINK Transmission Definitions - 9-16
9.4.1 MPLINK Transmission Configuration Window - 9-16
9.4.2 Transmission Parameters Tab Page - 9-18
9.4.3 Link Assignment Tab Page - 9-20
9.4.4 I/O Map Tab Page - 9-21
9.4.5 Status Tab Page - 9-22
9.4.6 Saving and Deleting MPLINK Transmission Definitions - 9-24

9.5 Transmission Methods in MPLINK/CP-215 Communication - - - - - - - - - - - - - - - 9-25
9.5.1 Link Communication - 9-25
9.5.2 Engineering communication - 9-29
9.5.3 Message Communication - 9-30
9.5.4 Token Passing - 9-32
9.5.5 Relay Function - 9-34

9.6 Connection Examples for MPLINK/CP-215 Communication - - - - - - - - - - - - - - 9-38
9.6.1 Link Communication Example - 9-38
9.6.2 Message Communication Example - 9-42

9.7 Message Send Function - 9-46
9.7.1 Basic Specifications - 9-46
9.7.2 Inputs and Outputs for the Message Send Function - 9-47
9.7.3 Parameter List for MPLINK/CP-215 Transmissions (MPLINK) - 9-51
9.7.4 Function Setting Example and Parameter Details for the Extended MEMOBUS Protocol - - - 9-52
9.7.5 Function Setting Example and Parameter Details for the Non-procedure Protocol - - - - - - - 9-59

9.8 Message Receive Function - 9-63
9.8.1 Basic Specifications - 9-63
9.8.2 Inputs and Outputs for the Message Receive Function - 9-64
9.8.3 Parameter List for MPLINK/CP-215 Transmissions (MPLINK) - 9-68
9.8.4 Function Setting Example and Parameter Details for the Extended MEMOBUS Protocol - - - 9-69
9.8.5 Function Setting Example and Parameter Details for the Non-procedure Protocol - - - - - - - 9-75

Appendices - A-1

Appendix A Supplementary Information on Message Send and Receive Functions - - A-2
A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND - - - - - - - - - - - - - - A-2
A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV - - - - - - - - - - - - - - A-5
A.3 Transmission Buffer Channels - A-9

Appendix B Details on the Extended MEMOBUS Protocol - - - - - - - - - - - - - - - - - - - A-14
B.1 Message Configuration - A-14
B.2 MEMOBUS Binary Mode - A-16
B.3 MEMOBUS ASCII Mode - A-24
B.4 General-purpose Messages in Binary Mode - A-24
B.5 General-purpose Messages in ASCII Mode - A-25

Appendix C Details on the MEMOBUS Protocol - A-26
C.1 Message Configuration - A-26
C.2 MEMOBUS RTU Mode - A-27
C.3 MEMOBUS ASCII Mode - A-32

Appendix D Details on Non-procedure Protocol - A-33
D.1 Message Configuration - A-33
D.2 General-purpose Messages in Binary Mode - A-34
D.3 General-purpose Messages in ASCII Mode - A-34

Appendix E Sample C Language Programs for the Master - - - - - - - - - - - - - - - - - - A-35
E.1 TCP (When Using Extended MEMOBUS Protocol (SFC = 09)) - A-35
E.2 UDP (When Using Extended MEMOBUS Protocol (SFC = 09)) - A-39

Appendix F Sample C Language Programs for a Slave - - - - - - - - - - - - - - - - - - - A-41
F.1 TCP (When Using Extended MEMOBUS Protocol) - A-41
F.2 UDP (When Using Extended MEMOBUS Protocol) - A-46
Appendix G Junction Box Connection Example for CP-215 Communications- - - - A-48
G.1 Connections with the 215AIF-01 Module - A-48
G.2 Connection to a Personal Computer - A-49
Index
Revision History

1

Starting Communication Modules

This chapter describes how to connect and start Communication Modules, including installing Communication Modules, setting the Communication Platform, using self-configuration.

1.1 Starting Modules ----------1-2
1.1.1 Overall Flow of Starting Modules ---------- 1-2
1.2 Preparing the Equipment ----------1-3
1.2.1 Checking the Equipment ---------- 1-3
1.2.2 Applicable Machine Controllers ---------- 1-3
1.2.3 Applicable Versions of CPU Module ---------- 1-5
1.2.4 Software Versions of Supported Communications Module ---------- 1-6
1.2.5 Applicable Versions of MPE720 ---------- 1-8
1.3 Mounting and Removing a Module on Machine Controller ----------1-12
1.3.1 Mounting a Communication Module ---------- 1-12
1.3.2 Removing a Communication Module ---------- 1-14
1.4 Setting the Communication Platform ----------1-17
1.4.1 Opening the Communication Platform ---------- 1-17
1.4.2 Setting the Communication Port ---------- 1-18
1.5 Executing Self-configuration ----------1-26
1.6 Starting the MPE720, and Setting Communication or Network Parameters ----------1-28
1.6.1 Starting MPE720 Ver. 7 and Setting Communication ---------- 1-28
1.7 Communication Module Transmission Definitions ----------1-29
1.7.1 Displaying the Module Configuration Window ---------- 1-29
1.7.2 Module Configuration Window ---------- 1-30
1.7.3 Transmission Definitions ---------- 1-30

1.1 Starting Modules

This section describes the procedures from installing a Communication Module on the Machine Controller to setting the transmission definitions using the MPE720 Programming Device.

1.1.1 Overall Flow of Starting Modules

1	Preparing the Equipment The Communication Module and other equipment are prepared.	→	Refer to *1.2 Preparing the Equipment* on page 1-3.
⇩			
2	Installing the Module on the Machine Controller The Communication Module is installed on the MP2000 Series Machine Controller.	→	Refer to *1.3 Mounting and Removing a Module on Machine Controller* on page 1-12.
⇩			
3	Connecting a Personal Computer and the Remote Devices A personal computer for engineering and the remote devices are connected.	→	Refer to the descriptions of connections in chapters 4 through 9 for the Communication Module being used.
⇩			
4	Setting the Communication Platform Communication between the personal computer and Machine Controller is set.	→	Refer to *1.4 Setting the Communication Platform* on page 1-17.
⇩			
5	Assignment in the MP2000 Series Machine Controller The self-configuration tool in the MP2000 Series Machine Controller is executed so that the Communication Module is recognized automatically.	→	Refer to *1.5 Executing Self-configuration* on page 1-26.
⇩			
6	Starting the MPE720 and Setting Communication The MPE720 is started, and communication or network parameters are set.	→	Refer to *1.6 Starting the MPE720, and Setting Communication or Network Parameters* on page 1-28.
⇩			
7	Setting the Transmission Definitions The transmission parameters and other items are set and saved to flash memory.	→	Refer to *1.7 Communication Module Transmission Definitions* on page 1-29.

1.2 Preparing the Equipment

1.2.1 Checking the Equipment

To enable using the MP2000 Series Communication Module, make sure that you have all of the following equipment.

- MP2000 Series Machine Controller
- Communication Module
- The remote devices to be connected
- Cables
- A personal computer with the MPE720 Programming Device installed

1.2.2 Applicable Machine Controllers

(1) MP2000 Series

The MP2000 Series Machine Controllers to which Communication Modules can be mounted are listed in the following table.

Name		Model	Max. No. of Connectable Modules	Remarks
MP2300		JEPMC-MP2300 (-E)	3 modules	–
MP2310		JEPMC-MP2310 (-E)	3 modules	–
MP2300S		JEPMC-MP2300S (-E)	1 module	–
MP2200*1	CPU-01	JAPMC-CP2200 (-E)	8 modules	The maximum number of connectable Modules is the total for the maximum expansion to four racks.*2
	CPU-02	JAPMC-CP2210 (-E)		
	CPU-03	JAPMC-CP2220-E		
	CPU-04	JAPMC-CP2230-E		
MP2100M		JAPMC-MC2140 (-E)	8 modules	The maximum number of connectable Modules is the total for the maximum expansion to three racks.*2
MP2101M		JAPMC-MC2142-E		
MP2101TM		JAPMC-MC2142T-E		

* 1. Mount a CPU module on the following base units.

Name	Model	Remarks
MBU-01	JEPMC-BU2200 (-E)	100/200-VAC input base unit (9 slots)
MBU-02	JEPMC-BU2210 (-E)	24-VDC input base unit (9 slots)
MBU-03	JEPMC-BU2220-E	24-VDC input base unit (4 slots)

* 2. The following module or board is required between racks.

Name	Model	Remarks
EXIOIF	JAPMC-EX2200 (-E)	Inter-rack connection module
MP2100MEX	JAPMC-EX2100 (-E)	I/F board for MP2100M, MP2101M, and MP2101TM

(2) MP3000 Series

The MP3000 Series Machine Controllers to which Communication Modules can be mounted are listed in the following table.

Name		Model	Max. No. of Connectable Modules	Remarks
MP3100	MP3100 (16 axes)	JAPMC-MC3100-1-E	16 modules	The maximum number of connectable Modules is the total for the maximum expansion to three racks.*2
	MP3100 (32 axes)	JAPMC-MC3100-2-E	16 modules	
MP3200	CPU-201	JEPMC-CP3201-E	16 modules	The maximum number of connectable Modules is the total for the maximum expansion to four racks.*2
	CPU-202	JEPMC-CP3202-E	16 modules	
MP3300*1	CPU-301 (16 axes)	JAPMC-CP3301-1-E	16 modules	
	CPU-301 (32axes)	JAPMC-CP3301-2-E	16 modules	
	CPU-302 (16 axes)	JAPMC-CP3302-1-E	16 modules	
	CPU-302 (32 axes)	JAPMC-CP3302-2-E	16 modules	

* 1. Mount a CPU module on the following base units.

Name	Model	Remarks
MBU-301	JEPMC-BU3301-E	100/200-VAC input base unit (8 slots)
MBU-302	JEPMC-BU3302-E	24-VDC input base unit (8 slots)
MBU-303	JEPMC-BU3303-E	24-VDC input base unit (3 slots)
MBU-304	JEPMC-BU3304-E	24-VDC input base unit (1 slot)

* 2. The following module or board is required between racks.

Name	Model	Remarks
EXIOIF	JAPMC-EX2200 (-E)	Inter-rack connection module
MP3100EX	JAPMC-EX3100-E	Can be connected to the Expansion Interface Module and the EXIOIF Module.
MP3101EX	JAPMC-EX3101-E	Can be connected to the EXIOIF Module.

1.2.3 Applicable Versions of CPU Module

The software versions of supported Communication Modules may differ according to the protocol.

<table>
<tr><th colspan="2" rowspan="2">Machine Controller</th><th colspan="3">Applicable Versions of CPU Module</th></tr>
<tr><th>Using the 217IF-01, 218IF-01, 260IF-01, and 261IF-01</th><th>Using the 218IF-02</th><th>Using the 215AIF-01</th></tr>
<tr><td colspan="2">MP2300</td><td rowspan="11">All versions</td><td>Ver. 2.60 or later</td><td>Ver. 2.41 or later</td></tr>
<tr><td colspan="2">MP2310</td><td rowspan="2">All versions</td><td rowspan="2">All versions</td></tr>
<tr><td colspan="2">MP2300S</td></tr>
<tr><td rowspan="4">MP2200</td><td>CPU-01</td><td rowspan="2">Ver. 2.60 or later</td><td rowspan="2">Ver. 2.41 or later</td></tr>
<tr><td>CPU-02</td></tr>
<tr><td>CPU-03</td><td rowspan="2">All versions</td><td rowspan="2">All versions</td></tr>
<tr><td>CPU-04</td></tr>
<tr><td colspan="2">MP2100M</td><td>Ver. 2.60 or later</td><td>Ver. 2.41 or later</td></tr>
<tr><td colspan="2">MP2101M</td><td rowspan="3">All versions</td><td rowspan="3">All versions</td></tr>
<tr><td colspan="2">MP2101TM</td></tr>
<tr><td rowspan="2">MP3100</td><td>MP3100 (16 axes)</td></tr>
<tr><td>MP3100 (32 axes)</td><td>All versions</td><td>All versions</td><td>All versions</td></tr>
<tr><td rowspan="2">MP3200</td><td>CPU-201</td><td rowspan="2">Ver. 1.02 or later</td><td rowspan="2">Ver. 1.02 or later</td><td rowspan="2">Ver. 1.02 or later</td></tr>
<tr><td>CPU-202</td></tr>
<tr><td rowspan="4">MP3300</td><td>CPU-301 (16 axes)</td><td rowspan="4">All versions</td><td rowspan="4">All versions</td><td rowspan="4">All versions</td></tr>
<tr><td>CPU-301 (32 axes)</td></tr>
<tr><td>CPU-302 (16 axes)</td></tr>
<tr><td>CPU-302 (32 axes)</td></tr>
</table>

1.2.4 Software Versions of Supported Communications Module

The software versions of supported Communication Modules may differ according to the protocol.
The system version is given on a label on the PCB of the Communication Module as "V**.**".

- For a portion of functions, the software versions of supported Communication Modules may differ from the software versions listed in the table. If the software versions are different, the software version is listed on the page that describes the functions.

(1) 217IF-01

MEMOBUS	MELSEC A-compatible 1C	OMRON Host Link	Non-procedure Protocol	Non-procedure Protocol FD
All versions	All versions	Ver. 1.05 or later	All versions	Ver. 2.00 or later

(2) 218IF-01

[a] Function Module: 217IF (Serial Communications)

MEMOBUS	MELSEC A-compatible 1C	OMRON Host Link	Non-procedure Protocol
All versions	All versions	Ver. 1.08 or later	All versions

[b] Function Module: 218IF (Ethernet Communications)

MEMOBUS	Extended MEMOBUS	MELSEC A-compatible 1E	MODBUS/TCP	Non-procedure Protocol
All versions	All versions	All versions	All versions	All versions

(3) 218IF-02

[a] Function Module: 217IF (Serial Communications)

MEMOBUS	MELSEC A-compatible 1C	OMRON Host Link	Non-procedure Protocol
All versions	All versions	All versions	All versions

[b] Function Module: 218IF (Ethernet Communications)

MEMOBUS	Extended MEMOBUS	MELSEC A-compatible 1E	MELSEC QnA-compatible 3E	MODBUS/TCP	OMRON FINS	TOYOPUC File memory system	Non-procedure Protocol
All versions	All versions	All versions	Ver. 1.03 or later	All versions	Ver. 1.03 or later	Ver. 1.05 or later	All versions

(4) 260IF-01

[a] Function Module: 217IF (Serial Communications)

MEMOBUS	MELSEC A-compatible 1C	OMRON Host Link	Non-procedure Protocol
All versions	All versions	Ver. 1.09 or later	All versions

[b] Function Module: 260IF (DeviceNet Communications)

Explicit
All versions

(5) 261IF-01

[a] Function Module: 217IF (Serial Communications)

MEMOBUS	MELSEC A-compatible 1C	OMRON Host Link	Non-procedure Protocol
All versions	All versions	Ver. 1.05 or later	All versions

[b] Function Module: 261IF (PROFIBUS communications)

This function module does not support any protocol.

(6) 215AIF-01

[a] Function Module: 217IF (Serial Communications)

MEMOBUS	MELSEC A-compatible 1C	OMRON Host Link	Non-procedure Protocol
All versions	All versions	All versions	All versions

[b] Function Module: MPLINK (MPLINK/CP-215 communications)

MEMOBUS	Extended MEMOBUS	Non-procedure Protocol
All versions	All versions	All versions

1.2.5 Applicable Versions of MPE720

For MPE720 Ver.5 and Ver.6, the supported software versions may differ according to the protocol.

(1) 217IF-01

<table>
<tr><td colspan="2" rowspan="3">Machine Controller</td><td colspan="7">MPE720</td></tr>
<tr><td colspan="3">Ver. 5</td><td colspan="2">Ver. 6</td><td colspan="2">Ver. 7</td></tr>
<tr><td>Non-procedure protocol FD</td><td>OMRON Host Link</td><td>Other than left</td><td>Non-procedure protocol FD</td><td>Other than left</td><td>Non-procedure protocol FD</td><td>Other than left</td></tr>
<tr><td colspan="2">MP2300</td><td rowspan="10">Not supported</td><td>Ver. 4.52 or later</td><td>Ver. 4.41A or later</td><td rowspan="10">Not supported</td><td>All versions</td><td rowspan="10">Ver. 7.24 or later</td><td rowspan="10">All versions</td></tr>
<tr><td colspan="2">MP2310</td><td colspan="2" rowspan="2">Ver. 5.38 or later</td><td>Ver. 6.04 or later</td></tr>
<tr><td colspan="2">MP2300S</td><td rowspan="4">All versions</td></tr>
<tr><td rowspan="3">MP2200</td><td>CPU-01</td><td colspan="2">Ver. 5.10 or later</td></tr>
<tr><td>CPU-02</td><td colspan="2">Ver. 5.31 or later</td></tr>
<tr><td>CPU-03</td><td colspan="2">Ver. 5.50 or later</td><td>Ver. 6.20 or later</td></tr>
<tr><td>CPU-04</td><td colspan="2">Ver. 5.52 or later</td><td>Ver. 6.22 or later</td></tr>
<tr><td colspan="2">MP2100M</td><td colspan="2">Ver. 5.10 or later</td><td>All versions</td></tr>
<tr><td colspan="2">MP2101M</td><td colspan="2" rowspan="2">Ver. 5.54 or later</td><td rowspan="2">Ver. 6.24 or later</td></tr>
<tr><td colspan="2">MP2101TM</td></tr>
<tr><td rowspan="2">MP3100</td><td>MP3100 (16 axes)</td><td rowspan="7">Not supported</td><td colspan="5" rowspan="7"></td><td colspan="2">Ver. 7.38 or later</td></tr>
<tr><td>MP3100 (32 axes)</td></tr>
<tr><td rowspan="2">MP3200</td><td>CPU-201</td><td rowspan="2">Ver. 7.24 or later</td><td rowspan="2">All versions</td></tr>
<tr><td>CPU-202</td></tr>
<tr><td rowspan="4">MP3300</td><td>CPU-301 (16 axes)</td><td colspan="2">Ver. 7.26 or later</td></tr>
<tr><td>CPU-301 (32 axes)</td><td colspan="2">Ver. 7.28 or later</td></tr>
<tr><td>CPU-302 (16 axes)</td><td colspan="2" rowspan="2">Ver. 7.33 or later</td></tr>
<tr><td>CPU-302 (32 axes)</td></tr>
</table>

(2) 218IF-01, 260IF-01

<table>
<tr><th colspan="2" rowspan="3">Machine Controller</th><th colspan="3">MPE720</th></tr>
<tr><th colspan="2">Ver. 5</th><th rowspan="2">Ver. 6</th><th rowspan="2">Ver. 7</th></tr>
<tr><th>OMRON Host Link</th><th>Other than left</th></tr>
<tr><td colspan="2">MP2300</td><td>Ver. 4.52 or later</td><td>Ver. 4.41A or later</td><td>All versions</td><td rowspan="10">All versions</td></tr>
<tr><td colspan="2">MP2310</td><td colspan="2" rowspan="2">Ver. 5.38 or later</td><td rowspan="2">Ver. 6.04 or later</td></tr>
<tr><td colspan="2">MP2300S</td></tr>
<tr><td rowspan="4">MP2200</td><td>CPU-01</td><td colspan="2">Ver. 5.10 or later</td><td rowspan="2">All versions</td></tr>
<tr><td>CPU-02</td><td colspan="2">Ver. 5.31 or later</td></tr>
<tr><td>CPU-03</td><td colspan="2">Ver. 5.50 or later</td><td>Ver. 6.20 or later</td></tr>
<tr><td>CPU-04</td><td colspan="2">Ver. 5.52 or later</td><td>Ver. 6.22 or later</td></tr>
<tr><td colspan="2">MP2100M</td><td colspan="2">Ver. 5.10 or later</td><td>All versions</td></tr>
<tr><td colspan="2">MP2101M</td><td colspan="2" rowspan="2">Ver. 5.54 or later</td><td rowspan="2">Ver. 6.24 or later</td></tr>
<tr><td colspan="2">MP2101TM</td></tr>
<tr><td rowspan="2">MP3100</td><td>MP3100 (16 axes)</td><td rowspan="8">Not supported</td><td rowspan="8">Not supported</td><td rowspan="8">Not supported</td><td>Ver. 7.38 or later</td></tr>
<tr><td>MP3100 (32 axes)</td></tr>
<tr><td rowspan="2">MP3200</td><td>CPU-201</td><td rowspan="2">All versions</td></tr>
<tr><td>CPU-202</td></tr>
<tr><td rowspan="4">MP3300</td><td>CPU-301 (16 axes)</td><td>Ver. 7.26 or later</td></tr>
<tr><td>CPU-301 (32 axes)</td><td>Ver. 7.28 or later</td></tr>
<tr><td>CPU-302 (16 axes)</td><td rowspan="2">Ver. 7.33 or later</td></tr>
<tr><td>CPU-302 (32 axes)</td></tr>
</table>

(3) 218IF-02

<table>
<tr><th colspan="2" rowspan="3">Machine Controller</th><th colspan="5">MPE720</th></tr>
<tr><th colspan="2">Ver. 5</th><th colspan="2">Ver. 6</th><th rowspan="2">Ver. 7</th></tr>
<tr><th>MELSEC QnA-compatible 3E and OMRON FINS</th><th>Other than left</th><th>MELSEC QnA-compatible 3E and OMRON FINS</th><th>Other than left</th></tr>
<tr><td colspan="2">MP2300</td><td rowspan="8">Ver. 5.52 or later</td><td rowspan="4">Ver. 5.38 or later</td><td rowspan="8">Ver. 6.22 or later</td><td rowspan="4">Ver. 6.04 or later</td><td rowspan="8">All versions</td></tr>
<tr><td colspan="2">MP2310</td></tr>
<tr><td colspan="2">MP2300S</td></tr>
<tr><td rowspan="4">MP2200</td><td>CPU-01</td></tr>
<tr><td>CPU-02</td></tr>
<tr><td>CPU-03</td><td>Ver. 5.50 or later</td><td>Ver. 6.20 or later</td></tr>
<tr><td>CPU-04</td><td>Ver. 5.52 or later</td><td>Ver. 6.22 or later</td></tr>
<tr><td colspan="2">MP2100M</td><td>Ver. 5.38 or later</td><td>Ver. 6.04 or later</td></tr>
<tr><td colspan="2">MP2101M</td><td colspan="2" rowspan="2">Ver. 5.54 or later</td><td colspan="2" rowspan="2">Ver. 6.24 or later</td></tr>
<tr><td colspan="2">MP2101TM</td></tr>
<tr><td rowspan="2">MP3100</td><td>MP3100 (16 axes)</td><td colspan="2" rowspan="7">Not supported</td><td colspan="2" rowspan="7">Not supported</td><td>Ver.7.38 or later</td></tr>
<tr><td>MP3100 (32 axes)</td></tr>
<tr><td rowspan="2">MP3200</td><td>CPU-201</td><td rowspan="2">All versions</td></tr>
<tr><td>CPU-202</td></tr>
<tr><td rowspan="4">MP3300</td><td>CPU-301 (16 axes)</td><td>Ver.7.26 or later</td></tr>
<tr><td>CPU-301 (32 axes)</td><td>Ver.7.28 or later</td></tr>
<tr><td>CPU-302 (16 axes)</td><td rowspan="2">Ver.7.33 or later</td></tr>
<tr><td>CPU-302 (32 axes)</td></tr>
</table>

(4) 215AIF-01

Machine Controller		MPE720		
		Ver. 5	Ver. 6	Ver. 7
MP2300		Ver. 5.31 or later	All versions	All versions
MP2310		Ver. 5.38 or later	Ver. 6.04 or later	
MP2300S				
MP2200	CPU-01	Ver. 5.31 or later	All versions	
	CPU-02			
	CPU-03	Ver. 5.50 or later	Ver. 6.20 or later	
	CPU-04	Ver. 5.52 or later	Ver. 6.22 or later	
MP2100M		Ver. 5.31 or later	All versions	
MP2101M		Ver. 5.54 or later	Ver. 6.24 or later	
MP2101TM				
MP3100	MP3100 (16 axes)	Not supported	Not supported	Ver. 7.38 or later
	MP3100 (32 axes)			
MP3200	CPU-201			All versions
	CPU-202			
MP3300	CPU-301 (16 axes)			Ver. 7.26 or later
	CPU-301 (32 axes)			Ver. 7.28 or later
	CPU-302 (16 axes)			Ver. 7.33 or later
	CPU-302 (32 axes)			

1.3 Mounting and Removing a Module on Machine Controller

This section describes mounting and removing a Communication Module.

1.3.1 Mounting a Communication Module

Use the following procedure to mount a Communication Module.

- When replacing a Communication Module, first refer to *1.3.2 Removing a Communication Module* on page 1-14 and remove the Communication Module that needs to be replaced.

(1) Preparation

1. Backup the Programs.

Save the programs written to the Machine Controller in the personal computer using MPE720.

2. Remove the Machine Controller and Expansion Rack.

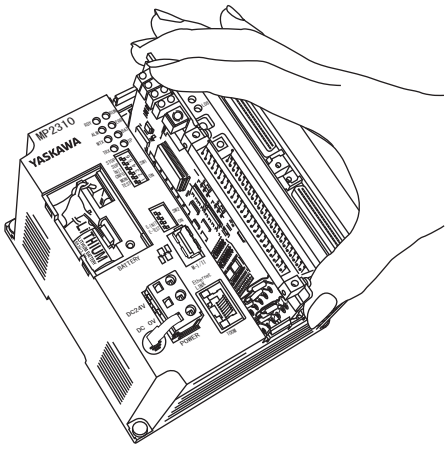
Turn OFF the power supply and remove all the cables connected to the Machine Controller or Expansion Rack (MP2200 Base Unit). Next, remove the Machine Controller and Expansion Rack from the panel or rack, and place them where there is sufficient space, such as on a work table.

(2) Installing Optional Modules

Use the following procedure to install Optional Modules.

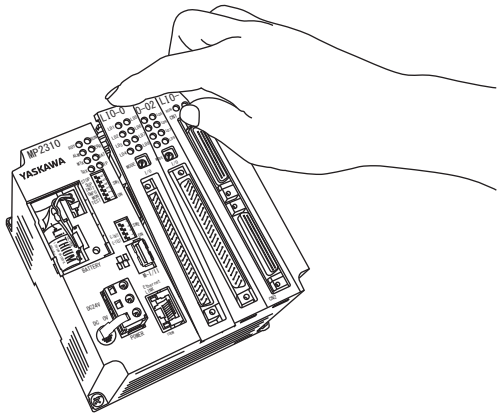
1. Hold the top and bottom of the Optional Module to be installed, line up the Module with the left side of the guide rail inside the option slot, and then insert the Module straight in.

- The FG bar inside and on the bottom may be damaged if the Module is not inserted along the guide rail.



2. After the Optional Module is completely inserted, place your hand on the front of the Optional Module and press the Optional Module firmly until it mates with the Mounting Base connectors in the Unit. The front of the Optional Module and the tabs will be aligned if the Optional Module has been installed properly.

3. Place the hole on the bottom of the panel of the Optional Module onto the tab on the bottom of the Unit. Next, hook the hole at the top of the panel of the Optional Module onto the tab on the Unit.



This completes the installation procedure.

- Always use Option Covers (model: JEPMC-OP2300) to cover unused slots.

1.3.2 Removing a Communication Module

Use the following procedure to remove a Communication Module.

(1) Preparation

1. Backup the Programs.

Save the program written to the Machine Controller in the personal computer using MPE720.

2. Remove the Machine Controller and Expansion Rack.

Turn OFF the power supply and remove the cables connected to the Machine Controller or Expansion Rack. Next, remove the Machine Controller and Expansion Rack from the panel or rack and place them where there is sufficient space, such as on a work table.

(2) Replacing and Adding Optional Modules

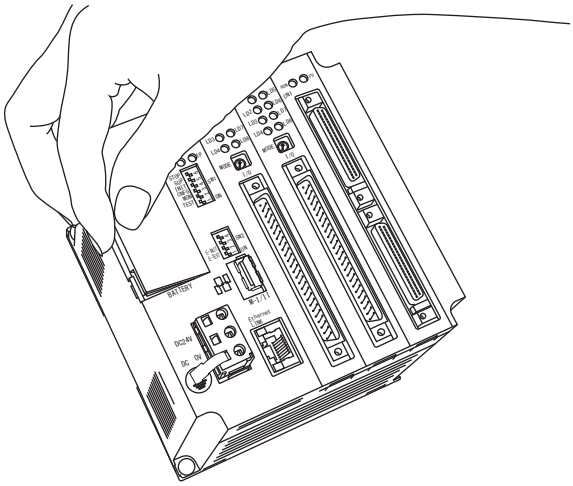
Use the following procedure to replace or add Optional Modules.

- Always create a backup before replacing or adding Optional Modules.
- Back up the program from the Machine Controller to the PC using the MPE720.

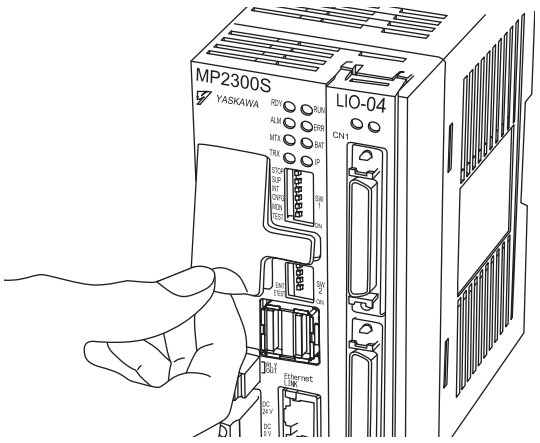
1. Turn OFF the power supply and disconnect all cables from the Machine Controller.

2. Pull the notch on the side toward you to remove the battery cover.

<MP2200/MP2300>

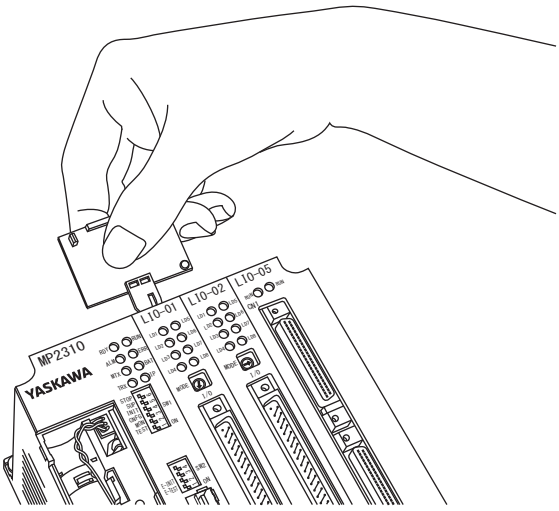


<MP2310/MP2300S>



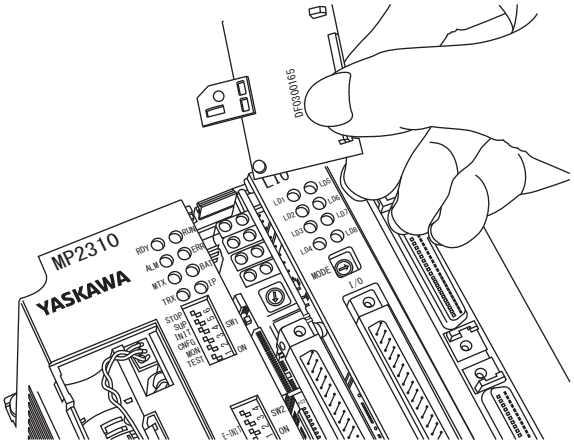
3. Insert the protruding part of the battery cover into the slot on top of the Optional Module panel to unhook the tab. Face the front of the battery cover toward you for this operation.

- Use the same method to remove the Option Cover from an unused slot before adding an Optional Module.

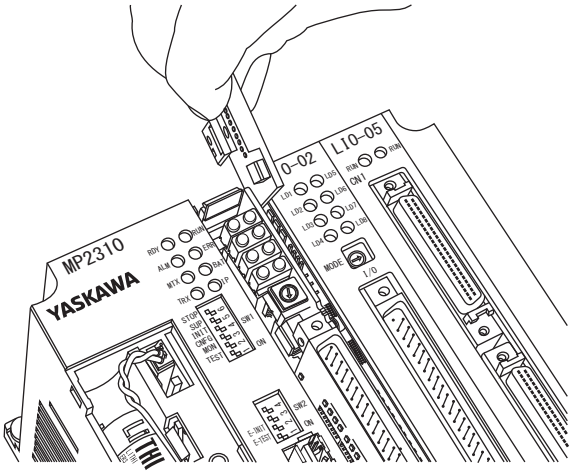


Unhook the bottom tab in the same way.

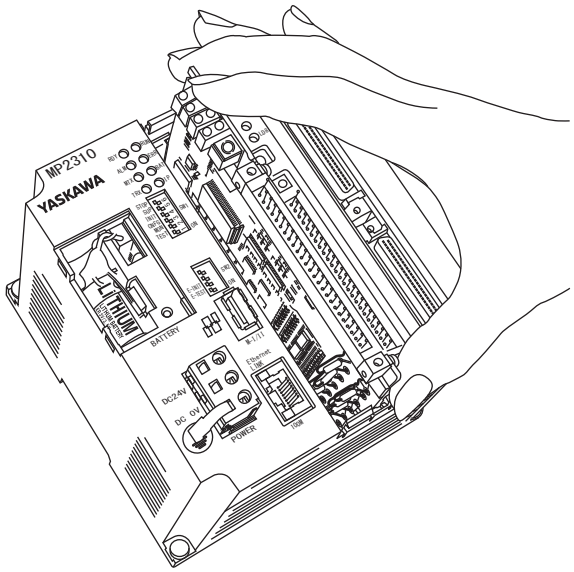
4. Pull the top of the Optional Module panel toward you and remove it. A notch on the Optional Module will be visible from the gap with the panel. Hook the round knob on the battery cover into the notch in the Optional Module.



5. Hold the center of the battery cover, and turn it around the round knob while pushing it toward the back to disconnect the Module from the Mounting Base connectors. Then, pull the Module forward.



6. Hold the Optional Module at the top and bottom and pull it straight out. Hold the edges of the Module and avoid touching the components on the Module.



- Put the Module that you removed into the bag that was supplied when you purchased it and store the Module in this bag.

1.4 Setting the Communication Platform

This section describes the software called the Communication Platform that is used to set the communication method for engineering communication between the personal computer running the MPE720 and the MP2000 Series Machine Controller.
Set the communication conditions with the Communication Platform after the MPE720 Programming Device has been installed. Once they have been set, it is unnecessary to set from the next startup except when other conditions are to be added.

1.4.1 Opening the Communication Platform

1. Click the Windows **Start** button and select ***Programs - YE_Applications - Communication Platform***.
   The **Communication Platform** icon will be displayed in the task tray at the right bottom of the window.

2. Double-click the **Communication Platform** icon.

Communication Platform icon

The **Communication Platform** window will be displayed.
Logical ports for up to 16 channels can be set in this window.

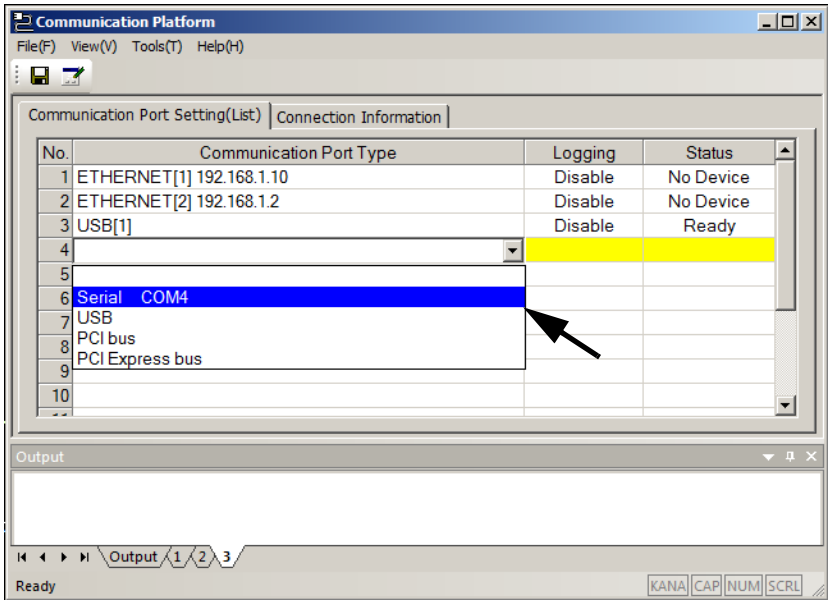
1.4.2 Setting the Communication Port

This section describes the procedure to set the Communication Platform for connecting the MPE720 and MP2000 Series Machine Controller for each type of communication port.

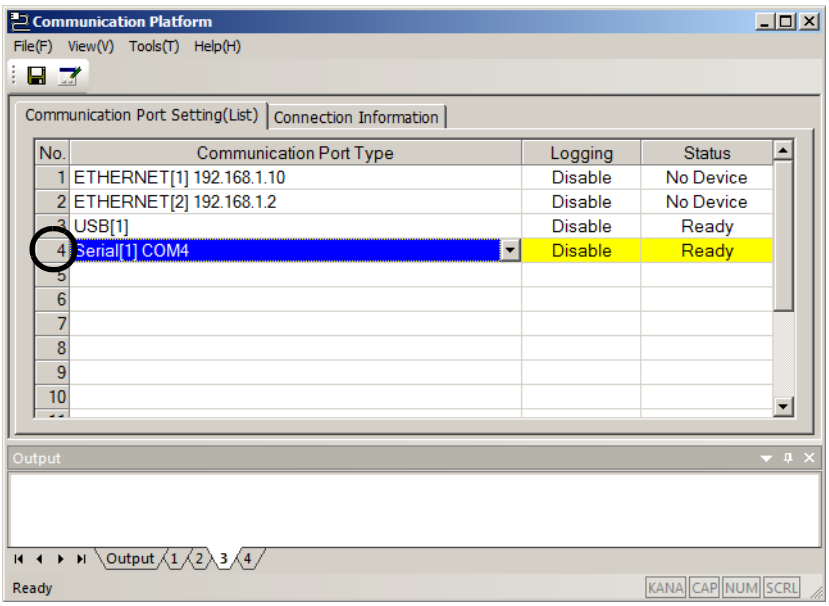
(1) Settings for Serial Communication Port (RS-232C Connection)

These settings are required to perform engineering communication via the serial (RS-232C) port of each Communication Module using the MPE720. Use the following procedure to make the settings.

1. Select **Serial COM4** under **Communication Port Type** in the **Communication Platform** Window.

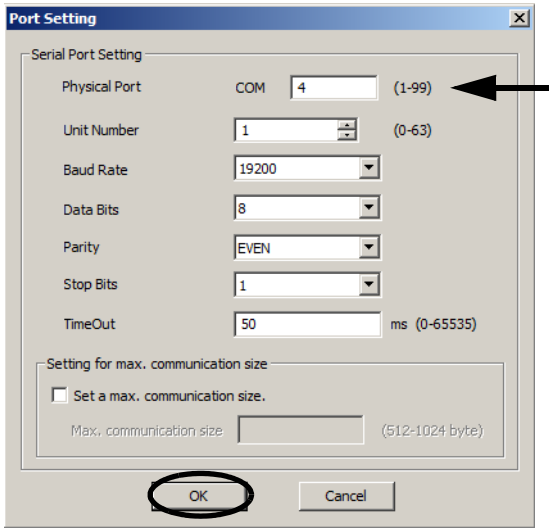


2. Double-click **No.4**.

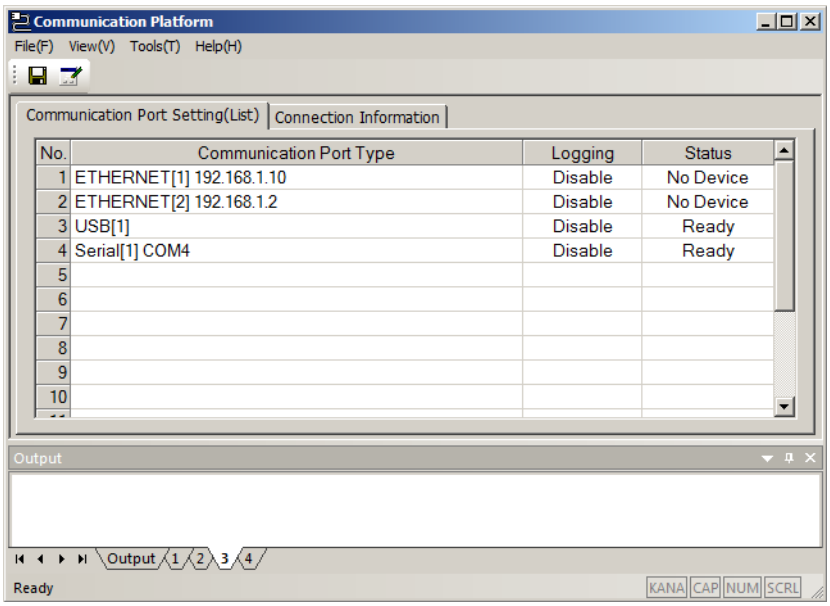


The **Port Setting** Dialog Box will appear.

3. Match the settings under **Physical Port** to the computer's serial communication port. Leave the other items on the default settings. Once the settings have been completed and checked, click the **OK** Button to close the dialog box.



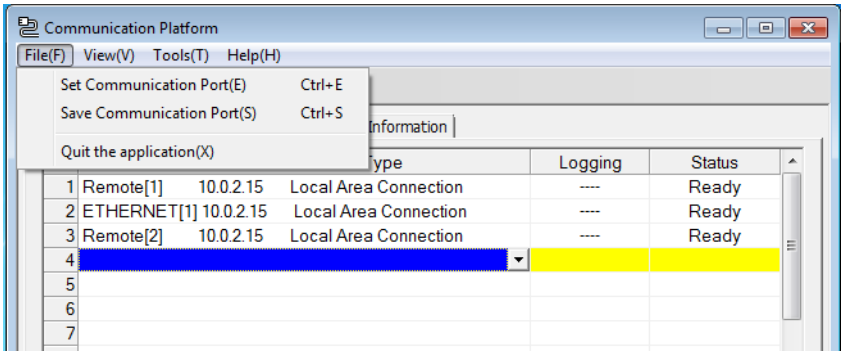
4. The **Logical Port Setting** Window will be displayed. Click the **OK** Button again. The display will return to the **Communication Platform** Window. Check that **Serial** has been assigned to **No.4** on the **Communication Port Setting (List)** Tab Page.



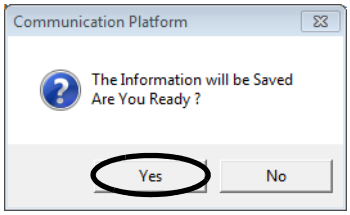
■ Saving the Communication Port Settings and Restarting the Communication Platform

Save the communication port settings, and restart the Communication Platform to validate the settings.

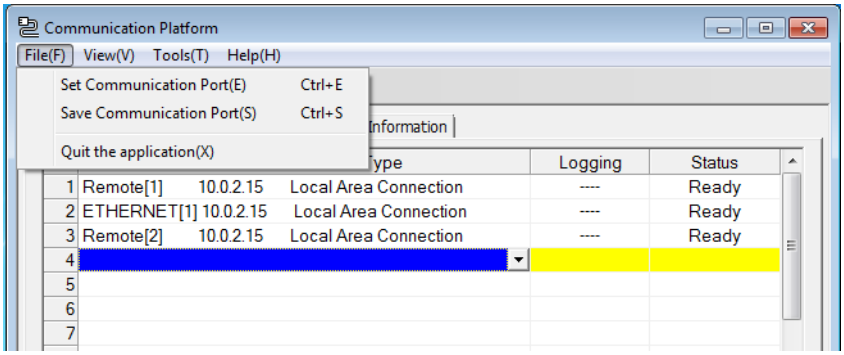
1. Select ***File - Save Communication Port*** in the Communication Platform Window.



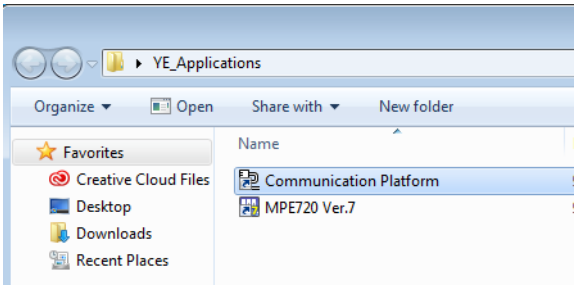
2. A save confirmation window will be displayed. Click the **Yes** Button.



3. Select ***File - Quit the application*** in the Communication Platform Window.



4. Click the **Communication Platform** Icon in the **YE_Applications Folder** to reopen the Communication Platform Window.



(2) Setting the Ethernet Communication Port

These settings are required to perform engineering via the 10Base-T communication port (Ethernet) of the 218IF-01 Module. For Ethernet connection, a general-purpose Ethernet board or PCMCIA Ethernet card must be mounted on the personal computer. Prior to make settings, the IP address of the personal computer must be set.

[a] Mounting an Ethernet Card

Mount a general-purpose Ethernet board or PCMCIA Ethernet card on the specified connector of the personal computer. Also, install the driver provided with the Ethernet card.

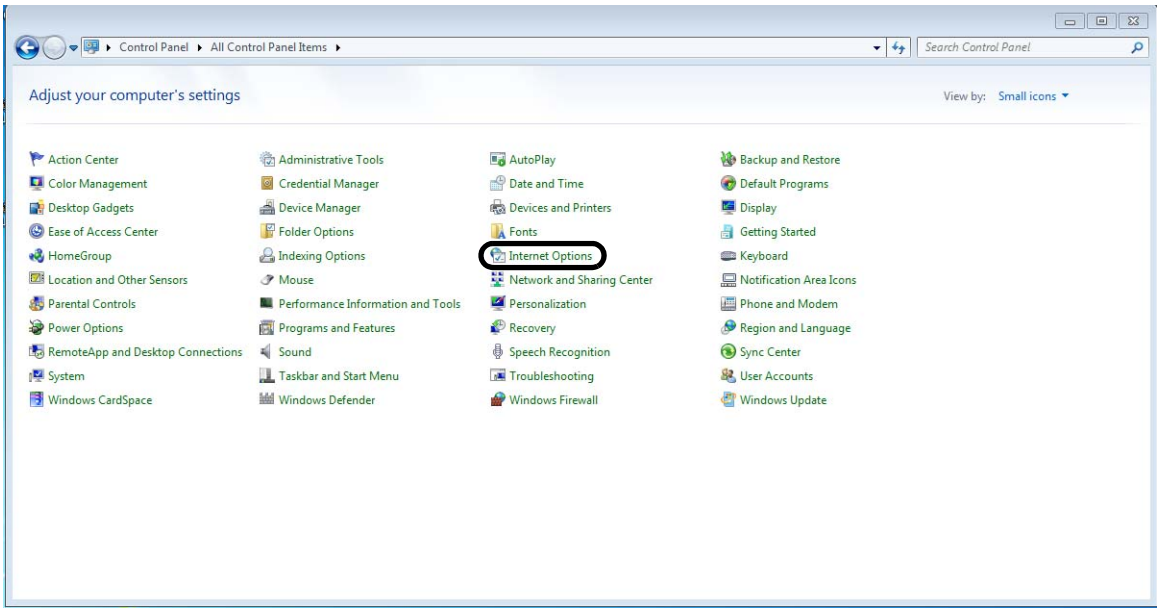
[b] Setting the IP Address

Prior to make settings for Ethernet connection, the IP address of the personal computer must be set. Set the IP address using the following procedure.

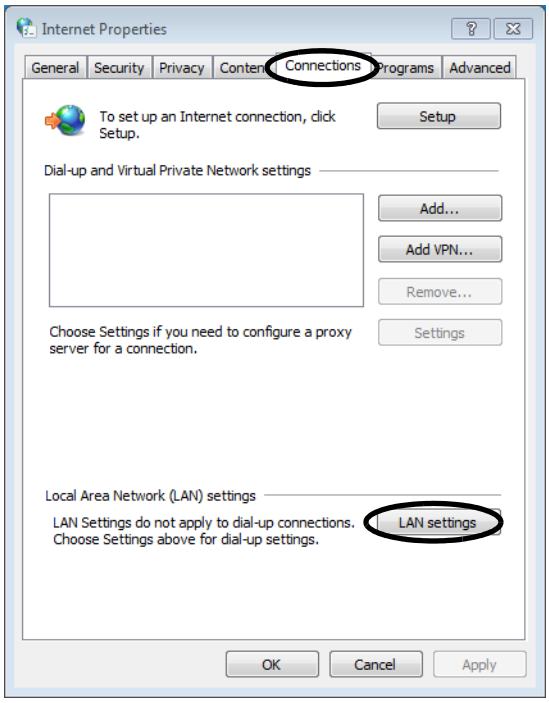
- Make the following settings with the LAN cable connected.

1. Click the Windows **Start** Button and select ***Settings - Control Panel***.

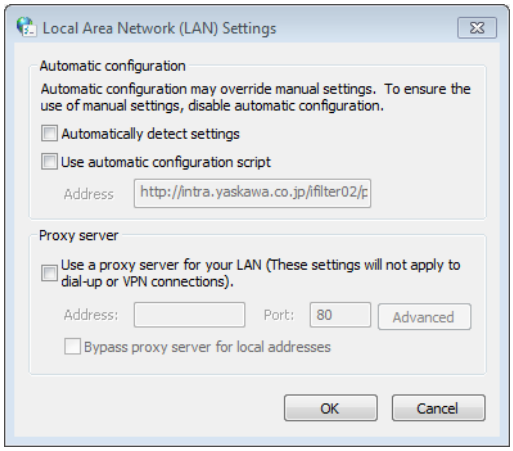
2. Click **Internet Options** in the **All Control Panel Items**.



3. Click **Connections** Tab in the **Internet Properties** Dialog Box to display the tab page. Click the **LAN Settings** Button.

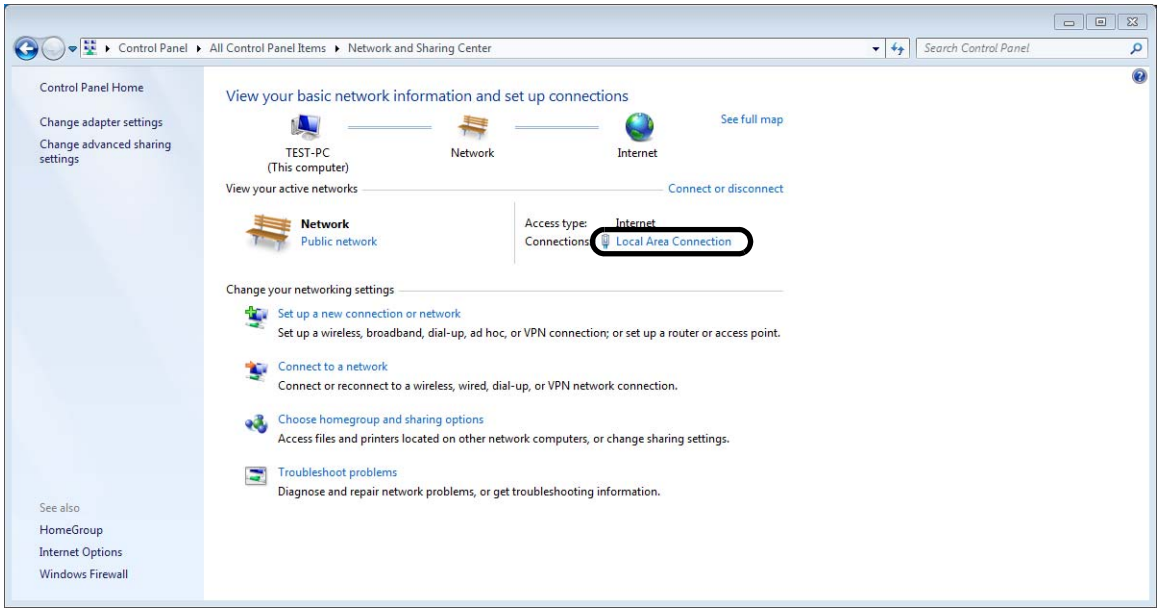


4. Check that the **Automatically detect settings** Check Box is cleared, and click the **OK** Button to close the dialog box.

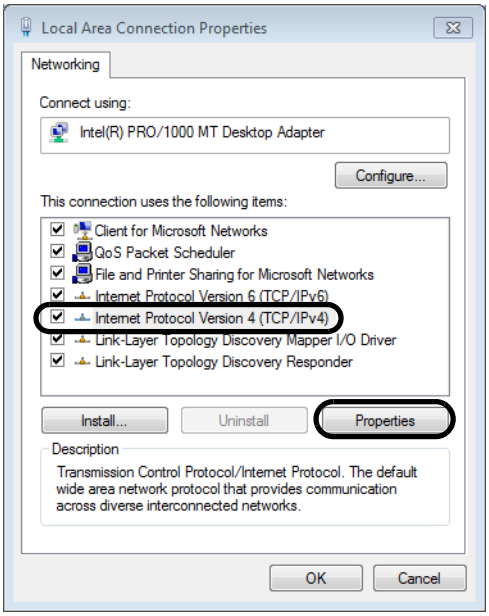


5. Click **Network and Sharing Center** in the **All Control Panel Items**.

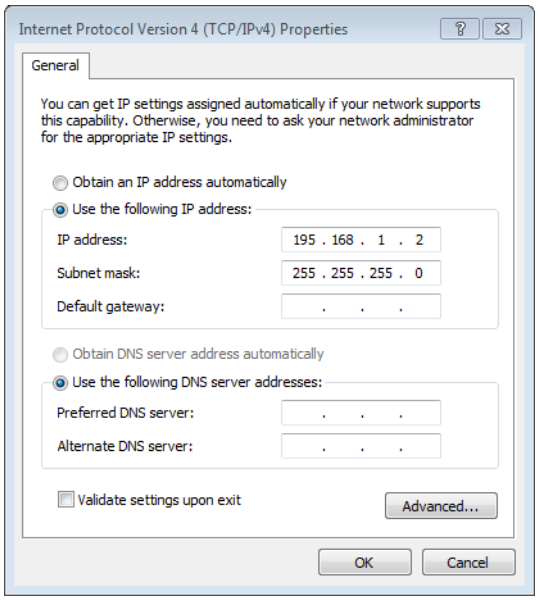
6. Click **Local Area Connection**.



7. Select **Internet Protocol Version 4 (TCP/IPv4)** in the **Local Area Connection Properties** Dialog Box and click the **Properties** Button.



8. Click **Use the following IP address** Option and enter 192 168 1 2 for **IP address** and 255 255 255 0 for **Subnet mask**.

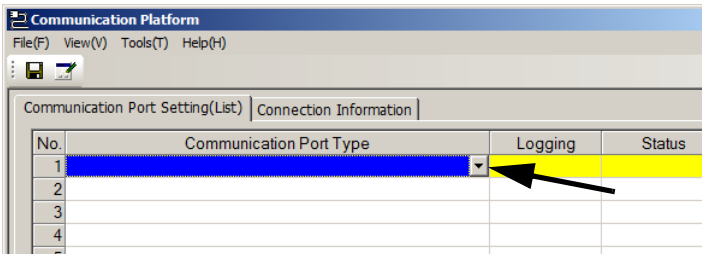


9. Click the **OK** Button to close the dialog box.

This completes the procedure for setting IP address.

[c] Setting the Ethernet Communication Port

1. In the **Communication Platform** Window, click the [▼] Button in the Communication Port Type column on a line which has not been set, and select the communication port type from the options.



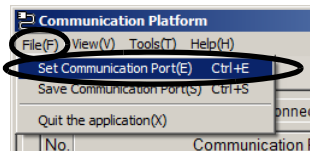
- The connector on the Machine Controller side can also be connected to a computer (MPE720) using an Ethernet communication port other than that on the 263IF-01 Module. Select the communication port type according to the connector on the Machine Controller side that will be used.

Connector Used on Machine Controller Side	Communication Port Type
Ethernet communication connector other than that on the 218IF-01 Module	Select one of the following options. • Ethernet (LP)*1 • Ethernet*1, *2 • CP-218*2
Ethernet communication connector on the 218IF-01 Module	Select one of the following options. • Ethernet*2 • CP-218*2

∗ 1. High-speed engineering communication is possible with the Ethernet (LP) communication port because the engineering message size has been expanded when compared to the Ethernet communication port. For this reason, selecting **Ethernet (LP)** is recommended.

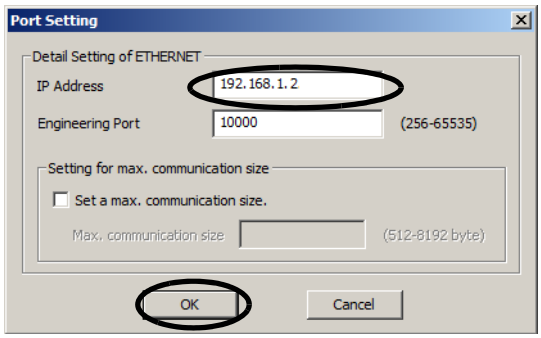
∗ 2. The specifications for **Ethernet** and **CP-218** are the same. No matter which option is selected, there are no differences in the communication specifications.

2. Confirm that the line set in step 1 is highlighted, and then click ***File - Set Communication Port***.

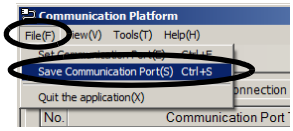


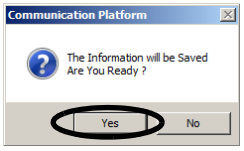
The **Port Setting** Dialog Box will be displayed.

3. For **IP Address**, enter the IP address of the computer, and then click the **OK** Button to close the dialog box.



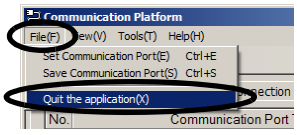
4. Click ***File - Save Communication Port***. A confirmation dialog box will be displayed. Click the **Yes** Button to save the communication port settings.

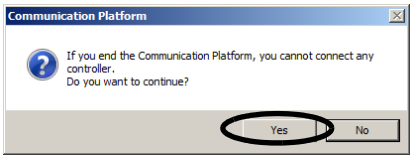




These settings will be used as the communication port information whenever the Communication Platform is started.

5. Click ***File - Quit the Application*** to close the **Communication Platform** Window. A confirmation dialog box will be displayed. Click the **Yes** Button to close the **Communication Platform** Window.





1.5 Executing Self-configuration

The self-configuration function automatically detects the Option Modules connected to the Machine Controller and automatically generates the files for the Module configuration definitions, MECHATROLINK transmission definitions, and SVB definitions. Executing self-configuration will greatly reduce the system startup procedure.
The methods used to execute self-configuration are described below.

■ Setting the CNFG DIP Switch Pin and Cycling Power

Self-configuration can be executed by turning ON the CNFG DIP switch pin on the Machine Controller and turning the power OFF and then ON again. The result will depend on the setting of the INIT DIP switch pin.

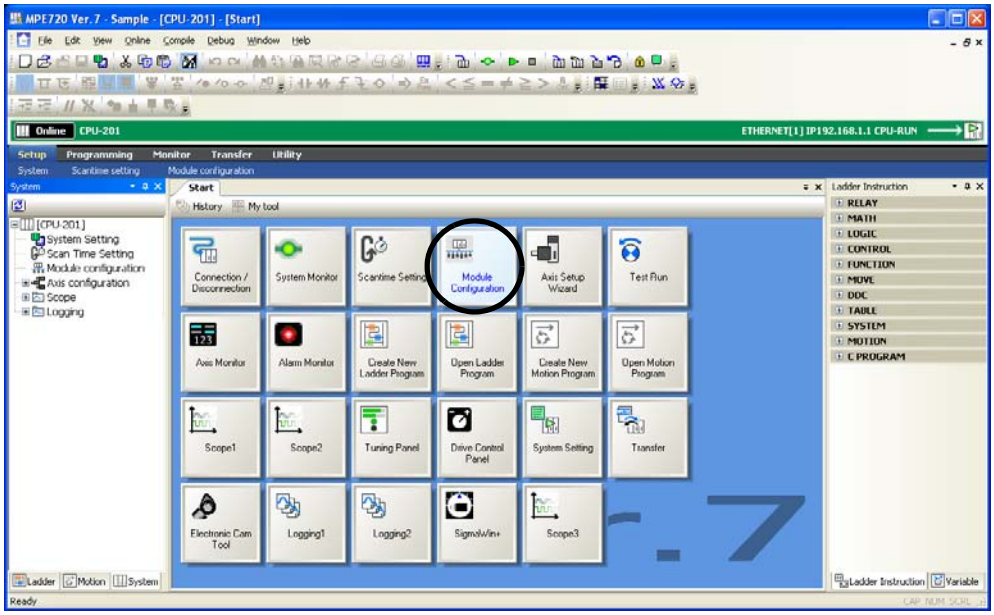
CNFG	INIT	Result
ON	ON	• The Module configuration definitions are updated. • The default assignments are made for all of the Communication Modules that are detected.
ON	OFF	• The Module configuration definitions are updated. • The definitions for any Modules for which there are already transmission definitions are not changed. • The default values are assigned in the transmission definitions for any new Modules that are detected.

After executing self-configuration, always save data to flash memory so that the results of self-configuration are saved to the Machine Controller.

- The DIP switch is not normally used for the MP2100M. For these Machine Controllers, use the MPE720 as described next.

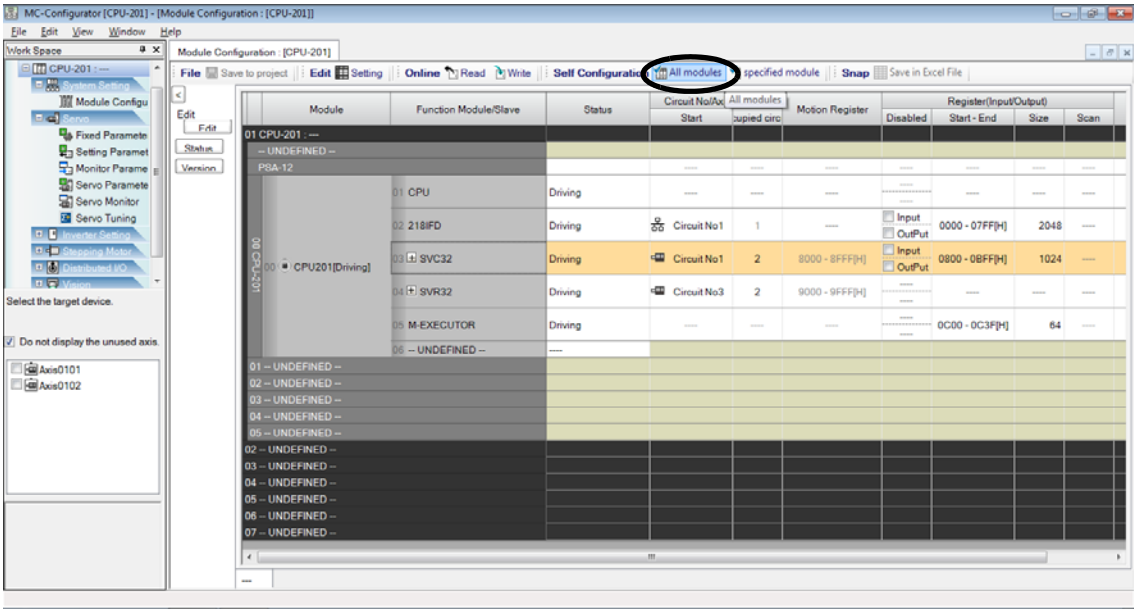
■ Using the MPE720

1. Click the **Module Configuration** Button on the My Tool View.



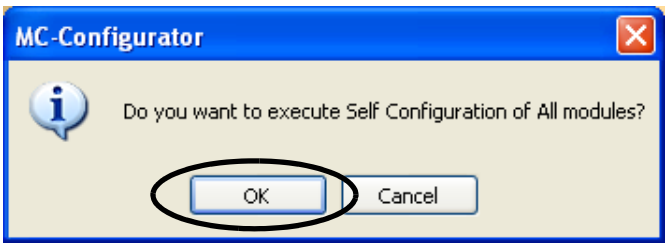
The Module Configuration Tab Page will be displayed.

2. Click the **All modules** Button.



The MC-Configurator Dialog Box will be displayed.

3. Click the **OK** Button.



Self configuration for all Modules will be executed.

Command	Result
Self-configuration for all Modules	• The Module configuration definitions are updated. • The definitions for any Modules for which there are already transmission definitions are not changed. • The default values are assigned in the transmission definitions for any new Modules that are detected.
Module Self-configuration	• Transmission definitions are assigned only for the selected Module. • The definitions for any Modules for which there are already transmission definitions are not changed. • The default values are assigned in the transmission definitions for any new Modules that are detected.

1.6 Starting the MPE720, and Setting Communication or Network Parameters

This section describes how to start the MPE720 and set the communication parameters in the project file.

1.6.1 Starting MPE720 Ver. 7 and Setting Communication

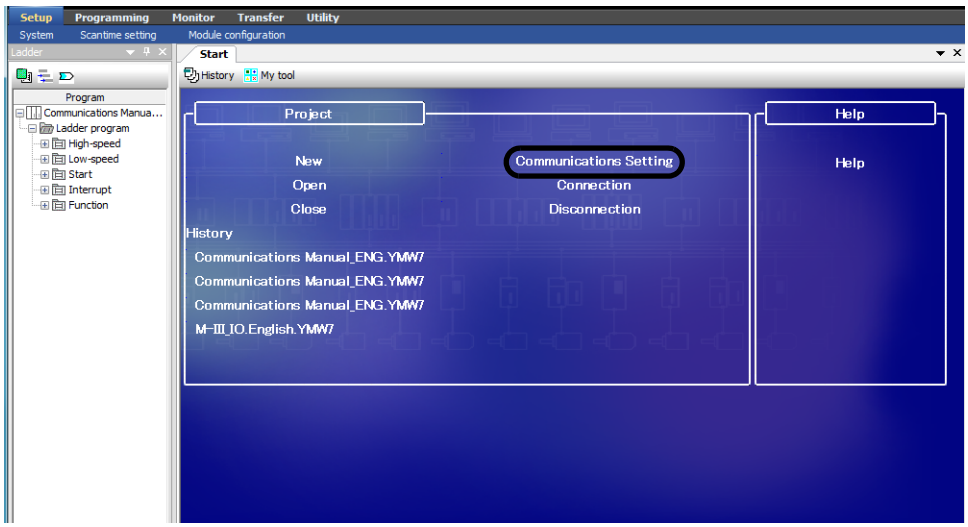
This section describes how to start the MPE720 Ver.7 and set the communication parameters in the project file.

- Disconnect the Machine Controller when setting the parameters.
- Create the project file in advance.

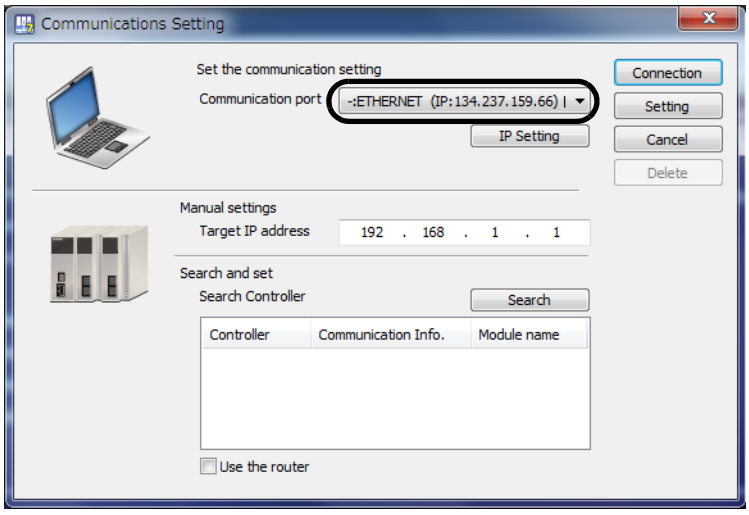
1. Click the **MPE720** Icon.

The MPE720 will start.

2. Click the **Communications Setting** Button on the My Tool View.



3. Select the port number of your PC under **Communication port** in the **Communication Setting** Dialog Box.



1.7 Communication Module Transmission Definitions

This section describes how to open the **Module Configuration** Window from the MPE720 and set the transmission definitions for the Communication Module.

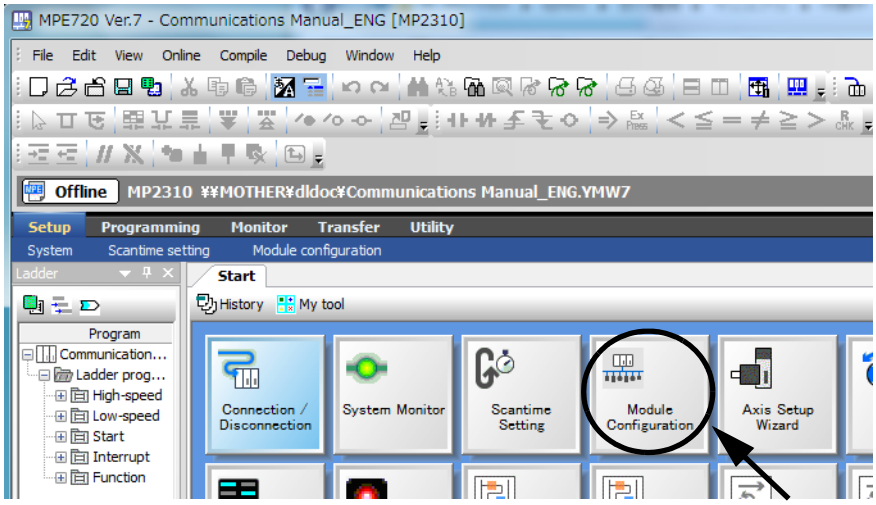
1.7.1 Displaying the Module Configuration Window

Use the following procedure to display the **Module Configuration** Window.

1. Start the MPE720 on the personal computer connected to the Machine Controller and open the project file.

- For information on starting the MPE720, refer to *Machine Controller MP2000/MP3000 Series Engineering Tool MPE720 Version 7 User's Manual (Manual No.: SIEP C880761 03)*.

2. Click the **Module Configuration** Button on My Tool View.

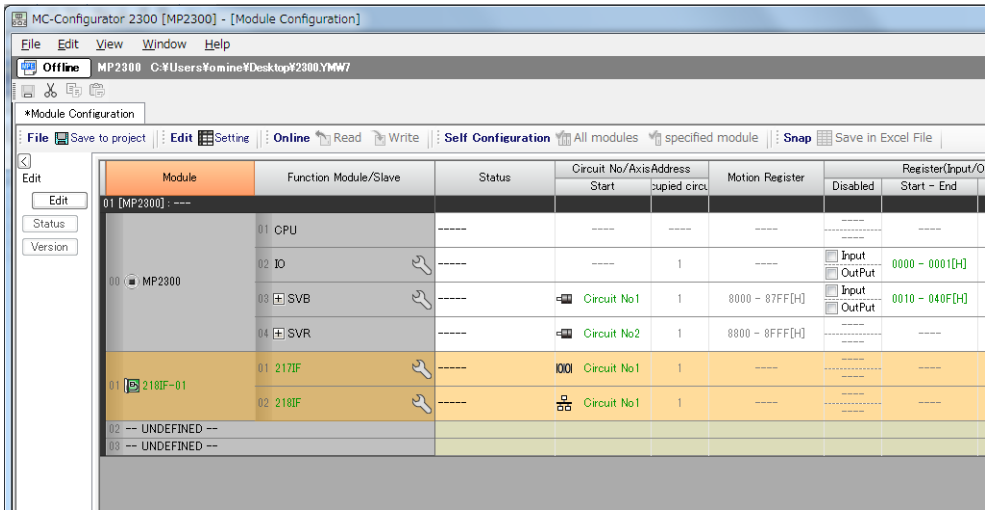


The **Module Configuration** Window will be displayed.

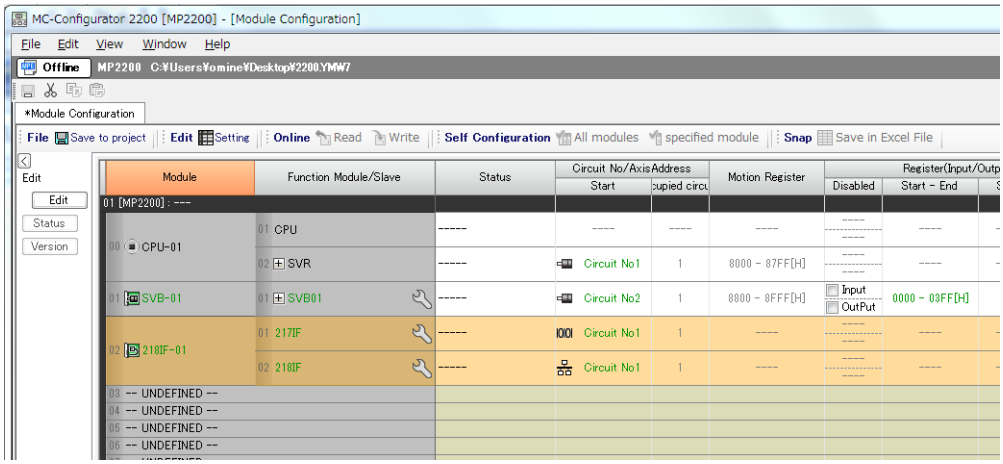
1.7.2 Module Configuration Window

As the result of executing self-configuration, all Option Modules connected to the Machine Controller will be displayed in the **Controller** Area of the **Module Configuration** Window.
If a Communication Module is selected in the **Controller** Area, configuration information on the communication ports on the selected Communication Module will be displayed in the **Module Details** Area.
The following example shows the **Module Configuration** Window when a 218IF-01 Module is connected.

<MP2300/MP2300S/MP2310>



<MP2100M/MP2200>



- Self-configuration can be executed from the **Module Configuration** Window in Online Mode. Refer to *1.5 Executing Self-configuration* on page 1-26 for details.
- Communication Module can be manually assigned to the **Module Type** Column of any slot in the **Controller** Area.
- After executing self-configuration or after manually allocating a Module, save the data to flash memory.

1.7.3 Transmission Definitions

To set the transmission definitions, the **Transmission Configuration** Window must be displayed from the **Module Configuration** Window.
Refer to the descriptions for the following Communication Module for information on displaying the **Transmission Configuration** Window and setting the transmission definitions.

- 218IF-01: *4.4 218IF Transmission Definitions* on page 4-11
- 218IF-02: *5.4 218IFB Transmission Definitions* on page 5-13
- 217IF-01: *3.3 217IF Transmission Definitions* on page 3-14
- 260IF-01: *7.4 260IF Transmission Definitions* on page 7-17
- 261IF-01: *8.4 261IF Transmission Definitions* on page 8-10
- 215AIF-01: *9.4 MPLINK Transmission Definitions* on page 9-16

2

Communication Modes and Protocols

This chapter describes the communication modes and protocols that can be used for each Communication Module.

2.1 Communication Modes ----------2-2
2.1.1 Overview ---------- 2-2
2.1.2 Message Communication ---------- 2-3
2.1.3 Engineering Communication ---------- 2-5
2.1.4 Link Communication ---------- 2-5
2.2 Message Communication Protocols ----------2-6
2.2.1 Types of Communication Protocol ---------- 2-6
2.2.2 MEMOBUS Protocol ---------- 2-6
2.2.3 Extended MEMOBUS Protocol ---------- 2-7
2.2.4 MELSEC Protocol ---------- 2-8
2.2.5 OMRON Protocol ---------- 2-9
2.2.6 MODBUS/TCP Protocol ---------- 2-9
2.2.7 Non-Procedure Protocol ---------- 2-10

2.1 Communication Modes

2.1.1 Overview

There are basically three communication modes in the Communication Modules for an MP2000 Series Machine Controller. The communication modes that can be used depend on the type of communication.

- Message communication: Data is transmitted by user programs using the protocols in the following table.
- Engineering communication: Communication is performed with the MPE720 Programming device using panel commands.
- Link communication and cycllic communication: Predefined areas are transferred between Machine Controllers at regular intervals.

Communication Mode	Transmission Method	Protocol	Yes: Supported, No: Not Supported							
			RS-232C (217IF-01)	RS-232C (other than 217F-01)	RS422/485 (217IF-01)	Ethernet (218IF-01)	Ethernet (218IF-02)	DeviceNet (260IF-01)	PROFIBUS (261IF-01)	MPLINK/CP-215 (215AIF-01)
Message Communication	MEMO-BUS	MEMOBUS	Yes	Yes	Yes	Yes	Yes	No	No	Yes
		Extended MEMOBUS	No	No	No	Yes	Yes	No	No	Yes
		MELSEC A-compatible 1C	Yes	Yes	Yes	No	No	No	No	No
		MELSEC A-compatible 1E	No	No	No	Yes	Yes	No	No	No
		MELSEC QnA-compatible 3E*1	No	No	No	No	Yes	No	No	No
		MODBUS/TCP	No	No	No	Yes	Yes	No	No	No
		OMRON Host Link	Yes	Yes	Yes	No	No	No	No	No
		OMRON FINS*1	No	No	No	No	Yes	No	No	No
		TOYOPUC File memory system*1	No	No	No	No	Yes	No	No	No
		Explicit	No	No	No	No	No	Yes*2	No	No
	General-purpuse	Non-procedure protocol	Yes	Yes	Yes	Yes	Yes	No	No	Yes
		Non-procedure protocol FD	Yes	No	Yes	No	No	No	No	No
Engineering Communication	–		Yes	Yes	No	Yes	Yes	No	No	Yes
Link Communication and Cyclic Communication	–		No	No	No	No	No	Yes	Yes	Yes

* 1. Information on the relevant protocols is not contained in this manual. Refer to the *MP2300S/MP2310/MP2400 Basic Module Supplement for Ethernet Communications* (Manual No.: SIEP C880700 37) for details.

* 2. Supported only for the Master.

2.1.2 Message Communication

The message communication function sends and receives messages between Machine Controllers one event at a time using the MESSAGE SEND function (MSG-SND) and the MESSAGE RECEIVE function (MSG-RCV). The following two message communication modes are supported:

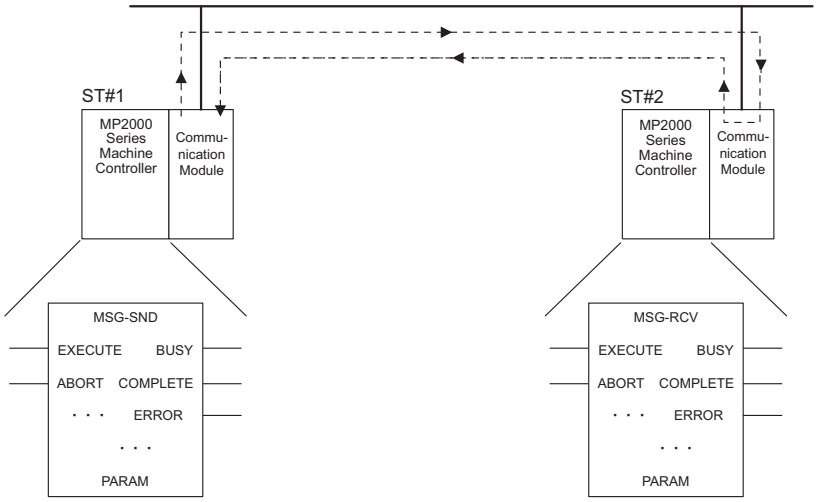
- MEMOBUS Message Communication Mode
- General-purpose Message Communication Mode

(1) MEMOBUS Message Communication Mode

MEMOBUS messages can be sent and received by user programs. A variety of communication protocols are provided to send and receive MEMOBUS messages.

◆ Refer to *2.2 Message Communication Protocols* on page 2-6 for details on each communication protocol.

The Master sends MEMOBUS messages to a Slave using the MSG-SND function, and the Slave returns response messages using the MSG-RCV function. The flow of MEMOBUS message communication is shown below.

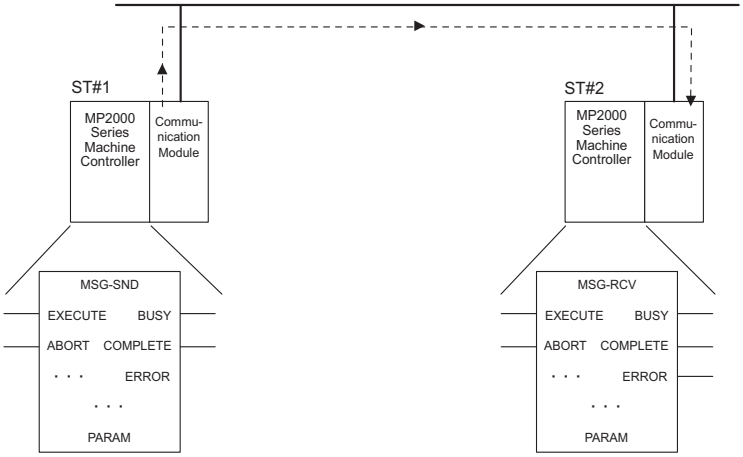


The message is sent from ST#1 to ST#2 via the line using the MSG-SND function.
At ST#2, the MSG-RCV function saves the data into the specified register and returns a response to ST#1.
The results of processing can be checked by reading PARAM (the parameter area) of the MSG-SND function.

◆ For details on the MSG-SND and MSG-RCV functions, refer to *Message Send Function* and *Message Receive Function* in chapters 5 to 9.

(2) General-purpose Message Communication Mode (Non-procedure)

The general-purpose message communication mode sends and receives data in consecutive M registers without processing them, using the MSG-SND and MSG-RCV functions in user programs. No predetermined protocol is used for non-procedure message communication.
An application-specific protocol can be established by sending and receiving the packet data using consecutive M registers.



Data is transferred with the non-procedure protocol via the line from ST#1 to ST#2 using the MSG-SND function.
ST#2 receives the data sent from ST#1 using the MSG-RCV function. No response is returned to ST#1.

■ Advantages of Using the MSG-RCV Function But Not the Automatic Reception Function

- The offset and write area can be changed in the ladder program.
- The results of communication processing and the status of communication can be monitored during debugging.
- Messages can be received faster than in low-speed scan processing.

If, however, the MSG-RCV function is executed in the ladder program with the automatic reception function disabled, memory area is used unnecessarily, lengthening execution time.

(3) Automatic Reception

[a] Overview

The automatic reception function enables message communication even if the MESSAGE RECEIVE function (MSG-RCV function) does not exist in the ladder program when a Communication Module of the MP2000 Series Machine Controller is used as a Slave. Communication Modules that can support the automatic reception function are listed in the following table.

Module Name	Communication Port	Yes: Supported, No: Not Supported
218IF-01	PORT (RS-232C)	Yes
	10Base-T	No
218IF-02	PORT (RS-232C)	Yes
	Ethernet	No
217IF-01	PORT (RS-232C)	Yes
	RS422/485	Yes
260IF-01	PORT (RS-232C)	Yes
	DeviceNet	No
261IF-01	PORT (RS-232C)	Yes
	PROFIBUS	No
215AIF-01	CN1 (CP-215)	No
	CN (RS-232C)	Yes

- The default setting at the ports that support the automatic reception function is to enable automatic reception.
- For the setting to enable or disable the automatic reception function, refer to Transmission Settings in each chapter.
- The automatic reception function cannot be used for the non-procedure communication protocol.

[b] Automatic Reception Execution Timing

The automatic reception function is executed by interrupting low-speed scan processing at fixed intervals of 50 ms. For details on scan execution processing, refer to the manual for your machine controller.

- If the MSG-RCV function is executed in the ladder program for the port for which the automatic reception function is enabled, the MSG-RCV function will be executed together with automatic reception, causing an error in receiving the message.

2.1.3 Engineering Communication

■ Overview

The engineering communication protocol is used for communication between the MPE720 Programming Device and a Machine Controller. When a Communication Module that supports the engineering communication mode is installed at a Machine Controller, engineering is enabled by directly connecting the MPE720 to the Machine Controller.
The following table shows the channels that can be used as the engineering port.

Module Name	Communication Port	Yes: Supported, No: Not Supported
218IF-01	PORT (RS-232C)	Yes
	10Base-T	Yes
218IF-02	PORT (RS-232C)	Yes
	Ethernet	Yes
217IF-01	PORT (RS-232C)	Yes
	RS422/485	No
260IF-01	PORT (RS-232C)	Yes
	DeviceNet	No
261IF-01	PORT (RS-232C)	Yes
	PROFIBUS	No
215AIF-01	CN1 (MPLINK/CP-215)	Yes
	CN2 (RS-232C)	Yes

2.1.4 Link Communication

Link communication mode automatically transfers I/O data (such as relays, coils, and registers) between Machine Controllers. It operates only between 215AIF-01 Modules. Links cannot be used with 217IF-01, 218IF-01, 218IF-02, 260IF-01, or 261IF-01 Modules.
Machine Controllers send and receive data at regular intervals by a transmission mechanism called token passing according to the link assignment map that has been set in advance.
Refer to *9.5.1 Link Communication* on page 9-25 for details on link communication. Also, refer to *9.5.4 Token Passing* on page 9-32 for details on token passing.

2.2 Message Communication Protocols

This section describes the communication protocols used for message communication.

2.2.1 Types of Communication Protocol

The communication protocols supported by the MP2000 Series Machine Controller are listed in the following table. The protocols that can actually be used depend on the Communication Module.

Communication Protocol	Overview	Yes: Supported, No: Not Supported				
		217IF	218IF	218IFB	260IF	CP-215/ MPLINK
MEMOBUS	Yaskawa's standard protocol	Yes	Yes	Yes	No	No
Extended MEMOBUS	Protocol with the Extended MEMOBUS protocol functions	No	Yes	Yes	No	Yes
MELSEC	Protocol used for connecting a controller manufactured by Mitsubishi Electric Corporation	Yes	Yes	Yes	No	No
MODBUS/TCP	Ethernet protocol, proposed by Modicon, used in industrial applications	No	Yes	Yes	No	No
OMRON	Protocol used for connecting a controller manufactured by OMRON Corporation	Yes	No	Yes	No	No
TOYOPUC	Protocol used for connecting a controller manufactured by JTEKT Corporation	No	No	Yes	No	No
Explicit	Protocol used for communication between DeviceNet devices	No	No	No	Yes*	No
Non-Procedure Protocol	Protocol provided to implement general-purpose message communication	Yes	Yes	Yes	×	Yes

* Supported only for the Master.

2.2.2 MEMOBUS Protocol

The MEMOBUS protocol is the standard protocol used up to now for Yaskawa RS-232C communication, and it is the core communication protocols for Yaskawa Controllers.
The Master sends an MEMOBUS-format instruction to a Slave, and the Slave returns a response message when it has received the instruction.

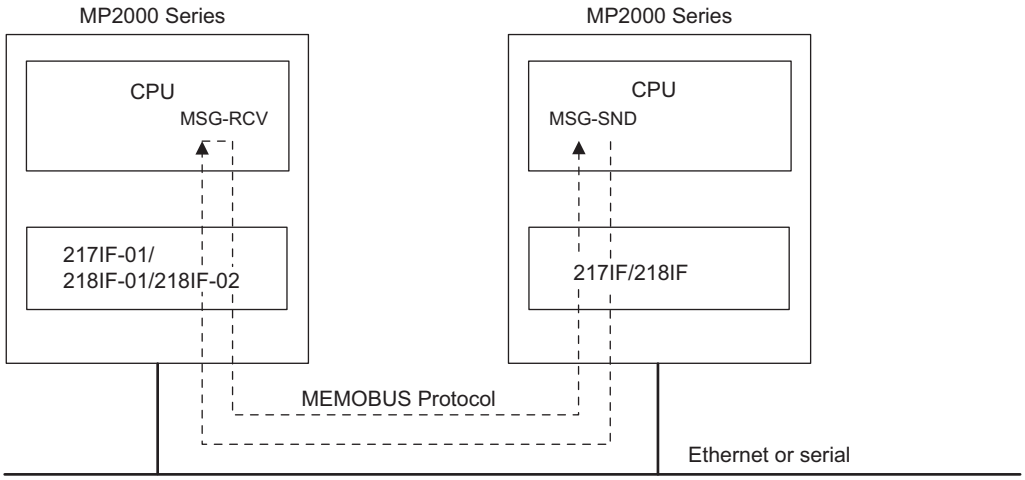


Fig. 2.1 Message Flow between MP-series Machine Controllers Using MEMOBUS Protocol

2.2.3 Extended MEMOBUS Protocol

The Extended MEMOBUS protocol is used for transmissions between Machine Controllers equipped with a 215IF, 218IF, or 218IFB Module.
With more extensive functions than the conventional MEMOBUS protocol used in RS-232C communication, the Extended MEMOBUS protocol provides more versatile functions and higher efficiency in communication.
The Extended MEMOBUS protocol supports the MEMOBUS message communication mode and the general-purpose message communication mode. The mode that is used is specified in a parameter in the MSG-SND and MSG-RCV functions.

(1) Transferring Data in the MEMOBUS Mode

The Extended MEMOBUS protocol is used for communication between the Yaskawa's MP-series and CP-series Machine Controllers (MP2100M, MP2200, MP2300, MP2300S, MP2310, and MP2400).
The Master sends an Extended MEMOBUS-format instruction to a Slave, and the Slave returns a response message when it has received the instruction.

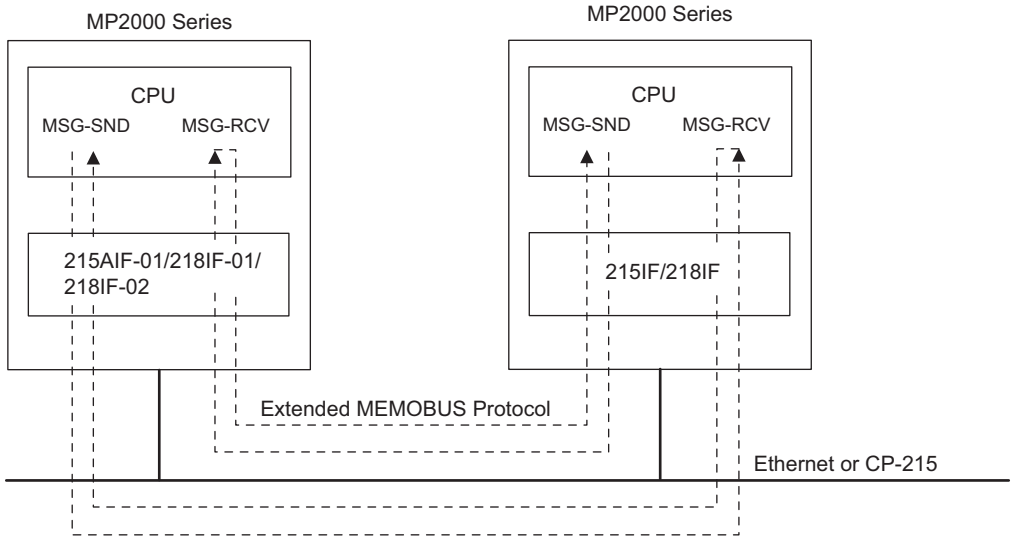
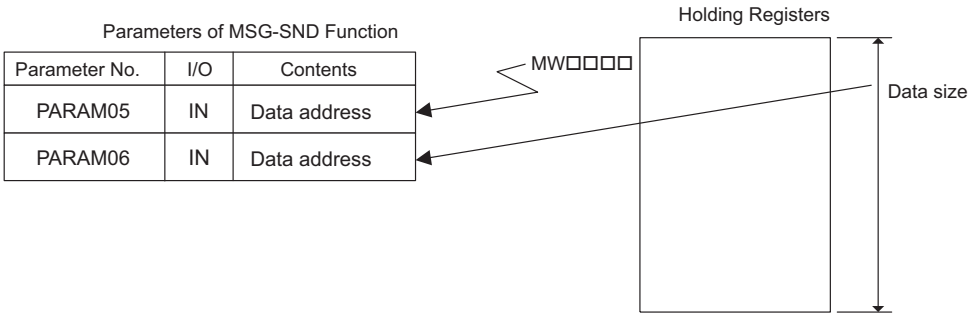


Fig. 2.2 Message Flow between MP-series Machine Controllers Using Extended MEMOBUS Protocol

(2) Transferring Data in the General-purpose Message Communication Mode (Only for Ethernet Communication of 218IF-01, 218IF-02)

In the general-purpose message communication mode, the contents of holding registers (M registers) of a Machine Controller can be sent or received.
As shown in the following diagram, the specified size of the data set in the consecutive holding registers in the Machine Controller is sent starting from the specified address without processing. The device that sends a message in this mode does not wait for a response.
The data size that can be handled in a single communication is from 1 to 510 words.



2.2.4 MELSEC Protocol

The MELSEC protocol is used for communication with Mitsubishi Electric's MELSEC A-series General-purpose Programmable Logic Controllers. There is no need to be aware of the communication protocol. To use the MELSEC protocol, a special-purpose Ethernet Module and a Computer Link Unit are necessary at the MELSEC A-series Programmable Logic Controller.
The MELSEC protocol is used to communicate between the MP2000 Series Machine Controller and MELSEC A-series Programmable Logic Controller. The Master sends a MELSEC-format message to the Slave and receives a response message from the Slave. In communication with a MELSEC A-series Programmable Logic Controller, the MSG-SND and MSG-RCV functions can be used in user programs in the same way as with the MEMOBUS protocol. The protocol is automatically converted inside the Communication Module.

■ Communicating with the MELSEC Protocol Using Serial Communication

Transmissions using the 217IF transmission system supports the special protocol type 1 functions of the MELSEC A-series Programmable Logic Controllers. Here, the MP2000 Series Machine Controller operates as the Master. A subset of the Common MELSEC ACPU commands is used.

• Refer to *3.2.2 MELSEC Protocol* on page 3-5 for details on the MELSEC protocol with serial communication.

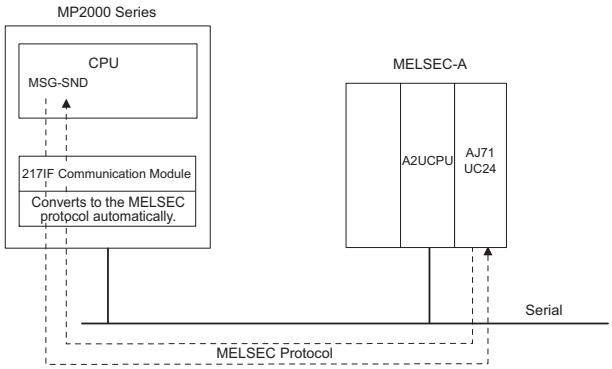


Fig. 2.3 Message Flow between MP2000 Series Machine Controller and MELSEC A-series PLC Using MELSEC Protocol with Serial Communication

■ Communicating with the MELSEC Protocol Using Ethernet Transmission

The Ethernet (218IF and 218IFB) transmission system supports the following three transmission functions of the MELSEC A-series Programmable Logic Controllers.

- Communication between CPUs
- Fixed buffer communication
- Random-access buffer communication

In fixed buffer communication, the MP2000 Series Machine Controller can be either the Master or a Slave. In communication between CPUs or random-access buffer communication, the MP2000 Series Machine Controller can only be the Master due to restrictions of the specifications of MELSEC A-series Programmable Logic Controllers.
A subset of the Common MELSEC ACPU commands is used between the CPUs.

• Refer to *4.5.5 MELSEC Protocol* on page 4-27 for details on the MELSEC protocol with Ethernet transmission.

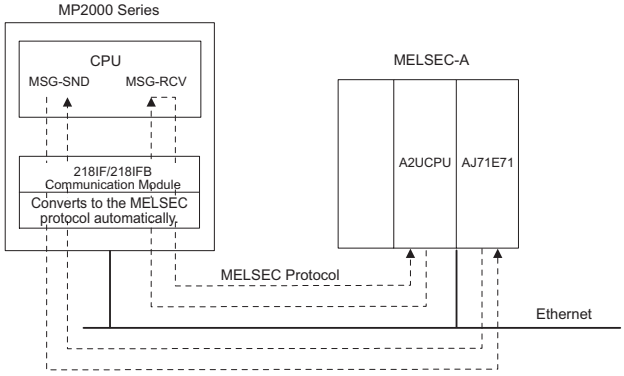


Fig. 2.4 Message Flow between MP2000 Series Machine Controller and MELSEC A-series PLC Using MELSEC Protocol with Ethernet Transmission

2.2.5 OMRON Protocol

The OMRON protocol is used to read and write word and bit data in OMRON SYSMAC Programmable Logic Controllers without having to be concerned with the protocol.

■ Communicating with the OMRON Protocol

The OMRON protocol is used to communicate between the MP2000 Series Machine Controller and OMRON SYSMAC Programmable Logic Controllers (general-purpose PLCs) and normally the MP2000 Series Machine Controller is used as the Master. The MP2000 Series Machine Controller as the Master sends commands in the OMRON format to a SYSMAC PLC as a Slave and receives response messages from the Slave. Conversion to the OMRON format is automatically performed inside the Communication Module. The MSG-SND function can be used in user program in the same way as with the MEMOBUS protocol.

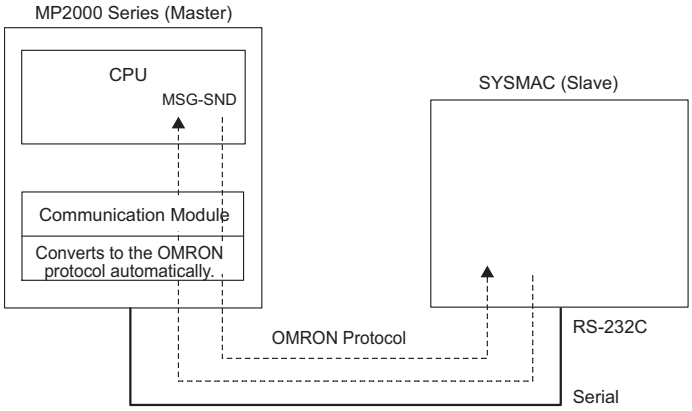


Fig. 2.5 Message Flow between MP2000 Series Machine Controller and OMRON SYSMAC PLC Using OMRON Protocol

* Although the MP2000 Series Machine Controller can be set also as a Slave, it is assumed that the OMRON PLC will be used as a Slave due to its characteristics (upward linking mode). If the MP2000 Series Machine Controller is set as a Slave, the MSG-RCV function is used.

2.2.6 MODBUS/TCP Protocol

The MODBUS/TCP protocol enables communication with devices of other manufacturers that support the MODBUS/TCP protocol.
In communication between an MP2000 Series Machine Controller and a device that supports the MODBUS/TCP, the 218IF-01 or 218IF-02 Module automatically converts the protocol to the MODBUS/TCP protocol. The MSG-SND and MSG-RCV functions can be used in user program in the same way as with the MEMOBUS protocol.

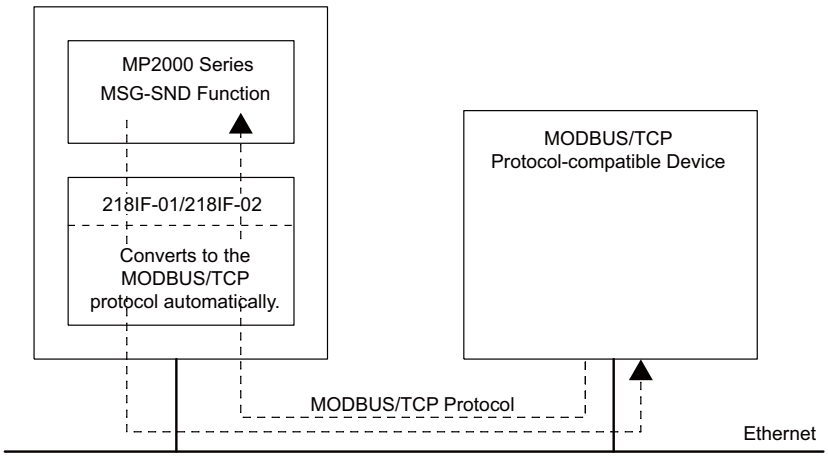


Fig. 2.6 Message Flow between MP2000 Series Machine Controller and MODBUS/TCP Protocol-compatible Device Using MODBUS/TCP Protocol

2.2.7 Non-Procedure Protocol

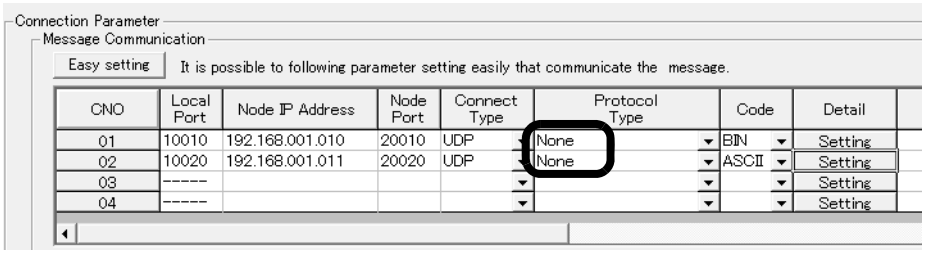
With non-procedure communication protocol, the data in consecutive MW registers in the MP2000 Series Machine Controller are sent and received as they are.

(1) Non-procedure Protocol Communication Function

MESSAGE Function	Function
PRO_TYP=2	Sends and receives messages in word units.
PRO_TYP=3	Sends and receives messages in byte units.

(2) Transmission Parameter Setting Example

The **Connection Parameter** Window shown below is used to set the protocol type.
To use the non-procedure communication mode, specify **None** in the **Protocol Type** column in the window.



(3) Communication Programs

Communication in the non-procedure mode are enabled by specifying No protocol 1 (PRO_TYP = 2) or No protocol 2 (PRO_TYP = 3) in the MSG-SND or MSG-RCV function in the user program. Messages are sent and received in word units when no protocol 1 is selected and in byte units when no protocol 2 is selected.

- For details on the MSG-SND and MSG-RCV functions, refer to *Message Send Function* and *Message Receive Function* in chapters 5 to 9.

3

Serial Communication

This chapter describes how to define parameters for the serial interface of the Communication Modules and provides connection and programming examples.

•Before reading this chapter, be sure to read *Chapter 1 Starting Communication Modules*.

3.1 Serial Communications ----- 3-3

3.2 Message Communication Protocols ----- 3-4

3.2.1 MEMOBUS Protocol ----- 3-4

3.2.2 MELSEC Protocol ----- 3-5

3.2.3 OMRON Protocol ----- 3-8

3.2.4 Non-procedure Protocol ----- 3-10

3.2.5 Non-procedure FD Protocol ----- 3-11

3.3 217IF Transmission Definitions ----- 3-14

3.3.1 217IF Transmission Configuration Window ----- 3-14

3.3.2 Setting the Parameters ----- 3-16

3.3.3 Saving and Deleting 217IF Transmission Definitions ----- 3-19

3.4 Time Required for Transmission ----- 3-20

3.4.1 Overview ----- 3-20

3.4.2 Estimating the Time Required for Each Step ----- 3-20

3.5 Serial Communication Connection Example ----- 3-22

3.5.1 Connecting to the 217IF-01 ----- 3-22

3.5.2 Connecting to an Operator Interface Using MEMOBUS Communication ----- 3-27

3.5.3 Connection to MELSEC ----- 3-32

3.5.4 Connection to an OMRON PLC ----- 3-37

3.5.5 Connection to a Temperature Controller ----- 3-42

3.5.6 Connection to an Inverter ----- 3-48

3.5.7 Connection to Programming Device ----- 3-51

3.6 Message Send Function ----- 3-53

3.6.1 Basic Specifications ----- 3-53

3.6.2 Inputs and Outputs for the Message Send Function ----- 3-54

3.6.3 Parameter List for Serial Communication (217IF) ----- 3-58

3.6.4 Function Setting Example and Parameter Details for the MEMOBUS Protocol ----- 3-59

3.6.5 Function Setting Example and Parameter Details for the MELSEC Protocol ----- 3-67

3.6.6 Function Setting Example and Parameter Details for the OMRON Protocol ----- 3-71

3.6.7 Function Setting Example and Parameter Details for the Non-procedure Protocol - - - 3-75

3.7 Message Receive Function - 3-79

3.7.1 Basic Specifications - 3-79
3.7.2 Inputs and Outputs for the Message Receive Function - 3-80
3.7.3 Parameter List for Serial Communication (217IF) - 3-84
3.7.4 Function Setting Example and Parameter Details for the MEMOBUS Protocol - - - - - - 3-85
3.7.5 Function Setting Example and Parameter Details for the MELSEC Protocol - - - - - - - - 3-91
3.7.6 Function Setting Example and Parameter Details for the OMRON Protocol - - - - - - - - 3-94
3.7.7 Function Setting Example and Parameter Details for the Non-procedure Protocol - - - 3-97

3.1 Serial Communications

Each of the MP2000-series Communication Modules provides an RS-232C communication interface through a connector labeled "PORT" as a serial interface. The 217IF-01 Module also provides an RS-422/485 interface through a connector labeled "RS-422/485."
The serial interfaces can be used for message communication with Yaskawa's MEMOBUS communication protocol, as well as the other protocols listed below.

- Refer to the page number given in the *Reference* column for details on a protocol.

Protocol	Remarks	Master/Slave	Reference
MEMOBUS	Yaskawa's standard MEMOBUS protocol	Master/Slave	P.3-4
MELSEC A-compatible 1C	A subset of the sub-protocol for Mitsubishi Electric's General-purpose Programmable Logic Controllers	Master/Slave	P.3-5
OMRON Host Link	A protocol used for connecting a controller manufactured by OMRON Corporation	Master	P.3-8
Non-procedure Protocol	A protocol provided to implement general-purpose message communication	Master/Slave	P.3-10
Non-procedure FD Protocol	A protocol provided to implement general-purpose message communication that supports full-duplex communication	Master/Slave	P.3-11

Message communication is used to send and receive data using functions, such as the MSG-SND and MSG-RCV functions, in a ladder program or function program.
RS-232C communication can also be used for engineering communication with the MPE720 running on a personal computer.

3.2 Message Communication Protocols

This section describes the protocols that can be used for serial communication with the 217 transmission system.

3.2.1 MEMOBUS Protocol

MEMOBUS transmissions use messages controlled by the MEMOBUS protocol. Messages are sent and received in Master-Slave relationships. The Master sends a message to a Slave and a response message is returned from the Slave that received the message.
The Master communicates with one Slave at a time. To enable this, Slaves are assigned unique address numbers. The Master specifies the address number of a specific Slave when sending data. On receiving a message from the Master, the Slave performs the specified function and returns a response to the Master.
Using these functions, the Master can read and modify the status of coils and contents of the registers in Slaves located at different places, enabling flexible centralized monitoring and control in a variety of production fields.

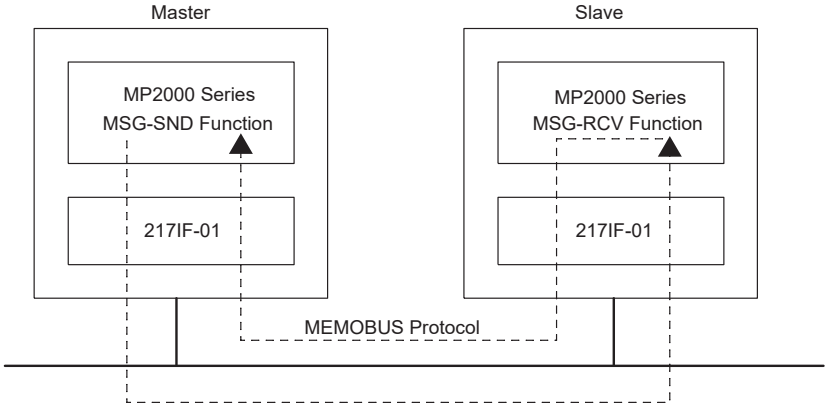


Fig. 3.1 Message Flow Using the MEMOBUS Protocol

(1) Function Codes in the MEMOBUS Protocol

The following table lists the function codes used in the MEMOBUS protocol. These functions are implemented by setting the function code in the MSG-SND function.

Function Code	Function	Data Address Setting Range	Data Size
01H	Read Coil Status	0 to 65535	1 to 2000 bits
02H	Read Input Relay Status	0 to 65535	1 to 2000 bits
03H	Read Holding Register Contents	0 to 65534	1 to 125 words
04H	Read Input Register Contents	0 to 32767	1 to 125 words
05H	Modify Status of a Single Coil	0 to 65535	–
06H	Write to a Single Holding Register	0 to 65534	–
08H	Loopback Test	–	–
0FH	Modify Status of Multiple Coils	0 to 65535	1 to 800 bits
10H	Write to Multiple Holding Registers	0 to 65534	1 to 100 words

3.2.2 MELSEC Protocol

The MELSEC protocol is used for communication between an MP2000 Series Machine Controller and a MELSEC A-series Programmable Logic Controller manufactured by Mitsubishi Electric Corporation.
The MELSEC protocol is used to communicate between the MP2000 Series Machine Controller and a MELSEC Programmable Logic Controller.
To achieve this, the 217IF-01 Module automatically converts the protocol to the MELSEC protocol. The MSG-SND and MSG-RCV functions can be used in user program in the same way as with the MEMOBUS protocol.
The 217IF transmission system supports the special protocol type 1 functions of the MELSEC protocol. With the 217IF transmission system, the MP2000 Series Machine Controller is the Master.

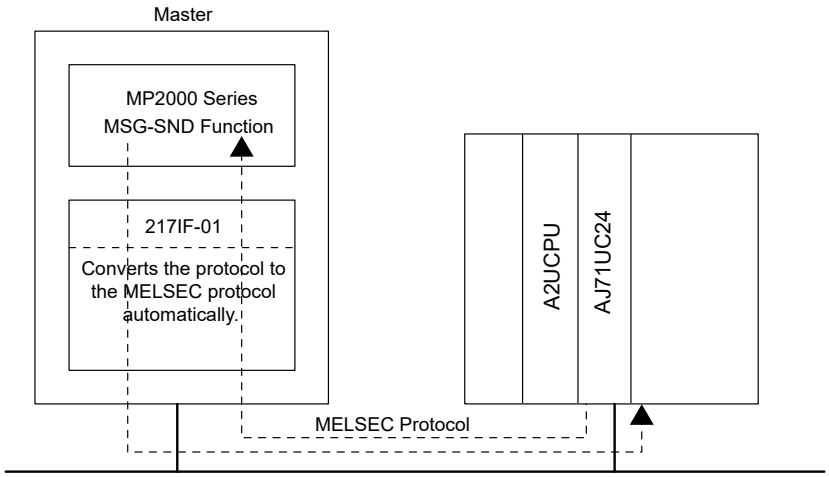


Fig. 3.2 Message Flow Using the MELSEC Protocol

(1) Command Function Code List

The following table lists the common MELSEC ACPU commands and MEMOBUS function codes supported by the 217IF transmission system. To communicate with a MELSEC A-series Programmable Logic Controller, set the MEMOBUS function code that corresponds to the MELSEC ACPU command in the MSG-SND function.

Common MELSEC ACPU Command	MEMOBUS Function Code	Function	Number of Points
WR	01H/02H	Reads data from bit-type devices in 16-point units.	32 words (512 points)
	03H/04H	Reads data from word-type devices in 1-point units.	64 points
WW	0FH	Writes data to bit-type devices in 16-point units.	10 words (160 points)
	10H	Writes data to word-type devices in 1-point units.	64 points
TT	08H	Executes a loopback test.	254 characters

- The special commands for the AnCPU are not supported. Use the common ACPU commands listed above when accessing an AnCPU. Accessing the extended registers in an AnCPU is not possible.

(2) MELSEC Devices

The following tables list the common ACPU command device ranges, MEMOBUS commands, and MP2000 Series Controller register numbers for MELSEC bit-type and word-type devices.
Use these tables as a basis for setting the parameters for the MSG-SND parameters when reading and writing MELSEC devices from an MP2000 Series Controller.

■ MELSEC Bit-type Devices

Device	Common ACPU Command Device Range	Decimal/ Hex	MEMOBUS Commands	Start No.	Register No.
X	X0000 to X07FF	Hex	02H: Input relays	0 to 2047	MB000000 to MB00127F
Y	Y0000 to Y07FF	Hex	01H/0FH: Coils	0 to 2047	MB000000 to MB00127F
M	M000 to M2047	Decimal	01H/0FH: Coils	2048 to 4095	MB001280 to MB00255F
M	M9000 to M9255	Decimal	01H/0FH: Coils	4096 to 4351	MB002560 to MB00271F
B	B0000 to B03FF	Hex	01H/0FH: Coils	4352 to 5375	MB002720 to MB00335F
F	F0000 to F0255	Decimal	01H/0FH: Coils	5376 to 5631	MB003360 to MB00351F
TS	TS000 to TS255	Decimal	02H: Input relays	2048 to 2303	MB001280 to MB00143F
TC	TC000 to TC255	Decimal	02H: Input relays	2304 to 2559	MB001440 to MB00159F
CS	CS000 to CS255	Decimal	02H: Input relays	2560 to 2815	MB001660 to MB00175F
CC	CC000 to CC255	Decimal	02H: Input relays	2816 to 3071	MB001760 to MB00191F

■ MELSEC Word-type Devices

Device	Common ACPU Command Device Range	Decimal/ Hex	MEMOBUS Commands	Start No.	Register No.
TN	TN0000 to TN255	Decimal	04H: Input registers	0 to 255	MW00000 to MW00255
CN	CN0000 to CN255	Decimal	04H: Input registers	256 to 511	MW00256 to MW00511
D	D0000 to D1023	Decimal	03H/10H: Holding registers	0 to 1023	MW00000 to MW01023
D (Special)	D9000 to D9255	Decimal	03H/10H: Holding registers	1024 to 1279	MW01024 to MW01279
W	W0000 to W03FF	Hex	03H/10H: Holding registers	1280 to 2303	MW01280 to MW02303
R	R0000 to R8191	Decimal	03H/10H: Holding registers	2304 to 10495	MW02304 to MW10495

The MP2000 Series Machine Controller register addresses can be adjusted by using the offset setting in the MSG-SND function, as shown in the following example.

Example: Reading 10 Words from MELSEC Data Registers (D0100)

From the table for MELSEC word-type devices, the register number for device D0100 is MW00100, the starting number is 100, and the MEMOBUS command to read the data is 03H.
These values are set in the parameters of the MSG-SND function.

- PARAM04: Function code = 03
- PARAM05: Data address = 100
- PARAM06: Data size = 10 (words)

The data address in PARAM05 serves as both the address to read and the storage address. The above settings thus read 10 words of data starting from D0100 and write it to the 10 words starting from MW00100.
To store the register at a location other than MW00100, the holding register offset in PARAM11 is used to adjust the storage location.
Example:

- PARAM11: Holding register offset =1000

The above setting stores the 10 words of data read starting at D0100 to the 10 words starting at MW01100 (i.e., MW00100 + 1000). Negative offsets cannot be used.

• Refer to *3.6 Message Send Function* on page 3-53 for details on message communication functions.

(3) MP2000 Series Machine Controller/MELSEC Register Map

The following register map is based on the device table in *(2) MELSEC Devices* on page 3-6 and shows the MP2000 Series Machine Controller M registers that correspond to each MELSEC device. Holding registers, input registers, input relays, and coils are assigned for each MELSEC device so that the MP2000 Series Machine Controller can use MEMOBUS commands as an interface to read and write data. Data read from a MELSEC device is stored in the corresponding M registers shown in the following map. Data written to the MELSEC devices is written from the corresponding device M registers shown in the following table.

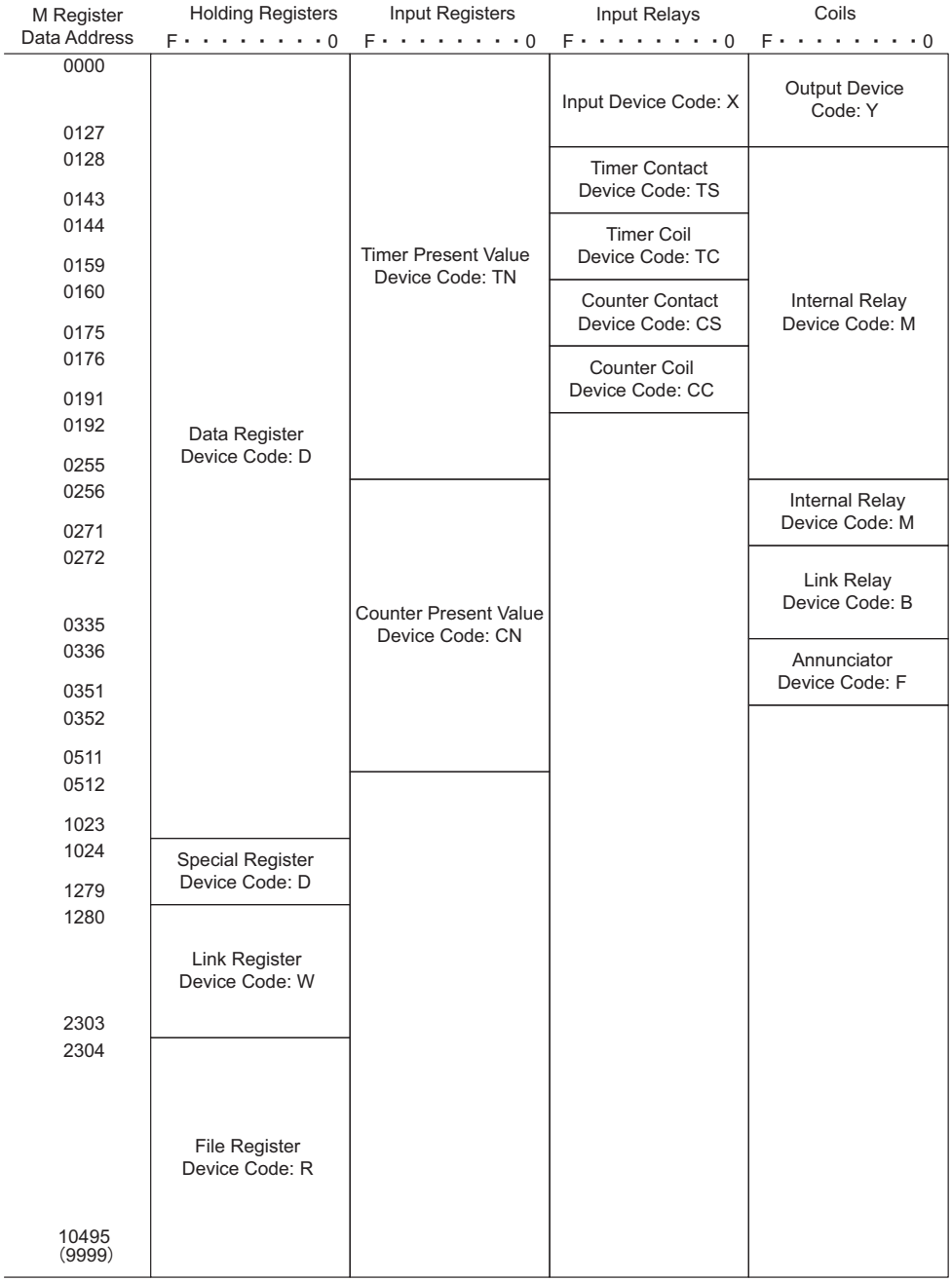


Fig. 3.3 Register Map

- This map is for when the offset parameters in the MSG-SND function are set to 0. An offset parameter (i.e., holding register offset, input register offset, input relay offset, or coil offset) can be set to shift the M register numbers given in this map backward by a relative amount.

3.2.3 OMRON Protocol

The OMRON protocol is used to communicate between the MP2000 Series Machine Controller and OMRON SYSMAC Programmable Logic Controllers (general-purpose PLCs).
The MP2000 Series Machine Controller as the Master sends commands in the OMRON format to SYSMAC PLC as a Slave and receives response messages from the Slave. Conversion to the OMRON format is automatically performed inside the Communication Module. The MSG-SND function can be used in a user program in the same way as with the MEMOBUS protocol.
Refer to *1.2.4 Software Versions of Supported Communications Module* and *1.2.5 Applicable Versions of MPE720* for details on the versions of software and MPE720 that support this protocol.

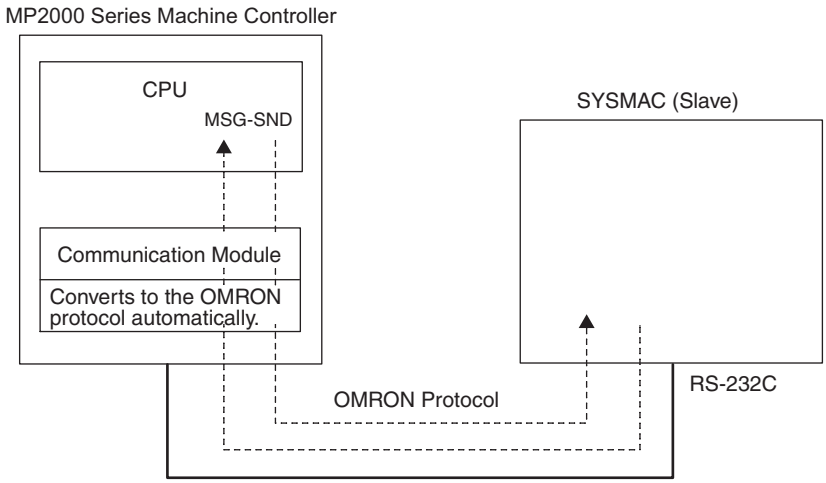


Fig. 3.4 Message Flow Using the OMRON Protocol

- Of the OMRON protocols, only the Host Link mode is supported.
  Normally the MP2000 Series Machine Controller is the Master. Although it can be set as a Slave, it is assumed that the OMRON PLC will be used as a Slave due to its characteristics (Host Link mode). If the MP2000 Series Machine Controller is set as the Slave, the MSG-RCV function is used.

(1) Command Function Code List

The following table shows the Host Link commands in the OMRON protocol and the MEMOBUS function codes supported by the MP2000 Series Machine Controllers.
To communicate using the OMRON protocol, set the MEMOBUS function code that corresponds to the command in the MSG-SND function.

OMRON Protocol Specifications			MEMOBUS Protocol Specifications	
OMRON Header Code	Command	Number of Words	Number of Words Accessible with One Command	Function Code
RR	Read I/O relay, internal auxiliary relay, and special auxiliary relay	256 words	125 words	01H
RD	Read data memory area	2000 words	125 words	03H
WR	Write I/O relay, internal auxiliary relay, and special auxiliary relay	252 words	50 words	0FH
WD	Write data memory area	2000 words	100 words	10H
TS	Test	–	–	08H

- Although the multi-block transmission format defined in the OMRON protocol is supported, the number of words that can be accessed with one command is restricted by the MEMOBUS protocol.

(2) OMRON Devices

The following table shows the OMRON relays and data memory that can be accessed from the MP2000 Series Machine Controller along with the corresponding register numbers and MEMOBUS commands.

OMRON Protocol Specifications			MEMOBUS Protocol Specifications		
Name	Channel Number	Relay Number	Function Code	Start No.	Register No.
I/O Relay	000 to 039	00000 to 03915	01H/0FH	0 to 639	MB000000 to MB00039F
Internal Auxiliary Relay	040 to 246	04000 to 24615	01H/0FH	640 to 3951	MB000400 to MB00246F
Special Auxiliary Relay	247 to 255	24700 to 25507	01H/0FH	3952 to 4088	MB002470 to MB002557
Data Memory	0000 to 9999	DM0000 to DM9999	03H/10H	0000 to 9999	MW00000 to MW09999

- OMRON protocol specifications depend on the model of OMRON PLC being used. Refer to the user documentation provided by OMRON for information on the PLC model and OMRON communication procedure.
- The MP2000 Series Machine Controller register number can be adjusted by using the offset setting in the MSG-SND function.

3.2.4 Non-procedure Protocol

With non-procedure communication, data in consecutive M registers in the MP2000 Series Machine Controller are sent and received without modification.
The MSG-SND function sends the data from the specified M registers without modification. There is no response.
The MSG-RCV function stores the received data without modification in the M registers specified in the MSG-RCV function. It does not return a response.
Thus, the non-procedure protocol sends data in just one direction without a response.
The non-procedure protocol is not based on command-response communication. It is thus necessary to make a user program in which sending and receiving are switched to perform command-response communication.
The size of data that can be sent or received in non-procedure communication mode is from 1 to 254 words.

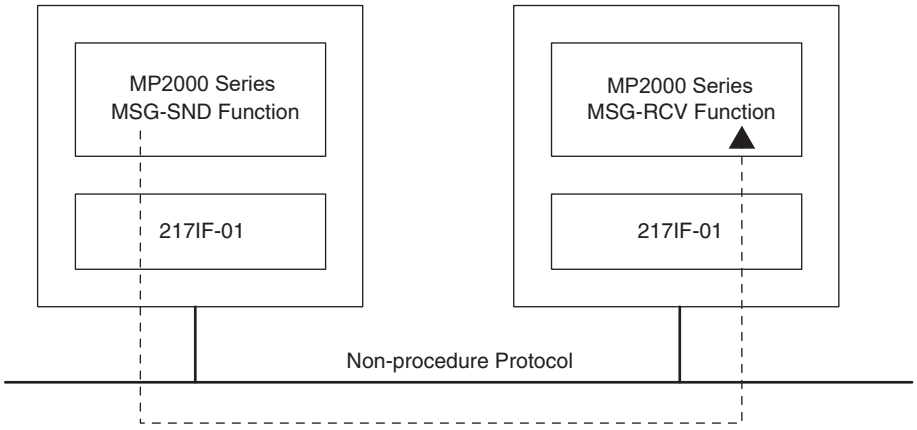
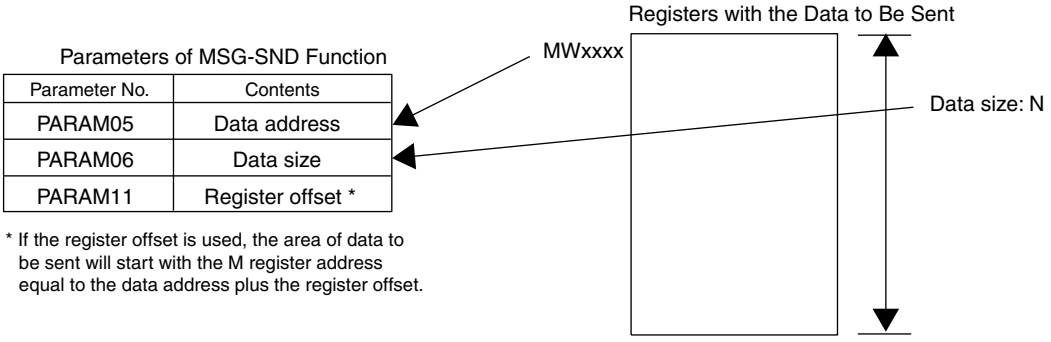
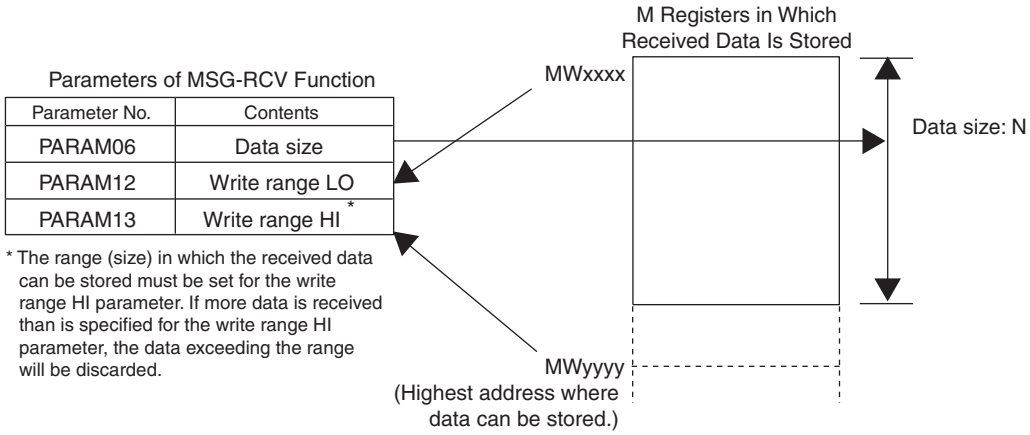


Fig. 3.5 Message Flow Using the Non-procedure Protocol

With the non-procedure protocol, the address of the first M register and the data size are specified in the MSG-SND function to enable sending the contents of M registers.



With the MSG-RCV function, the write range LO and write range HI can be set to store the received data in the specified M registers.



3.2.5 Non-procedure FD Protocol

The non-procedure FD protocol is a full-duplex version (simultaneous communications are possible for both sending and receiving) of the non-procedure protocol, which is for half-duplex communications (simultaneous communications are not possible for both sending and receiving). With the standard non-procedure protocol, it is necessary to make a user program in which sending and receiving are switched to perform command-response communications. With the non-procedure FD protocol, you can continuously execute reception (MSG-RCV function) and sending (MSG-SND function) whenever required. Sequential programming for the MSG-RCV function and MSG-SND function is not required.

Refer to *1.2.4 Software Versions of Supported Communications Module* and *1.2.5 Applicable Versions of MPE720* for details on the versions of software and MPE720 that support this protocol.

(1) Specifications

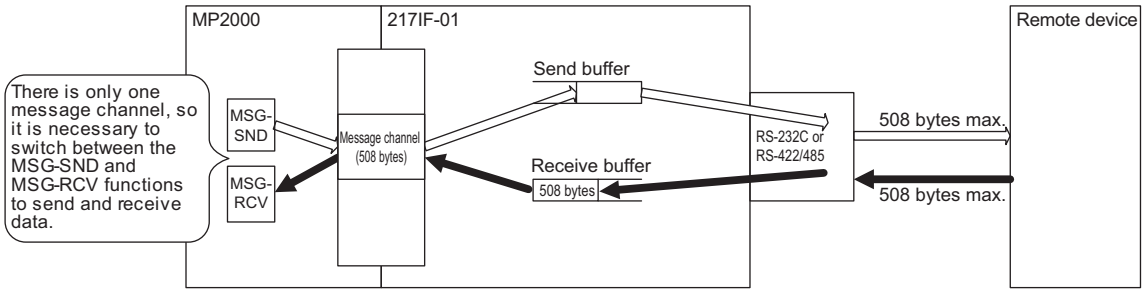
The differences in specifications between the standard non-procedure protocol and non-procedure FD protocol are given in the following table.

	Non-procedure Protocol	Non-procedure FD Protocol
Full-duplex Communications	Not supported.	Supported.
Number of Message Channels	1 (Used for both MSG-SND and MSG-RCV.)	2 (1 for MSG-SND and 1 for MSG-RCV)
Maximum Sending Size	254 words	254 words
Maximum Receiving Size	254 words	5,080 words
Processing for Data Errors	Entire message is discarded.	Only the error byte is discarded.
Data Error Detection Reporting	None.	Reported in MSG-RCV function.

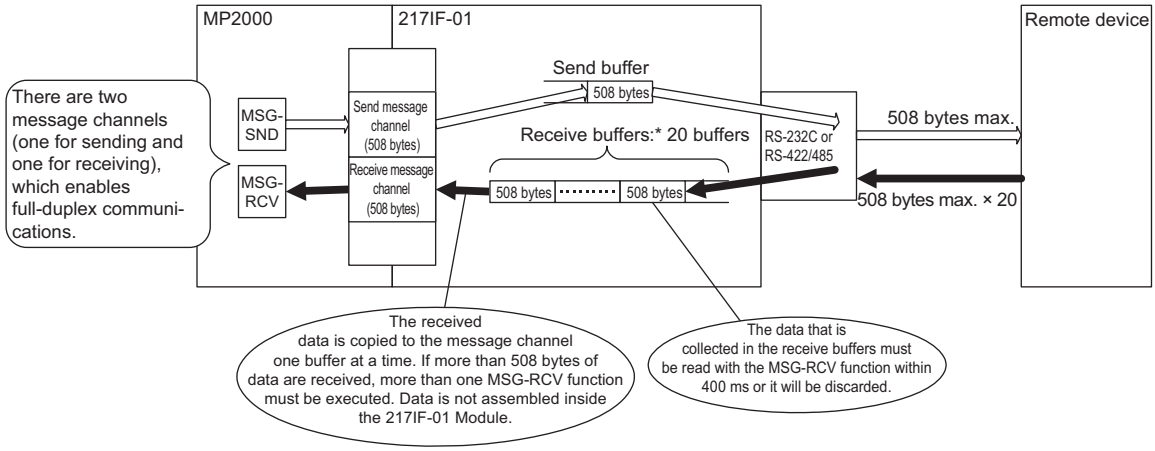
(2) Protocol Operation

This section shows the differences in operation of the 217IF-01 Module for the standard non-procedure protocol and the non-procedure FD protocol. The number of message channels was increased from 1 to 2 channels for the non-procedure FD protocol. Therefore, the MSG-SND function and MSG-RCV function can be executed at the same time, which enables full-duplex communications. Also, the number of reception buffers was increased, allowing up to 5,080 words of continuous data to be received.

Non-procedure Protocol Operation



Non-procedure FD Protocol Operation



* Rules for Storing Receive Data in Receive Buffers

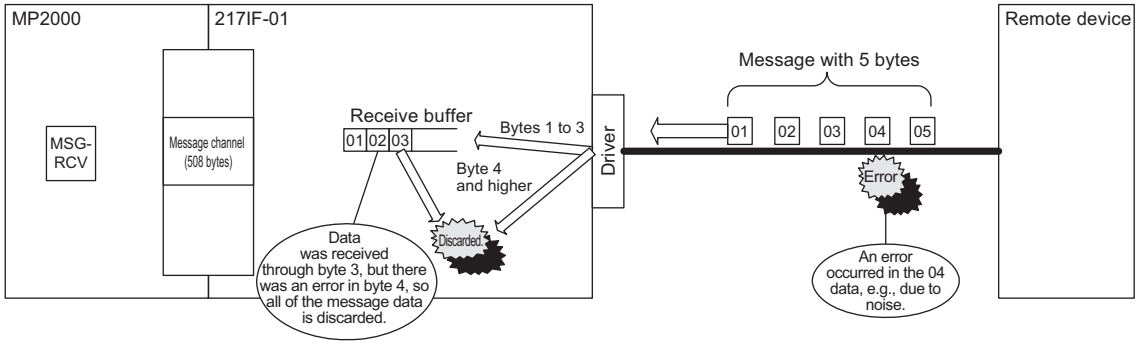
- Receive data delimitation is monitored on a time basis. If the time set for the reception monitoring time in the communications parameters of the 217IF-01 Module elapses, the data up to that time is stored in a reception buffer as one message.
- If 508 bytes of data or less are received, they are stored as receive data in one receive buffer. For example, if one byte of data is detected as a message, only that one byte of data is stored in a receive buffer.
- If 509 bytes or more of data are received, the continuous data is received as long as receive buffers are available (20 buffers max.) and 508 bytes are stored in each receive buffer.

- To perform full-duplex communications, sending and receiving must be wired separately. For RS422/485, use a 4-wire connection (and not a 2-wire connection).

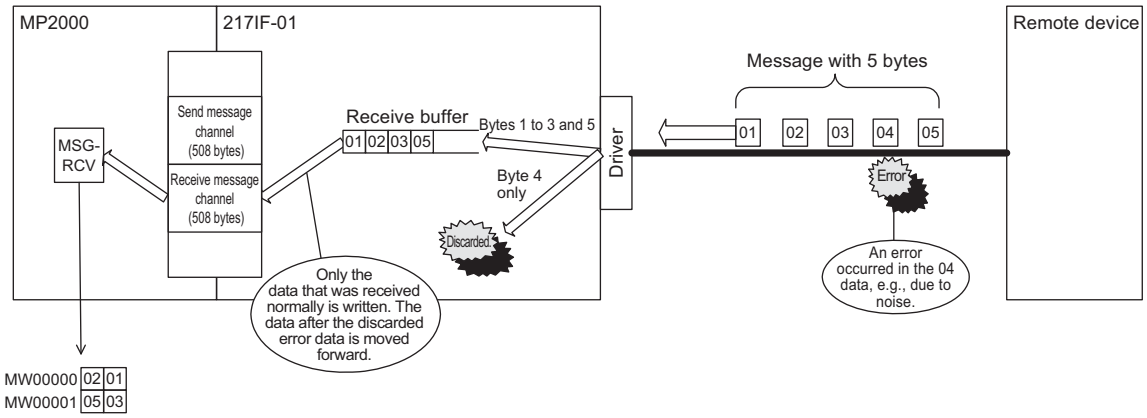
(3) Processing for Receive Data Errors

This section shows the differences in processing errors in received data for standard non-procedure protocol operation and non-procedure FD protocol operation of the 217IF-01 Module.
The standard non-procedure protocol functions in the same way as the MEMOBUS and other protocols, i.e., all of the message data is discarded if there is an error in even one byte in the message. One message is all of the data received until the reception monitor timer times out. With the non-procedure FD protocol, the definition of a message is the same as the standard non-procedure protocol, but even if an error occurs, only the byte with the error is discarded and the rest of the data is stored in the receive buffers. Information is passed to a parameter in the MSG-RCV function to report that there was an error in the data in the received message.

Processing for Receive Data Errors for the Non-procedure Protocol

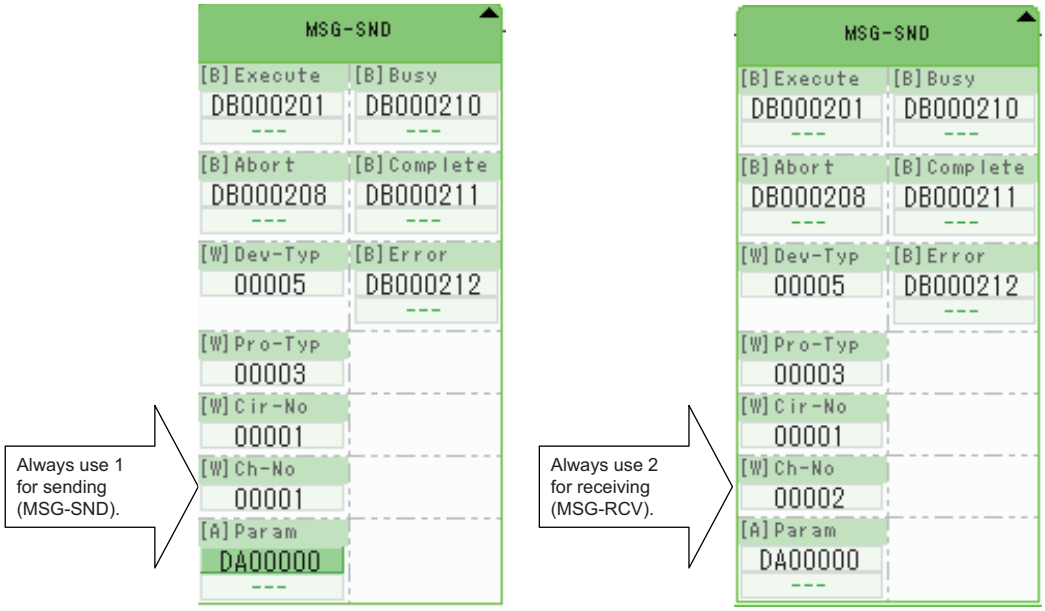


Processing for Receive Data Errors for the Non-procedure FD Protocol



(4) Precautions for Non-procedure FD Protocol

Set Ch-No (channel number) in the MSG-SND function to 1 and set Ch-No in the MSG-RCV function to 2.



3.3 217IF Transmission Definitions

To perform serial communication with the MP2000 Series Communication Modules, the transmission definition file must be created for the serial interface. This file is called the 217IF transmission definitions file.
This section describes how to define transmissions for the serial interface.

3.3.1 217IF Transmission Configuration Window

To define the 217IF transmissions, first the **Module Configuration** Window is opened on the MPE720, and then the **217IF Transmission Configuration** Window is opened.

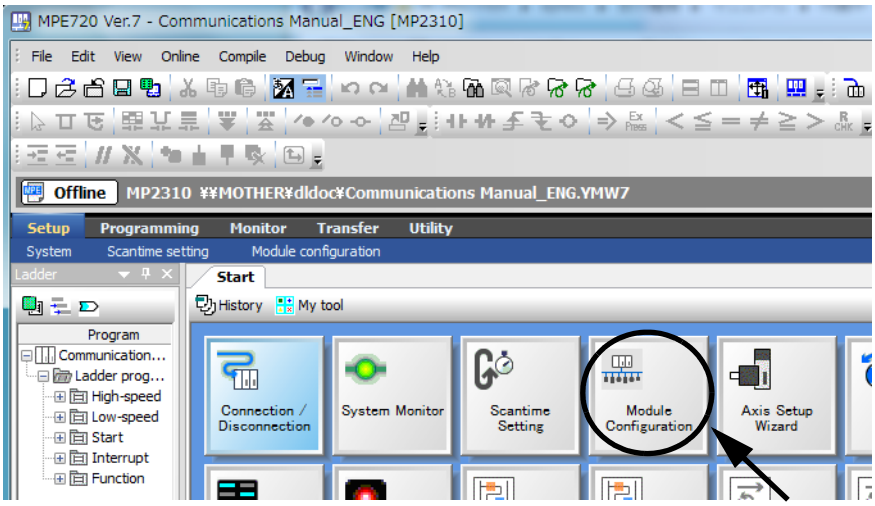
(1) Displaying the Module Configuration Window

Use the following procedure to display the **Module Configuration** Window.

1. Start the MPE720 on the personal computer connected to the Machine Controller and open the project file.

- For information on starting the MPE720, refer to *MP2000/MP3000 Series Engineering Tool MPE720 Version 7 User's Manual (Manual No.: SIEP C880761 03)*.

2. Click the **Module Configuration** Button on My Tool View.



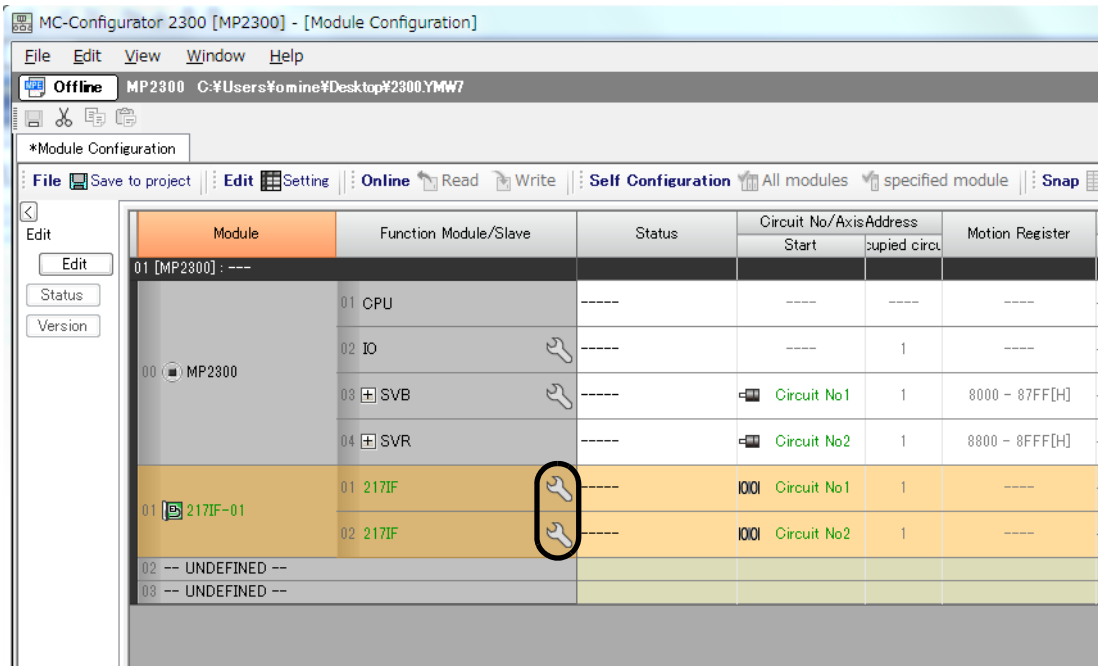
The **Module Configuration** Window will be displayed.

(2) Displaying the 217IF Transmission Configuration Window from the Module Configuration Window

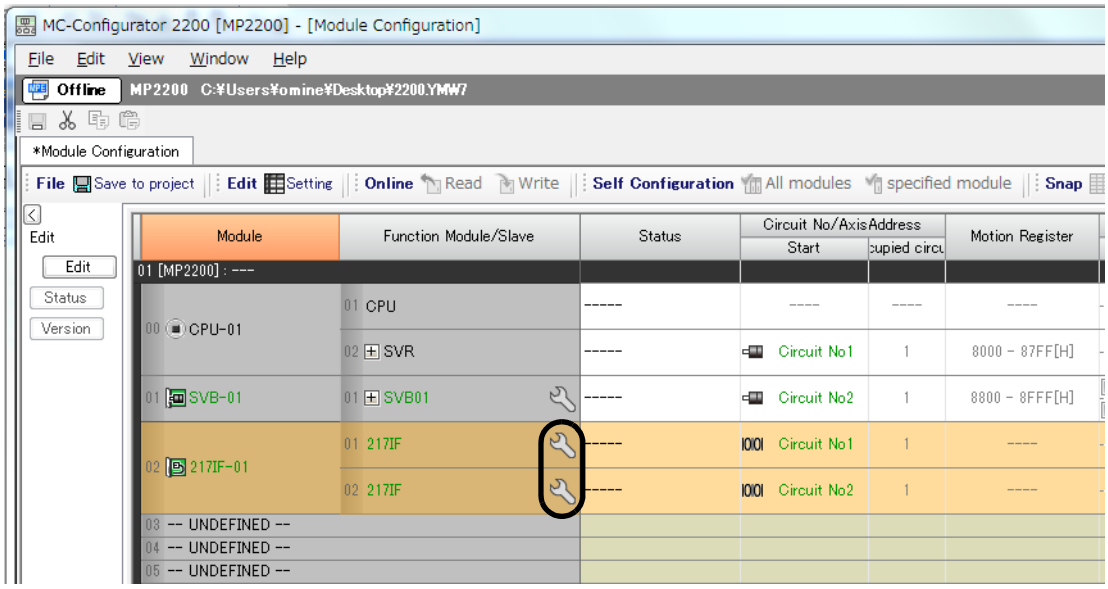
As the result of executing self-configuration, all Option Modules connected to the Machine Controller will be displayed in the **Controller** Area of the **Module Configuration** Window. (Refer to *1.5 Executing Self-configuration* on page 1-26.)

In the **Controller** Area, select the Communication Module for which to define transmissions, and then double-click the slot number cell for the 217IF in the **Module Details** Area. The **217IF Transmission Configuration** Window (see next page) will be displayed.

<MP2300/MP2300S/MP2310 Module Configuration Window>



<MP2100M/MP2200 Module Configuration Window>

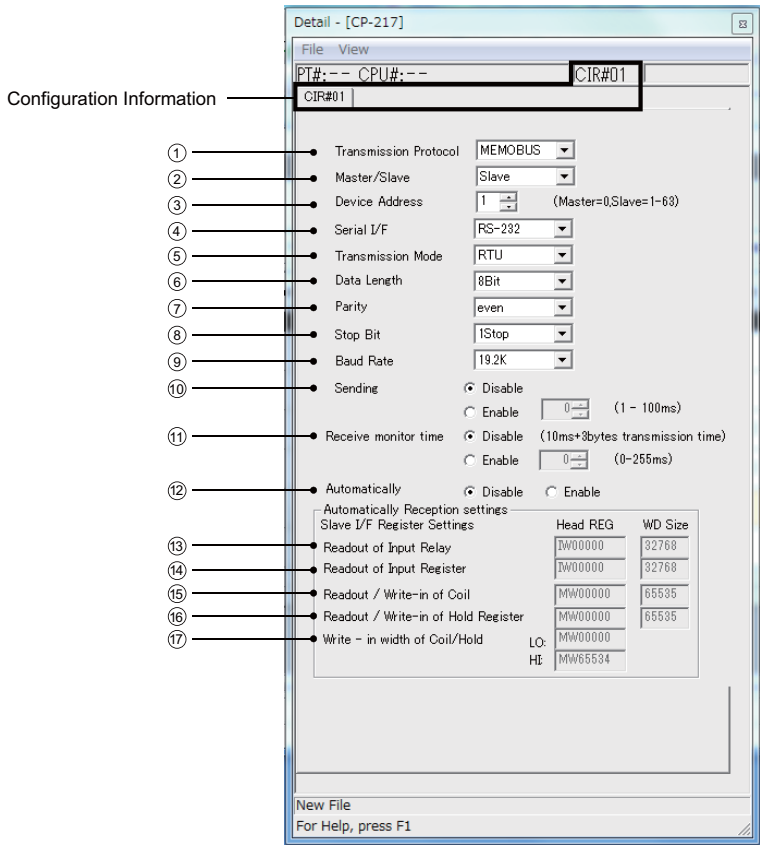


In Online Mode, the 217IF transmission definitions data saved in the Machine Controller are displayed. In Offline Mode, the definitions data saved in the personal computer on which MPE720 is running is displayed.

- If the **Transmission Configuration** Window is being opened for the first time, a "new file" message box will be displayed and the **217IF Transmission Configuration** Window will be displayed when the **OK** Button is clicked.
- For the 217IF-01 Module, “217IF” will be displayed in two different columns in the **Module Details** Area. The column on the upper side is for the RS-232C settings, and the column on the lower side is for the RS-422/485 settings.

3.3.2 Setting the Parameters

Set the parameters required for using serial communication.



- Parameters ⑬ to ⑰ can be set after automatic reception ⑫ is enabled.

■ Configuration Information

The configuration information is the same as the information displayed in the **Module Details** Area in the **Module Configuration** Window.

RACK#: The number of the rack in which the Communication Module is defined.

CIR# (circuit number): The circuit number of the serial interface is displayed.

- The configuration information can also be changed in the **Module Details** Area of the **Module Configuration** Window. Do not use a circuit number that is already being used for another serial interface.
- When the MSG-SND or MSG-RCV function is used, set the circuit number displayed here for CIR# (circuit number).

■ Parameters

① Transmission Protocol (Communication Protocol)

Select the desired protocol from the following.

MEMOBUS: Yaskawa's standard MEMOBUS protocol

MELSEC: Protocol (special protocol type 1) used for Mitsubishi MELSEC A-series general-purpose programmable logic controllers

OMRON: OMRON's protocol (Host Link mode)

Non-procedure protocol: Data can be sent or received at any timing using the user application program (half-duplex communications).

Non-procedure FD protocol: Data can be sent and received at any timing using the user application program (full-duplex communications).

② Master/Slave

Select whether the MP2000 Series Machine Controller should operate as a Master or a Slave if **MEMOBUS**, **MELSEC**, or **OMRON** is selected for the **Transmission Protocol** parameter.

③ Device Address

Set the device address. When the Machine Controller is being used as a **Master**, set the device address to 0.
When the Machine Controller is being used as a **Slave**, set the device address to a unique value between 1 and 63.

④ Serial I/F

For any Module other than the 217IF-01, **RS-232C** will be displayed.
For the 217IF-01 Module, **RS-232C** will be displayed if the left **217IF** column in the **Module Configuration** Window was selected and **RS-485** will be displayed if the right **217IF** column was selected.
This setting cannot be changed.

⑤ Transmission Mode (Communication Mode)

Select the communication mode from the following.

RTU: Specifies the RTU mode when the MEMOBUS protocol is selected.
ASCII: Specifies the ASCII mode when the MEMOBUS protocol is selected.
None: Select this mode when the communications protocol is set to MELSEC, OMRON, Normal (Non-procedure), or Normal FD (Non-procedure FD).

⑥ Data Length

Set the number of bits that comprise one character.

8 Bit: 8 bits
7 Bit: 7 bits

⑦ Parity

Specify the use of the parity bit from the following.

even: Even parity
odd: Odd parity
none: No parity check

⑧ Stop Bit

Set the number of stop bits.

1 stop: 1 bit
2 stop: 2 bits

⑨ Baud Rate

Set the baud rate (unit: bps).

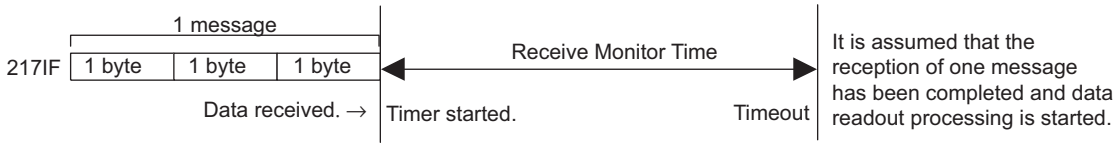
⑩ Sending (Send Delay)

If **Enable** is selected, a delay until the beginning of data transmission can be set between 1 and 100 ms. The meaning of the delay is different for a master and a slave.

Master: The delay will be included between executing the MSG-SND function and sending the command.
Slave: The delay will be included between receiving the command by the MSG-RCV function and sending the response.

⑪ Receive Monitor Time

Set the length of time (i.e., the receive monitor time) when no data is received that will be used to determine the completion of the reception of one message when receiving data for serial communication.



If **Disable** is selected (default), the receive monitor time will be the transmission time for three bytes plus 10 ms.
If **Enable** is selected, the receive monitor time can be set to between 0 and 255 ms.

- Set the time to longer than the transmission time for one byte.
  The optimum time depends on the user's environment. Adjust the value according to the environment.

- If 0 ms is set, the transmission time for three bytes will be used. This might be longer than 1 ms, depending on the baud rate that is selected.
- The receive monitor time cannot be set for the following:
  1) When the communication protocol is set to **MEMOBUS** and the communication mode is set to **ASCII**
  2) When the communication protocol is set to **OMRON**
- The communications module software versions that support this function are listed bellow.
  217IF-01 Module: Ver. 1.08 or later
  218IF-01 Module: Ver. 1.11 or later
  218IF-02 Module: Ver. 1.00 or later
  260IF-01 Module: Ver. 1.12 or later
  261IF-01 Module: Ver. 1.07 or later
  215AIF-01 Module: Ver. 2.02 or later
  The system version is given on a label on the PCB of the Communication Module as "V**.**".

  The receive monitor time cannot be set with versions earlier than those given above. The same time as for a setting of **Disable** will be used (i.e., the transmission time for three bytes plus 10 ms).

⑫ Automatically Reception (Automatic Reception)

This parameter can be enabled only when the Communication Module is set as a slave.
Specify whether or not a response is returned automatically for a command from the Master.
If automatic reception ⑫ is enabled, parameters ⑬ to ⑰ can be set.

- It is not necessary to make these settings when messages are not transmitted between the Master and Slave.
- Also, disable automatic reception when using a ladder program that transmits a response message with the MSG-RCV function.
- The automatic reception function is disabled when the **Communication Protocol** is set to **Normal** (non-procedure).

⑬ Readout of Input Relay

Set the leading register number and the range of input relays that can be read through serial communication.
Refer to the table for item ⑯, below, for settings.

⑭ Readout of Input Register

Set the leading register number and the range of input registers that can be read through serial communication.
Refer to the table for item ⑯, below, for settings.

⑮ Readout/Write-in of Coil

Set the leading register number and the range of coils that can be read or written through serial communication.
Refer to the table for item ⑯, below, for settings.

⑯ Readout/Write-in of Hold Register

Set the leading register number and the range of holding registers that can be read or written through serial communication. See the following table for settings.

	Leading Register	Number of Words
Readout of Input Relay	IW0000	32768
Readout of Input Register	IW0000	32768
Readout/Write-in of Coil	MW00000	65535
Readout/Write-in of Hold Register	MW00000	65535

⑰ Write-in Width of Coil/Hold Register

Set the overall range of coils and holding registers that can be written.

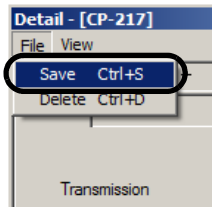
3.3.3 Saving and Deleting 217IF Transmission Definitions

- When changing, saving, or deleting 217IF transmission definitions data, save the data to flash memory in the Machine Controller, and then turn the power to the Machine Controller OFF and then turn it ON again.

(1) Saving the 217IF Transmission Definitions Data

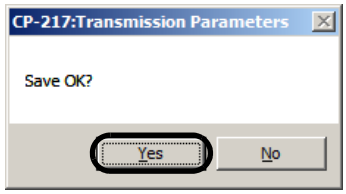
Use the following procedure to save the 217IF transmission definitions data if it has been set or changed. In Online Mode, the definitions data is saved in the Machine Controller and in personal computer running the MPE720. In Offline Mode, the definitions data is saved in the personal computer running the MPE720.

1. Select ***File - Save*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

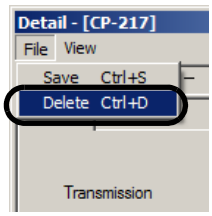
2. Click the **Yes** Button.



(2) Deleting the 217IF Transmission Definitions Data

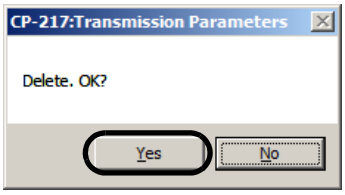
Use the following procedure to delete all of the 217IF transmission definitions data. In Online Mode, the definitions data is deleted from the MP2000 Series Machine Controller and the personal computer running the MPE720. In Offline Mode, the definitions data is deleted from the personal computer running the MPE720.

1. Select ***File - Delete*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

2. Click the **Yes** Button.



3.4 Time Required for Transmission

This section explains the time required for signal transmissions between a Master and a Slave using the MEMOBUS protocol as an example.

3.4.1 Overview

In the MEMOBUS system, the time required for transmission can be roughly calculated from the time required for the seven steps of transmission described below.

- Transmission processing time by the Master for query message
- Delay time of the modem at the Master
- Transmission time for query message
- Processing time at the Slave
- Delay time of the modem at the Slave
- Transmission time for response message
- Processing time by the Master for response message

If multiple Slaves are connected to the ports of the same Master, the overall time required for communication with the connected Slaves can be obtained by totaling the required communication time for the individual Slaves.

3.4.2 Estimating the Time Required for Each Step

The time required for each of seven steps listed above is explained in more detail in this section.

(1) Transmission Processing Time by the Master for Query Message

This is the length of time taken by the Master, such as a computer, to prepare the query message at the MEMOBUS port.
This time is determined by the processing time of the Master.
For the MP2000 Series Machine Controller, the time depends on the scan time, and this step usually takes one scan time.

(2) Delay Time of the Modem at the Master

This is the length of time taken by the modem at the Master to return the clear-to-send (CTS) signal to the Master after it receives the request-to-send (RTS) signal from the Master.
If a Yaskawa modem is used, this time may be ignored because it is less than 5 ms.

(3) Transmission Time for Query Message

This is the length of time taken to transmit the query message from the communication port of the Master.
This time is determined by the length of the query message and the baud rate, and can be calculated by using the following equation.

$$\text{Transmission time} = \frac{\text{Number of characters in query message} \times \text{Number of bits per character} \times 1{,}000\ (\text{ms})}{\text{Baud rate}}$$

In this calculation, add the number of data bits (8 or 7), the number of start bits (1), the number of stop bits (1 or 2), and the number of parity bits (1 or 0) to obtain the total number of bits per character.

(4) Processing Time at the Slave

This is the length of time taken by the Slave to process the query message after receiving it and then to prepare the response message at the MEMOBUS port.
This time is determined by the scan time of the Machine Controller, the number of coils and registers specified in the query message, and the number of devices that the Machine Controller can process per scan.
With the MP2000 Series Machine Controller, all functions can be processed in one scan. Therefore, this time will be one MP2000 Series Machine Controller scan time.

(5) Delay Time of the Modem at the Slave

This is the length of time taken by the modem at the Slave to return the clear-to-send (CTS) signal to the Slave after it receives the request-to-send (RTS) signal from the Slave.
If a Yaskawa modem is used, this time may be ignored because it is less than 5 ms.

(6) Transmission Time for Response Message

This is the length of time taken to transmit the response message from the communication port of the Slave. It can be calculated using the following equation, just like with the query message.

$$\text{Transmission time} = \frac{\text{Number of characters in query message x Number of bits per character x 1,000 (ms)}}{\text{Baud rate}}$$

In this calculation, add the number of data bits (8 or 7), the number of start bits (1), the number of stop bits (1 or 2), and the number of parity bits (1 or 0) to obtain the total number of bits per character.

(7) Processing Time by the Master for Response Message

This is the length of time taken by the Master, such as a computer, to process the response message from the Slave. This time is determined by the processing time of the Master.
For the MP2000 Series Machine Controller, the time depends on the scan time, and this step usually takes one or two scan times.

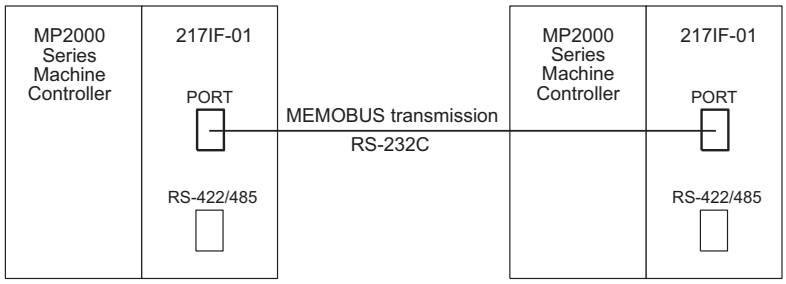
3.5 Serial Communication Connection Example

3.5.1 Connecting to the 217IF-01

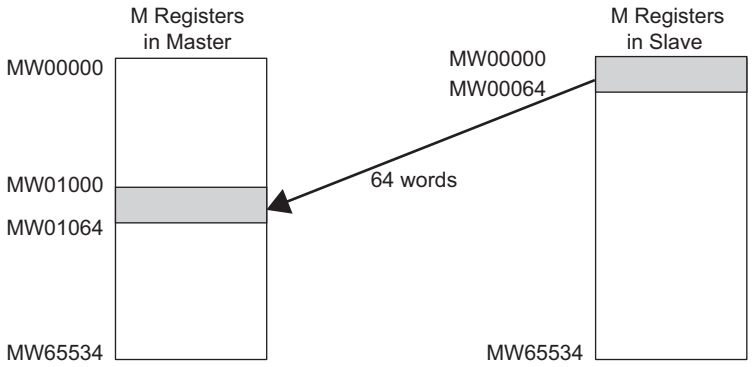
This section describes a system in which 217IF-01 Modules are connected to each other.

(1) System Configuration Example

The following example system configuration for message communication connects the RS-232C ports on two 217IF-01 Modules. This enables using any of the protocols supported by the 217IF-01 Modules for message communication.



In this configuration example, the Master MP2000 Series Machine Controller reads the contents of 64 words starting from MW00000 from the Slave MP2000 Series Machine Controller and writes the data in the Master starting at MW01000 using message communication.



(2) Cable Specifications

<Connections for a Remote Station with a 25-pin D-sub Connector>

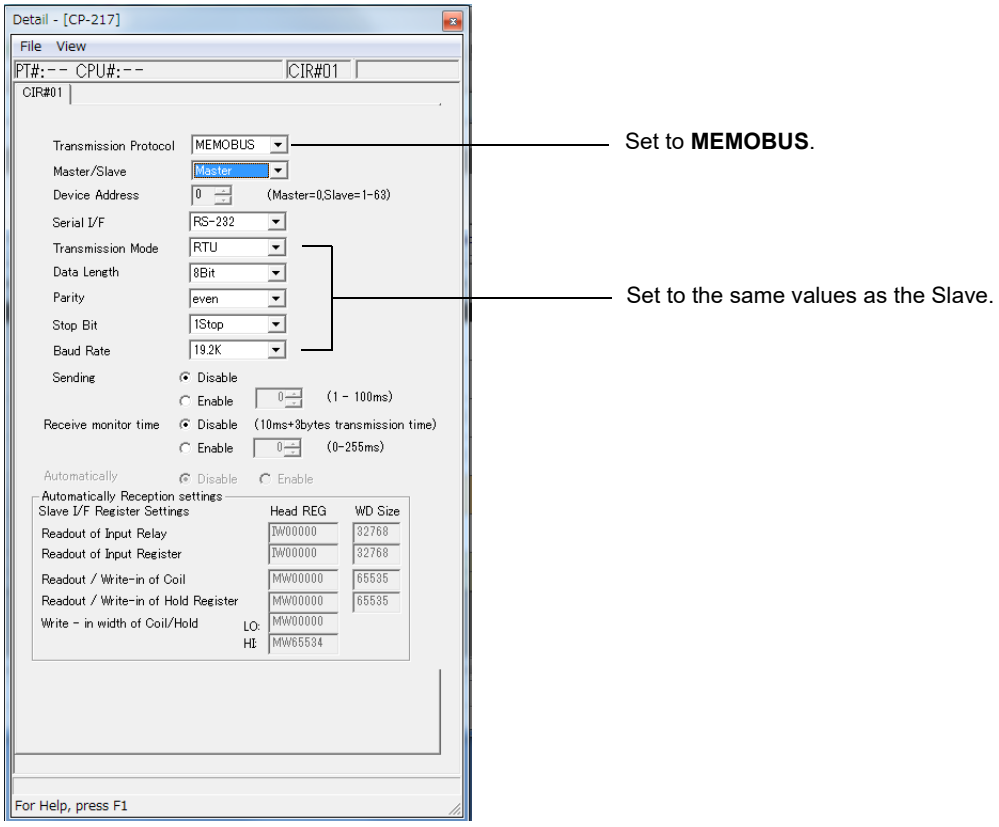
MP2000 Series Machine Controller (PORT Connector)		Cable Connections and Signal Direction	Remote Station (25-pin D-sub Connector)	
Signal Name	Pin Number		Pin Number	Signal Name
FG	1		1	FG
SD (TXD)	2		2	SD (TXD)
RD (RXD)	3		3	RD (RXD)
RS (RTS)	4		4	RS (RTS)
CS (CTS)	5		5	CS (CTS)
–	6		6	DR (DSR)
SG (GND)	7		7	SG (GND)
–	8		8	–
ER (DTR)	9		20	ER (DTR)

<Connections for a Remote Station with a Yaskawa 9-pin D-sub Connector>

MP2000 Series Machine Controller (PORT Connector)		Cable Connections and Signal Direction	Remote Station (9-pin D-sub Connector) (Yaskawa Specifications)	
Signal Name	Pin Number		Pin Number	Signal Name
FG	1		1	FG
SD (TXD)	2		2	SD (TXD)
RD (RXD)	3		3	RD (RXD)
RS (RTS)	4		4	RS (RTS)
CS (CTS)	5		5	CS (CTS)
–	6		6	DR (DSR)
SG (GND)	7		7	SG (GND)
–	8		8	–
ER (DTR)	9		9	ER (DTR)

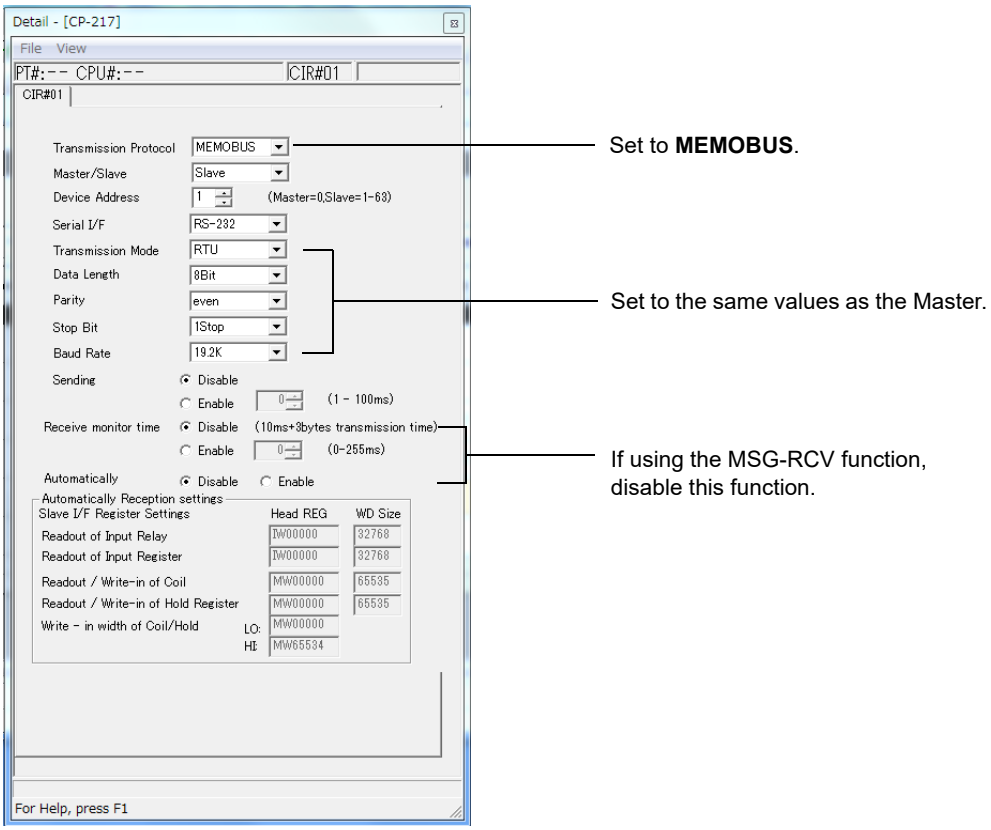
(3) 217IF Transmission Settings for the Master

Set the transmission definitions in the **217IF Transmission Configuration** Window (RS-232C) as shown below.



(4) 217IF Transmission Settings for the Slave

Set the transmission definitions in the **217IF Transmission Configuration** Window (RS-232C) as shown below.



(5) Procedure to Start Communication

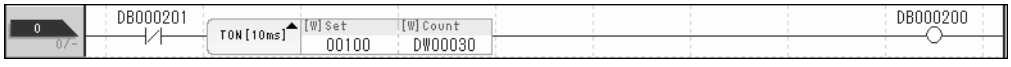
Use the following procedure to start communication.

1. Start executing the MSG-RCV function in the Slave 217IF Module.
2. Turn ON the **Execute** parameter in the MSG-SND function in the Master 217IF Module to send a message.

Example ladder programming using the MSG-SND and MSG-RCV functions is shown below in item (6).
In *[b] Ladder Programming Example Using the Message Receive Function (MSG-RCV)* on page 3-26, MSG-RCV function execution is started with SB000004 (Always ON Coil). Execution of the MSG-RCV function will thus start as soon as the power is turned ON to the Slave MP2000 Series Machine Controller. In *[a] Ladder Programming Example Using the Message Send Function (MSG-SND)* on page 3-25, a message will be sent as soon as DB000201 (the register set for **Execute** in the MSG-SND function) turns ON, thus starting communication with the Slave 217IF Module.

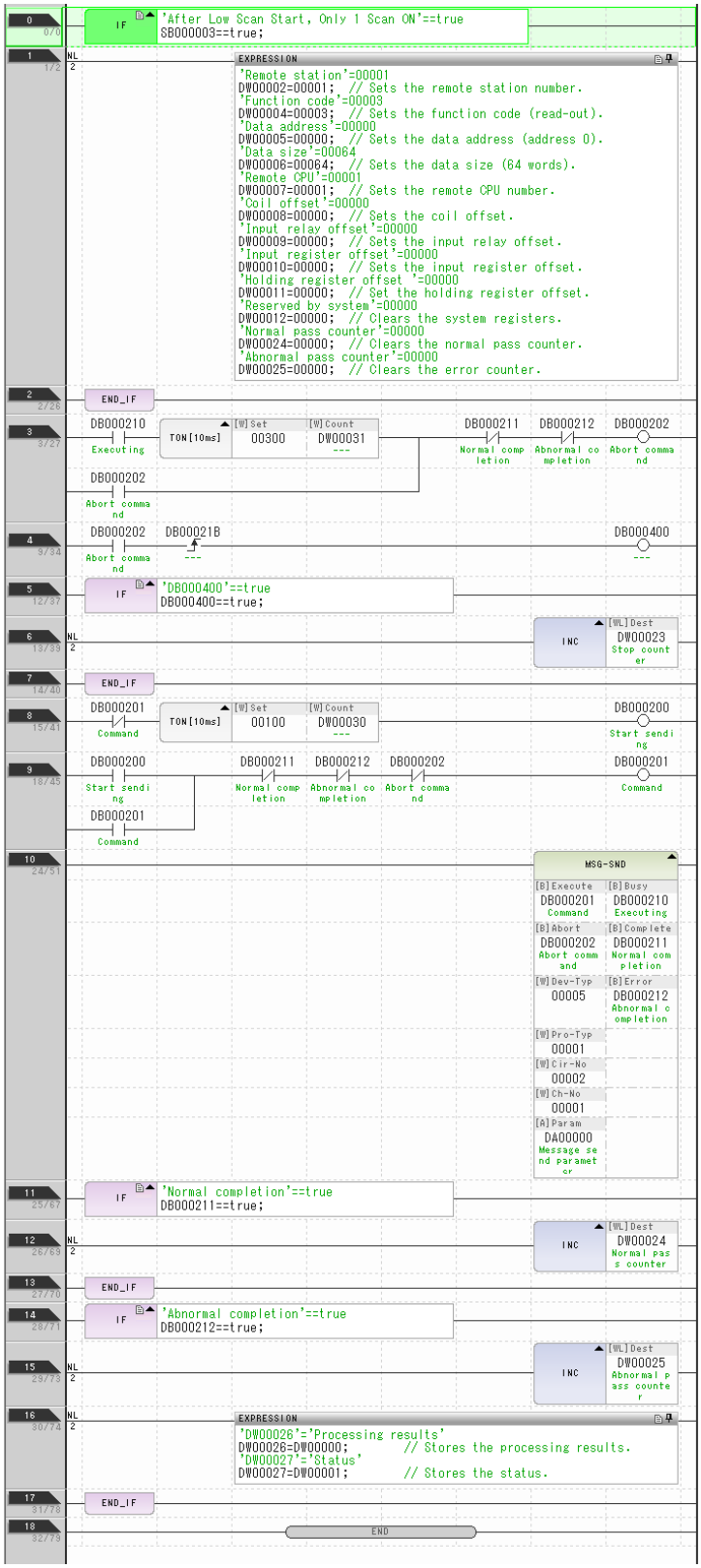
I/O Definition	No.	Name	Setting Example	Description
Input	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

- In *[a] Ladder Programming Example Using the Message Send Function (MSG-SND)* on page 3-25, the programming is written so that a message will be sent every second after one second has expired after the low-speed (or high-speed) scan has started. To change the interval at which messages are sent, change the set value of the timer shown below.



(6) Programming Example

[a] Ladder Programming Example Using the Message Send Function (MSG-SND)



Sets parameters in the first scan after startup.
(Low-speed scan (DWG.L): SB000003,
High-speed scan (DWG.H): SB000001)
Sets PARAM02 to PARAM11, clears the system registers, and clears the error counter.

Forced stop if processing is not completed within 3 seconds after sending the Send command.

Increments the stop counter when the Abort command is sent.

Starts sending every 1 second.

Function Send command

Sends a message to the remote station. The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: Always 1 (MEMOBUS communication)
Cir-No. (circuit number): 2 (connection port)
Ch-No. (channel number): 1 (always 1 for the 217IF-01)

Turns ON after the command message is sent and the response message is received.
Clears the normal pass counter.

Ended with transmission errors.

Increments the abnormal pass counter.

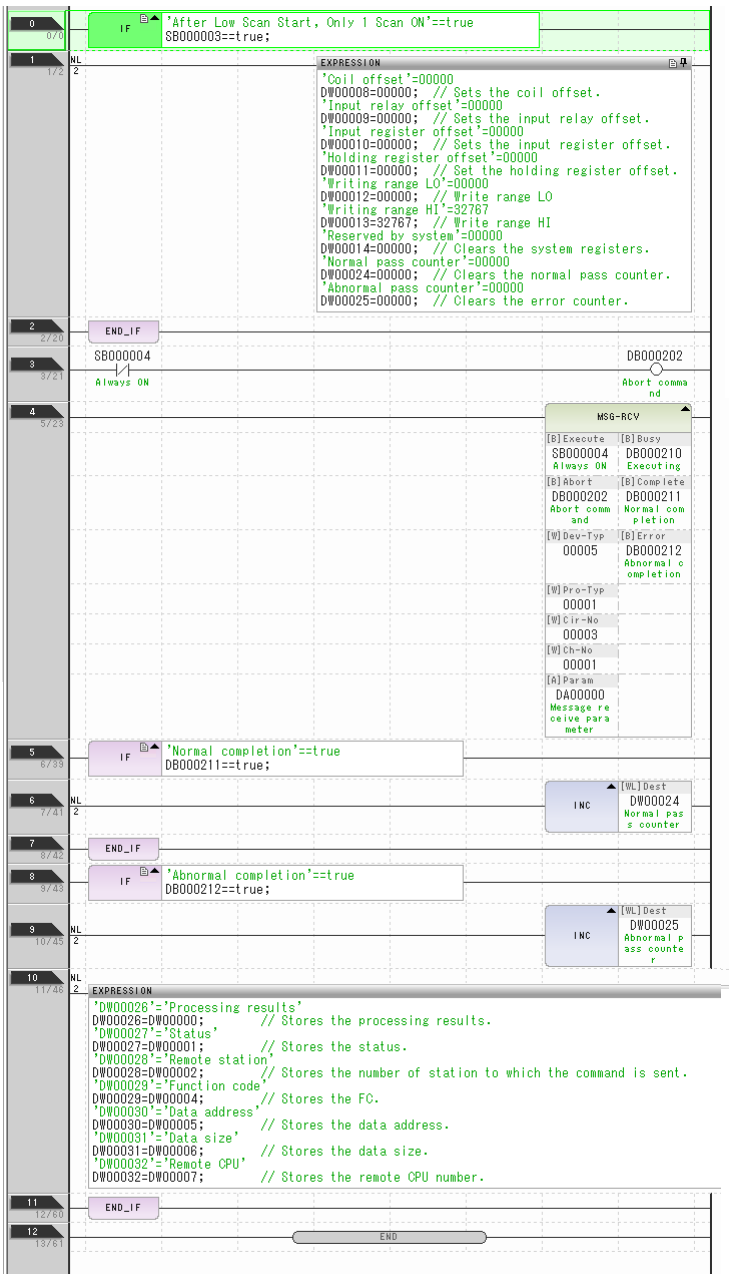
Stores the processing results and the status.

[b] Ladder Programming Example Using the Message Receive Function (MSG-RCV)

The MSG-RCV function is necessary if **Disable** is selected for **Automatically Reception** in the setting for the 217IF transmission system.

In this example, the number of the D registers is changed to 33. (Initial setting of the D registers is 32: DW00000 to DW00031.)

To change the number of registers, use the **Program Property** window of **MPE720**.



Sets parameters in the first scan after startup.

(Low-speed scan (DWG.L): SB000003,
High-speed scan (DWG.H): SB000001)

Sets PARAM08 to PARAM13, clears the system registers, and clears the error counter.

Parameter settings completed.

Always turns the Abort command OFF.

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 1 (MEMOBUS communication)
Cir-No. (circuit number): 3 (connection port)
Ch-No. (channel number): 1 (always 1 for the 217IF-01)

The normal completion coil turns ON at normal completion.

Increments the normal pass counter.

The abnormal completion coil turns ON at abnormal completion.

Increments the abnormal pass counter.

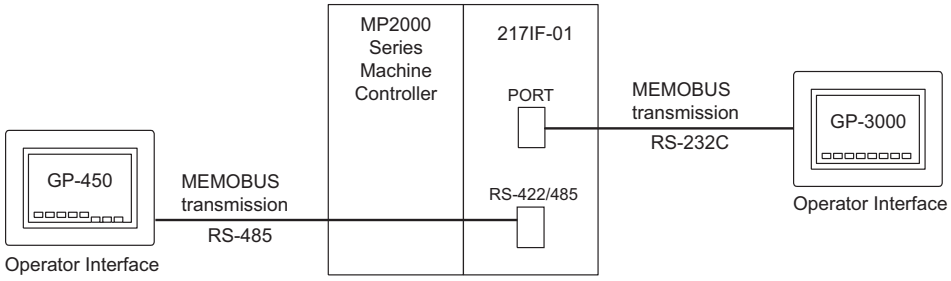
Stores the processing results, the status, the number of station to which the command is sent, the FC, the data address, the data size, and the remote CPU number.

3.5.2 Connecting to an Operator Interface Using MEMOBUS Communication

This section describes connection of an Operator Interface that is compatible with MEMOBUS communication.

(1) System Configuration Example

In the following configuration, a GP-3000 Operator Interface from Digital Electronics Corporation (commercially available) is connected to the RS-232C port of a 217IF-01 Module and a GP-450 Operator Interface from Digital Electronics Corporation (commercially available) is connected to the RS-485 port of the same Module. These Operator Interfaces are used to monitor the status of the MP2000 Series Machine Controller.

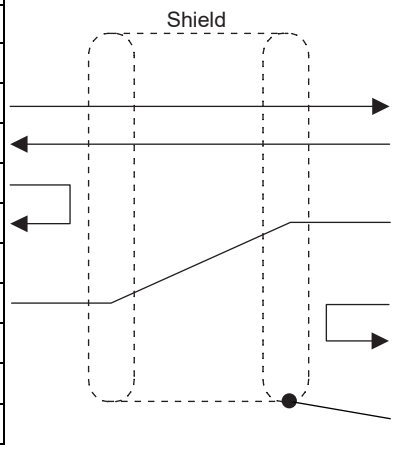


(2) Cable Specifications

■ PORT Cable

MP2000 Series Machine Controller (PORT Connector)	
Signal Name	Pin Number
FG	1
SD (TXD)	2
RD (RXD)	3
RS (RTS)	4
CS (CTS)	5
DR (DSR)	6
SG (GND)	7
–	8
ER (DTR)	9
–	–

Cable Connections and Signal Directions

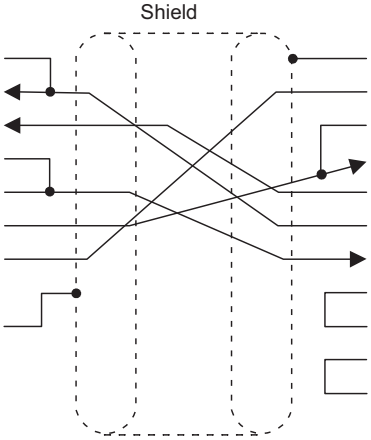


GP3000 Connector	
9-pin D-sub Connector	
Pin Number	Signal Name
1	CD
2	RD
3	SD
4	ER
5	SG
6	DR
7	RS
8	CS
9	CI
Shell	FG

■ RS-422/485 Cable

217IF-01	
14-pin MDR Connector	
Signal Name	Pin Number
RXR	7
RX-	4
RX+	3
TXR	11
TX-	2
TX+	1
GND	14
–	–
Connector shell	

Cable Connections and Signal Directions



GP	
25-pin D-sub Connector	
Pin Number	Signal Name
1	FG
7	SG
9	–
10	RDA
11	SDA
15	SDB
16	RDB
18	–
19	–
21	–
22	–

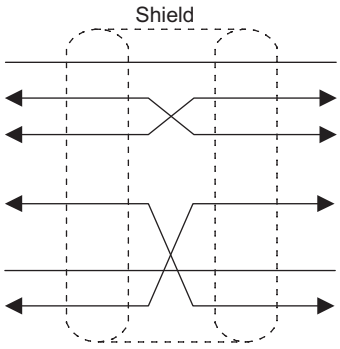
- Connect the cable so that the terminator for the 217IF-01 Module is used.
- The connector shell of the RS-422/485 port is connected to the FG terminal of the MP2000 Series Machine Controller.

■ Connection with GP450 through RS-232C Communication

The GP450 can also monitor the status of the MP2000 Series Machine Controller through RS-232C communication.
The cable connections between the GP450 and the RS-232C port (PORT) are shown below.

217IF-01	
9-pin D-sub Connector	
Signal Name	Pin Number
FG	1
SD (TXD)	2
RD (RXD)	3
RS (RTS)	4
CS (CTS)	5
–	6
SG (GND)	7
ER (DTR)	9

Cable Connections and Signal Directions

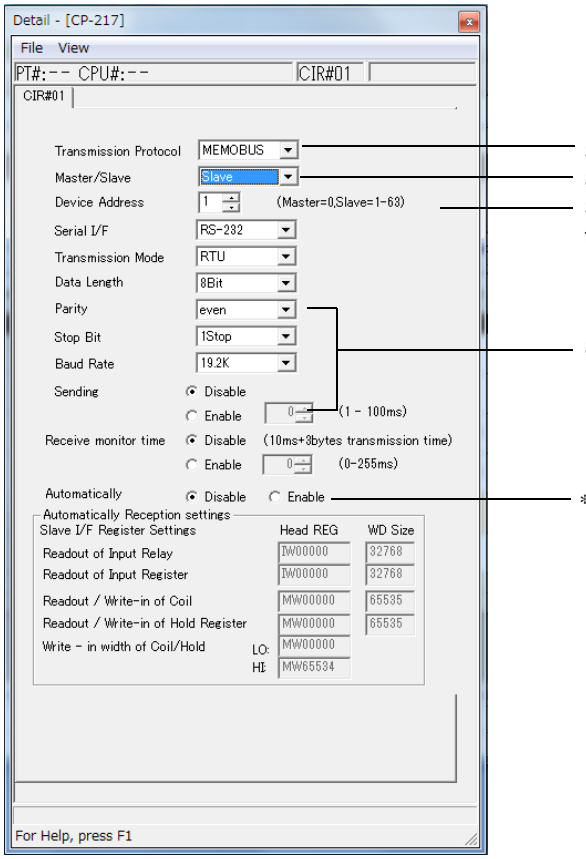


GP	
25-pin D-sub Connector	
Pin Number	Signal Name
1	FG
2	SD (TXD)
3	RD (RXD)
4	RS (RTS)
5	CS (CTS)
6	DSR (DR)
7	SG (GND)
20	DTR (ER)

(3) 217IF Transmission Settings

[a] 217IF Transmission Definitions for RS-232C Port

Select **MEMOBUS** for the communication protocol in the **217IF Transmission Configuration** Window (RS-232C).

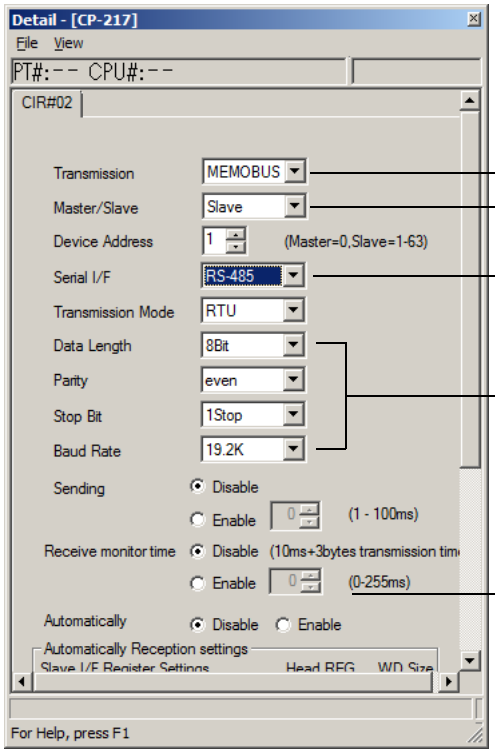


Set to **MEMOBUS**.
Set to **Slave** (the Operator Interface is the Master).
Set according to the unit number of the Operator Interface.
Set to the same values as the Operator Interface.
*

* If automatic reception is enabled, the system will interrupt the low-speed scan every 50 ms and perform the same processing as is performed for MSG-RCV. Therefore, if automatic reception is enabled, the programming example in *(6) Programming Example* on page 3-31 is not required.
If automatic reception is disabled, ladder program as shown in the programming example in *(6) Programming Example* on page 3-31 is required. The ladder program can be executed in the high-speed scan to increase the response speed.

[b] 217IF Transmission Definitions for RS-422/485 Port

Select **MEMOBUS** for the communication protocol in the **217IF Transmission Configuration** Window (RS-422/485).



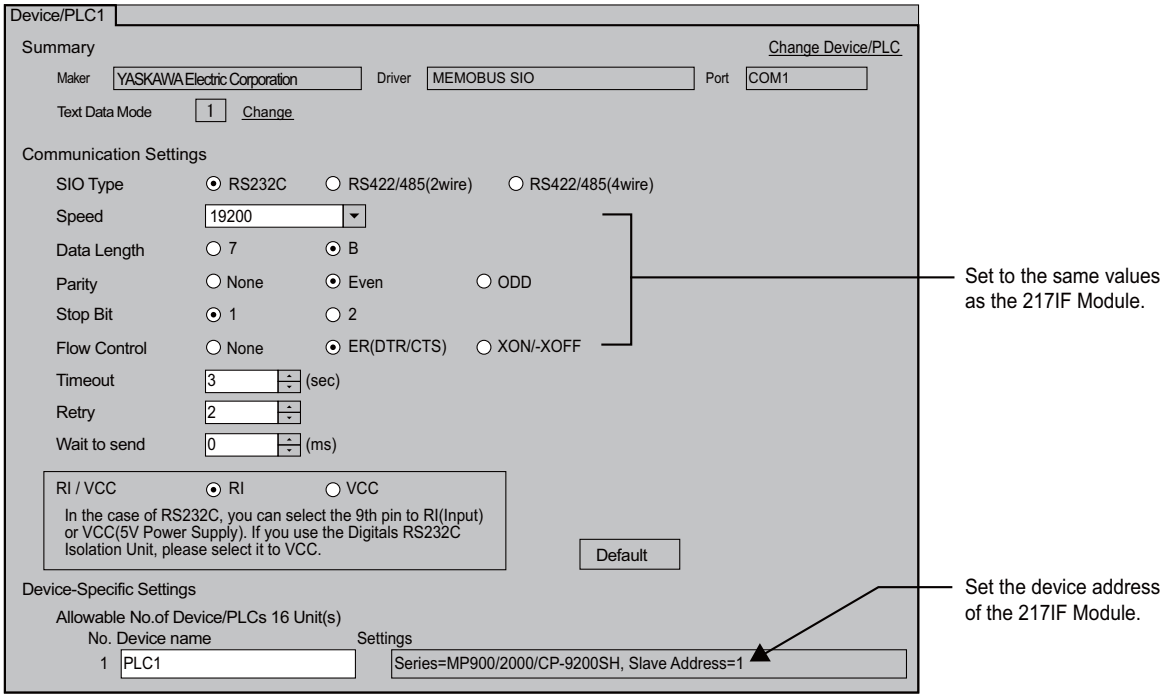
Set to **MEMOBUS**.
Set to **Slave** (the Operator Interface is the Master).
Set according to the unit number of the Operator Interface.
Always set to RS-485. *1
Set to the same values as the Operator Interface.
*2

* 1. The serial interface is always set to **RS-485** and cannot be switched between RS-422 and RS-485. Use the 485 DIP switch on the 217IF-01 Module to switch between RS-422 and RS-485 (ON = RS-485, OFF = RS-422). Even if the DIP switch is set for RS-422, the serial interface parameter setting will show **RS-485**.
Refer to *6.1.5 Switch Settings* on page 6-7 for information on the DIP switch on the 217IF-01 Module.
* 2. If automatic reception is enabled, the system will interrupt the low-speed scan every 50 ms and perform the same processing as is performed for MSG-RCV. Therefore, if automatic reception is enabled, the programming example in *(6) Programming Example* on page 3-31 is not required.
If automatic reception is disabled, ladder program as shown in the programming example in *(6) Programming Example* on page 3-31 is required. The ladder program can be executed in the high-speed scan to increase the response speed.

(4) Transmission Settings on the Operator Interface

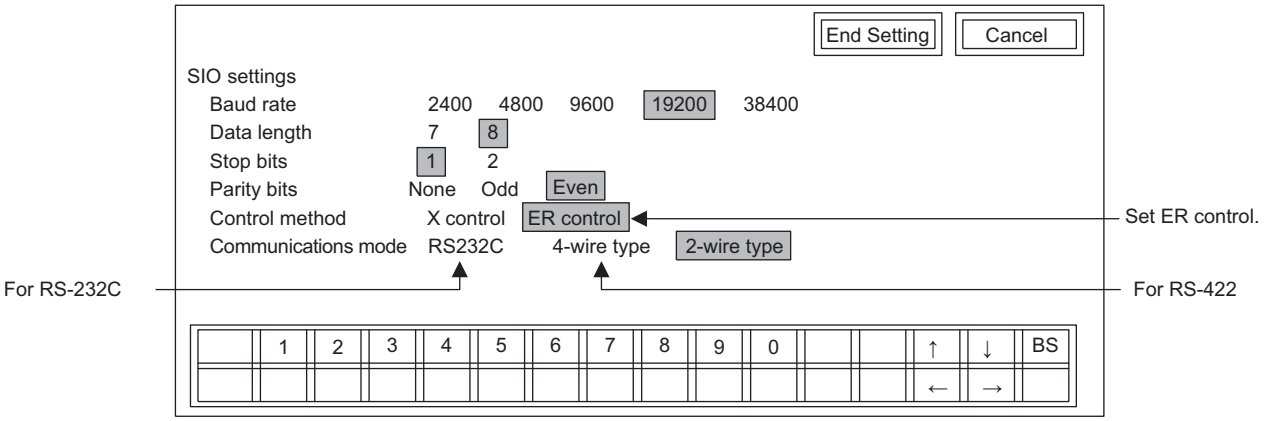
[a] GP-3000 Settings

The following figure shows the settings for the GP-3000.
When setting the Operator Interface, set the maker to **Yaskawa Electric Corporation** and set the Driver to **MEMOBUS SIO**.



[b] GP-450 Settings

The following figure shows the settings for the GP-450.



(5) Procedure to Start Communication

Use the following procedure to start communication.

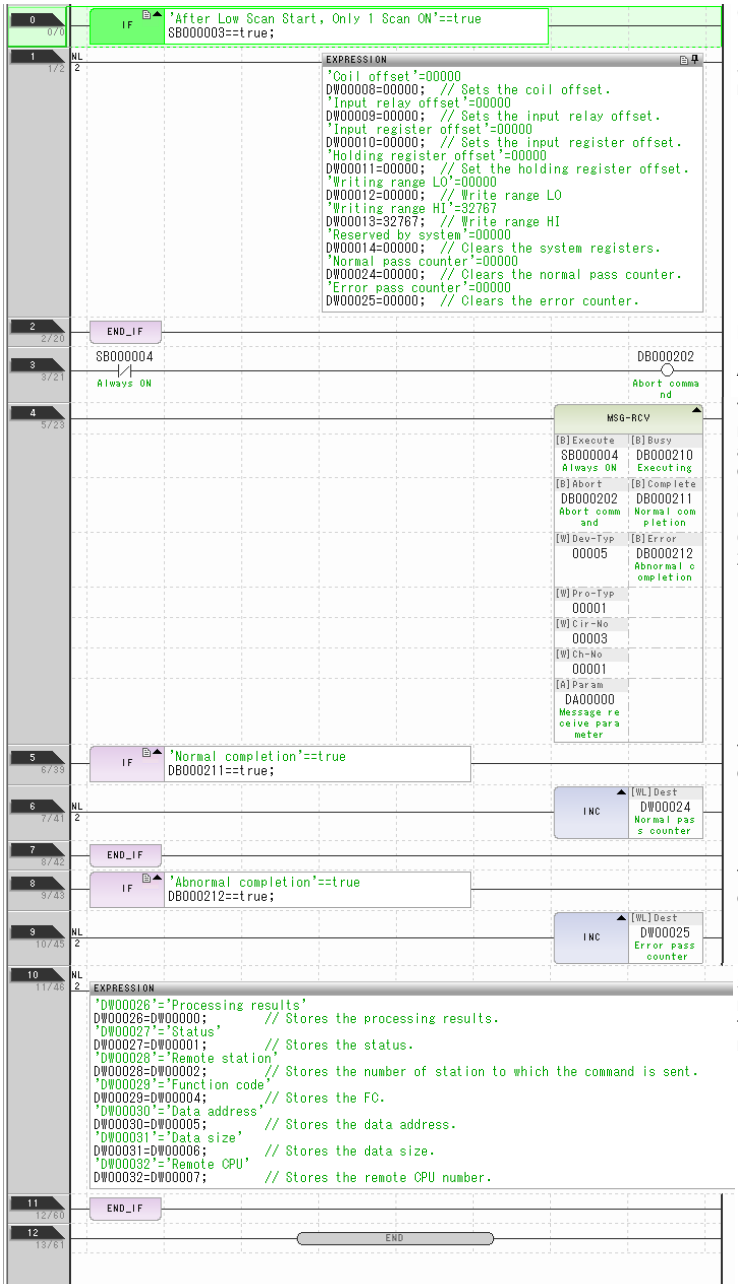
1. Start executing the Message Receive function at the MP2000 Series Machine Controller.

2. Start sending messages from the Operator Interface.

Example ladder programming using the MSG-RCV function is shown in *(6) Programming Example* on page 3-31. In the programming example, MSG-RCV function execution is started with SB000004 (Always ON Coil). Execution of the MSG-RCV function will thus start as soon as the power is turned ON to the Slave MP2000 Series Machine Controller.

(6) Programming Example

Ladder programming that uses the MSG-RCV function is shown below. The MSG-RCV function is necessary if **Disable** is selected for **Automatically Reception** in the settings for the 217IF transmission system.



Sets parameters in the first scan after startup.

(Low-speed scan (DWG.L): SB000003,
High-speed scan (DWG.H): SB000001)

Sets PARAM08 to PARAM13, clears the system registers, and clears the error counter.

Parameter settings completed.

Always turns the Abort command OFF.

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 1 (MEMOBUS communication)
Cir-No. (circuit number): 3 (connection port)
Ch-No. (channel number): 1 (always 1 for the 217IF-01)

The normal completion coil turns ON at normal completion.

Increments the normal pass counter.

The abnormal completion coil turns ON at abnormal completion.

Increments the abnormal pass counter.

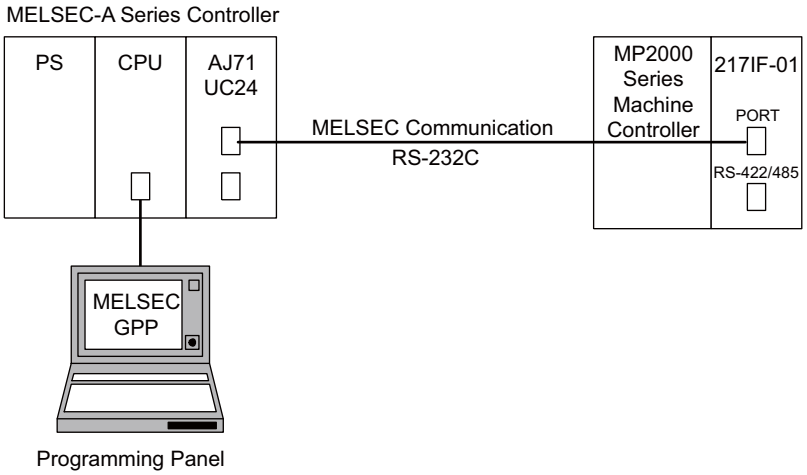
Stores the processing results, the status, the number of station to which the command is sent, the FC, the data address, the data size, and the remote CPU number.

3.5.3 Connection to MELSEC

This section describes communication with a MELSEC-A Series Programmable Logic Controller (MELSEC Master).

(1) System Configuration Example

In the following configuration, the MP2000 Series Machine Controller is connected with a Mitsubishi Electric MELSEC A-series Programmable Logic Controller via the RS-232C (PORT) port of the 217IF-01 Module to monitor data in the MELSEC A-series Controller. The MP2000 Series Machine Controller is the Master and the MELSEC-A is the Slave.



(2) Cable Specifications

■ PORT Cable

217IF-01		Cable Connections and Signal Directions	MELSEC	
9-pin D-sub Connector			9-pin D-sub Connector	
Signal Name	Pin Number		Pin Number	Signal Name
FG	1	Shield	1	CD
SD (TXD)	2		2	RD
RD (RXD)	3		3	SD
RS (RTS)	4		4	DTR
CS (CTS)	5		5	SG
–	6		6	DSR
SG (GND)	7		7	RS
ER (DTR)	9		8	CS

- Mitsubishi Electric
  Model: A1SJ71C24

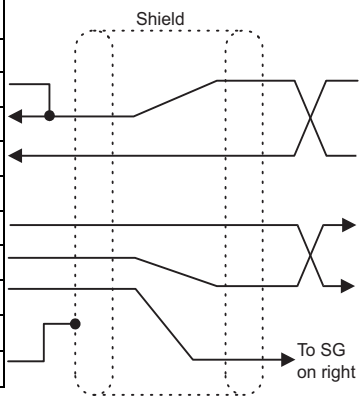
■ Connection to MELSEC through RS-485 Communication

With the 217IF-01 Module, communication with a MELSEC A-series Programmable Logic Controller is also possible using the RS-485 port. An example of connecting the RS-485 port (CN2) is shown below.

CN2 Cable

217IF-01	
MDR-14 Connector	
Signal Name	Pin Number
RXR	7
RX-	4
RX+	3
TXR	11
TX-	9
TX+	8
GND	14
Connector shell	

Cable Connections and Signal Directions



Screw Terminal Block *	
Signal Name	
Left	Right
SD A	
	SG
SD B	
	FG
RD A	
	N.C.
RD B	

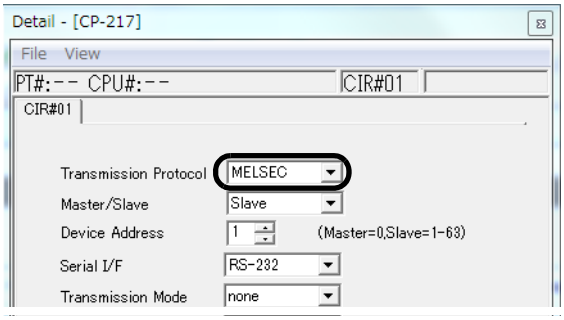
- MELSEC-A series (manufactured by Mitsubishi Electric)
  Model: A1SJ71C24
  AJ71UC24
  A171QC24
  A1SJ71UC24-R4/R2*

- As shown in the above wiring diagram, connect the cable so that the terminator for the 217IF-01 Module is used.
- The RS-422/485 connector shell is connected to the FG terminal of the MP2000 Series Machine Controller.

* If the R2 model is used, the remote station number set in the MSG-SND function must be H0100 because the station number of R2 is always 0.

(3) 217IF Transmission Settings

Select **MELSEC** for the communication protocol in the **217IF Transmission Configuration** Window (RS-232C).



(4) Setting the MELSEC Module (Example for AJ71UC24)

[a] Buffer Memory Settings

Change the two settings shown in the following table.
Buffer memory is not backed up, so the settings must be written using the user program.

Address	Name	Default Setting	Recommended Setting
10BH	RS-232C CD terminal check setting area	0 (Check)	1 (No check)
11AH	Transmission control specification area	0 (DTR control)	1 (No DTR control)

- Use the default setting for other addresses.
- The above addresses are used when the 217IF-01 Module is mounted in slot 1. These addresses change if another slot is used.

[b] Switch Settings

■ Station Number Setting Switches

Set a value between 01 and 31. (Recommended setting: 01)

Switch Name	Setting Item	Recommended Setting
10 (rotary switch: 0 to 9)	The 10s digit of the station number	0
1 (rotary switch: 0 to 9)	The 1s digit of the station number	1

■ Transmission Specifications Setting Switches (Example of AJ71UC24 Module)

Switch Name	Setting Item	Setting Contents		Set Value*1
		ON	OFF	
SW11	Main channel setting	RS-422	RS-232C	OFF
SW12	Data bit setting	8 bits	7 bits	ON
SW13	Baud rate setting	See the following table.*2		OFF
SW14				ON
SW15				ON
SW16	Parity bit setting	Yes	No	ON
SW17	Even or odd parity	Even	Odd	ON
SW18	Stop bit setting	2 bits	1 bit	OFF
SW21	Checksum setting	Yes	No	ON
SW22	Write enable/disable during RUN	Enable	Disable	ON
SW23	Computer link or multi-drop link	Computer link	Multi-drop link	ON
SW24	Not used.	−		OFF

* 1. The settings in shaded cells are example settings.
* 2. The relationship of the SW13, SW14, and SW15 settings and the baud rate is as follows:

bps	300	600	1200	2400	4800	9600	19200
SW13	OFF	ON	OFF	ON	OFF	ON	OFF
SW14	OFF	OFF	ON	ON	OFF	OFF	ON
SW15	OFF	OFF	OFF	OFF	ON	ON	ON

- With the AJ71C24-S8, the ON/OFF settings for the terminators on the sending side and receiving side depend on the wiring.

■ Mode Setting Switches

Switch Name	Setting	Port Operation Mode			Set Value
		RS-232C Port		RS-422/485 Port	
MODE (rotary switch: 0 to F)	0	Cannot be used.			RS-232C connection: 1 RS-422/485 connection: 5
	1	Type 1 protocol mode		Non-procedure protocol	
	2	Type 2 protocol mode		Non-procedure protocol	
	3	Type 3 protocol mode		Non-procedure protocol	
	4	Type 4 protocol mode		Non-procedure protocol	
	5	Non-procedure protocol		Type 1 protocol mode	
	6	Non-procedure protocol		Type 2 protocol mode	
	7	Non-procedure protocol		Type 3 protocol mode	
	8	Non-procedure protocol		Type 4 protocol mode	
	9	Non-procedure protocol	↔	Non-procedure protocol	
	A	Type 1 protocol mode	↔	Type 1 protocol mode	
	B	Type 2 protocol mode	↔	Type 2 protocol mode	
	C	Type 3 protocol mode	↔	Type 3 protocol mode	
	D	Type 4 protocol mode	↔	Type 4 protocol mode	
	E	Cannot be used.			
	F	For unit test			

(5) Procedure to Start Communication

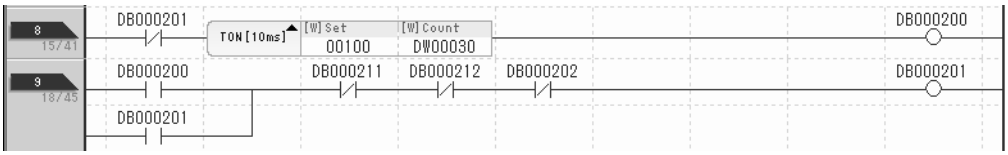
Use the following procedure to start communication.

1. Make the settings for serial communication at the MELSEC Controller and enable message reception.
2. Start sending messages from the 217IF Module of the MP2000 Series Machine Controller.

Example ladder programming using the MSG-SND function is shown in *(6) Programming Example* on page 3-36. In this programming example, a message is sent when DB000201 turns ON, starting communication with the MELSEC Controller.

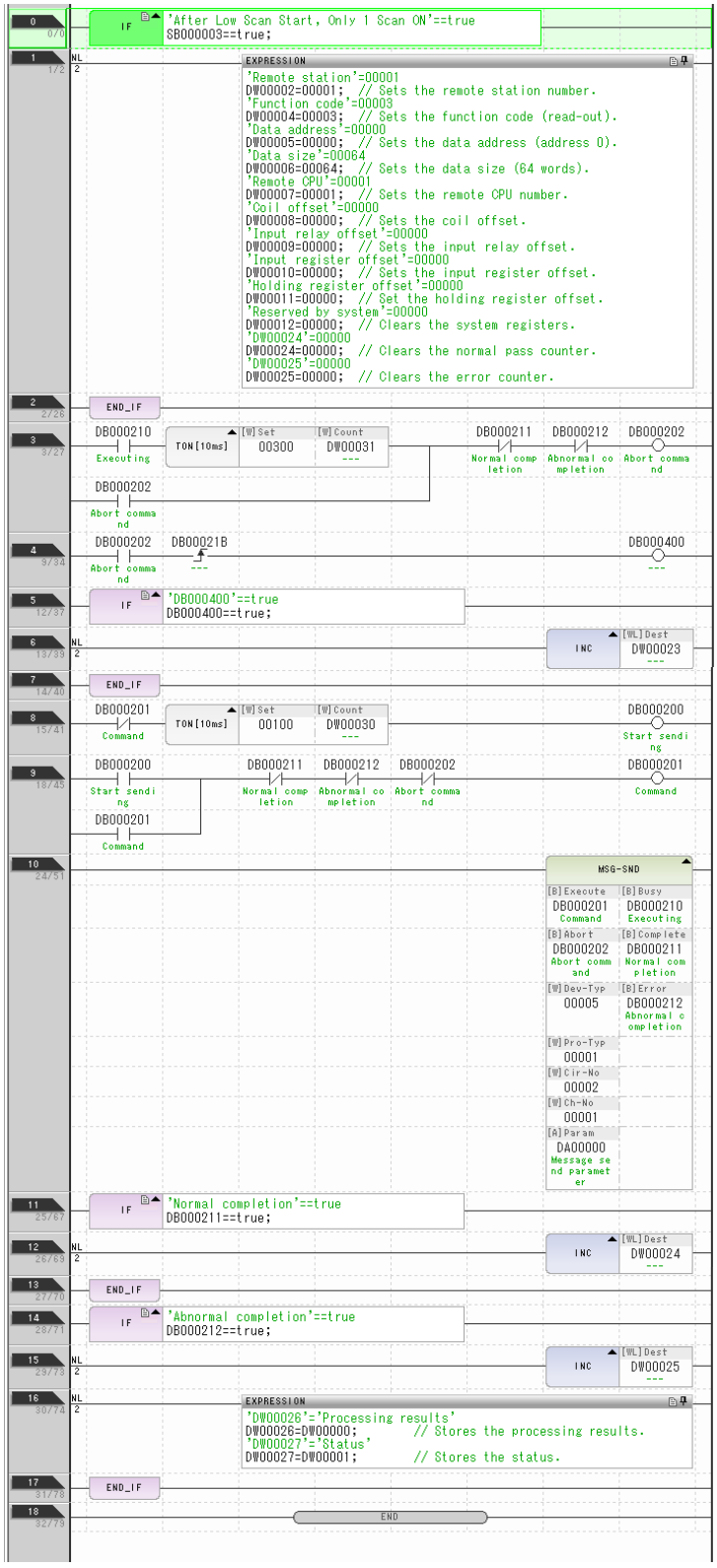
I/O Definition	No.	Name	Setting Example	Description
Input	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

- In *(6) Programming Example* on page 3-36, the programming is written so that a message will be sent every second after one second has expired after the low-speed (or high-speed) scan has started. To change the interval at which messages are sent, change the set value of the timer shown below.



(6) Programming Example

An example of a ladder program that uses the MSG-SND function is shown below. No programming is required in the MELSEC-A Controller.



Sets parameters in the first scan after startup.
(Low-speed scan (DWG.L): SB000003,
High-speed scan (DWG.H): SB000001)

Sets PARAM02 to PARAM11, clears the system registers, and clears the error counter.

Forced stop if processing is not completed within 3 seconds after sending the Send command.

Increments the stop counter when the Abort command is sent.

Starts sending every 1 second.

Function Send command

Sends a message to the remote station. The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: Always 1 (MEMOBUS communication)
Cir-No. (circuit number): 2 (connection port)
Ch-No. (channel number): 1 (always 1 for the 217IF-01)

Turns ON after the command message is sent and the response message is received.

Clears the normal pass counter.

Ended with transmission errors.

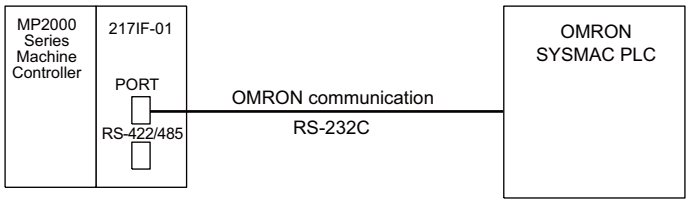
Stores the processing results and the status.

3.5.4 Connection to an OMRON PLC

This section describes how to connect to an OMRON PLC.

(1) System Configuration Example

In this example, an OMRON SYSMAC PLC is connected to the PORT connector (RS-232C) of the Communication Module.



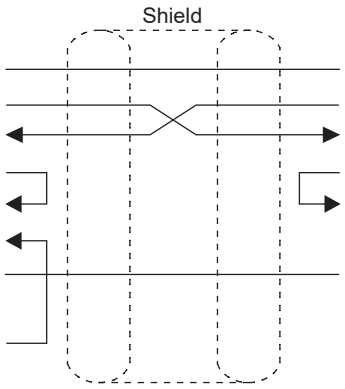
(2) PORT Cable Specifications

The pin arrangement of the PORT connector (RS-232C) is the same for all Communication Modules.

- To enable the terminator (120 Ω) for a Communication Module, connect RXR and RX (-), and connect TXR and TX(-).
- Insert terminators only at the stations at both ends of the system.

Communication Module PORT	
9-pin D-sub Connector	
Signal Name	Pin Number
FG	1
TXD	2
RXD	3
RTS	4
CTS	5
DSR	6
SG	7
–	8
DTR	9

Cable Connections and Signal Directions

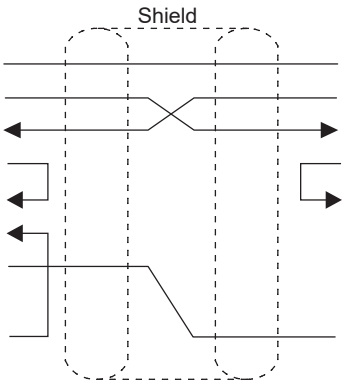


OMRON	
9-pin D-sub Connector	
Pin Number	Signal Name
1	FG
2	SD
3	RD
4	RS
5	CS
6	5V
7	SG
8	–
9	–

- SYSMAC manufactured by OMRON
  Model: C20H
  C28H
  C40H

Communication Module PORT	
9-pin D-sub Connector	
Signal Name	Pin Number
FG	1
TXD	2
RXD	3
RTS	4
CTS	5
DSR	6
SG	7
–	8
DTR	9

Cable Connections and Signal Directions

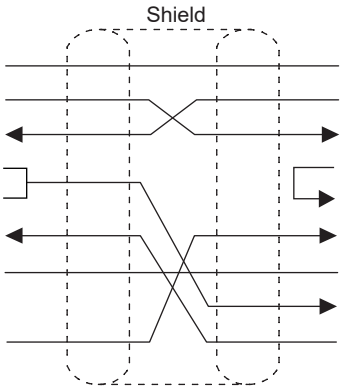


OMRON	
9-pin D-sub Connector	
Pin Number	Signal Name
1	FG
2	SD
3	RD
4	RS
5	CS
6	5V
7	DR
8	ER
9	SG

- SYSMAC manufactured by OMRON
  Model: CS1G-CPU□□-V1
  CS1H-CPU□□-V1

Communication Module PORT	
9-pin D-sub Connector	
Signal Name	Pin Number
FG	1
TXD	2
RXD	3
RTS	4
CTS	5
DSR	6
SG	7
–	8
DTR	9

Cable Connections and Signal Directions



OMRON	
9-pin D-sub Connector	
Pin Number	Signal Name
1	FG
2	SD
3	RD
4	RS
5	CS
6	DSK
7	SG
8	CD
20	DTR

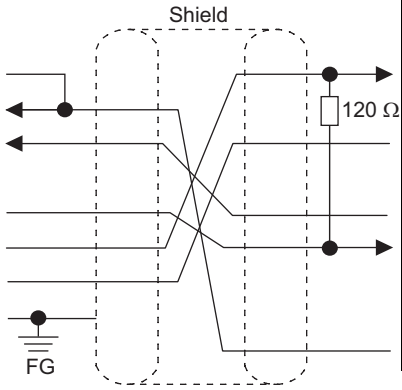
- SYSMAC manufactured by OMRON
  Model: C200H-ASC02
  C200H-LK201-V1
  C500-LK201-V1
  C120-LK201-V1L

■ RS-422/485 Communication Cables

The 217IF-01 Module can be connected to an OMRON PLC through RS-422/485 transmissions.
The wiring diagram of RS-422/485 communication cable is shown below.

Communication Module RS-422/485	
14-pin MDR Connector	
Signal Name	Pin Number
RXR	7
RX (–)	4
RX (+)	3
TXR	11
TX	9
TX	8
SG	14
SH	

Cable Connections and Signal Directions

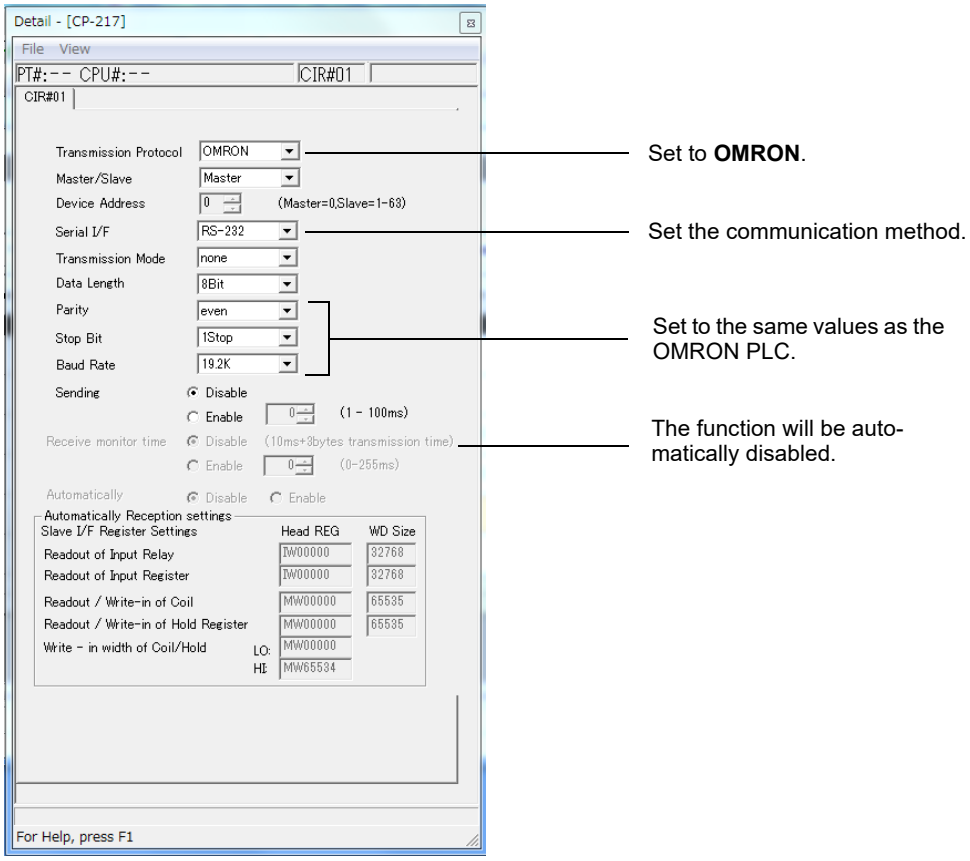


OMRON	
9-pin D-sub Connector	
Pin Number	Signal Name
1	RD B
2	–
3	SG
4	–
5	SD B
6	RD A
7	FG
8	–
9	SD A

- SYSMAC manufactured by OMRON
  Model: C200H-LK202-V1
  C500-LK201-V1
  C120-LK202-V1

(3) 217IF Transmission Settings

Set the transmission definitions in the **217IF Transmission Configuration** Window (RS-232C) as shown below.



(4) OMRON PLC Settings

The settings for the OMRON PLC depend on the model being used. Refer to user documentation provided by OMRON.

Item	Recommended Setting
Baud rate	Set to the same value as the Communication Module.
Start bits	1 bit
Data length	7 or 8 bits (Set to same value as the 217IF.)
Stop bits	1 or 2 bits (Set to same value as the 217IF.)
Parity	Even, odd, or none (Set to same value as the 217IF.)
Communications mode	Host Link
RS/CS control	None
Unit number	Set any value other than 0. *

* When the MP2000 Series Machine Controller is the Master, its device address will be 0. The OMRON PLC must therefore be set to a device address (unit number) other than 0. The default value is 0, so be sure to change this setting.

(5) Procedure to Start Communication

Use the following procedure to start communication.

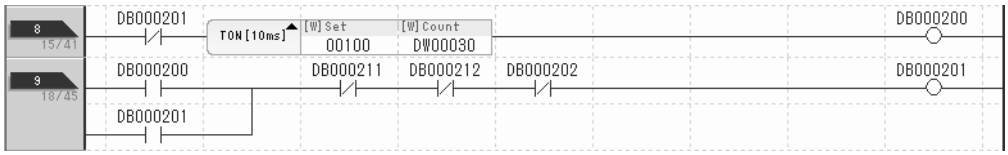
1. Make the settings for serial communication at the OMRON PLC and enable message reception.

2. Start sending messages from the 217IF Module of the MP2000 Series Machine Controller.

Example ladder programming using the MSG-SND function is shown in *(6) Programming Example* on page 3-41. In this programming example, a message is sent when DB000201 turns ON, starting communication with the OMRON PLC.

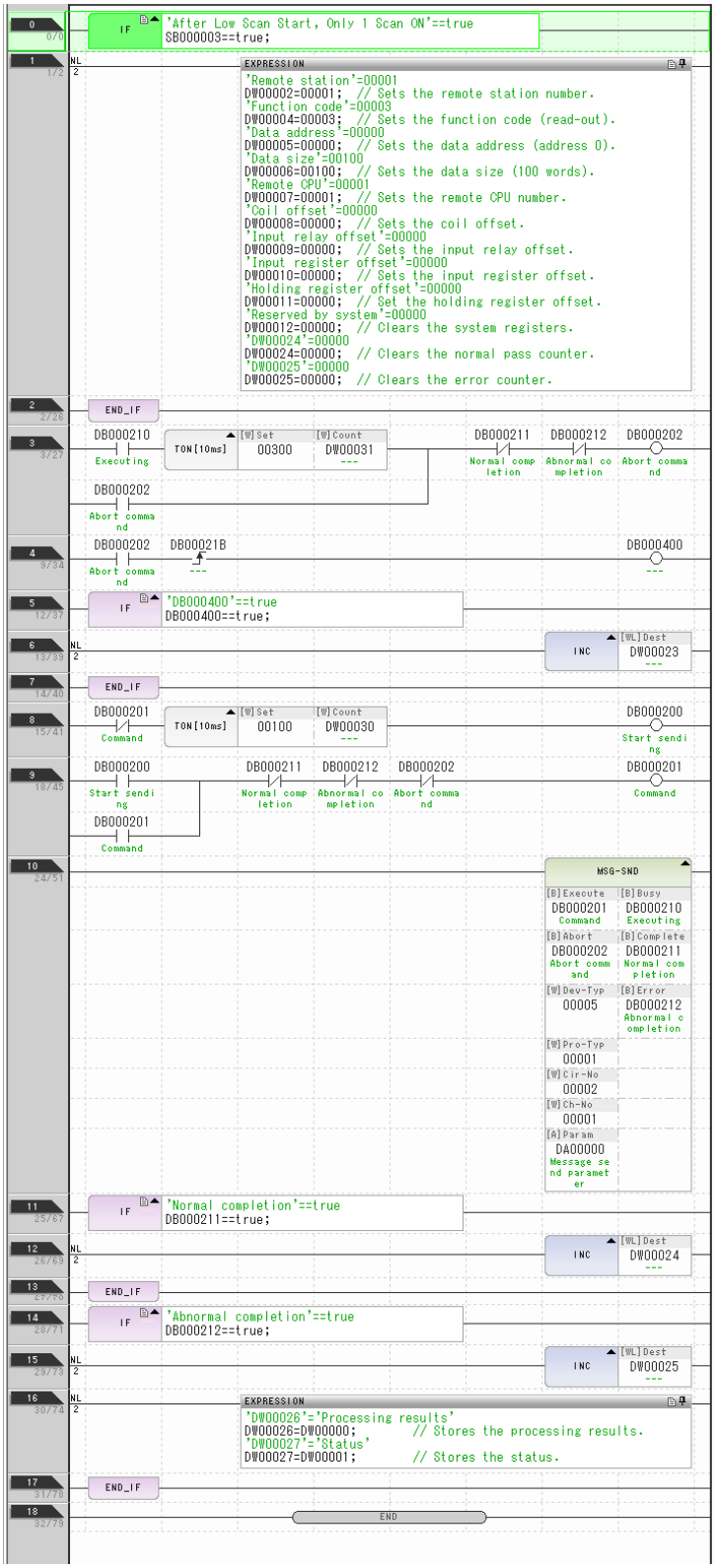
I/O Definition	No.	Name	Setting Example	Description
Input	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

- In *(6) Programming Example* on page 3-41, the programming is written so that a message will be sent every second after one second has expired after the low-speed (or high-speed) scan has started. To change the interval at which messages are sent, change the set value of the timer shown below.



(6) Programming Example

An example of a ladder program that uses the MSG-SND function is shown below. No programming is required in the OMRON PLC.



Sets parameters in the first scan after startup. (Low-speed scan (DWG.L): SB000003, High-speed scan (DWG.H): SB000001)

Sets PARAM02 to PARAM11, clears the system registers, and clears the error counter.

Forced stop if processing is not completed within 3 seconds after sending the Send command.

Increments the stop counter when the Abort command is sent.

Starts sending every 1 second.

Function Send command

Sends a message to the remote station. The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: Always 1 (MEMOBUS communication)
Cir-No. (circuit number): 2 (connection port)
Ch-No. (channel number): 1 (always 1 for the 217IF-01)

Turns ON after the command message is sent and the response message is received.

Clears the normal pass counter.

Ended with transmission errors.

Increments the abnormal pass counter.

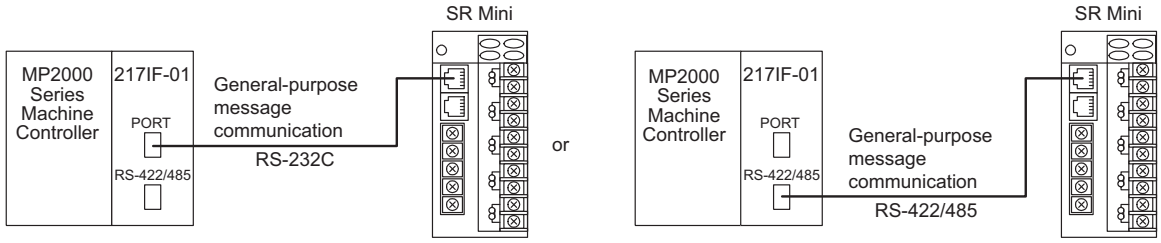
Stores the processing results and the status.

3.5.5 Connection to a Temperature Controller

This section describes an example of connection to a Temperature Controller (SR Mini) manufactured by RKC Instrument Incorporated.

(1) System Configuration Example

In this example, the SR Mini Temperature Controller is connected to the RS-232C (PORT) port or the RS-422/485 port of the 217IF-01 Module to read temperature data.

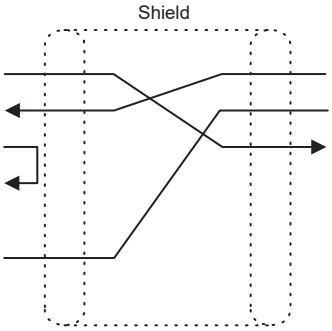


(2) Cable Specifications

■ PORT (RS-232C) Cable

Communication Module	
9-pin D-sub Connector	
Signal Name	Pin Number
FG	1
SD (TXD)	2
RD (RXD)	3
RS (RTS)	4
CS (CTS)	5
–	6
SG (GND)	7
ER (DTR)	9

Cable Connections and Signal Directions



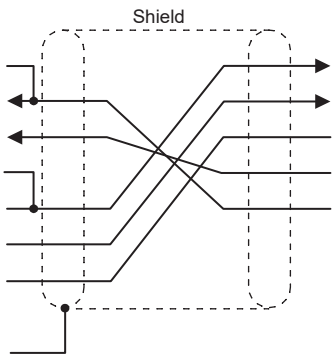
SR Mini	
6-pin Modular Connector	
Pin Number	Signal Name
1	–
2	SD
3	SG
4	RD
5	–
6	FG

- Three interfaces listed below are supported by the SR Mini. Specify the model when ordering.
  RS-232C
  RS-422
  RS-485

■ RS-422 Cable

217IF-01	
14-pin MDR Connector	
Signal Name	Pin Number
RXR	7
RX-	4
RX+	3
TXR	11
TX-	9
TX+	8
GND	14
Connector shell	

Cable Connections and Signal Directions



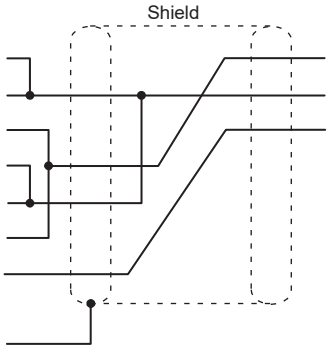
SR Mini	
6-pin Modular Connector	
Pin Number	Signal Name
1	R (A)
2	R (B)
3	SG
4	T (B)
5	T (A)
6	FG

- Connect the cable so that the terminator for the 217IF-01 Module is used.

■ RS-485 Cable

217IF-01	
14-pin MDR Connector	
Signal Name	Pin Number
RXR	7
RX-	4
RX+	3
TXR	11
TX-	9
TX+	8
GND	14
Connector shell	

Cable Connections



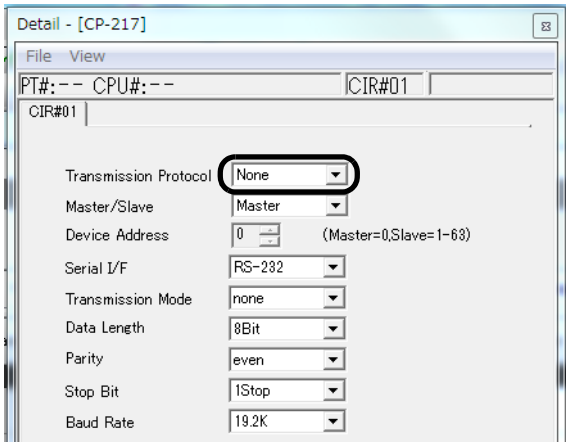
SR Mini	
6-pin Modular Connector	
Pin Number	Signal Name
1	R (A)
2	R (B)
3	SG
4	T (B)
5	T (A)
6	FG

RS-485

- Connect the cable so that the terminator for the 217IF-01 Module is used.

(3) 217IF Transmission Settings

Select **None (non-procedure)** for the communication protocol in the **217IF Transmission Configuration** Window (RS-232C or RS-422/485).



(4) Setting the Temperature Controller

The settings of the switches on the SR Mini Temperature Controller are shown in the following tables.

Bit 1	OFF	Always OFF.
Bit 2	OFF	Always OFF.
Bit 3	ON	Set according to the baud rate.
Bit 4	ON	Set according to the baud rate.

Bit 3	Bit 4	Baud Rate
OFF	OFF	2400 bps
OFF	ON	4800 bps
ON	OFF	9600 bps (factory default setting)
ON	ON	19200 bps

(5) Procedure to Start Communication

Use the following procedure to start communication.

1. Make the settings for serial communication at the Temperature Controller and enable communication.

2. Start sending messages from the 217IF Module of the MP2000 Series Machine Controller.

Example ladder programming using the MSG-SND and MSG-RCV functions is shown in *(6) Programming Example* on page 3-45.
In this programming example, a message is sent when the register set for **Execute** in the MSG-SND function (DB000201) turns ON. Also, the Communication Module will wait to receive a message when the register set for **Execute** in the MSG-RCV function (DB000601) turns ON.

<MSG-SND Function>

I/O Definition	No.	Name	Setting Example	Description
Input	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

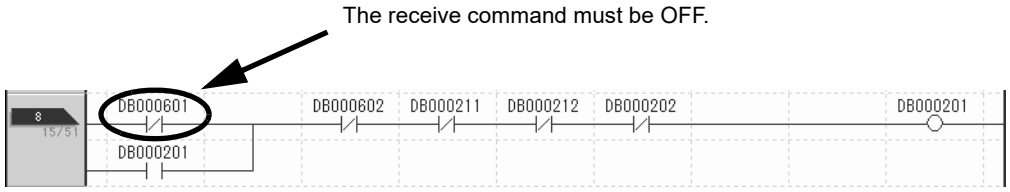
<MSG-RCV Function>

I/O Definition	No.	Name	Setting Example	Description
Input	1	Execute	DB000601	Message receive processing is started when the receive command (Execute) turns ON.

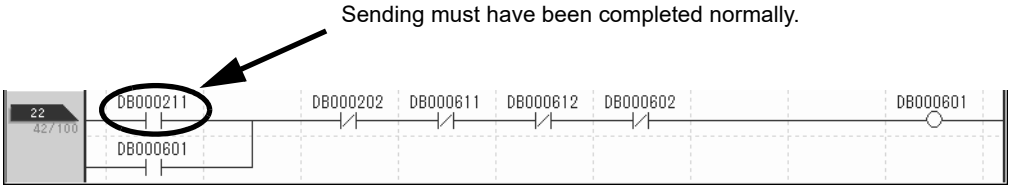
In *(6) Programming Example* on page 3-45, the programming is written so that the 217IF-01 Module will send a message once when the low-speed (or high-speed) scan is started. (See below for the execution conditions.)
If sending the message is completed normally, the MSG-RCV function will be executed to wait to receive a message. (See below for the execution conditions for receiving a message.)
If receiving the message is completed normally, the MSG-SND function will be executed again to send a message. In this way, the MSG-SND function and MSG-RCV function are executed alternately to repeatedly send and receive messages.

- Do not execute the MSG-SND and MSG-RCV functions at the same time.

<Execution Conditions for Sending a Message>



<Execution Conditions for Receiving a Message>

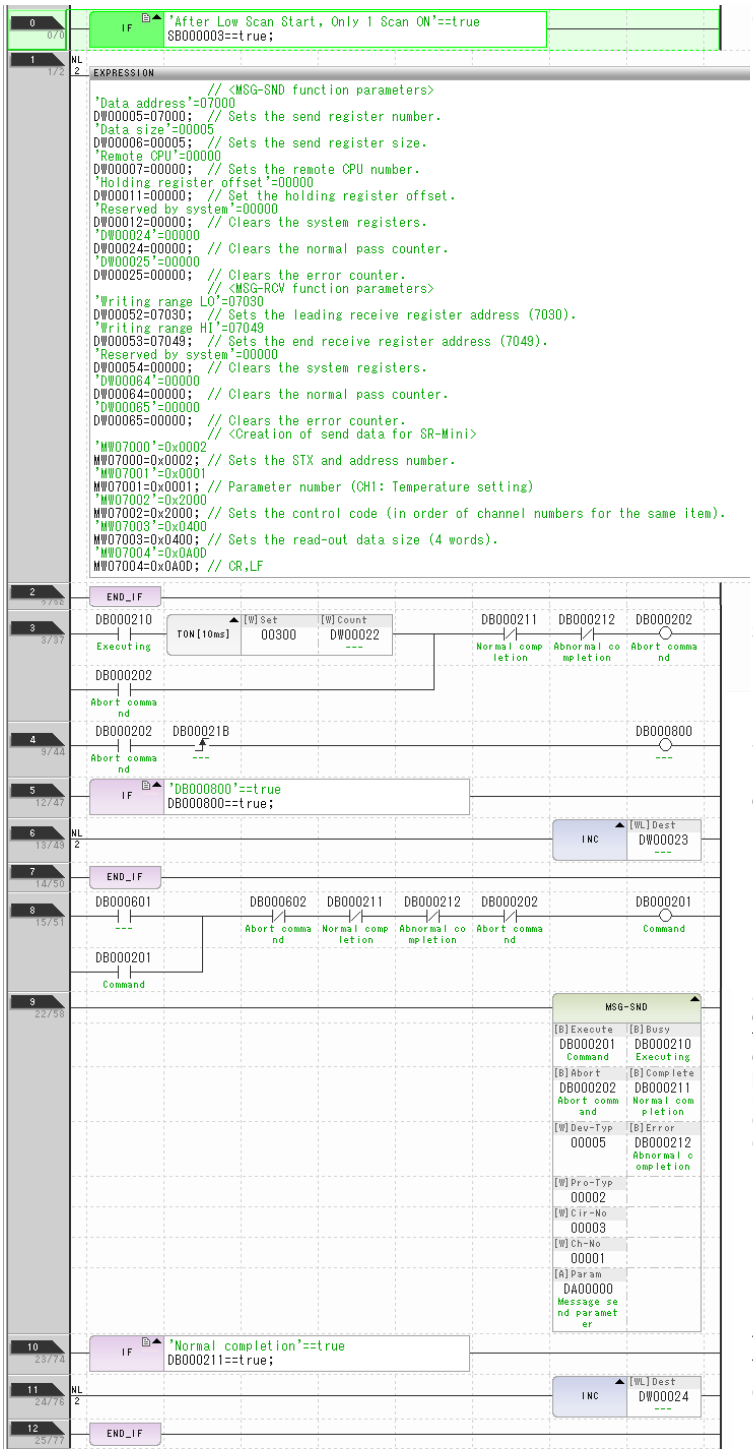


(6) Programming Example

Example ladder programming for the MP2000 Series Machine Controller using the MSG-SND and MSG-RCV functions is shown below. When connected to a Temperature Controller, the MSG-SND function is used to send a command and the MSG-RCV function is used to receive the response.
Here, the temperature inputs CH0 to CH3 from the SR Mini are stored in registers starting with MW07030.

Sets parameters in the first scan after startup.

(Low-speed scan (DWG.L): SB000003,
High-speed scan (DWG.H): SB000001)



Forced stop if processing is not completed within 3 seconds after sending the Send command.

Sends the Abort command.

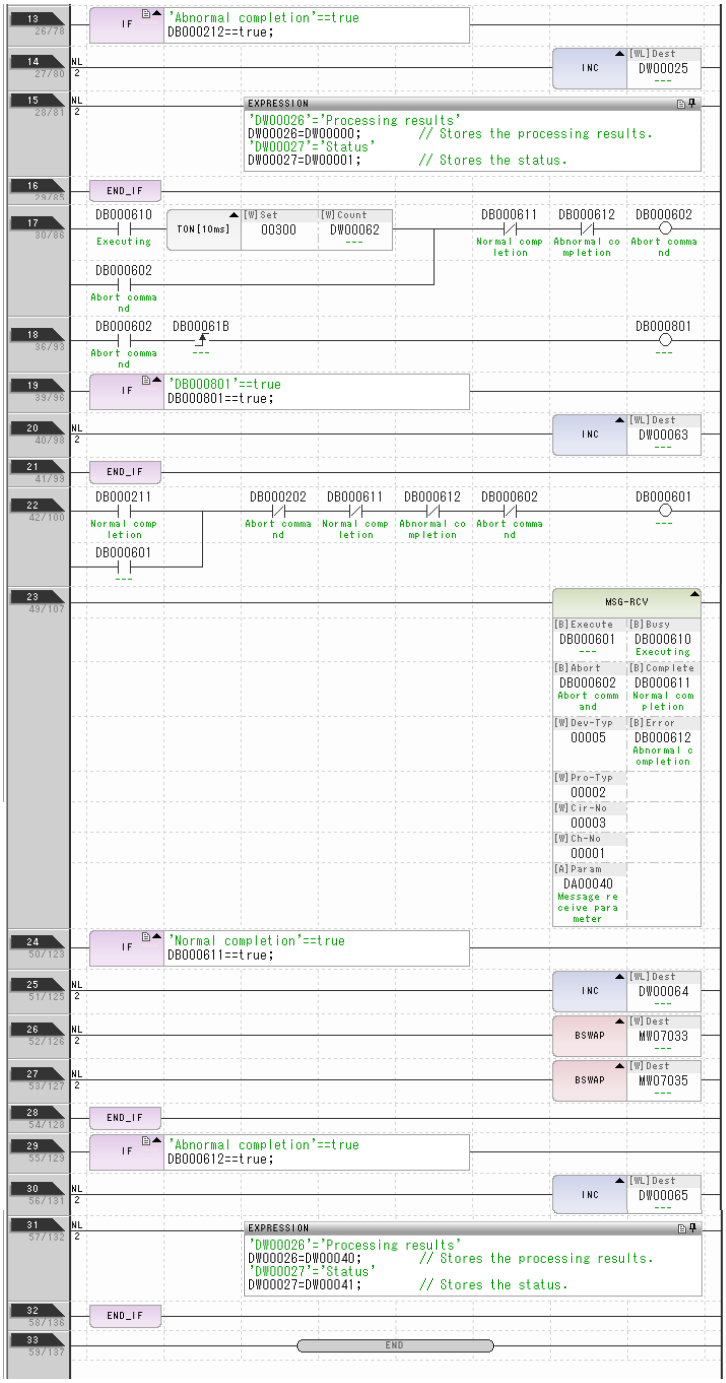
Increments the stop counter when the Abort command is sent.

Function Send command

Sends a message to the remote station. The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 2 (general-purpose communication)
Cir-No. (circuit number): 3 (connection port)
Ch-No. (channel number): 1

Turns ON after the command message is sent and the response message is received.

Clears the normal pass counter.



Ended with transmission errors.

Increments the abnormal pass counter.

Stores the processing results and the status.

Forced stop if processing is not completed within 3 seconds after sending the Send command.

Sends the Abort command.

Increments the stop counter when the Abort command is sent.

Function Send command

Receives a message from the remote station.
The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 2 (general-purpose communication)
Cir-No. (circuit number): 3 (connection port)
Ch-No. (channel number): 1

Turns ON after the message is received.

Clears the normal pass counter.

Swaps receive data bytes (1°C temperature).

Swaps receive data bytes (1°C temperature).

Ended with transmission errors.

Increments the abnormal pass counter.

Stores the processing results and the status.

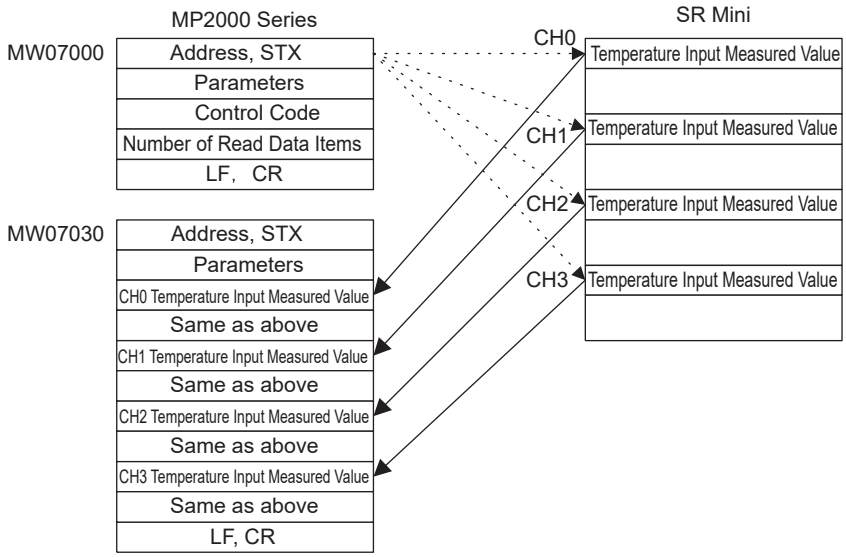


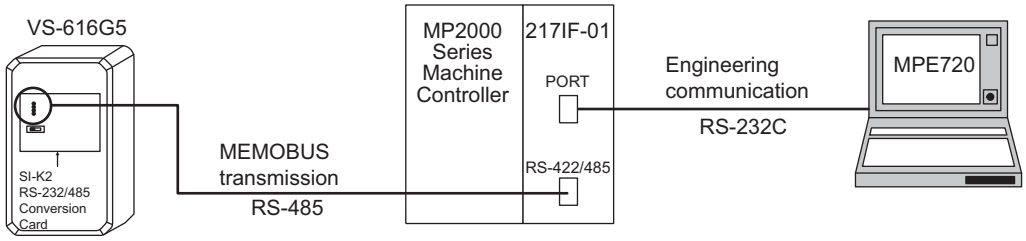
Fig. 3.6 Data Flow in the System Using the Programming Example

3.5.6 Connection to an Inverter

This section describes connection to a Yaskawa VS-616G5 Inverter.

(1) System Configuration Example

In this example, the VS-616G5 Inverter manufactured by Yaskawa is connected to the RS-485 port of the 217IF-01 Module. In this connection, multi-drop connections and control of multiple inverters is possible using 1:N communication.
The personal computer running the MPE720 is connected to the RS-232C (PORT) port of the 217IF-01 Module to set up the engineering environment.

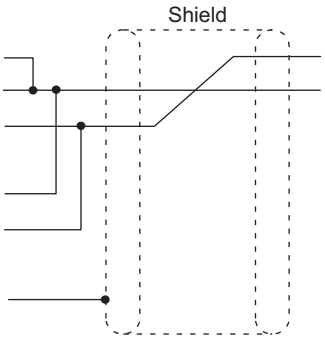


(2) Cable Specifications

■ RS-422/485 Cable

217IF-01	
14-pin MDR Connector	
Signal Name	Pin Number
RXR	7
RX-	4
RX+	3
TXR	11
TX-	9
TX+	8
GND	14
Connector shell	

Cable Connections

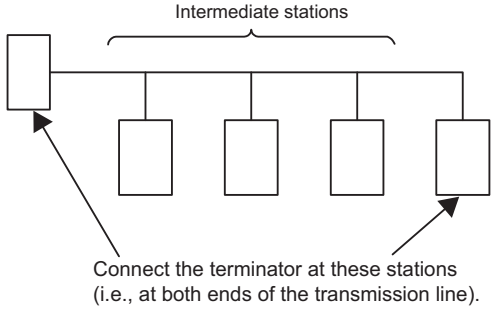


VS-616G5	
Screw Terminal	
Pin Number	Signal Name
1	SRD (+)
2	SRD (-)
3	SRD (+)
4	SRD (-)

- As shown in the above wiring diagram, connect the cable so that the terminator for the 217IF-01 Module is used.
- To connect the terminator in the VS-616G5, turn ON the switch on the PCB.
  When using RS-485 connections, turn OFF the switch on the PCB at all stations except for those on the ends of the transmission path.

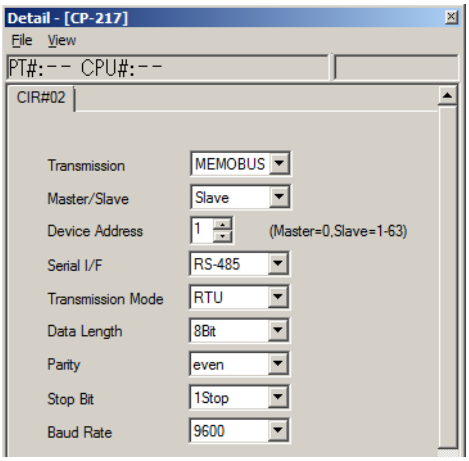
■ Systems in Which the Terminator Cannot Be Connected in the Communication Module

If a terminator cannot be connected in the Communication Module, connect a 120-Ω terminator outside the Module. Connect the terminator at the stations at both ends of the transmission path. Do not connect the terminator at any intermediate stations.



(3) 217IF Transmission Settings

The following figure shows the settings for the 217IF transmission system.



(4) Inverter Parameter Settings

The parameter settings for the VS-616G5 are shown below.

Transmission address	H5-01 = 1
Baud rate	H5-02 = 9600 bps
Transmission parity	H5-03 = Even parity

(5) Procedure to Start Communication

Use the following procedure to start communication.

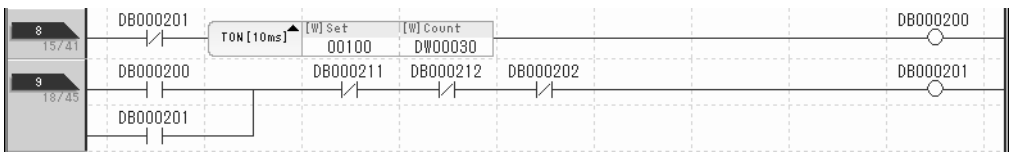
1. Make the settings for serial communication at the Inverter and enable message reception.

2. Start sending messages from the 217IF Module of the MP2000 Series Machine Controller.

Example ladder programming using the MSG-SND function is shown in *(6) Programming Example* on page 3-50. In this programming example, a message is sent when the register set for **Execute** in the MSG-SND function (DB000201) turns ON to start communication with the 217IF Module (Slave).

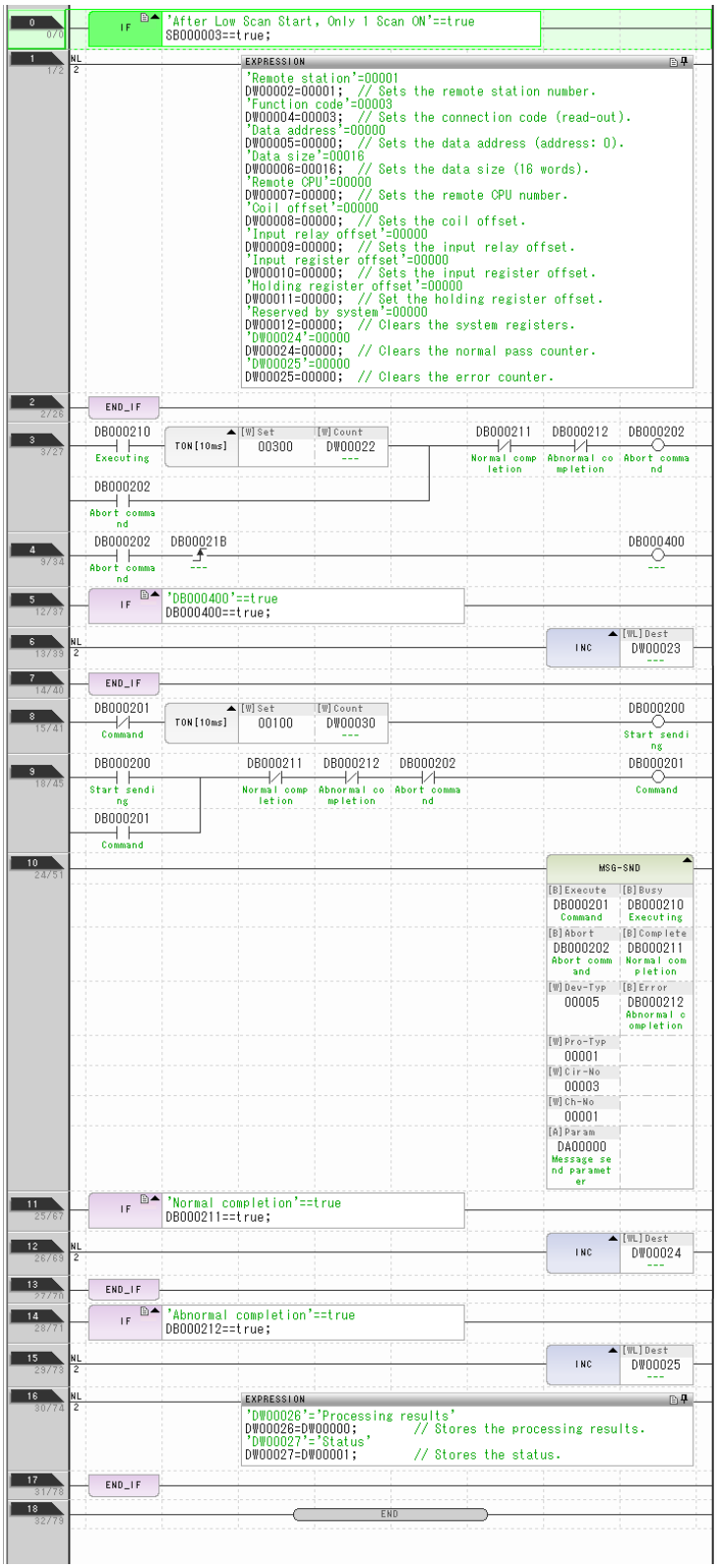
I/O Definition	No.	Name	Setting Example	Description
Input	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

- In *(6) Programming Example* on page 3-50, the programming is written so that a message will be sent every second after one second has expired after the low-speed (or high-speed) scan has started. To change the interval at which messages are sent, change the set value of the timer shown below.



(6) Programming Example

An example of a ladder program using the MSG-SND function of MP2000 Series Machine Controller is shown below.



Sets parameters in the first scan after startup. (Low-speed scan (DWG.L): SB000003, High-speed scan (DWG.H): SB000001)

Sets PARAM02 to PARAM11, clears the system registers, and clears the error counter.

Forced stop if processing is not completed within 3 seconds after sending the Send command.

Sends the Abort command.

Increments the stop counter when the Abort command is sent.

Starts sending every 1 second.

Function Send command

Sends a message to the remote station. The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: Always 1 (MEMOBUS communication)
Cir-No. (circuit number): 3 (connection port)
Ch-No. (channel number): 1 (always 1 for the 217IF-01)

Turns ON after the command message is sent and the response message is received.

Clears the normal pass counter.

Ended with transmission errors.

Increments the abnormal pass counter.

Stores the processing results and the status.

3.5.7 Connection to Programming Device

This section describes a system in which a 217IF-01 Module is connected to the personal computer running the MPE720.

(1) System Configuration Example

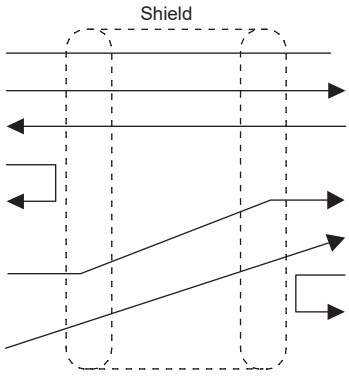
In the following example system configuration, the personal computer running the MPE720 is connected to the RS-232C port on the 217IF-01 Module to set up the engineering environment.

(2) Cable Specifications

■ PORT Cable for IBM PC/AT or Compatible with 9-pin D-sub Connector

MP2000 Series Machine Controller (PORT Connector)	
Signal Name	Pin Number
FG	1
SD (TXD)	2
RD (RXD)	3
RS (RTS)	4
CS (CTS)	5
DR (DSR)	6
SG (GND)	7
–	8
ER (DTR)	9

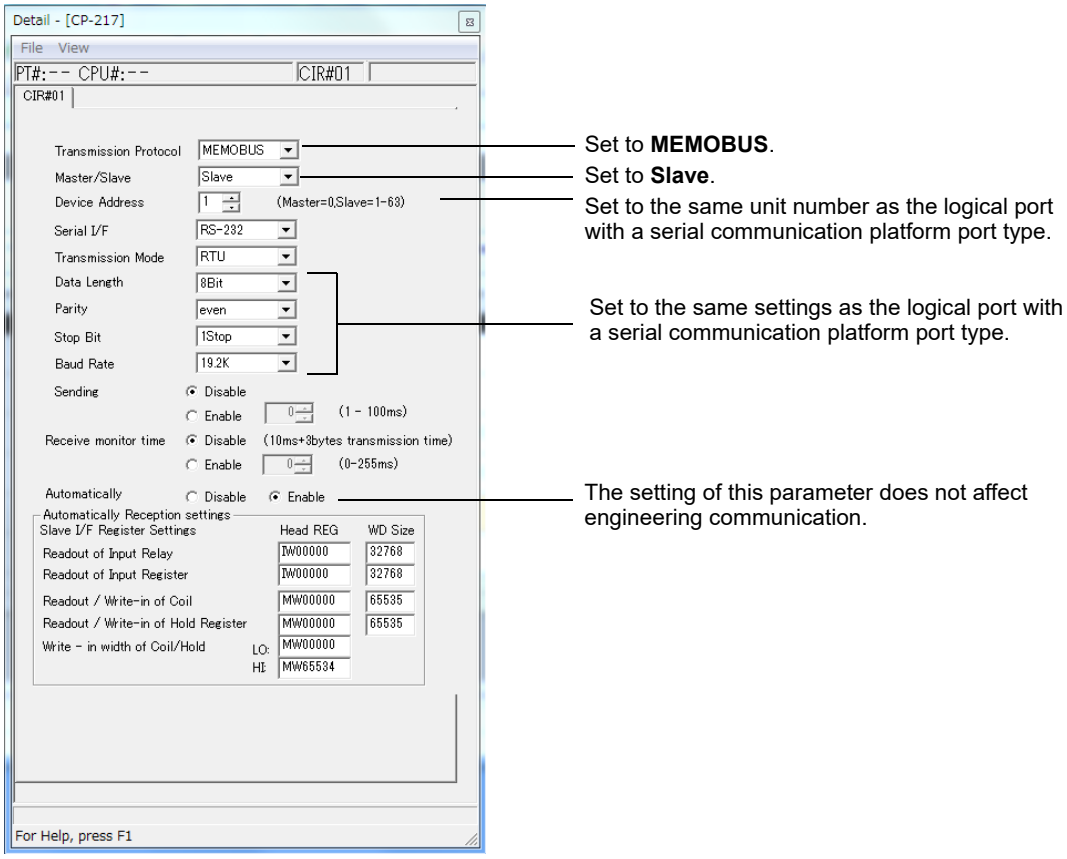
Cable Connections and Signal Directions



IBM PC/AT or compatible (9-pin D-sub Connector)	
Pin Number	Signal Name
1	FG
2	RD (RXD)
3	SD (TXD)
4	ER (DTR)
5	SG (GND)
6	DR (DSR)
7	RS (RTS)
8	CS (CTS)
9	–

(3) 217IF Transmission Settings

Select **MEMOBUS** for the communication protocol in the **217IF Transmission Configuration** Window (RS-232C) and set **Slave** for the **Master/Slave** parameter.



(4) MPE720 (Personal Computer) Settings

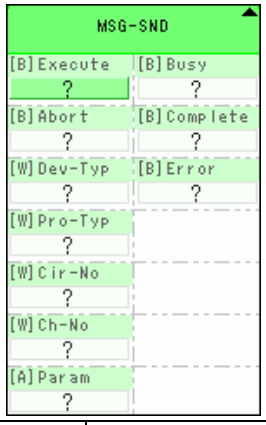
Refer to *1.4 Setting the Communication Platform* on page 1-17.

(5) Programming Example

Ladder programming with the MSG-SND function is not required for engineering communication.

3.6 Message Send Function

3.6.1 Basic Specifications

Function Name	MSG-SND			
Function	Sends a message to a remote station on the line specified by the Transmission Device Type parameter (DEV-TYP). The function supports multiple protocols. The Execute command must be held ON until the Complete or Error output turns ON.			
Function Definition				
I/O Definitions	No.	Name	I/O Designation*1	Description
Inputs	1	Execute	B-VAL	Send Message Command
	2	Abort	B-VAL	Send Message Abort Command
	3	Dev-Typ	I-REG	Transmission Device Type MPLINK/CP-215 (MPLINK) = 1, RS-232C/422/485 (217IF) = 5, Ethernet (218IF) = 6, DeviceNet (260IF) = 11, Ethernet (218IFA, 218IFB, 218IFC) = 16
	4	Pro-Typ	I-REG	Communication Protocol MEMOBUS = 1*2, Non-procedure 1*3 = 2, Non-procedure 2*3 = 3
	5	Cir-No	I-REG	Circuit Number MPLINK/CP-215 (MPLINK) = 1 to 8, RS-232C/422/485 (217IF) = 1 to 16, Ethernet (218IF) = 1 to 8, DeviceNet (260IF) = 1 to 8, Ethernet (218IFA, 218IFB, 218IFC) = 1 to 8
	6	Ch-No	I-REG	Transmission Buffer Channel Number MPLINK/CP-215 (MPLINK) = 1 to 12, RS-232C/422/485 (217IF) = 1, Ethernet (218IF) = 1 to 10, DeviceNet (260IF) = 1 to 4, Ethernet (218IFA) = 1 to 4, Ethernet (218IFB, 218IFC) = 1 to 10
	7	Param	Address input	Parameter List Leading Address (MA, DA)
Outputs	8	Busy	B-VAL	Processing in progress.
	9	Complete	B-VAL	Processing completed.
	10	Error	B-VAL	Error has occurred.

* 1. The I/O designations are as follows:
B-VAL: I/O is specified as bit data.
I-REG: I/O is specified as integer data. Specify the number of an integer register.
Constants (immediate data) can also be specified for inputs.
Address input: The address of the specified register (any integer register) is passed to the function.

* 2. To use the MEMOBUS, Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, select the MEMOBUS protocol by setting the Communication Protocol parameter (PRO-TYP) to 1. The transmission device will automatically convert to the proper protocol.

* 3. Non-procedure 1: Data is sent in word units.
Non-procedure 2: Data is sent in byte units.

3.6.2 Inputs and Outputs for the Message Send Function

(1) Inputs

The following table gives the registers that can be used for the inputs.

Input	I/O Designation	Applicable Registers
Execute Abort	B-VAL	Any bit registers (including those with subscripts) except for # and C registers
Dev-Typ Pro-Typ Cir-No Ch-No	I-REG	Any integer registers (including those with subscripts) Constants
Param	Address input	Any register addresses (including those with subscripts) except for # and C registers

The following sections describe the inputs in more detail.

[a] Execute (Send Message Execute Command)

Specify the bit that will be used to control execution of the Message Send function.
Message send processing is started when the Execute command turns ON. To execute processing, this bit must be turned ON and OFF, e.g., from the ladder program.

- The Execute command must be held at ON until the Complete or Error output turns ON. The message is sent when the Execute command turns ON. To send another message, always turn OFF the Execute command for at least one scan.

[b] Abort (Send Message Abort Command)

Specify the bit that will be used to abort the Message Send function.
Sending the message will be aborted when the Abort command turns ON. The Abort command takes priority over the Execute command.
To abort processing, this bit must be turned ON and OFF, e.g., from the ladder program.

[c] Dev-Typ (Transmission Device Type)

The Dev-Typ input specifies the transmission device type as shown in the following table.

Device	Type Code
MPLINK/CP-215 (MPLINK)	1
RS-232C/422/485 (217IF)	5
Ethernet (218IF)	6
DeviceNet (260IF)	11
Ethernet (218IFA, 218IFB, 218IFC)	16

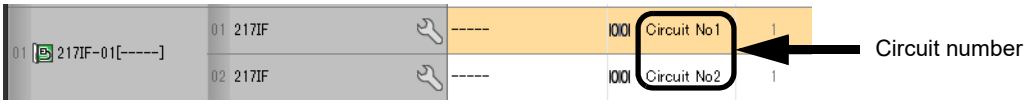
[d] Pro-Typ (Communication Protocol)

The Pro-Typ input specifies the communication protocol as shown in the following table.

Type Code	Communication Protocol	Remarks
1	MEMOBUS	To use the Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, set the type code to 1. The transmission device will automatically convert to the proper protocol.
2	Non-procedure 1 (word units)	Data will be sent in word units using the non-procedure protocol. No response is received from the remote station.
3	Non-procedure 2 (byte units)	Data will be sent in byte units using the non-procedure protocol. No response is received from the remote station.

[e] Cir-No (Circuit Number)

The Cir-No input specifies the circuit number of the transmission device.
Set the circuit number to the value displayed in the **MPE720 Module Configuration** Window.



The valid circuit number ranges are given in the following table.

Device	Valid Circuit Numbers
MPLINK/CP-215 (MPLINK)	1 to 8
RS-232C/422/485 (217IF)	1 to 16
Ethernet (218IF)	1 to 8
DeviceNet (218IFA, 260IF, 218IFC)	1 to 8
Ethernet (218IFA, 218IFB, 218IFC)	1 to 8

[f] Ch-No (Channel Number)

The Ch-No input specifies the channel number of the transmission buffer.

Any channel number can be specified as long as it is within the valid range. If more than one function is being executed at the same time, do not specify the same channel number more than once for the same modem number. (The same channel number can be used as long as the functions are not executed at the same time.)

The valid channel number ranges are given in the following table.

Device	Valid Circuit Numbers
MPLINK/CP-215 (MPLINK)	1 to 12
RS-232C/422/485 (217IF)	1
Ethernet (218IF)	1 to 10
DeviceNet (260IF)	1 to 4
Ethernet (218IFA)	1 to 4
Ethernet (218IFB, 218IFC)	1 to 10

<Examples>

If the RS-232C/RS-422/RS-485 (217IF) transmission device is used, there is only one channel of transmission buffers for both sending and receiving, so only one message can be sent or received at the same time.

- One MSG-SND (or MSG-RCV) function must be programmed for each circuit being used at the same time.
- Refer to *A.3 Transmission Buffer Channels* on page A-9 for information on transmission buffer channels.

[g] Param (Parameter List Leading Address)

The PARAM input specifies the leading address of the parameter list. A parameter list will be automatically created from the 17 words starting with the specified address. Use the parameter list to input the function code and other related parameters. The processing results and status are also output to the parameter list.

- Refer to *3.6.3 Parameter List for Serial Communication (217IF)* on page 3-58 for information on the parameter list.

Example: The following parameter list will be created when the Parameter List Leading Address is set to DA000000.

Register	Parameter F · · · · · · · 0
DW000000	PARAM00
DW000001	PARAM01
DW000002	PARAM02
DW000003	PARAM03
DW000004	PARAM04
DW000005	PARAM05
DW000006	PARAM06
DW000007	PARAM07
DW000008	PARAM08
DW000009	PARAM09
DW000010	PARAM10
DW000011	PARAM11
DW000012	PARAM12
DW000013	PARAM13
DW000014	PARAM14
DW000015	PARAM15
DW000016	PARAM16

(2) Outputs

The following table gives the registers that can be used for the outputs.

Outputs	I/O Designation	Applicable Registers
Busy Complete Error	B-VAL	Any bit registers (including those with subscripts) except for # and C registers

The following sections describe the outputs in more detail.

[a] Busy (Processing in Progress)

Specify the bit that will report when sending the message is being processed.
The Busy output will be ON while message send processing or abort processing is in progress.
Keep the Execute command or Abort command ON while the Busy output is ON.

[b] Complete (Processing Completed)

Specify the bit that will report when processing to send the message has been completed.
The Complete output will be ON for only one scan after message send processing or abort processing has been completed normally.

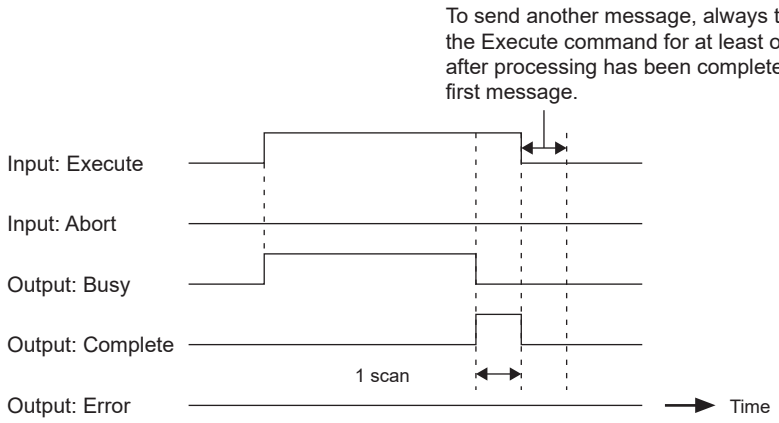
[c] ERROR (Error Occurred)

Specify the bit that will report when an error has occurred while sending the message.
The Error output will turn ON for only one scan when an error occurs.

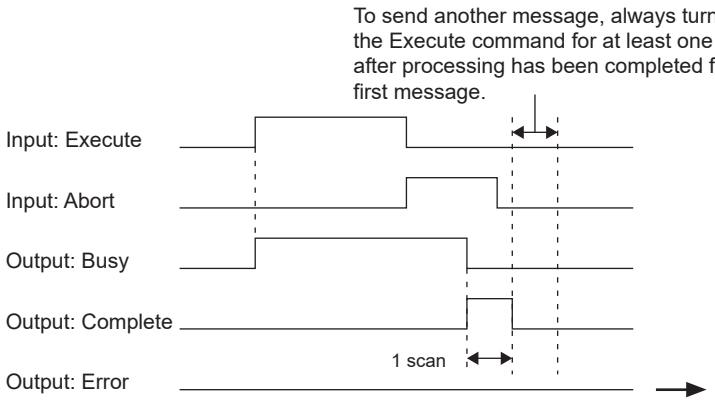
(3) I/O Timing Chart

The following timing charts show the bit inputs and outputs used with the MSG-SND function.

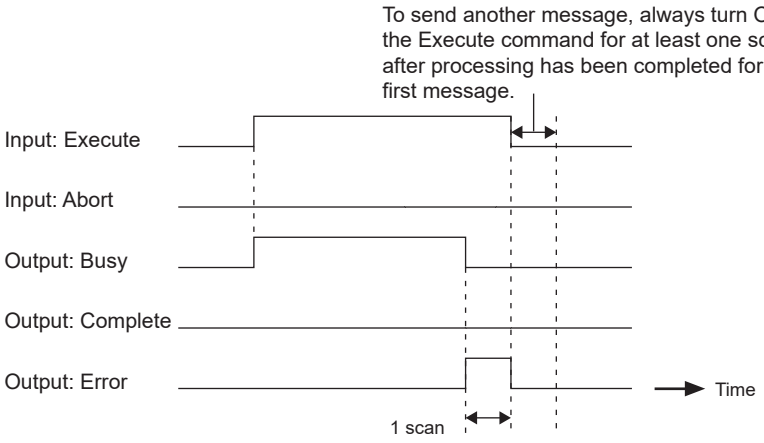
<Normal Processing>



<Abort Processing>



<Error Processing>



3.6.3 Parameter List for Serial Communication (217IF)

The Param input to the MSG-SND function is a parameter list structure consisting of 17 words. The value of the Param input is the leading address (MA or DA) of the parameter list.
Use the parameter list to input the connection number, function code, and other related parameters. The processing results and status are also output to the parameter list.
The parameter lists for the MEMOBUS and non-procedure communication protocols are given below.

■ Parameter List for MEMOBUS Protocol

Param No.	IN/OUT	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	IN	Remote station number	Specifies the remote station number.
03	IN	Option	Sets options specific to each transmission device.
04	IN	Function code	Sets the function code to be sent.
05	IN	Data address	Specifies the leading address of the data.
06	IN	Data size	Sets the data size to be read or written.
07	IN	Remote CPU number	Sets the remote CPU number.
08	IN	Coil offset	Sets the offset word address of the coils.
09	IN	Input relay offset	Sets the offset word address of the input relays.
10	IN	Input register offset	Sets the offset word address of the input registers.
11	IN	Holding register offset	Sets the offset word address of the holding registers.
12	SYS	Reserved by the system (1).	
13 to 16	SYS	Reserved by the system (2).	

- IN: Input, OUT: Output, SYS: Used by the system.
- Refer to *3.6.4 Function Setting Example and Parameter Details for the MEMOBUS Protocol* on page 3-59, *3.6.5 Function Setting Example and Parameter Details for the MELSEC Protocol* on page 3-67, and *3.6.6 Function Setting Example and Parameter Details for the OMRON Protocol* on page 3-71 for details on the parameters.

■ Parameter List for Non-procedure Protocol

Param No.	IN/OUT	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	IN	(Not used.)	
03	IN	(Not used.)	
04	IN	(Not used.)	
05	IN	Data address	Specifies the leading address of the data.
06	IN	Data size	Sets the data size to be written.
07	IN	(Not used.)	
08	IN	(Not used.)	
09	IN	(Not used.)	
10	IN	(Not used.)	
11	IN	Register offset	Sets the offset word address of the registers.
12	SYS	Reserved by the system (1).	
13 to 16	SYS	Reserved by the system (2).	

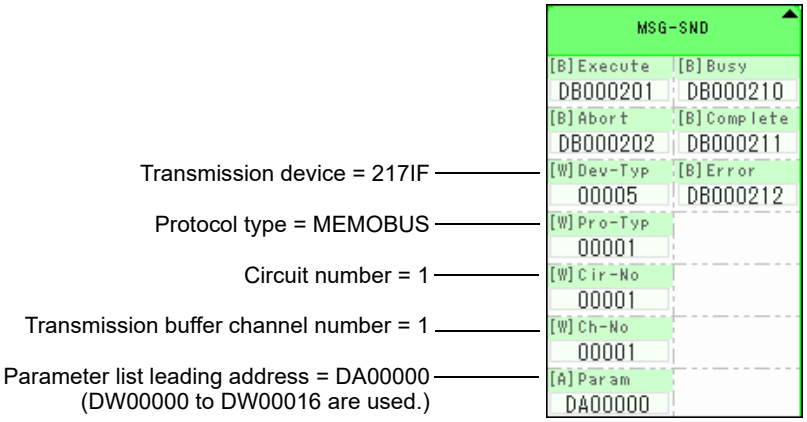
- IN: Input, OUT: Output, SYS: Used by the system.
- Refer to *3.6.7 Function Setting Example and Parameter Details for the Non-procedure Protocol* on page 3-75 for details on the parameters.

3.6.4 Function Setting Example and Parameter Details for the MEMOBUS Protocol

This section provides an MSG-SND function setting example and describes the parameter list in detail for serial communication (217IF) using the MEMOBUS protocol.

(1) MSG-SND Function Setting Example

This example shows the settings for using the 217IF as the transmission device.



The protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 217IF transmission device.
The transmission buffer channel number is set to 1. This value must be set to 1.
For details on settings, refer to *3.6.2 Inputs and Outputs for the Message Send Function* on page 3-54.

(2) Parameter List for MEMOBUS Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	07	IN	Remote CPU number
01	OUT	Status	08	IN	Coil offset
02	IN	Remote station number	09	IN	Input relay offset
03	IN	Option	10	IN	Input register offset
04	IN	Function code	11	IN	Holding register offset
05	IN	Data address	12	SYS	Reserved by the system (1).
06	IN	Data size	13 to 16	SYS	Reserved by the system (2).

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for MEMOBUS Protocol

This section describes the parameter list in detail for using the MEMOBUS protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARAM00. The lower-place byte is for system analysis.

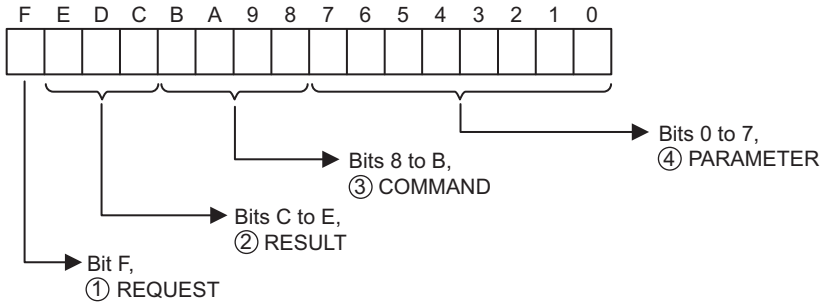
Value of Processing Result	Meaning
00xxH	Processing in progress (Busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	Function code error	An undefined function code was sent or received. Check PARAM04 (function code).
82xxH	Address setting error	One of the following settings in not within the valid range. Check the settings. PARAM05 (data address) PARAM08 (coil offset) PARAM09 (input relay offset) PARAM10 (input register offset) PARAM11 (holding register offset)
83xxH	Data size error	The size of the sent or received data is not within the allowable range. Check PARAM06 (data size).
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-SND function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-SND function.
86xxH	Station address error	The station number is not within the allowable range. Check PARAM02 (remote station number).
87xxH	–	–
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-SND function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-SND function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-SND function is output to these bits.

Code	Abbreviation	Meaning
1	SEND_OK	Sending has been completed normally.
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-SND function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send General-purpose Message with Non-procedure Protocol
2	U_REC	Receive General-purpose Message with Non-procedure Protocol
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response.
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response.
C	MR_SEND	Send MEMOBUS response

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the remote station number is output.

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Connection number out of range
	02	MEMOBUS response receive monitor time error
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error
	10H	Function code error OR format conversion error
Other values of RESULT	XX	Remote station number

[c] PARAM02: Remote Station Number

The number of the destination station is output to PARAM02.
The valid setting range is given in the following table.

Transmission Device	Connection Number	Remarks
RS-232C/RS-422/RS-485 (217IF)	0	The message is sent to all the stations (broadcast transmission).
	1 to 254	The message is sent to the remote station with the designated station number.

- With a broadcast transmission, only the message is sent. Response data is not received. Broadcast transmission can thus be used only for write commands.
- If the remote station is the 217IF, the remote station number is set to the device address of the slave.

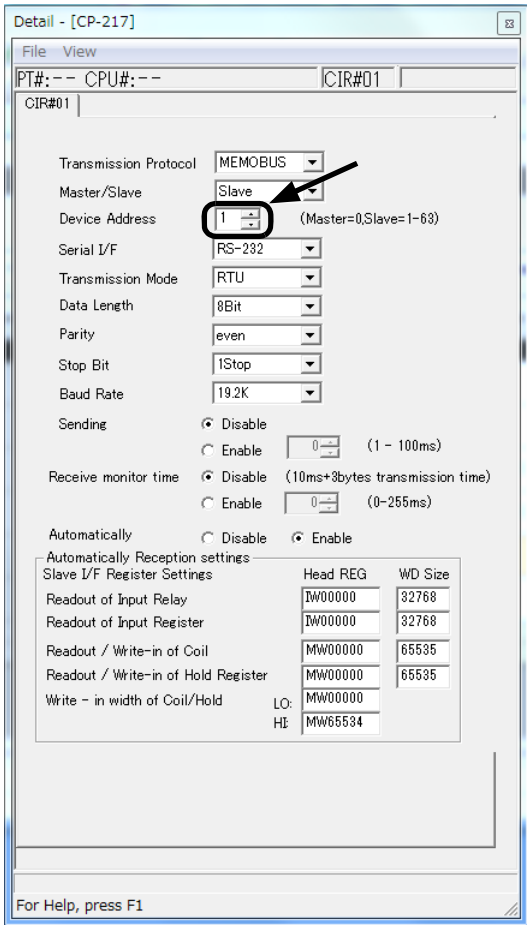


Fig. 3.7 The **217IF Parameter Setting** Tab Page of the **MPE720 Module Configuration** Window

[d] PARAM03: Option

PARAM03 sets options specific to each transmission device.
PARAM03 is not used for the MEMOBUS protocol. Nothing needs to be set.

[e] PARAM04: Function Code

PARAM04 sets the function code to be sent.
The function registered to the function code is used by specifying the function code here. Function include reading coil and input relay status and writing holding registers.
The function codes used for the MEMOBUS protocol are listed in the following table.

<Function Codes>

Function Code	Applicable Data Type	Function	Protocol Type
			MEMOBUS
00H	–	Not used.	–
01H	B	Read Coil Status	○
02H	B	Read Input Relay Status	○
03H	W	Read Holding Register Contents	○
04H	W	Read Input Register Contents	○
05H	B	Modify Status of a Single Coil	○
06H	W	Write to a Single Holding Register	○
07H	–	Not used.	–
08H	–	Loopback Test	○
09H	W	Read Holding Register Contents (Extended)	×
0AH	W	Read Input Register Contents (Extended)	×
0BH	W	Write to Holding Register (Extended)	×
0CH	–	Not used.	–
0DH	W	Read Discontinuous Holding Register Contents (Extended)	×
0EH	W	Write to Discontinuous Holding Registers (Extended)	×
0FH	B	Modify Status of Multiple Coils	○
10H	W	Write to Multiple Holding Registers	○

- B: Bit data, W: Word data
- ○: Can be set, ×: Cannot be set.
- Only MW (MB) registers may be used for sending and receiving messages when operating as the Master.
- MB, MW, IB, and IW registers can be used as coils, holding registers, input relays, and input registers, respectively, when operating as a Slave.

[f] PARAM05: Data Address

PARAM05 specifies the leading address of the data.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

The data address setting depends on the function code.
The data address setting ranges used for the MEMOBUS protocol are listed in the following table.

<Data Address Setting Ranges for MEMOBUS>

Function Code	Applicable Data Type	Function	Data Address Setting Range
00H	–	Not used.	Invalid
01H	B	Read Coil Status*1	0 to 65535 (0 to FFFFH)
02H	B	Read Input Relay Status*1	0 to 65535 (0 to FFFFH)
03H	W	Read Holding Register Contents*2	0 to 65534 (0 to FFFEH)
04H	W	Read Input Register Contents*2	0 to 32767 (0 to 7FFFH)
05H	B	Modify Status of a Single Coil*1	0 to 65535 (0 to FFFFH)
06H	W	Write to a Single Holding Register*2	0 to 65534 (0 to FFFEH)
07H	–	Not used.	Invalid
08H	–	Loopback Test	Invalid
09H	W	Read Holding Register Contents (Extended)	Cannot be used.
0AH	W	Read Input Register Contents (Extended)	Cannot be used.
0BH	W	Write to Holding Register (Extended)	Cannot be used.
0CH	–	Not used.	Invalid
0DH	W	Read Discontinuous Holding Register Contents (Extended)	Cannot be used.
0EH	W	Write to Discontinuous Holding Registers (Extended)	Cannot be used.
0FH	B	Modify Status of Multiple Coils*1	0 to 65535 (0 to FFFFH)
10H	W	Write to Multiple Holding Registers*2	0 to 65534 (0 to FFFEH)

* 1. Read/write request for coils and relays: Set the leading bit address of the data.
* 2. Continuous read/write request for registers: Set the leading word address of the data.

[g] PARAM06: Data Size

PARAM06 specified the data size to be read or written as the number of bits or words.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size setting depends on the function code and the transmission device.
The data size setting ranges used for the MEMOBUS protocol are listed in the following table.

<Data Size Setting Ranges for MEMOBUS or Extended MEMOBUS>

Function Code	Applicable Data Type	Function	Data Size Setting Range
00H	–	Not used.	Invalid
01H	B	Read Coil Status*1	1 to 2000
02H	B	Read Input Relay Status*1	1 to 2000
03H	W	Read Holding Register Contents*2	1 to 125
04H	W	Read Input Register Contents*2	1 to 125
05H	B	Modify Status of a Single Coil*1	Invalid
06H	W	Write to a Single Holding Register	Invalid
07H	–	Not used.	Invalid
08H	–	Loopback Test	Invalid
09H	W	Read Holding Register Contents (Extended)	Cannot be used.
0AH	W	Read Input Register Contents (Extended)	Cannot be used.
0BH	W	Write to Holding Register (Extended)	Cannot be used.
0CH	–	Not used.	Invalid
0DH	W	Read Discontinuous Holding Register Contents (Extended)	Cannot be used.
0EH	W	Write to Discontinuous Holding Registers (Extended)	Cannot be used.
0FH	B	Modify Status of Multiple Coils*1	1 to 800
10H	W	Write to Multiple Holding Registers*2	1 to 100

* 1. Set the number of bits.
* 2. Set the number of words.

- The data sizes listed in the table are decimal values.

[h] PARAM07: Remote CPU Number

PARAM07 sets the remote CPU number.
PARAM07 is not used for the MEMOBUS protocol. Nothing needs to be set.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the offsets for the data read or write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Meaning	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter	Protocol Type
			MEMOBUS
01H	Read Coil Status	PARAM08	○
02H	Read Input Relay Status	PARAM09	○
03H	Read Holding Register Contents	PARAM11	○
04H	Read Input Register Contents	PARAM10	○
05H	Modify Status of a Single Coil	PARAM08	○
06H	Write to a Single Holding Register	PARAM11	○
09H	Read Holding Register Contents (Extended)	PARAM11	×
0AH	Read Input Register Contents (Extended)	PARAM10	×
0BH	Write to Holding Register (Extended)	PARAM11	×
0DH	Read Discontinuous Holding Register Contents (Extended)	PARAM11	×
0EH	Write to Discontinuous Holding Registers (Extended)	PARAM11	×
0FH	Modify Status of Multiple Coils	PARAM08	○
10H	Write to Multiple Holding Registers	PARAM11	○

- ○: Can be set, ×: Cannot be set.

[j] PARAM12: Reserved by the System (1)

PARAM12 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[k] PARAM13 to PARAM16: Reserved by the System (2)

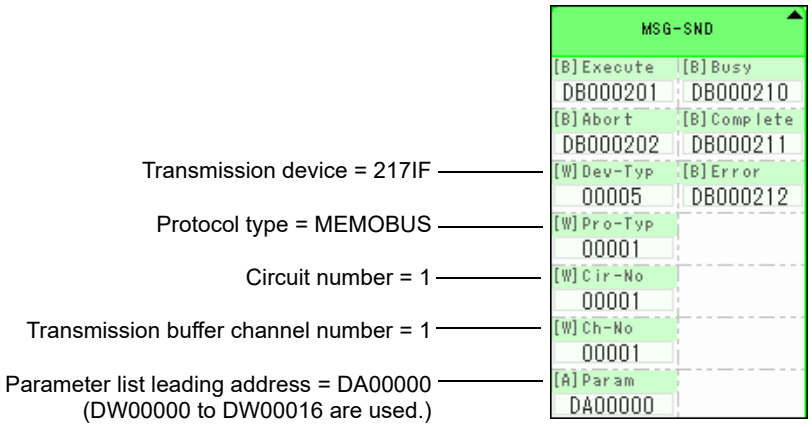
These parameters are used by the system. Do not change the values set for these parameters.

3.6.5 Function Setting Example and Parameter Details for the MELSEC Protocol

This section provides an MSG-SND function setting example and describes the parameter list in detail for serial communication (217IF) using the MELSEC protocol.

(1) MSG-SND Function Setting Example

This example shows the settings for using the 217IF as the transmission device.



To use the MELSEC protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 217IF transmission device.
The transmission buffer channel number is set to 1. This value must be set to 1.
For details on settings, refer to *3.6.2 Inputs and Outputs for the Message Send Function* on page 3-54.

(2) Parameter List for MELSEC Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	07	IN	Remote CPU number
01	OUT	Status	08	IN	Coil offset
02	IN	Remote station number	09	IN	Input relay offset
03	IN	Option	10	IN	Input register offset
04	IN	Function Code	11	IN	Holding register offset
05	IN	Data address	12	SYS	Reserved by the system (1).
06	IN	Data size	13 to 16	SYS	Reserved by the system (2).

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for MELSEC Protocol

This section describes the parameter list in detail for using the MELSEC protocol.

[a] PARAM00: Processing Result

Refer to *3.6.4 (3) [a] PARAM00: Processing Result* on page 3-60.

[b] PARAM01: Status

Refer to *3.6.4 (3) [b] PARAM01: Status* on page 3-60.

[c] PARAM02: Remote Station Number

The number of the destination station is output to PARAM02. The valid setting range is given in the following table.

Transmission Device	Connection Number	Remarks
RS-232C/RS-422/RS-485 (217IF)	0x100	The message is sent to the remote station set to station number 0.
	1 to 254	The message is sent to the remote station with the designated station number.

- Depending on the MELSEC Serial Unit, the station number setting may be fixed to 0. If that is the case, specify 0x100 for the station number.

[d] PARAM03: Option

PARAM03 sets options specific to each transmission device.
PARAM03 is not used for the MELSEC protocol. Nothing needs to be set.

[e] PARAM04: Function Code

PARAM04 sets the function code to be sent.
The function registered to the function code is used by specifying the function code here. Functions include reading bit and word devices and writing word devices.
The function codes used for the MELSEC protocol are listed in the following table.

<Function Codes for MELSEC>

Common MELSEC ACPU Command	Function Code	Applicable Data Type	Function
WR	01H/02H	B	Reads data from bit-type devices in 16-point units.
	03H/04H	W	Reads data from word-type devices in 1-point units.
WW	0FH	B	Writes data to bit-type devices in 16-point units.
	10H	W	Writes data to word-type devices in 1-point units.
TT	08H	–	Executes a loopback test.

- B: Bit data, W: Integer data
- The special commands for the AnCPU are not supported. Use the common ACPU commands listed above when accessing an AnCPU. Accessing the extended file registers in an AnCPU is not possible.

[f] PARAM05: Data Address

PARAM05 specifies the leading address of the data.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

The setting ranges for function codes and data addresses depend on the MELSEC device type and device range.
The data address setting ranges used for the MELSEC protocol are listed in the following tables.

<Data Address Setting Ranges for MELSEC Bit Devices>

Device	Common ACPU Command Device Range	Decimal/ Hex	Function Code	Data Address Setting Range	Corresponding Register Numbers
X	X0000 to X07FF	Hex	02H: Input relays	0 to 2047	MB000000 to MB00127F
Y	Y0000 to Y07FF	Hex	01H/0FH: Coils	0 to 2047	MB000000 to MB00127F
M	M000 to M2047	Decimal	01H/0FH: Coils	2048 to 4095	MB001280 to MB00255F
M	M9000 to M9255	Decimal	01H/0FH: Coils	4096 to 4351	MB002560 to MB002715F
B	B0000 to B03FF	Hex	01H/0FH: Coils	4352 to 5375	MB002720 to MB00335F
F	F0000 to F0255	Decimal	01H/0FH: Coils	5376 to 5631	MB003360 to MB00351F
TS	TS000 to TS255	Decimal	02H: Input relays	2048 to 2303	MB001280 to MB00143F
TC	TC000 to TC255	Decimal	02H: Input relays	2304 to 2559	MB001440 to MB00159F
CS	CS000 to CS255	Decimal	02H: Input relays	2560 to 2815	MB001660 to MB00175F
CC	CC000 to CC255	Decimal	02H: Input relays	2816 to 3071	MB001760 to MB00191F

- Even for the same device ranges, the device area ranges depend on the MELSEC Sequencer. Refer to the MELSEC manuals for details.
- The MP2000 Series Machine Controller register addresses can be adjusted by using the offset setting in the MSG-SND function.

<Data Address Setting Ranges for MELSEC Word Devices>

Device	Common ACPU Command Device Range	Decimal/ Hex	MEMOBUS Commands	Start No.	Corresponding Register Numbers
TN	TN0000 to TN255	Decimal	04H: Input registers	0 to 255	MW00000 to MW00255
CN	CN0000 to CN255	Decimal	04H: Input registers	256 to 511	MW00256 to MW00511
D	D0000 to D1023	Decimal	03H/10H: Holding registers	0 to 1023	MW00000 to MW01023
D (Special)	D9000 to D9255	Decimal	03H/10H: Holding registers	1024 to 1279	MW01024 to MW01279
W	W0000 to W03FF	Hex	03H/10H: Holding registers	1280 to 2303	MW01280 to MW02303
R	R0000 to R8191	Decimal	03H/10H: Holding registers	2304 to 10495	MW02304 to MW10495

- Even for the same device ranges, the device area ranges depend on the MELSEC Sequencer. Refer to the MELSEC manuals for details.
- The MP2000 Series Machine Controller register addresses can be adjusted by using the offset setting in the MSG-SND function.

[g] PARAM06: Data Size

PARAM06 specified the data size to be read or written as the number of bits or words.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size setting depends on function code and the transmission device.
The data size setting ranges used for the MELSEC protocol are listed in the following table.

<Data Size Setting Ranges for MELSEC>

Common MELSEC ACPU Command	Function Code	Function	Data Size Setting Range
WR	01H/02H	Reads data from bit-type devices in 16-point units.	1 to 512 bits (32 words)
	03H/04H	Reads data from word-type devices in 1-point units.	1 to 64 points
WW	0FH	Writes data to bit-type devices in 16-point units.	1 to 160 bits (10 words)
	10H	Writes data to word-type devices in 1-point units.	1 to 64 points
TT	08H	Executes a loopback test.	–

[h] PARAM07: Remote CPU Number

Refer to *3.6.4 (3) [h] PARAM07: Remote CPU Number* on page 3-65.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the offsets for the data read or write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Meaning	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter
01H	Read Coil Status	PARAM08
02H	Read Input Relay Status	PARAM09
03H	Read Holding Register Contents	PARAM11
04H	Read Input Register Contents	PARAM10
0FH	Modify Status of Multiple Coils	PARAM08
10H	Write to Multiple Holding Registers	PARAM11

[j] PARAM12: Reserved by the System (1)

PARAM12 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[k] PARAM13 to PARAM16: Reserved by the System (2)

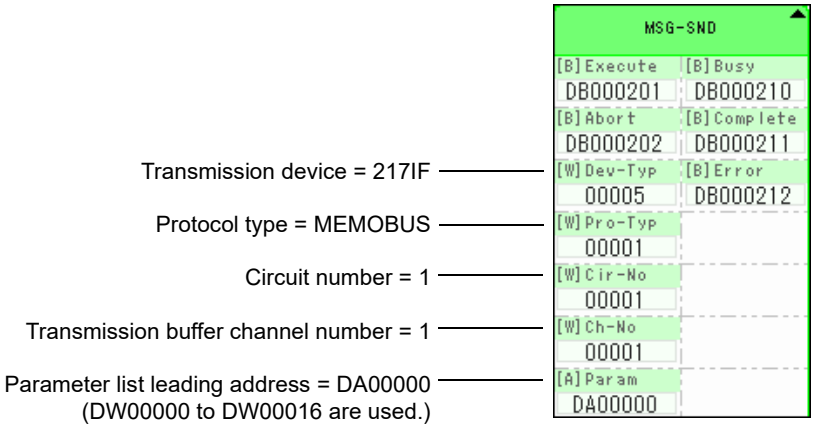
These parameters are used by the system. Do not change the values set for these parameters.

3.6.6 Function Setting Example and Parameter Details for the OMRON Protocol

This section provides an MSG-SND function setting example and describes the parameter list in detail for serial communication (217IF) using the OMRON protocol.

(1) MSG-SND Function Setting Example

This example shows the settings for using the 217IF as the transmission device.



To use the OMRON protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 217IF transmission device.
The transmission buffer channel number is set to 1. This value must be set to 1.
For details on settings, refer to *3.6.2 Inputs and Outputs for the Message Send Function* on page 3-54.

(2) Parameter List for OMRON Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	07	IN	Remote CPU number
01	OUT	Status	08	IN	Coil offset
02	IN	Remote station number	09	IN	Input relay offset
03	IN	Option	10	IN	Input register offset
04	IN	Function Code	11	IN	Holding register offset
05	IN	Data address	12	SYS	Reserved by the system (1).
06	IN	Data size	13 to 16	SYS	Reserved by the system (2).

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for OMRON Protocol

This section describes the parameter list in detail for using the OMRON protocol.

[a] PARAM00: Processing Result

Refer to *3.6.4 (3) [a] PARAM00: Processing Result* on page 3-60.

[b] PARAM01: Status

Refer to *3.6.4 (3) [b] PARAM01: Status* on page 3-60.

[c] PARAM02: Remote Station Number

The number of the destination station is output to PARAM02.
The valid setting range is given in the following table.

Transmission Device	Connection Number	Remarks
RS-232C/RS-422/RS-485 (217IF)	0	The message is sent to all the stations (broadcast transmission).
	1 to 254	The message is sent to the remote station with the designated station number.

- With a broadcast transmission, only the message is sent. Response data is not received. Broadcast transmission can thus be used only for write commands.

[d] PARAM03: Option

PARAM03 sets options specific to each transmission device.
PARAM03 is not used for the OMRON protocol. Nothing needs to be set.

[e] PARAM04: Function Code

PARAM04 sets the function code to be sent.
The function registered to the function code is used by specifying the function code here. Functions include reading bit and word devices and writing word devices.
The function codes used for the OMRON protocol are listed in the following table.

<Function Codes for OMRON>

OMRON Header Code	Function Code	Applicable Data Type	Function
RR	01H	B	Reads I/O relays, internal auxiliary relays, and special auxiliary relays.
RD	03H	W	Reads data memory area.
WR	0FH	B	Writes I/O relays, internal auxiliary relays, and special auxiliary relays.
WD	10H	W	Writes data memory area.
TS	08H	–	Test

- B: Bit data, W: Integer data

[f] PARAM05: Data Address

PARAM05 specifies the leading address of the data.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

The data address setting ranges used for the OMRON protocol are listed in the following table.

<Data Address Setting Ranges for OMRON>

Device	Channel Number	Relay Number	Function Code	Data Address Setting Range	Corresponding Register Numbers
I/O Relay	000 to 039	00000 to 03915	01H/0FH: Coils	0 to 639	MB000000 to MB00039F
Internal Auxiliary Relay	040 to 246	04000 to 24615	01H/0FH: Coils	640 to 3951	MB000400 to MB00246F
Special Auxiliary Relay	247 to 255	24700 to 25507	01H/0FH: Coils	3952 to 4088	MB002470 to MB002557
Data Memory	0000 to 9999	DM 0000 to DM 9999	03H/10H: Holding registers	0000 to 9999	MW00000 to MW09999

[g] PARAM06: Data Size

PARAM06 specifies the data size to be read or written as the number of bits or words.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size setting depends on function code and the transmission device.
The data size setting ranges used for the OMRON protocol are listed in the following table.

<Data Size Setting Ranges for OMRON>

OMRON Header Code	Function Code	Function	Data Size Setting Range
RR	01H	Reads I/O relays, internal auxiliary relays, and special auxiliary relays.	1 to 2000 bits (125 words)*
RD	03H	Reads data memory area.	1 to 125 words
WR	0FH	Writes I/O relays, internal auxiliary relays, and special auxiliary relays.	1 to 800 bits (50 words)*
WD	10H	Writes data memory area.	1 to 100 words
TS	08H	Test	–

* Specify in units of 16 bits.

- The data size setting ranges give the maximum number of words that can be accessed with one command due to restrictions in the MEMOBUS protocol.

[h] PARAM07: Remote CPU Number

PARAM07 sets the remote CPU number.
PARAM07 is not used for the OMRON protocol. Nothing needs to be set.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the offsets for the data read or write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameters	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter
01H	Read Coil Status	PARAM08
03H	Read Holding Register Contents	PARAM11
0FH	Modify Status of Multiple Coils	PARAM08
10H	Write to Multiple Holding Registers	PARAM11

[j] PARAM12: Reserved by the System (1)

PARAM12 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[k] PARAM13 to PARAM16: Reserved by the System (2)

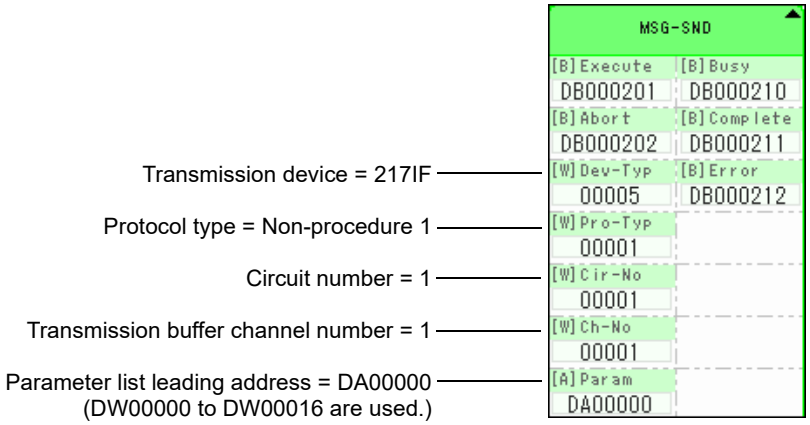
These parameters are used by the system. Do not change the values set for these parameters.

3.6.7 Function Setting Example and Parameter Details for the Non-procedure Protocol

This section provides an MSG-SND function setting example and describes the parameter list in detail for serial communication (217IF) using the non-procedure protocol.

(1) MSG-SND Function Setting Example

This example shows the settings for using the 217IF as the transmission device.



The protocol type (Pro-Typ) is set to 2 to use non-procedure protocol 1 and to 3 to use non-procedure protocol 2.
The circuit number is set to the circuit number assigned to the 217IF transmission device.
The transmission buffer channel number is set to 1. This value must be set to 1.
For details on settings, refer to *3.6.2 Inputs and Outputs for the Message Send Function* on page 3-54.

(2) Parameter List for Non-procedure Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	07	IN	(Not used.)
01	OUT	Status	08	IN	(Not used.)
02	IN	(Not used.)	09	IN	(Not used.)
03	IN	(Not used.)	10	IN	(Not used.)
04	IN	(Not used.)	11	IN	Register offset
05	IN	Data address	12	SYS	Reserved by the system (1).
06	IN	Data size	13 to 16	SYS	Reserved by the system (2).

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for Non-procedure Protocol

This section describes the parameter list in detail for using the non-procedure protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARAM00. The lower-place byte is for system analysis.

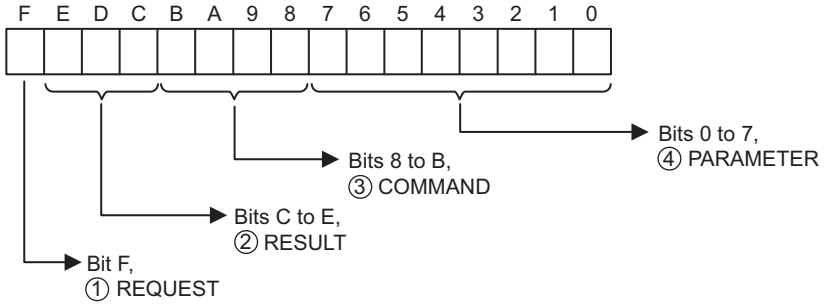
Value of Processing Result	Contents
00xxH	Processing in progress (busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	–	–
82xxH	Address setting error	One of the following settings is not within the valid range. Check the settings. PARAM05 (data address) PARAM11 (holding register offset)
83xxH	Data size error	The size of the sent or received data is not within the allowable range. Check PARAM06 (data size).
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-SND function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-SND function.
86xxH	–	–
87xxH	–	Reserved by the system.
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-SND function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-SND function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-SND function is output to these bits.

Code	Abbreviation	Meaning
1	SEND_OK	Sending has been completed normally.
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-SND function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send general-purpose message.
2	U_REC	Receive general-purpose message.
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response. (For MEMOBUS Protocol)
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response. (For MEMOBUS Protocol)
C	MR_SEND	Send MEMOBUS response (For MEMOBUS Protocol)

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error).

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Station address outside the allowable range
	02	MEMOBUS response receive monitor time error (For MEMOBUS Protocol)
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error (For MEMOBUS Protocol)

[c] PARAM02: Remote Station Number

Specifies the remote station number.
PARAM02 is not used for the non-procedure protocol. Nothing needs to be set.

[d] PARAM05: Data Address

PARAM05 specifies the leading address of the data.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

The valid data address setting ranges are given in the following table.

<Data Address Setting Ranges for Non-procedure Protocol>

Non-procedure Protocol Type	Applicable Data Type	Function	Data Address Setting Range
Non-procedure 1	W	Sends data in word units.	0 to 65534 (0 to FFFEH)
Non-procedure 2	B	Sends data in byte units.*	0 to 65534 (0 to FFFEH)

* The address setting must be in word units.

[e] PARAM06: Data Size

Set the data size to be written as the number of words or bytes.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size setting depends on function code and the transmission device.

<Data Size Setting Ranges for Non-procedure Protocol>

Non-procedure Protocol Type	Applicable Data Type	Function	Data Address Setting Range
Non-procedure 1	W	Sends data in word units. *1	1 to 254
Non-procedure 2	B	Sends data in byte units.*2	1 to 508

* 1. Set the number of words.
* 2. Set the number of bytes.

- The data sizes listed in the table are decimal values.

[f] PARAM11: Register Offset

PARAM11 sets the offset for the data write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

Example: PARAM11 is set as follows to offset the register address by 1000 words.

PARAM11 = 1000

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

[g] PARAM12: Reserved by the System (1)

PARAM12 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

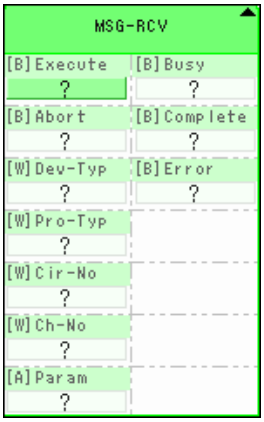
[h] PARAM13 to PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

3.7 Message Receive Function

This section describes the Message Receive function (MSG-RCV) used in a ladder program to receive a message.

3.7.1 Basic Specifications

Function Name	MSG-RCV
Function	Receives a message from a remote station on the line specified by the Transmission Device Type parameter (DEV-TYP). The function supports multiple protocols. The Execute command must be held ON until the Complete or Error output turns ON.
Function Definition	

I/O Definitions	No.	Name	I/O Designation*1	Description
Inputs	1	Execute	B-VAL	Receive message command
	2	Abort	B-VAL	Reception abort command
	3	Dev-Typ	I-REG	Transmission device type MPLINK/CP-215 (MPLINK) = 1, RS-232C/RS-422/RS-485 (217IF) = 5, Ethernet (218IF) = 6, Ethernet (218IFA, 218IFB, 218IFC) = 16
	4	Pro-Typ	I-REG	Communication Protocol MEMOBUS = 1*2, Non-procedure 1*3 = 2, Non-procedure 2*3 = 3
	5	Cir-No	I-REG	Circuit number MPLINK/CP-215 (MPLINK) = 1 to 8, RS-232C/RS-422/RS-485 (217IF) = 1 to 16, Ethernet (218IF) = 1 to 8, Ethernet (218IFB) = 1 to 8
	6	Ch-No	I-REG	Transmission buffer channel number MPLINK/CP-215 (MPLINK) = 1 to 12, RS-232C/RS-422/RS-485 (217IF) = 1, Ethernet (218IF) = 1 to 10, Ethernet (218IFB) = 1 to 10
	7	Param	Address input	Parameter List Leading Address (MA, DA)
Outputs	8	Busy	B-VAL	Processing in progress.
	9	Complete	B-VAL	Processing completed.
	10	Error	B-VAL	Error has occurred.

* 1. The I/O designations are as follows:
B-VAL: I/O is specified as bit data.
I-REG: I/O is specified as integer data. Specify the number of an integer register.
Constants (immediate data) can also be specified for inputs.
Address input: The address of the specified register (any integer register) is passed to the function.

* 2. To use the MEMOBUS, Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, select the MEMOBUS protocol by setting the Communication Protocol parameter (PRO-TYP) to 1. The transmission device will automatically convert to the proper protocol.

* 3. Non-procedure 1: Data is received in word units.
Non-procedure 2: Data is received in byte units.

3.7.2 Inputs and Outputs for the Message Receive Function

(1) Inputs

The following table gives the registers that can be used for the inputs.

Inputs	I/O Designation	Applicable Registers
Execute Abort	B-VAL	Any bit registers (including those with subscripts) except for # and C registers
Dev-Typ Pro-Typ Cir-No Ch-No	I-REG	Any integer registers (including those with subscripts) Constants
Param	Address input	Any register addresses (including those with subscripts) except for # and C registers

The following sections describe the inputs in more detail.

[a] Execute (Receive Message Execute Command)

Specify the bit that will be used to control execution of the Message Receive function.
Message receive processing is started when the Execute command turns ON. To execute processing, this bit must be turned ON and OFF, e.g., from the ladder program.

- The Execute command must be held at ON until the Complete or Error output turns ON. A message is received when the Execute command turns ON. To receive another message, always turn OFF the Execute command for at least one scan.

[b] Abort (Receive Message Abort Command)

Specify the bit that will be used to abort the Message Receive function.
Receiving the message will be aborted when the Abort command turns ON. The Abort command takes priority over the Execute command.
To abort processing, this bit must be turned ON and OFF, e.g., from the ladder program.

[c] Dev-Typ (Transmission Device Type)

The Dev-Typ input specifies the transmission device type as shown in the following table.

Device	Type Code
MPLINK/CP-215 (MPLINK)	1
RS-232C/422/485 (217IF)	5
Ethernet (218IF)	6
Ethernet (218IFA, 218IFB, 218IFC)	16

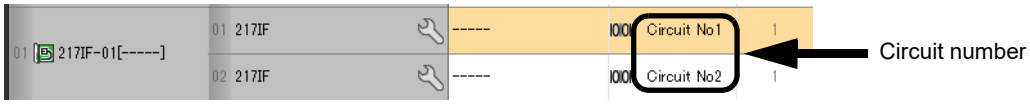
[d] Pro-Typ (Communication Protocol)

The Pro-Typ input specifies the communication protocol as shown in the following table.

Type Code	Communication Protocol	Remarks
1	MEMOBUS	To use the Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to receive a message, set the type code to 1. The transmission device will automatically convert to the proper protocol.
2	Non-procedure 1 (word units)	Data will be received in word units using the non-procedure protocol. No response is received from the remote station.
3	Non-procedure 2 (byte units)	Data will be received in byte units using the non-procedure protocol. No response is received from the remote station.

[e] Cir-No (Circuit Number)

The Cir-No input specifies the circuit number of the transmission device.
Set the circuit number to the value displayed in the **MPE720 Module Configuration** Window.



The valid circuit number ranges are given in the following table.

Device	Valid Circuit Numbers
MPLINK/CP-215 (MPLINK)	1 to 8
RS-232C/422/485 (217IF)	1 to 16
Ethernet (218IF)	1 to 8
Ethernet (218IFB)	1 to 8

[f] Ch-No (Channel Number)

The Ch-No input specifies the channel number of the transmission buffer.
Any channel number can be specified as long as it is within the valid range. If more than one function is being executed at the same time, do not specify the same channel number more than once for the same modem number. (The same channel number can be used as long as the functions are not executed at the same time.)
The valid channel number ranges are given in the following table.

Device	Valid Channel Numbers
MPLINK/CP-215 (MPLINK)	1 to 12
RS-232C/422/485 (217IF)	1
Ethernet (218IF)	1 to 10
Ethernet (218IFB)	1 to 10

<Examples>

If the RS-232C/RS-422/RS-485 (217IF) transmission device is used, there is only one channel of transmission buffers for both sending and receiving, so only one message can be sent or received at the same time.

- One MSG-RCV (or MSG-SND) function must be programmed for each circuit being used at the same time.
- Refer to *A.3 Transmission Buffer Channels* on page A-9 for information on transmission buffer channels.

[g] Param (Parameter List Leading Address)

The PARAM input specifies the leading address of the parameter list. A parameter list will be automatically created from the 17 words starting with the specified address. Use the parameter list to input the function code and other related parameters. The processing results and status are also output to the parameter list.

- Refer to *3.7.3 Parameter List for Serial Communication (217IF)* on page 3-84 for information on the parameter list.

Example: The following parameter list will be created when the Parameter List Leading Address is set to DA000000.

Register	Parameter F · · · · · · · 0
DW000000	PARAM00
DW000001	PARAM01
DW000002	PARAM02
DW000003	PARAM03
DW000004	PARAM04
DW000005	PARAM05
DW000006	PARAM06
DW000007	PARAM07
DW000008	PARAM08
DW000009	PARAM09
DW000010	PARAM10
DW000011	PARAM11
DW000012	PARAM12
DW000013	PARAM13
DW000014	PARAM14
DW000015	PARAM15
DW000016	PARAM16

(2) Outputs

The following table gives the registers that can be used for the outputs.

Inputs	I/O Designation	Applicable Registers
Busy Complete Error	B-VAL	Any bit registers (including those with subscripts) except for # and C registers

The following sections describe the outputs in more detail.

[a] Busy (Processing in Progress)

Specify the bit that will report when receiving the message is being processed.
The Busy output will be ON while message receive processing or abort processing is in progress.
Keep the Execute command or Abort command ON while the Busy output is ON.

[b] Complete (Processing Completed)

Specify the bit that will report when processing to receive the message has been completed.
The Complete output will be ON for only one scan after message receive processing or abort processing has been completed normally.

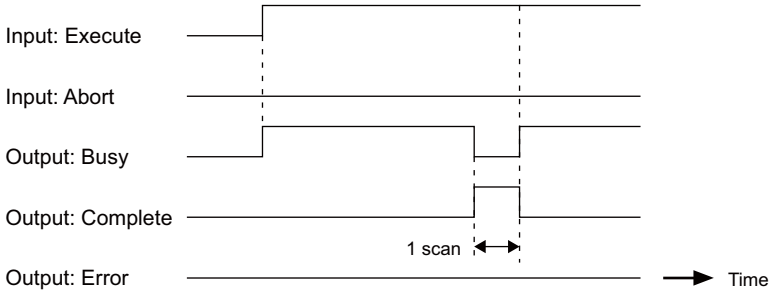
[c] ERROR (Error Occurred)

Specify the bit that will report when an error has occurred while receiving the message.
The Error output will turn ON for only one scan when an error occurs.

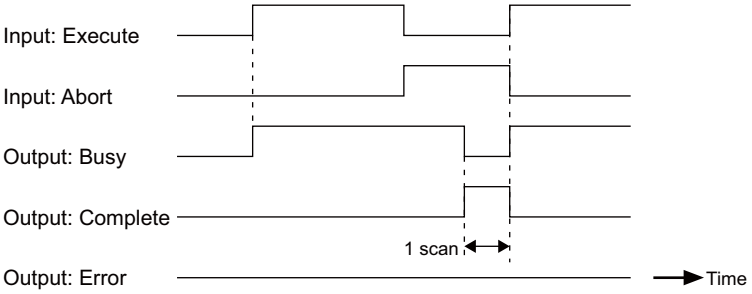
(3) I/O Timing Chart

The following timing charts show the bit inputs and outputs used with the MSG-RCV function.

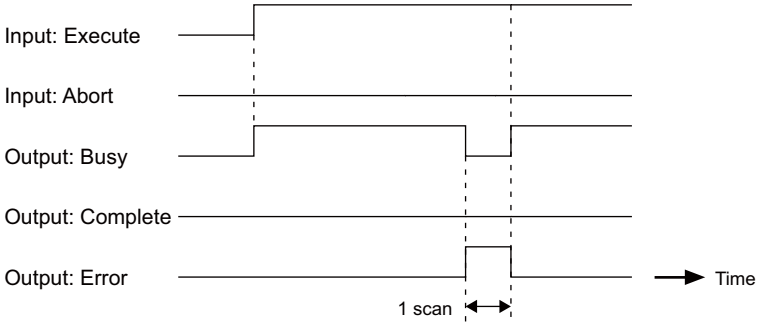
<Normal Processing>



<Abort Processing>



<Error Processing>



3.7.3 Parameter List for Serial Communication (217IF)

The Param input to the MSG-RCV function is a parameter list structure consisting of 17 words. The value of the Param input is the leading address (MA or DA) of the parameter list.
Use the parameter list to input the connection number, function code, and other related parameters. The processing results and status are also output to the parameter list.
The parameter lists for the MEMOBUS and non-procedure communication protocols are given below for 217IF serial communication.

■ Parameter List for MEMOBUS (MELSEC or OMRON) Protocol

Param No.	IN/OUT*	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	OUT	Local station address	The number of the source station is output.
03	OUT	Option	Options specific to each transmission device are output here.
04	OUT	Function code	The function code requested by the sending station is output here.
05	OUT	Data address	The leading address of the data requested by the sending station is output here.
06	OUT	Data size	The read or write data size requested by the sending station is output here.
07	OUT	Remote CPU number	The remote CPU number is output here.
08	IN	Coil offset	Sets the offset word address of the coils.
09	IN	Input relay offset	Sets the offset word address of the input relays.
10	IN	Input register offset	Sets the offset word address of the input registers.
11	IN	Holding register offset	Sets the offset word address of the holding registers.
12	IN	Write range LO	Set the leading address of the write range.
13	IN	Write range HI	Set the final address of the write range.
14	SYS	Reserved by the system (1).	–
15 and 16	SYS	Reserved by the system (2).	–

* IN: Input, OUT: Output, SYS: Used by the system.

- Refer to *3.7.4 Function Setting Example and Parameter Details for the MEMOBUS Protocol* on page 3-85, *3.7.5 Function Setting Example and Parameter Details for the MELSEC Protocol* on page 3-91, and *3.7.6 Function Setting Example and Parameter Details for the OMRON Protocol* on page 3-94 for details on the parameters.

■ Parameter List for Non-procedure Protocol

Param No.	IN/OUT*	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	OUT	(Not used.)	–
03	OUT	Option	Outputs the reception status (non-procedure FD protocol only).
04	OUT	(Not used.)	–
05	OUT	(Not used.)	–
06	OUT	Data size	Set the data size to be read or written.
07	OUT	(Not used.)	–
08 to 11	IN	(Not used.)	–
12	IN	Register offset	Sets the offset word address of the registers.
13	IN	Write range HI	Set the final address of the write range.
14	SYS	Reserved by the system (1).	–
15 and 16	SYS	Reserved by the system (2).	–

* IN: Input, OUT: Output, SYS: Used by the system.

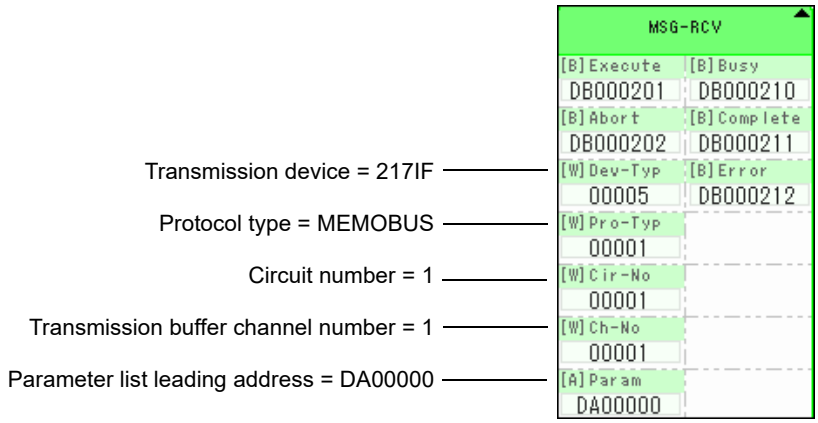
- Refer to *3.7.7 Function Setting Example and Parameter Details for the Non-procedure Protocol* on page 3-97 for details on the parameters.

3.7.4 Function Setting Example and Parameter Details for the MEMOBUS Protocol

This section provides an MSG-RCV function setting example and describes the parameter list in detail for serial communication (217IF) using the MEMOBUS protocol.

(1) MSG-RCV Function Setting Example

This example shows the settings for using the 217IF as the transmission device.



The protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 217IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *3.7.2 Inputs and Outputs for the Message Receive Function* on page 3-80.

(2) Parameter List for MEMOBUS Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	08	IN	Coil offset
01	OUT	Status	09	IN	Input relay offset
02	OUT	Local station address	10	IN	Input register offset
03	OUT	Option	11	IN	Holding register offset
04	OUT	Function code	12	IN	Write range LO
05	OUT	Data address	13	IN	Write range HI
06	OUT	Data size	14	SYS	Reserved by the system (1).
07	OUT	Remote CPU number	15 and 16	SYS	Reserved by the system (2).

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for MEMOBUS Protocol

This section describes the parameter list in detail for using the MEMOBUS protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARM00. The lower-place byte is for system analysis.

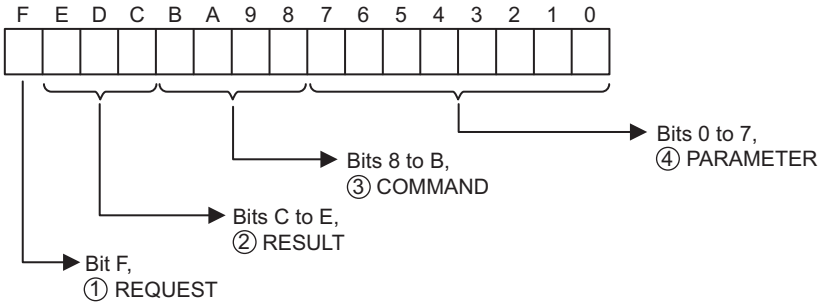
Value of Processing Result	Meaning
00xxH	Processing in progress (busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	Function code error	An undefined function code was received. Check the function code that was received.
82xxH	Address setting error	One of the following settings in not within the valid range. Check the settings. Data address (in request from sender) PARAM08 (coil offset) PARAM09 (input relay offset) PARAM10 (input register offset) PARAM11 (holding register offset)
83xxH	Data size error	The size of the sent or received data is not within the allowable range. Check PARAM06 (data size).
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-RCV function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-RCV function.
86xxH	–	–
87xxH	–	Reserved by the system.
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-RCV function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-RCV function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-RCV function is output to these bits.

Code	Abbreviation	Meaning
1	SEND_OK	Sending has been completed normally.
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-RCV function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send General-purpose Message with Non-procedure Protocol
2	U_REC	Receive General-purpose Message with Non-procedure Protocol
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response.
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response.
C	MR_SEND	Send MEMOBUS response

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the station number is output.

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Connection number out of range
	02	MEMOBUS response receive monitor time error
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error or format conversion error
Other values of RESULT	XX	Station number

[c] PARAM02: Station Number

The station number specified by the source station is output.
The valid station number ranges are given in the following table.

Transmission Device	Station Number	Remarks
RS-232C/RS-422/RS-485 (217IF)	0 to 63	The station number specified by the source station is output.

[d] PARAM03: Option

Options specific to each transmission device are output here.
PARAM03 is not used for the MEMOBUS protocol.

[e] PARAM04: Function Code

The function code that was received is output to PARAM04.
The function codes used for the MEMOBUS protocol are listed in the following table.

<Function Codes for MEMOBUS>

Function Code	Applicable Data Type	Function	Protocol Type
			MEMOBUS
00H	–	Not used.	–
01H	B	Read Coil Status	○
02H	B	Read Input Relay Status	○
03H	W	Read Holding Register Contents	○
04H	W	Read Input Register Contents	○
05H	B	Modify Status of a Single Coil	○
06H	W	Write to a Single Holding Register	○
07H	–	Not used.	–
08H	–	Loopback Test	○
09H	W	Read Holding Register Contents (Extended)	×
0AH	W	Read Input Register Contents (Extended)	×
0BH	W	Write to Holding Register (Extended)	×
0CH	–	Not used.	–
0DH	W	Read Discontinuous Holding Register Contents (Extended)	×
0EH	W	Write to Discontinuous Holding Registers (Extended)	×
0FH	B	Modify Status of Multiple Coils	○
10H	W	Write to Multiple Holding Registers	○

- B: Bit data, W: Word data
- ○: Can be set. ×: Cannot be set.
- Only MW (MB) registers may be used for sending and receiving messages when operating as the Master.
- MB, MW, IB, and IW registers can be used as coils, holding registers, input relays, and input registers, respectively, when operating as a Slave.

[f] PARAM05: Data Address

The requested data address is output to PARAM05.

[g] PARAM06: Data Size

The data size requested by the sending station to be read or written as the number of bits or words is output to PARAM06.

[h] PARAM07: Remote CPU Number

PARAM07 is not used for the MEMOBUS protocol. A value of 0 will be output.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the data address offsets from the receiving station.
The position of the data is shifted backward by the number of words set for the offset at the device receiving the message.

- For details, refer to *A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV* on page A-5.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameters	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter	Protocol Type
			MEMOBUS
01H	Read Coil Status	PARAM08	○
02H	Read Input Relay Status	PARAM09	○
03H	Read Holding Register Contents	PARAM11	○
04H	Read Input Register Contents	PARAM10	○
05H	Modify Status of a Single Coil	PARAM08	○
06H	Write to a Single Holding Register	PARAM11	○
09H	Read Holding Register Contents (Extended)	PARAM11	×
0AH	Read Input Register Contents (Extended)	PARAM10	×
0BH	Write to Holding Register (Extended)	PARAM11	×
0DH	Read Discontinuous Holding Register Contents (Extended)	PARAM11	×
0EH	Write to Discontinuous Holding Registers (Extended)	PARAM11	×
0FH	Modify Status of Multiple Coils	PARAM08	○
10H	Write to Multiple Holding Registers	PARAM11	○

- ○: Can be set. ×: Cannot be set.

[j] PARAM12 and PARAM13: Write Range

These parameters set the range of addresses that can be written for write requested from the sending station. Any write request that exceeds this address range will result in an error.
The write ranges in PARAM12 and PARAM13 are set as word addresses.

- The MP3000/MP2000 Series Machine Controllers store all data for write requested from the sending station to M registers.

The write range parameters can be used to set a range of M registers that can be written using messages. The write range parameters are listed in the following table.

Parameter	Contents	Description
PARAM12	Write range LO	Leading address of the write range
PARAM13	Write range HI	Final address of the write range

Set the write ranges so that the following conditions are satisfied.

0 ≤ Write range LO ≤ Write range HI ≤ Maximum value of M addresses

The write range is valid for the following function codes.

05H (Modify Status of a Single Coil)
06H (Write to a Single Holding Register)
0FH (Modify Status of Multiple Coils)
10H (Write to Multiple Holding Registers)

Example: The following settings would be used to permit writing to the M registers with addresses 1000 to 1999.

PARAM12=1000
PARAM13=1999

With these settings, the receiving station will output an error and not perform the write if a write request is received for any registers not between MW01000 and MW01999.

[k] PARAM14: Reserved by the System (1)

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[l] PARAM15 and PARAM16: Reserved by the System (2)

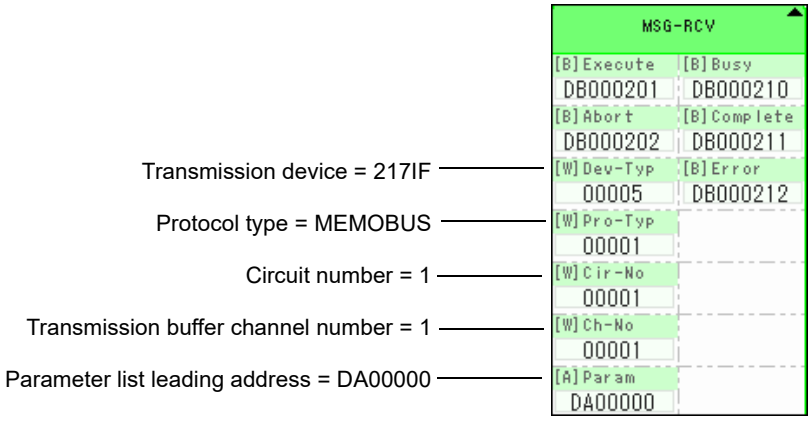
These parameters are used by the system. Do not change the values set for these parameters.

3.7.5 Function Setting Example and Parameter Details for the MELSEC Protocol

This section provides an MSG-RCV function setting example and describes the parameter list in detail for serial communication (217IF) using the MELSEC protocol.

(1) MSG-RCV Function Setting Example

This example shows the settings for using the 217IF as the transmission device.



To use the MELSEC protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 217IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *3.7.2 Inputs and Outputs for the Message Receive Function* on page 3-80.

(2) Parameter List for MELSEC Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	08	IN	Coil offset
01	OUT	Status	09	IN	Input relay offset
02	OUT	Station Number	10	IN	Input register offset
03	OUT	Option	11	IN	Holding register offset
04	OUT	Function Code	12	IN	Write range LO
05	OUT	Data address	13	IN	Write range HI
06	OUT	Data size	14	SYS	Reserved by the system (1).
07	OUT	Remote CPU number	15 and 16	SYS	Reserved by the system (2).

IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for MELSEC Protocol

This section describes the parameter list in detail for using the MELSEC protocol.

[a] PARAM00: Processing Result

Refer to *3.7.4 (3) [a] PARAM00: Processing Result* on page 3-85.

[b] PARAM01: Status

Refer to *3.7.4 (3) [b] PARAM01: Status* on page 3-86.

[c] PARAM02: Station Number

Refer to *3.7.4 (3) [c] PARAM02: Station Number* on page 3-87.

[d] PARAM03: Option

Options specific to each transmission device are output here.
PARAM03 is not used for the MELSEC protocol.

[e] PARAM04: Function Code

The function code that was received is output to PARAM04.
The function codes used for the MELSEC protocol are listed in the following table.

<Function Codes for MELSEC>

Common MELSEC ACPU Command	Function Code	Applicable Data Type	Function
WR	01H/02H	B	Reads data from bit-type devices in 16-point units.
	03H/04H	W	Reads data from word-type devices in 1-point units.
WW	0FH	B	Writes data to bit-type devices in 16-point units.
	10H	W	Writes data to word-type devices in 1-point units.
TT	08H	–	Executes a loopback test.

- B: Bit data, W: Integer data
- The special commands for the AnCPU are not supported. Commands for expanded file registers are not supported.

[f] PARAM05: Data address

The requested data address is output to PARAM05.

[g] PARAM06: Data Size

The data size requested by the sending station to be read or written as the number of bits or words is output to PARAM06.

[h] PARAM07: Remote CPU Number

Refer to *3.7.4 (3) [h] PARAM07: Remote CPU Number* on page 3-88.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the data address offsets from the sending station.
The position of the data is shifted backward by the number of words set for the offset at the device receiving the message.

- For details, refer to *A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV* on page A-5.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter
01H	Read Coil Status	PARAM08
02H	Read Input Relay Status	PARAM09
03H	Read Holding Register Contents	PARAM11
04H	Read Input Register Contents	PARAM10
0FH	Modify Status of Multiple Coils	PARAM08
10H	Write to Multiple Holding Registers	PARAM11

[j] PARAM12 and PARAM13: Write Range

These parameters set the range of addresses that can be written for write requested from the sending station. Any write request that exceeds this address range will result in an error.
The write ranges in PARAM12 and PARAM13 are set as word addresses.

- The MP3000/MP2000 Series Machine Controllers store all data for write requested from the sending station to M registers.

The write range parameters can be used to set a range of M registers that can be written using messages. The write range parameters are listed in the following table.

Parameter	Contents	Description
PARAM12	Write range LO	Leading address of the write range
PARAM13	Write range HI	Final address of the write range

Set the write ranges so that the following conditions are satisfied.

0 ≤ Write range LO ≤ Write range HI ≤ Maximum value of M addresses

The write range is valid for the following function codes.

0FH (Modify Status of Multiple Coils)
10H (Write to Multiple Holding Registers)

Example: The following settings would be used to permit writing to the M registers with addresses 1000 to 1999.
PARAM12=1000
PARAM13=1999
With these settings, the receiving station will output an error and not perform the write if a write request is received for any registers not between MW01000 and MW01999.

[k] PARAM14: Reserved by the System (1)

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[l] PARAM15 and PARAM16: Reserved by the System (2)

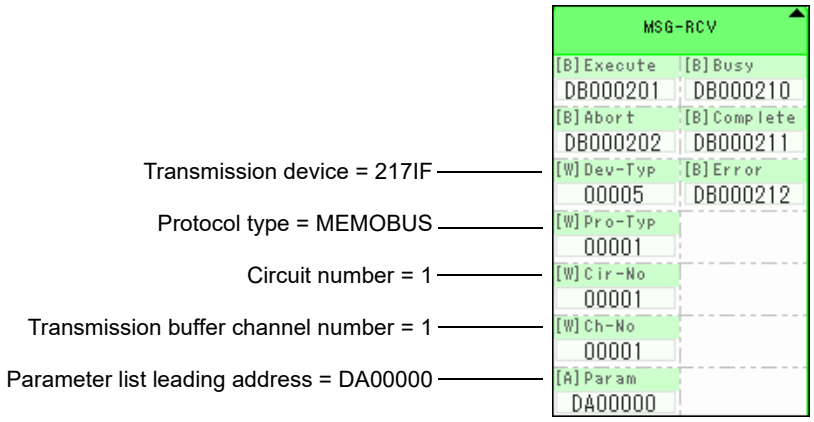
These parameters are used by the system. Do not change the values set for these parameters.

3.7.6 Function Setting Example and Parameter Details for the OMRON Protocol

This section provides an MSG-RCV function setting example and describes the parameter list in detail for serial communication (217IF) using the OMRON protocol.

(1) MSG-RCV Function Setting Example

This example shows the settings for using the 217IF as the transmission device.



To use the OMRON protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 217IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *3.7.2 Inputs and Outputs for the Message Receive Function* on page 3-80.

(2) Parameter List for OMRON Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	08	IN	Coil offset
01	OUT	Status	09	IN	Input relay offset
02	OUT	Station Number	10	IN	Input register offset
03	OUT	Option	11	IN	Holding register offset
04	OUT	Function Code	12	IN	Write range LO
05	OUT	Data address	13	IN	Write range HI
06	OUT	Data size	14	SYS	Reserved by the system (1).
07	OUT	Remote CPU number	15 and 16	SYS	Reserved by the system (2).

IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for OMRON Protocol

This section describes the parameter list in detail for using the OMRON protocol.

[a] PARAM00: Processing Result

Refer to *3.7.4 (3) [a] PARAM00: Processing Result* on page 3-85.

[b] PARAM01: Status

Refer to *3.7.4 (3) [b] PARAM01: Status* on page 3-86.

[c] PARAM02: Station Number

Refer to *3.7.4 (3) [c] PARAM02: Station Number* on page 3-87.

[d] PARAM03: Option

Options specific to each transmission device are output here.
PARAM03 is not used for the OMRON protocol.

[e] PARAM04: Function Code

The function code that was received is output to PARAM04.
The function codes used for the OMRON protocol are listed in the following table.

<Function Codes for OMRON>

OMRON Header Code	Function Code	Applicable Data Type	Function
RR	01H	B	Reads I/O relays, internal auxiliary relays, and special auxiliary relays.
RD	03H	W	Reads data memory area.
WR	0FH	B	Writes I/O relays, internal auxiliary relays, and special auxiliary relays.
RD	10H	W	Writes data memory area.
TS	08H	–	Test

- B: Bit data, W: Integer data

[f] PARAM05: Data address

The requested data address is output to PARAM05.

[g] PARAM06: Data Size

The data size requested by the sending station to be read or written as the number of bits or words is output to PARAM06.

[h] PARAM07: Remote CPU Number

Refer to *3.7.4 (3) [h] PARAM07: Remote CPU Number* on page 3-88.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the data address offsets from the sending station.
The position of the data is shifted backward by the number of words set for the offset at the device receiving the message.

- For details, refer to *A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV* on page A-5.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter
01H	Read Coil Status	PARAM08
03H	Read Holding Register Contents	PARAM11
0FH	Modify Status of Multiple Coils	PARAM08
10H	Write to Multiple Holding Registers	PARAM11

[j] PARAM12 and PARAM13: Write Range

These parameters set the range of addresses that can be written for write requested from the sending station. Any write request that exceeds this address range will result in an error.
The write ranges in PARAM12 and PARAM13 are set as word addresses.

- The MP3000/MP2000 Series Machine Controllers store all data for write requested from the sending station to M registers.

The write range parameters can be used to set a range of M registers that can be written using messages. The write range parameters are listed in the following table.

Parameter	Contents	Description
PARAM12	Write range LO	Leading address of the write range
PARAM13	Write range HI	Final address of the write range

Set the write ranges so that the following conditions are satisfied.

$0 \leq$ Write range LO $\leq$ Write range HI $\leq$ Maximum value of M addresses

The write range is valid for the following function codes.

0FH (Modify Status of Multiple Coils)
10H (Write to Multiple Holding Registers)

Example: The following settings would be used to permit writing to the M registers with addresses 1000 to 1999.
PARAM12=1000
PARAM13=1999
With these settings, the receiving station will output an error and not perform the write if a write request is received for any registers not between MW01000 and MW01999.

[k] PARAM14: Reserved by the System (1)

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[l] PARAM15 and PARAM16: Reserved by the System (2)

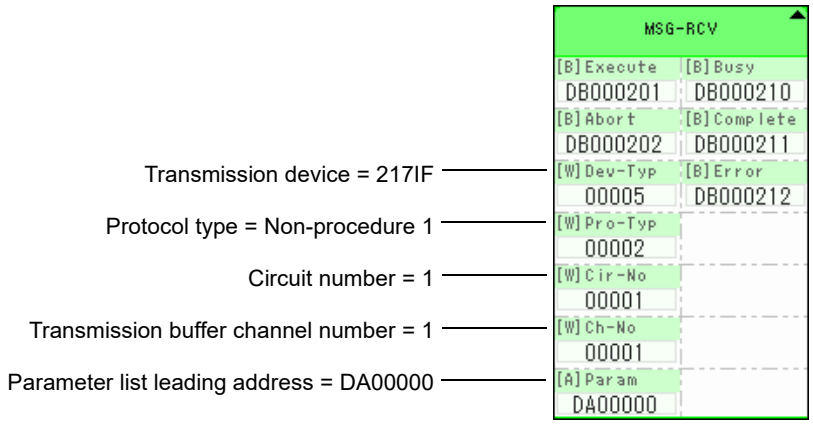
These parameters are used by the system. Do not change the values set for these parameters.

3.7.7 Function Setting Example and Parameter Details for the Non-procedure Protocol

With the non-procedure protocol, the received data is not converted and stored as is in the M registers. This enables receiving data using any protocol of the remote station.
This section provides an MSG-RCV function setting example and describes the parameter list in detail for serial communication (217IF) using the non-procedure protocol.

(1) MSG-RCV Function Setting Example

This example shows the settings for using the 217IF as the transmission device.



The protocol type (Pro-Typ) is set to 2 to use non-procedure protocol 1 and to 3 to use non-procedure protocol 2.
The circuit number is set to the circuit number assigned to the 217IF transmission device.
The transmission buffer channel number is set to 1.
For details on settings, refer to *3.7.2 Inputs and Outputs for the Message Receive Function* on page 3-80.

(2) Parameter List for Non-procedure Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	07	OUT	(Not used.)
01	OUT	Status	08 to 11	IN	(Not used.)
02	OUT	(Not used.)	12	IN	Write range LO
03	OUT	Option	13	IN	Write range HI
04	OUT	(Not used.)	14	SYS	Reserved by the system (1).
05	OUT	(Not used.)	15 and 16	SYS	Reserved by the system (2).
06	OUT	Data size			

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for Non-procedure Protocol

This section describes the parameter list in detail for using the non-procedure protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARM00. The lower-place byte is for system analysis.

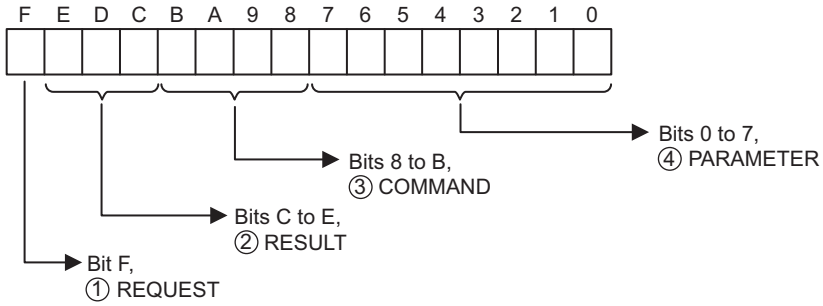
Value of Processing Result	Meaning
00xxH	Processing in progress (Busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	–	–
82xxH	Address setting error	The following setting is not within the valid range. Check the settings. PARAM12 (register offset)
83xxH	Data size error	The size of the received data is not within the allowable range. Check the data size from the sending station.
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-RCV function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-RCV function.
86xxH	–	–
87xxH	–	Reserved by the system.
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-RCV function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-RCV function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-RCV function is output to these bits.

Code	Abbreviation	Meaning
1	SEND_OK	Sending has been completed normally.
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-RCV function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send general-purpose message.
2	U_REC	Receive general-purpose message.
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response. (For MEMOBUS Protocol)
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response. (For MEMOBUS Protocol)
C	MR_SEND	Send MEMOBUS response (For MEMOBUS Protocol)

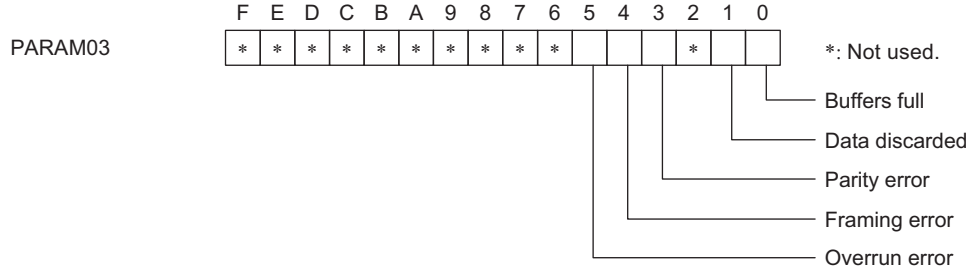
④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error).

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Connection number out of range
	02	MEMOBUS response receive monitor time error (For MEMOBUS Protocol)
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error (For MEMOBUS Protocol)

[c] PARAM03: Option

Reception status is output here when an error occurs.
This parameter is used only when the non-procedure FD protocol is selected for the communications protocol in the 217IF detailed definitions. You cannot use it if you select the standard non-procedure protocol.



Bit	Name	Meaning	Description
Bit 0	Buffers full	Turns ON when all 20 receive buffers have been used and reception is no longer possible. 0: No buffer overflow 1: Buffer overflow	The buffers became full before the data that is currently being read. The data could not be received.
Bit 1	Data discarded	Turns ON when received data is not all read out with the MSG-RCV function within 400 ms. The unread data is discarded by the system. 0: Data not discarded for timeout 1: Data discarded for timeout	A timeout occurred before the data that is currently being read. The previous data was discarded.
Bit 3	Parity error	Turns ON when a parity error occurs. 0: No parity error 1: Parity error	A parity error occurred in the data that is currently being read. The data could not be received.
Bit 4	Framing error	Turns ON when a framing error occurs. 0: No framing error 1: Framing error	A framing error occurred in the data that is currently being read. The data could not be received.
Bit 5	Overrun error	Turns ON when an overrun error occurs. 0: No overrun error 1: Overrun error	An overrun error occurred in the data that is currently being read. The data could not be received.

[d] PARAM06: Data Size

The size of received data is output to this parameter.
In the non-procedure 1 communication mode, the data size is output as the number of words.
In the non-procedure 2 communication mode, the data size is output as the number of bytes.

[e] PARAM12 and PARAM13: Write Range

With the non-procedure protocol, the received continuous data is stored in M registers. An error will occur if there is a request to write a value that does not correspond to the addresses from PARAM12 (Write Range LO) to PARAM13 (Write Range HI).
Set PARAM12 and PARAM13 (Write Range) as word addresses.

Set the write ranges so that the following conditions are satisfied.

$0 \leq$ Write range $\leq$ Maximum value of M addresses

Example: The following settings would be used to permit writing to the M registers with addresses 1000 to 1999.
PARAM12 = 1000
PARAM13 = 1999

[f] PARAM14: Reserved by the System (1)

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[g] PARAM15 and PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

4

218IF-01 Module

This chapter describes the 218IF-01 Module in detail and includes definition methods and programming examples.

- Before reading this chapter, be sure to read *Chapter 1 Starting Communication Modules*.

4.1 Overview ----- 4-2
4.1.1 Appearance and Connectors ----- 4-2
4.1.2 System Configuration Example ----- 4-3
4.1.3 Module Specifications ----- 4-4
4.1.4 Indicators ----- 4-6
4.1.5 Switch Settings ----- 4-7
4.1.6 Default Parameters at Initial Startup ----- 4-8

4.2 Connection Specifications ----- 4-9
4.2.1 Connector Specifications ----- 4-9
4.2.2 Cable Specifications ----- 4-9

4.3 RS-232C Communication ----- 4-10

4.4 218IF Transmission Definitions ----- 4-11
4.4.1 218IF Transmission Configuration Window ----- 4-11
4.4.2 Transmission Parameters Tab Page ----- 4-13
4.4.3 Status Tab Page ----- 4-20
4.4.4 Saving and Deleting 218IF Transmission Definitions ----- 4-23

4.5 Ethernet Communication ----- 4-24
4.5.1 Overview ----- 4-24
4.5.2 Ethernet Communication Protocol ----- 4-24
4.5.3 MEMOBUS Protocol ----- 4-25
4.5.4 Extended MEMOBUS Protocol ----- 4-26
4.5.5 MELSEC Protocol ----- 4-27
4.5.6 MODBUS/TCP Protocol ----- 4-31
4.5.7 Non-Procedure Protocol ----- 4-32

4.6 Connection Examples for Ethernet Communication ----- 4-34
4.6.1 Connecting 218IF-01 Modules ----- 4-34
4.6.2 Connection to MELSEC ----- 4-39
4.6.3 Connection to a Personal Computer ----- 4-43

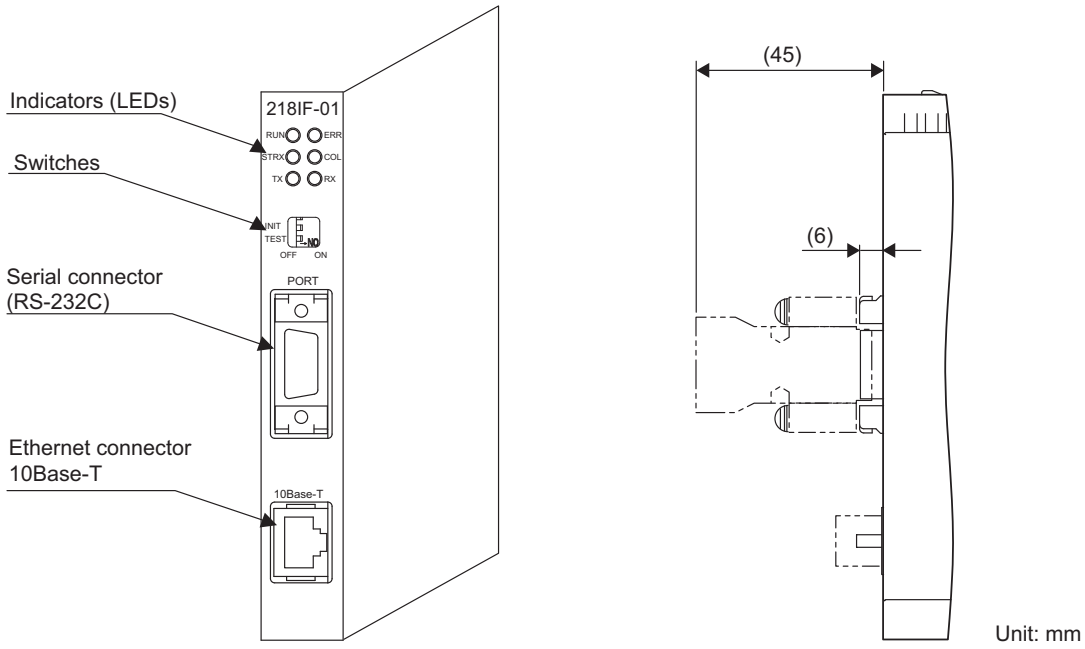
4.7 Message Send Function and Message Receive Function ----- 4-47

4.1 Overview

The 218IF-01 Module is a Communication Module that has an RS-232C serial interface and an Ethernet interface mounted in it. Personal computers, HMI devices, and controllers manufactured by other companies can be connected to the 218IF-01 Module via the PORT serial interface and 10Base-T connectors.
Message communication and engineering communication are supported. The MEMOBUS, MELSEC, OMRON, and non-procedure protocols are supported for serial communication, and the Extended MEMOBUS, MELSEC, MODBUS/TCP, and non-procedure protocols are supported for Ethernet communication.
The Ethernet interface on the 218IF-01 supports the Ethernet and CP-218 port types are supported for the Communication Platform.
The rest of this chapter describes the appearance, specifications, and switch settings of the 218IF-01 Module.

4.1.1 Appearance and Connectors

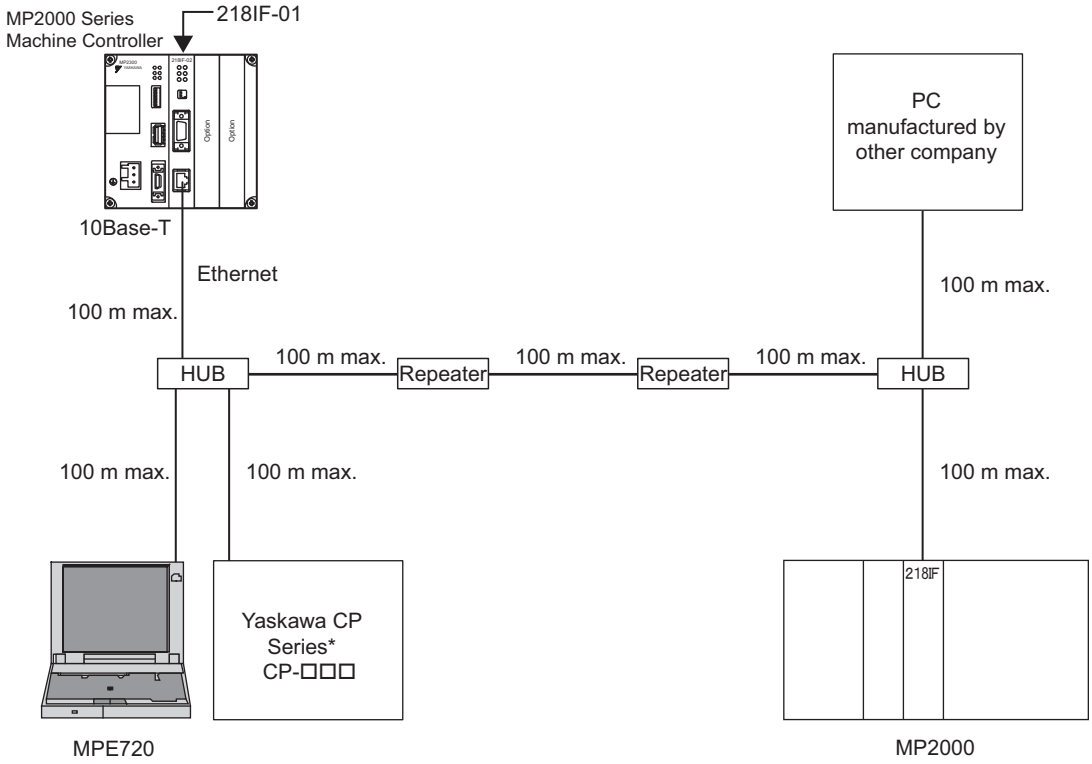
The following diagram shows the appearance of the 218IF-01 Module and gives the external dimensions of the connectors.



4.1.2 System Configuration Example

(1) Example of Ethernet Connections

The following example shows an Ethernet connection to the 10Base-T port.

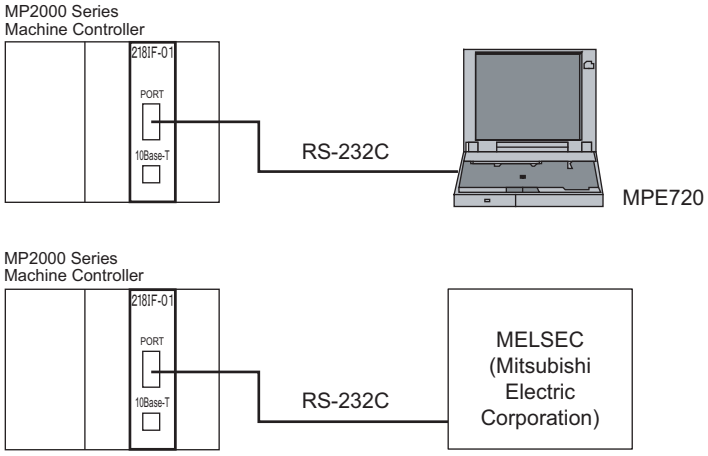


- Maximum length between two nodes: 100 m, Total length between all nodes: 500 m, Hubs between nodes: 4 max.

* Yaskawa CP Series: Large-scale Programmable Logic Controller Series (mainly used for plant control)

(2) Examples of Serial Connections

The following examples show serial connections to the PORT connector (RS-232C). The 218IF-01 Module is connected to a personal computer or a controller manufactured by another company on a one-to-one basis.



4.1.3 Module Specifications

This section provides the specifications of the 218IF-01 Module.

(1) Hardware Specifications

Item	Specification
Name	218IF-01
Model Number	JAPMC-CM2300
Communication Ports	RS-232C: 1 port (PORT)
	Ethernet: 1 port (10Base-T)
Indicators	Module status LED indicators RUN (green) ERR (red) STRX (green) COL (green) TX (green) RX (green)
Switches	INIT TEST
Dimensions (mm)	125 x 95 mm (H x D)
Mass	85 g

(2) Transmission Specifications

■ RS-232C Transmission Specifications

Item	Specification
Connector	9-pin D-sub (female)
Transmission Distance	15 m max.
Baud Rate	9600 or 19200 bps
Access Mode	Asynchronous (start-stop synchronization)
Communication Type	Half duplex
Communication Modes	Message communication and engineering communication
Communication Protocols	MEMOBUS, MELSEC, OMRON, and Non-procedure
Media Access Control Method	1:1
Transmission Format (Can be set.)	Data bit length: 7 or 8 bits Stop bits: 1 or 2 bits Parity: Odd, even, or none

■ Ethernet Transmission Specifications

Item	Specification
Interface	10Base-T: RJ-45
Isolation Method	Transformer coupled
Transmission Distance	100 m/segment, Total length: 500 m (when 4 repeaters are connected)
Baud Rate	10 Mbps
Access Mode	IEEE 802.3, CSMA/CD
Communication Type	Half duplex *
Communication Modes	Message communication and engineering communication
Communication Protocols	MEMOBUS, Extended MEMOBUS, MELSEC, MODBUS/TCP, and non-procedure
Frames	Ethernet, Ver. 2 (DIX specifications)
Connection Types	TCP, UDP, IP, and ARP
Max. Number of Nodes	Depends on the hubs and network.
Max. Number of Transmission Words	512 words (1024 Bytes)
Max. Number of Segments	5
Max. Number of Connections	20 (max. of 10 connections for simultaneous communication)

* Ethernet communication for the 218IF-01 Module must be half-duplex, 10Base-T communications. The following table lists the communication types of remote devices (the hub or communication device directly connected to the 218IF-01) with which the 218IF-01 can communicate.

218IF-01's Communication Type	Remote Device's Communication Type				
Half-duplex 10Base-T	Auto-negotiation	Half-duplex 10Base-T	Full-duplex 10Base-T	Half-duplex 100Base-TX	Full-duplex 100Base-TX
	Depends on remote device.	Communicates using half-duplex 10Base-T.	Communication not possible. (An error will occur.)	Communication not possible	Communication not possible.

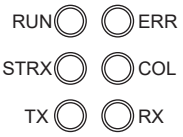
If the remote device supports half-duplex 10Base-T communication, then communication is possible.
If the remote device supports auto-negotiation, then the communication type will be selected automatically.
Communication is not possible with remote devices that support only full-duplex 10Base-T communication or with remote devices that support only 100Base-TX communication. The remote device must therefore use either auto-negotiation or half-duplex 10Base-T communication.

(3) Operating Environment Specifications

Item		Specification
Environmental Conditions	Ambient Operating Temperature	0 to 55 °C
	Ambient Storage Temperature	-25 to 85 °C
	Ambient Operating Humidity	30% to 95% (with no condensation)
	Ambient Storage Humidity	5% to 95% (with no condensation)
	Pollution Level	Pollution level 2 (conforming to JIS B 3502)
	Corrosive Gas	There must be no combustible or corrosive gas.
	Operating Altitude	2,000 m above sea level or lower
Mechanical Operating Conditions	Vibration Resistance	Conforming to JIS B 3502 Vibration and acceleration: 10 to 57 Hz with single-amplitude of 0.075 mm 57 to 150 Hz with fixed acceleration of 9.8 m/s^2 10 sweeps each in X, Y, and Z directions (sweep time: 1 octave/min)
	Shock Resistance	Conforming to JIS B 3502 Peak acceleration of 147 m/s^2 (15 G) twice for 11 ms each in the X, Y, and Z directions
Electrical Operating Conditions	Noise Resistance	Conforming to EN 61000-6-2, EN 61000-6-4, EN 55011 (Group 1 Class A)
Installation Requirements	Ground	Ground to 100 Ω max.
	Cooling Method	Natural cooling

4.1.4 Indicators

The following table shows the status of the 218IF-01 Module shown by the LED indicators.



Indicator	Color	Meaning When Lit	Meaning When Not Lit
RUN	Green	Operating normally	An error has occurred.
ERR	Red	A failure has occurred (lit or blinking).	Normal
STRX	Green	Sending or receiving RS-232C data	No RS-232C data communication
COL	Green	Ethernet collision has occurred.	No collisions have occurred.
TX	Green	Sending Ethernet data	Not sending Ethernet data
RX	Green	Receiving Ethernet data	Not receiving Ethernet data

■ Status Indication by LED Indicators

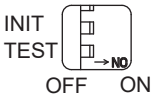
The following table shows the LED indications and their meanings for operating status and errors.

Status	Indicators						Meaning	Description
	RUN	ERR	COL	STRX	TX	RX		
Power ON	○	●	○	○	○	○	Initializing	This is the status immediately after the power to the 218IF-01 is turned ON (initializing). When initialization has been completed, the ERR indicator goes out and the 218IF-01 enters initialized status.
Initialized	●	○	○	–	○	–	Waiting for initial settings	The 218IF-01 starts normally and waits for initial settings from the controller (CPU). The STRX and RX indicators may be lit depending on the connection status.
Normal	●	○	○	–	–	–	Operating normally	The 218IF-01 is performing Ethernet communication normally. The STRX indicator is lit during serial communication.
Error	●	○	●	–	●	●	Transmission error	A collision was detected in Ethernet communication. The STRX indicator is lit during serial communication.
	–	★	○	–	○	–	Hardware failure (No. of blinks:) 1: ROM diagnosis error 2: RAM diagnosis error 3: Shared memory diagnosis error	The 218IF-01 hardware failed. The Module must be replaced.
	–	★	○	–	○	–	Software or hardware error (No. of blinks:) 15: Watchdog timeout error	An error occurred in the 218IF-01 hardware or software. If a software error occurred, there is a good chance that the processing load was too high. Example: The reception processing load was too high because of high Ethernet traffic.

- ● : Lit
  ○ : Not lit
  ★ : Blinking
  – : Not relevant

4.1.5 Switch Settings

The following table shows the 218IF-01 Module switch settings.



Label (Switch No.)	Name	Status	Function	Factory Setting
INIT (2)	Initial Startup	ON	For engineering communication. Starts using the default parameters except setting for automatic reception. Given higher priority than the Machine Controller flash startup and self-configuration startup.	OFF
		OFF	Set to OFF for Machine Controller flash startup or self-configuration startup.	
TEST (1)	TEST	ON	System use.	OFF
		OFF	Normal operation (Always leave set to OFF.)	

- Always leave the unused switches (3 and 4) set to OFF.

- If the unused switch (4) is turned ON, the RUN indicator will light up, but communication will not be enabled.

4.1.6 Default Parameters at Initial Startup

At initial startup (INIT switch set to ON), the following parameter settings will be made.

■ Ethernet Interface (The setting of INIT switch to ON is valid only when the power is turned ON.)

Item	Default Setting
This Station IP Address (Local Station's IP Address)	192.168.1.1
Subnet Mask	255.255.255.0
Gateway IP Address	0.0.0.0
System Port (Engineering Port)	10000 (UDP)
TCP Zero Window Timer Value	3 s
TCP Retry Time	500 ms
TCP Close Time	60 s
IP Assemble Timer	30 s
Max. Packet Length	1500 bytes

■ RS-232C Interface (The setting of INIT switch to ON is valid while the power is being supplied.)

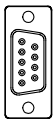
Item	Default Setting		
Communication Protocol	MEMOBUS		
Master/Slave	Slave		
Device Address	1		
Serial I/F	RS-232C		
Communication Mode	RTU		
Data Length	8 bits		
Parity	Even		
Stop Bits	1 bit		
Baud Rate	19.2 kbps		
Sending (Send Delay)	Disable		
Receive Monitor Time	Disable		
Automatically Reception (Automatic Reception)	Enable		
Interface Register Settings in the Slave		Leading Register	Number of Words
	Read Input Relay Status	IW0000	32768
	Read Input Register Status	IW0000	32768
	Read/Write Coil Status	MW00000	65535
	Read/Write Holding Register Status	MW00000	65535
	Coil/holding Register Write Range	LO MW00000 HI MW65534	

4.2 Connection Specifications

4.2.1 Connector Specifications

This section provides the connector specifications for the 218IF-01 Module.

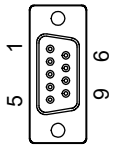
(1) Connector Specifications

Connector Shape	Name	Connector Name	No. of Pins	Connector Model		
				Module	Cable	Manufacturer
PORT 	RS-232C	PORT	9	17LE-13090-27 (D2BC) D-sub 9-pin female connector	17JE-23090-02 (D8B) 9-pin D-sub male connector (Shell included.)	DDK Ltd.
10Base-T	Ethernet	10Base-T	8	555153-1 10Base-T Ethernet Connector (modular jack)	–	Tyco Electronics AMP K.K.

(2) Connector Pin Arrangement

■ PORT Connector

The PORT connector is used to connect the MP2000 Series Machine Controller to computers and HMI devices via an RS-232C connection.



Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	FG	Frame ground	6	–	–
2	SD	Send data	7	SG	Signal ground (0V)
3	RD	Receive data	8	–	–
4	RS	Ready to send	9	ER	Data terminal ready
5	CS	Clear to send	–	–	–

■ Ethernet Connector (10Base-T)

The Ethernet connector is used to connect the MP2000 Series Machine Controller to computers and HMI devices via an Ethernet (10Base-T) connection.

Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	TXD+	Send data +	5	–	–
2	TXD-	Send data -	6	RXD-	Receive data -
3	RXD+	Receive data +	7	–	–
4	–	–	8	–	–

4.2.2 Cable Specifications

This section provides the cable specifications for the 218IF-01 Module.

(1) RS-232C Cable (for MPE720 Engineering)

Model Number	Length	External Appearance (JEPMC-W5311-□□-E)
JEPMC-W5311-03-E	2.5 m	
JEPMC-W5311-15-E	15 m	

(2) Ethernet Cable

Yaskawa does not provide Ethernet cables. Obtain a commercially available 10Base-T cross or straight cable.

4.3 RS-232C Communication

Refer to *Chapter 3 Serial Communication* for information on the transmission definitions, connection examples, programming examples, the Message Send function, the Message Receive function, and other details on communication through the RS-232C port on the 218IF-01 Module.

4.4 218IF Transmission Definitions

To perform Ethernet communication with the MP2000 Series Machine Controller, the 218IF transmission (Ethernet communication) definition file must be created.
This section describes how to define 218IF transmissions using MPE720 windows.

4.4.1 218IF Transmission Configuration Window

To define the 218IF transmissions, first the **Module Configuration** Window is opened on the MPE720, and then the **218IF Transmission Configuration** Window is opened.

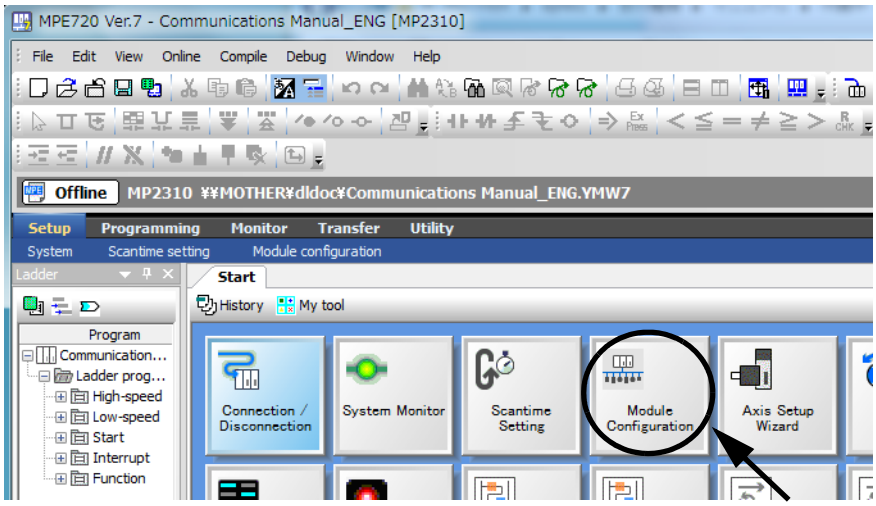
(1) Displaying the Module Configuration Window

Use the following procedure to display the **Module Configuration** Window.

1. Start the MPE720 on the personal computer connected to the Machine Controller and open the project file.

- For information on starting the MPE720, refer to *Machine Controller MP2000/MP3000 Series Engineering Tool MPE720 Version 7 User's Manual (Manual No.: SIEP C880761 03)*.

2. Click the **Module Configuration** Button on My Tool View.



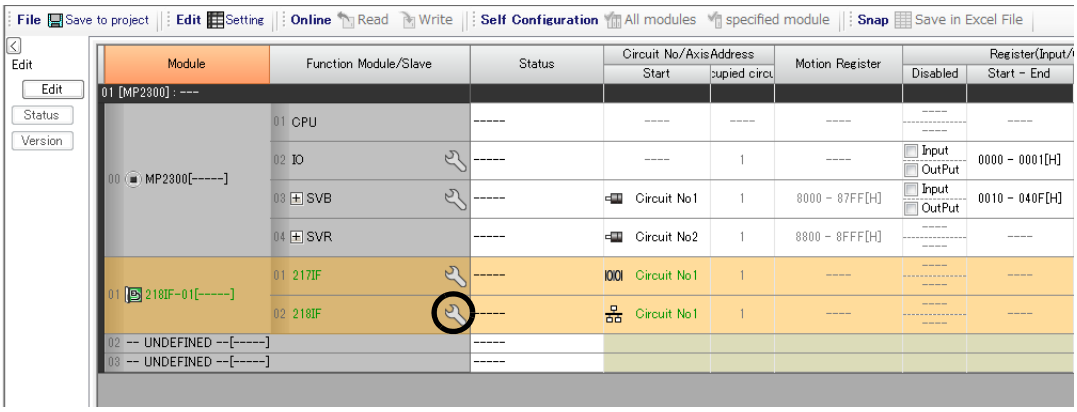
The **Module Configuration** Window will be displayed.

(2) Displaying the 218IF Transmission Configuration Window from the Module Configuration Window

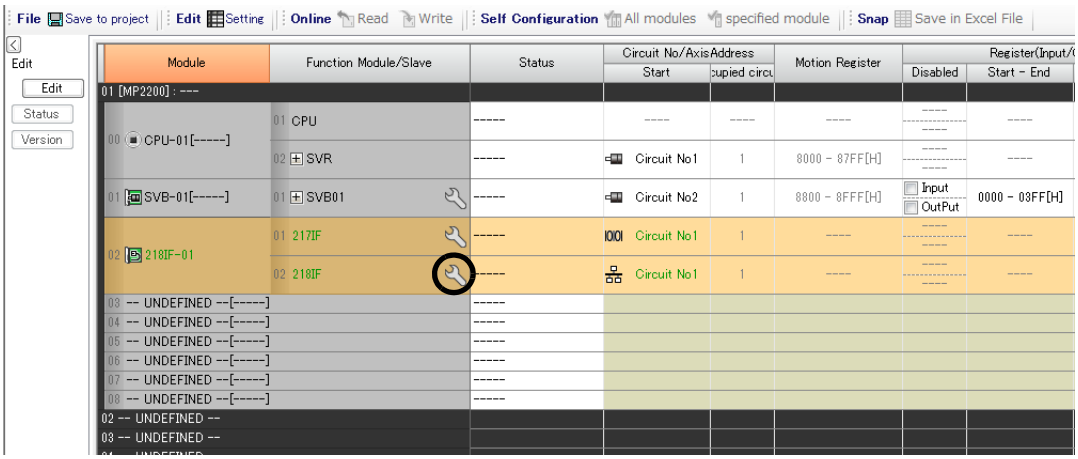
As the result of executing self-configuration, all Option Modules connected to the Machine Controller will be displayed in the **Controller** Area of the **Module Configuration** Window. (Refer to *1.5 Executing Self-configuration* on page 1-26.)

In the **Controller** Area, select the cell for 218IF-01, and then double-click the slot number cell for the 218IF in the **Module Details** Area. The **218IF Transmission Configuration** Window (see next page) will be displayed.

<MP2300/MP2300S/MP2310 Module Configuration Window>



<MP2100M/MP2200 Module Configuration Window>



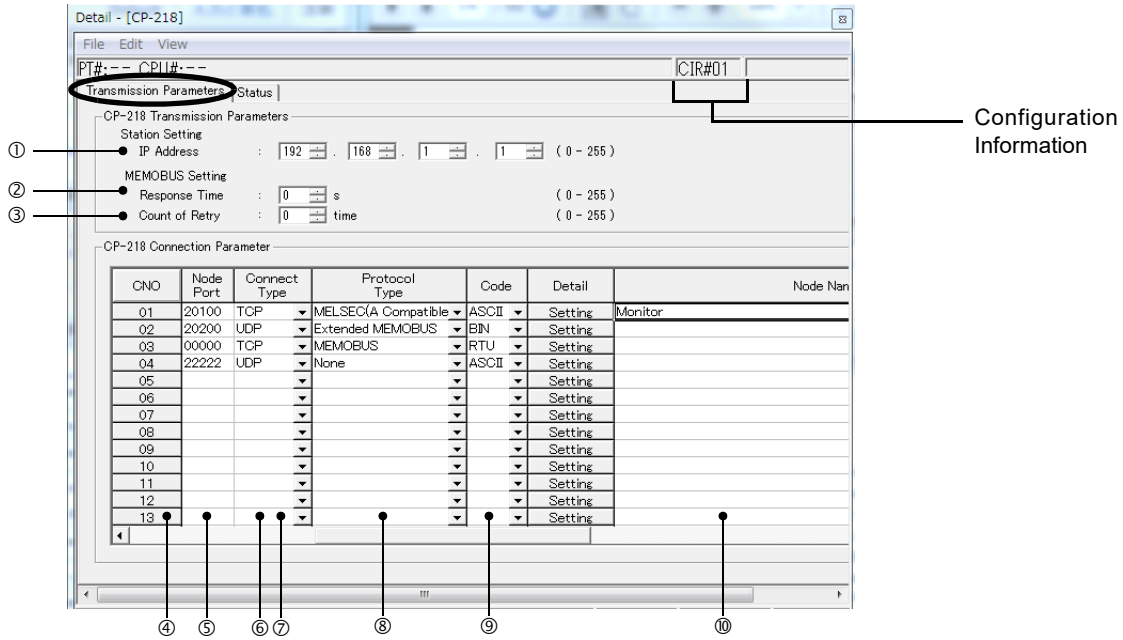
In Online Mode, the 218IF transmission definitions data saved in the Machine Controller are displayed. In Offline Mode, the definitions data saved in the personal computer on which MPE720 is running is displayed.

The **218IF Transmission Configuration** Window is composed of two tab pages: **Transmission Parameters** and **Status**. These tab pages are used to set the definitions and monitor the settings.

- If the **Transmission Configuration** Window is being opened for the first time, a "new file" message box will be displayed and the **218IF Transmission Configuration** Window will be displayed when the **OK** Button is clicked.
- If the cell for the 217IF in the **Module Details** Area is double-clicked, the **217IF Transmission (serial communication) Configuration** Window will be displayed. Refer to *Chapter 3 Serial Communication* for details on serial communication.

4.4.2 Transmission Parameters Tab Page

Set the parameters required for using Ethernet communication.



■ Configuration Information

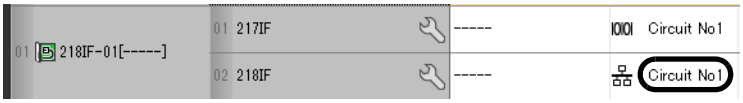
Displays the 218IF-01 Module configuration information. The configuration information is the same as the information displayed in the **Module Details** Area in the **Module Configuration** Window.

RACK#: The rack number of the rack in which the 218IF-01 is defined.

Slot#: The slot number of the slot in which the 218IF-01 is defined.

CIR#: The circuit number of the Ethernet port on the 218IF-01.

- The configuration information can also be changed in the **Module Details** Area of the **Module Configuration** Window. Do not use a circuit number that is already being used for another Ethernet Module (218IF or 218IFB).



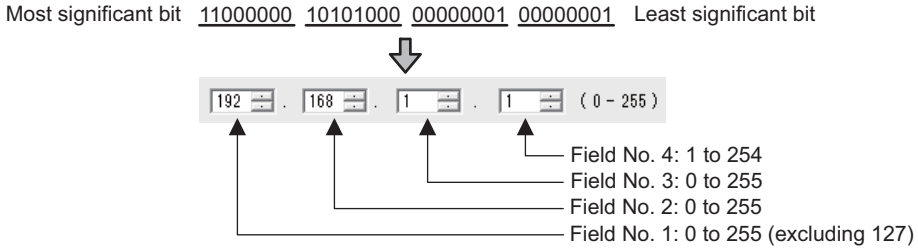
- When the MSG-SND or MSG-RCV function is used, set the circuit number displayed here for **CIR#** (circuit number).

■ Setting the Transmission Parameters

Set the parameters as follows. After making the required settings, save the parameters to flush memory. The settings will be valid after the power is turned OFF and then ON.

① IP Address

Enter the IP address of the local station. There are four 8-bit fields delimited by periods (.). Input a decimal number for each field.



- Do not use the same address as another node in the Ethernet network.
- IP address 192.168.1.1 to 192.168.1.254 are recommended as private addresses. Check with your network administrator when setting an IP address.

② Response Time

Enter the time (0 to 255 s) that the transmission system waits for a response after sending a command using the MSG-SND function.

The **Count of Retry** parameter is not valid if 0 is set for the **Response Time.**

- Enter 0 for the **Response Time** if 0 is set for the **Count of Retry** in the MEMOBUS setting (if **MEMOBUS** is set in the **Protocol Type** column of the **Transmission Parameters** Tab Page.)

③ Count of Retry

Enter the number of retries (0 to 255) to be attempted if a timeout is detected after sending a command with the MSG-SND function. An error is returned for the MSG-SND function if a response is not returned after the set number of retries.

- If only the TCP protocol is set as the connection type, the number of retries does not need to be set. Set parameters ② (**Response Time**) and ③ (**Count of Retry**) to 0.

■ Setting the Connection Parameters

Set the parameters as follows. After making the required settings, save the parameters to flush memory. The settings will be valid after the power is turned OFF and then ON.

④ CNO (Connection Number)

The connection number (1 to 20) is displayed.

With Ethernet communication, the connection number is used to identify remote stations.

The connection numbers corresponds to the **Remote Connection #** (parameter 02) of the MSG-SND and MSG-RCV functions.

⑤ Local Port

Enter the port number of the local station (256 to 65534) for each connection. Do not enter a port number used for another connection or for diagnostics. (Refer to *(4) Local Station and TCP/IP Settings* on page 4-17.) If 0 is entered for this parameter, the data for that connection number is cleared and "---" is displayed in the **Local Port** column.

⑥ Node IP Address (Remote Station's IP Address)

Enter the IP address of the remote station for each connection.

The data input range depends on the field of the IP address.

Field No. 1: 0 to 255 (excluding 127)
Field No. 2: 0 to 255
Field No. 3: 0 to 255
Field No. 4: 1 to 254

Enter 0 in all four fields for the unpassive open mode (see *Explanation* below.)

■ Terminology: Connection

A series of operations to confirm communication and transfer data in one-to-one communication between local station programs and remote station programs.

■ Terminology: Port Number

The port number is used to identify the communication target program in the remote station. There is a one-to-one correspondence between port numbers and communication programs. A port number is entered in the header section of the data, together with the IP address and other information. The remote station transfers the data to the target program identified by the destination port number.

The port number at the local station also enables each local station and remote station communication service program to be identified.

■ Explanation: Unpassive Open Mode

If the remote station's address is set to 000.000.000.000 and the remote station's port number is set to 0, the connection is set in the Unpassive Open mode.

In Unpassive Open mode, the 218IF-01 Module is connected to the station that has accessed the connection number of the 218IF-01 Module, and if more than one station accesses the 218IF-01 Module's connection number, the connection will be established with the station that sent the connection request first. If a connection request is sent from a station while the connection is established with another station in Unpassive Open mode, the connection that was established first is disconnected and a connection will be established with the station that sent the connection request later.

---

⑦ Node Port (Remote Station's Port Number)

Enter the port number (0 or 256 to 65535) of the remote station for each connection. Do not use a combination of the **Node IP Address** and the **Node Port** that is the same as the one assigned for any other connection.

⑧ Connect Type (Connection Type)

Select the transport-layer protocol.

TCP: Communication are performed using TCP (Transmission Control Protocol).

UDP: Communication are performed using UDP (User Datagram Protocol).

⑨ Protocol Type

Select the application-layer protocol for each connection according to the protocol supported by the remote station. The following table outlines each protocol type.

- Refer to *4.5.2 Ethernet Communication Protocol* on page 4-24 for details on the protocols.

Protocol Type	Description
Extended MEMOBUS	Yaskawa's extended MEMOBUS protocol
MEMOBUS	Yaskawa's standard MEMOBUS protocol
MELSEC	Ethernet interface protocol used with Mitsubishi MELSEC A-series General-purpose Programmable Logic Controllers.
Non-procedure protocol	A protocol provided to implement general-purpose message communication Data is sent or received and written to consecutive MW registers without processing.
MODBUS/TCP	Ethernet protocol, proposed by Modicon, used in industrial applications.

⑩ Code

The code of the data to be transmitted at each connection is handled according to the code set at the remote station.

RTU: Specifies RTU mode when the MEMOBUS protocol is being used.

ASCII: Specifies ASCII mode.

BIN: Specifies binary mode.

The code that can be selected is restricted by the selection of the protocol type in the Protocol Type column as shown in the following table.

Protocol Type	Code			Protocol Type	Code		
	RTU	ASCII	BIN		RTU	ASCII	BIN
Extended MEMOBUS	×	○	○	Non-procedure protocol	×	○	○
MEMOBUS	○	○	×	MODBUS/TCP	×	×	○
MELSEC	×	○	○				

- ○: Can be selected. ×: Cannot be selected.

⑪ Node Name (Remote Station Name)

Enter a connection comment of up to 32 characters.

■ Explanation: TCP and UDP

TCP is a connection protocol and UDP is a connectionless protocol.

A connection protocol ensures reliable communication because it performs various steps of communication control, such as arrival checks, error detection and correction, sequence number checks, and send data size control.

In contrast, a connectionless protocol does not provide the procedures that ensure the quality of communication, and data transmissions are performed in only one direction. Connectionless protocols thus provide high-speed communication, but less communication reliability.

The selection of TCP or UDP depends on the requirements for communication. Select TCP if reliability is important and select UDP if speed is important.

---

(3) Setting the Parameters

[a] Setting Default Values

The transmission parameters can be set to the following defaults by selecting ***Edit - Transmission Parameters Default Settings*** from main menu of the **Transmission Parameters** Tab Page of the **Module Configuration** Window.

	Parameter	Default Setting
Transmission Parameters	IP Address	192.168.001.001
	Response Time	0
	Count of Retry	0
This Station Settings and TCP/IP Setting*	Subnet Mask	000.000.000.000
	Gateway IP Address	000.000.000.000
	DIAG. Port No. (Diagnostic Port Number)	10000
	TCP Zero Window Timer Value	3
	TCP Retry Time	500
	TCP Close Time	60
	IP Assemble Time	30
	Max. Packet Length	1500

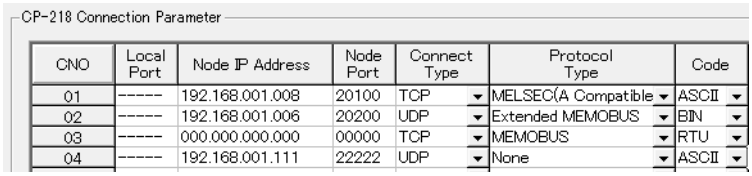
* Refer to ■ *Setting Default Values* on page 4-18 for details on local station and TCP/IP setting parameters.

- The **This Station Settings** and **TCP/IP Settings** will also be set to the above defaults. Refer to *(4) Local Station and TCP/IP Settings* on page 4-17 for details on The **This Station Settings** and **TCP/IP Settings**.
- The default value setting operation does not reset the connection parameters and the Ethernet address of the remote station to the default values.

[b] Deleting Connection Parameters

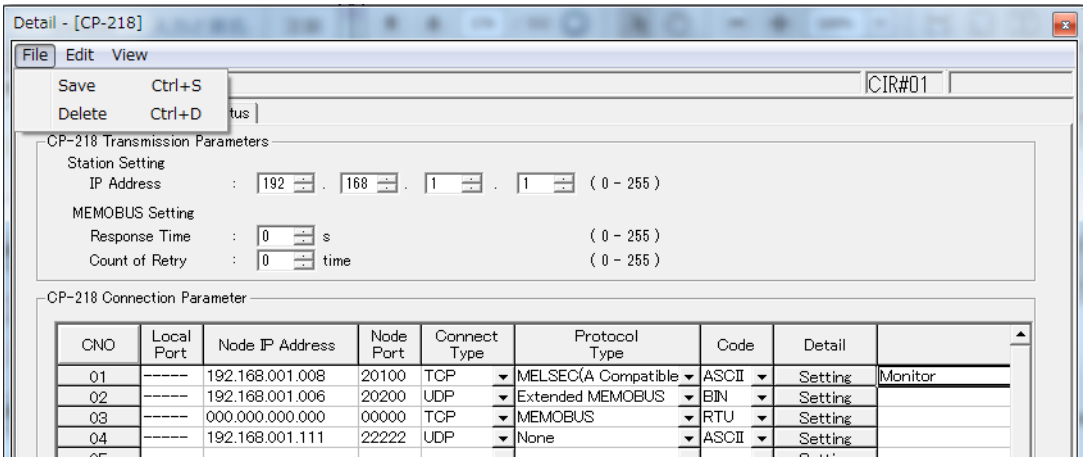
Use the following procedure to delete the assignment data for one connection.

1. Move the cursor to the cell for the connection for which the assignment data is to be deleted.



2. Select ***Edit - Assignment Delete*** from the Main Menu.

The assignment data of the connection selected in step 1 will be deleted.

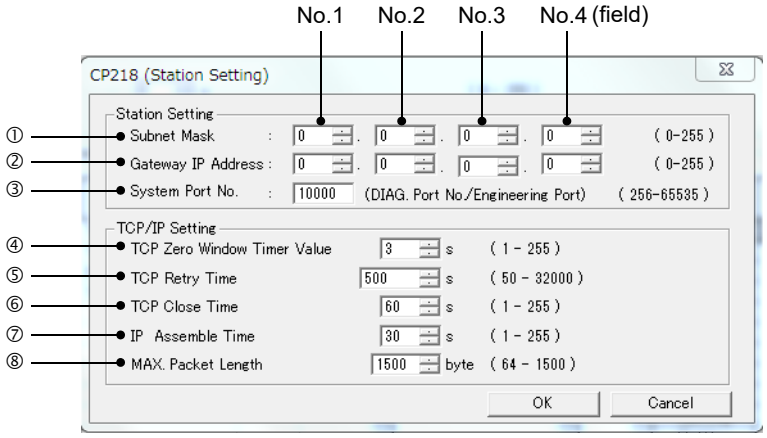


- The assignment data can also be deleted by entering 0 in the Local Port column.

(4) Local Station and TCP/IP Settings

The local station and TCP/IP settings described in this section must be set to connect the 218IF-01 Module and other stations through a server or router.

The **CP-218 (Station Setting)** Dialog Box shown below will be displayed if ***Edit - Local Port: TCP/IP Setting*** is selected from the main menu of the **Transmission Parameters** Tab Page of the **218IF Module Configuration** Window.



- The settings described here are not required if the 218IF-01 Module is connected directly to another station. In that case, use the default settings.

■ Local Station Settings

① Subnet Mask

Enter the subnet mask for the IP address of the local station. The range that can be entered in fields 1 to 4 is 0 to 255. When a subnet mask is not being used, enter 0 in fields 1 to 4.

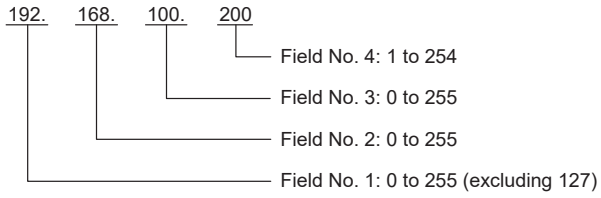
② Gateway IP Address

When communication is performed with other networks connected through a gateway (router), enter the IP address of the gateway.

If there is no gateway or a gateway is not used, enter 0 for the four fields.

The data input range depends on the field.

Setting Example



- Do not set the same address more than once for the IP addresses and gateway IP addresses.

■ Terminology: Subnet Mask

A subnet mask is a mask value used to get the network address of the subnet from the IP address. The result of AND operation of the IP address and the subnet mask will be the subnet address.

Contact your network administrator when setting a subnet mask.

■ Terminology: Gateway IP Address

The IP address of the gateway device (router) through which communication is performed between multiple network segments. Contact your network administrator when setting a gateway IP address.

---

③ System Port No.

Set the 218IF Ethernet port number to use for engineering communication with the MPE720 to between 256 and 65535. The default is 10000.

- If this value is changed, the port set in the detail settings of the **Logical Port Setting** Dialog Box for the MPE720 Communication Platform must also be changed.
- Do not set the same port number for the system port and the local port. (Refer to ■ *Setting the Connection Parameters* on page 4-14.)

■ TCP/IP Settings

④ TCP Zero Window Timer Value

Enter the time (1 to 255) before sending the Transmission Window confirmation packet after the value of the **TCP Transmission** Window size is zeroed. The default is 3 seconds.

⑤ TCP Retry Time

Enter the time (50 to 32000) before resending the data if ACK (acknowledgment) is not returned when opening a connection or sending data in the TCP mode. The default is 500 ms.

⑥ TCP Close Time

Enter the time (1 to 255) for the 218IF transmission system waits while the remote station normally closes the connection in TCP mode. The default is 60 seconds.

⑦ IP Assemble Time

Enter the time (1 to 255) for the IP to wait to receive the next fragment of the data when it receives fragmented data. The default is 30 seconds.

⑧ MAX. Packet Length

Enter the maximum length of packet (64 to 1500) allowed in the 218IF transmission system. The default is 1500.

- When setting a value smaller than the default (1500 bytes) for the maximum packet length with non-procedure protocol set for the application protocol, set a value at least 40 bytes more than the number of data item (bytes) to be sent.

■ Setting Default Values

The transmission parameters can be set to the following defaults by selecting ***Edit - Transmission Parameters Default Settings*** from main menu of the **Transmission Parameters** Tab Page of the **Module Configuration** Window.

	Parameter	Default Setting
Transmission Parameters	IP Address	192.168.001.001
	Response Time	0
	Count of Retry	0
This Station Settings and TCP/IP Setting	Subnet Mask	000.000.000.000
	Gateway IP Address	000.000.000.000
	DIAG. Port No. (Diagnostic Port Number)	10000
	TCP Zero Window Timer Value	3
	TCP Retry Time	500
	TCP Close Time	60
	IP Assemble Time	30
	Max. Packet Length	1500

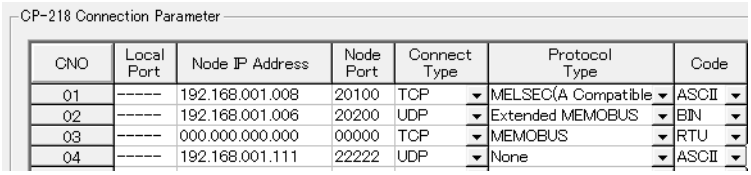
- The default value setting operation does not reset the connection parameters and the Ethernet address of the remote station to the default values.

(5) Setting the Ethernet Address of the Remote Station

Use the following procedure to set the Ethernet address (physical address) of the remote station if it does not support the ARP (Address Resolution Protocol).

- Contact your network administrator when setting a remote Ethernet address.

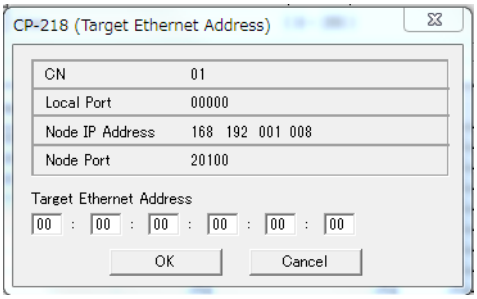
1. Move the cursor to the connection where the remote station's Ethernet address is set.



CNO	Local Port	Node IP Address	Node Port	Connect Type	Protocol Type	Code
01	-----	192.168.001.008	20100	TCP	MELSEC(A Compatible	ASCII
02	-----	192.168.001.006	20200	UDP	Extended MEMOBUS	BIN
03	-----	000.000.000.000	00000	TCP	MEMOBUS	RTU
04	-----	192.168.001.111	22222	UDP	None	ASCII

2. Select ***Edit - Ethernet Address Setting*** from the main window to display the **CP-218 (Target Ethernet Address)** Dialog Box.

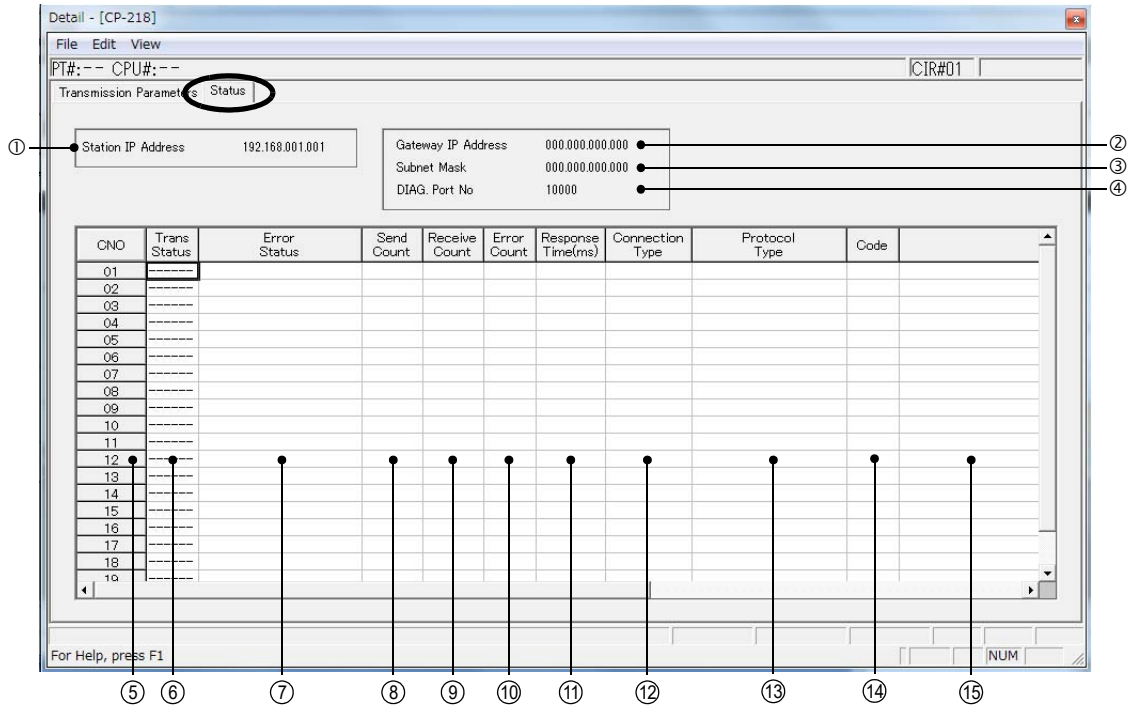
3. Set Ethernet address of the remote station in the dialog box and click the **OK** Button.



- If the node IP address and node port number are both all zeros, set the target Ethernet address to all zeros as well.
- If the same Ethernet address is set for more than one connection, set the same node IP address as well. This restriction does not apply if the Ethernet address is set to all zeros.

4.4.3 Status Tab Page

The **Status** Tab Page displays the 218IF-01 Module configuration information and transmission status. The settings cannot be changed.



- If the **Status** Tab is clicked without saving 218IF transmission definition data, a confirmation message to save data or an error message (if inconsistencies exist in the data) may be displayed. If a confirmation message to save data is displayed, click the **Yes** Button to save the setting data and the **Status** Tab Page will be displayed. If an error message is displayed, click the **OK** Button, enter correct parameters, and save the data. Then, click the **Status** Tab.
- The data on the **Status** Tab Page is displayed only in online mode. Nothing is displayed in offline mode.

■ Status Display Details

① Station IP Address

Displays the local station's IP address set on the **Transmission Parameters** Tab Page.

② Gateway IP Address

Displays the gateway (router) IP address set on the **Transmission Parameters** Tab Page.

③ Subnet Mask

Displays the subnet mask set on the **Transmission Parameters** Tab Page.

④ DIAG. Port No. (Diagnostic Port Number)

Displays the system port number (diagnostic/engineering port number) that was set on the **Transmission Parameters** Tab Page.

⑤ CNO (Connection Number)

The connection number (1 to 20) is displayed.

⑥ Trans Status (Transmission Status)

Displays the status of each connection as shown below.

Status Display	Status
IDLE	Idle status
WAIT	Wait status (waiting for connection)
CONNECT	Connected status (data send/receive enabled)
–	Unused connection

⑦ Error Status

The **Error Status** Column displays details on the error if an error has occurred in the transmission status. The error status displays and meanings are listed in the following table.

Status Display	Meaning	Description
No error	Normal	
Socket Creation Error	System error	Failure in socket generation
Local Port No Error	Local station port number setting error (Binds the same address during disconnection of the TCP connection.)	Binding error (port number duplication)
		When the ABORT function was executed using the MSG function, a binding error occurred during disconnection. This error occurs if Execute is turned ON within 1 minute after Abort is input and then Complete is automatically turned ON.
		Before the completion of connection processing, the command was sent to the same remote station by another function.
Changing Socket Attribute	System error (in TCP mode)	The error occurred while setting the socket attribute.
Connection Error (M-SND)	Connection error (Connection was rejected by the remote station when establishing a connection with an unpassive open in TCP mode.)	The command was reset since the remote station rejected the connection although the connection was attempted by MSG-SND function.
		Connection was disabled even at 1-minute retry (default) after the cable was disconnected.
Connection Error (M-RCV)	Connection error (Connection with a passive open in TCP mode)	MSG-RCV function communication acceptance error
System Error	System error	Socket polling (SELECT used) while receiving data.
TCP Data Send Error	Data sending error (The remote station does not exist or the remote station has not started when using TCP.)	A response transmission error occurred with the MSG-RCV function. The same error occurred with the MSG-SND function. This error occurs when the target remote station for sending the data does not exist, or the remote station is rebooted (only with a TCP connection).
UDP Data Send Error	Data sending error (When using UDP)	The data send request command was sent to a socket that does not exist.
TCP Data Receive Error	Data receiving error (The Module received a request to disconnect the connection from the remote station when using TCP.)	This error occurs when the connection is disconnected by the remote station. The error occurs even if close processing is performed normally.
Data Receiving Error (UDP)	Data receiving error (when using UDP)	The data receiving command was executed for a socket that does not exist.
Changing Socket Option Error	System error	An error occurred while changing the socket option.
Converting Data Error	Data conversion error	Protocol conversion error

- The last error displayed in the error status will be retained until the power is turned OFF and then back ON even if communication recovers. To clear the display, turn the power to the Controller OFF and then back ON.

⑧ Send Count (Transmission Pass Counter)

Displays the amount of data (number of packets) sent to the remote station.

⑨ Receive Count (Reception Pass Counter)

Displays the amount of data (number of packets) received from the remote station.

⑩ Error Count

Displays the number of times an error has occurred for each connection.

⑪ Response Time [ms]

Displays the time (ms) taken to receive a response for a command sent using the MSG-SND function.

⑫ Connection Type

Displays the Connect Type (TCP or UDP) set on the **Transmission Parameters** Tab Page.

⑬ Protocol Type

Displays the Protocol Type (MEMOBUS, Extended MEMOBUS, MELSEC, None (non-procedure), MODBUS, or TCP) set on the **Transmission Parameters** Tab Page.

⑭ Code

Displays the Code (ASCII, binary, or RTU) set on the **Transmission Parameters** Tab Page.

⑮ Node Name (Remote Station Name)

Displays the Node Station name set on the **Transmission Parameters** Tab Page.

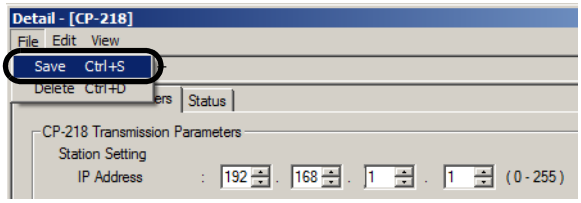
4.4.4 Saving and Deleting 218IF Transmission Definitions

- When changing, saving, or deleting 218IF transmission definitions data, save the data to flash memory in the Machine Controller, and then turn the power to the Machine Controller OFF and then turn it ON again.

(1) Saving the 218IF Transmission Definitions Data

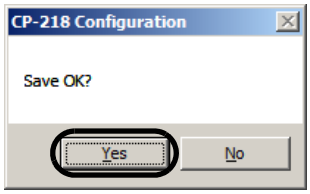
Use the following procedure to save the 218IF transmission definitions data if it has been set or changed. In Online Mode, the definitions data is saved in the Machine Controller and in personal computer running the MPE720. In Offline Mode, the definitions data is saved in the personal computer running the MPE720.

1. Select ***File - Save*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

2. Click the **Yes** Button.

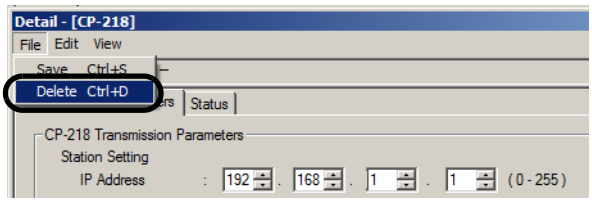


- An error message will be displayed if the input values for the parameters do not pass the consistency check. In this case, correct the cause of the error by referring to the following manual and try to save the data again. *MPE720 Software for Programming Device User's Manual (Manual No.: SIEP C880700 05)*.

(2) Deleting the 218IF Transmission Definitions Data

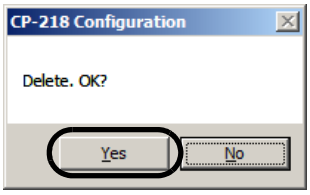
Use the following procedure to delete all of the 218IF transmission definitions data. In Online Mode, the definitions data is deleted from the MP2000 Series Machine Controller and the personal computer running the MPE720. In Offline Mode, the definitions data is deleted from the personal computer running the MPE720.

1. Select ***File - Delete*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

2. Click the **Yes** Button.



4.5 Ethernet Communication

4.5.1 Overview

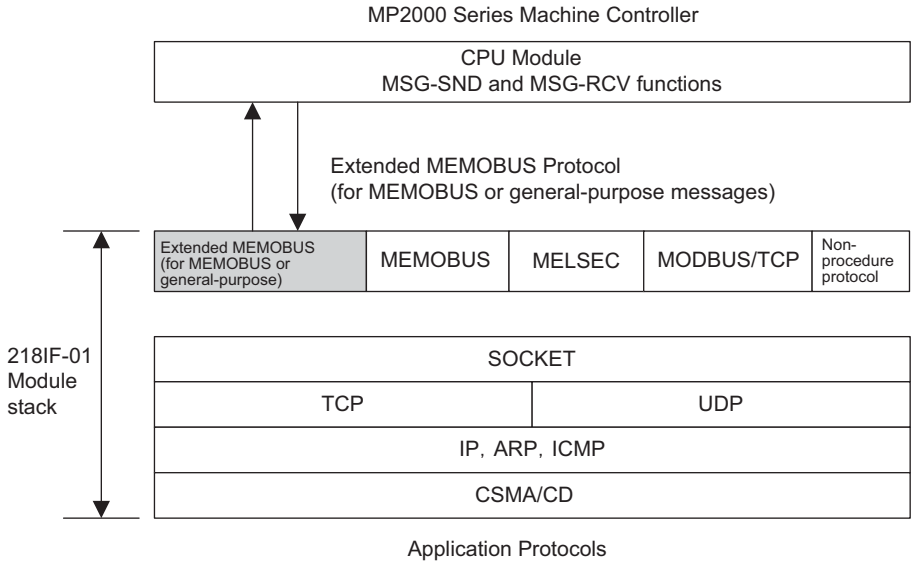
The 218IF-01 Module is used to connect the MP2000 Series Machine Controller to Ethernet devices. The 218IF-01 Module is equipped with one 10Base-T Ethernet port that conforms to IEEE802.3 and can be easily connected to a controller manufactured by other companies, a personal computer, or other types of computers. By connecting a computer running the MPE720, engineering of the MP2000 Series Machine Controller from the computer is enabled.
By installing a 218IF-01 Module, the MP2000 Series Machine Controller can send and receive data through the Ethernet line. Ethernet communication can be used for message communication and engineering communication. Transmissions using Ethernet communication are performed by specifying the MSG-SND and MSG-RCV functions in drawing or function programs.
This section describes the 218IF protocol for Ethernet communication, setting methods, and connection methods, and provides programming examples.

4.5.2 Ethernet Communication Protocol

The 218IF-01 Module supports the five types of protocol listed below in the TCP/IP application layer.

Protocol	Remarks	Master/Slave
MEMOBUS	Yaskawa's standard MEMOBUS protocol	Master/Slave
Extended MEMOBUS	Yaskawa's extended MEMOBUS protocol.	Master/Slave
MELSEC	A subset of the sub-protocol for Mitsubishi Electric's General-purpose Programmable Logic Controllers	Master/Slave
MODBUS/TCP	Ethernet protocol, proposed by Modicon, used in industrial applications.	Master/Slave
Non-procedure protocol	A protocol provided to implement general-purpose message communication	Master/Slave

The Extended MEMOBUS protocol is used for all message communication between an MP2000 Series Machine Controller CPU Module and the 218IF-01 Module.
The following diagram illustrates this.



This section describes the communication protocols that can be used with the 218IF.

4.5.3 MEMOBUS Protocol

(1) Overview

MEMOBUS transmissions use messages controlled by the MEMOBUS protocol. Data transmissions are performed in Master-Slave relationships in which the Master sends a message to a Slave and the Slave returns a response to the Master.
The Master communicates with one Slave at a time. To enable this, Slaves are assigned unique address numbers. The Master specifies the address number of a specific Slave when sending data.
On receiving a message from the Master, the Slave performs the specified function and returns a response to the Master.
Using these functions, the Master can read and modify the status of coils and contents of the registers in Slaves located at different places, enabling flexible centralized monitoring and control in a variety of production fields.

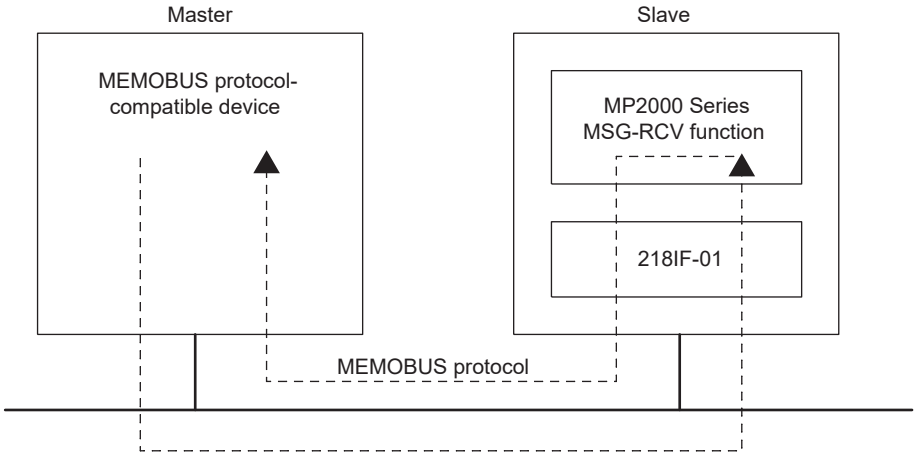


Fig. 4.1 Message Flow Using the MEMOBUS Protocol

(2) Function Codes in the MEMOBUS Protocol

The following table lists the function codes used in the MEMOBUS protocol. These functions are implemented by setting the function code in the MSG-SND function.

Function Code	Function	Data Address Setting Range	Data Size
01H	Read Coil Status	0 to 65535	1 to 2000 bits
02H	Read Input Relay Status	0 to 65535	1 to 2000 bits
03H	Read Holding Register Contents	0 to 65534	1 to 125 words
04H	Read Input Register Contents	0 to 32767	1 to 125 words
05H	Modify Status of a Single Coil	0 to 65535	–
06H	Write to a Single Holding Register	0 to 65534	–
08H	Loopback Test	–	–
0FH	Modify Status of Multiple Coils	0 to 65535	1 to 800 bits
10H	Write to Multiple Holding Registers	0 to 65534	1 to 100 words

4.5.4 Extended MEMOBUS Protocol

(1) Overview

The Extended MEMOBUS protocol is an enhanced version of the MEMOBUS protocol. It provides more versatile function codes and is capable of handling a larger data size. Data transmissions are performed in Master-Slave relationships in which the Master sends a message to a Slave and the Slave returns a response to the Master.
The Extended MEMOBUS protocol supports the MEMOBUS message communication mode and a general-purpose message communication mode. The mode that is used is specified in a parameter in the MSG-SND and MSG-RCV functions.

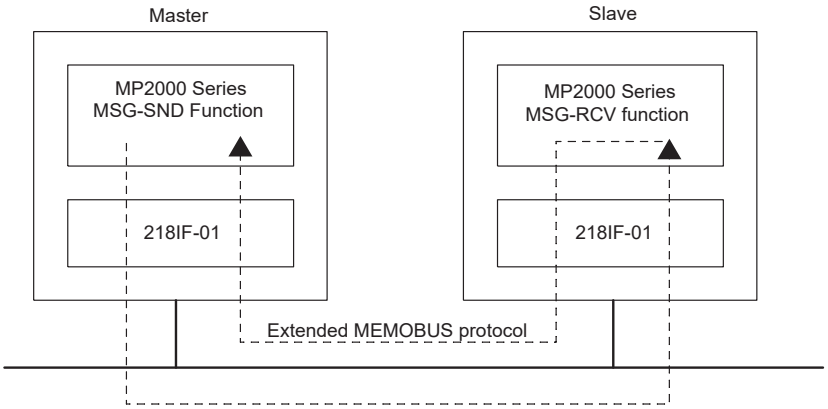


Fig. 4.2 Message Flow Using the Extended MEMOBUS Protocol

(2) Function Codes in the Extended MEMOBUS Protocol

The following table shows the function codes that can be used with the Extended MEMOBUS protocol. These functions are implemented by setting the function code in the MSG-SND function.

Function Code	Function	Data Address Setting Range	Data Size
01H	Read Coil Status	0 to 65535	1 to 2000 bits
02H	Read Input Relay Status	0 to 65535	1 to 2000 bits
03H	Read Holding Register Contents	0 to 65534	1 to 125 words
04H	Read Input Register Contents	0 to 32767	1 to 125 words
05H	Modify Status of a Single Coil	0 to 65535	–
06H	Write to a Single Holding Register	0 to 65534	–
08H	Loopback Test	–	–
09H	Read Holding Register Contents (Extended)	0 to 65534	1 to 508 words
0AH	Read Input Register Contents (Extended)	0 to 32767	1 to 508 words
0BH	Write to Holding Register (Extended)	0 to 65534	1 to 507 words
0DH	Read Discontinuous Holding Register Contents (Extended)	0 to 65534	1 to 508 words
0EH	Write to Discontinuous Holding Registers (Extended)	0 to 65534	1 to 254 words
0FH	Modify Status of Multiple Coils	0 to 65535	1 to 800 bits
10H	Write to Multiple Holding Registers	0 to 65534	1 to 100 words

4.5.5 MELSEC Protocol

(1) Overview

The MELSEC protocol is used for communication between an MP2000 Series Machine Controller and a MELSEC A-series Programmable Logic Controller manufactured by Mitsubishi Electric Corporation.
The MELSEC protocol is used to communicate between the MP2000 Series Machine Controller and a MELSEC Programmable Logic Controller. To achieve this, the 218IF-01 Module automatically converts the protocol to the MELSEC protocol. The MSG-SND and MSG-RCV functions can be used in a user program in the same way as with the MEMOBUS protocol.
The 218IF supports the following three functions of the MELSEC protocol.

[a] Communication between CPUs

The MSG-SND function can be used to read and write data in the CPU of the MELSEC Programmable Logic Controllers. The A-compatible 1E frame format is used and a subset of the A-compatible 1E commands is supported. In communication between CPUs, the MP2000 Series Machine Controller can only be a Master due to restrictions of the specifications of MELSEC A-series Programmable Logic Controllers.

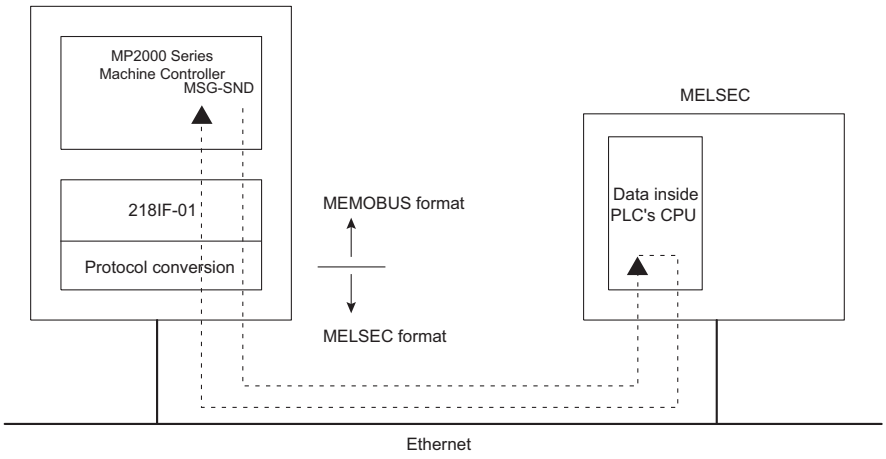


Fig. 4.3 Message Flow for Communication between CPUs

[b] Fixed Buffer Communication

The MSG-SND function can be used to write the contents of M registers to the fixed buffers in the MELSEC Ethernet Module. The MSG-RCV function can be used to receive data sent to the MP2000 Series Machine Controller from the fixed buffers and store it in M registers.

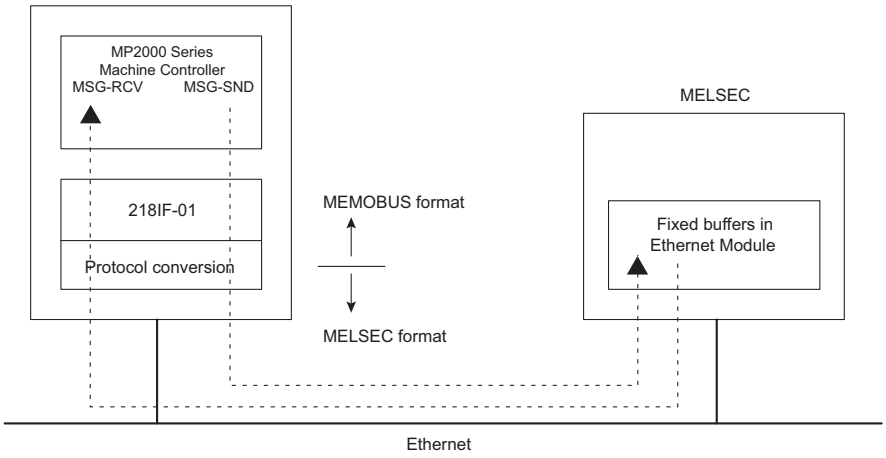


Fig. 4.4 Message Flow for Fixed Buffer Communication

[c] Random-access Buffer Communication

The MSG-SND function can be used to read and write the random-access buffers in the MELSEC Ethernet Module. The contents of M registers can be written to the random-access buffers, and the contents read from the random-access buffers can be stored in M registers.
In random-access buffer communication, the MP2000 Series Machine Controller can only be a Master due to restrictions of the specifications of MELSEC A-series Programmable Logic Controllers.

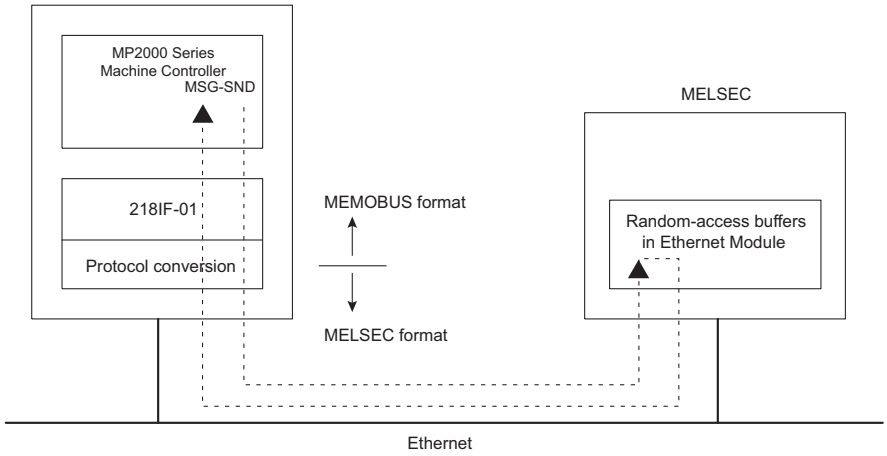


Fig. 4.5 Message Flow for Random-access Buffer Communication

(2) Command Function Code List

The following table lists the common MELSEC ACPU commands and MEMOBUS function codes supported by the 218IF. To communicate with a MELSEC A-series Programmable Logic Controller, set the MEMOBUS function code that corresponds to the MELSEC ACPU command in the MSG-SND function.

Common MELSEC ACPU Command	MEMOBUS Function Code	Function	Number of Words
00H	01H/02H	Reads data from bit-type devices in 1-point units.	256 points
01H	03H/04H/09H/0AH	Reads data from word-type devices in 1-point units.	256 points
02H	05H/0FH	Writes data to bit-type devices in 1-point units.	256 points
03H	06H/0BH/10H	Writes data to word-type devices in 1-point units.	256 points
16H	08H	Executes a loopback test.	256 bytes
05H	0EH	Sets/resets data in word devices specified at random in 1-point units.	40 points
60H	31H	Fixed buffer communication	507 words
61H	32H	Random-access buffer read	508 words
62H	33H	Random-access buffer write	507 words

- The special commands for the AnCPU are not supported. Use the common ACPU commands listed above when accessing an AnCPU. Accessing the extended file registers in an AnCPU is not possible.

(3) MELSEC Devices

The following tables list the common ACPU command device ranges, MEMOBUS commands, and MP2000 Series Controller register numbers for MELSEC bit-type and word devices.
Use these tables as a basis for setting the parameters for the MSG-SND function when reading and writing MELSEC devices from an MP2000 Series Controller.

■ MELSEC Bit Devices

Device	Common ACPU Command Device Range	Decimal/ Hex	MEMOBUS Commands	Start No.	Register No.
X	X0000 to X07FF	Hex	02H: Input relays	0 to 2047	MB000000 to MB00127F
Y	Y0000 to Y07FF	Hex	01H/0FH: Coils	0 to 2047	MB000000 to MB00127F
M	M0000 to M2047	Decimal	01H/05H/0FH: Coils	2048 to 4095	MB001280 to MB00255F
M	M9000 to M9255	Decimal	01H/05H/0FH: Coils	4096 to 4351	MB002560 to MB002715F
B	B0000 to B03FF	Hex	01H/05H/0FH: Coils	4352 to 5375	MB002720 to MB00335F
F	F0000 to F0255	Decimal	01H/05H/0FH: Coils	5376 to 5631	MB003360 to MB00351F
TS	TS000 to TS255	Decimal	02H: Input relays	2048 to 2303	MB001280 to MB00143F
TC	TC000 to TC255	Decimal	02H: Input relays	2304 to 2559	MB001440 to MB00159F
CS	CS000 to CS255	Decimal	02H: Input relays	2560 to 2815	MB001660 to MB00175F
CC	CC000 to CC255	Decimal	02H: Input relays	2816 to 3071	MB001760 to MB00191F

- Even for the same device ranges, the device area ranges depend on the MELSEC Sequencer. Refer to the MELSEC manuals for details.

■ MELSEC Word Devices

Device	Common ACPU Command Device Range	Decimal/ Hex	MEMOBUS Commands	Start No.	Register No.
TN	TN0000 to TN255	Decimal	04H/0AH: Input registers	0 to 255	MW00000 to MW00255
CN	CN0000 to CN255	Decimal	04H/0AH: Input registers	256 to 511	MW00256 to MW00511
D	D0000 to D1023	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	0 to 1023	MW00000 to MW01023
D (Special)	D9000 to D9255	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	1024 to 1279	MW01024 to MW01279
W	W0000 to W03FF	Hex	03H/06H/09H/0BH/0EH/ 10H: Holding registers	1280 to 2303	MW01280 to MW02303
R	R0000 to R8191	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	2304 to 10495	MW02304 to MW10495

- Even for the same device ranges, the device area ranges depend on the MELSEC Sequencer. Refer to the MELSEC manuals for details.

The MP2000 Series Machine Controller register addresses can be adjusted by using the offset setting in the MSG-SND function, as shown in the following example.

Example: Reading 32 Bits from MELSEC Relays (from M0016)

From the table for MELSEC bit devices, the register number for device M0016 is MB001290, the starting number is 2064, and the MEMOBUS command to read the data is 01H.
These values are set in the parameters of the MSG-SND function.

- PARAM04: Function code = 01
- PARAM05: Data address = 2064
- PARAM06: Data size = 32 (bits)

The data address in PARAM05 serves as both the address to read and the storage address. The above settings thus read 32 bits of data starting from M0016 and write it to the 32 bits starting from MB001290.
To store the register at a location other than MB001290, the coil offset in PARAM08 is used to adjust the storage location.
Example:

- PARAM08: Coil offset = 1000

The above setting stores the 32 bits of data read starting at M0016 to the 32 bits starting at MB011290 (i.e., MB001290 + 1000). Negative offsets cannot be used.

- Refer to *5.7 Message Send Function* on page 5-48 for details on message communication functions.

(4) MP2000 Series Machine Controller/MELSEC Register Map

The following register map is based on the device table in *(3) MELSEC Devices* on page 4-28 and shows the MP2000 Series Machine Controller M registers that correspond to each MELSEC device. Holding registers, input registers, input relays, and coils are assigned for each MELSEC device so that the MP2000 Series Machine Controller can use MEMOBUS commands as an interface to read and write data. Data read from a MELSEC device is stored in the corresponding M registers shown in the following map. Data written to the MELSEC devices is written from the corresponding device M registers shown in the following table.

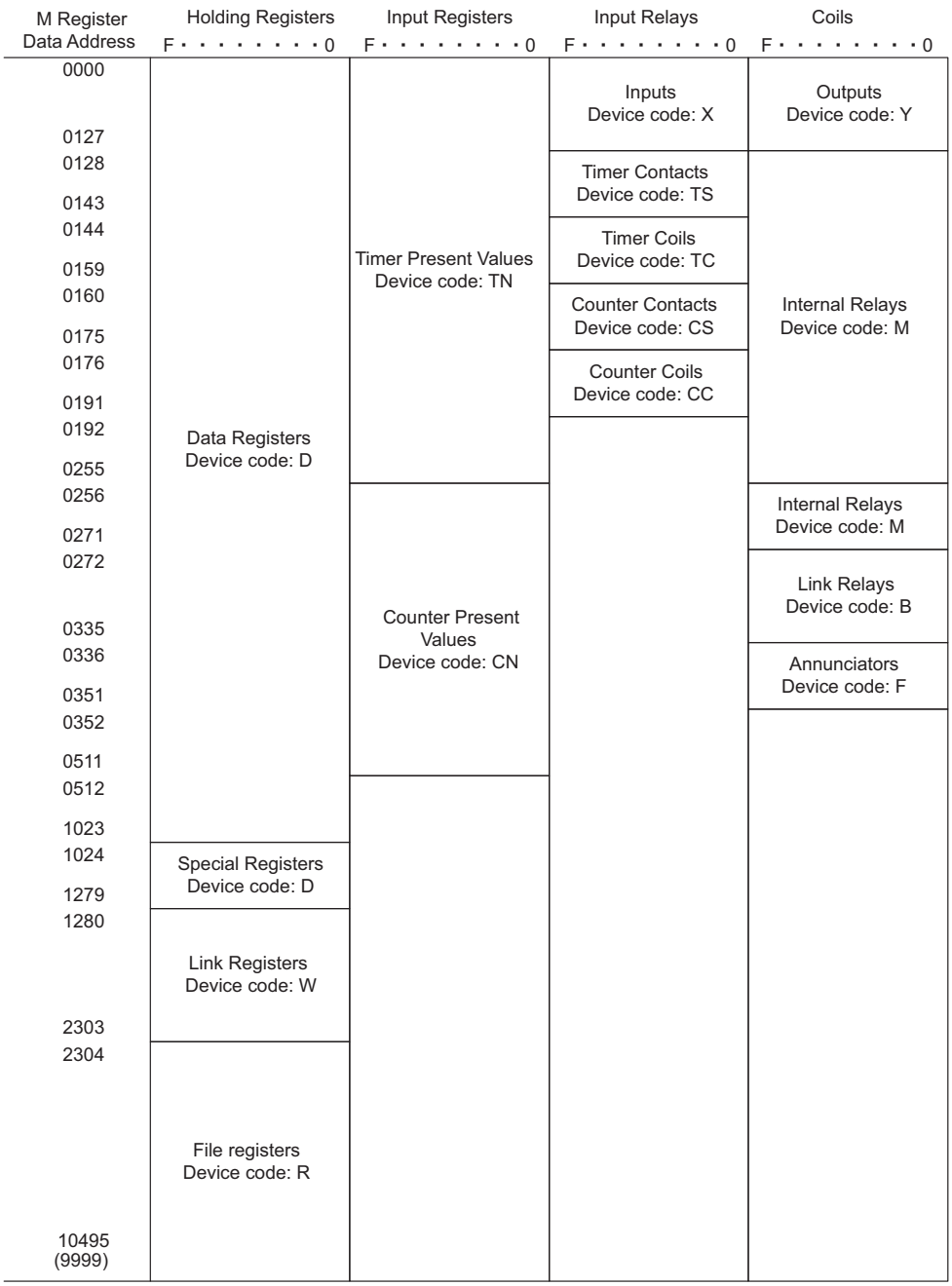


Fig. 4.6 Register Map

- This map is for when the offset parameters in the MSG-SND function are set to 0. An offset parameter (i.e., holding register offset, input register offset, input relay offset, or coil offset) can be set to shift the M register numbers given in this map backward by a relative amount.

4.5.6 MODBUS/TCP Protocol

(1) Overview

The MODBUS/TCP protocol enables communication with devices of other manufacturers that support the MODBUS/TCP protocol.
In communication between an MP2000 Series Machine Controller and a device that supports the MODBUS/TCP, the 218IF-01 Module automatically converts the protocol to the MODBUS/TCP protocol. The MSG-SND and MSG-RCV functions can be used in a user program in the same way as with the MEMOBUS protocol.

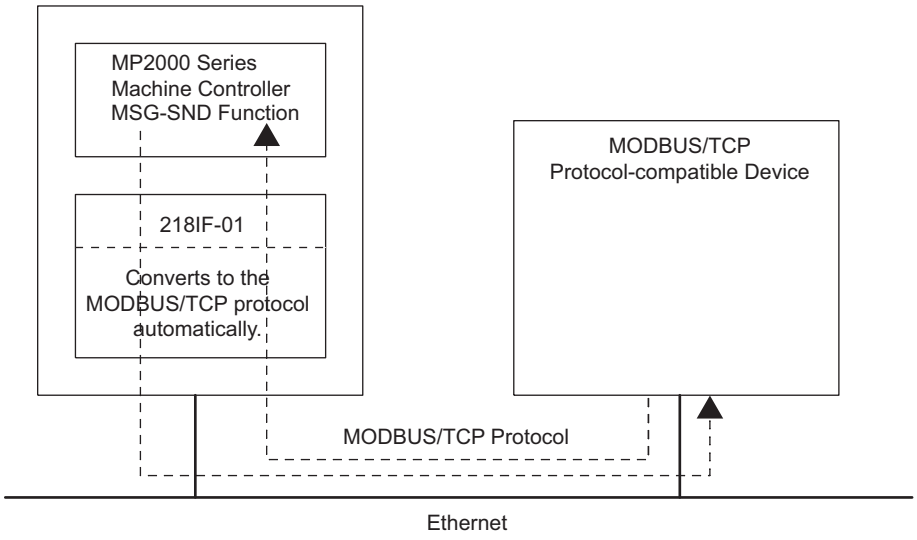


Fig. 4.7 Message Flow between MP2000 Series Machine Controller and MODBUS/TCP Protocol-compatible Device Using MODBUS/TCP Protocol

(2) Function Codes

The following table shows the function codes used in the MODBUS/TCP protocol. These functions are implemented by setting the function code in the MSG-SND function.

Function Code	Function	Data Size
01H	Read Coil Status	1 to 2000 bits
02H	Read Input Relay Status	1 to 2000 bits
03H	Read Holding Register Contents	1 to 125 words
04H	Read Input Register Contents	1 to 125 words
05H	Modify Status of a Single Coil	1
06H	Write to a Single Holding Register	1 words
0FH	Modify Status of Multiple Coils	1 to 800 bits
10H	Write to Multiple Holding Registers	1 to 100 words

4.5.7 Non-Procedure Protocol

(1) Overview

The non-procedure protocol enables message communication with devices from other manufacturers using any protocol. Data in consecutive M registers in the MP2000 Series Machine Controller is sent and received without modification.
The MSG-SND function sends the data from the specified M registers without modification. There is no response.
The MSG-RCV function stores the received data without modification in the M registers specified in the MSG-RCV function. It does not return a response.
Thus, the non-procedure protocol sends data in just one direction without a response.
The non-procedure protocol is not based on command-response communication. It is thus necessary to make a user program in which sending and receiving are switched to perform command-response communication.
The size of data that can be sent or received in non-procedure communication mode is from 1 to 510 words.

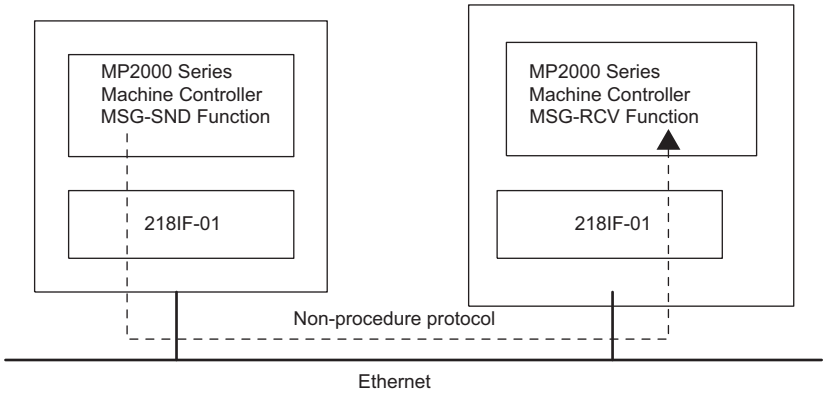
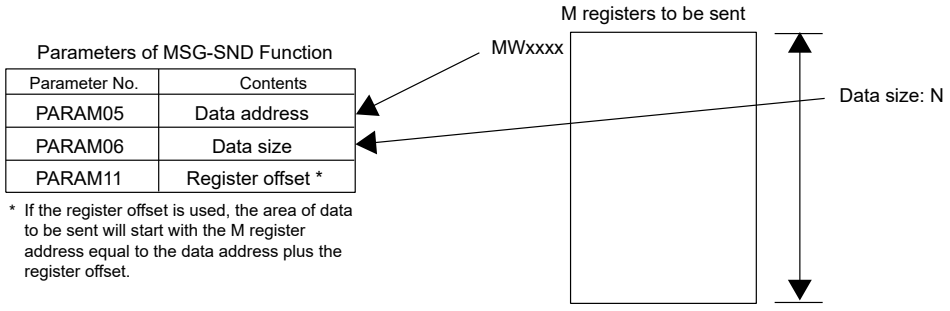
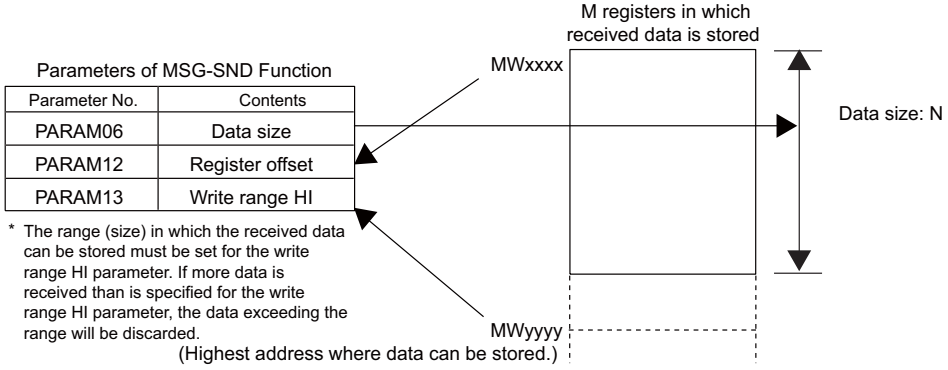


Fig. 4.8 Message Flow Using the Non-procedure Protocol

With the non-procedure protocol, the address of the first M register and the data size are specified in the MSG-SND function to enable sending the contents of M registers.



With the MSG-RCV function, the register offset and write range HI can be set to store the received data in the specified M registers.



- For both the MSG-SND and MSG-RCV functions, set the connection number to use for communication in PARAM02.

■ Precautions on Using TCP with Non-procedure Communication

- To avoid complicated byte stream processing when using TCP with non-procedure communication, use one of the following methods so that the remote device does not consecutively send TCP packets.
  - Increase the packet send interval (recommendation: 1 s or longer).
  - Create a command-response application protocol.
- Set the system so that the remote device does not separate send data into packets. Design the system so that the remote device does not send data separated into packets.
  - TCP is a byte-stream protocol. If data is sent continuously, the number of data blocks sent in one packet will depend on the send timing because the data blocks that are sent may be combined at the sending device. Also, the number of data blocks that is received will depend on the receive timing because the data blocks sent in multiple packets may be combined at the receive device. Combining data blocks at the sending device and receive device is performed in the transport layer* (TCP) and network layer* (IP) of the protocol stack of the 218IF-01 Module. Therefore, it is not possible to determine how many items of data will be sent using non-procedure communication without a field that indicates the send data length.

■ Precautions on Using TCP and UDP with Non-procedure Communication

If data is continuously received using the same connection, set a data send interval for remote devices that is longer than the cycle for which the MSG-RCV function can read data into the user application.

- If the 218IF-01 Module receives the next data before reading the data into the user application using the MSG-RCV function, the old data will be overwritten with the most recent data. This is because there is only one reception buffer per connection in the protocol stack application layer* (non-procedure) in the 218IF-01 Module.
- For the MSG-RCV function, at least two scans are required after data reception has been completed until the next time data reception starts. Asynchronous processing is performed between the 218IF-01 Module and the MSG-RCV function, and so at least two scans are required if the scan cycle is shorter than the operation cycle of the 218IF-01 Module. As a rough guide, set a data send interval of at least two scans of the execution scan cycle of the MSG-RCV function at the sending device if the execution scan cycle of the MSG-RCV function is 6 ms or longer. Set a data send interval of 12 ms or longer at the sending device if the execution scan cycle of the MSG-RCV function is 5 ms or shorter. The operation cycle of the 218IF-01 Module depends on the communication load of the Module. If the communication load is large, adjust the data send interval to an appropriate time.

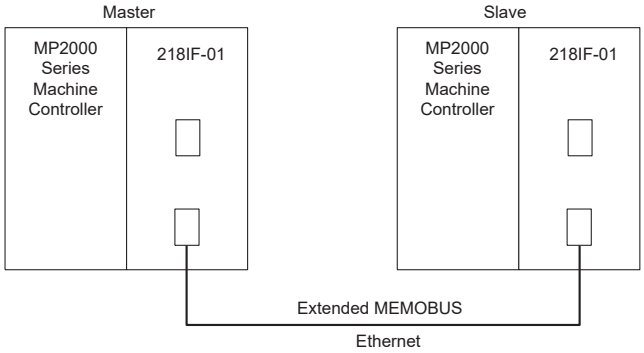
* Seven Layers for OSI Reference Model

4.6 Connection Examples for Ethernet Communication

This section shows examples of connecting to the Ethernet port of the 218IF-01.

4.6.1 Connecting 218IF-01 Modules

The 218IF-01 Modules are mounted on MP2000 Series Machine Controllers. The MP2000 Series Machine Controller that operates as the Master reads data from the other MP2000 Series Machine Controller that operates as a Slave using Ethernet. The setup is shown below.

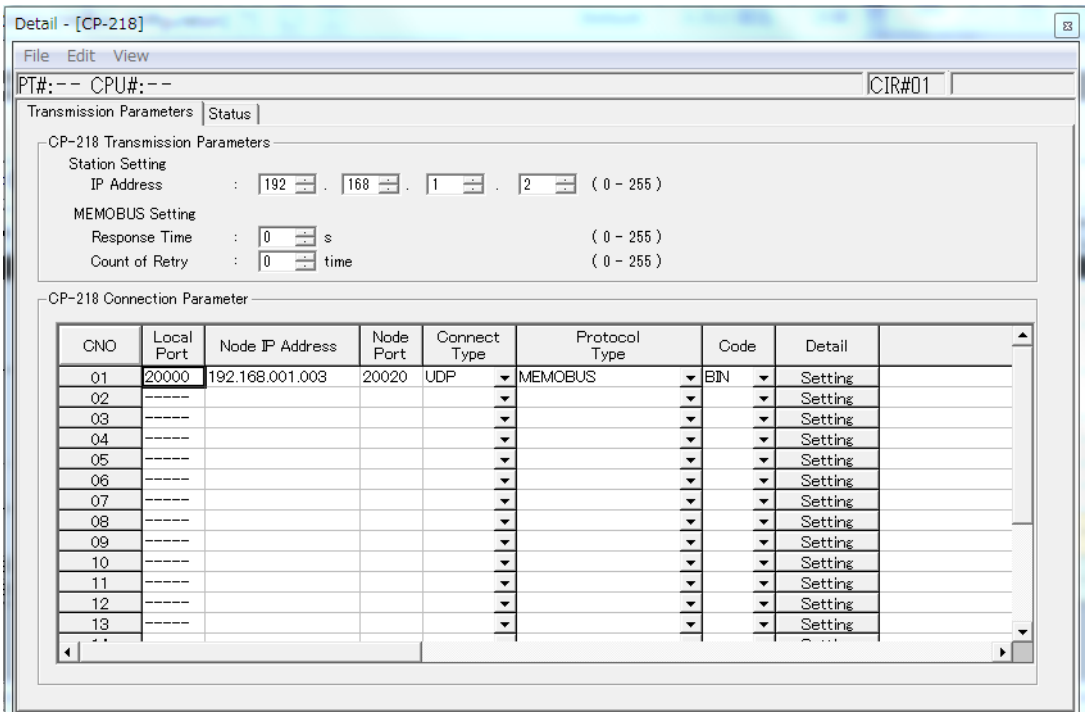


The IP address settings of the Master and Slave are shown below.

Item	Master	Slave
IP Address	192.168.001.002	192.168.001.003
Subnet Mask	255.255.255.000	255.255.255.000
Gateway IP Address	000.000.000.000	000.000.000.000

(1) 218IF Transmission Settings for the Master

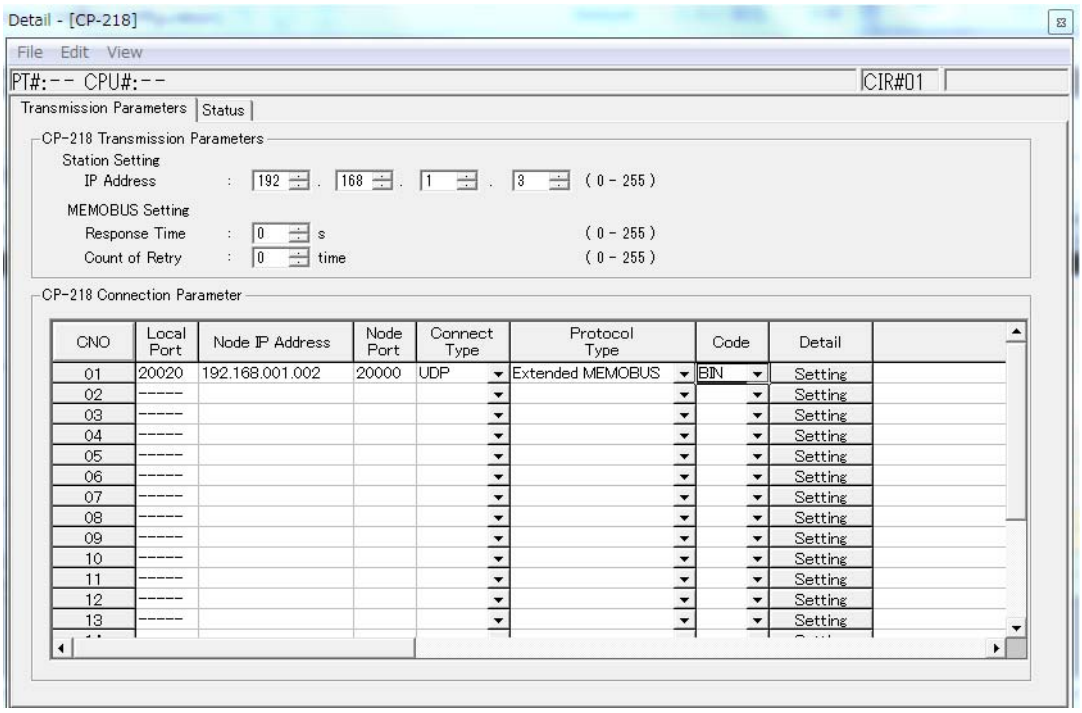
The 218IF transmission settings for the Master are shown below.



Local Port: Set the port number of the MP2000 Series Machine Controller operating as the Master.
Node IP Address: Set the IP address of the MP2000 Series Machine Controller operating as a Slave.
Node Port: Set the port number of the MP2000 Series Machine Controller operating as a Slave.

(2) 218IF Transmission Settings for the Slave

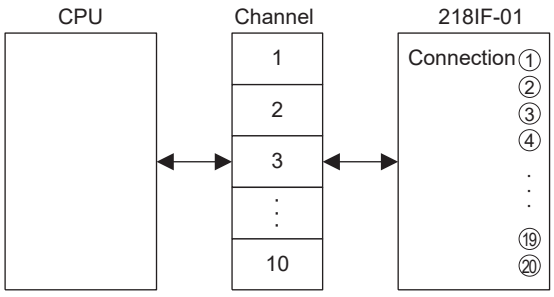
The 218IF transmission settings for the Slave are shown below.



Local Port: Set the port number of the MP2000 Series Machine Controller operating as a Slave.
Node IP Address: Set the IP address of the MP2000 Series Machine Controller operating as the Master.
Node Port: Set the port number of the MP2000 Series Machine Controller operating as the Master.

(3) Programming Precaution When Communicating through More Than 10 Connections

Although the maximum number of connections that can be set for a 218IF transmission system is 20, simultaneous communication is allowed for only up to 10 connections.
This is because only 10 channels are provided for communication between the CPU and a 218IF-01 Module and these channels are occupied until the MSG-SND or MSG-RCV function has been completed. When the 218IF transmission system is used with 11 stations or more, one channel must be used by two stations and the communication designation must be switched using the ladder program so that the 218IF transmission system communicates with the second station only after it has finished communicating with the first station. In programming cases like this, the execute input for the MSG-SND and MSG-RCV functions must not be always ON.



(4) Procedure to Start Communication

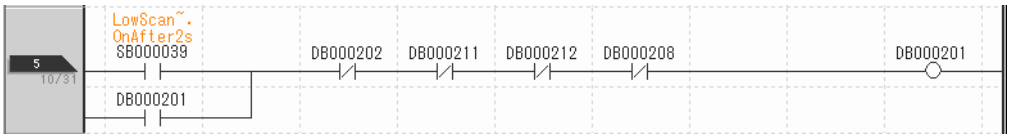
Use the following procedure to start communication.

1. Start executing the MSG-RCV function in the Slave 218IF Module.
   - Always execute the MSG-RCV function first.
2. Turn ON the **Execute** parameter in the MSG-SND function in the Master 218IF Module to send a message.

Example ladder programming using the MSG-SND and MSG-RCV functions is shown below in item (5).
In *[b] Ladder Programming Example Using the Message Receive Function (MSG-RCV)* on page 4-38 MSG-RCV function execution is started with SB000004 (Always ON Coil). Execution of the MSG-RCV function will thus start as soon as the power is turned ON to the Slave MP2000 Series Machine Controller. In *[a] Ladder Programming Example Using the Message Send Function (MSG-SND)* on page 4-37, a message will be sent as soon as DB000201 (the register set for **Execute** in the MSG-SND function) turns ON, thus starting communication with the Slave 218IF Module.

I/O Definitions	No.	Name	Setting Example	Description
Inputs	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

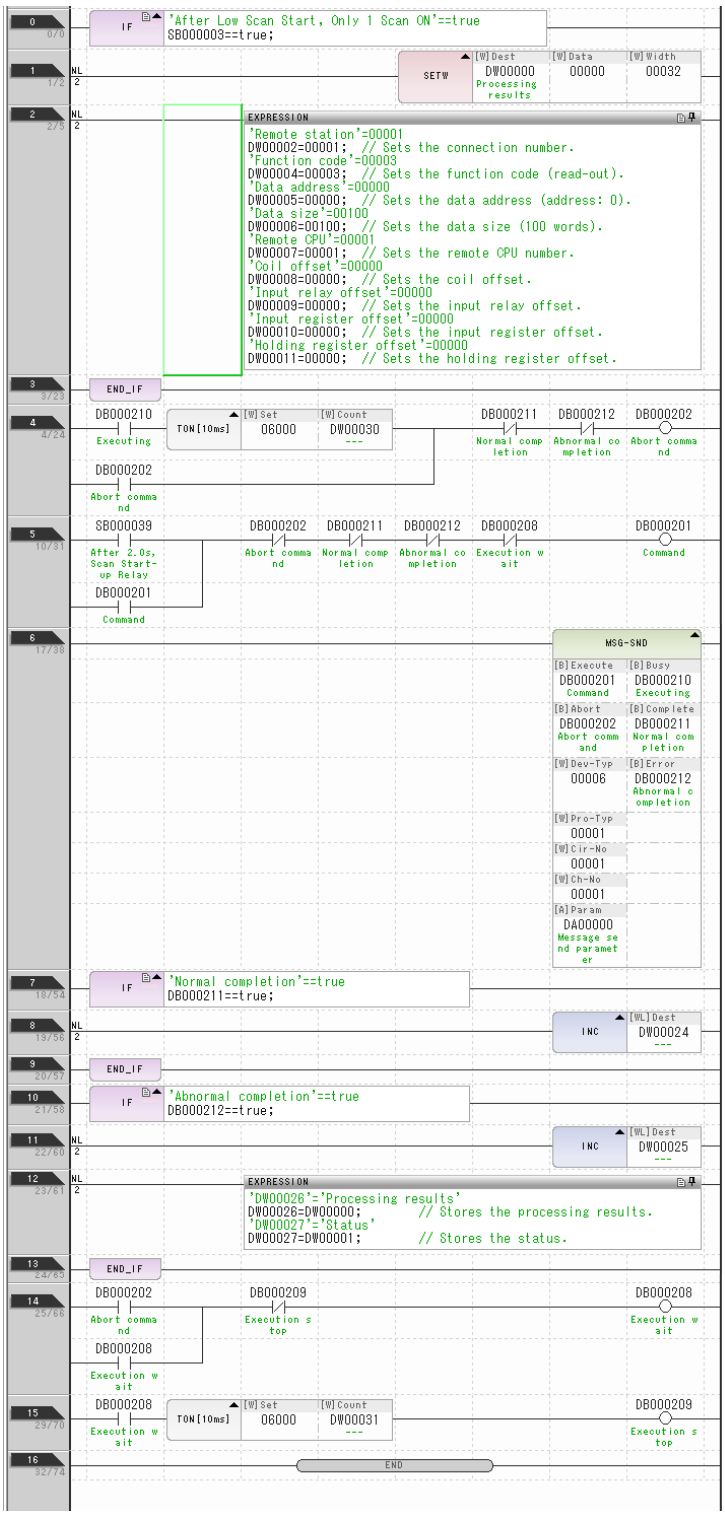
- In *[a] Ladder Programming Example Using the Message Send Function (MSG-SND)* on page 4-37, the programming is written so that a message will be sent every second after two seconds has expired after the low-speed (or high-speed) scan has started. When the MSG-SND function has been completed normally, one scan will be skipped and then execution of the MSG-SND function will be started again, repeatedly sending a message each time sending the previous message has been completed normally.



(5) Programming Example

An example of ladder programming using the Message Send and Message Receive function for 218IF transmissions is shown below.

[a] Ladder Programming Example Using the Message Send Function (MSG-SND)



Sets parameters in the first scan after startup.
(Low-speed scan (DWG.L): SB000003,
High-speed scan (DWG.H): SB000001)

Clears DW register to zero.

Sets PARAM02 to PARAM11.

Forced stop if processing is not completed within 3 seconds after sending the Send command.

Sends the Abort command.

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 1
Cir-No. (circuit number): 2 (connection port)
Ch-No. (channel number): 1 (always 1 for the 218IF-01)

Normal completion

Increments the normal pass counter.

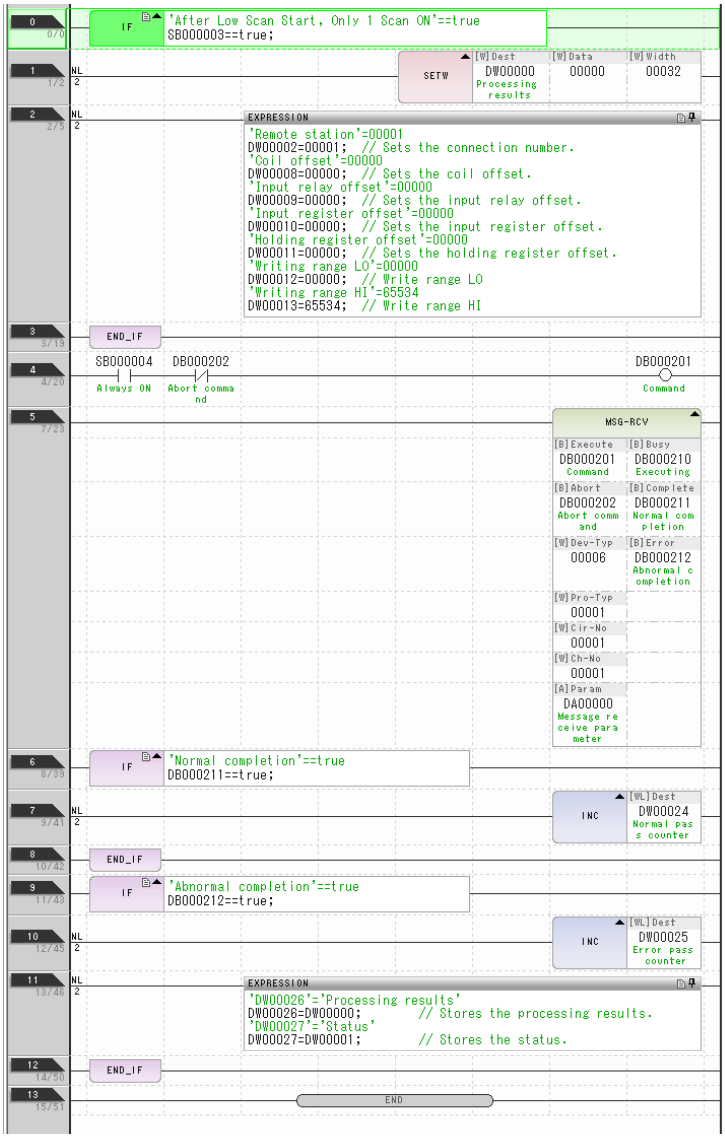
Ended with transmission errors.

Increments the abnormal pass counter.

Stores the processing results and the status.

Function Send command

[b] Ladder Programming Example Using the Message Receive Function (MSG-RCV)



Sets parameters in the first scan after startup. (Low-speed scan (DWG.L): SB000003, High-speed scan (DWG.H): SB000001)

Clears DW register to zero.

Sets PARAM07 to PARAM13.

Starts receiving.

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 1 for extended MEMOBUS or MELSEC, 2 for non-procedure
Cir-No. (circuit number): 2 (connection port)
Ch-No. (channel number): 1 (always 1 for the 218IF-01)

Turns ON after the command message is sent and the response message is received.

Increments the normal pass counter.

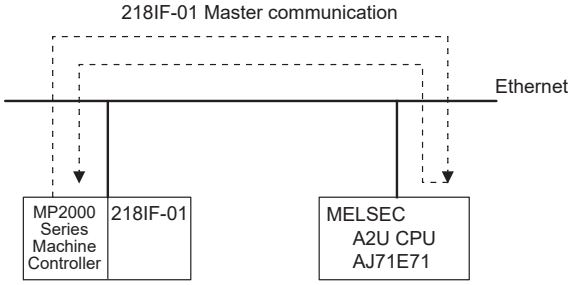
Ended with transmission errors.

Increments the abnormal pass counter.

Stores the processing results and the status.

4.6.2 Connection to MELSEC

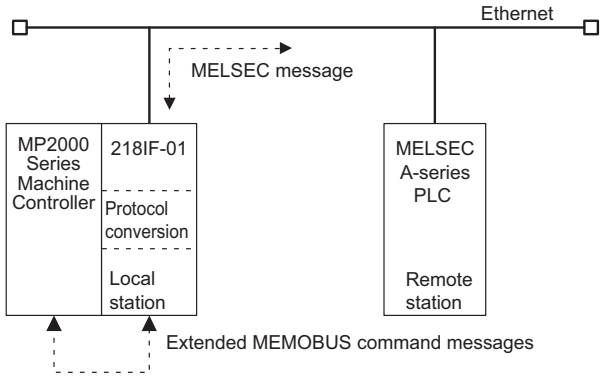
This section describes the connection example shown below in which the MP2000 Series Machine Controller is connected to a MELSEC A-series Programmable Logic Controller through Ethernet using a 218IF-01 Module.



- The 218IF-01 Module is set as the Master and the MELSEC as a Slave.

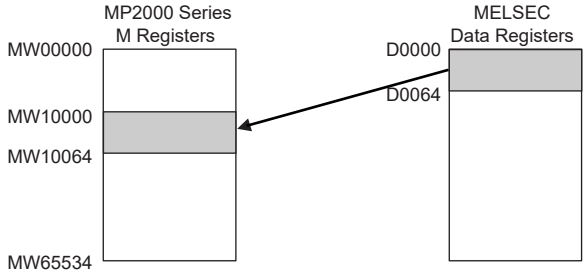
(1) Message Flow

In the above example, messages are exchanged between the MP2000 Series Machine Controller and 218IF-01 Module using Extended MEMOBUS communication and between the 218IF-01 and the MELSEC-A PLC using MELSEC communication.



- MELSEC communication is checked with a combination of the A2U CPU and AJ71E71.

In this example, the MELSEC protocol is used for communication between CPUs. The 218IF-01 on the MP2000 Series Machine Controller at the Master reads 64 words starting from data register D0000 at the MELSEC Slave and stores the data in starting from MW10000.

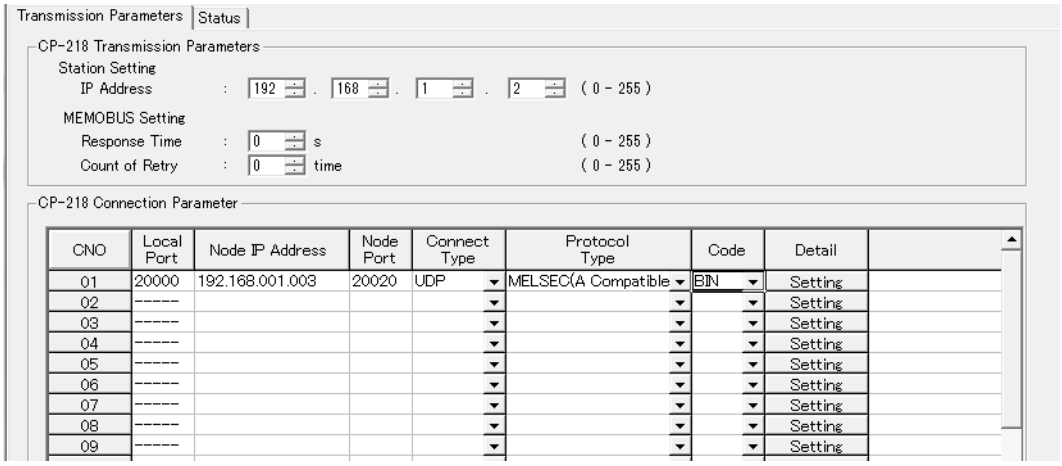


The following settings are used.

Item	MP2000 Series Machine Controller (Master)	MELSEC (Slave)
IP Address	192.168.001.002	192.168.001.003
Port Number	20000	20020
Subnet Mask	255.255.255.000	255.255.255.000
Gateway IP Address	000.000.000.000	000.000.000.000
Communication Protocol	UDP	UDP
Application Protocol	MELSEC	–

(2) 218IF Transmission Settings for the Master

The 218IF transmission settings for the Master are shown below.



Local Port: Set the port number of the MP2000 Series Machine Controller operating as the Master.
Node IP Address: Set the IP address of the MELSEC operating as a Slave.
Node Port: Set the port number of the MELSEC operating as a Slave.

(3) MELSEC Slave Settings

When using a MELSEC A-series Ethernet Module (AJ71E71), ladder programming to initialize the IP address, port number, and other settings must be created. To create the initializing ladder program, refer to the procedure for communicating with other nodes explained in the MELSEC manual.
When using a MELSEC Q-series Ethernet Module (QJ71E71-100), set the IP address and port number in the Ethernet setting window.

(4) Programming in the MP2000 Series Machine Controller

The MP2000 Series Machine Controller performs communication with a MELSEC A-series Programmable Logic Controller by converting messages from the Extended MEMOBUS protocol to the MELSEC protocol. Accordingly, it is not necessary to be aware of the MELSEC message structure in user application programs. All the user must do is to set MELSEC as the protocol in the connection parameters for the 218IF transmission system and enter the MSG-SND function in the program to easily communicate with the MELSEC Controller.
The commands used in the MSG-SND function are the MEMOBUS commands that correspond to MELSEC commands. Refer to *4.5.5 (2) Command Function Code List* on page 4-28 for the MEMOBUS commands.
For example, the MEMOBUS command that corresponds to the MELSEC command 00 (to read bit devices in 1-point units) is 01 or 02.

(5) Procedure to Start Communication

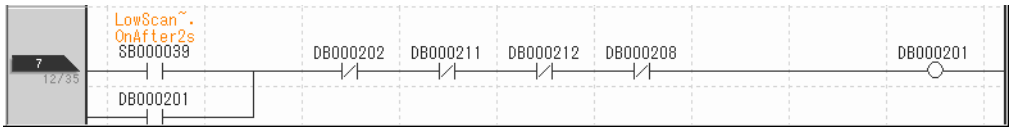
Use the following procedure to start communication.

1. Start the MELSEC Slave.
2. Turn ON the **Execute** parameter in the MSG-SND function in the Master 218IF Module to send a message.

Example ladder programming using the MSG-SND function is shown in *(6) Programming Example* on page 4-42. In this programming example, a message is sent when the register set for **Execute** in the MSG-SND function (DB000201) turns ON to start communication with the MELSEC Slave.

I/O Definitions	No.	Name	Setting Example	Description
Inputs	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

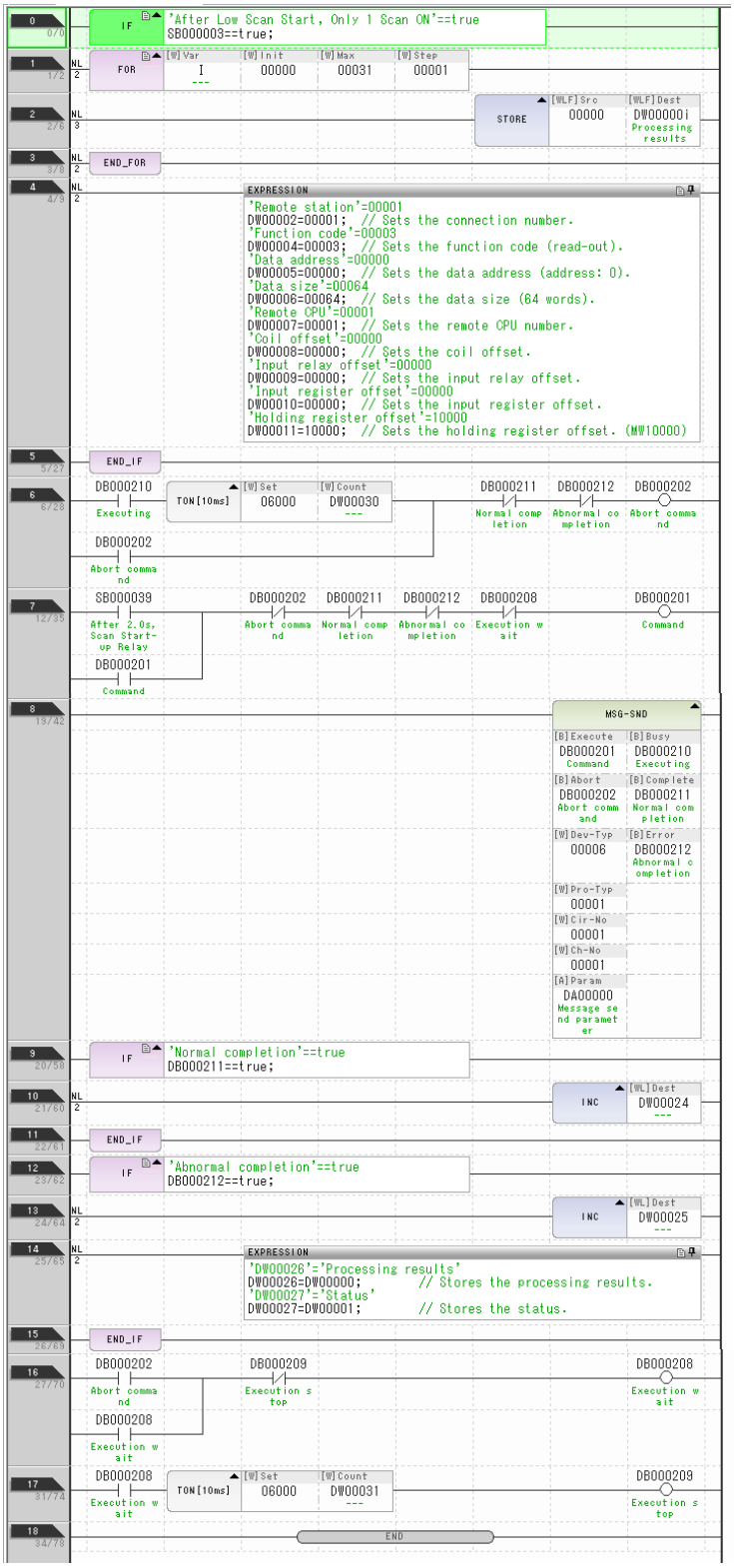
- In *(6) Programming Example* on page 4-42, the programming is written so that a message will be sent after two seconds has expired after the low-speed (or high-speed) scan has started. When the MSG-SND function has been completed normally, one scan will be skipped and then execution of the MSG-SND function will be started again, repeatedly sending a message each time sending the previous message has been completed normally.



(6) Programming Example

An example of ladder programming using the Message Send function for 218IF transmissions is shown below.

- For communication between CPUs using the MELSEC protocol, ladder programming for message communication is not required in the MELSEC PLC.



Sets parameters in the first scan after startup. (Low-speed scan (DWG.L): SB000003, High-speed scan (DWG.H): SB000001)

Clears DW register to zero.

Sets PARAM02 to PARAM11.

Forced stop if processing is not completed within 60 seconds after sending the Send command.

Function Send command
(Execution started 2 s after first ladder execution.)

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 1 for extended MEMOBUS or MELSEC, 2 for non-procedure
Cir-No. (circuit number): 2 (connection port)
Ch-No. (channel number): 1 (always 1 for the 218IF-01)

The normal completion coil turns ON at normal completion.

Increments the normal pass counter.

Ended with transmission errors.

Increments the abnormal pass counter.

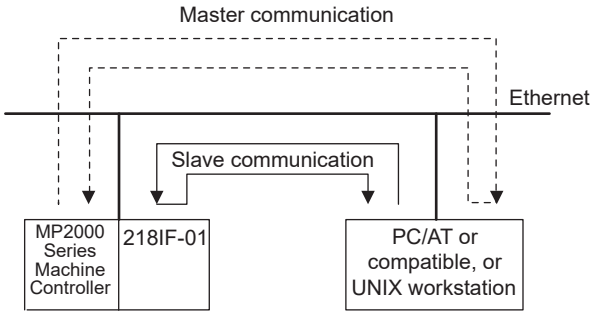
Stores the processing results and the status.

Function Send command

Enables sending the Send command when 60 seconds elapses.

4.6.3 Connection to a Personal Computer

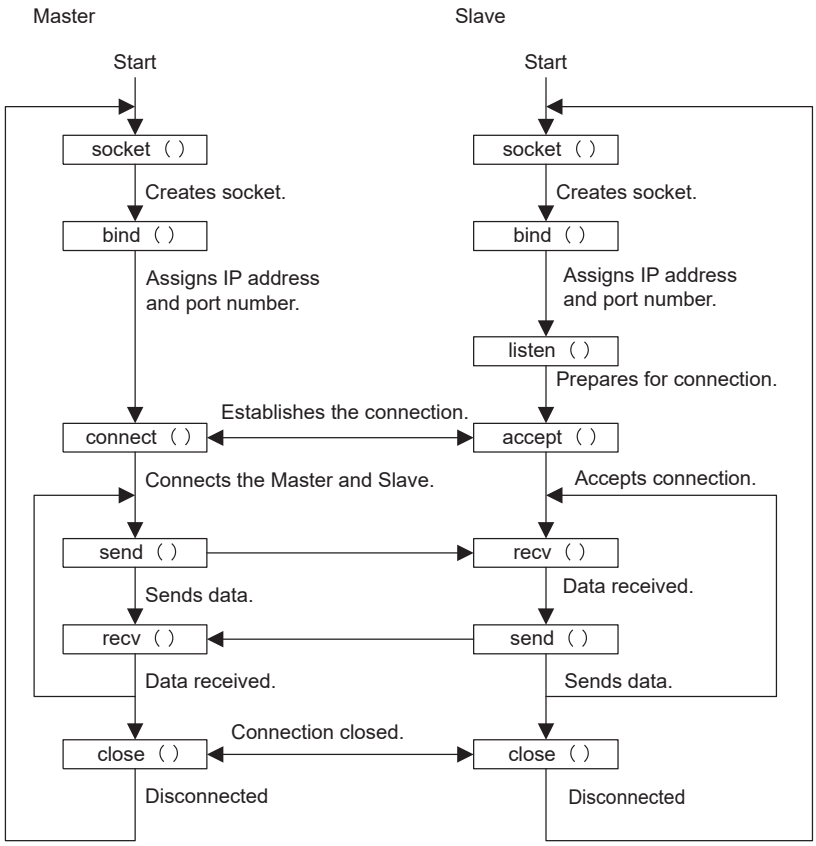
This section describes the network configuration in which a PC/AT or compatible computer or a UNIX workstation is connected to the 218IF-01 Module through an Ethernet interface of the computer or the workstation.
Original applications can be developed using the socket function in the computer or other device.



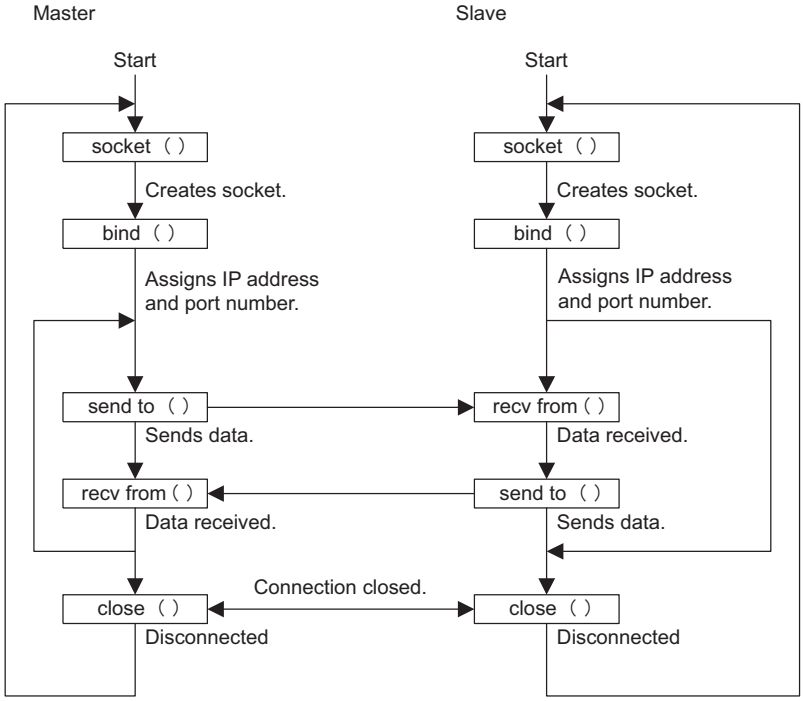
(1) Socket Communication Flow

The following diagram shows the logical programming flow using a socket interface.
When programming the computer using socket function, program according to the correct sequence depending on whether you are programming the Master or a Slave.
If execution of the MSG-SND function is started in the Master and execution of the MSG-RCV function is started in the Slave, the 218IF-01 will automatically perform the sequences for the Master and Slave.

[a] TCP

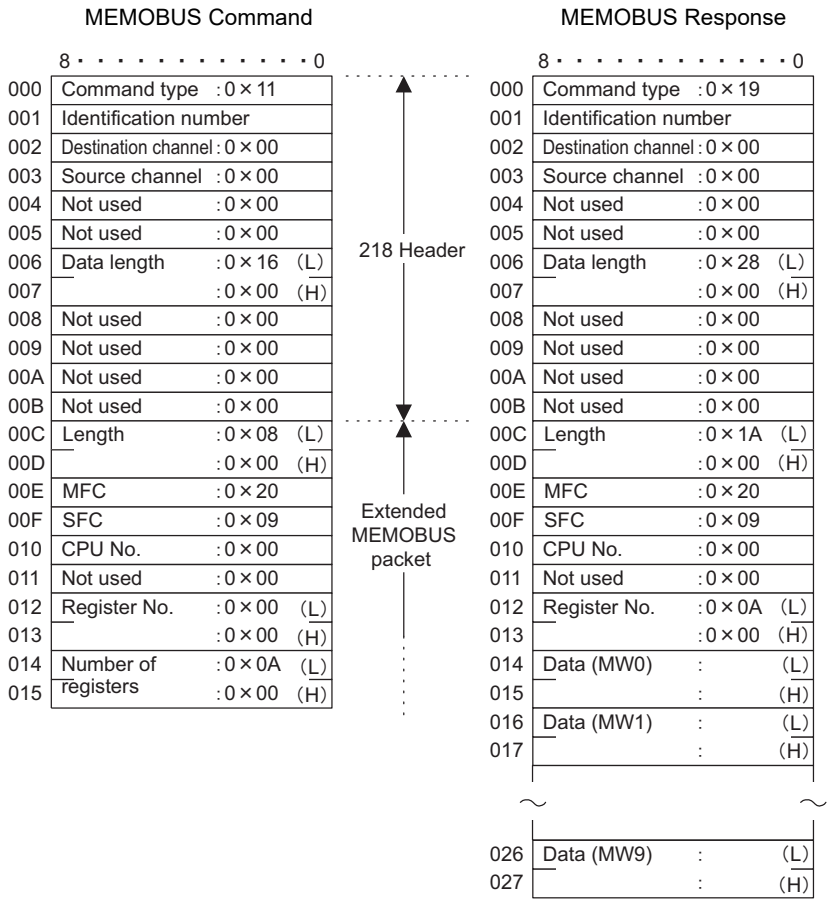


[b] UDP


Master
Slave
Start
Start
socket ( )
socket ( )
Creates socket.
Creates socket.
bind ( )
bind ( )
Assigns IP address
and port number.
Assigns IP address
and port number.
send to ( )
recv from ( )
Sends data.
Data received.
recv from ( )
send to ( )
Data received.
Sends data.
Connection closed.
close ( )
close ( )
Disconnected
Disconnected

(2) Data Format Example

The following diagram shows the data format when 10 words of data in holding registers of addresses 0 to 9 are read using the function code 09 (Holding Register Read (Extended) command), which is one of the Extended MEMOBUS commands.

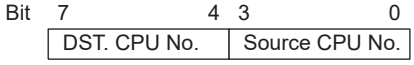


(3) Precautions on Creating Data Using a Personal Computer

[a] Using a Personal Computer as the Master

■ MEMOBUS Command Data

- Set the identification numbers each time communication is performed in the following increment data order: 0x00→0x01→0x02→....→0xff→0x00.
- Specify 0 for the destination channel.
- Specify 0 for the source channel.
- When the remote station has more than one CPU, set the CPU number as follows.



- DST. CPU No. (Destination CPU No.): Set to 1 for the MP2000 Series Machine Controller.
  Source CPU No.: Set to 0.

■ MEMOBUS Response Data

- Copy the identification numbers set in the MEMOBUS command data.
- Copy the destination channel numbers set in the MEMOBUS command data.
- A Machine Controller sets a unique value for the source channel. The computer may disregard for this number.
- The upper-place 4 bits and lower-place 4 bits of the CPU number set by the MEMOBUS command data are reversed for the CPU number. If the CPU number is not required at the computer, it would be disregarded.

[b] Using a Personal Computer as a Slave

■ MEMOBUS Response Data

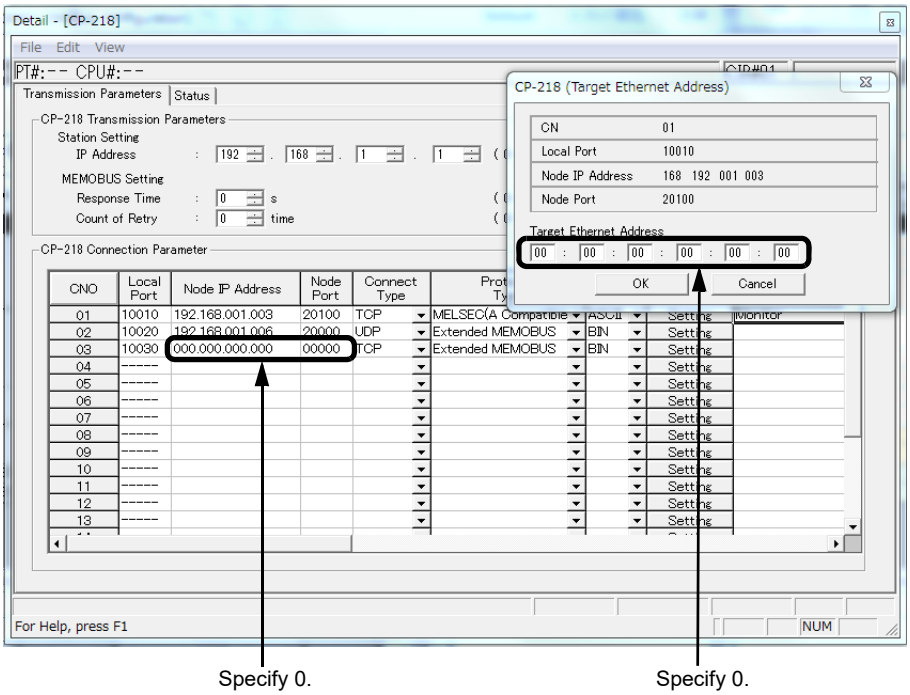
- Set the number that is set in the MEMOBUS command data for the identification number.
- Set the destination channel number that is set in the MEMOBUS command data.
- Set the source channel number that is set in the MEMOBUS command data.
- Reverse the upper-place 4 bits and lower-place 4 bits of the CPU number set by the MEMOBUS command data for the CPU number.

[c] Sample Program

Refer to *Appendix E Sample C Language Programs for the Master* on page A-35 and *Appendix F Sample C Language Programs for a Slave* on page A-41 for computer programming.

(4) Precautions on Programming

In Master programming created for TCP, the local IP address and port number may not be assigned with the bind( ) function depending on the application. If this happens, the system will assign an available port when the connect( ) function is called. In such cases, the remote IP address and remote port number will not agree with the values set in the **217IF Transmission Configuration** Window, causing the 218IF-01 to refuse the connection.
When using Master programming that may result in this problem, set the node IP address and node port in the **Transmission Parameters** Tab Page of the **218IF Transmission Configuration** Window to 0. With these settings, a connection request will be accepted from any station (unpassive open mode; refer to page 4-14). In this case, it is necessary to set all zeros for the remote station's Ethernet address as well.



(Refer to *4.4.2 Transmission Parameters Tab Page* on page 4-13.)

(Refer to *4.4.2 (5) Setting the Ethernet Address of the Remote Station* on page 4-19.)

4.7 Message Send Function and Message Receive Function

Refer to *5.7 Message Send Function* on page 5-48 for details on the Message Send function (MSG-SND) for Ethernet communication (218IF).
Refer to *5.8 Message Receive Function* on page 5-78 for details on the Message Receive function (MSG-RCV) for Ethernet communication (218IF).

5

218IF-02 Module

This chapter describes the 218IF-02 Module in detail and includes definition methods and programming examples.

- Before reading this chapter, be sure to read *Chapter 1 Starting Communication Modules*.

5.1 Overview ---- 5-3
5.1.1 Appearance and Connectors ---- 5-3
5.1.2 System Configuration Examples ---- 5-4
5.1.3 Module Specifications ---- 5-6
5.1.4 Indicators ---- 5-8
5.1.5 Switch Settings ---- 5-9
5.1.6 Default Parameters at Initial Startup ---- 5-10
5.2 Connection Specifications ---- 5-11
5.2.1 Connector Specifications ---- 5-11
5.2.2 Cable Specifications ---- 5-11
5.3 RS-232C Communication ---- 5-12
5.4 218IFB Transmission Definitions ---- 5-13
5.4.1 218IFB Transmission Configuration Window ---- 5-13
5.4.2 Transmission Parameters Tab Page ---- 5-15
5.4.3 Status Tab Page ---- 5-20
5.4.4 Saving and Deleting 218IFB Transmission Definitions ---- 5-23
5.5 Ethernet Communication ---- 5-24
5.5.1 Overview ---- 5-24
5.5.2 Ethernet Communication Protocols ---- 5-24
5.5.3 MEMOBUS Protocol ---- 5-25
5.5.4 Extended MEMOBUS Protocol ---- 5-26
5.5.5 MELSEC Protocol ---- 5-27
5.5.6 MODBUS/TCP Protocol ---- 5-32
5.5.7 Non-Procedure Protocol ---- 5-33
5.6 Connection Examples for Ethernet Communication ---- 5-35
5.6.1 Connecting 218IF-02 Modules ---- 5-35
5.6.2 Connection to MELSEC ---- 5-40
5.6.3 Connection to a Personal Computer ---- 5-44

5.7 Message Send Function ----------5-48
5.7.1 Basic Specifications ----------5-48
5.7.2 Inputs and Outputs for the Message Send Function ----------5-49
5.7.3 Parameter List for Ethernet Communication (218IF or 218IFB) ----------5-53
5.7.4 Function Setting Example and Parameter Details for the MEMOBUS and Extended MEMOBUS Protocol ----------5-54
5.7.5 Function Setting Example and Parameter Details for the MELSEC Protocol --------5-62
5.7.6 Function Setting Example and Parameter Details for the MODBUS/TCP Protocol ---5-67
5.7.7 Function Setting Example and Parameter Details for the Non-procedure Protocol ---5-73
5.8 Message Receive Function ----------5-78
5.8.1 Basic Specifications ----------5-78
5.8.2 Inputs and Outputs for the Message Receive Function ----------5-79
5.8.3 Parameter List for Ethernet Communication (218IF or 218IFB) ----------5-83
5.8.4 Function Setting Example and Parameter Details for the MEMOBUS and Extended MEMOBUS Protocol ----------5-84
5.8.5 Function Setting Example and Parameter Details for the MELSEC Protocol --------5-91
5.8.6 Function Setting Example and Parameter Details for the MODBUS/TCP Protocol ---5-95
5.8.7 Function Setting Example and Parameter Details for the Non-procedure Protocol --5-100

5.1 Overview

The 218IF-02 Module is a Communication Module that has an RS-232C serial interface and an Ethernet interface (100Base-TX or 10Base-T) mounted in it. Personal computers, HMI devices, and controllers manufactured by other companies can be connected to the 218IF-01 Module via the PORT serial interface and 100Base-TX/10Base-T connectors.

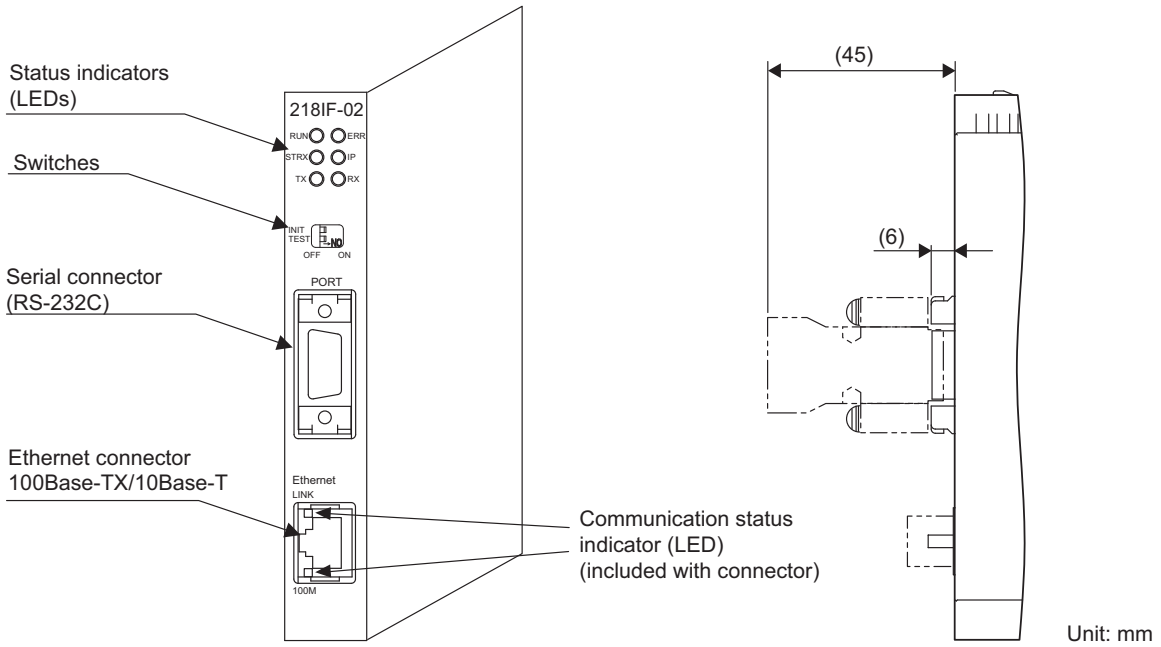
Message communication and engineering communication are supported. The MEMOBUS, MELSEC, OMRON, and non-procedure protocols are supported for serial communication, and the Extended MEMOBUS, MELSEC, MODBUS/TCP, and non-procedure protocols are supported for Ethernet communication.

The Ethernet interface on the 218IF-02 supports the Ethernet and CP-218 port types of the Communication Platform.

The rest of this chapter describes the appearance, specifications, and switch settings of the 218IF-02 Module.

5.1.1 Appearance and Connectors

The following diagram shows the appearance of the 218IF-02 Module and gives the external dimensions of the connectors.

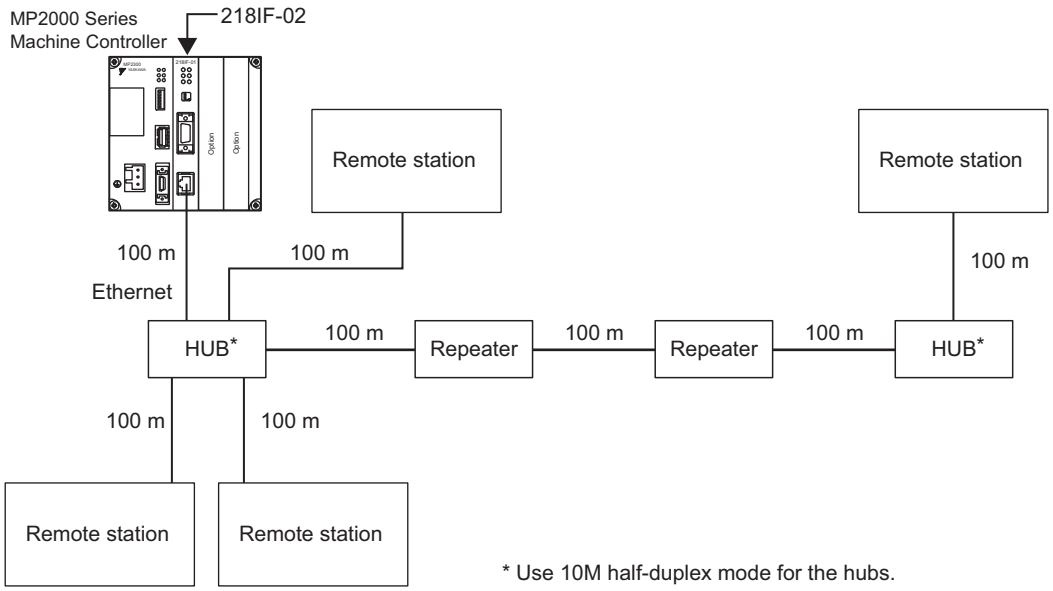


5.1.2 System Configuration Examples

(1) Examples of Ethernet Connections

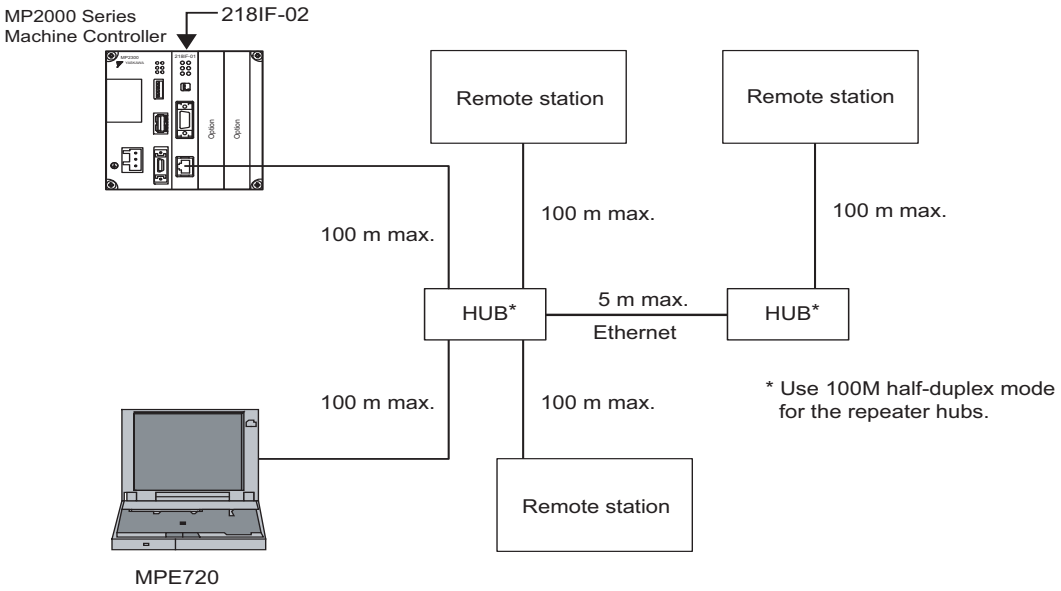
■ Connection Example 1

The following figure shows connections to Ethernet using 10Base-T.
The maximum length between the end nodes is 500 m with 10Base-T connections, and the maximum number of nodes is 5 nodes.



■ Connection Example 2

The following figure shows an Ethernet connection example for 100Base-TX using repeater hubs. The connection specifications are also shown.

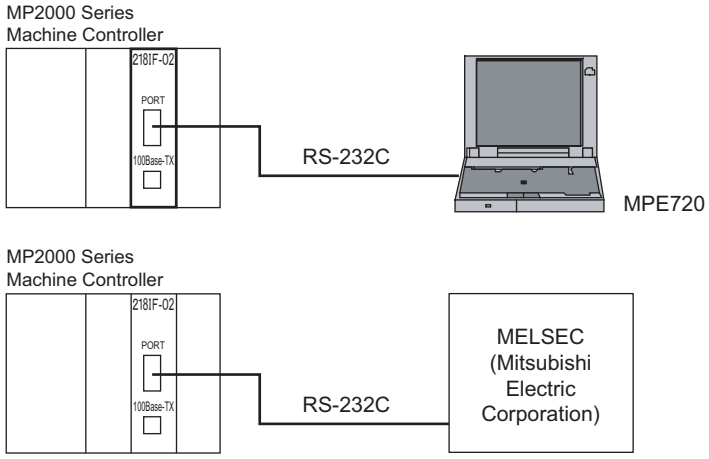


<Connection Specifications>

Item	With Repeater Hubs	With Switching Hubs
Cable length between MP2000 and hub	100 m max.	100 m max.
Cable length between hubs	5 m max.	100 m max.
Number of hubs between nodes	Up to 2	No restriction

(2) Example of Serial Connections

The following examples show serial connections to the PORT connector (RS-232C). The 218IF-02 Module is connected to a personal computer or a controller manufactured by another company on a one-to-one basis.



5.1.3 Module Specifications

This section provides the specifications of the 218IF-02 Module.

(1) Hardware Specifications

Item	Specification
Name	218IF-02
Model Number	JAPMC-CM2302-E
Communication Ports	RS-232C: 1 port (PORT)
	Ethernet: 1 port (100Base-TX)
Indicators	Module status LED indicators RUN (green) ERR (red) STRX (green) IP (green) TX (green) RX (green)
	Ethernet status indicator LEDs 100M (green), LINK (orange)
Switches	INIT, TEST
Dimensions (mm)	125 x 95 mm (H x D)
Mass	85 g

(2) Transmission Specifications

■ RS-232C Transmission Specifications

Item	Specification
Connector	9-pin D-sub (female)
Transmission Distance	15 m max.
Baud Rate	9600/19200/38400/57600/76800/115200 bps
Access Mode	Asynchronous (start-stop synchronization)
Communication Type	Half duplex
Communication Modes	Message communication and engineering communication
Communication Protocols	MEMOBUS, MELSEC, OMRON, and non-procedure
Media Access Control Method	1:1
Transmission Format (Can be set.)	Data bit length: 7 or 8 bits Stop bits: 1 or 2 bits Parity: Odd, even, or none

■ Ethernet Transmission Specifications

Item		100Base-TX Specifications	10Base-T Specifications
Interface		RJ-45	
Baud Rate		100 Mbps	10 Mbps
Access Mode		IEEE802.3u CSMA/CD	IEEE802.3i CSMA/CD
Communication Type		Full duplex/half duplex *	
Communication Modes		Message communication and engineering communication	
Communication Protocols		MEMOBUS, Extended MEMOBUS, MELSEC, MODBUS/TCP, and non-procedure	
Maximum Number of Cascade Connections		2 layers	4 layers
Maximum Message Length		100 m (distance between hub and node when using UTP)	
Connection Type		TCP/UDP/IP/ARP/ICMP	
Transmis-sion Media	Twisted-pair cable (UTP)	Category 5 or 5e	Category 3, 4, 5, or 5e
	Twisted-pair cable (STP)	Category 5 or 5e (100 Ω)	Category 3, 4, 5, or 5e (100 Ω)
Maximum Number of Transmission Words		MEMOBUS: 100 words for writing, 125 words for reading Extended MEMOBUS: 2043 words for writing, 2044 words for reading MELSEC: 1017 words for writing, 1017 words for reading MODBUS/TCP: 100 words for writing, 125 words for reading Non-procedure: 2046 words for writing	
Maximum Number of Connections		20 (maximum of 10 connections for simultaneous communication)	

* The Ethernet communication type and baud rate will be automatically determined by the auto-negotiation function. The following table lists the communication types of remote devices (the hub or communication device directly connected to the 218IF-02) with which the 218IF-02 can communicate.

218IF-02's communication type	Remote Device's Communication Type				
Auto-negotiation	Auto-negotiation	Half-duplex 10Base-T	Full-duplex 10Base-T	Half-duplex 100Base-TX	Full-duplex 100Base-TX
	Depends on remote device.	Communicates using half-duplex 10Base-T.	Communication not possible. (A communication error will occur.)	Communicates using half-duplex 100Base-TX.	Communication not possible. (A communication error will occur.)

Communication is not possible with remote devices that support only full-duplex communication. Set the remote device to half-duplex 10Base-T or half-duplex 100Base-TX.
If the remote device supports auto-negotiation, then the communication type will be selected automatically. The 218IF-02 will attempt to connect in the following order: Full-duplex 100Base-TX, half-duplex 100Base-TX, full-duplex 10Base-T, and then half-duplex 10Base-T.

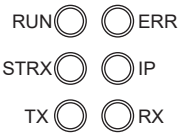
(3) Operating Environment Specifications

Item		Specification
Environmental Conditions	Ambient Operating Temperature	0 to 55°C
	Ambient Storage Temperature	–25 to 85°C
	Ambient Operating Humidity	30% to 95% (with no condensation)
	Ambient Storage Humidity	5% to 95% (with no condensation)
	Pollution Level	Pollution level 2 (conforming to JIS B 3502)
	Corrosive Gas	There must be no combustible or corrosive gas.
	Operating Altitude	2,000 m above sea level or lower
Mechanical Operating Conditions	Vibration Resistance	Conforming to JIS B 3502 Vibration and acceleration: 10 to 57 Hz with single-amplitude of 0.075 mm 57 to 150 Hz with fixed acceleration of 9.8 m/s^2 10 sweeps each in X, Y, and Z directions (sweep time: 1 octave/min)
	Shock Resistance	Conforming to JIS B 3502 Peak acceleration of 147 m/s^2 (15 G) twice for 11 ms each in the X, Y, and Z directions
Electrical Operating Conditions	Noise Resistance	Conforming to EN 61000-6-2, EN 61000-6-4, EN 55011 (Group 1 Class A)
Installation Requirements	Ground	Ground to 100 Ω max.
	Cooling Method	Natural cooling

5.1.4 Indicators

(1) Status Indicators (LEDs)

The following table shows the status of the 218IF-02 Module shown by the LED indicators.



Indicator	Color	Meaning When Lit	Meaning When Not Lit
RUN	Green	Operating normally	An error has occurred.
ERR	Red	A failure has occurred (lit or blinking).	Normal
STRX	Green	Sending or receiving RS-232C data	No RS-232C data communication
IP	Green	IP address has been set.	IP address has not been set.
TX	Green	Sending Ethernet data	Not sending Ethernet data
RX	Green	Receiving Ethernet data	Not receiving Ethernet data

■ Status Indication by LED Indicators

The following table shows the LED indications and their meanings for operating status and errors.

Status	Indicators						Meaning	Description
	RUN	ERR	IP	STRX	TX	RX		
Power ON	○	●	–	–	–	–	Initializing	This is the status immediately after the power to the 218IF-02 is turned ON (initializing). When initialization has been completed, the ERR indicator goes out and the 218IF-02 enters initialized status.
Initialized	●	○	–	–	–	–	Waiting for initial settings	The 218IF-02 starts normally and waits for initial settings from the controller (CPU). The STRX and RX indicators may be lit depending on the connection status.
Normal	●	○	–	–	–	–	Operating normally	The 218IF-02 is performing serial or Ethernet communication normally. The STRX indicator blinks during serial communication. The TX and RX indicators blink during Ethernet communication. The IP indicator is lit when the IP address is set.
Error	★	★	–	–	–	–	Hardware failure (number of blinks) 1: ROM diagnosis error 2: RAM diagnosis error 3: Shared memory diagnosis error	The 218IF-02 hardware failed. The Module must be replaced.
	–	★	–	–	–	–	Software or hardware error (number of blinks) 15: Watchdog timeout error	215AIF-02 processor error. The Module must be replaced.

- ● : Lit
  ○ : Not lit
  ★ : Blinking
  – : Not relevant

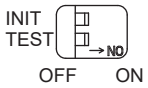
(2) Communication Status Indicators (LED) (Included with Ethernet Connector)

The indicators (LEDs) included with the Ethernet connector show the status of Ethernet communication.

Indicator	Color	Meaning When Lit	Meaning When Not Lit
LINK	Orange	Ethernet link established.	Ethernet link not established.
100M	Green	100 Mbps	10 Mbps or not connected

5.1.5 Switch Settings

The following table shows the 218IF-02 Module switch settings.



Label (Switch No.)	Name	Status	Function	Factory Setting
INIT (2)	Transmission Parameters Startup Selection	ON	For engineering communication. Starts using the default parameters except setting for automatic reception. Given higher priority than the Machine Controller flash startup and self-configuration startup.	OFF
		OFF	Set to OFF for Machine Controller flash startup or self-configuration startup.	
TEST (1)	Operating Mode Selection	ON	System use.	OFF
		OFF	Normal operation (Always leave set to OFF.)	

- Always leave the unused switches (3 and 4) set to OFF.

5.1.6 Default Parameters at Initial Startup

At initial startup (INIT switch set to ON), the following parameter settings will be made.

■ Ethernet Interface (The setting of INIT switch to ON is valid only when the power is turned ON.)

Item	Default Setting
Local IP Address	192.168.1.1
Baud Rate	Automatic
Subnet Mask	255.255.255.0
Gateway IP Address	0.0.0.0
System Port (Engineering Port)	9999 (UDP)

■ RS-232C Interface (The setting of INIT switch to ON is valid while the power is being supplied.)

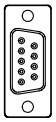
Item	Default Setting		
Communication Protocol	MEMOBUS		
Master/Slave	Slave		
Device Address	1		
Serial I/F	RS-232C		
Communication Modes	RTU		
Data Length	8 bits		
Parity	Even		
Stop Bits	1 bit		
Baud Rate	19.2 kbps		
Send Delay	Disable		
Receive Monitor Time	Disable		
Automatic Reception	Enable		
Interface Register Settings in the Slave		Leading Register	Number of Words
	Read Input Relay Status	IW0000	32768
	Read Input Register Status	IW0000	32768
	Read/Write Coil Status	MW00000	65535
	Read/Write Holding Register Status	MW00000	65535
	Coil/holding Register Write Range	LO MW00000	
		HI MW65534	

5.2 Connection Specifications

5.2.1 Connector Specifications

This section provides the connector specifications for the 218IF-02 Module.

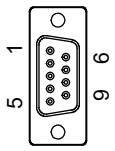
(1) Connector Specifications

Connector Shape	Name	Connector Name	No. of Pins	Connector Model		
				Module	Cable	Manufacturer
PORT 	RS-232C	PORT	9	17LE-13090-27 (D2BC) D-sub 9-pin female connector	17JE-23090-02 (D8B) 9-pin D-sub male connector (Shell included.)	DDK Ltd.
Ethernet	100Base-TX	Ethernet	8	JOG-0001NL (modular jack)	–	Pulse Engineering

(2) Connector Pin Arrangement

■ PORT Connector

The PORT connector is used to connect the MP2000 Series Machine Controller to computers and HMI devices via an RS-232C connection.



Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	FG	Frame ground	6	–	–
2	SD	Send data	7	SG	Signal ground (0V)
3	RD	Receive data	8	–	–
4	RS	Ready to send	9	ER	Data terminal ready
5	CS	Clear to send	–	–	–

■ Ethernet Connector (100Base-TX)

The Ethernet connector is used to connect the MP2000 Series Machine Controller to computers and HMI devices via an Ethernet (100Base-TX) connection.

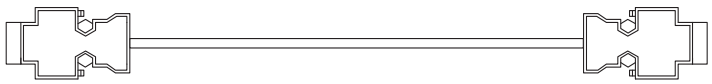
Ethernet

Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	TXD+	Send data +	5	–	–
2	TXD-	Send data -	6	RXD-	Receive data -
3	RXD+	Receive data +	7	–	–
4	–	–	8	–	–

5.2.2 Cable Specifications

This section provides the cable specifications for the 218IF-02 Module.

(1) RS-232C Cable (for MPE720 Engineering)

Model Number	Length	External Appearance (JEPMC-W5311-□□-E)
JEPMC-W5311-03-E	2.5 m	
JEPMC-W5311-15-E	15 m	

(2) Ethernet Cable

Yaskawa does not provide Ethernet cables. Obtain a commercially available 100Base-TX cross or straight cable.

5.3 RS-232C Communication

Refer to *Chapter 3 Serial Communication* for information on the transmission definitions, connection examples, programming examples, the Message Send function, the Message Receive function, and other details on communication through the RS-232C port on the 218IF-02 Module.

5.4 218IFB Transmission Definitions

To perform 218IF-02 Ethernet communication with the MP2000 Series Machine Controller, the 218IFB transmission (Ethernet communication) definition file must be created.
This section describes how to define 218IFB transmissions using MPE720 windows.

5.4.1 218IFB Transmission Configuration Window

To define the 218IFB transmissions, first the **Module Configuration** Window is opened on the MPE720, and then the **218IFB Transmission Configuration** Window is opened.

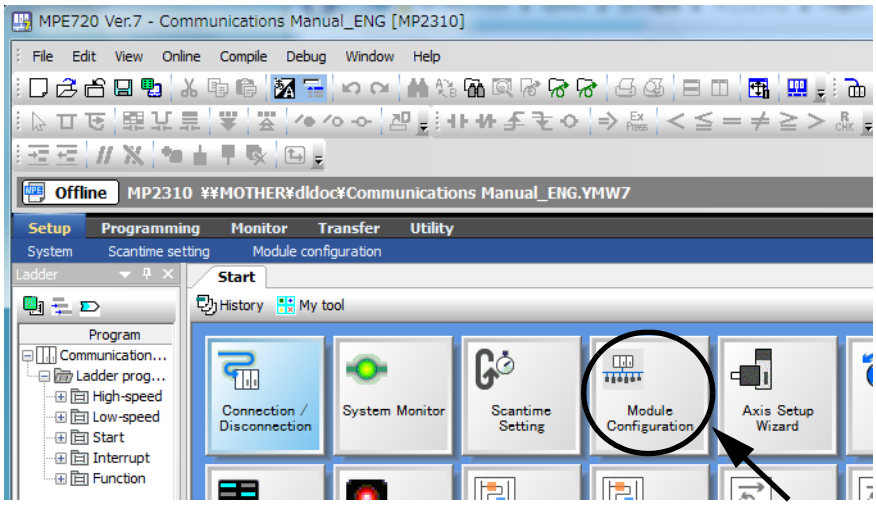
(1) Displaying the Module Configuration Window

Use the following procedure to display the **Module Configuration** Window.

1. Start the MPE720 on the personal computer connected to the Machine Controller and open the project file.

- For information on starting the MPE720, refer to *MP2000/MP3000 Series Engineering Tool MPE720 Version 7 User's Manual (Manual No.: SIEP C880761 03)*.

2. Click the **Module Configuration** Button on My Tool View.



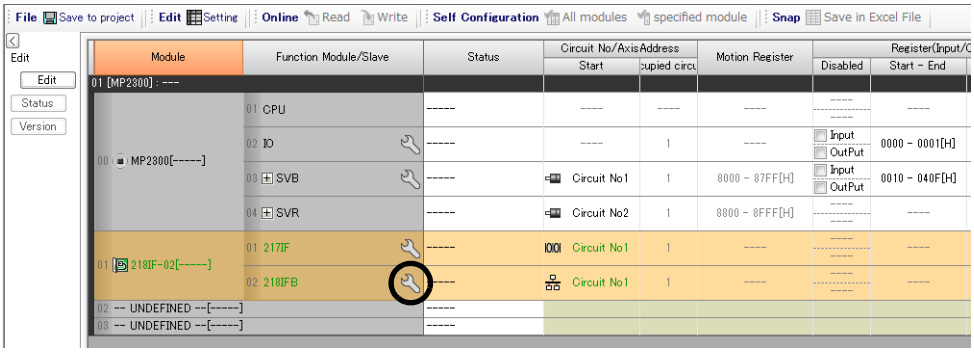
The **Module Configuration** Window will be displayed.

(2) Displaying the 218IFB Transmission Configuration Window from the Module Configuration Window

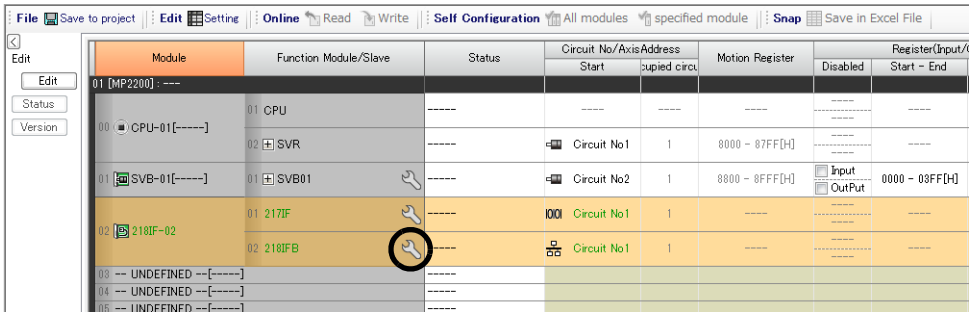
As the result of executing self-configuration, all Option Modules connected to the Machine Controller will be displayed in the **Controller** Area of the **Module Configuration** Window. (Refer to *1.5 Executing Self-configuration* on page 1-26.)

In the **Controller** Area, select the cell for 218IF-02, and then double-click the slot number cell for the 218IFB in the **Module Details** Area. The **218IFB Transmission Configuration** Window (see next page) will be displayed.

<MP2300/MP2300S/MP2310 Module Configuration Window>



<MP2100M/MP2200 Module Configuration Window>



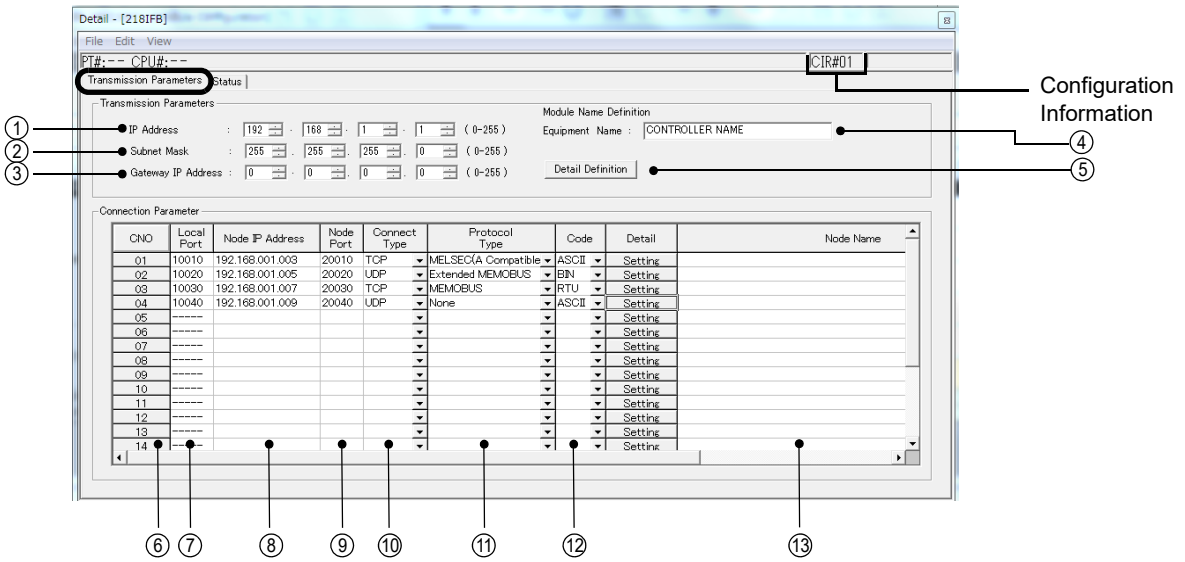
In Online Mode, the 218IFB transmission definitions data saved in the Machine Controller are displayed. In Offline Mode, the definitions data saved in the personal computer on which MPE720 is running is displayed.

The **218IFB Transmission Configuration** Window is composed of two tab pages: **Transmission Parameters** and **Status**. These tab pages are used to set the definitions and monitor the settings.

- If the **Transmission Configuration** Window is being opened for the first time, a "new file" message box will be displayed and the **218IFB Transmission Configuration** Window will be displayed when the **OK** Button is clicked.
- If the cell for the 217IF in the **Module Details** Area is double-clicked, the **217IF Transmission (serial communication) Configuration** Window will be displayed. Refer to *Chapter 3 Serial Communication* for details on serial communication.

5.4.2 Transmission Parameters Tab Page

Set the parameters required for using Ethernet communication.



■ Configuration Information

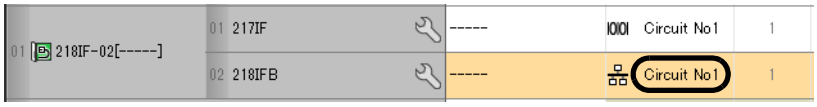
Displays the 218IF-02 Module configuration information. The configuration information is the same as the information displayed in the **Module Details** Area in the **Module Configuration** Window.

RACK#: The rack number of the rack in which the 218IF-02 is defined.

Slot#: The slot number of the slot in which the 218IF-02 is defined.

CIR#: The circuit number of the Ethernet port on the 218IF-02.

- The configuration information can also be changed in the **Module Details** Area of the **Module Configuration** Window. Do not use a circuit number that is already being used for another Ethernet Module (218IF or 218IFB).

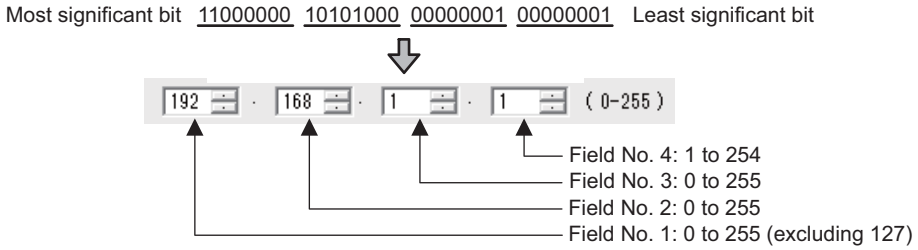


- When the MSG-SND or MSG-RCV function is used, set the circuit number displayed here for **CIR#** (circuit number).

■ Setting the Transmission Parameters

① IP Address

Enter the IP address of the local station. There are four 8-bit fields delimited by periods (.). Input a decimal number for each field.



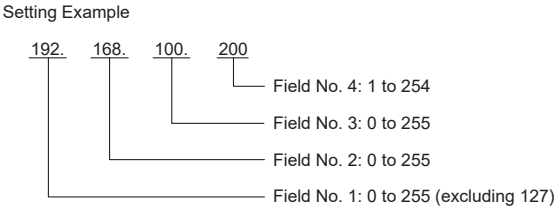
- Do not use the same address as another node in the Ethernet network.
- IP address 192.168.1.1 to 192.168.1.254 are recommended as private addresses. Check with your network administrator when setting an IP address.

② Subnet Mask

Enter the subnet mask for the IP address of the local station. The range that can be entered in fields 1 to 3 is 0 to 255. The range that can be entered in field 4 is 1 to 254. When a subnet mask is not being used, enter 0 in fields 1 to 4.

③ Gateway IP Address

When communication is performed with other networks connected through a gateway (router), enter the IP address of the gateway.
If there is no gateway or a gateway is not used, enter 0 for the four fields.
The data input range depends on the field.



- Do not set the same address more than once for the IP addresses and gateway IP addresses.

④ Module Name Definition

Set a name of 16 characters or less for the 218IF-02.

■ Terminology: Subnet Mask

A subnet mask is a mask value used to get the network address of the subnet from the IP address. The result of AND operation of the IP address and the subnet mask will be the subnet address.
Contact your network administrator when setting a subnet mask.

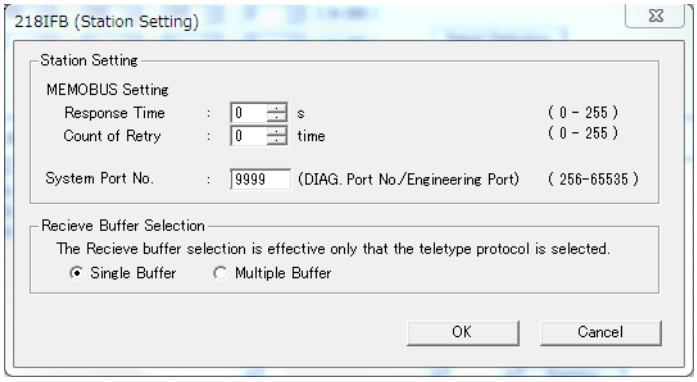
■ Terminology: Gateway IP Address

The IP address of the gateway device (router) through which communication is performed between multiple network segments.
Contact your network administrator when setting a gateway IP address.

---

⑤ Detail Definition Button

Click this button to display the **Station Setting** Dialog Box and enable setting the response time, number of retries, and system port number.



Response Time

Enter the time (0 to 255) that the 218IFB transmission system waits for a response after sending a command using the MSG-SND function. If a response is not returned, causing a timeout, the transmission will be retried.
The transmission will be retried as many times as set in **Count of Retry** parameter described below.
The **Count of Retry** parameter is not valid if 0 is set for the **Response Time**.

- Enter 0 for the **Response Time** if 0 is set for the **Count of Retry** in the MEMOBUS setting (if **MEMOBUS** is set in the **Protocol Type** column of the **Transmission Parameters** Tab Page.)

Count of Retry

Enter the number of retries (0 to 255) to be attempted if a timeout is detected after sending a command with the MSG-SND function. An error is returned for the MSG-SND function if a response is not returned after the set number of retries.

- If only the TCP protocol is set as the connection type, the number of retries does not need to be set. Set 0.

System Port No.

Set the 218IFB Ethernet port number to use for engineering communication with the MPE720 to between 256 and 65535. The default is 9999.

- If this value is changed, the port set in the detail settings of the **Logical Port Setting** Dialog Box for the MPE720 Communication Platform must also be changed.
- Do not set the same port number as the local port number.
- Do not set 9998 or 10000. These are used by the system.

Receive Buffer Selection

Select single buffer or multiple buffers when receiving data with the non-procedure protocol.
The following table gives the characteristics of single buffer and multiple buffers.

	Single Buffer	Multiple Buffers
Number of buffers in the controller	1	20
When the MSG-RCV function is executed	Received data stored in the buffer in the controller is read out.	The oldest received data stored in the buffer in the controller is read out. After the data is read out, the buffer is emptied.
When data is next received before the MSG-RCV function is executed	The buffer in the controller is updated with the next received data.	• If there is an empty buffer The next received data is stored in the empty buffer. • If there are no empty buffers The next received data is discarded.
Received data in controller	The received data is retained for five to 10 seconds after it is received. If the received data is not read out by the MSG-RCV function within five seconds after the data is received, the buffer is emptied.	Each item of received data is retained for five to 10 seconds after it is received. If the received data is not read out by the MSG-RCV function within five seconds after the data is received, the buffer is emptied.

- The communications module software versions and MPE720 versions that support this function are shown below.
  Software versions of supported communications module: Ver. 1.03 or later
  MPE720 Ver.5: Ver. 5.52 or later
  MPE720 Ver.6: Ver. 6.22 or later
  MPE720 Ver.7: Ver. 7.10 or later
  The software version is given on a label on the PCB of the Communication Module as "V**.**".
- For a protocol other than the non-procedure protocol, the received data is processed in the same manner as when single buffer is selected, regardless of whether multiple buffers are selected.

■ Setting the Connection Parameters

⑥ CNO (Connection Number)

The connection number (1 to 20) is displayed.
With Ethernet communication, the connection number is used to identify remote stations.
The connection numbers corresponds to the **Remote Connection #** (parameter 02) of the MSG-SND and MSG-RCV functions.

⑦ Local Port

Enter the port number of the local station (256 to 65535) for each connection. Do not enter a port number used for another connection, for the system port, or for diagnostics. (Refer to *4.4.2 (4) Local Station and TCP/IP Settings* on page 4-17.) If 0 is entered for this parameter, the data for that connection number is cleared and "---" is displayed in the **Local Port** column.

- The port number set for the system port, as well as 9998 and 10000, cannot be used for UDP.

⑧ Node IP Address (Remote Station's IP Address)

Enter the IP address of the remote station for each connection.

The data input range depends on the field of the IP address.

Field No. 1: 0 to 255 (excluding 127)
Field No. 2: 0 to 255
Field No. 3: 0 to 255
Field No. 4: 1 to 254

Enter 0 in all four fields for the unpassive open mode (see *Explanation* below.)

⑨ Node Port (Remote Station's Port Number)

Enter the port number (0 or 256 to 65535) of the remote station for each connection. Do not use a combination of the **Node IP Address** and the **Node Port** that is the same as the one assigned for any other connection.

⑩ Connect Type (Connection Type)

Select the transport-layer protocol.

TCP: Communication is performed using TCP (Transmission Control Protocol).
UDP: Communication is performed using UDP (User Datagram Protocol).

⑪ Protocol Type

Select the application-layer protocol for each connection according to the protocol supported by the remote station. The following table outlines each protocol type.

• Refer to *5.5.2 Ethernet Communication Protocols* on page 5-24 for details on the protocols.

Protocol Type	Description
Extended MEMOBUS	Yaskawa's extended MEMOBUS protocol
MEMOBUS	Yaskawa's standard MEMOBUS protocol
MELSEC	Ethernet interface protocol used with Mitsubishi MELSEC A-series General-purpose Programmable Logic Controllers.
Non-procedure protocol	A protocol provided to implement general-purpose message communication. Data is sent or received and written to consecutive MW registers without processing.
MODBUS/TCP	Ethernet protocol, proposed by Modicon, used in industrial applications.

■ Terminology: Connection

A series of operations to confirm communication and transfer data in one-to-one communication between local station programs and remote station programs.

■ Terminology: Port Number

The port number is used to identify the communication target program in the remote station. There is a one-to-one correspondence between port numbers and communication programs. A port number is entered in the header section of the data, together with the IP address and other information. The remote station transfers the data to the target program identified by the destination port number.

The port number at the local station also enables each local station and remote station communication service program to be identified.

■ Explanation: Unpassive Open Mode

If the remote station's address is set to 000.000.000.000 and the remote station's port number is set to 0, the connection is set in the Unpassive Open mode.

In Unpassive Open mode, the 218IF-02 Module is connected to the station that has accessed the connection number of the 218IF-02 Module, and if more than one station accesses the 218IF-02 Module's connection number, the connection will be established with the station that sent the connection request first. If a connection request is sent from a station while the connection is established with another station in Unpassive Open mode, the connection that was established first is disconnected and a connection will be established with the station that sent the connection request later.

■ Explanation: TCP and UDP

TCP is a connection protocol and UDP is a connectionless protocol.
A connection protocol ensures reliable communication because it performs various steps of communication control, such as arrival checks, error detection and correction, sequence number checks, and send data size control.
In contrast, a connectionless protocol does not provide the procedures that ensure the quality of communication, and data transmissions are performed in only one direction. Connectionless protocols thus provide high-speed communication, but less communication reliability.
The selection of TCP or UDP depends on the requirements for communication. Select TCP if reliability is important and select UDP if speed is important.

⑫ Code

The code of the data to be transmitted at each connection is handled according to the code set at the remote station.

RTU: Specifies RTU mode when the MEMOBUS protocol is being used.
ASCII: Specifies ASCII mode.
BIN: Specifies binary mode.
The code that can be selected is restricted by the selection of the protocol type in the Protocol Type column as shown in the following table.

Protocol Type	Code			Protocol Type	Code		
	RTU	ASCII	BIN		RTU	ASCII	BIN
Extended MEMOBUS	×	○	○	Non-procedure protocol	×	○	○
MEMOBUS	○	○	×	MODBUS/TCP	×	×	○
MELSEC	×	○	○				

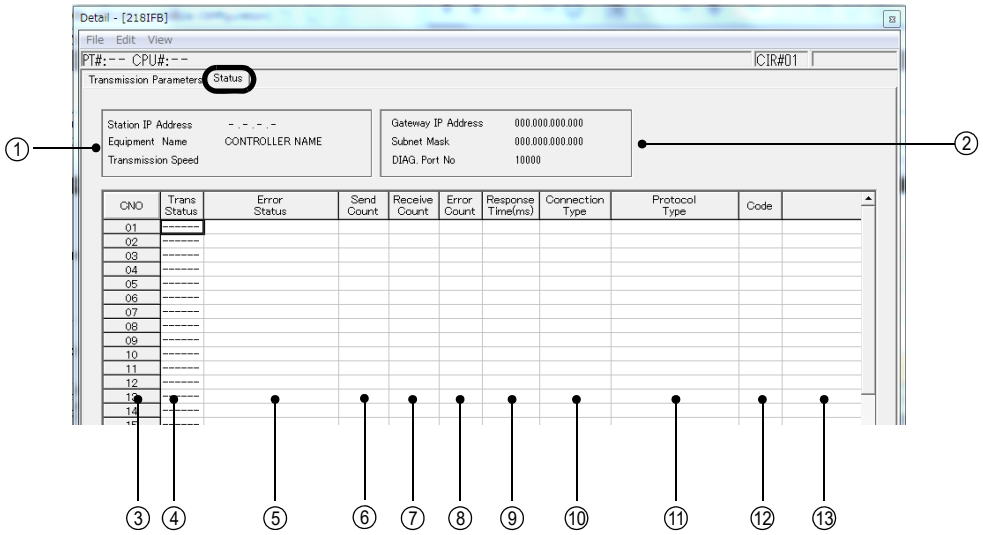
- ○: Can be selected.
  ×: Cannot be selected.

⑬ Node Name (Remote Station Name)

Enter a connection comment of up to 32 characters.

5.4.3 Status Tab Page

The **Status** Tab Page displays the 218IFB Module configuration information and transmission status. The settings cannot be changed.



- If the **Status** Tab is clicked without saving 218IFB transmission definition data, a confirmation message to save data or an error message (if inconsistencies exist in the data) may be displayed. If a confirmation message to save data is displayed, click the **Yes** Button to save the setting data and the **Status** Tab Page will be displayed. If an error message is displayed, click the **OK** Button, enter correct parameters, and save the data. Then, click the **Status** Tab.
- The data on the **Status** Tab Page is displayed only in online mode. Nothing is displayed in offline mode.

■ Status Display Details

① Station IP Address, Equipment Name, and Transmission Speed (Baud Rate)

Displays the local station's IP address and name set on the **Transmission Parameters** Tab Page.

② Gateway IP Address, Subnet Mask, and DIAG. Port No. (System Port)

Displays the subnet mask, gateway (router) IP address, and system port number set on the **Transmission Parameters** Tab Page.

③ CNO (Connection Number)

The connection number (1 to 20) is displayed.

④ Trans Status (Transmission Status)

Displays the status of each connection as shown below.

Status Display	Status
IDLE	Idle status
WAIT	Wait status (waiting for connection)
CONNECT	Connected status (data send/receive enabled)
–	Unused connection

⑤ Error Status

The **Error Status** Column displays details on the error if an error has occurs in the transmission status. The error status displays and meanings are listed in the following table.

Status Display	Meaning	Description
No error	Normal	
Socket Creation Error	System error	Failure in socket generation
Local Port No Error	Local station port number setting error (Binds the same address during disconnection of the TCP connection.)	Binding error (port number duplication)
		When the ABORT function was executed using the MSG function, a binding error occurred during disconnection. This error occurs if Execute is turned ON within 1 minute after Abort is input and then Complete is automatically turned ON.
		Before the completion of connection processing, the command was sent to the same remote station by another function.
Changing Socket Attribute	System error (in TCP mode)	The error occurred while setting the socket attribute.
Connection Error (M-SND)	Connection error (Connection was rejected by the remote station when establishing a connection with an unpassive open in TCP mode.)	The command was reset since the remote station rejected the connection although the connection was attempted by the MSG-SND function.
		Connection was disabled even at 1-minute retry (default) after the cable was disconnected.
Connection Error (M-RCV)	Connection error (connection with a passive open in TCP mode)	MSG-RCV function communication acceptance error
System error	System error	Socket polling (SELECT used) while receiving data.
TCP Data Send Error	Data sending error (The remote station does not exist or the remote station has not started when using TCP.)	A response transmission error occurred with the MSG-RCV function. The same error occurred with the MSG-SND function. This error occurs when the target remote station for sending the data does not exist, or the remote station is rebooted (only with a TCP connection).
UDP Data Send Error	Data sending error (When using UDP)	The data send request command was sent to a socket that does not exist.
TCP Data Receive Error	Data receiving error (The Module received a request to disconnect the connection from the remote station when using TCP.)	This error occurs when the connection is disconnected by the remote station. The error occurs even if close processing is performed normally.
Data Receiving Error (UDP)	Data receiving error (when using UDP)	The data receiving command was executed for a socket that does not exist.
Changing Socket Option Error	System error	An error occurred while changing the socket option.
Converting Data error	Data conversion error	Protocol conversion error

- The last error displayed in the error status will be retained until the power is turned OFF and then back ON even if communication recovers. To clear the display, turn the power to the Controller OFF and then back ON.

⑥ Send Count (Transmission Pass Counter)

Displays the amount of data (number of packets) sent to the remote station.

⑦ Receive Count (Reception Pass Counter)

Displays the amount of data (number of packets) received from the remote station.

⑧ Error Count

Displays the number of times an error has occurred for each connection.

⑨ Response Time [ms]

Displays the time (ms) taken to receive a response for a command sent using the MSG-SND function.

⑩ Connection Type

Displays the Connect Type (TCP or UDP) set on the **Transmission Parameters** Tab Page.

⑪ Protocol Type

Displays the Protocol Type (MEMOBUS, Extended MEMOBUS, MELSEC, None (non-procedure), MODBUS, or TCP) set on the **Transmission Parameters** Tab Page.

⑫ Code

Displays the Code (ASCII, binary, or RTU) set on the **Transmission Parameters** Tab Page.

⑬ Node Name (Remote Station Name)

Displays the Node Station name set on the **Transmission Parameters** Tab Page.

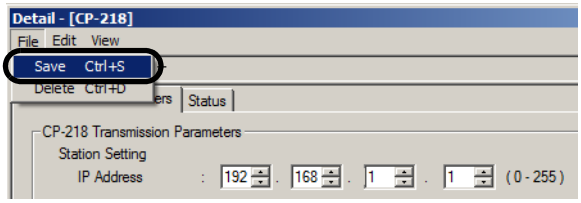
5.4.4 Saving and Deleting 218IFB Transmission Definitions

- When changing, saving, or deleting 218IFB transmission definitions data, save the data to flash memory in the Machine Controller, and then turn the power to the Machine Controller OFF and then turn it ON again.

(1) Saving the 218IFB Transmission Definitions Data

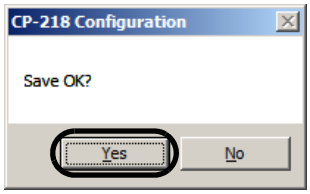
Use the following procedure to save the 218IFB transmission definitions data if it has been set or changed. In Online Mode, the definitions data is saved in the Machine Controller and in personal computer running the MPE720. In Offline Mode, the definitions data is saved in the personal computer running the MPE720.

1. Select ***File - Save*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

2. Click the **Yes** Button.

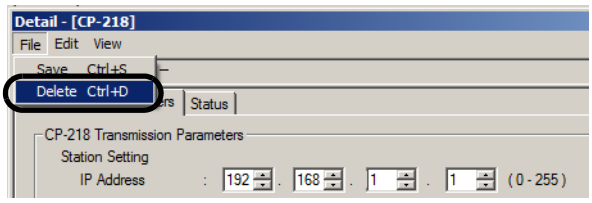


- An error message will be displayed if the input values for the parameters do not pass the consistency check. In this case, correct the cause of the error by referring to the following manual and try to save the data again. *MPE720 Software for Programming Device User's Manual (Manual No.: SIEP C880700 05)*.

(2) Deleting the 218IFB Transmission Definitions Data

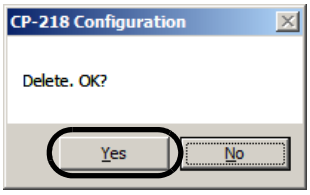
Use the following procedure to delete all of the 218IFB transmission definitions data. In Online Mode, the definitions data is deleted from the MP2000 Series Machine Controller and the personal computer running the MPE720. In Offline Mode, the definitions data is deleted from the personal computer running the MPE720.

1. Select ***File - Delete*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

2. Click the **Yes** Button.



5.5 Ethernet Communication

5.5.1 Overview

The 218IF-02 Module is used to connect the MP2000 Series Machine Controller to Ethernet devices. The 218IF-02 Module is equipped with one 100Base-TX Ethernet port that conforms to IEEE802.3u and can be easily connected to a controller manufactured by other companies, a personal computer, or other types of computers. By connecting a computer running the MPE720, engineering of the MP2000 Series Machine Controller from the computer is enabled.
By installing a 218IF-02 Module, the MP2000 Series Machine Controller can send and receive data through the Ethernet line. Ethernet communication can be used for message communication and engineering communication. Transmissions using Ethernet communication are performed by specifying the MSG-SND and MSG-RCV functions in drawing or function programs.
This section describes the 218IFB protocol for Ethernet communication, setting methods, and connection methods, and provides programming examples.

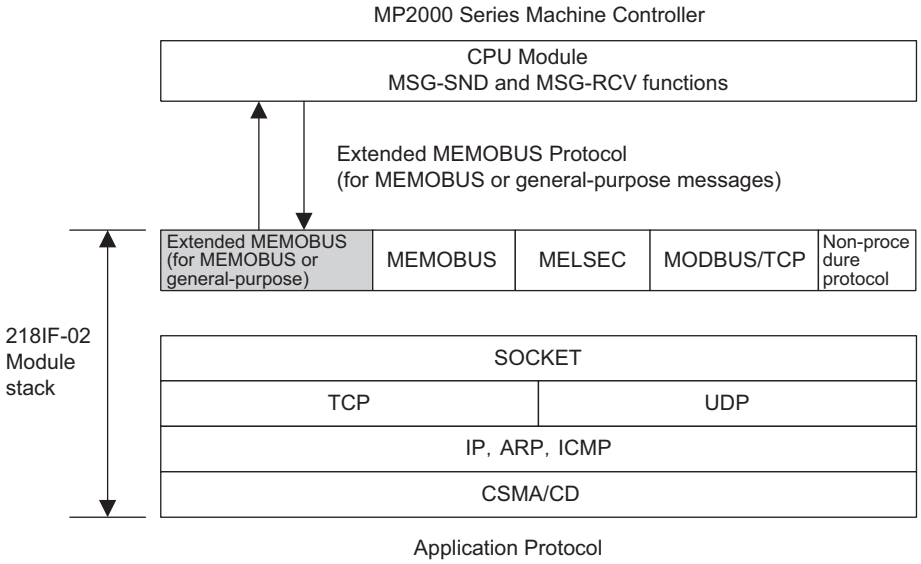
5.5.2 Ethernet Communication Protocols

The 218IF-02 Module supports the five types of protocol listed below in the TCP/IP application layer.

Protocol	Remarks	Master/Slave
MEMOBUS	Yaskawa's standard MEMOBUS protocol	Master/Slave
Extended MEMOBUS	Yaskawa's extended MEMOBUS protocol.	Master/Slave
MELSEC A-compatible 1E	PLC protocol manufactured by Mitsubishi Electric's General	Master/Slave
MELSEC QnA-compatible 3E	PLC protocol manufactured by Mitsubishi Electric's General	Master/Slave
MODBUS/TCP	Ethernet protocol, proposed by Modicon, used in industrial applications.	Master/Slave
OMRON FINS*	PLC protocol manufactured by OMRON Corporation	Master/Slave
TOYOPUC File memory system	PLC protocol manufactured by JTEKT Corporation	Master/Slave
Non-procedure protocol	A protocol provided to implement general-purpose message communication	Master/Slave

* Information on the relevant protocols is not contained in this manual. Refer to the *MP2300S/MP2310/MP2400 Basic Module Supplement for Ethernet Communications* (Manual No.: SIEP C880700 37) for details.

The Extended MEMOBUS protocol is used for all message communication between an MP2000 Series Machine Controller CPU Module and the 218IF-02 Module.
The following diagram illustrates this.



This section describes the communication protocols that can be used with the 218IFB.

5.5.3 MEMOBUS Protocol

(1) Overview

MEMOBUS transmissions use messages controlled by the MEMOBUS protocol. Data transmissions are performed in Master-Slave relationships in which the Master sends a message to a Slave and the Slave returns a response to the Master.

The Master communicates with one Slave at a time. To enable this, Slaves are assigned unique address numbers. The Master specifies the address number of a specific Slave when sending data.

On receiving a message from the Master, the Slave performs the specified function and returns a response to the Master.

Using these functions, the Master can read and modify the status of coils and contents of the registers in Slaves located at different places, enabling flexible centralized monitoring and control in a variety of production fields.

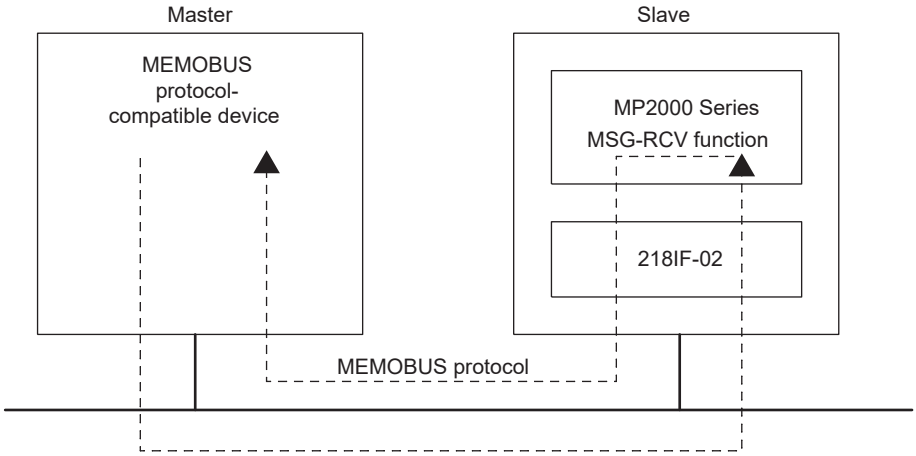


Fig. 5.1 Message Flow Using the MEMOBUS Protocol

(2) Function Codes in the MEMOBUS Protocol

The following table lists the function codes used in the MEMOBUS protocol. These functions are implemented by setting the function code in the MSG-SND function.

Function Code	Function	Data Address Setting Range	Data Size
01H	Read Coil Status	0 to 65535	1 to 2000 bits
02H	Read Input Relay Status	0 to 65535	1 to 2000 bits
03H	Read Holding Register Contents	0 to 65534	1 to 125 words
04H	Read Input Register Contents	0 to 32767	1 to 125 words
05H	Modify Status of a Single Coil	0 to 65535	–
06H	Write to a Single Holding Register	0 to 65534	–
08H	Loopback Test	–	–
0FH	Modify Status of Multiple Coils	0 to 65535	1 to 800 bits
10H	Write to Multiple Holding Registers	0 to 65534	1 to 100 words

5.5.4 Extended MEMOBUS Protocol

(1) Overview

The Extended MEMOBUS protocol is an enhanced version of the MEMOBUS protocol. It provides more versatile function codes and is capable of handling a larger data size. Data transmissions are performed in Master-Slave relationships in which the Master sends a message to a Slave and the Slave returns a response to the Master.
The Extended MEMOBUS protocol supports the MEMOBUS message communication mode and a general-purpose message communication mode. The mode that is used is specified in a parameter in the MSG-SND and MSG-RCV functions.

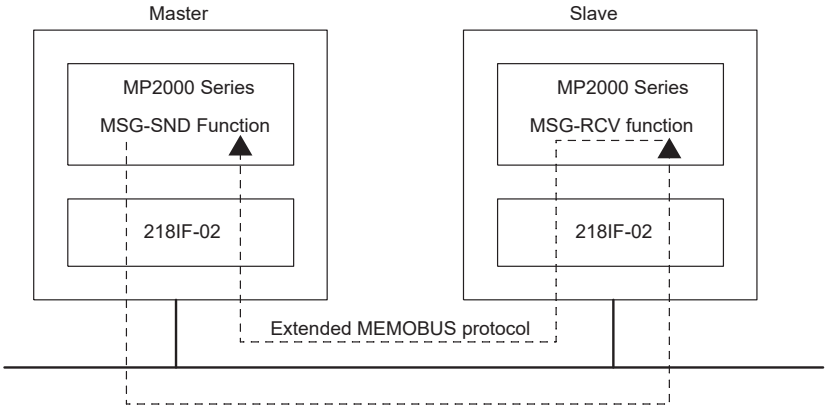


Fig. 5.2 Message Flow Using the Extended MEMOBUS Protocol

(2) Function Codes in the Extended MEMOBUS Protocol

The following table shows the function codes that can be used with the Extended MEMOBUS protocol. These functions are implemented by setting the function code in the MSG-SND function.

Function Code	Function	Data Address Setting Range	Data Size
01H	Read Coil Status	0 to 65535	1 to 2000 bits
02H	Read Input Relay Status	0 to 65535	1 to 2000 bits
03H	Read Holding Register Contents	0 to 65534	1 to 125 words
04H	Read Input Register Contents	0 to 32767	1 to 125 words
05H	Modify Status of a Single Coil	0 to 65535	–
06H	Write to a Single Holding Register	0 to 65534	–
08H	Loopback Test	–	–
09H	Read Holding Register Contents (Extended)	0 to 65534	1 to 2044 words (ASCII = 1020 words)
0AH	Read Input Register Contents (Extended)	0 to 32767	1 to 2044 words (ASCII = 1020 words)
0BH	Write to Holding Register (Extended)	0 to 65534	1 to 2043 words (ASCII = 1019 words)
0DH	Read Discontinuous Holding Register Contents (Extended)	0 to 65534	1 to 2044 words (ASCII = 1020 words)
0EH	Write to Discontinuous Holding Registers (Extended)	0 to 65534	1 to 1022 words (ASCII = 510 words)
0FH	Modify Status of Multiple Coils	0 to 65535	1 to 800 bits
10H	Write to Multiple Holding Registers	0 to 65534	1 to 100 words

- Maximum length of MEMOBUS command (MEMOBUS header + data size) is 512 words. However, expansion of the user message area allows up to 2048 words for some expansion functions.

5.5.5 MELSEC Protocol

(1) Overview

The MELSEC protocol is used for communication between an MP2000 Series Machine Controller and a MELSEC A-series Programmable Logic Controller manufactured by Mitsubishi Electric Corporation.
The MELSEC protocol is used to communicate between the MP2000 Series Machine Controller and a MELSEC Programmable Logic Controller. To achieve this, the 218IF-02 Module automatically converts the protocol to the MELSEC protocol. The MSG-SND and MSG-RCV functions can be used in a user program in the same way as with the MEMOBUS protocol.
The 218IF-02 Ethernet transmission system supports the following three transmissions functions of the MELSEC A-series Programmable Logic Controllers.

[a] Communication between CPUs

The MSG-SND function can be used to read and write data in the CPU of the MELSEC Programmable Logic Controllers. The A-compatible 1E frame format is used and a subset of the A-compatible 1E commands is supported. In communication between CPUs, the MP2000 Series Machine Controller can only be a Master due to restrictions of the specifications of MELSEC A-series Programmable Logic Controllers.

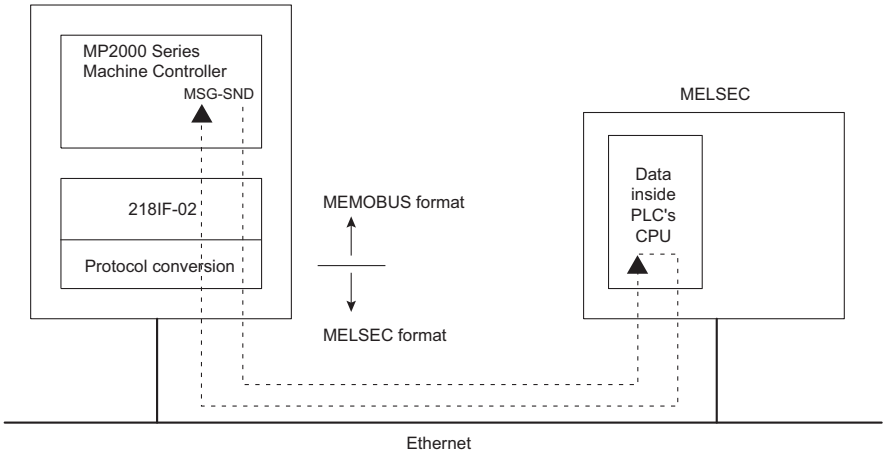


Fig. 5.3 Message Flow for Communication between CPUs

[b] Fixed Buffer Communication

The MSG-SND function can be used to write the contents of M registers to the fixed buffers in the MELSEC Ethernet Module. The MSG-RCV function can be used to receive data sent to the MP2000 Series Machine Controller from the fixed buffers and store it in M registers.

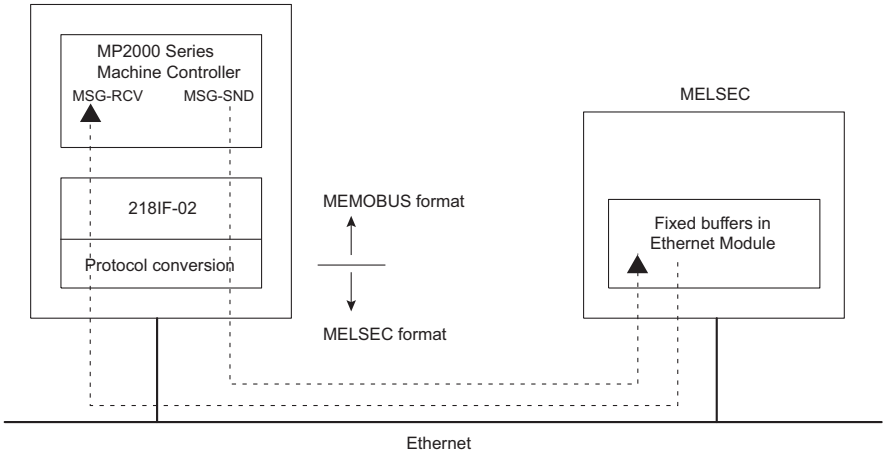


Fig. 5.4 Message Flow for Fixed Buffer Communication

[c] Random-access Buffer Communication

The MSG-SND function can be used to read and write the random-access buffers in the MELSEC Ethernet Module. The contents of M registers can be written to the random-access buffers, and the contents read from the random-access buffers can be stored in M registers.
In random-access buffer communication, the MP2000 Series Machine Controller can only be a Master due to restrictions of the specifications of MELSEC A-series Programmable Logic Controllers.

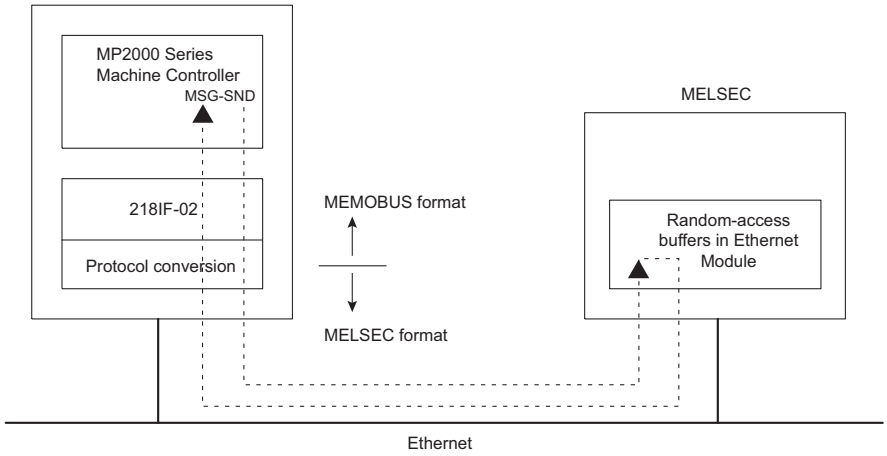


Fig. 5.5 Message Flow for Random-access Buffer Communication

(2) Command Function Code List

The following table lists the common MELSEC ACPU commands and MEMOBUS function codes supported by the 218IF. To communicate with a MELSEC A-series Programmable Logic Controller, set the MEMOBUS function code that corresponds to the MELSEC ACPU command in the MSG-SND function.

Common MELSEC ACPU Command	MEMOBUS Function Code	Function	Number of Words
00H	01H/02H	Reads data from bit-type devices in 1-point units.	256 points
01H	03H/04H/09H/0AH	Reads data from word-type devices in 1-point units.	256 points
02H	05H/0FH	Writes data to bit-type devices in 1-point units.	256 points
03H	06H/0BH/10H	Writes data to word-type devices in 1-point units.	256 points
16H	08H	Executes a loopback test.	256 bytes
05H	0EH	Sets/resets data in word devices specified at random in 1-point units.	40 points
60H	31H	Fixed Buffer Communication	1017 words (ASCII = 508 words)
61H	32H	Random-access buffer read	1017 words (ASCII = 508 words)
62H	33H	Random-access buffer write	1017 words (ASCII = 508 words)

- The special commands for the AnCPU are not supported. Use the common ACPU commands listed above when accessing an AnCPU. Accessing the extended file registers in an AnCPU is not possible.

(3) MELSEC Devices

The following tables list the common ACPU command device ranges, MEMOBUS commands, and MP2000 Series Controller register numbers for MELSEC bit-type and word devices.
Use these tables as a basis for setting the parameters for the MSG-SND function when reading and writing MELSEC devices from an MP2000 Series Controller.

- Even for the same device ranges, the device area ranges depend on the MELSEC Sequencer. Refer to the MELSEC manuals for details.

■ MELSEC Bit Devices

Device	Common ACPU Command Device Range	Decimal/ Hex	MEMOBUS Commands	Start No.	Register No.
X	X0000 to X07FF	Hex	02H: Input relays	0 to 2047	MB000000 to MB00127F
Y	Y0000 to Y07FF	Hex	01H/0FH: Coils	0 to 2047	MB000000 to MB00127F
M	M0000 to M2047	Decimal	01H/05H/0FH: Coils	2048 to 4095	MB001280 to MB00255F
M	M9000 to M9255	Decimal	01H/05H/0FH: Coils	4096 to 4351	MB002560 to MB00271F
B	B0000 to B03FF	Hex	01H/05H/0FH: Coils	4352 to 5375	MB002720 to MB00335F
F	F0000 to F0255	Decimal	01H/05H/0FH: Coils	5376 to 5631	MB003360 to MB00351F
TS	TS000 to TS255	Decimal	02H: Input relays	2048 to 2303	MB001280 to MB00143F
TC	TC000 to TC255	Decimal	02H: Input relays	2304 to 2559	MB001440 to MB00159F
CS	CS000 to CS255	Decimal	02H: Input relays	2560 to 2815	MB001660 to MB00175F
CC	CC000 to CC255	Decimal	02H: Input relays	2816 to 3071	MB001760 to MB00191F
M	M2048 to M8191	Decimal	01H/05H/0FH: Coils	8192 to 14335	MB005120 to MB00895F

■ MELSEC Word Devices

Device	Common ACPU Command Device Range	Decimal/ Hex	MEMOBUS Commands	Start No.	Register No.
TN	TN0000 to TN255	Decimal	04H/0AH: Input registers	0 to 255	MW00000 to MW00255
CN	CN0000 to CN255	Decimal	04H/0AH: Input registers	256 to 511	MW00256 to MW00511
D	D0000 to D1023	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	0 to 1023	MW00000 to MW01023
D (Special)	D9000 to D9255	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	1024 to 1279	MW01024 to MW01279
W	W0000 to W03FF	Hex	03H/06H/09H/0BH/0EH/ 10H: Holding registers	1280 to 2303	MW01280 to MW02303
R	R0000 to R8191	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	2304 to 10495	MW02304 to MW10495
D	D1024 to D6143	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	10496 to 15615	MW10496 to MW15615

The MP2000 Series Machine Controller register addresses can be adjusted by using the offset setting in the MSG-SND function, as shown in the following example.

Example: Writing 20 Words to MELSEC Data Registers (from D1030)

From the table for MELSEC word devices, the register number for device D1030 is MW10502, the starting number is 10502, and the MEMOBUS command to write the data is 10H.
These values are set in the parameters of the MSG-SND function.

- PARAM04: Function Code = 10H
- PARAM05: Data address = 10502
- PARAM06: Data size = 20 (words)

The data address in PARAM05 serves as both the address to write and the source address. The above settings thus read 20 words of data starting from MW10502 and write it to the 20 words starting from D1030.
To write data from the register at a location other than MW10502, the holding register offset in PARAM11 is used to adjust the source location.
Example:

- PARAM11: Holding register offset = 10000

If the holding register offset in PARAM11 is set to 10000, the 20 words of data starting at MW20502 (i.e., MW10502 + 10000) will be written to 20 words starting at D1030. Negative offsets cannot be used.

⬩ Refer to *5.7 Message Send Function* on page 5-48 for details on message communication functions.

(4) MP2000 Series Machine Controller/MELSEC Register Map

The following register map is based on the device table in *(3) MELSEC Devices* on page 5-28 and shows the MP2000 Series Machine Controller M registers that correspond to each MELSEC device. Holding registers, input registers, input relays, and coils are assigned for each MELSEC device so that the MP2000 Series Machine Controller can use MEMOBUS commands as an interface to read and write data. Data read from a MELSEC device is stored in the corresponding M registers shown in the following map. Data written to the MELSEC devices is written from the corresponding device M registers shown in the following table.

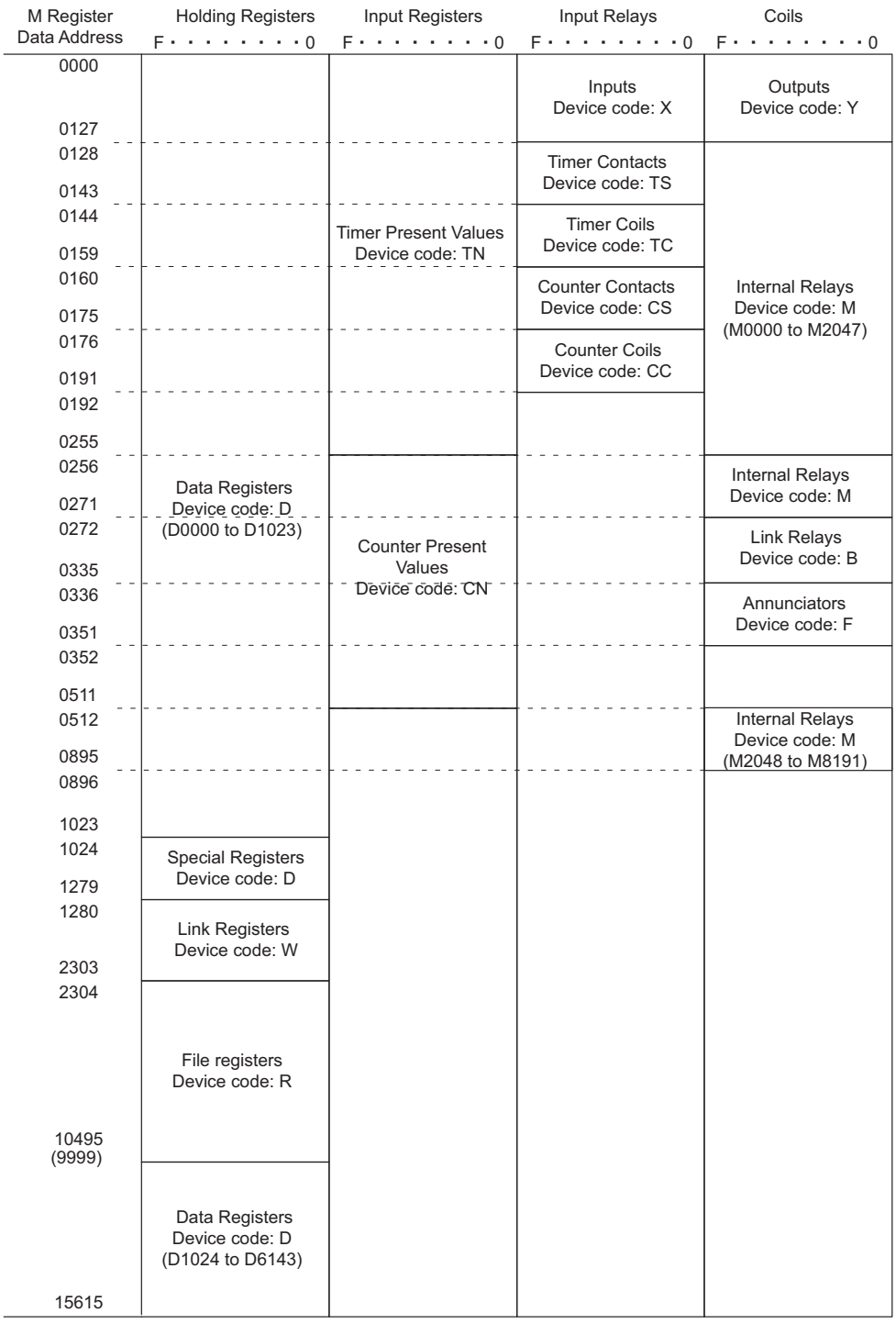


Fig. 5.6 Register Map

- This map is for when the offset parameters in the MSG-SND function are set to 0. An offset parameter (i.e., holding register offset, input register offset, input relay offset, or coil offset) can be set to shift the M register numbers given in this map backward by a relative amount.

(5) Precaution on Data Sizes When Using the MELSEC Protocol

The following table lists restrictions in data sizes when using the MELSEC protocol.

MELSEC Protocol		Maximum data sizes when MTU = 1500	
		UDP	TCP
Communication between CPUs		There are no restrictions for binary or ASCII data within the applicable ranges.	There are no restrictions for binary or ASCII data within the applicable ranges.
Fixed Buffer Communication		Binary = 1017 words ASCII = 508 words	Binary = 727 words ASCII = 362 words
Random-access buffer communication	Writing data	Binary = 1017 words ASCII = 508 words	Binary = 726 words ASCII = 361 words
	Reading data	Binary = 1017 words ASCII = 508 words	Binary = 728 words ASCII = 363 words

- The maximum data size for TCP is the maximum size that can be sent with one segment. A "segment" is a unit of data transfer for TCP. It is determined by the MTU (maximum transfer unit).
- If the maximum data size is exceeded, the data will be split into more than one segment, but the 218IF may not be able to rebuild data separated into more than one segment. Do not specify more data than can be sent with one segment.

5.5.6 MODBUS/TCP Protocol

(1) Overview

The MODBUS/TCP protocol enables communication with devices of other manufacturers that support the MODBUS/TCP protocol.
In communication between an MP2000 Series Machine Controller and a device that supports the MODBUS/TCP, the 218IF-02 Module automatically converts the protocol to the MODBUS/TCP protocol. The MSG-SND and MSG-RCV functions can be used in a user program in the same way as with the MEMOBUS protocol.

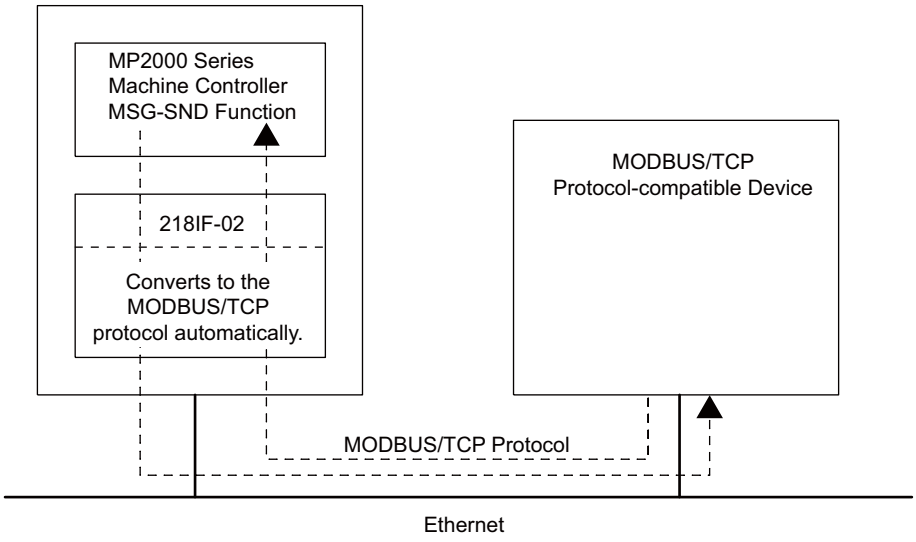


Fig. 5.7 Message Flow between MP2000 Series Machine Controller and MODBUS/TCP Protocol-compatible Device Using MODBUS/TCP Protocol

(2) Function Codes

The following table shows the function codes used in the MODBUS/TCP protocol. These functions are implemented by setting the function code in the MSG-SND function.

Function Code	Function	Data Address Setting Range	Data Size
01H	Read Coil Status	0 to 65535	1 to 2000 bits
02H	Read Input Relay Status	0 to 65535	1 to 2000 bits
03H	Read Holding Register Contents	0 to 65534	1 to 125 words
04H	Read Input Register Contents	0 to 32767	1 to 125 words
05H	Modify Status of a Single Coil	0 to 65535	–
06H	Write to a Single Holding Register	0 to 65534	–
0FH	Modify Status of Multiple Coils	0 to 65535	1 to 800 bits
10H	Write to Multiple Holding Registers	0 to 65534	1 to 100 words
16H	Mask Write to Holding Registers	0 to 65534	–
17H	Read/Write to Multiple Holding Registers	0 to 65534	Read: 1 to 125 words Write: 1 to 100 words

5.5.7 Non-Procedure Protocol

(1) Overview

The non-procedure protocol enables message communication with devices from other manufacturers using any protocol. Data in consecutive M registers in the MP2000 Series Machine Controller is sent and received without modification.
The MSG-SND function sends the data from the specified M registers without modification. There is no response.
The MSG-RCV function stores the received data without modification in the M registers specified in the MSG-RCV function. It does not return a response.
Thus, the non-procedure protocol sends data in just one direction without a response.
The non-procedure protocol is not based on command-response communication. It is thus necessary to make a user program in which sending and receiving are switched to perform command-response communication.
The size of data that can be sent or received in non-procedure communication mode is from 1 to 510 words.

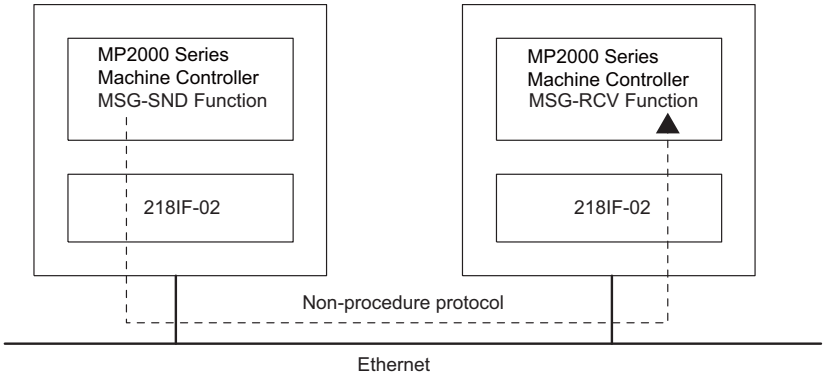
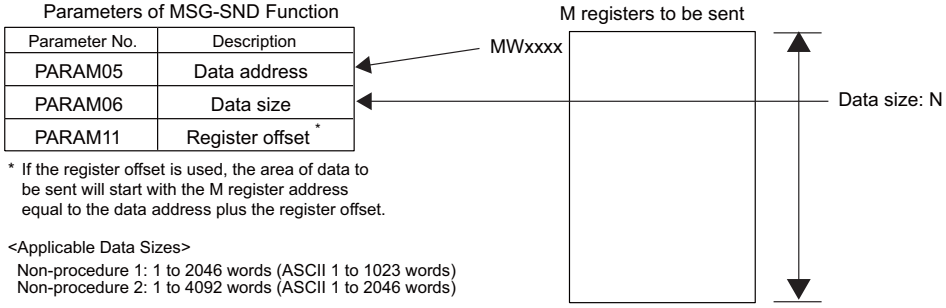
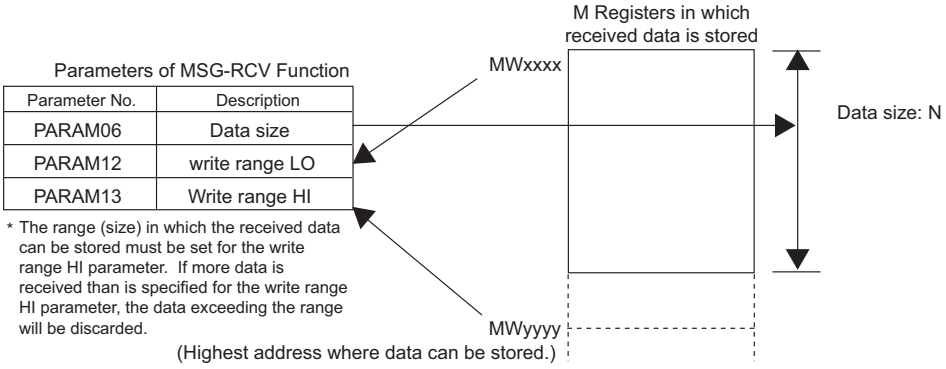


Fig. 5.8 Message Flow Using the Non-procedure Protocol

With the non-procedure protocol, the address of the first M register and the data size are specified in the MSG-SND function to enable sending the contents of M registers.



With the MSG-RCV function, write range LO and write range HI can be set to store the received data in the specified M registers.



- For both the MSG-SND and MSG-RCV functions, set the connection number to use for communication in PARAM02.

■ Precautions on Using TCP with Non-procedure Communication

- To avoid complicated byte stream processing when using TCP with non-procedure communication, use one of the following methods so that the remote device does not consecutively send TCP packets.
  - Increase the packet send interval (recommendation: 1 s or longer).
  - Create a command-response application protocol.
- Set the system so that the remote device does not separate send data into packets. Design the system so that the remote device does not send data separated into packets.
  - TCP is a byte-stream protocol. If data is sent continuously, the number of data blocks sent in one packet will depend on the send timing because the data blocks that are sent may be combined at the sending device. Also, the number of data blocks that is received will depend on the receive timing because the data blocks sent in multiple packets may be combined at the receive device. Combining data blocks at the sending device and receive device is performed in the transport layer* (TCP) and network layer* (IP) of the protocol stack of the 218IF-02 Module. Therefore, it is not possible to determine how many items of data will be sent using non-procedure communication without a field that indicates the send data length.

■ Precautions on Using TCP and UDP with Non-procedure Communication

If data is continuously received using the same connection, set a data send interval for remote devices that is longer than the cycle for which the MSG-RCV function can read data into the user application.

- If the 218IF-02 Module receives the next data before reading the data into the user application using the MSG-RCV function, the old data will be overwritten with the most recent data. This is because there is only one reception buffer per connection in the protocol stack application layer* (non-procedure) in the 218IF-02 Module.
- For the MSG-RCV function, at least two scans are required after data reception has been completed until the next time data reception starts. Asynchronous processing is performed between the 218IF-02 Module and the MSG-RCV function, and so at least two scans are required if the scan cycle is shorter than the operation cycle of the 218IF-02 Module. As a rough guide, set a data send interval of at least two scans of the execution scan cycle of the MSG-RCV function at the sending device if the execution scan cycle of the MSG-RCV function is 2 ms or longer. Set a data send interval of 4 ms or longer at the sending device if the execution scan cycle of the MSG-RCV function is less than 2 ms. The operation cycle of the 218IF-02 Module depends on the communication load of the Module. If the communication load is large, adjust the data send interval to an appropriate time.

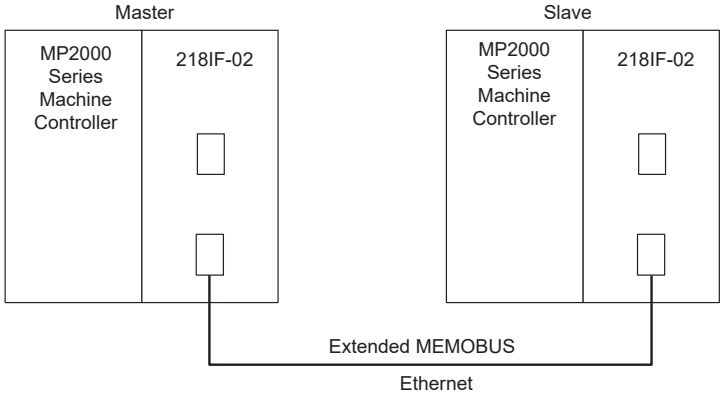
* Seven Layers for OSI Reference Model

5.6 Connection Examples for Ethernet Communication

This section shows examples of connecting to the Ethernet port of the 218IF-02.

5.6.1 Connecting 218IF-02 Modules

The 218IF-02 Modules are mounted on MP2000 Series Machine Controllers. The MP2000 Series Machine Controller that operates as the Master reads data from the other MP2000 Series Machine Controller that operates as a Slave using Ethernet. The setup is shown below.

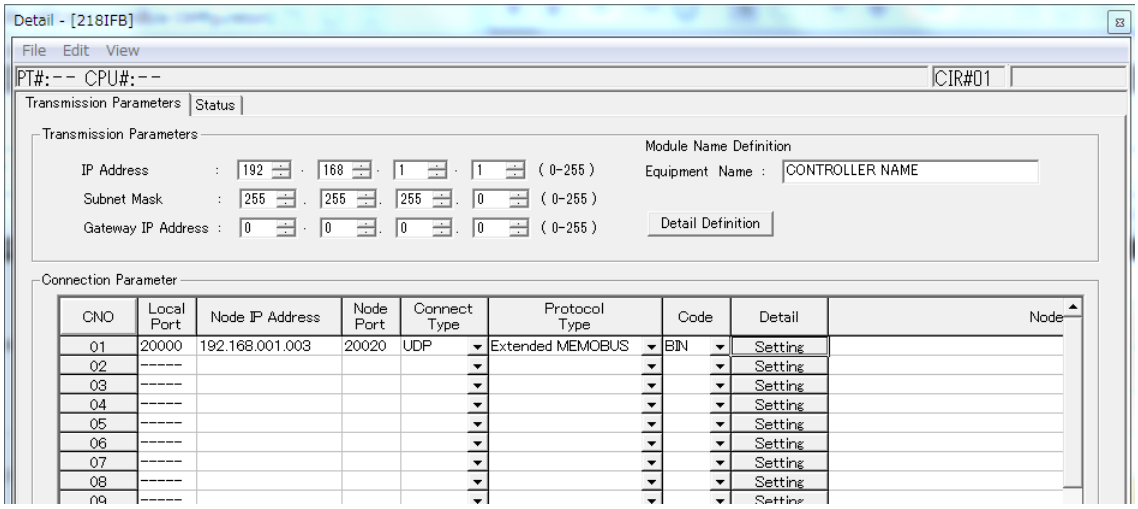


The IP address settings of the Master and Slave are shown below.

Item	Master	Slave
IP Address	192.168.001.002	192.168.001.003
Subnet Mask	255.255.255.000	255.255.255.000
Gateway IP Address	000.000.000.000	000.000.000.000

(1) 218IFB Transmission Settings for the Master

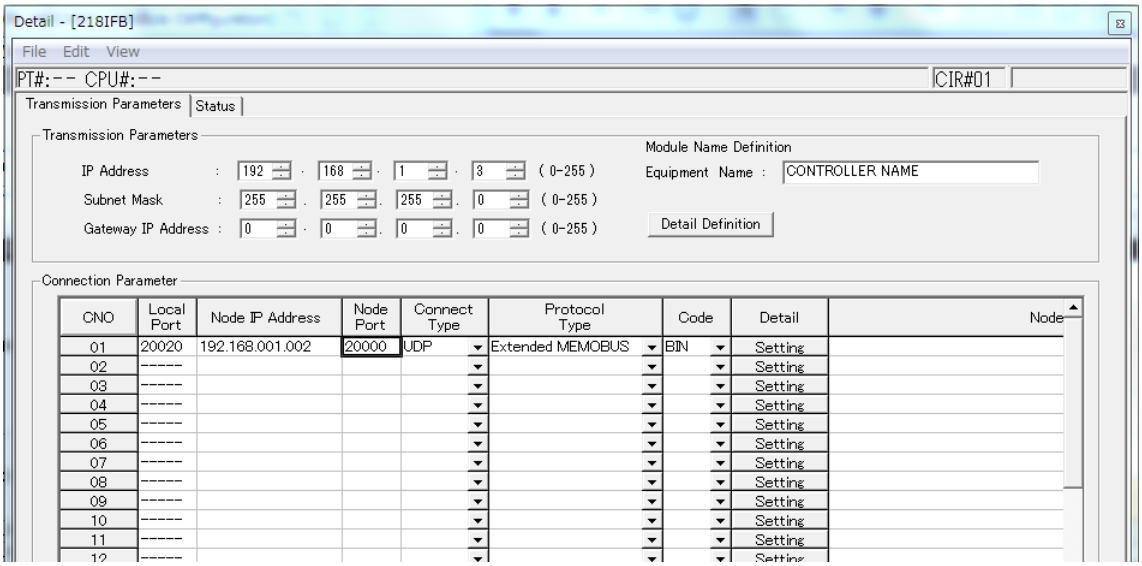
The 218IFB transmission settings for the Master are shown below.



Local Port: Set the port number of the MP2000 Series Machine Controller operating as the Master.
Node IP Address: Set the IP address of the MP2000 Series Machine Controller operating as a Slave.
Node Port: Set the port number of the MP2000 Series Machine Controller operating as a Slave.

(2) 218IFB Transmission Settings for the Slave

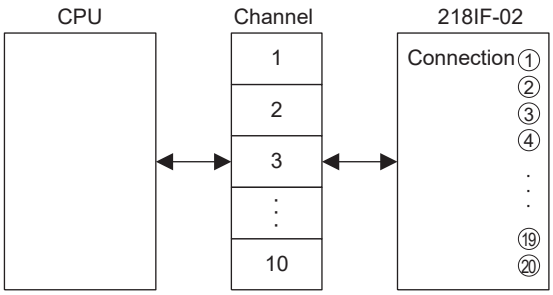
The 218IFB transmission settings for the Slave are shown below.



Local Port: Set the port number of the MP2000 Series Machine Controller operating as a Slave.
Node IP Address: Set the IP address of the MP2000 Series Machine Controller operating as the Master.
Node Port: Set the port number of the MP2000 Series Machine Controller operating as the Master.

(3) Programming Precaution When Communicating through More Than 10 Connections

Although the maximum number of connections that can be set for a 218IFB transmission system is 20, simultaneous communication is allowed for only up to 10 connections.
This is because only 10 channels are provided for communication between the CPU and a 218IF-02 Module and these channels are occupied until the MSG-SND or MSG-RCV function has been completed. When the 218IF transmission system is used with 11 stations or more, one channel must be used by two stations and the communication designation must be switched using the ladder program so that the 218IF transmission system communicates with the second station only after it has finished communicating with the first station. In programming cases like this, the execute input for the MSG-SND and MSG-RCV functions must not be always ON.



(4) Procedure to Start Communication

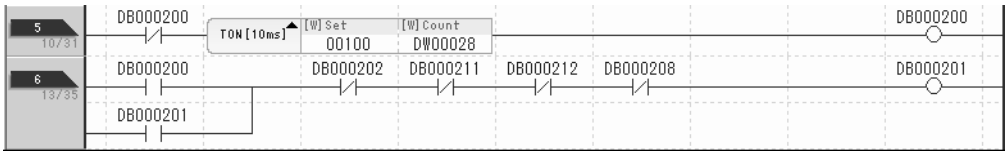
Use the following procedure to start communication.

1. Start executing the MSG-RCV function in the Slave 218IFB Module.
   - Always execute the MSG-RCV function first.
2. Turn ON the **Execute** parameter in the MSG-SND function in the Master 218IFB Module to send a message.

Example ladder programming using the MSG-SND and MSG-RCV functions is shown below in item (5).
In *[b] Ladder Programming Example Using the Message Receive Function (MSG-RCV)* on page 5-39, MSG-RCV function execution is started with SB000004 (Always ON Coil). Execution of the MSG-RCV function will thus start as soon as the power is turned ON to the Slave MP2000 Series Machine Controller. In *[a] Ladder Programming Example Using the Message Send Function (MSG-SND)* on page 5-38, a message will be sent as soon as DB000201 (the register set for **Execute** in the MSG-SND function) turns ON, thus starting communication with the Slave 218IFB Module.

I/O Definitions	No.	Name	Setting Example	Description
Inputs	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

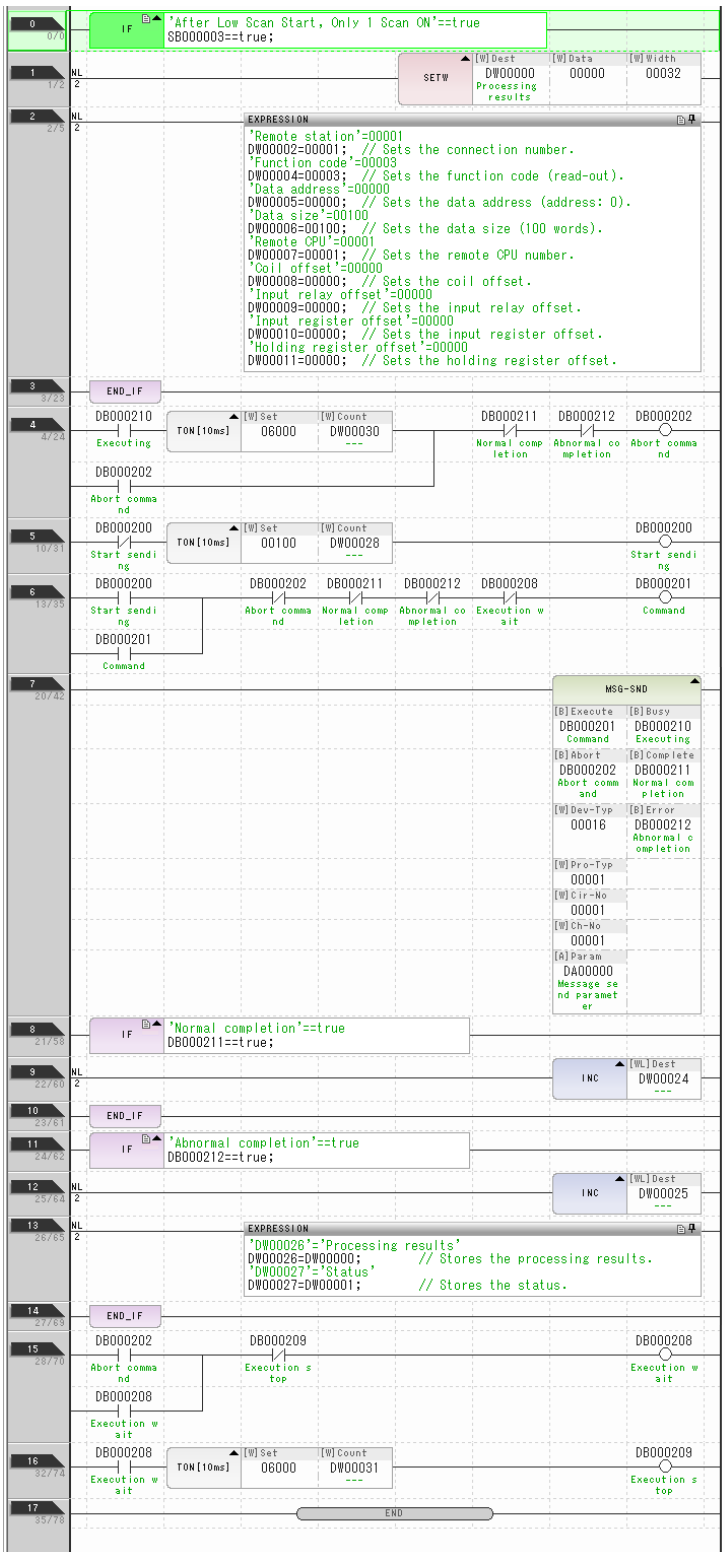
- In *[a] Ladder Programming Example Using the Message Send Function (MSG-SND)* on page 5-38, the programming is written so that a message will be sent every second after one second has expired after the low-speed (or high-speed) scan has started. To change the interval at which messages are sent, change the set value of the timer shown below.



(5) Programming Example

An example of ladder programming using the Message Send and Message Receive function for 218IFB transmissions is shown below.

[a] Ladder Programming Example Using the Message Send Function (MSG-SND)



Sets parameters in the first scan after startup. (Low-speed scan (DWG.L): SB000003, High-speed scan (DWG.H): SB000001)

Clears DW register to zero.

Sets PARAM02 to PARAM11.

Forced stop if processing is not completed within 60 seconds after sending the Send command.

Starts sending every 1 second.

Function Send command

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 1 for extended MEMOBUS or MELSEC, 2 for non-procedure
Cir-No. (circuit number): 2 (connection port)
Ch-No. (channel number): 1 (always 1 for the 218FB)

Turns ON after the command message is sent and the response message is received.

Increments the normal pass counter.

Ended with transmission errors.

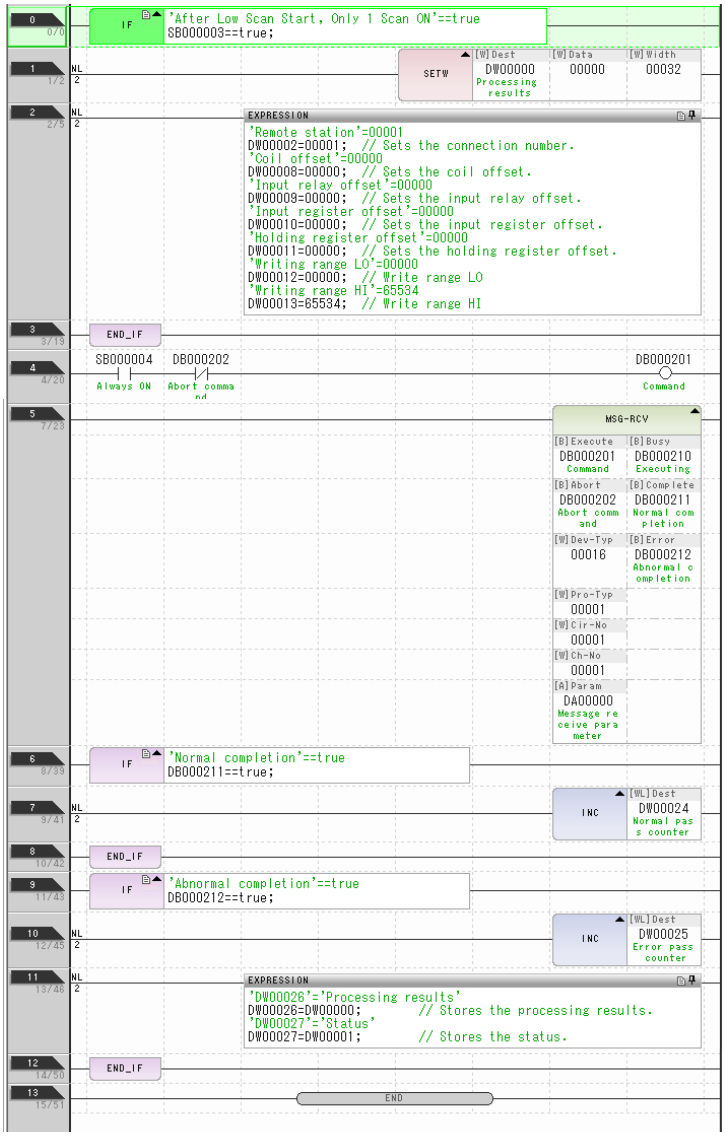
Increments the abnormal pass counter.

Stores the processing results and the status.

Function Send command

Enables sending the Send command when 60 seconds elapses.

[b] Ladder Programming Example Using the Message Receive Function (MSG-RCV)



Sets parameters in the first scan after startup.
(Low-speed scan (DWG.L): SB000003,
High-speed scan (DWG.H): SB000001)

Clears DW register to zero.

Sets PARAM07 to PARAM13.

Starts receiving.

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 1 for extended MEMOBUS or MELSEC, 2 for non-procedure
Cir-No. (circuit number): 2 (connection port)
Ch-No. (channel number): 1 (always 1 for the 218FB)

The normal completion coil turns ON at normal completion.

Increments the normal pass counter.

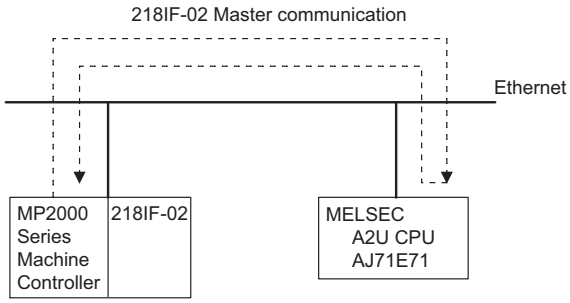
Ended with transmission errors.

Increments the abnormal pass counter.

Stores the processing results and the status.

5.6.2 Connection to MELSEC

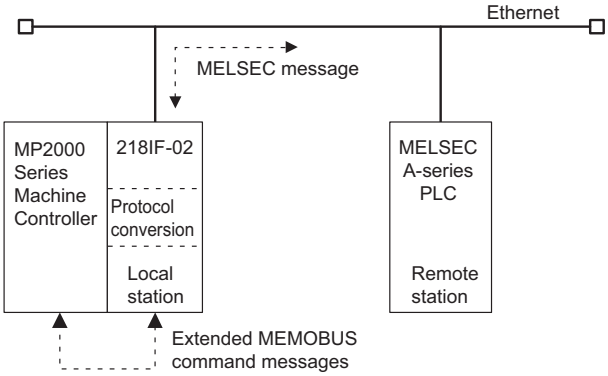
This section describes the connection example shown below in which the MP2000 Series Machine Controller is connected to a MELSEC A-series Programmable Logic Controller through Ethernet using a 218IF-02 Module.



- The 218IF-02 Module is set as the Master and the MELSEC as a Slave.

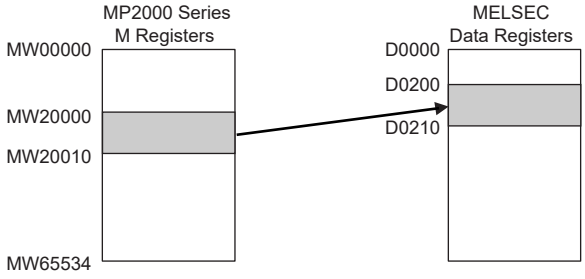
(1) Message Flow

In the above example, messages are exchanged between the MP2000 Series Machine Controller and 218IF-02 Module using Extended MEMOBUS communication and between the 218IF-02 and the MELSEC-A PLC using MELSEC communication.



- MELSEC communication is checked with a combination of the A2U CPU and AJ71E71.

In this example, the MELSEC protocol is used for communication between CPUs. The 218IF-02 on the MP2000 Series Machine Controller at the Master reads the contents of 10 words starting from MW20000 at the MELSEC Slave and writes the data to 10 words starting at D0200.

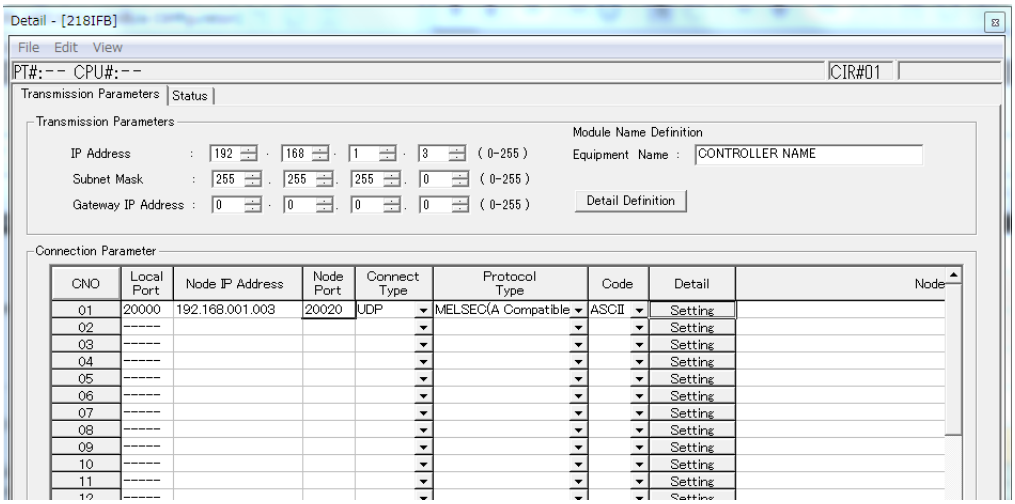


The following settings are used.

Item	MP2000 Series Machine Controller (Master)	MELSEC (Slave)
IP Address	192.168.001.002	192.168.001.003
Port Number	20000	20020
Subnet Mask	255.255.255.000	255.255.255.000
Gateway IP Address	000.000.000.000	000.000.000.000
Communication Protocols	UDP	UDP
Application Protocol	MELSEC	–

(2) 218IFB Transmission Settings for the Master

The 218IFB transmission settings for the Master are shown below.



Local Port: Set the port number of the MP2000 Series Machine Controller operating as the Master.
Node IP Address: Set the IP address of the MELSEC operating as a Slave.
Node Port: Set the port number of the MELSEC operating as a Slave.

(3) MELSEC Slave Settings

When using a MELSEC A-series Ethernet Module (AJ71E71), ladder programming to initialize the IP address, port number, and other settings must be created. To create the initializing ladder program, refer to the procedure for communicating with other nodes explained in the MELSEC manual.
When using a MELSEC Q-series Ethernet Module (QJ71E71-100), set the IP address and port number in the Ethernet setting window.

(4) Programming in the MP2000 Series Machine Controller

The MP2000 Series Machine Controller performs communication with a MELSEC A-series Programmable Logic Controller by converting messages from the Extended MEMOBUS protocol to the MELSEC protocol. Accordingly, it is not necessary to be aware of the MELSEC message structure in user application programs. All the user must do is to set MELSEC as the protocol in the connection parameters for the 218IF-02 transmission system and enter the MSG-SND function in the program to easily communicate with the MELSEC Controller.
The commands used in the MSG-SND function are the MEMOBUS commands that correspond to MELSEC commands. Refer to *5.5.5 (2) Command Function Code List* on page 5-28 for the MEMOBUS commands.
For example, the MEMOBUS command that corresponds to the MELSEC command 00 (to read bit devices in 1-point units) is 01 or 02.

(5) Procedure to Start Communication

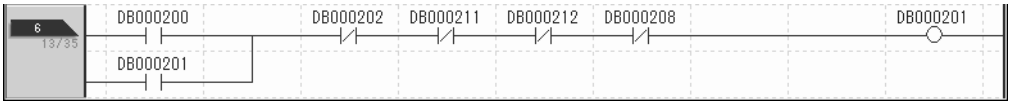
Use the following procedure to start communication.

1. Start the MELSEC Slave.
2. Turn ON the **Execute** parameter in the MSG-SND function in the Master 218IF Module to send a message.

Example ladder programming using the MSG-SND function is shown in *(6) Programming Example* on page 5-43. In this programming example, a message is sent when the register set for **Execute** in the MSG-SND function (DB000201) turns ON to start communication with the MELSEC Slave.

I/O Definitions	No.	Name	Setting Example	Description
Inputs	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

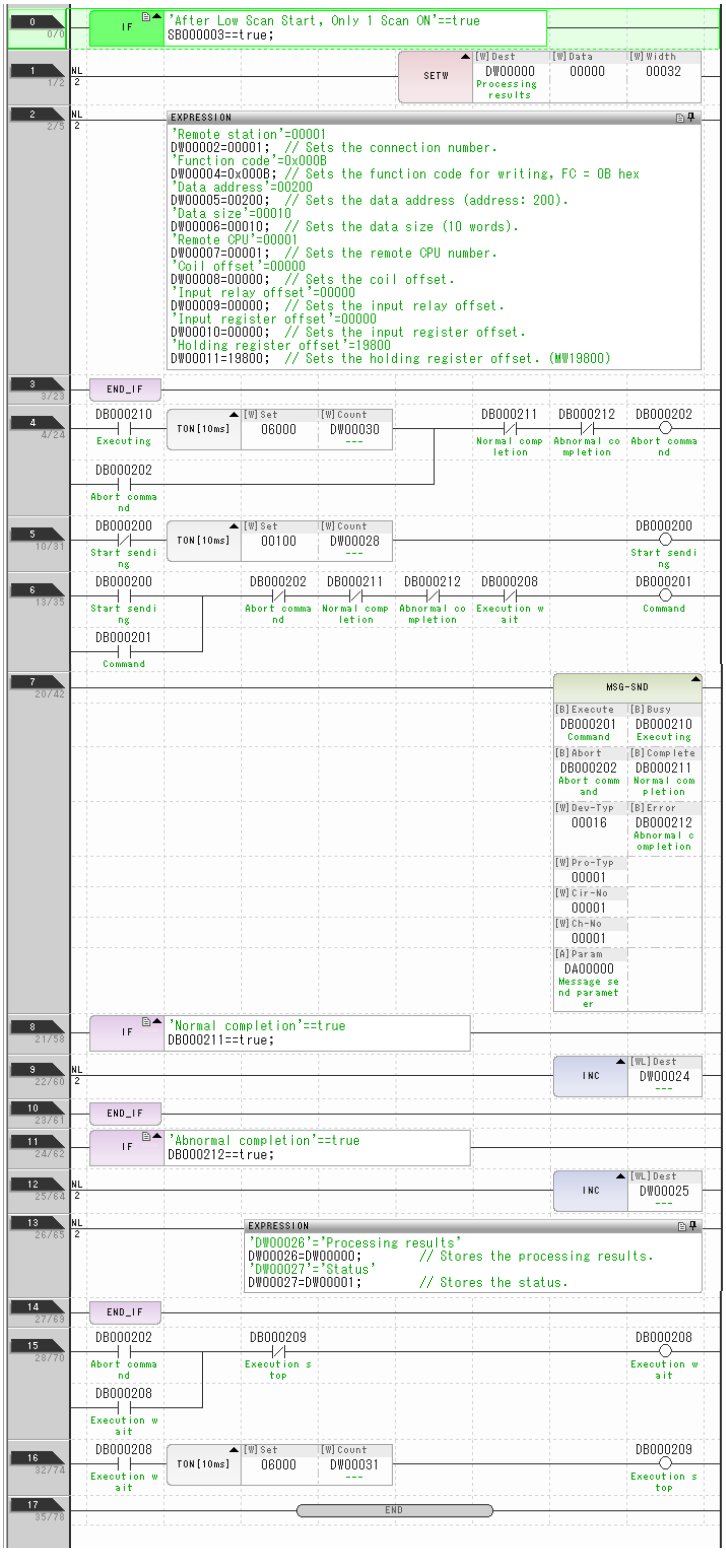
* In *(6) Programming Example* on page 5-43, the programming is written so that a message will be sent every second after one second has expired after the low-speed (or high-speed) scan has started. To change the interval at which messages are sent, change the set value of the timer shown below.



(6) Programming Example

An example of ladder programming using the Message Send function for 218IFB transmissions is shown below.

- For communication between CPUs using the MELSEC protocol, ladder programming for message communication is not required in the MELSEC PLC.



Sets parameters in the first scan after startup. (Low-speed scan (DWG.L): SB000003, High-speed scan (DWG.H): SB000001)

Clears DW register to zero.

Sets PARAM02 to PARAM11.

Forced stop if processing is not completed within 60 seconds after sending the Send command.

Starts sending every 1 second.

Function Send command

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 1 for extended MEMOBUS or MELSEC, 2 for non-procedure
Cir-No. (circuit number): 2 (connection port)
Ch-No. (channel number): 1 (always 1 for the 218FB)

Turns ON after the command message is sent and the response message is received.

Increments the normal pass counter.

Increments the abnormal pass counter.

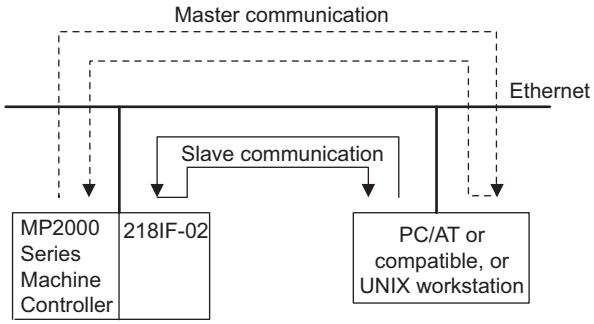
Stores the processing results and the status.

Function Send command

Enables sending the Send command when 60 seconds elapses.

5.6.3 Connection to a Personal Computer

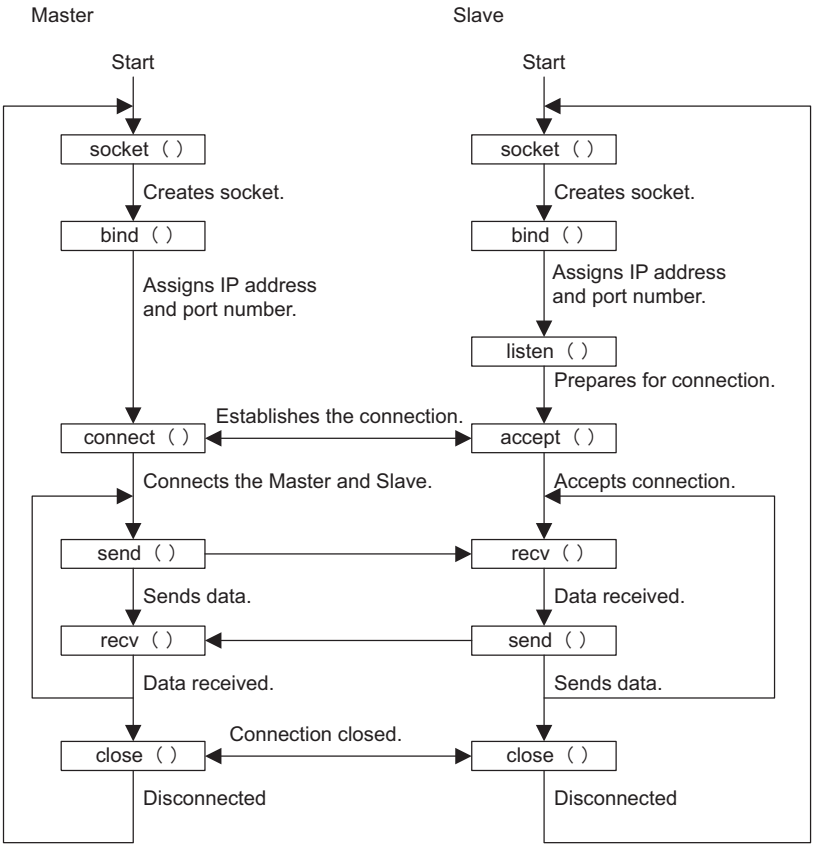
This section describes the network configuration in which a PC/AT or compatible computer or a UNIX workstation is connected to the 218IF-02 Module through an Ethernet interface of the computer or the workstation.
Original applications can be developed using the socket function in the computer or other device.



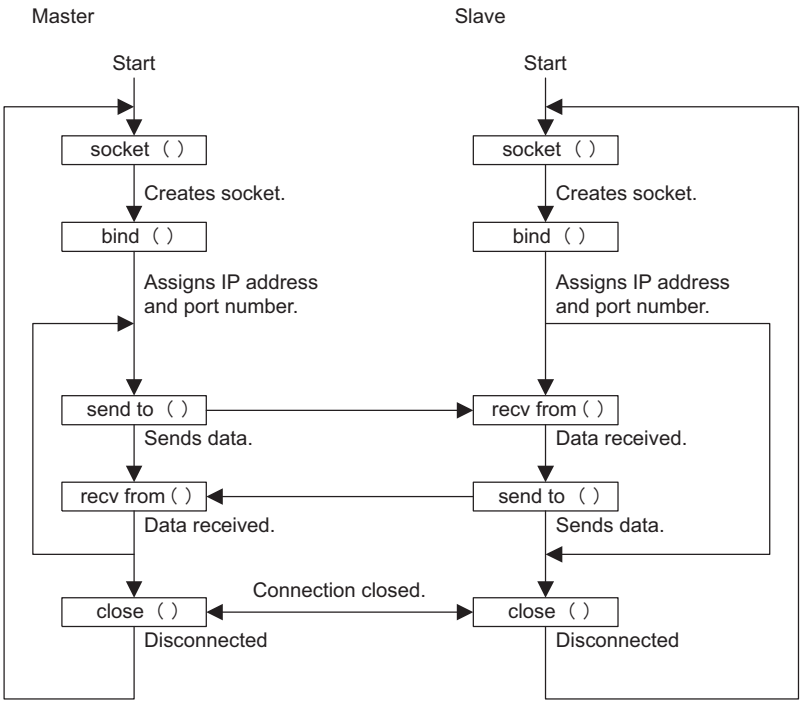
(1) Socket Communication Flow

The following diagram shows the logical programming flow using a socket interface.
When programming the computer using socket function, program according to the correct sequence depending on whether you are programming the Master or a Slave.
If execution of the MSG-SND function is started in the Master and execution of the MSG-RCV function is started in the Slave, the 218IF-02 will automatically perform the sequences for the Master and Slave.

[a] TCP

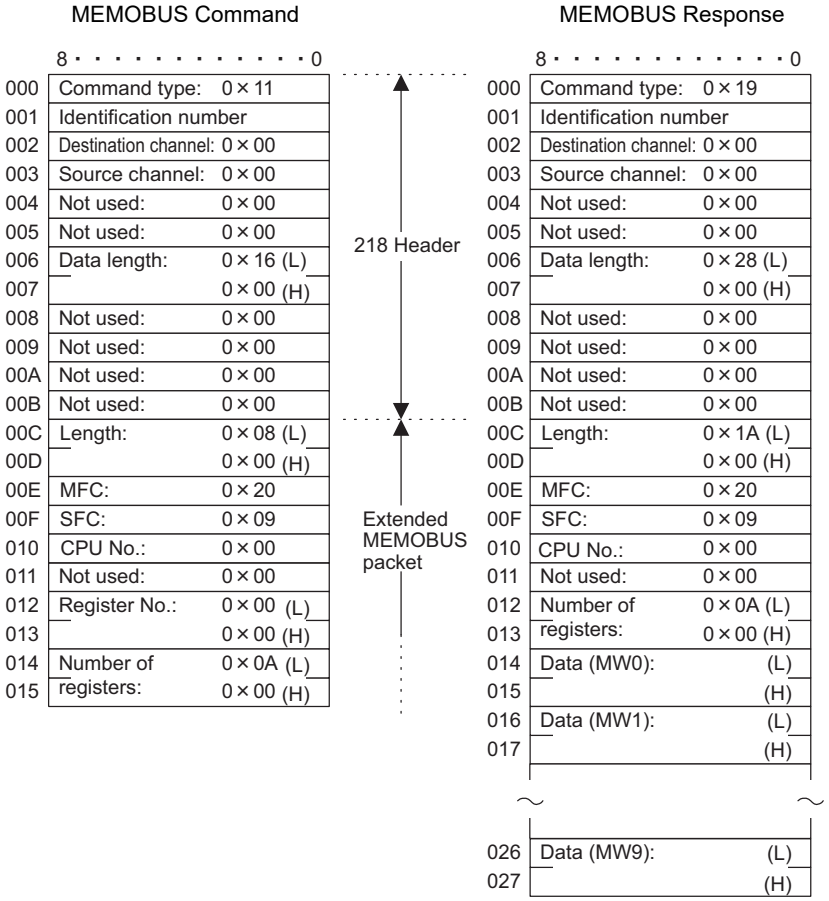


[b] UDP


Master
Slave
Start
Start
socket ( )
socket ( )
Creates socket.
Creates socket.
bind ( )
bind ( )
Assigns IP address
and port number.
Assigns IP address
and port number.
send to ( )
recv from ( )
Sends data.
Data received.
recv from ( )
send to ( )
Data received.
Sends data.
Connection closed.
close ( )
close ( )
Disconnected
Disconnected

(2) Data Format Example

The following diagram shows the data format when 10 words of data in holding registers of addresses 0 to 9 are read using the function code 09 (Holding Register Read (Extended) command), which is one of the Extended MEMOBUS commands.



(3) Precautions on Creating Data Using a Personal Computer

[a] Using a Personal Computer as the Master

■ MEMOBUS Command Data

- Set the identification numbers each time communication is performed in the following increment data order: 0x00→0x01→0x02→....→0xff→0x00.
- Specify 0 for the destination channel.
- Specify 0 for the source channel.
- When the remote station has more than one CPU, set the CPU number as follows.

Bit 7 – 4	3 – 0
DST. CPU No.	Source CPU No.

- DST. CPU No. (Destination CPU No.): Set to 1 for the MP2000 Series Machine Controller.
  Source CPU No.: Set to 0.

■ MEMOBUS Response Data

- Copy the identification numbers set in the MEMOBUS command data.
- Copy the destination channel numbers set in the MEMOBUS command data.
- A Machine Controller sets a unique value for the source channel. The computer may disregard for this number.
- The upper-place 4 bits and lower-place 4 bits of the CPU number set by the MEMOBUS command data are reversed for the CPU number. If the CPU number is not required at the computer, it would be disregarded.

[b] Using a Personal Computer as a Slave

■ MEMOBUS Response Data

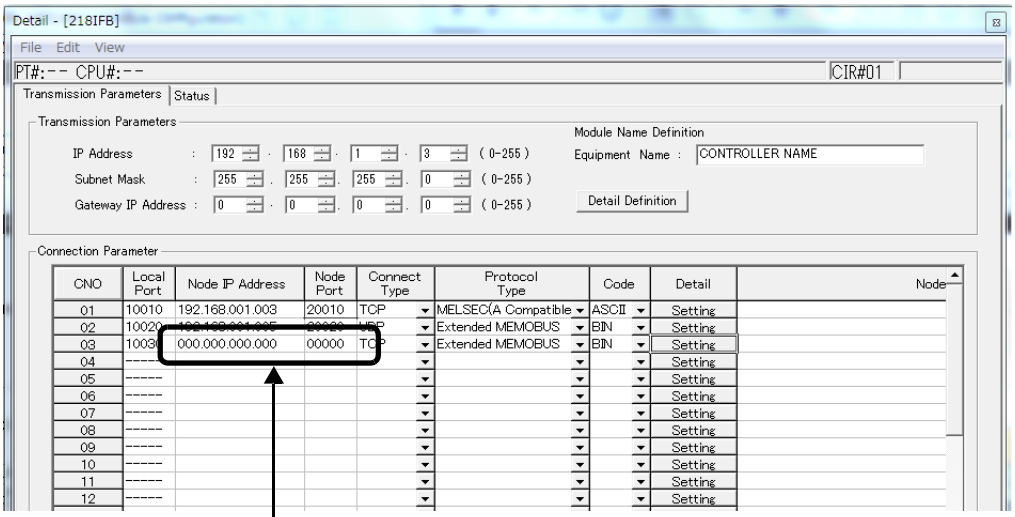
- Set the number that is set in the MEMOBUS command data for the identification number.
- Set the destination channel number that is set in the MEMOBUS command data.
- Set the source channel number that is set in the MEMOBUS command data.
- Reverse the upper-place 4 bits and lower-place 4 bits of the CPU number set by the MEMOBUS command data for the CPU number.

[c] Sample Program

Refer to *Appendix E Sample C Language Programs for the Master* on page A-35 and *Appendix F Sample C Language Programs for a Slave* on page A-41 for computer programming.

(4) Precautions on Programming

In Master programming created for TCP, the local IP address and port number may not be assigned with the bind( ) function depending on the application. If this happens, the system will assign an available port when the connect( ) function is called. In such cases, the remote IP address and remote port number will not agree with the values set in the **217IFB Transmission Configuratio**n Window, causing the 218IF-02 to refuse the connection.
When using Master programming that may result in this problem, set the node IP address and node port in the **Transmission Parameters** Tab Page of the **218IFB Transmission Configuration** Window to 0. With these settings, a connection request will be accepted from any station (unpassive open mode; refer to page 5-18).

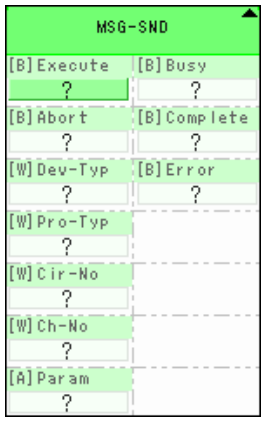


Specify 0.

Refer to *5.4.3 Status Tab Page* on page 5-20.

5.7 Message Send Function

5.7.1 Basic Specifications

Function Name	MSG-SND			
Function	Sends a message to a remote station on the line specified by the Transmission Device Type parameter (DEV-TYP). The function supports multiple protocols. The Execute command must be held ON until the Complete or Error output turns ON.			
Function Definition				
I/O Definitions	No.	Name	I/O Designation*1	Description
Inputs	1	Execute	B-VAL	Send Message command
	2	Abort	B-VAL	Send Message Abort command
	3	Dev-Typ	I-REG	Transmission Device Type MPLINK/CP-215 (MPLINK) = 1, RS-232C/RS-422/RS-485 (217IF) = 5, Ethernet (218IF) = 6, DeviceNet (260IF) = 11, Ethernet (218IFA, 218IFB, 218IFC) = 16
	4	Pro-Typ	I-REG	Communication Protocol MEMOBUS = 1*2, Non-procedure 1*3 = 2, Non-procedure 2*3 = 3
	5	Cir-No	I-REG	Circuit Number MPLINK/CP-215 (MPLINK) = 1 to 8, RS-232C/RS-422/RS-485 (217IF) = 1 to 16, Ethernet (218IF) = 1 to 8, DeviceNet (260IF) = 1 to 8, Ethernet (218IFB) = 1 to 8
	6	Ch-No	I-REG	Transmission Buffer Channel Number MPLINK/CP-215 (MPLINK) = 1 to 12, RS-232C/RS-422/RS-485 (217IF) = 1, Ethernet (218IF) = 1 to 10, DeviceNet (260IF) = 1 to 4, Ethernet (218IFB) = 1 to 10
	7	Param	Address input	Parameter List Leading Address (MA, DA)
Outputs	8	Busy	B-VAL	Processing in progress.
	9	Complete	B-VAL	Processing completed.
	10	Error	B-VAL	Error has occurred.

* 1. The I/O designations are as follows:
B-VAL: I/O is specified as bit data.
I-REG: I/O is specified as integer data. Specify the number of an integer register.
Constants (immediate data) can also be specified for inputs.
Address input: The address of the specified register (any integer register) is passed to the function.

* 2. To use the MEMOBUS, Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, select the MEMOBUS protocol by setting the Communication Protocol parameter (PRO-TYP) to 1. The transmission device will automatically convert to the proper protocol.

* 3. Non-procedure 1: Data is sent in word units.
Non-procedure 2: Data is sent in byte units.

5.7.2 Inputs and Outputs for the Message Send Function

(1) Inputs

The following table gives the registers that can be used for the inputs.

Inputs	I/O Designation	Applicable Registers
Execute Abort	B-VAL	Any bit registers (including those with subscripts) except for # and C registers
Dev-Typ Pro-Typ Cir-No Ch-No	I-REG	Any integer registers (including those with subscripts) Constants
Param	Address input	Any register addresses (including those with subscripts) except for # and C registers

The following sections describe the inputs in more detail.

[a] Execute (Send Message Execute Command)

Specify the bit that will be used to control execution of the Message Send function.
Message send processing is started when the Execute command turns ON. To execute processing, this bit must be turned ON and OFF, e.g., from the ladder program.

- The Execute command must be held at ON until the Complete or Error output turns ON. The message is sent when the Execute command turns ON. To send another message, always turn OFF the Execute command for at least one scan.

[b] Abort (Send Message Abort Command)

Specify the bit that will be used to abort the Message Send function.
Sending the message will be aborted when the Abort command turns ON. The Abort command takes priority over the Execute command.
To abort processing, this bit must be turned ON and OFF, e.g., from the ladder program.

[c] Dev-Typ (Transmission Device Type)

The Dev-Typ input specifies the transmission device type as shown in the following table.

Device	Type Code
MPLINK/CP-215 (MPLINK)	1
RS-232C/422/485 (217IF)	5
Ethernet (218IF)	6
DeviceNet (260IF)	11
Ethernet (218IFA, 218IFB, 218IFC)	16

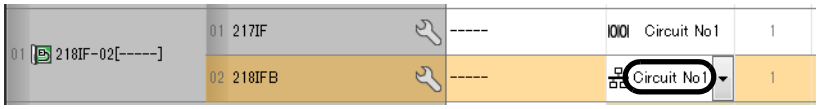
[d] Pro-Typ (Communication Protocol)

The Pro-Typ input specifies the communication protocol as shown in the following table.

Type Code	Communication Protocol	Remarks
1	MEMOBUS	To use the Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, set the type code to 1. The transmission device will automatically convert to the proper protocol.
2	Non-procedure 1 (word units)	Data will be sent in word units using the non-procedure protocol. No response is received from the remote station.
3	Non-procedure 2 (byte units)	Data will be sent in byte units using the non-procedure protocol. No response is received from the remote station.

[e] Cir-No (Circuit Number)

The Cir-No input specifies the circuit number of the transmission device.
Set the circuit number to the value displayed in the **MPE720 Module Configuration** Window.



The valid circuit number ranges are given in the following table.

Device	Valid Circuit Numbers
MPLINK/CP-215 (MPLINK)	1 to 8
RS-232C/422/485 (217IF)	1 to 16
Ethernet (218IF)	1 to 8
DeviceNet (260IF)	1 to 8
Ethernet (218IFB)	1 to 8

[f] Ch-No (Channel Number)

The Ch-No input specifies the channel number of the transmission buffer.
Any channel number can be specified as long as it is within the valid range. If more than one function is being executed at the same time, do not specify the same channel number more than once for the same modem number. (The same channel number can be used as long as the functions are not executed at the same time.)

The valid channel number ranges are given in the following table.

Device	Valid Channel Numbers
MPLINK/CP-215 (MPLINK)	1 to 12
RS-232C/422/485 (217IF)	1
Ethernet (218IF)	1 to 10
DeviceNet (260IF)	1 to 4
Ethernet (218IFB)	1 to 10

<Example>

If an Ethernet (218IF or 218IFB) transmission device is used, there are 10 channels of transmission buffers from 1 to 10 for sending and receiving combined, so up to 10 messages can be sent and received at the same time.

- One MSG-SND (or MSG-RCV) function must be programmed for each circuit being used at the same time.
- Refer to *A.3 Transmission Buffer Channels* on page A-9 for information on transmission buffer channels.

[g] Param (Parameter List Leading Address)

The PARAM input specifies the leading address of the parameter list. A parameter list will be automatically created from the 17 words starting with the specified address. Use the parameter list to input the function code and other related parameters. The processing results and status are also output to the parameter list.

- Refer to *5.7.3 Parameter List for Ethernet Communication (218IF or 218IFB)* on page 5-53 for information on the parameter list.

Example: The following parameter list will be created when the Parameter List Leading Address is set to DA000000.

Register	Parameter F・・・・・・0
DW000000	PARAM00
DW000001	PARAM01
DW000002	PARAM02
DW000003	PARAM03
DW000004	PARAM04
DW000005	PARAM05
DW000006	PARAM06
DW000007	PARAM07
DW000008	PARAM08
DW000009	PARAM09
DW000010	PARAM10
DW000011	PARAM11
DW000012	PARAM12
DW000013	PARAM13
DW000014	PARAM14
DW000015	PARAM15
DW000016	PARAM16

(2) Outputs

The following table gives the registers that can be used for the outputs.

Inputs	I/O Designation	Applicable Registers
Busy Complete Error	B-VAL	Any bit registers (including those with subscripts) except for # and C registers

The following sections describe the outputs in more detail.

[a] Busy (Processing in Progress)

Specify the bit that will report when sending the message is being processed.
The Busy output will be ON while message send processing or abort processing is in progress.
Keep the Execute command or Abort command ON while the Busy output is ON.

[b] Complete (Processing Completed)

Specify the bit that will report when processing to send the message has been completed.
The Complete output will be ON for only one scan after message send processing or abort processing has been completed normally.

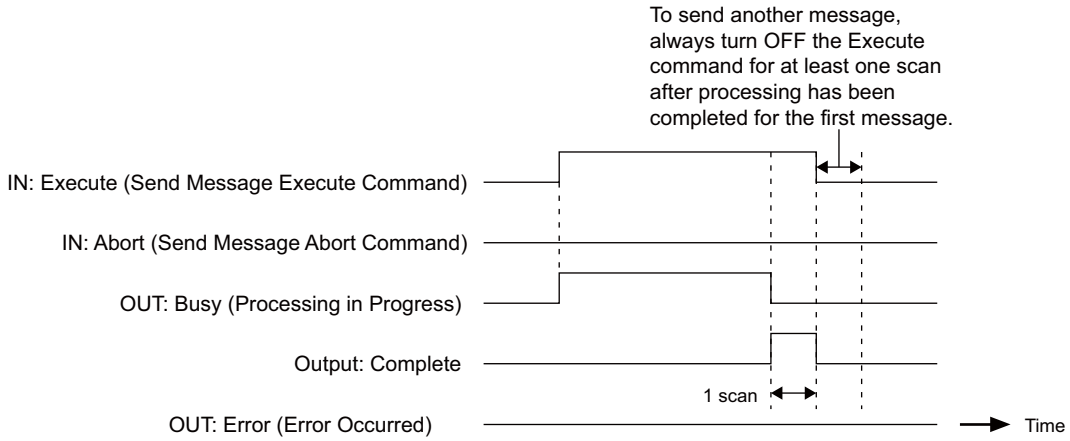
[c] ERROR (Error Occurred)

Specify the bit that will report when an error has occurred while sending the message.
The Error output will turn ON for only one scan when an error occurs.

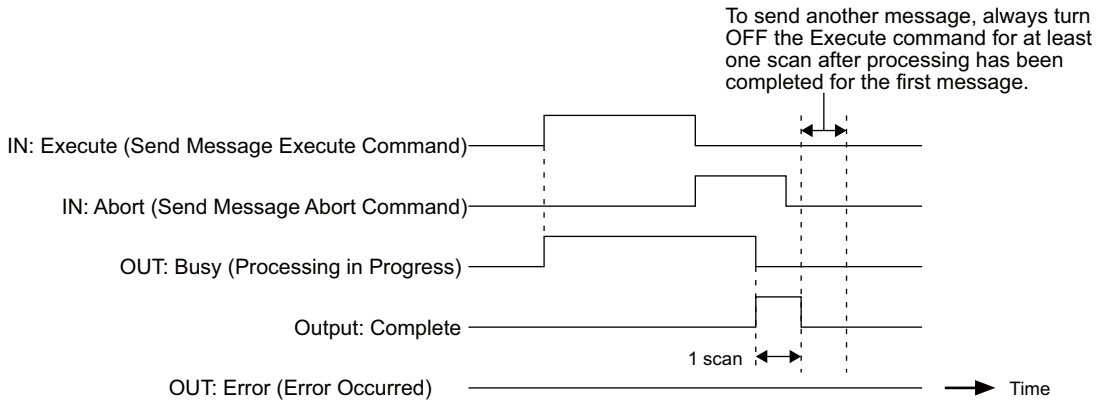
(3) I/O Timing Chart

The following timing charts show the bit inputs and outputs used with the MSG-SND function.

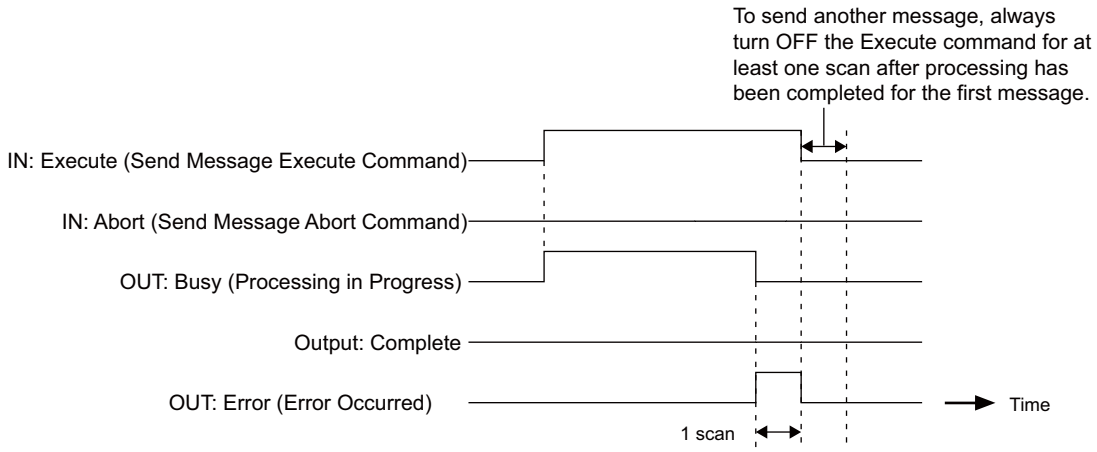
<Normal Processing>



<Abort Processing>



<Error Processing>



5.7.3 Parameter List for Ethernet Communication (218IF or 218IFB)

The Param input to the MSG-SND function is a parameter list structure consisting of 17 words. The value of the Param input is the leading address (MA or DA) of the parameter list.
Use the parameter list to input the connection number, function code, and other related parameters. The processing results and status are also output to the parameter list.
The parameter lists for the MEMOBUS and non-procedure communication protocols are given below.

■ Parameter List for MEMOBUS (Extended MEMOBUS, MELSEC or MODBUS/TCP) Protocol

PARAM No.	IN/OUT	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	IN	Connection number	Specifies the remote station number.
03	IN	Option	Sets options specific to each transmission device.
04	IN	Function code	Sets the function code to be sent.
05	IN	Data address	Specifies the leading address of the data.
06	IN	Data size	Sets the data size to be read or written.
07	IN	Remote CPU number	Sets the remote CPU number.
08	IN	Coil offset	Sets the offset word address of the coils.
09	IN	Input relay offset	Sets the offset word address of the input relays.
10	IN	Input register offset	Sets the offset word address of the input registers.
11	IN	Holding register offset	Sets the offset word address of the holding registers.
12	SYS	Reserved by the system (1).	
13 to 16	SYS	Reserved by the system (2).	

- IN: Input, OUT: Output, SYS: Used by the system.
- Refer to *5.7.4 Function Setting Example and Parameter Details for the MEMOBUS and Extended MEMOBUS Protocol* on page 5-54, *5.7.5 Function Setting Example and Parameter Details for the MELSEC Protocol* on page 5-62, and *5.7.6 Function Setting Example and Parameter Details for the MODBUS/TCP Protocol* on page 5-67 on for details on the parameters.

■ Parameter List for Non-procedure Protocol

PARAM No.	IN/OUT	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	IN	Connection number	Specifies the remote station number.
03	IN	(Not used.)	
04	IN	(Not used.)	
05	IN	Data address	Specifies the leading address of the data.
06	IN	Data size	Sets the data size to be read or written.
07	IN	(Not used.)	
08	IN	(Not used.)	
09	IN	(Not used.)	
10	IN	(Not used.)	
11	IN	Register offset	Sets the offset word address of the registers.
12	SYS	Reserved by the system (1).	
13 to 16	SYS	Reserved by the system (2).	

- IN: Input, OUT: Output, SYS: Used by the system.
- Refer to *5.7.7 Function Setting Example and Parameter Details for the Non-procedure Protocol* on page 5-73 for details on the parameters.

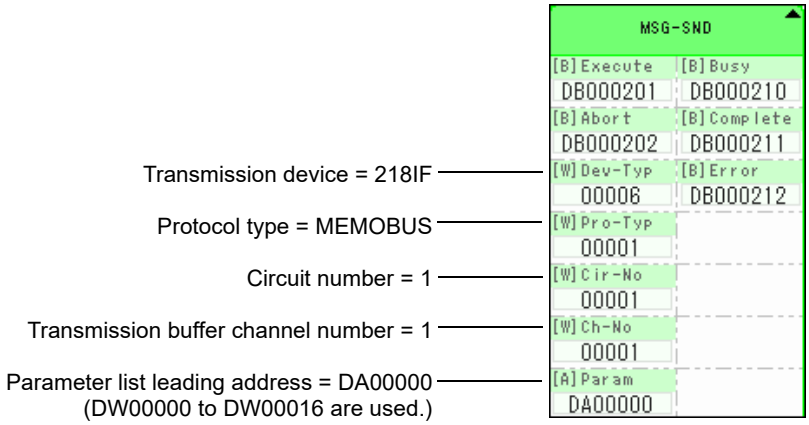
5.7.4 Function Setting Example and Parameter Details for the MEMOBUS and Extended MEMOBUS Protocol

This section provides an MSG-SND function setting example and describes the parameter list in detail for Ethernet communication (218IF or 218IFB) using the MEMOBUS protocol.

(1) MSG-SND Function Setting Example

<218IF Setting Example>

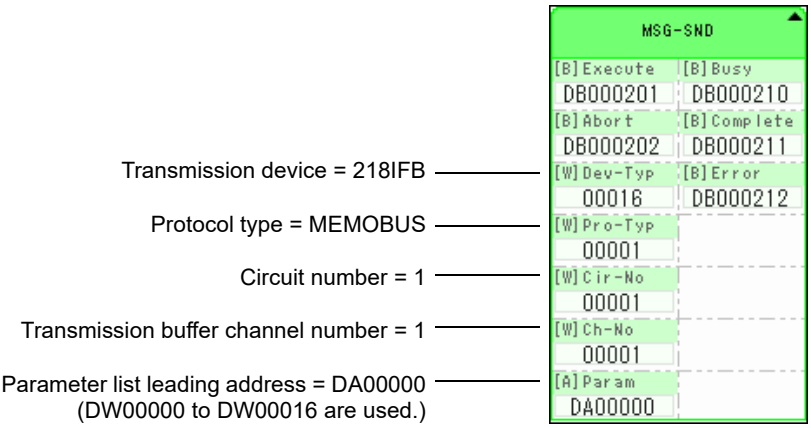
This example shows the settings for using the 218IF as the transmission device.



To use the Extended MEMOBUS protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.7.2 Inputs and Outputs for the Message Send Function* on page 5-49.

<218IFB Setting Example>

This example shows the settings for using the 218IFB as the transmission device.



To use the Extended MEMOBUS protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IFB transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings refer to *5.7.2 Inputs and Outputs for the Message Send Function* on page 5-49.

(2) Parameter List for MEMOBUS or Extended MEMOBUS Protocol

PARAM No.	IN/OUT	Contents	PARAM No.	IN/OUT	Contents
00	OUT	Processing result	07	IN	Remote CPU number
01	OUT	Status	08	IN	Coil offset
02	IN	Connection number	09	IN	Input relay offset
03	IN	Option	10	IN	Input register offset
04	IN	Function code	11	IN	Holding register offset
05	IN	Data address	12	SYS	Reserved by the system (1).
06	IN	Data size	13 to 16	SYS	Reserved by the system (2).

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for MEMOBUS or Extended MEMOBUS Protocol

This section describes the parameter list in detail for using the MEMOBUS or Extended MEMOBUS protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARAM00. The lower-place byte is for system analysis.

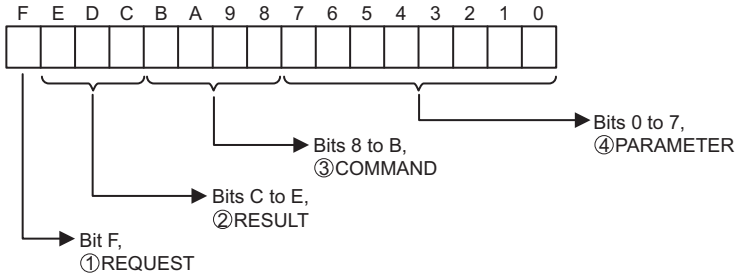
Value of Processing Result	Meaning
00xxH	Processing in progress (busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	Function code error	An undefined function code was sent or received. Check PARAM04 (function code).
82xxH	Address setting error	One of the following settings in not within the valid range. Check the settings. PARAM05 (data address) PARAM08 (coil offset) PARAM09 (input relay offset) PARAM10 (input register offset) PARAM11 (holding register offset)
83xxH	Data size error	The size of the sent or received data is not within the allowable range. Check PARAM06 (data size).
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-SND function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-SND function.
86xxH	Connection number error	The connection number is not within the allowable range. Check PARAM02 (connection number).
87xxH	–	–
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-SND function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-SND function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-SND function is output to these bits.

Code	Abbreviation	Meaning
0	CONN_NG	Sending or connection has ended abnormally for Ethernet communication.
1	SEND_OK	Sending has been completed normally.
2	REC_OK	Receiving has been completed normally.
3	Abort_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG or INIT_NG	Command sequence error
6	RESET_NG or O_RING_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-SND function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send General-purpose Message with Non-procedure Protocol
2	U_REC	Receive General-purpose Message with Non-procedure Protocol
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response.
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response.
C	MR_SEND	Send MEMOBUS response

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the connection number is output.

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Connection number out of range
	02	MEMOBUS response receive monitor time error
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error
Other values of RESULT	XX	Connection Number

[c] PARAM02: Connection Number

Specifies the remote station number.
Set the connection number when the transmission device is Ethernet (218IF or 218IFB). The valid setting ranges are given in the following table.

Transmission Device	Connection Number	Remarks
Ethernet (218IF or 218IFB)	1 to 20	The message is sent to the remote station with the designated connection number.

- Enter the same connection number as the one set in the **218IF/218IFB Transmission Parameters Tab** Page or the **MPE720 Module Configuration** Window.

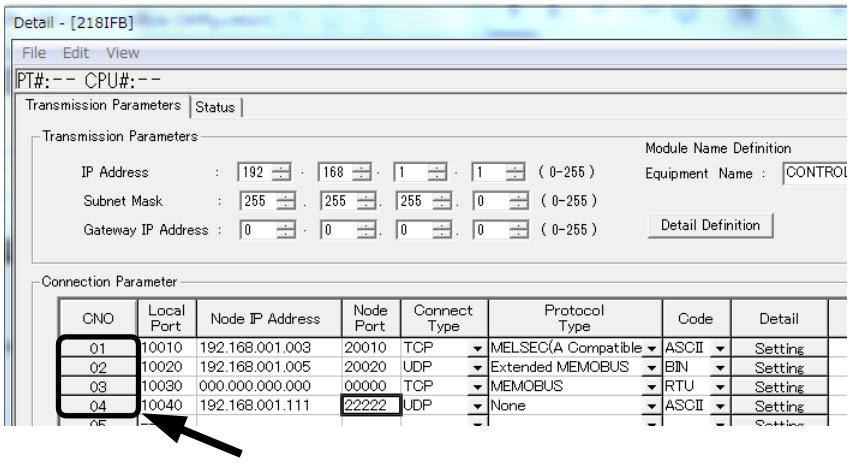


Fig. 5.9 The **218IF Transmission Parameters** Tab Page of the **MPE720 Module Configuration** Window

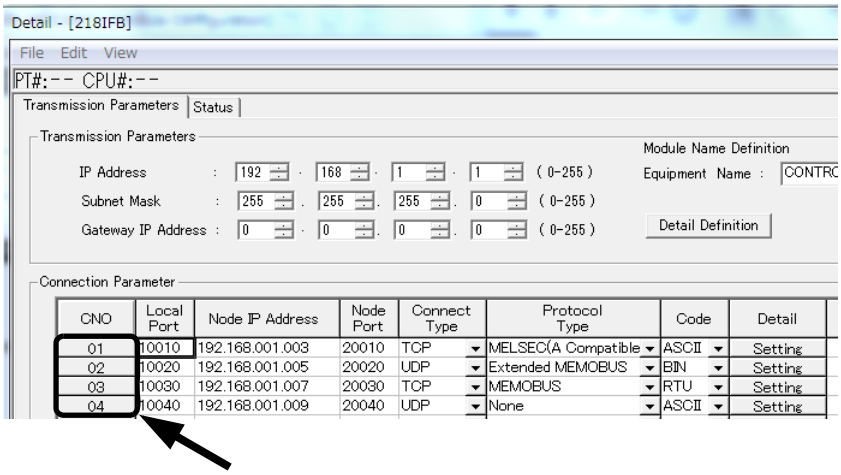


Fig. 5.10 The **218IFB Transmission Parameters** Tab Page of the **MPE720 Module Configuration** Window

[d] PARAM03: Option

PARAM03 sets options specific to each transmission device.
PARAM03 is not used for the MEMOBUS or Extended MEMOBUS protocol. Nothing needs to be set.

[e] PARAM04: Function Code

PARAM04 sets the function code to be sent.
The function registered to the function code is used by specifying the function code here. Functions include reading coil and input relay status and writing holding registers.
The function codes used for the MEMOBUS or Extended MEMOBUS protocol are listed in the following table.

<Function Codes>

Function Code	Applicable Data Type	Function	Protocol Type	
			Extended MEMOBUS	MEMOBUS
00H	–	Not used.	–	–
01H	B	Read Coil Status	○	○
02H	B	Read Input Relay Status	○	○
03H	W	Read Holding Register Contents	○	○
04H	W	Read Input Register Contents	○	○
05H	B	Modify Status of a Single Coil	○	○
06H	W	Write to a Single Holding Register	○	○
07H	–	Not used.	–	–
08H	–	Loopback Test	○	○
09H	W	Read Holding Register Contents (Extended)	○	×
0AH	W	Read Input Register Contents (Extended)	○	×
0BH	W	Write to Holding Register (Extended)	○	×
0CH	–	Not used.	–	–
0DH	W	Read Discontinuous Holding Register Contents (Extended)	○	×
0EH	W	Write to Discontinuous Holding Registers (Extended)	○	×
0FH	B	Modify Status of Multiple Coils	○	○
10H	W	Write to Multiple Holding Registers	○	○

- B: Bit data, W: Word data
- ○: Can be set, ×: Cannot be set.
- Only MW (MB) registers may be used for sending and receiving messages when operating as the Master.
- MB, MW, IB, and IW registers can be used as coils, holding registers, input relays, and input registers, respectively, when operating as a Slave.

[f] PARAM05: Data Address

PARAM05 specifies the leading address of the data.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

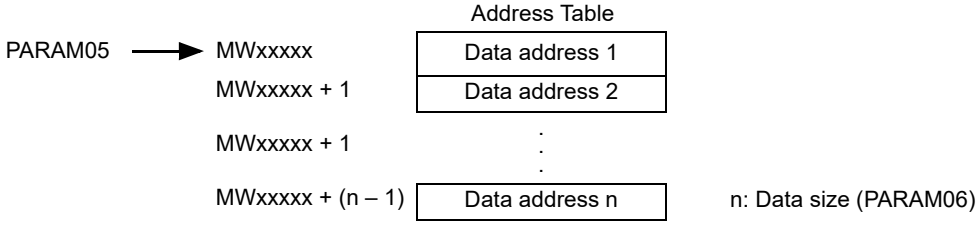
The data address setting depends on function code.
The data address setting ranges used for the MEMOBUS or Extended MEMOBUS protocol are listed in the following table.

<Data Address Setting Ranges for MEMOBUS or Extended MEMOBUS>

Function Code	Applicable Data Type	Function	Data Address Setting Range	
			Ethernet (218IF)	Ethernet (218IFB)
00H	–	Not used.	Invalid	
01H	B	Read Coil Status*1	0 to 65535 (0 to FFFFH)	
02H	B	Read Input Relay Status*1	0 to 65535 (0 to FFFFH)	
03H	W	Read Holding Register Contents*2	0 to 65534 (0 to FFFEH)	
04H	W	Read Input Register Contents*2	0 to 32767 (0 to 7FFFH)	0 to 65535 (0 to FFFFH)
05H	B	Modify Status of a Single Coil*1	0 to 65535 (0 to FFFFH)	
06H	W	Write to a Single Holding Register*2	0 to 65534 (0 to FFFEH)	
07H	–	Not used.	Invalid	
08H	–	Loopback Test	Invalid	
09H	W	Read Holding Register Contents (Extended)*2	0 to 65534 (0 to FFFEH)	
0AH	W	Read Input Register Contents (Extended)*2	0 to 32767 (0 to 7FFFH)	0 to 65535 (0 to FFFFH)
0BH	W	Write to Holding Register (Extended)*2	0 to 65534 (0 to FFFEH)	
0CH	–	Not used.	Invalid	
0DH	W	Read Discontinuous Holding Register Contents (Extended)*3	0 to 65534 (0 to FFFEH)	
0EH	W	Write to Discontinuous Holding Registers (Extended)*3	0 to 65534 (0 to FFFEH)	
0FH	B	Modify Status of Multiple Coils*1	0 to 65535 (0 to FFFFH)	
10H	W	Write to Multiple Holding Registers*2	0 to 65534 (0 to FFFEH)	

* 1. Read/write request for coils and relays: Set the leading bit address of the data.
* 2. Continuous read/write request for registers: Set the leading word address of the data.
* 3. Read/write request for discontinuous registers: Set the leading M register number of the address table.

- Address Table
  An address table specifies indirect addressing for discontinuous data. The M register specified by PARAM05 (data address) specifies the start of the address table and PARAM06 (data size) specifies the size of the address table.
  When reading data, set data addresses 1 to n to the address to be read at the remote station. The read data will be stored in the local station according to data addresses 1 to n.
  When writing data, the data stored in data addresses 1 to n at the local station will be read and then written to data addresses 1 to n at the remote station.
  The contents of an address table used to read/write discontinuous registers is illustrated below.



[g] PARAM06: Data Size

PARAM06 specifies the data size to be read or written as the number of bits or words.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size setting depends on function code and the transmission device.
The data size setting ranges used for the MEMOBUS or Extended MEMOBUS protocol are listed in the following table.

<Data Size Setting Ranges for MEMOBUS or Extended MEMOBUS>

Function Code	Applicable Data Type	Function	Data Address Setting Range	
			Ethernet (218IF)	Ethernet (218IFB)
00H	–	Not used.	Invalid	
01H	B	Read Coil Status*1	1 to 2000	
02H	B	Read Input Relay Status*1	1 to 2000	
03H	W	Read Holding Register Contents*2	1 to 125	
04H	W	Read Input Register Contents*2	1 to 125	
05H	B	Modify Status of a Single Coil*1	Invalid	
06H	W	Write to a Single Holding Register*2	Invalid	
07H	–	Not used.	Invalid	
08H	–	Loopback Test	Invalid	
09H	W	Read Holding Register Contents (Extended)*2	1 to 508	1 to 2044 (BIN) 1 to 1020 (ASCII)
0AH	W	Read Input Register Contents (Extended)*2	1 to 508	1 to 2044 (BIN) 1 to 1020 (ASCII)
0BH	W	Write to Holding Register (Extended)*2	1 to 507	1 to 2043 (BIN) 1 to 1019 (ASCII)
0CH	–	Not used.	Invalid	
0DH	W	Read Discontinuous Holding Register Contents (Extended)*2	1 to 508	1 to 2044 (BIN) 1 to 1020 (ASCII)
0EH	W	Write to Discontinuous Holding Registers (Extended)*2	1 to 254	1 to 1022 (BIN) 1 to 510 (ASCII)
0FH	B	Modify Status of Multiple Coils*1	1 to 800	
10H	W	Write to Multiple Holding Registers*2	1 to 100	

* 1. Set the number of bits.
* 2. Set the number of words.

- The data sizes listed in the table are decimal values.

[h] PARAM07: Remote CPU Number

PARAM07 sets the remote CPU number.
Set the remote CPU number to 1 if the remote device is an MP2000 Series Machine Controller.
If the remote device is a Yaskawa Controller, but not in the MP2000 Series and it consists of more than one CPU Module, set the destination CPU number.
In all other cases, set 0.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the offsets for the data read or write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter	Protocol Type	
			Extended MEMOBUS	MEMOBUS
01H	Read Coil Status	PARAM08	○	○
02H	Read Input Relay Status	PARAM09	○	○
03H	Read Holding Register Contents	PARAM11	○	○
04H	Read Input Register Contents	PARAM10	○	○
05H	Modify Status of a Single Coil	PARAM08	○	○
06H	Write to a Single Holding Register	PARAM11	○	○
09H	Read Holding Register Contents (Extended)	PARAM11	○	×
0AH	Read Input Register Contents (Extended)	PARAM10	○	×
0BH	Write to Holding Register (Extended)	PARAM11	○	×
0DH	Read Discontinuous Holding Register Contents (Extended)	PARAM11	○	×
0EH	Write to Discontinuous Holding Registers (Extended)	PARAM11	○	×
0FH	Modify Status of Multiple Coils	PARAM08	○	○
10H	Write to Multiple Holding Registers	PARAM11	○	○

- ○: Can be set, ×: Cannot be set.

[j] PARAM12: Reserved by the system (1).

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[k] PARAM13 to PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

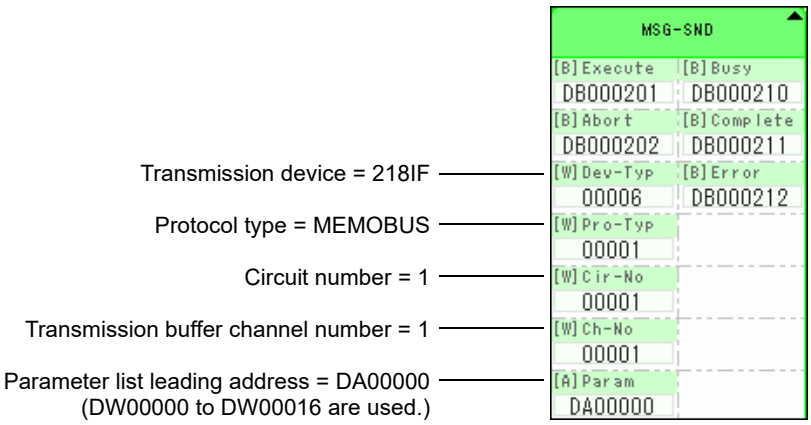
5.7.5 Function Setting Example and Parameter Details for the MELSEC Protocol

This section provides an MSG-SND function setting example and describes the parameter list in detail for Ethernet communication (218IF or 218IFB) using the MELSEC protocol.

(1) MSG-SND Function Setting Example

<218IF Setting Example>

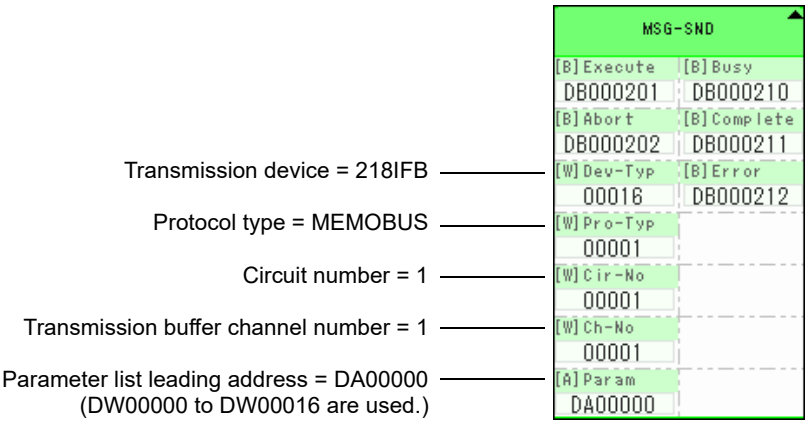
This example shows the settings for using the 218IF as the transmission device.



To use the MELSEC protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on the settings, refer to *5.7.2 Inputs and Outputs for the Message Send Function* on page 5-49.

<218IFB Setting Example>

This example shows the settings for using the 218IFB as the transmission device.



To use the MELSEC protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IFB transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.7.2 Inputs and Outputs for the Message Send Function* on page 5-49.

(2) Parameter List for MELSEC Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	07	IN	Remote CPU number
01	OUT	Status	08	IN	Coil offset
02	IN	Connection number	09	IN	Input relay offset
03	IN	Option	10	IN	Input register offset
04	IN	Function code	11	IN	Holding register offset
05	IN	Data address	12	SYS	Reserved by the system (1).
06	IN	Data size	13 to 16	SYS	Reserved by the system (2).

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for MELSEC Protocol

This section describes the parameter list in detail for using the MELSEC protocol.

[a] PARAM00: Processing Result

Refer to *5.7.4 (3) [a] PARAM00: Processing Result* on page 5-55.

[b] PARAM01: Status

Refer to *5.7.4 (3) [b] PARAM01: Status* on page 5-56.

[c] PARAM02: Connection Number

Refer to *5.7.4 (3) [c] PARAM02: Connection Number* on page 5-57.

[d] PARAM03: Option

If the protocol type is MELSEC, this parameter is not used and does not need to be set.

[e] PARAM04: Function Code

PARAM04 sets the function code to be sent.
The function registered to the function code is used by specifying the function code here. Functions include reading bit and word devices and writing word devices.
The function codes used for the MELSEC protocol are listed in the following table.

<Function Codes for MELSEC>

Common MELSEC ACPU Command	Function Code	Applicable Data Type	Function
00H	01H/02H	B	Reads data from bit-type devices in 1-point units.
01H	03H/04H/09H/0AH	W	Reads data from word-type devices in 1-point units.
02H	05H/0FH	B	Writes data to bit-type devices in 1-point units.
03H	06H/0BH/10H	W	Writes data to word-type devices in 1-point units.
16H	08H	–	Executes a loopback test.
05H	0EH	B	Sets/resets data in word devices specified at random in 1-point units.
60H	31H	W	Fixed Buffer Communication
61H	32H	W	Reads data in word units from the random-access buffers.
62H	33H	W	Writes data in word units to the random-access buffers.

- B: Bit data, W: Integer data
- The special commands for the AnCPU are not supported. Use the common ACPU commands listed above when accessing an AnCPU. Accessing the extended file registers in an AnCPU is not possible.

[f] PARAM05: Data Address

PARAM05 specifies the leading address of the data.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

The setting ranges for function codes and data addresses depend on the MELSEC device type and device range.
The data address setting ranges used for the MELSEC protocol are listed in the following tables.

<Data Address Setting Ranges for MELSEC Bit Devices>

Device	Common ACPU Command Device Range	Decimal/ Hex	Function Code	Data Address Setting Range	Corresponding Register Numbers
X	X0000 to X07FF	Hex	02H: Input relays	0 to 2047	MB000000 to MB00127F
Y	Y0000 to Y07FF	Hex	01H/0FH: Coils	0 to 2047	MB000000 to MB00127F
M	M000 to M2047	Decimal	01H/05H/0FH: Coils	2048 to 4095	MB001280 to MB00255F
M	M9000 to M9255	Decimal	01H/05H/0FH: Coils	4096 to 4351	MB002560 to MB00271F
B	B0000 to B03FF	Hex	01H/05H/0FH: Coils	4352 to 5375	MB002720 to MB00335F
F	F0000 to F0255	Decimal	01H/05H/0FH: Coils	5376 to 5631	MB003360 to MB00351F
TS	TS000 to TS255	Decimal	02H: Input relays	2048 to 2303	MB001280 to MB00143F
TC	TC000 to TC255	Decimal	02H: Input relays	2304 to 2559	MB001440 to MB00159F
CS	CS000 to CS255	Decimal	02H: Input relays	2560 to 2815	MB001660 to MB00175F
CC	CC000 to CC255	Decimal	02H: Input relays	2816 to 3071	MB001760 to MB00191F
M*	M2048 to M8191	Decimal	01H/05H/0FH: Coils	8192 to 14335	MB005120 to MB00895F

* Usable only with the 218IFB.

- Even for the same device ranges, the device area ranges depend on the MELSEC Sequencer. Refer to the MELSEC manuals for details.

<Data Address Setting Ranges for MELSEC Word Devices>

Device	Common ACPU Command Device Range	Decimal/ Hex	Function Code	Data Address Setting Range	Corresponding Register Numbers
TN	TN0000 to TN255	Decimal	04H/0AH: Input registers	0 to 255	MW00000 to MW00255
CN	CN0000 to CN255	Decimal	04H/0AH: Input registers	256 to 511	MW00256 to MW00511
D	D0000 to D1023	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	0 to 1023	MW00000 to MW01023
D (Special)	D9000 to D9255	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	1024 to 1279	MW01024 to MW01279
W	W0000 to W03FF	Hex	03H/06H/09H/0BH/0EH/ 10H: Holding registers	1280 to 2303	MW01280 to MW02303
R	R0000 to R8191	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	2304 to 10495	MW02304 to MW10495
D*	D1024 to D6143	Decimal	03H/06H/09H/0BH/0EH/ 10H: Holding registers	10496 to 15615	MW10496 to MW15615

* Usable only with the 218IFB.

- Even for the same device ranges, the device area ranges depend on the MELSEC Sequencer. Refer to the MELSEC manuals for details.
- The MP2000 Series Machine Controller register addresses can be adjusted by using the offset setting in the MSG-SND function.

[g] PARAM06: Data Size

PARAM06 specifies the data size to be read or written as the number of bits or words.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size setting depends on function code and the transmission device.
The data size setting ranges used for the MELSEC protocol are listed in the following table.

<Data Size Setting Ranges for MELSEC>

Common MELSEC ACPU Command	MEMOBUS Function Code	Function	Connection Type	Code	Data Size Setting Range	
					Ethernet (218IF)	Ethernet (218IFB)
00H	01H/02H	Reads data from bit-type devices in 1-point units.	–		1 to 256 points	
01H	03H/04H/09H/0AH	Reads data from word-type devices in 1-point units.	–		1 to 256 points	
02H	05H/0FH	Writes data to bit-type devices in 1-point units.	–		1 to 256 points	
03H	06H/0BH/10H	Writes data to word-type devices in 1-point units.	–		1 to 256 points	
16H	08H	Executes a loopback test.	–		–	
05H	0EH	Sets/resets data in word devices specified at random in 1-point units.	–		1 to 40 points	
60H	31H	Writes data in word units to the fixed buffers.	TCP	BIN	1 to 507 words	1 to 727 words
				ASCII	1 to 362 words	1 to 362 words
			UDP	BIN	1 to 507 words	1 to 1017 words
				ASCII	1 to 507 words	1 to 508 words
61H	32H	Reads data in word units from the random-access buffers.	TCP	BIN	1 to 508 words	1 to 728 words
				ASCII	1 to 363 words	1 to 363 words
			UDP	BIN	1 to 508 words	1 to 1017 words
				ASCII	1 to 508 words	1 to 508 words
62H	33H	Writes data in word units to the random-access buffers.	TCP	BIN	1 to 507 words	1 to 726 words
				ASCII	1 to 361 words	1 to 361 words
			UDP	BIN	1 to 508 words	1 to 1017 words
				ASCII	1 to 508 words	1 to 508 words

- The maximum data size for TCP is the maximum size that can be sent with one segment. A "segment" is a unit of data transfer for TCP. It is determined by the MTU (maximum transfer unit).
  The data size setting ranges given above are for when the MTU is set to 1500 bytes.

[h] PARAM07: Remote CPU Number

Refer to *5.7.4 (3) [h] PARAM07: Remote CPU Number* on page 5-60.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the offsets for the data read or write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter
01H	Read Coil Status	PARAM08
02H	Read Input Relay Status	PARAM09
03H	Read Holding Register Contents	PARAM11
04H	Read Input Register Contents	PARAM10
05H	Modify Status of a Single Coil	PARAM08
06H	Write to a Single Holding Register	PARAM11
09H	Read Holding Register Contents (Extended)	PARAM11
0AH	Read Input Register Contents (Extended)	PARAM10
0BH	Write to Holding Register (Extended)	PARAM11
0EH	Write to Discontinuous Holding Registers (Extended)	PARAM11
0FH	Modify Status of Multiple Coils	PARAM08
10H	Write to Multiple Holding Registers	PARAM11
31H	Write to Fixed Buffers	PARAM11
32H	Read from Random-access Buffers	PARAM11
33H	Write to Random-access Buffers	PARAM11

[j] PARAM12: Reserved by the System (1)

PARAM12 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[k] PARAM13 to PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

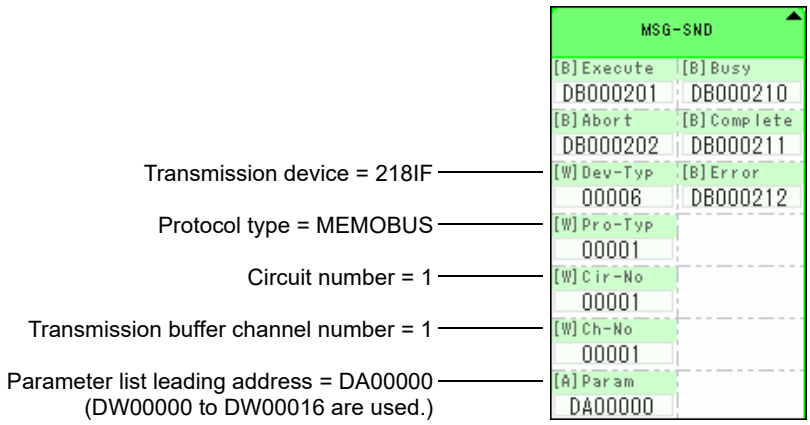
5.7.6 Function Setting Example and Parameter Details for the MODBUS/TCP Protocol

This section provides an MSG-SND function setting example and describes the parameter list in detail for Ethernet communication (218IF or 218IFB) using the MODBUS/TCP protocol.

(1) MSG-SND Function Setting Example

<218IF Setting Example>

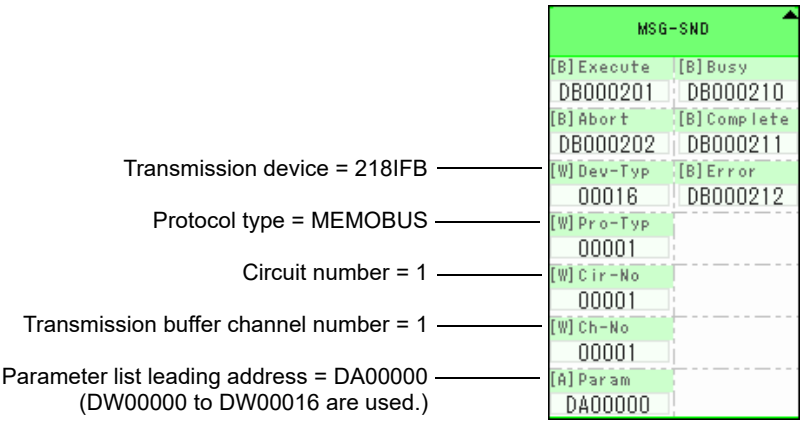
This example shows the settings for using the 218IF as the transmission device.



To use the MODBUS/TCP protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.7.2 Inputs and Outputs for the Message Send Function* on page 5-49.

<218IFB Setting Example>

This example shows the settings for using the 218IFB as the transmission device.



To use the MODBUS/TCP protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IFB transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.7.2 Inputs and Outputs for the Message Send Function* on page 5-49.

(2) Parameter List for MODBUS/TCP Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	07	IN	Remote CPU number
01	OUT	Status	08	IN	Coil offset
02	IN	Connection number	09	IN	Input relay offset
03	IN	Option	10	IN	Input register offset
04	IN	Function code	11	IN	Holding register offset
05	IN	Data address	12	SYS	Reserved by the system (1).
06	IN	Data size	13 to 16	SYS	Reserved by the system (2).

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for MODBUS/TCP Protocol

This section describes the parameter list in detail for using the MODBUS/TCP protocol.

[a] PARAM00: Processing Result

Refer to *5.7.4 (3) [a] PARAM00: Processing Result* on page 5-55.

[b] PARAM01: Status

Refer to *5.7.4 (3) [b] PARAM01: Status* on page 5-56.

[c] PARAM02: Connection Number

Refer to *5.7.4 (3) [c] PARAM02: Connection Number* on page 5-57.

[d] PARAM03: Option

PARAM03 sets options specific to each transmission device. The valid setting ranges are given in the following table.

Transmission Device	Setting Range	Remarks
Ethernet (218IF or 218IFB)	0 to 247, 255	Set the remote unit ID for communication using MODBUS/TCP protocol. Set the unit ID to 0 if the remote device is an MP2000 Series Machine Controller.

[e] PARAM04: Function Code

PARAM04 sets the function code to be sent.
The function registered to the function code is used by specifying the function code here. Functions include reading bit and word devices and writing word devices.
The function codes used for the MODBUS/TCP protocol are listed in the following table.

<Function Codes for MODBUS/TCP>

Function Code	Applicable Data Type	Function	Protocol Type	
			Ethernet (218IF)	Ethernet (218IFB)
00H	–	Not used.	–	–
01H	B	Read Coil Status	○	○
02H	B	Read Input Relay Status	○	○
03H	W	Read Holding Register Contents	○	○
04H	W	Read Input Register Contents	○	○
05H	B	Modify Status of a Single Coil	○	○
06H	W	Write to a Single Holding Register	○	○
07H : 0EH	–	Not used.	–	–
0FH	B	Modify Status of Multiple Coils	○	○
10H	W	Write to Multiple Holding Registers	○	○
11H : 15H	–	Not used.	–	–
16H	W	Mask Write to Holding Registers	×	○
17H	W	Read/Write to Multiple Holding Registers	×	○

- B: Bit data, W: Word data
- ○: Can be set, ×: Cannot be set.
- Only MW (MB) registers may be used for sending and receiving messages when operating as the Master.
- MB, MW, IB, and IW registers can be used as coils, holding registers, input relays, and input registers, respectively, when operating as a Slave.

[f] PARAM05: Data Address

PARAM05 specifies the leading address of the data.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

The data address setting ranges used for the MODBUS/TCP protocol are listed in the following tables.

<Data Address Setting Ranges for MODBUS/TCP>

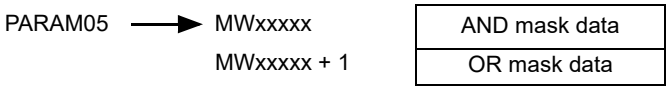
Function Code	Applicable Data Type	Function	Data Address Setting Range
00H	–	Not used.	Invalid
01H	B	Read Coil Status	0 to 65535 (0 to FFFFH)
02H	B	Read Input Relay Status	0 to 65535 (0 to FFFFH)
03H	W	Read Holding Register Contents	0 to 65534 (0 to FFFEH)
04H	W	Read Input Register Contents	0 to 32767 (0 to 7FFFH)
05H	B	Modify Status of a Single Coil	0 to 65535 (0 to FFFFH)
06H	W	Write to a Single Holding Register	0 to 65534 (0 to FFFEH)
07H : 0EH	–	Not used.	Invalid
0FH	B	Modify Status of Multiple Coils	0 to 65535 (0 to FFFFH)
10H	W	Write to Multiple Holding Registers	0 to 65534 (0 to FFFEH)
11H : 15H	–	Not used.	Invalid
16H	W	Mask Write to Holding Registers*1 *2	0 to 65534 (0 to FFFEH)
17H	W	Read/Write to Multiple Holding Registers*1 *3	0 to 65534 (0 to FFFEH)

* 1. Only for Ethernet (218IFB).
* 2. Mask write request for holding registers: Set the leading M register number of the remote and local address table.
* 3. Read/write request for multiple holding registers: Set the leading M register number of the address table.

- Data Table
  The data table used for mask writes to holding registers stores the mask data. The M register specified by PARAM05 (data address) specifies the start of a two-word data table.
  AND mask and OR mask data are set in the data table. PARAM05 (data address) used by the Mask Write to Holding Registers designates both the leading address at the remote station where the mask is being written and the leading M register of the data table at the local station.

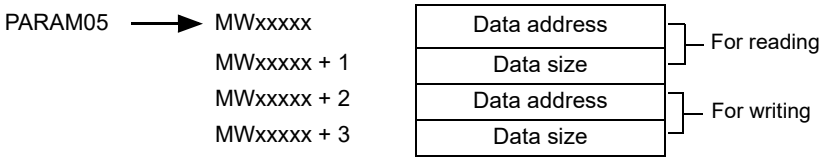
  The contents of a data table used to write a holding register mask is illustrated below.

Data Table for Mask Write to Holding Registers



- Address Table
  The address table used to read or write multiple holding registers specifies indirect addressing of the data to be read or written. The M register specified by PARAM05 (data address) specifies the start of a four-word address table. The address table contains the data address and data size to be read and the data address and size to be written.
  The read operation is the same as that of Read Holding Register Contents (function code 30H). The write operation is the same as that of Write to Multiple Holding Register Contents (function code 10H).
  The contents of an address table used to read/write multiple holding registers is illustrated below.

Address Table to Read/Write to Multiple Holding Registers



[g] PARAM06: Data Size

PARAM06 specifies the data size to be read or written as the number of bits or words.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size setting depends on function code and the transmission device.
The data address setting ranges used for the MODBUS/TCP protocol are listed in the following tables.

<Data Size Setting Ranges for MODBUS/TCP>

Function Code	Applicable Data Type	Function	Data Size Setting Range
00H	–	Not used.	Invalid
01H	B	Read Coil Status*1	1 to 2000
02H	B	Read Input Relay Status*1	1 to 2000
03H	W	Read Holding Register Contents*2	1 to 125
04H	W	Read Input Register Contents*2	1 to 125
05H	B	Modify Status of a Single Coil*1	Invalid
06H	W	Write to a Single Holding Register*2	Invalid
07H : 0EH	–	Not used.	Invalid
0FH	B	Modify Status of Multiple Coils*1	1 to 800
10H	W	Write to Multiple Holding Registers*2	1 to 100
11H : 15H	–	Not used.	Invalid
16H	W	Mask Write to Holding Registers*2 *3	Invalid
17H	W	Read/Write to Multiple Holding Registers*2 *3	Reading data: 1 to 125 Writing data: 1 to 100

* 1. Set the number of bits.
* 2. Set the number of words.
* 3. These settings are not valid when using Ethernet (218IF).

- The data sizes listed in the table are decimal values.

[h] PARAM07: Remote CPU Number

Refer to *5.7.4 (3) [h] PARAM07: Remote CPU Number* on page 5-60.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the offsets for the data read or write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter
01H	Read Coil Status	PARAM08
02H	Read Input Relay Status	PARAM09
03H	Read Holding Register Contents	PARAM11
04H	Read Input Register Contents	PARAM10
05H	Modify Status of a Single Coil	PARAM08
06H	Write to a Single Holding Register	PARAM11
0FH	Modify Status of Multiple Coils	PARAM08
10H	Write to Multiple Holding Registers	PARAM11
16H	Mask Write to Holding Registers	PARAM11
17H	Read/Write to Multiple Holding Registers	PARAM11

[j] PARAM12: Reserved by the System (1)

PARAM12 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[k] PARAM13 to PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

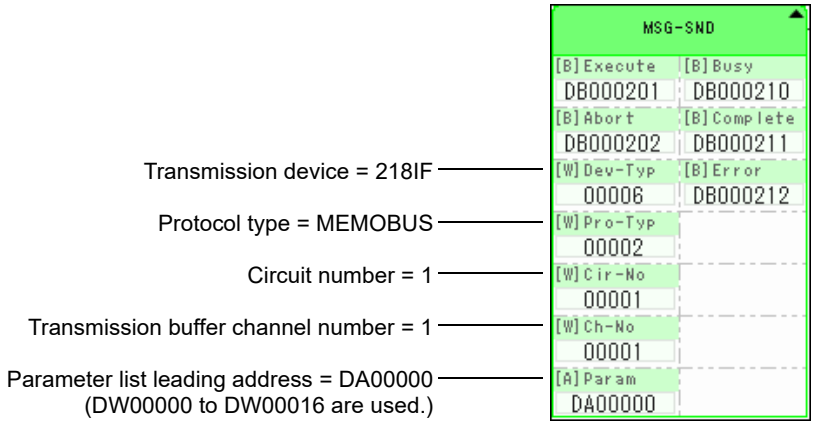
5.7.7 Function Setting Example and Parameter Details for the Non-procedure Protocol

The non-procedure protocol sends the contents of M registers without converting it in any way. This enables creating any protocol required by the remote station.
This section provides an MSG-SND function setting example and describes the parameter list in detail for Ethernet communication (218IF or 218IFB) using the non-procedure protocol.

(1) MSG-SND Function Setting Example

<218IF Setting Example>

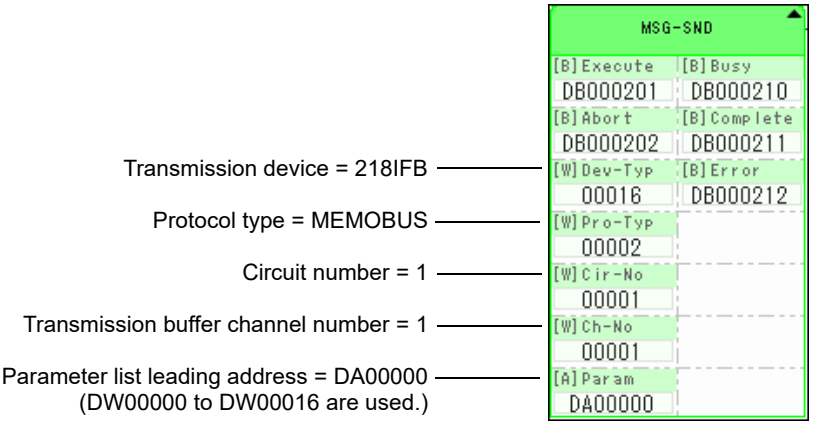
This example shows the settings for using the 218IF as the transmission device.



The protocol type (Pro-Typ) is set to 2 to use non-procedure protocol 1 and to 3 to use non-procedure protocol 2.
The circuit number is set to the circuit number assigned to the 218IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.7.2 Inputs and Outputs for the Message Send Function* on page 5-49.

<218IFB Setting Example>

This example shows the settings for using the 218IFB as the transmission device.



The protocol type (Pro-Typ) is set to 2 to use non-procedure protocol 1 and to 3 to use non-procedure protocol 2.
The circuit number is set to the circuit number assigned to the 218IFB transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.7.2 Inputs and Outputs for the Message Send Function* on page 5-49.

(2) Parameter List for Non-procedure Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	07	IN	(Not used.)
01	OUT	Status	08	IN	(Not used.)
02	IN	Connection number	09	IN	(Not used.)
03	IN	(Not used.)	10	IN	(Not used.)
04	IN	(Not used.)	11	IN	Register offset
05	IN	Data address	12	SYS	Reserved by the system (1).
06	IN	Data size	13 to 16	SYS	Reserved by the system (2).

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for Non-procedure Protocol

This section describes the parameter list in detail for using the non-procedure protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARAM00. The lower-place byte is for system analysis.

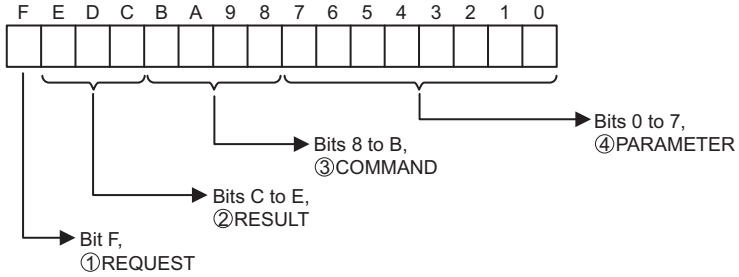
Value of Processing Result	Meaning
00xxH	Processing in progress (busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	–	–
82xxH	Address setting error	One of the following settings is not within the valid range. Check the settings. PARAM05 (data address) PARAM11 (holding register offset)
83xxH	Data size error	The size of the sent or received data is not within the allowable range. Check PARAM06 (data size).
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-SND function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-SND function.
86xxH	Connection number error	The connection number is not within the allowable range. Check PARAM02 (connection number).
87xxH	–	Reserved by the system.
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-SND function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-SND function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-SND function is output to these bits.

Code	Abbreviation	Meaning
0	CONN_NG	Sending or connection has ended abnormally for Ethernet communication.
1	SEND_OK	Sending has been completed normally.
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG or INIT_NG	Command sequence error
6	RESET_NG or O_RING_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-SND function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send general-purpose message.
2	U_REC	Receive general-purpose message.
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response. (For MEMOBUS Protocol)
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response. (For MEMOBUS Protocol)
C	MR_SEND	Send MEMOBUS response (For MEMOBUS Protocol)

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the connection number is output.

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Connection number out of range
	02	MEMOBUS response receive monitor time error (for MEMOBUS protocol)
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error (for MEMOBUS protocol)
Other values of RESULT	XX	Connection Number

[c] PARAM02: Connection Number

Refer to *5.7.4 (3) [c] PARAM02: Connection Number* on page 5-57.

[d] PARAM05: Data Address

PARAM05 specifies the leading address of the data.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

The valid data address setting ranges are given in the following table.

<Data Address Setting Ranges for Non-procedure Protocol>

Non-procedure Protocol Type	Applicable Data Type	Function	Data Address Setting Range	
			Ethernet (218IF)	Ethernet (218IFB)
Non-procedure 1	W	Sends data in word units.	0 to 65534 (0 to FFFEH)	
Non-procedure 2	B	Sends data in byte units.*	0 to 65534 (0 to FFFEH)	

* The address setting must be in word units.

[e] PARAM06: Data Size

PARAM06 specifies the data size to be read or written as the number of bits or words.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size setting depends on function code and the transmission device.

<Data Size Setting Ranges for Non-procedure Protocol>

Non-procedure Protocol Type	Applicable Data Type	Function	Data Address Setting Range	
			Ethernet (218IF)	Ethernet (218IFB)
Non-procedure 1	W	Sends data in word units.*1	1 to 510	1 to 2046 (BIN) 1 to 1023 (ASCII)
Non-procedure 2	B	Sends data in byte units.*2	1 to 1020	1 to 4092 (BIN) 1 to 2046 (ASCII)

* 1. Set the number of words.
* 2. Set the number of bytes.

- The data sizes listed in the table are decimal values.

[f] PARAM11: Offset

PARAM11 sets the offset for the data write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

Example: PARAM11 is set as follows to offset the register address by 1000 words.

PARAM11 = 1000

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

[g] PARAM12: Reserved by the System (1)

PARAM12 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

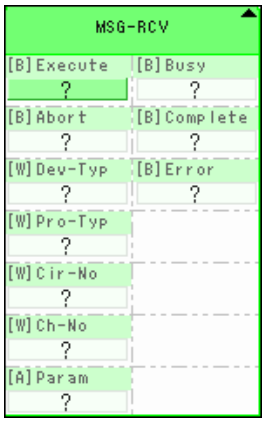
[h] PARAM13 to PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

5.8 Message Receive Function

This section describes the Message Receive function (MSG-RCV) used in a ladder program to receive a message.

5.8.1 Basic Specifications

Function Name	MSG-RCV			
Function	Receives a message from a remote station on the line specified by the Transmission Device Type parameter (DEV-TYP). The function supports multiple protocols. The Execute command must be held ON until the Complete or Error output turns ON.			
Function Definition				
I/O Definitions	No.	Name	I/O Designation*1	Description
Inputs	1	Execute	B-VAL	Receive message command
	2	Abort	B-VAL	Reception abort command
	3	Dev-Typ	I-REG	Transmission device type MPLINK/CP-215 (MPLINK) = 1, RS-232C/RS-422/RS-485 (217IF) = 5, Ethernet (218IF) = 6, Ethernet (218IFA, 218IFB, 218IFC) = 16
	4	Pro-Typ	I-REG	Communication protocol MEMOBUS = 1*2, Non-procedure 1*3 = 2, Non-procedure 2*3 = 3
	5	Cir-No	I-REG	Circuit number MPLINK/CP-215 (MPLINK) = 1 to 8, RS-232C/RS-422/RS-485 (217IF) = 1 to 16, Ethernet (218IF) = 1 to 8, Ethernet (218IFB) = 1 to 8
	6	Ch-No	I-REG	Transmission buffer channel number MPLINK/CP-215 (MPLINK) = 1 to 12, RS-232C/RS-422/RS-485 (217IF) = 1, Ethernet (218IF) = 1 to 10, Ethernet (218IFB) = 1 to 10
	7	Param	Address input	Parameter List Leading Address (MA, DA)
Outputs	8	Busy	B-VAL	Processing in progress.
	9	Complete	B-VAL	Processing completed.
	10	Error	B-VAL	Error has occurred.

* 1. The I/O designations are as follows:
B-VAL: I/O is specified as bit data.
I-REG: I/O is specified as integer data. Specify the number of an integer register.
Constants (immediate data) can also be specified for inputs.
Address input: The address of the specified register (any integer register) is passed to the function.

* 2. To use the MEMOBUS, Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, select the MEMOBUS protocol by setting the Communication Protocol parameter (PRO-TYP) to 1. The transmission device will automatically convert to the proper protocol.

* 3. Non-procedure 1: Data is received in word units.
Non-procedure 2: Data is received in byte units.

5.8.2 Inputs and Outputs for the Message Receive Function

(1) Inputs

The following table gives the registers that can be used for the inputs.

Inputs	I/O Designation	Applicable Registers
Execute Abort	B-VAL	Any bit registers (including those with subscripts) except for # and C registers
Dev-Typ Pro-Typ Cir-No Ch-No	I-REG	Any integer registers (including those with subscripts) Constants
Param	Address input	Any register addresses (including those with subscripts) except for # and C registers

The following sections describe the inputs in more detail.

[a] Execute (Receive Message Execute Command)

Specify the bit that will be used to control execution of the Message Receive function.
Message receive processing is started when the Execute command turns ON. To execute processing, this bit must be turned ON and OFF, e.g., from the ladder program.

- The Execute command must be held at ON until the Complete or Error output turns ON. A message is received when the Execute command turns ON.

[b] Abort (Receive Message Abort Command)

Specify the bit that will be used to abort the Message Receive function.
Receiving the message will be aborted when the Abort command turns ON. The Abort command takes priority over the Execute command.
To abort processing, this bit must be turned ON and OFF, e.g., from the ladder program.

[c] Dev-Typ (Transmission Device Type)

The Dev-Typ input specifies the transmission device type as show in the following table.

Device	Type Code
MPLINK/CP-215 (MPLINK)	1
RS-232C/422/485 (217IF)	5
Ethernet (218IF)	6
Ethernet (218IFA, 218IFB, 218IFC)	16

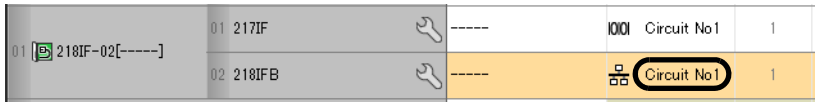
[d] Pro-Typ (Communication Protocol)

The Pro-Typ input specifies the communication protocol as shown in the following table.

Type Code	Communication Protocol	Remarks
1	MEMOBUS	To use the Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, set the type code to 1. The transmission device will automatically convert to the proper protocol.
2	Non-procedure 1 (word units)	Data will be received in word units using the non-procedure protocol. No response is received from the remote station.
3	Non-procedure 2 (byte units)	Data will be received in byte units using the non-procedure protocol. No response is received from the remote station.

[e] Cir-No (Circuit Number)

The Cir-No input specifies the circuit number of the transmission device.
Set the circuit number to the value displayed in the **MPE720 Module Configuration** Window.



The valid circuit number ranges are given in the following table.

Device	Valid Circuit Numbers
MPLINK/CP-215 (MPLINK)	1 to 8
RS-232C/422/485 (217IF)	1 to 16
Ethernet (218IF)	1 to 8
Ethernet (218IFB)	1 to 8

[f] Ch-No (Channel Number)

The Ch-No input specifies the channel number of the transmission buffer.

Any channel number can be specified as long as it is within the valid range. If more than one function is being executed at the same time, do not specify the same channel number more than once for the same modem number. (The same channel number can be used as long as the functions are not executed at the same time.)

The valid channel number ranges are given in the following table.

Device	Valid Channel Numbers
MPLINK/CP-215 (MPLINK)	1 to 12
RS-232C/422/485 (217IF)	1
Ethernet (218IF)	1 to 10
Ethernet (218IFB)	1 to 10

<Example>

If an Ethernet (218IF or 218IFB) transmission device is used, there are 10 channels of transmission buffers from 1 to 10 for sending and receiving combined, so up to 10 messages can be sent and received at the same time.

- One MSG-RCV (or MSG-SND) function must be programmed for each circuit being used at the same time.
- Refer to *A.3 Transmission Buffer Channels* on page A-9 for information on transmission buffer channels.

[g] Param (Parameter List Leading Address)

The PARAM input specifies the leading address of the parameter list. A parameter list will be automatically created from the 17 words starting with the specified address. Use the parameter list to input the function code and other related parameters. The processing results and status are also output to the parameter list.

- Refer to *5.8.3 Parameter List for Ethernet Communication (218IF or 218IFB)* on page 5-83 for information on the parameter list.

Example: The following parameter list will be created when the Parameter List Leading Address is set to DA000000.

Register	Parameter F · · · · · · · 0
DW000000	PARAM00
DW000001	PARAM01
DW000002	PARAM02
DW000003	PARAM03
DW000004	PARAM04
DW000005	PARAM05
DW000006	PARAM06
DW000007	PARAM07
DW000008	PARAM08
DW000009	PARAM09
DW000010	PARAM10
DW000011	PARAM11
DW000012	PARAM12
DW000013	PARAM13
DW000014	PARAM14
DW000015	PARAM15
DW000016	PARAM16

(2) Outputs

The following table gives the registers that can be used for the outputs.

Inputs	I/O Designation	Applicable Registers
Busy Complete Error	B-VAL	Any bit registers (including those with subscripts) except for # and C registers

The following sections describe the outputs in more detail.

[a] Busy (Processing in Progress)

Specify the bit that will report when receiving the message is being processed.
The Busy output will be ON while message receive processing or abort processing is in progress.
Keep the Execute command or Abort command ON while the Busy output is ON.

[b] Complete (Processing Completed)

Specify the bit that will report when processing to receive the message has been completed.
The Complete output will be ON for only one scan after message receive processing or abort processing has been completed normally.

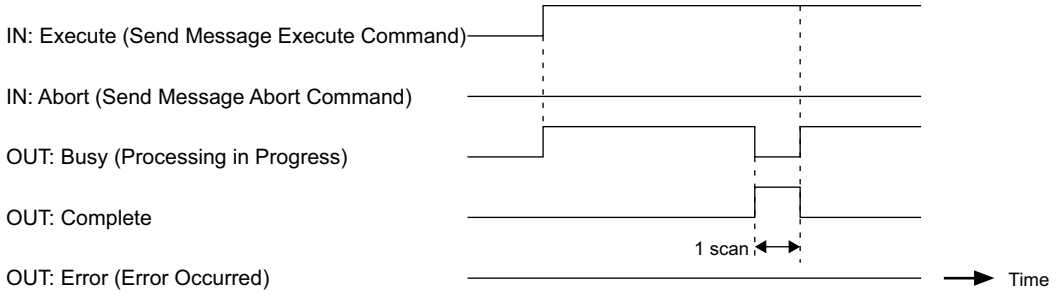
[c] ERROR (Error Occurred)

Specify the bit that will report when an error has occurred while receiving the message.
The Error output will turn ON for only one scan when an error occurs.

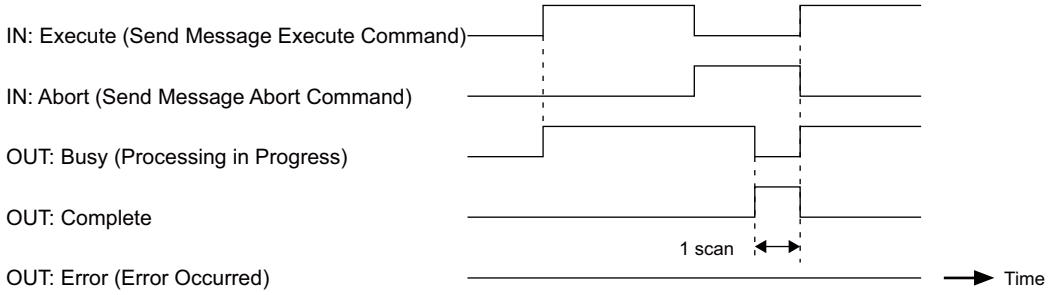
(3) I/O Timing Chart

The following timing charts show the bit inputs and outputs used with the MSG-RCV function.

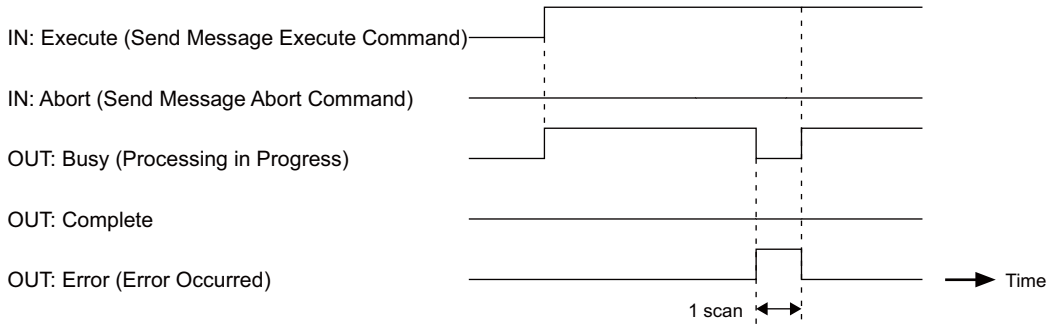
<Normal Processing>



<Abort Processing>



<Error Processing>



5.8.3 Parameter List for Ethernet Communication (218IF or 218IFB)

The Param input to the MSG-RCV function is a parameter list structure consisting of 17 words. The value of the Param input is the leading address (MA or DA) of the parameter list.
Use the parameter list to input the connection number, function code, and other related parameters. The processing results and status are also output to the parameter list.
The parameter lists for the MEMOBUS and non-procedure communication protocols are given below for 218IF or 218IFB Ethernet communication.

■ Parameter List for MEMOBUS (Extended MEMOBUS, MELSEC or MODBUS/TCP) Protocol

PARAM No.	IN/OUT*1	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	IN	Connection number	Specifies the remote station.
03	OUT	Option	Options specific to each transmission device are output here.
04	OUT	Function code	The function code requested by the sending station is output here.
05	OUT*2	Data address	The leading address of the data requested by the sending station is output here.
06	OUT	Data size	The read or write data size requested by the sending station is output here.
07	OUT	Remote CPU number	The remote CPU number is output here.
08	IN	Coil offset	Sets the offset word address of the coils.
09	IN	Input relay offset	Sets the offset word address of the input relays.
10	IN	Input register offset	Sets the offset word address of the input registers.
11	IN	Holding register offset	Sets the offset word address of the holding registers.
12	IN	Write range LO	Set the leading address of the write range.
13	IN	Write range HI	Set the final address of the write range.
14	SYS	Reserved by the system (1).	
15 and 16	SYS	Reserved by the system (2).	

* 1. IN: Input, OUT: Output, SYS: Used by the system.
* 2. IN/OUT for MODBUS/TCP.

- Refer to *5.8.4 Function Setting Example and Parameter Details for the MEMOBUS and Extended MEMOBUS Protocol* on page 5-84, *5.8.5 Function Setting Example and Parameter Details for the MELSEC Protocol* on page 5-91, and *5.8.6 Function Setting Example and Parameter Details for the MODBUS/TCP Protocol* on page 5-95 for details on the parameters.

■ Parameter List for Non-procedure Protocol

PARAM No.	IN/OUT*	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	IN	Connection number	Specifies the remote station.
03	OUT	(Not used.)	
04	OUT	(Not used.)	
05	OUT	(Not used.)	
06	OUT	Data size	Set the data size to be read or written.
07	OUT	(Not used.)	
08 to 11	IN	(Not used.)	
12	IN	Register offset	Sets the offset word address of the registers.
13	IN	Write range HI	Set the final address of the write range.
14	SYS	Reserved by the system (1).	
15 and 16	SYS	Reserved by the system (2).	

* IN: Input, OUT: Output, SYS: Used by the system.

- Refer to *5.8.7 Function Setting Example and Parameter Details for the Non-procedure Protocol* on page 5-100 for details on the parameters.

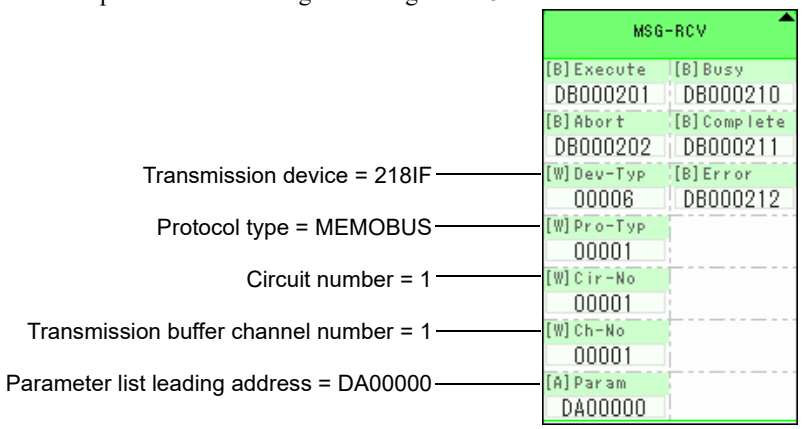
5.8.4 Function Setting Example and Parameter Details for the MEMOBUS and Extended MEMOBUS Protocol

This section provides an MSG-RCV function setting example and describes the parameter list in detail for Ethernet communication (218IF or 218IFB) using the MEMOBUS or Extended MEMOBUS protocol.

(1) MSG-RCV Function Setting Example

<218IF Setting Example>

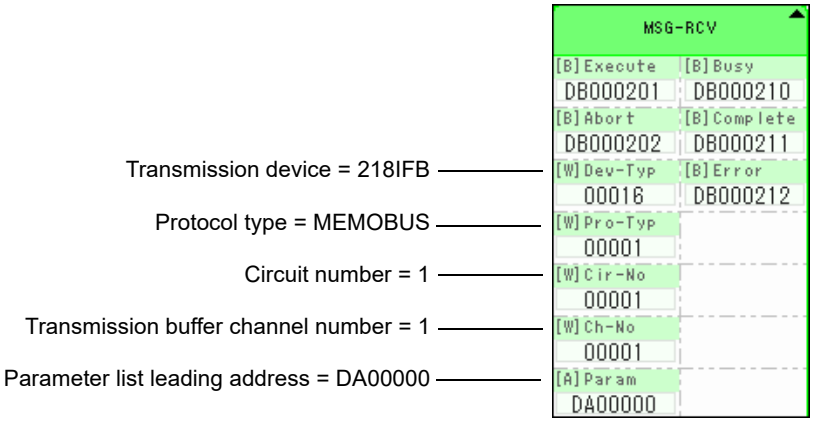
This example shows the settings for using the 218IF as the transmission device.



To use the Extended MEMOBUS protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.8.2 Inputs and Outputs for the Message Receive Function* on page 5-79.

<218IFB Setting Example>

This example shows the settings for using the 218IFB as the transmission device.



To use the Extended MEMOBUS protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IFB transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.8.2 Inputs and Outputs for the Message Receive Function* on page 5-79.

(2) Parameter List for MEMOBUS or Extended MEMOBUS Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	08	IN	Coil offset
01	OUT	Status	09	IN	Input relay offset
02	IN	Connection number	10	IN	Input register offset
03	OUT	Option	11	IN	Holding register offset
04	OUT	Function Code	12	IN	Write range LO
05	OUT	Data address	13	IN	Write range HI
06	OUT	Data size	14	SYS	Reserved by the system (1).
07	OUT	Remote CPU number	15 and 16	SYS	Reserved by the system (2).

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for MEMOBUS or Extended MEMOBUS Protocol

This section describes the parameter list in detail for using the MEMOBUS or Extended MEMOBUS protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARM00. The lower-place byte is for system analysis.

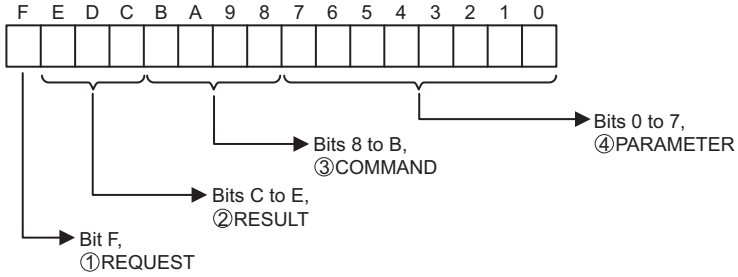
Value of Processing Result	Meaning
00xxH	Processing in progress (busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	Function code error	An undefined function code was received. Check the function code that was received.
82xxH	Address setting error	One of the following settings is not within the valid range. Check the settings. Data address (in request from sender) PARAM08 (coil offset) PARAM09 (input relay offset) PARAM10 (input register offset) PARAM11 (holding register offset)
83xxH	Data size error	The size of the sent or received data is not within the allowable range. Check PARAM06 (data size).
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-RCV function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-RCV function.
86xxH	Connection number error	The connection number is not within the allowable range. Check PARAM02 (connection number).
87xxH	–	–
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-RCV function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-RCV function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-RCV function is output to these bits.

Code	Abbreviation	Meaning
0	CONN_NG	Sending or connection has ended abnormally for Ethernet communication.
1	SEND_OK	Sending has been completed normally.
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-RCV function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send General-purpose Message with Non-procedure Protocol
2	U_REC	Receive General-purpose Message with Non-procedure Protocol
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response.
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response.
C	MR_SEND	Send MEMOBUS response

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the connection number is output.

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Connection number out of range
	02	MEMOBUS response receive monitor time error
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error
Other values of RESULT	XX	Connection Number

[c] PARAM02: Connection Number

PARAM02 specifies the sending station.
Set the connection number when the transmission device is Ethernet (218IF or 218IFB). The valid setting ranges are given in the following table.

Transmission Device	Connection Number	Remarks
Ethernet (218IF or 218IFB)	1 to 20	The message is received from the remote station with the designated connection number.

- Enter the same connection number as the one set in the 218IF/218IFB Transmission Parameters Tab Page or the MPE720 Module Configuration Window.

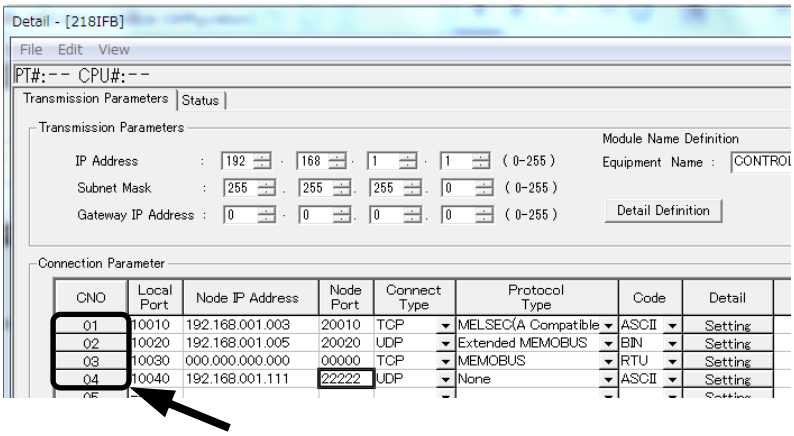


Fig. 5.11 The 218IF Parameter Setting Tab Page of the MPE720 Module Configuration Window

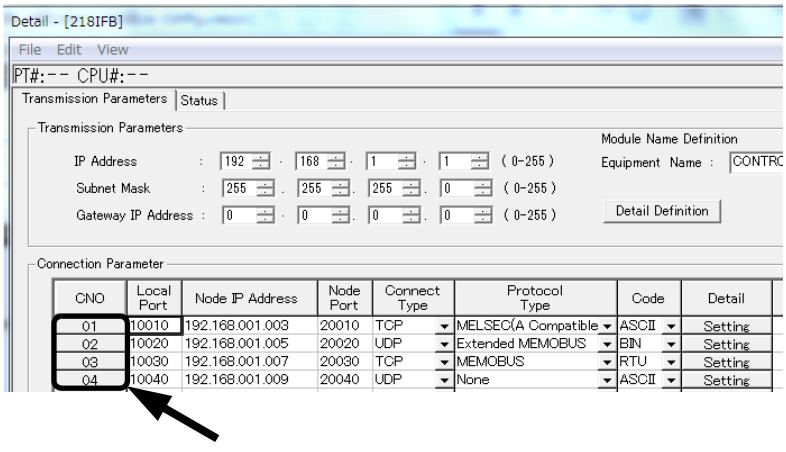


Fig. 5.12 The 218IFB Parameter Setting Tab Page of the MPE720 Module Configuration Window

[d] PARAM03: Option

Options specific to each transmission device are output here.
PARAM03 is not used for the MEMOBUS or Extended MEMOBUS protocol.

[e] PARAM04: Function Code

The function code that was received is output to PARAM04.
The function codes used for the MEMOBUS or Extended MEMOBUS protocol are listed in the following table.

<Function Codes for MEMOBUS or Extended MEMOBUS>

Function Code	Applicable Data Type	Function	Protocol Type	
			Extended MEMOBUS	MEMOBUS
00H	−	Not used.	−	−
01H	B	Read Coil Status	○	○
02H	B	Read Input Relay Status	○	○
03H	W	Read Holding Register Contents	○	○
04H	W	Read Input Register Contents	○	○
05H	B	Modify Status of a Single Coil	○	○
06H	W	Write to a Single Holding Register	○	○
07H	−	Not used.	−	−
08H	−	Loopback Test	○	○
09H	W	Read Holding Register Contents (Extended)	○	×
0AH	W	Read Input Register Contents (Extended)	○	×
0BH	W	Write to Holding Register (Extended)	○	×
0CH	−	Not used.	−	−
0DH	W	Read Discontinuous Holding Register Contents (Extended)	○	×
0EH	W	Write to Discontinuous Holding Registers (Extended)	○	×
0FH	B	Modify Status of Multiple Coils	○	○
10H	W	Write to Multiple Holding Registers	○	○

- B: Bit data, W: Word data
- ○: Can be set. ×: Cannot be set.
- Only MW (MB) registers may be used for sending and receiving messages when operating as the Master.
- MB, MW, IB, and IW registers can be used as coils, holding registers, input relays, and input registers, respectively, when operating as a Slave.

[f] PARAM05: Data Address

The requested data address is output to PARAM05.

[g] PARAM06: Data Size

The data size requested by the sending station to be read or written as the number of bits or words is output to PARAM06.

[h] PARAM07: Remote CPU Number

The remote CPU number will be output as 1 if the remote device is an MP2000 Series Machine Controller.
If the remote device is a Yaskawa Controller, but not in the MP2000 Series and it consists of more than one CPU Module, the destination CPU number will be output.
In all other cases, 0 will be output.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the data address offsets from the sending station.
The position of the data is shifted backward by the number of words set for the offset at the device receiving the message.

- For details, refer to *A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV* on page A-5.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter	Protocol Type	
			Extended MEMOBUS	MEMOBUS
01H	Read Coil Status	PARAM08	○	○
02H	Read Input Relay Status	PARAM09	○	○
03H	Read Holding Register Contents	PARAM11	○	○
04H	Read Input Register Contents	PARAM10	○	○
05H	Modify Status of a Single Coil	PARAM08	○	○
06H	Write to a Single Holding Register	PARAM11	○	○
09H	Read Holding Register Contents (Extended)	PARAM11	○	×
0AH	Read Input Register Contents (Extended)	PARAM10	○	×
0BH	Write to Holding Register (Extended)	PARAM11	○	×
0DH	Read Discontinuous Holding Register Contents (Extended)	PARAM11	○	×
0EH	Write to Discontinuous Holding Registers (Extended)	PARAM11	○	×
0FH	Modify Status of Multiple Coils	PARAM08	○	○
10H	Write to Multiple Holding Registers	PARAM11	○	○

- ○: Can be set. ×: Cannot be set.

[j] PARAM12 and PARAM13: Write Range

This parameter sets the range of addresses that can be written for a write requested from the sending station. Any write request that exceeds this address range will result in an error.
The write ranges in PARAM12 and PARAM13 are set as word addresses.

- The MP3000/MP2000 Series Machine Controllers store all data for write requested from the sending station to M registers.

The write range parameter can be used to set a range of M registers that can be written using messages.
The write range parameters are listed in the following table.

Parameter	Contents	Description
PARAM12	Write range LO	Leading address of the write range
PARAM13	Write range HI	Final address of the write range

Set the write ranges so that the following conditions are satisfied.

$0 \leq$ Write range LO $\leq$ Write range HI $\leq$ Maximum value of M addresses

The write range is valid for the following function codes.

05H (Modify Status of a Single Coil)
06H (Write to a Single Holding Register)
0BH (Write to Holding Register (Extended))
0EH (Write to Discontinuous Holding Registers (Extended))
0FH (Modify Status of Multiple Coils)
10H (Write to Multiple Holding Registers)

Example: The following settings would be used to permit writing to the M registers with addresses 1000 to 1999.

PARAM12=1000
PARAM13=1999

With these settings, the receiving station will output an error and not perform the write if a write request is received for any registers not between MW01000 and MW01999.

[k] PARAM14: Reserved by the System (1)

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[l] PARAM15 and PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

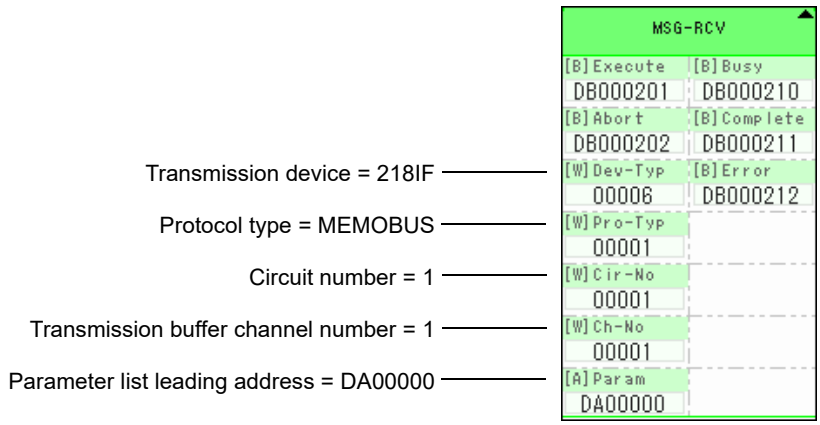
5.8.5 Function Setting Example and Parameter Details for the MELSEC Protocol

This section provides an MSG-RCV function setting example and describes the parameter list in detail for Ethernet communication (218IF or 218IFB) using the MELSEC protocol.

(1) MSG-RCV Function Setting Example

<218IF Setting Example>

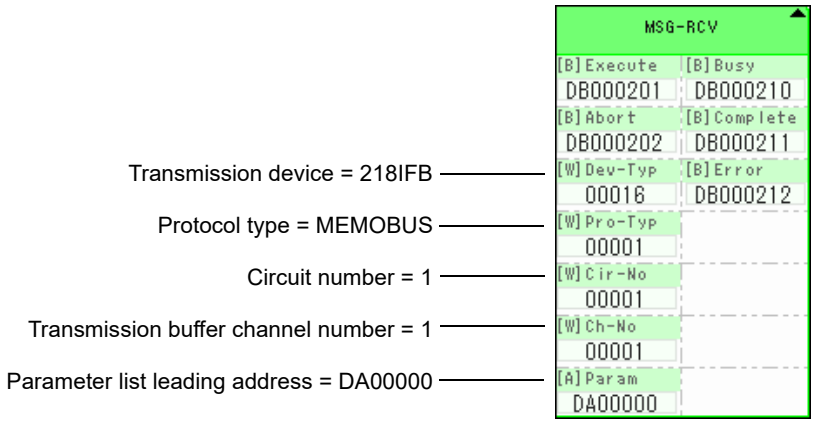
This example shows the settings for using the 218IF as the transmission device.



To use the MELSEC protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.8.2 Inputs and Outputs for the Message Receive Function* on page 5-79.

<218IFB Setting Example>

This example shows the settings for using the 218IFB as the transmission device.



To use the MELSEC protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IFB transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.8.2 Inputs and Outputs for the Message Receive Function* on page 5-79.

(2) Parameter List for MELSEC Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	08	IN	Coil offset
01	OUT	Status	09	IN	Input relay offset
02	IN	Connection number	10	IN	Input register offset
03	OUT	Option	11	IN	Holding register offset
04	OUT	Function code	12	IN	Write range LO
05	OUT	Data address	13	IN	Write range HI
06	OUT	Data size	14	SYS	Reserved by the system (1).
07	OUT	Remote CPU number	15 and 16	SYS	Reserved by the system (2).

IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for MELSEC Protocol

This section describes the parameter list in detail for using the MELSEC protocol.

[a] PARAM00: Processing Result

Refer to *5.8.4 (3) [a] PARAM00: Processing Result* on page 5-85.

[b] PARAM01: Status

Refer to *5.8.4 (3) [b] PARAM01: Status* on page 5-86.

[c] PARAM02: Connection Number

Refer to *5.8.4 (3) [c] PARAM02: Connection Number* on page 5-87.

[d] PARAM03: Option

Options specific to each transmission device are output here.
PARAM03 is not used for the MELSEC protocol.

[e] PARAM04: Function Code

The function code that was received is output to PARAM04.
The function codes used for the MELSEC protocol are listed in the following table.

<Function Codes for MELSEC>

Common MELSEC ACPU Command	Function Code	Applicable Data Type	Function
00H	01H/02H	B	Reads data from bit-type devices in 1-point units.
01H	03H/04H/09H/0AH	W	Reads data from word-type devices in 1-point units.
02H	05H/0FH	B	Writes data to bit-type devices in 1-point units.
03H	06H/0BH/10H	W	Writes data to word-type devices in 1-point units.
16H	08H	–	Executes a loopback test.
05H	0EH	B	Sets/resets data in word devices specified at random in 1-point units.
60H	31H	W	Writes data in word units to the fixed buffers.
61H	32H	W	Reads data in word units from the random-access buffers. (Cannot be received by an MP2000 Series Machine Controller.)
62H	33H	W	Writes data in word units to the random-access buffers.

- B: Bit data, W: Integer data
- The special commands for the AnCPU are not supported. Use the common ACPU commands listed above when accessing an AnCPU. Accessing the extended file registers in an AnCPU is not possible.

[f] PARAM05: Data Address

The requested data address is output to PARAM05.

[g] PARAM06: Data Size

The data size requested by the sending station to be read or written as the number of bits or words is output to PARAM06.

[h] PARAM07: Remote CPU Number

Refer to *5.8.4 (3) [h] PARAM07: Remote CPU Number* on page 5-88.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the data address offsets from the sending station.
The position of the data is shifted backward by the number of words set for the offset at the device receiving the message.

- For details, refer to *A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV* on page A-5.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter
01H	Read Coil Status	PARAM08
02H	Read Input Relay Status	PARAM09
03H	Read Holding Register Contents	PARAM11
04H	Read Input Register Contents	PARAM10
05H	Modify Status of a Single Coil	PARAM08
06H	Write to a Single Holding Register	PARAM11
09H	Read Holding Register Contents (Extended)	PARAM11
0AH	Read Input Register Contents (Extended)	PARAM10
0BH	Write to Holding Register (Extended)	PARAM11
0DH	Read Discontinuous Holding Register Contents (Extended)	PARAM11
0EH	Write to Discontinuous Holding Registers (Extended)	PARAM11
0FH	Modify Status of Multiple Coils	PARAM08
10H	Write to Multiple Holding Registers	PARAM11

[j] PARAM12 and PARAM13: Write Range

This parameter sets the range of addresses that can be written for a write requested from the sending station. Any write request that exceeds this address range will result in an error.
The write ranges in PARAM12 and PARAM13 are set as word addresses.

- The MP3000/MP2000 Series Machine Controllers store all data for write requested from the sending station to M registers.

The write range parameter can be used to set a range of M registers that can be written using messages.
The write range parameters are listed in the following table.

Parameter	Contents	Description
PARAM12	Write range LO	Leading address of the write range
PARAM13	Write range HI	Final address of the write range

Set the write ranges so that the following conditions are satisfied.

0 ≤ Write range LO ≤ Write range HI ≤ Maximum value of M addresses

The write range is valid for the following function codes.

05H (Modify Status of a Single Coil)
06H (Write to a Single Holding Register)
0BH (Write to Holding Register (Extended))
0EH (Write to Discontinuous Holding Registers (Extended))
0FH (Modify Status of Multiple Coils)
10H (Write to Multiple Holding Registers)
31H (Write to Fixed Buffers)
33H (Write to Random-access Buffers)

Example: The following settings would be used to permit writing to the M registers with addresses 1000 to 1999.
PARAM12=1000
PARAM13=1999
With these settings, the receiving station will output an error an not perform the write if a write request is received for any registers not between MW01000 and MW01999.

[k] PARAM14: Reserved by the System (1)

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[l] PARAM15 and PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

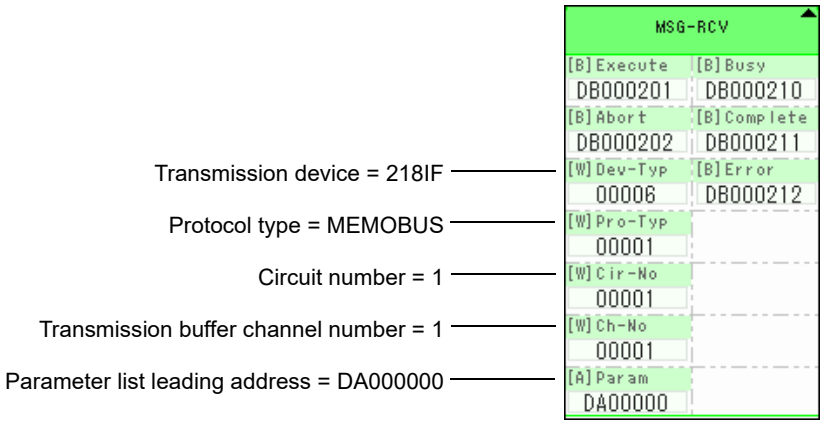
5.8.6 Function Setting Example and Parameter Details for the MODBUS/TCP Protocol

This section provides an MSG-RCV function setting example and describes the parameter list in detail for Ethernet communication (218IF or 218IFB) using the MODBUS/TCP protocol.

(1) MSG-RCV Function Setting Example

<218IF Setting Example>

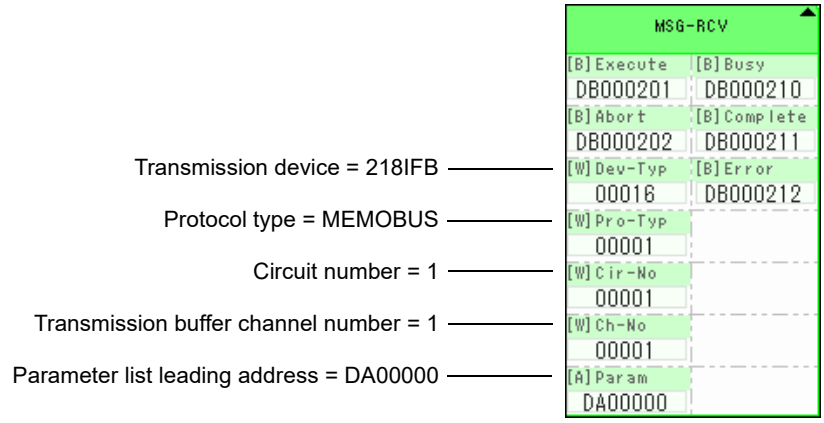
This example shows the settings for using the 218IF as the transmission device.



To use the MODBUS/TCP protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.8.2 Inputs and Outputs for the Message Receive Function* on page 5-79.

<218IFB Setting Example>

This example shows the settings for using the 218IFB as the transmission device.



To use the MODBUS/TCP protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 218IFB transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.8.2 Inputs and Outputs for the Message Receive Function* on page 5-79.

(2) Parameter List for MODBUS/TCP Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	08	IN	Coil offset
01	OUT	Status	09	IN	Input relay offset
02	IN	Connection number	10	IN	Input register offset
03	OUT	Option	11	IN	Holding register offset
04	OUT	Function code	12	IN	Write range LO
05	IN/OUT	Data address	13	IN	Write range HI
06	OUT	Data size	14	SYS	Reserved by the system (1).
07	OUT	Remote CPU number	15 and 16	SYS	Reserved by the system (2).

IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for MODBUS/TCP Protocol

This section describes the parameter list in detail for using the MODBUS/TCP protocol.

[a] PARAM00: Processing Result

Refer to *5.8.4 (3) [a] PARAM00: Processing Result* on page 5-85.

[b] PARAM01: Status

Refer to *5.8.4 (3) [b] PARAM01: Status* on page 5-86.

[c] PARAM02: Connection Number

Refer to *5.8.4 (3) [c] PARAM02: Connection Number* on page 5-87.

[d] PARAM03: Option

Options specific to each transmission device are output here.
The output contents is as shown below.

Transmission Device	Output Contents
Ethernet (218IF or 218IFB)	The remote unit ID for communication using MODBUS/TCP protocol is output.

[e] PARAM04: Function Code

The function code that was received is output to PARAM04.
The function codes used for the MODBUS/TCP protocol are listed in the following table.

<Function Codes for MODBUS/TCP>

Function Code	Applicable Data Type	Function	Protocol Type	
			Ethernet (218IF)	Ethernet (218IFB)
00H	–	Not used.	–	–
01H	B	Read Coil Status	○	○
02H	B	Read Input Relay Status	○	○
03H	W	Read Holding Register Contents	○	○
04H	W	Read Input Register Contents	○	○
05H	B	Modify Status of a Single Coil	○	○
06H	W	Write to a Single Holding Register	○	○
07H : 0EH	–	Not used.	–	–
0FH	B	Modify Status of Multiple Coils	○	○
10H	W	Write to Multiple Holding Registers	○	○
11H : 15H	–	Not used.	–	–
16H	W	Mask Write to Holding Registers	×	○
17H	W	Read/Write to Multiple Holding Registers	×	○

- B: Bit data, W: Word data
- ○: Can be set. ×: Cannot be set.
- Only MW (MB) registers may be used for sending and receiving messages when operating as the Master.
- MB, MW, IB, and IW registers can be used as coils, holding registers, input relays, and input registers, respectively, when operating as a Slave.

[f] PARAM05: Data Address

The requested data address is output to PARAM05.
However, when reading/writing multiple holding registers (function code 17H), input the leading M register number of the address table. The M register specified by PARAM05 (data address) specifies the start of a four-word address table. The requested read address, read size, write address, and write size are output to this address table.
For details on the address table, refer to *5.7.6 (3) [f] PARAM05: Data Address* on page 5-70.

[g] PARAM06: Data Size

The data size requested by the sending station to be read or written as the number of bits or words is output to PARAM06.

[h] PARAM07: Remote CPU Number

Refer to *5.8.4 (3) [h] PARAM07: Remote CPU Number* on page 5-88.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the data address offsets from the sending station.
The position of the data is shifted backward by the number of words set for the offset at the device receiving the message.

- For details, refer to *A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV* on page A-5.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter
01H	Read Coil Status	PARAM08
02H	Read Input Relay Status	PARAM09
03H	Read Holding Register Contents	PARAM11
04H	Read Input Register Contents	PARAM10
05H	Modify Status of a Single Coil	PARAM08
06H	Write to a Single Holding Register	PARAM11
0FH	Modify Status of Multiple Coils	PARAM08
10H	Write to Multiple Holding Registers	PARAM11
16H	Mask Write to Holding Registers	PARAM11
17H	Read/Write to Multiple Holding Registers	PARAM11

[j] PARAM12 and PARAM13: Write Range

This parameter sets the range of addresses that can be written for a write requested from the sending station. Any write request that exceeds this address range will result in an error.
The write ranges in PARAM12 and PARAM13 are set as word addresses.

- The MP3000/MP2000 Series Machine Controllers store all data for write requested from the sending station to M registers.

The write range parameter can be used to set a range of M registers that can be written using messages.
The write range parameters are listed in the following table.

Parameter	Contents	Description
PARAM12	Write range LO	Leading address of the write range
PARAM13	Write range HI	Final address of the write range

Set the write ranges so that the following conditions are satisfied.

0 ≤ Write range LO ≤ Write range HI ≤ Maximum value of M addresses

The write range is valid for the following function codes.

05H (Modify Status of a Single Coil)
06H (Write to a Single Holding Register)
0BH (Write to Holding Register (Extended))
0FH (Modify Status of Multiple Coils)
10H (Write to Multiple Holding Registers)
16H (Mask Write to Holding Registers)
17H (Read/Write to Multiple Holding Registers)

Example: The following settings would be used to permit writing to the M registers with addresses 1000 to 1999.

PARAM12=1000
PARAM13=1999

With these settings, the receiving station will output an error and not perform the write if a write request is received for any registers not between MW01000 and MW01999.

[k] PARAM14: Reserved by the System (1)

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[l] PARAM15 and PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

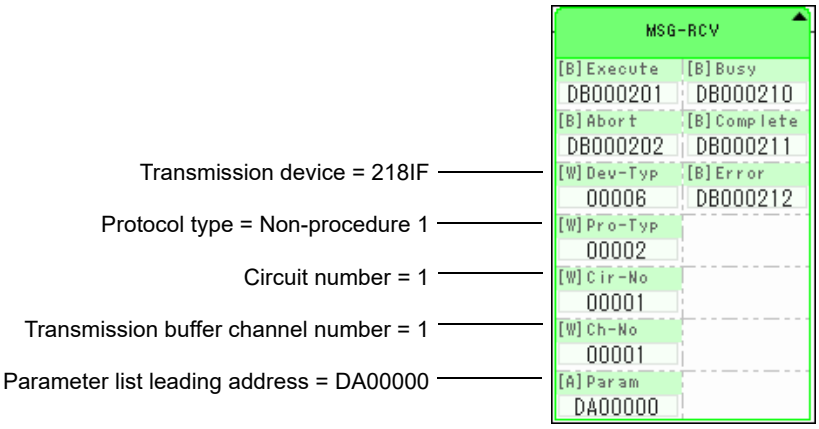
5.8.7 Function Setting Example and Parameter Details for the Non-procedure Protocol

With the non-procedure protocol, the received data is not converted and stored as is in the M registers. This enables receiving data using any protocol of the remote station.
This section provides an MSG-RCV function setting example and describes the parameter list in detail for Ethernet communication (218IF or 218IFB) using the non-procedure protocol.

(1) MSG-RCV Function Setting Example

<218IF Setting Example>

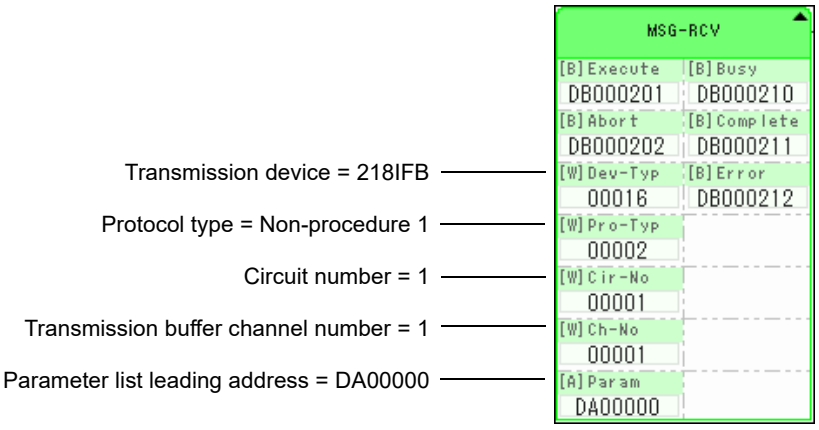
This example shows the settings for using the 218IF as the transmission device.



The protocol type (Pro-Typ) is set to 2 to use non-procedure protocol 1 and to 3 to use non-procedure protocol 2.
The circuit number is set to the circuit number assigned to the 218IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.8.2 Inputs and Outputs for the Message Receive Function* on page 5-79.

<218IFB Setting Example>

This example shows the settings for using the 218IFB as the transmission device.



The protocol type (Pro-Typ) is set to 2 to use non-procedure protocol 1 and to 3 to use non-procedure protocol 2.
The circuit number is set to the circuit number assigned to the 218IFB transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *5.8.2 Inputs and Outputs for the Message Receive Function* on page 5-79.

(2) Parameter List for Non-procedure Protocol

Param No.	IN/OUT	Contents	Param No.	IN/OUT	Contents
00	OUT	Processing result	07	OUT	(Not used.)
01	OUT	Status	08 to 11	IN	(Not used.)
02	IN	Connection number	12	IN	Write range LO
03	OUT	(Not used.)	13	IN	Write range HI
04	OUT	(Not used.)	14	SYS	Reserved by the system (1).
05	OUT	(Not used.)	15 and 16	SYS	Reserved by the system (2).
06	OUT	Data size			

- IN: Input, OUT: Output, SYS: Used by the system.

(3) Parameter Details for Non-procedure Protocol

This section describes the parameter list in detail for using the non-procedure protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARAM00. The lower-place byte is for system analysis.

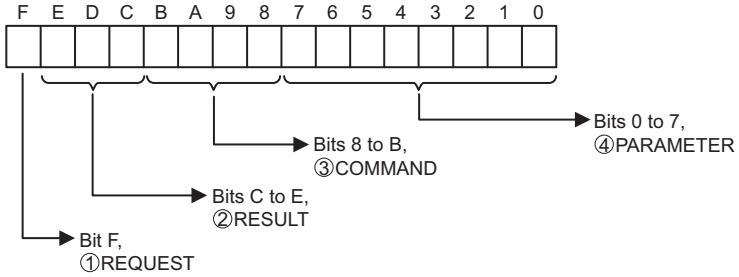
Value of Processing Result	Meaning
00xxH	Processing in progress (busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	–	–
82xxH	Address setting error	The following setting is not within the valid range. Check the settings. PARAM12 (register offset)
83xxH	Data size error	The size of the received data is not within the allowable range. Check the data size from the sending station.
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-RCV function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-RCV function.
86xxH	Connection number error	The connection number is not within the allowable range. Check PARAM02 (connection number).
87xxH	–	Reserved by the system.
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-RCV function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-RCV function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-RCV function is output to these bits.

Code	Abbreviation	Meaning
0	CONN_NG	Sending or connection has ended abnormally for Ethernet communication.
1	SEND_OK	Sending has been completed normally.
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-RCV function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send general-purpose message.
2	U_REC	Receive general-purpose message.
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response. (For MEMOBUS Protocol)
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response. (For MEMOBUS Protocol)
C	MR_SEND	Send MEMOBUS response (For MEMOBUS Protocol)

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the connection number is output.

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Connection number out of range
	02	MEMOBUS response receive monitor time error (for MEMOBUS Protocol)
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error (for MEMOBUS Protocol)
Other values of RESULT	XX	Connection Number

[c] PARAM02: Connection Number

Refer to *5.8.4 (1) [c] PARAM02: Connection Number* on page 5-87.

[d] PARAM06: Data Size

The size of received data is output to this parameter.
In the non-procedure 1 communication mode, the data size is output as the number of words.
In the non-procedure 2 communication mode, the data size is output as the number of bytes.

[e] PARAM13: Write Range HI

With the non-procedure protocol, the continuous received data is stored in M registers.
An error will occur if there is a request to write a value that does not correspond to the addresses from PARAM12 (Write Range LO) to PARAM13 (Write Range HI).
Set PARAM12 and PARAM13 (Write Range) as word addresses.

Set a write range that satisfies the following equation.

$0 \leq$ Write range $\leq$ Maximum M register address

Example: To set the range of M registers addresses that can be written from 1000 to 1999.

PARAM12=1000
PARAM13=1999

[f] PARAM14: Reserved by the System (1)

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[g] PARAM15 and PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

6

217IF-01 Module

This chapter describes the 217IF-01 Module in detail.

- Before reading this chapter, be sure to read *Chapter 1 Starting Communication Modules*.

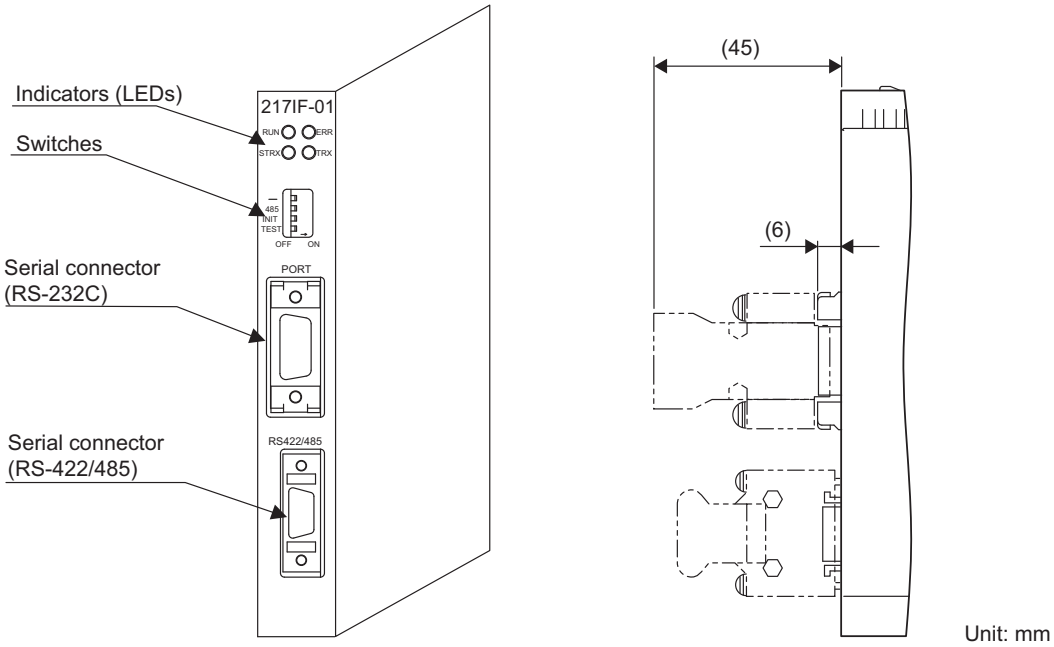
6.1 Overview ----------6-2
6.1.1 Appearance and Connectors ---------- 6-2
6.1.2 System Configuration Examples ---------- 6-3
6.1.3 Module Specifications ---------- 6-4
6.1.4 Indicators ---------- 6-6
6.1.5 Switch Settings ---------- 6-7
6.1.6 Default Parameters at Initial Startup ---------- 6-7
6.1.7 Online Switching of Transmission Parameter Definitions by Setting INIT Switch (Software Version 1.10 or Later) ---------- 6-8
6.2 Connection Specifications ----------6-9
6.2.1 Connector Specifications ---------- 6-9
6.2.2 Cable Specifications ---------- 6-10
6.2.3 Module Connection Examples ---------- 6-11
6.3 Serial Communication ----------6-12

6.1 Overview

The 217IF-01 Module is a Communication Module with two serial interfaces: RS-232C and RS-422/485. Personal computers, HMI devices, and controllers manufactured by other companies can be connected to the 217IF-01 Module via the PORT (RS-232C) or RS-422/485 connectors. Communication modes include message communication and engineering communication, and MEMOBUS, MELSEC, OMRON, and non-procedure protocols are supported.
The rest of this chapter describes the appearance, specifications, and switch settings of the 217IF-01 Module.

6.1.1 Appearance and Connectors

The following diagram shows the appearance of the 217IF-01 Module and gives the external dimensions of the connectors.



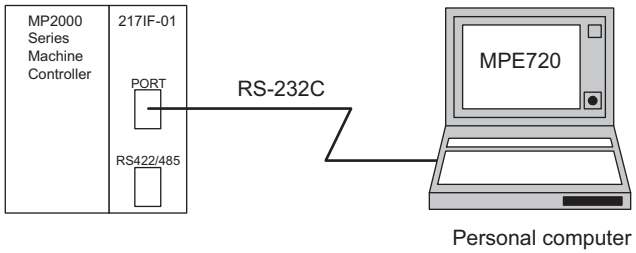
Unit: mm

6.1.2 System Configuration Examples

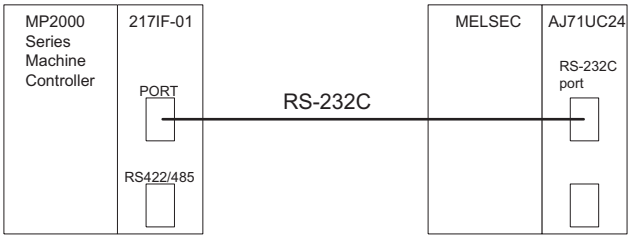
This section gives an overview of the configuration of a system that uses a 217IF-01 Module.

(1) Example of RS-232C Connections

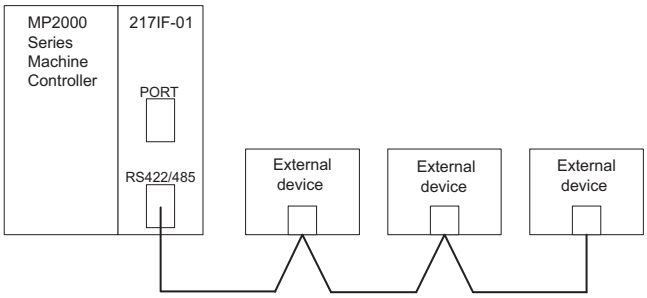
■ Connection to a Personal Computer Running MPE720



■ Connection to a Mitsubishi MELSEC A-series Controller



(2) Example of RS-422/485 Connections



6.1.3 Module Specifications

This section provides the specifications of the 217IF-01 Module.

(1) Hardware Specifications

Item	Specification
Name	217IF-01
Model Number	JAPMC-CM2310
Communication Ports	RS-232C: 1 port (PORT)
	RS-422/485: 1 port (RS-422/485)
Indicators	Module status LED indicators RUN (green) ERR (red) STRX (green) TRX (green)
Setting Switches	– 485 INIT TEST
Dimensions (mm)	125 x 95 mm (H x D)
Mass	90 g

(2) Transmission Specifications

■ RS-232C Transmission Specifications

Item	Specification
Interface	1 port (PORT)
Connector	9-pin D-sub (female)
Transmission Distance	15 m max.
Baud Rate	9.6/14.4/19.2/28.8/38.4/48.0/57.6/76.8 kbps
Access Mode	Asynchronous (start-stop synchronization)
Communication Type	Half duplex
Communication Modes	Message communication and engineering communication
Communication Protocols	MEMOBUS, MELSEC, OMRON, and Non-procedure
Media Access Control Method	1:1
Transmission Format (Can be set.)	Data bit length: 7 or 8 bits Stop bits: 1 or 2 bits Parity: Odd, even, or none

- Although the 217IF-01 Module can be used with a baud rate up to 76.8 kbps, connection may not be possible depending on the characteristics of the connected devices. If connection is not possible, decrease the setting of the baud rate.

■ RS-422/485 Transmission Specifications

Item	Specification
Interface	1 port (RS-422/485)
Transmission Distance	300 m max.
Baud Rate	4.8/9.6/14.4/19.2/28.8/38.4/48.0/57.6/76.8 kbps
Communication Protocols	MEMOBUS, MELSEC, OMRON, and Non-procedure
Connector	MDR14-pin (female)
Synchronization Mode	Asynchronous (start-stop synchronization)
Media Access Control Method	1:1 (RS-422), 1:N (RS-485)*
Transmission Format (Can be set.)	Data bit length: 7 or 8 bits Stop bits: 1 or 2 bits Parity: Odd, even, or none

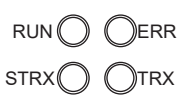
* N: 31 units maximum

(3) Operating Environment Specifications

Item		Specification
Environmental Conditions	Ambient Operating Temperature	0 to 55°C
	Ambient Storage Temperature	–25 to 85°C
	Ambient Operating Humidity	30% to 95% (with no condensation)
	Ambient Storage Humidity	5% to 95% (with no condensation)
	Pollution Level	Pollution level 2 (conforming to JIS B 3502)
	Corrosive Gas	There must be no combustible or corrosive gas.
	Operating Altitude	2,000 m above sea level or lower
Mechanical Operating Conditions	Vibration Resistance	Conforming to JIS B 3502 Vibration and acceleration: 10 to 57 Hz with single-amplitude of 0.075 mm 57 to 150 Hz with fixed acceleration of 9.8 m/s^2 10 sweeps each in X, Y, and Z directions (sweep time: 1 octave/min)
	Shock Resistance	Conforming to JIS B 3502 Peak acceleration of 147 m/s^2 (15 G) twice for 11 ms each in the X, Y, and Z directions
Electrical Operating Conditions	Noise Resistance	Conforming to EN 61000-6-2, EN 61000-6-4, EN 55011 (Group 1 Class A)
Installation Requirements	Ground	Ground to 100 Ω max.
	Cooling Method	Natural cooling

6.1.4 Indicators

The following table shows the status of the 217IF-01 Module shown by the LED indicators.



Indicator	Color	Meaning When Lit	Meaning When Not Lit
RUN	Green	Operating normally	An error has occurred.
ERR	Red	A failure has occurred (lit or blinking).	Normal
STRX	Green	Sending or receiving RS-232C data (PORT connector).	No RS-232C data communication
TRX	Green	Sending or receiving RS-422/485 data (RS-422/485 connector).	No RS-422/485 data communication

■ Status Indication by LED Indicators

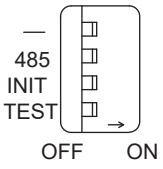
The following table shows the LED indications and their meanings for operating status and errors.

Status	Indicators				Meaning	Description
	RUN	ERR	STRX	TRX		
Power ON	○	●	○	○	Initializing	This is the status immediately after the power to the 217IF-01 is turned ON (initializing). When initialization has been completed, the ERR indicator goes out and the 217IF-01 enters initialized status.
Initialized	○	○	–	–	Waiting for initial settings	The 217IF-01 starts normally and waits for initial settings from the controller (CPU). The STRX and TRX indicators may be lit depending on the connection status.
Normal	●	○	–	–	Operating normally	This is the normal 217IF-01 operating status. The STRX and TRX indicators blink during serial communication.
Error	○	★	–	○	Hardware Error (No. of blinks:) 1: ROM diagnosis error 2: RAM diagnosis error 3: Shared memory diagnosis error 8: Slot illegal instruction error 9: CPU address error 11: User break error 12: Trap instruction error 15: Watchdog timeout error	The 217IF-01 hardware or processor failed. The Module must be replaced.

- ● : Lit
  ○ : Not lit
  ★ : Blinking
  – : Not relevant

6.1.5 Switch Settings

The following table shows the 217IF-01 Module switch settings.



Label	Name	Status	Function	Factory Setting
–	Reserved	–	Always leave set to OFF.	OFF
485	485 Mode	ON	Uses the RS-422/485 port for RS-485.	OFF
		OFF	Uses the RS-422/485 port for RS-422.	
INIT	Initial Startup	ON	For engineering communication. Starts the RS-232C (PORT) port using the default parameters except settings of automatic reception function. The RS-422/485 port is disabled. Given higher priority than the MP2000 Series Machine Controller Flash Startup and Self-configuration Startup.	OFF
		OFF	Set to OFF for MP2000 Series Machine Controller Flash Startup and Self-configuration Startup.	
TEST	TEST	ON	System use.	OFF
		OFF	Normal operation (Always leave set to OFF.)	

6.1.6 Default Parameters at Initial Startup

At initial startup (INIT switch set to ON), the following parameter settings will be made for the RS-232C interface.

■ RS-232C Interface (The setting of INIT switch to ON is valid while the power is being supplied.)

Item	Default Setting		
Communication Protocol	MEMOBUS		
Master/Slave	Slave		
Device Address	1		
Serial I/F	RS-232C		
Communication Mode	RTU		
Data Length	8 bits		
Parity	Even		
Stop bits	1 bit		
Baud Rate	19.2 kbps		
Send Delay	Disable		
Receive Monitor Time	Disable		
Automatic Reception	Enable		
Interface Register Settings in the Slave		Leading Register	Number of Words
	Read Input Relays	IW0000	32768
	Read Input Registers	IW0000	32768
	Read/Write Coils	MW00000	65535
	Read/Write Holding Registers	MW00000	65535
	Coil/Holding Register Write Range	LO MW00000	
		HI MW65534	

6.1.7 Online Switching of Transmission Parameter Definitions by Setting INIT Switch (Software Version 1.10 or Later)

The transmission parameters for RS-232C can be switched online from the default definitions to user-set definitions (i.e., definitions following the module configuration definitions) by turning OFF the INIT switch.
The 217IF-01 Module will operate as follows if the INIT switch setting is changed while the system is online.

Module Configuration Definitions: 217IF-01 Definitions	INIT Switch Status Change	217IF-01 Module Operation
Not set	Turned ON	Operation is performed using the default definitions.
	Turned OFF	Operation continues using the default definitions because there are no module configuration definitions.
Set	Turned ON	Operation is performed after changing from user-set definitions to default definitions.
	Turned OFF	Operation is performed after changing from default definitions to user-set definitions.

■ Precautions on Changing the INIT Switch Setting

The MSG-SND function will end in an error and communication will stop when the INIT switch is turned ON during message communication using the MSG-SND function in a user application. In that case, the command-response sequence may end in an error and communication may stop between the Module and the remote serial device. When the INIT switch is turned OFF while communication with the remote serial device is stopped, the MSG-SND function will start operation normally, but communication may not start again due to a sequence error between the Module and the remote serial device. In that case, set a timeout time and resend the MSG-SND function in the user application that executes the MSG-SND function.

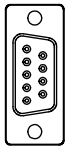
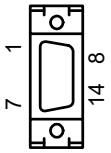
- The online switching of transmission parameter definitions by setting INIT switch is possible only when using RS-232C communication for the 217IF-01 Module.

6.2 Connection Specifications

6.2.1 Connector Specifications

This section provides the connector specifications for the 217IF-01 Module.

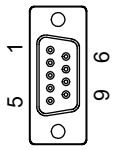
(1) Connector Specifications

Connector Shape	Name	Connector Name	No. of Pins	Connector Model		
				Module	Cable	Manufacturer
PORT 	RS-232C	PORT	9	17LE-13090-27(D2BC) D-sub 9-pin female connector	17JE-23090-02 (D8B) 9-pin D-sub male connector (Shell included.)	DDK Ltd.
RS422/485 	RS422/485 port	RS422/485	14	10214-52A2PL connector	• Connector body: 10114-3000PE • Shell: 10314-52A0-008 (Screw locking) 10314-52F0-008 (One-touch locking)	3M Japan Limited

(2) Connector Pin Arrangement

■ PORT Connector

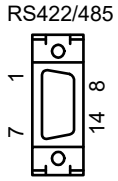
The PORT connector is used to connect the MP2000 Series Machine Controller to computers and HMI devices via an RS-232C connection.



Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	FG	Frame ground	6	–	–
2	SD	Send data	7	SG	Signal ground (0V)
3	RD	Receive data	8	–	–
4	RS	Ready to send	9	ER	Data terminal ready
5	CS	Clear to send	–	–	–

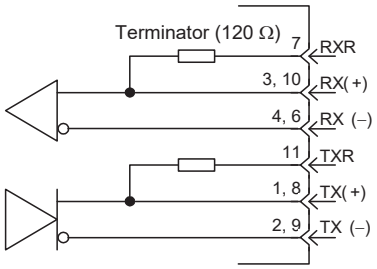
■ RS-422/485 Connector

The RS-422/485 connector is used to connect the 217IF-01 Module to computers and HMI devices via an RS-422/485 connection.



Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	TX+	Send data +	8	TX+	Send data +
2	TX–	Send data –	9	TX–	Send data –
3	RX+	Receive data +	10	RX+	Receive data +
4	RX–	Receive data –	11	TXR	Send data terminator
5	–	–	12	–	–
6	RX–	Receive data –	13	Reserved by the system.	Do not connect.
7	RXR	Receive data termina-tor	14	GND	Ground

- A terminator (120 Ω) has been included, as shown in the following diagram. If you need to add terminator, connect RXR to RX (–) and TXR to TX (–). Leave RXR and TXR open if not adding a terminator.



- Always keep the transmission cable separate from the drive, control, and power supply lines, and other transmission systems.
- The maximum length of RS-422/485 cable is 300 m. Keep all cables as short as possible.
- The 217IF-01 Module's RS-422/485 interface is not an isolated system. Noise from connected terminals may cause malfunction. If malfunctions occur, use a shielded cable, modem, or other measure to reduce noise.
- For RS-422 connections, add a terminator to the reception terminal if required.
- For RS-485 connections, add a terminator to the node at each end of the transmission path.

6.2.2 Cable Specifications

This section provides the cable specifications for the 217IF-01 Module.

(1) RS-232C Cable (for MPE720 Engineering)

Model Number	Length	External Appearance (JEPMC-W5311-□□-E)
JEPMC-W5311-03-E	2.5 m	
JEPMC-W5311-15-E	15 m	

(2) RS-422/485 Cable

A standard cable is not provided.
Make your own cable using a commercially available connector (10114-3000VE connector and 10314-52A0-008 shell, both from 3M Japan Limited) and a commercially available cable.

- The maximum cable length is 300 m. Use a shielded cable, modem, or other measure to reduce noise.

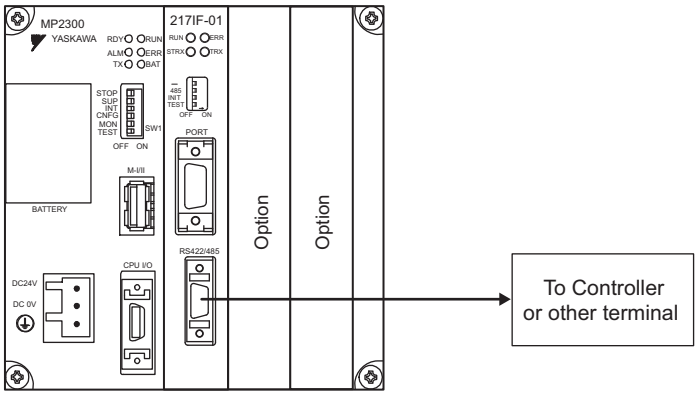
6.2.3 Module Connection Examples

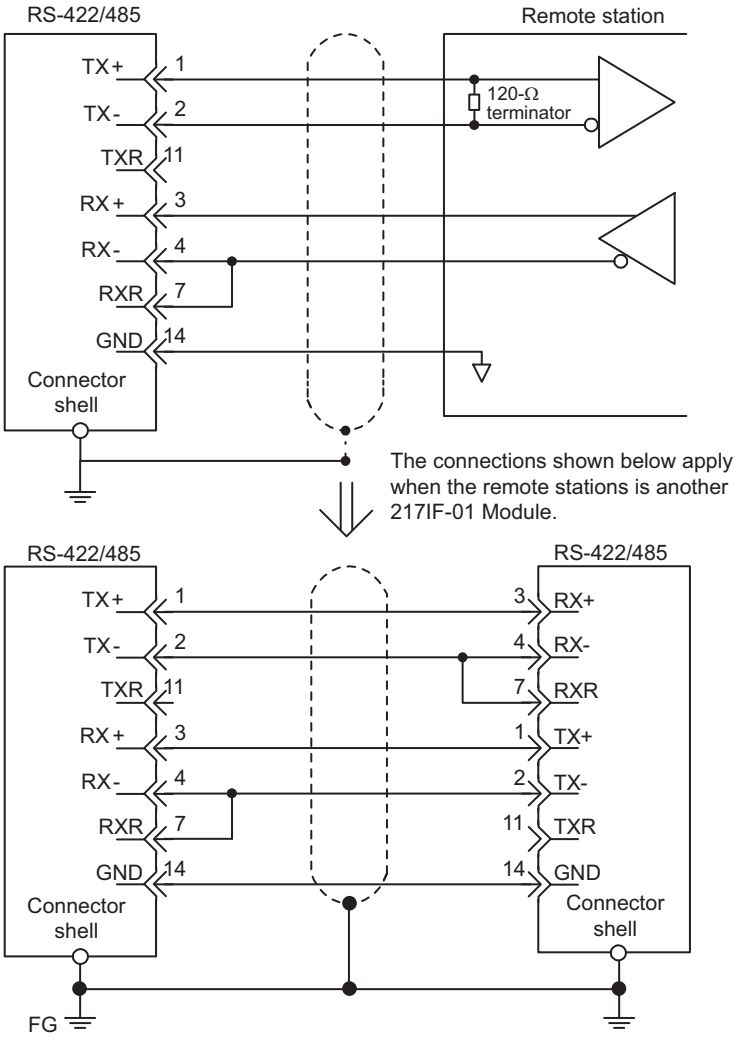
(1) PORT Connector Connections

Refer to *3.5 Serial Communication Connection Example* on page 3-22 for PORT connector connections.

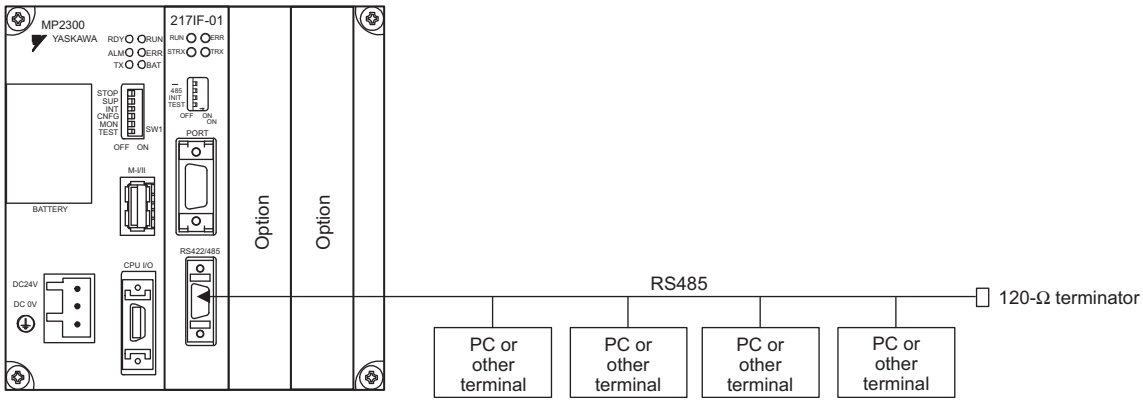
(2) RS-422/485 Connections

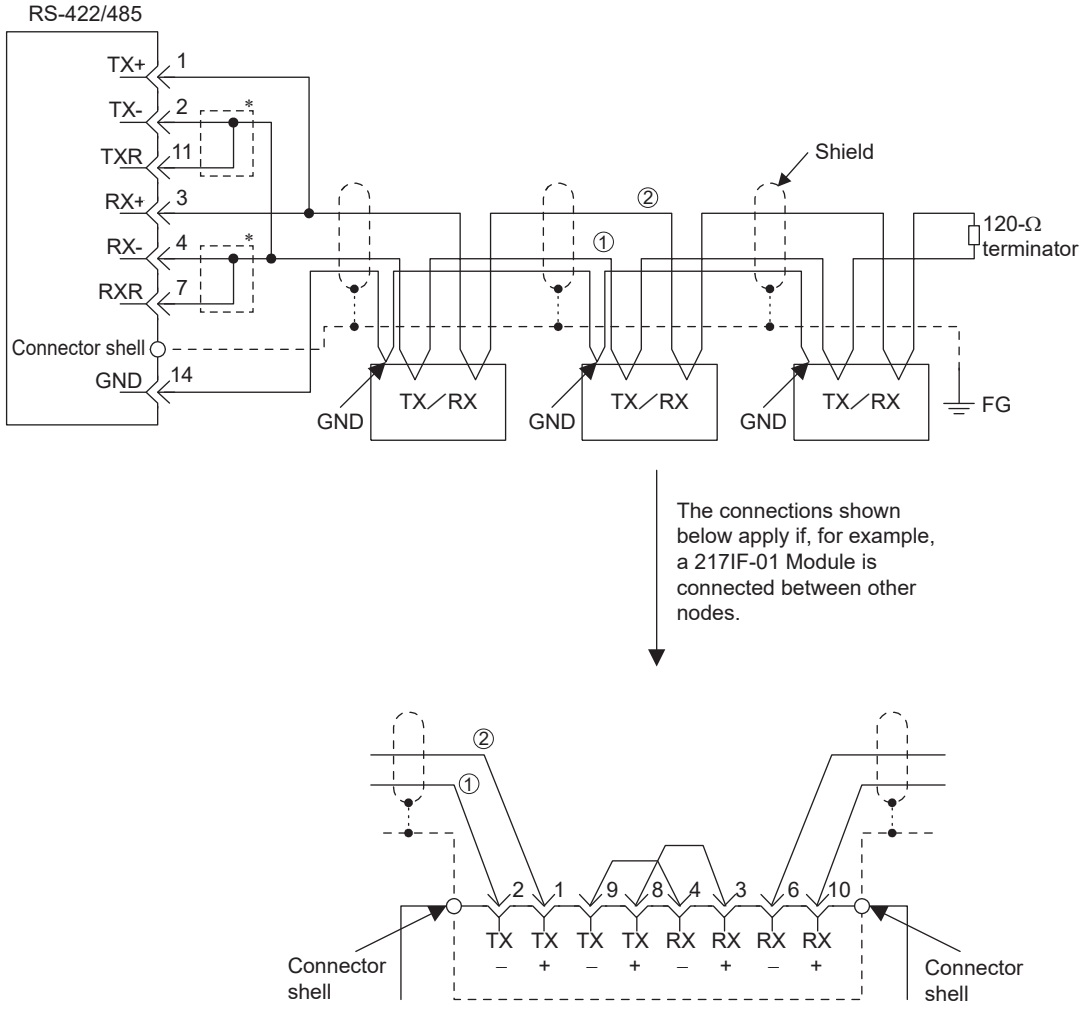
■ RS-422 Wiring





■ RS-485 Wiring





The connections shown below apply if, for example, a 217IF-01 Module is connected between other nodes.

* When wiring with a 2-wire system, connect only TXR or RXR.

- The terminator is enabled by connecting terminals 2-11 and terminals 4-7 for RS-422/485 ports.

6.3 Serial Communication

Refer to *Chapter 3 Serial Communication* for information on the transmission definitions, connection examples, programming examples, the Message Send function, the Message Receive function, and other details on serial communication (RS-232C communication or RS-422/485 communication).

7

260IF-01 Module

This chapter describes the 260IF-01 Module in detail and includes definition methods and programming examples.

- Before reading this chapter, be sure to read *Chapter 1 Starting Communication Modules*.

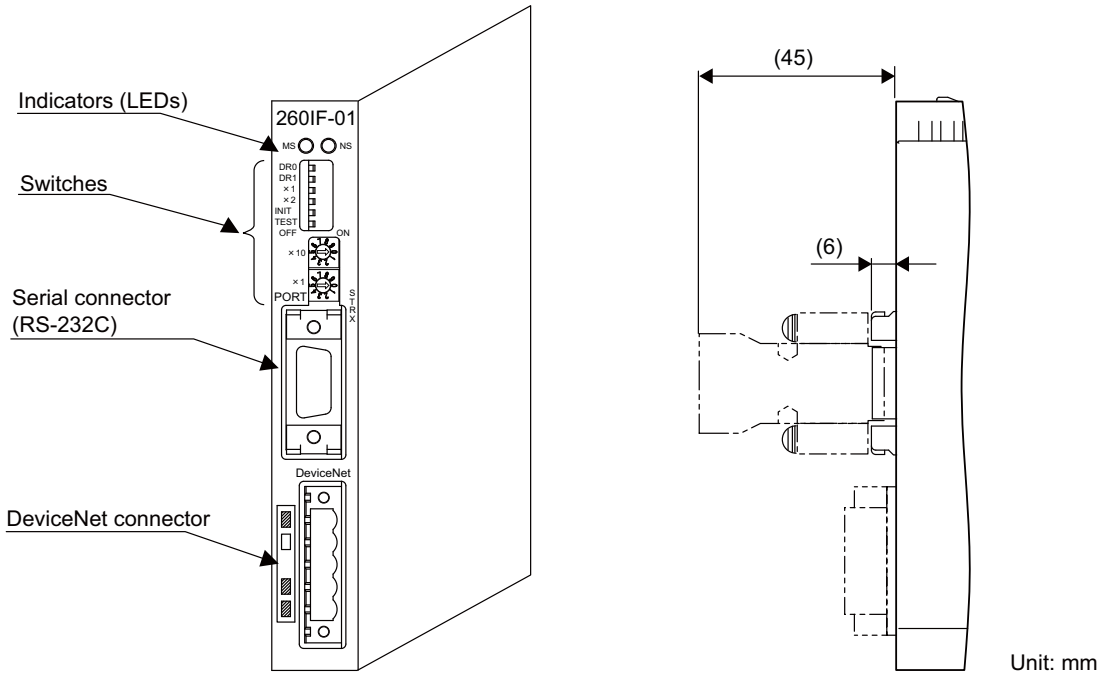
7.1 Overview ---- 7-2
7.1.1 Appearance and Connectors ---- 7-2
7.1.2 Network Configuration ---- 7-3
7.1.3 System Configuration Examples ---- 7-8
7.1.4 Module Specifications ---- 7-9
7.1.5 Indicators ---- 7-11
7.1.6 Switch Settings ---- 7-12
7.1.7 Default Parameters at Initial Startup ---- 7-12
7.1.8 Operation for Self-configuration ---- 7-13
7.2 Connection Specifications ---- 7-14
7.2.1 Connector Specifications ---- 7-14
7.2.2 Cable Specifications ---- 7-15
7.2.3 DeviceNet System Configuration Example ---- 7-15
7.3 RS-232C Communication ---- 7-16
7.4 260IF Transmission Definitions ---- 7-17
7.4.1 260IF Transmission Configuration Window ---- 7-17
7.4.2 Transmission Parameters Tab Page ---- 7-19
7.4.3 I/O Status Tab Page ---- 7-21
7.4.4 Status Tab Page ---- 7-22
7.4.5 Saving and Deleting 260IF Transmission Definitions ---- 7-23
7.4.6 Parameter Setting Examples ---- 7-24
7.4.7 Communication Cycle Time ---- 7-26
7.4.8 Troubleshooting ---- 7-28
7.5 Connection Examples for DeviceNet Communication ---- 7-32
7.5.1 I/O Communication Example ---- 7-32
7.5.2 Setting Example for Explicit Message Communication ---- 7-35
7.6 Message Send Function ---- 7-40
7.6.1 Basic Specifications ---- 7-40
7.6.2 Inputs and Outputs for the Message Send Function ---- 7-41
7.6.3 Parameter List for DeviceNet Communication (260IF) ---- 7-45
7.6.4 Function Setting Example and Parameter Details for Explicit Messages ---- 7-45

7.1 Overview

The 260IF-01 Module has an RS-232C serial interface and a DeviceNet interface mounted in it. Personal computers, HMI devices, and controllers manufactured by other companies can be connected to the 260IF-01 Module via the PORT or DeviceNet connectors. Serial communication modes include message communication and engineering communication, and MEMOBUS, MELSEC, OMRON, and non-procedure protocols are supported.
By installing a 260IF-01 Module, the MP2000 Series Machine Controller can perform I/O communication and explicit message communication (master only) through a DeviceNet line. Explicit message communication is performed by specifying the MSG-SND function in the ladder programs or functions.

7.1.1 Appearance and Connectors

The following diagram shows the appearance of the 260IF-01 Module and gives the external dimensions of the connectors.



7.1.2 Network Configuration

This section outlines the network configuration.

(1) Basic Network Configuration

[a] Connection Methods

The following diagram shows an example network connection.

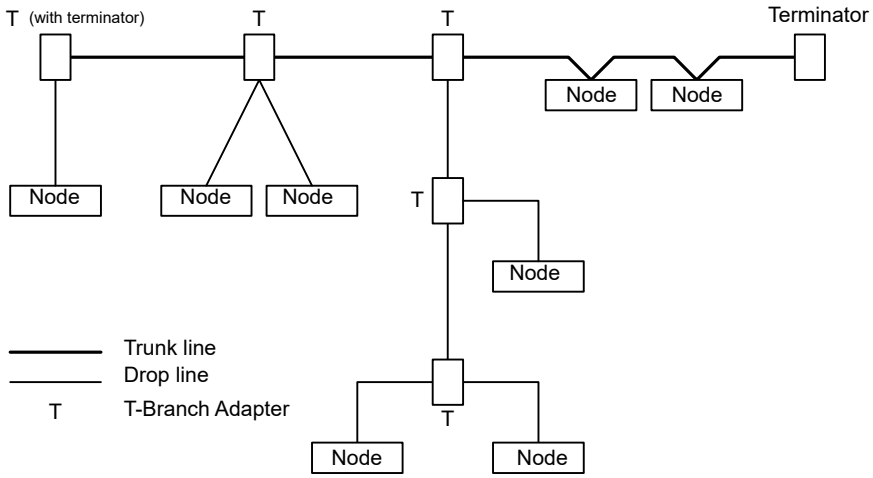


Fig. 7.1 Network Connection

[b] Configuration Elements

The network is configured from the following elements.

■ Nodes

A node is either a slave that connects to external I/O or the Master that controls the I/O of the slaves.
No restrictions apply to the locations of the Master or Slaves. Any node in the figure above can be the Master or a Slave.

■ Trunk Line and Drop Lines

A cable with a terminator on each end is the trunk line. Any cable branching from the trunk line is a drop line.

■ Connection Methods

A node is connected using the T-branch method or multi-drop method. A T-Branch Adapter is used to connect a node with the T-branch method. A node is directly connected to the trunk line or a drop line with the multi-drop method. Both T-branch and multi-drop methods can be used together in the same network, as shown in the diagram above.

■ Terminators

Both ends of the trunk line between pins 2 and 4 (for the CAN_H and CAN_L signals respectively) must have a terminator (121Ω) installed to decrease signal reflection and ensure stable network communications.

■ Communications Power Supply

The communications connectors of each node must be provided with communications power supply through the communications cable for DeviceNet communications. Communications power supply, internal circuit power supply, and I/O power supply must all be provided separately.

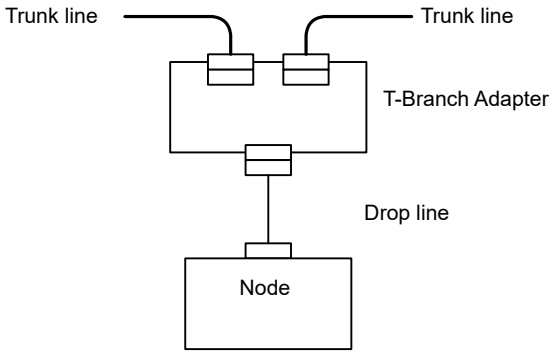
- Use only DeviceNet cables as communications cables.
- Always install terminators on both ends of the trunk line.
- Do not connect lightning arresters or any devices to the network other than the 260IF Module and DeviceNet products.

(2) Branching Methods

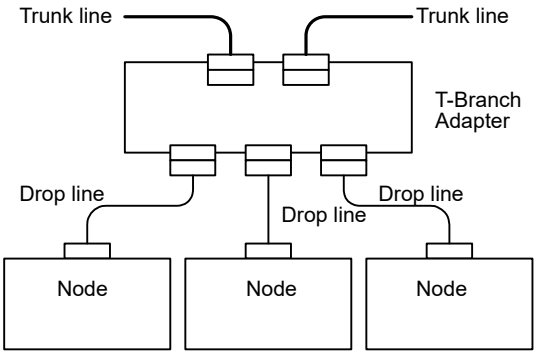
[a] Branching from the Trunk Line

Three methods can be used to branch from the trunk line.

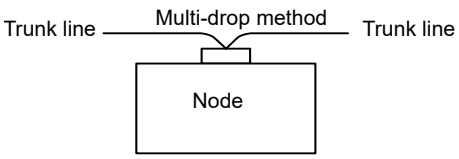
■ Branching to One Drop Line



■ Branching to Three Drop Lines



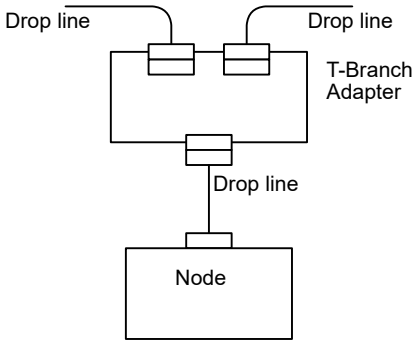
■ Direct Node Connection



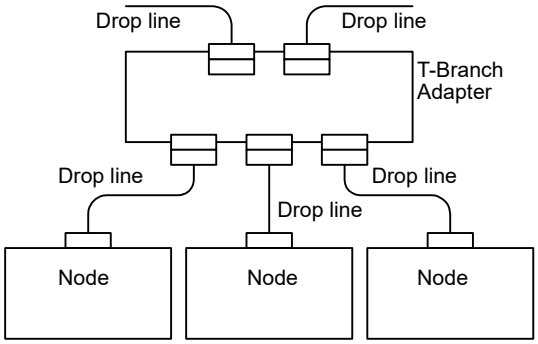
[b] Branching from Drop Lines

There are three methods that can be used to branch from drop lines.

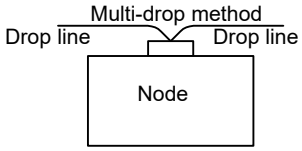
■ Branching to One Drop Line



■ Branching to Three Drop Lines



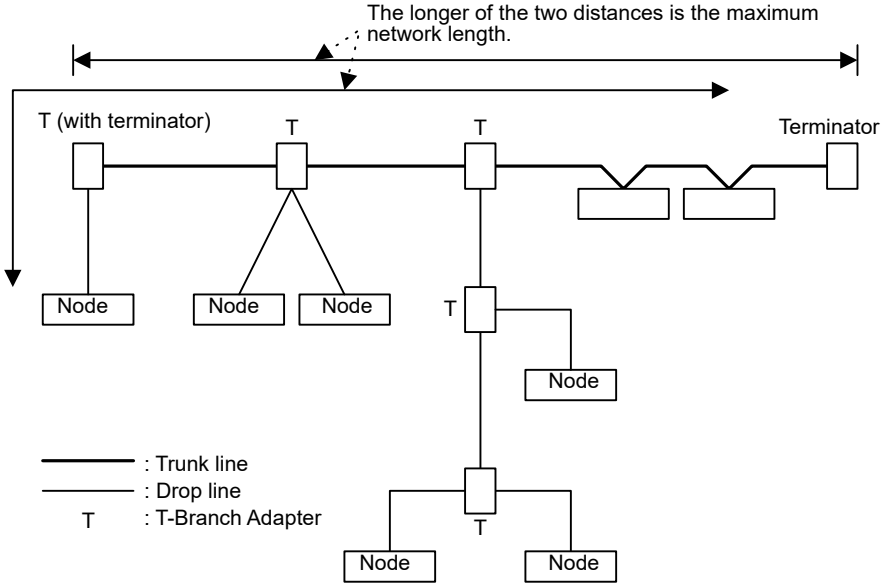
■ Direct Node Connection



(3) Network Configuration Restrictions

[a] Maximum Network Length

The maximum network length is either the line length between the two nodes located farthest from each other or the line length between the terminators on the ends of the trunk line, whichever is longer.



Both thick and thin cables can be used.
The thick cables are stiff and difficult to bend but they have little signal degradation and can be used for comparatively long transmission distances.
Thin cables are supple and easy to bend but signal degradation is greater and they are not, therefore, suited to communications over long distances.
The maximum network length is determined by the type of cable, as shown in the following table.

Baud Rate (kbps)	Maximum Network Length (m)	
	Thick Cable	Thin Cable
500	100	100
250	250	100
125	500	100

Using Thick Cable and Thin Cable Together

The line connecting two nodes located farthest from each other can use both thick and thin cables provided that the length of each cable satisfies the conditions in the following table.

Baud Rate (kbps)	Maximum Network Length (m)
500	$L_{Thick} + L_{Thin} \leq 100$
250	$L_{Thick} + 2.5 \times L_{Thin} \leq 250$
125	$L_{Thick} + 5 \times L_{Thin} \leq 500$

Note: L_{Thick}: Thick cable length L_{Thin}: Thin cable length

[b] Drop Line Length

The drop line length is the line length between the branch point on the trunk line to the farthest node that is located on the drop line.
The maximum drop line length is 6 m. A drop line can be branched out into other drop lines.

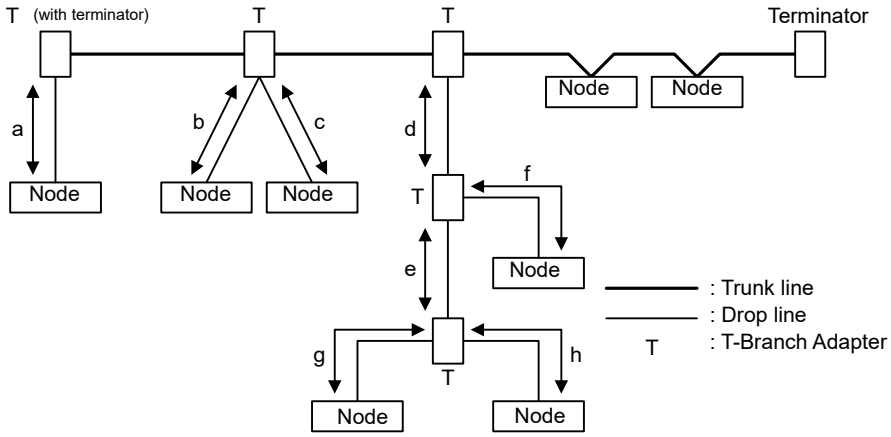
[c] Total Drop Line Length

The total drop line length is the total of all drop line lengths. The total drop line length must be within the allowable range and even then, each drop line must be 6 m or less.
The allowable range of total drop line length varies with the baud rate as shown in the following table.

Baud Rate (kbps)	Total Branch Length (m)
500	39 max.
250	78 max.
125	156 max.

■ Configuration Example

The following example is for a baud rate of 500 kbps.



The above example must satisfy the following conditions.

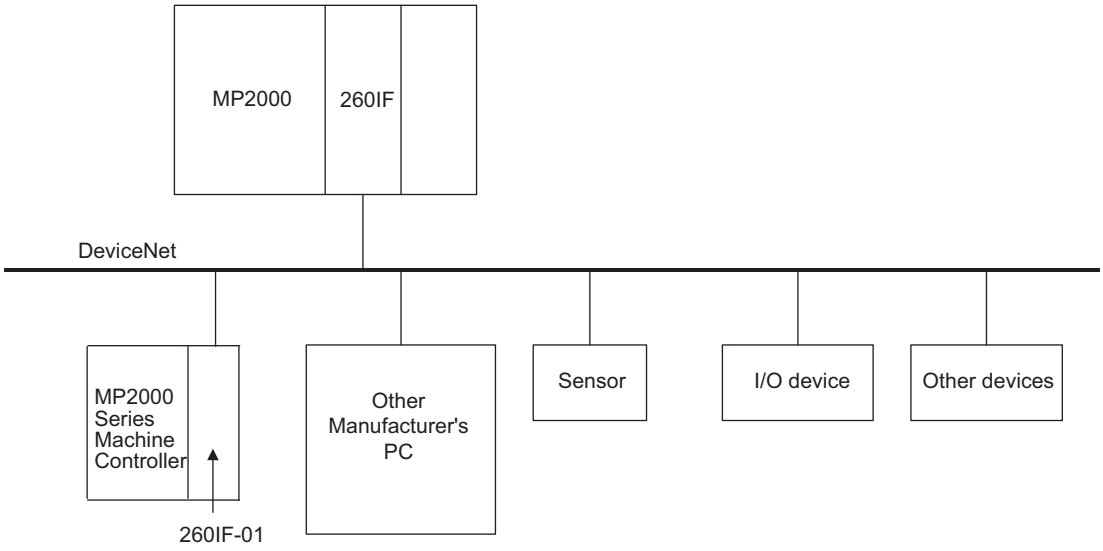
Drop lengths:
$a \leq 6$ m $b \leq 6$ m $c \leq 6$ m $d \leq 6$ m $d + f \leq 6$ m
$d + e + g \leq 6$ m $d + e + h \leq 6$ m
Total drop length = $a + b + c + d + e + f + g + h \leq 39$ m

7.1.3 System Configuration Examples

This section gives an overview of the configuration of a system that uses a 260IF-01 Module.

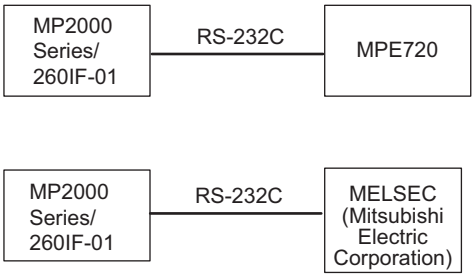
(1) Example of DeviceNet Connection

The following diagram shows an example of a system connected to the DeviceNet.



(2) Example of Serial Connections

The following diagram shows an example of serial connection. The 260IF-01 Module is connected to a personal computer or a controller manufactured by another company on a one-to-one basis.



7.1.4 Module Specifications

This section provides the specifications of the 260IF-01 Module.

(1) Hardware Specifications

Item	Specification
Name	260IF-01
Model Number	JAPMC-CM2320
Communication Ports	RS-232C: 1 port (indicated name: PORT)
	DeviceNet: 1 port (indicated name: DeviceNet)
Indicators	Module status LED indicators MS (green, red) NS (green, red)
Setting Switches	DIP switch: DR0, DR1, X1, X2, INIT, TEST Rotary switches: x10, x1
Dimensions (mm)	125 x 95 mm (H x D)
Mass	85 g

(2) Transmission Specifications

■ RS-232C Transmission Specifications

Item	Specification
Connector	9-pin D-sub (female)
Transmission Distance	15 m max.
Baud Rate	9600 or 19200 bps
Access Mode	Asynchronous (start-stop synchronization)
Communication Type	Half duplex
Communication Modes	Message communication and engineering communication
Communication Protocols	MEMOBUS, MELSEC, OMRON, and non-procedure
Media Access Control Method	1:1
Transmission Format (Can be set.)	Data bit length: 7 or 8 bits Stop bits: 1 or 2 bits Parity: Odd, even, or none

■ DeviceNet Transmission Specifications

Item		Specification
Number of Circuits		1
Supported Communication Methods		• I/O communication functions (polled and bit-strobed) • Explicit messages (supports master only)
I/O Communication	Max. Number of Slaves	63
	Max. Number of I/O Bytes	2048 bytes, 256 bytes/node for max. number of I/O bytes.
Message Communication (Only for Master)	Max. Number of Nodes for Message	63 Max. number of nodes for simultaneous communication: 4
	Max. Message Length	256 bytes
	Function for Execution	MSG-SND Function
Settings		2 rotary switches on front panel: Node address
		DIP switch on front panel: Band rate, Master/Slave mode
Indicators		2 LEDs: MS, NS
Power Supply Voltage for Communication		24 VDC ±10% (supplied by special cable)
Current Consumption		Communication power supply: 45 mA max. (supplied from communication connector). Internal circuit power supply (supplied from Basic Module).

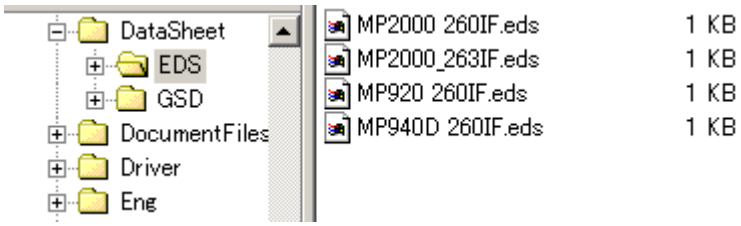
■ Device Profile

The device profile standardizes and models device structure and operation for each device.
The following is the device profile of the 260IF-01 Module.

Item	Code	Description
Vendor code	0x2C	"Yaskawa Electric America, Inc."
Product type	0x0C	"Communications Adapter"
Product code	0x0223	"MP2300/260IF-01"

<EDS File>

EDS files are device definition files that are used by DeviceNet devices. A device profile is defined by installing the EDS file.
The EDS file is stored on the MPE720 installation disk.
Under the root directory, open the **DataSheet** Folder and then the **EDS** Folder. Use the MP2000 260IF.eds file stored in the **EDS** Folder.



The file can also be downloaded from the ODVA website (http://www.odva.org).

- The EDS file is not required if assignments are defined using the MPE720. The EDS file is required if you use another company's configuration software.

(3) Operating Environment Specifications

Item		Specification
Environmental Conditions	Ambient Operating Temperature	0 to 55°C
	Ambient Storage Temperature	–25 to 85°C
	Ambient Operating Humidity	30% to 95% (with no condensation)
	Ambient Storage Humidity	5% to 95% (with no condensation)
	Pollution Level	Pollution level 2 (conforming to JIS B 3502)
	Corrosive Gas	There must be no combustible or corrosive gas.
	Operating Altitude	2,000 m above sea level or lower
Mechanical Operating Conditions	Vibration Resistance	Conforming to JIS B 3502 Vibration and acceleration: 10 to 57 Hz with single-amplitude of 0.075 mm 57 to 150 Hz with fixed acceleration of 9.8 m/s^2 10 sweeps each in X, Y, and Z directions (sweep time: 1 octave/min)
	Shock Resistance	Conforming to JIS B 3502 Peak acceleration of 147 m/s^2 (15 G) twice for 11 ms each in the X, Y, and Z directions
Electrical Operating Conditions	Noise Resistance	Conforming to EN 61000-6-2, EN 61000-6-4, EN 55011 (Group 1 Class A)
Installation Requirements	Ground	Ground to 100 Ω max.
	Cooling Method	Natural cooling

7.1.5 Indicators

The following table shows the status of the 260IF-01 Module LED indicators.

Indicator Name	Indicator	Status
MS (2-color LED)	Lit green	Operating normally
	Lit red	Module error
	Not lit	Module power supply disconnected
	Lit alternately red and green	Communication not possible
NS (2-color LED)	Lit green	Operating normally
	Green blinking	No I/O assignments or connection being established
	Lit red	Error (Bus OFF, duplicated MAC ID)
	Red blinking	Communication error
	Not lit	Communication power supply disconnected, checking for duplicated MAC ID
	Lit alternately red and green	Communication not possible
STRX (mounted on PCB)	Lit/blinking green	Sending or receiving RS-232C data
	Not lit	No RS-232C data communication

■ Status Indication by LED Indicators

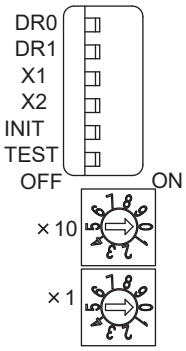
The following table shows the LED indications and their meanings for operating status and errors.

Status	Indicators			Meaning	Description
	MS	NS	STRX		
Power ON	○	○	○	Initializing	This is the status immediately after the power to the 260IF-01 is turned ON (initializing). The initialized status will be entered as soon as initialization has been completed.
Error	Red ●	○	–	Hardware error	The hardware in the Module has failed. The Module must be replaced.
	Red ●	Red ●	–		
Others			–	Normal status	Refer to the Status columns in *7.4.8 Troubleshooting* on page 7-28.

- ● : Lit
  ○ : Not lit
  – : Not relevant

7.1.6 Switch Settings

The following table shows the 260IF-01 Module switch settings.



Label	Name	Status	Function			Factory Setting
			DR1	DR0	Setting	
DR0	Baud Rate Setting 0	ON	OFF	OFF	125 kbps	OFF
		OFF	OFF	ON	250 kbps	
DR1	Baud Rate Setting 1	ON	ON	OFF	500 kbps	
		OFF	ON	ON	Communication not possible.	
X1	Master/Slave Mode	ON	Used in Master mode.			OFF
		OFF	Used in Slave mode.			
X2	Self-diagnosis (DeviceNet)	ON	System use.			OFF
		OFF	Normal operation (Always leave set to OFF.)			
INIT	Initial Startup	ON	For engineering communication. Starts the RS-232C (PORT) port using the default parameters except settings of automatic reception function. The connector is not valid. Given higher priority than the MP2000 Series Machine Controller flash startup and self-configuration startup.			OFF
		OFF	Set to OFF for MP2000 Series Machine Controller flash startup and self-configuration startup.			
TEST	TEST	ON	System use.			OFF
		OFF	Normal operation (Always leave set to OFF.)			
×10	Node Address 10s Digit Setting	–	Sets the node address. (Rotary decimal switch)			0
×1	Node Address 1s Digit Setting	–	Sets the node address. (Rotary decimal switch)			0

7.1.7 Default Parameters at Initial Startup

At initial startup (INIT switch set to ON), the following parameter settings will be made for RS-232C interface.

■ RS-232C Interface (The setting of INIT switch to ON is valid while the power is being supplied.)

Item	Setting		
Communication Protocol	MEMOBUS		
Master/Slave	Slave		
Device Address	1		
Serial I/F	RS-232C		
Communication Mode	RTU		
Data Length	8 bits		
Parity	Even		
Stop Bits	1 bit		
Baud Rate	19.2 kbps		
Send Delay	Disable		
Receive Monitor Time	Disable		
Automatic Reception	Enable		
Interface Register Settings in the Slave		Leading Register	Number of Words
	Read Input Relays	IW0000	32768
	Read Input Registers	IW0000	32768
	Read/Write Coils	MW00000	65535
	Read/Write Holding Registers	MW00000	65535
	Coil/Holding Register Write Range	LO MW00000	
		HI MW65534	

7.1.8 Operation for Self-configuration

The operation of the 260IF for self-configuration is described in this section. (Refer to *1.5 Executing Self-configuration* on page 1-26.)

(1) Operation as the Master

If the 260IF-01 is set as the DeviceNet Master (x1 DIP switch pin turned ON) and self-configuration is executed, the 260IF-01 will search for Slaves detected to the DeviceNet cable and automatically assign I/O. The baud rate set on the DIP switch pins DR0 and DR1 and the local node address set on the rotary switches will be used.
Self-configuration will not be completed normally in the following cases.

- If the 260IF is connected to a DeviceNet cable and network power is being supply, but the Slaves are as follows:
  - Slaves do not exist.
  - The Slaves are not operating normally.
  - The Slaves are using a different baud rate.

If any of these occur, correct the Slaves so that they are in a normal state and execute self-configuration again.
If a DeviceNet cable is not connected to the 260IF-01, or the cable is connected but network power is not being supplied, self-configuration will be completed normally, but I/O will not be assigned.

(2) Operation as a Slave

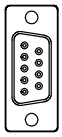
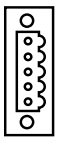
If the 260IF-01 is set as a DeviceNet Slave (x1 DIP switch pin turned OFF) and self-configuration is executed, the 260IF-01 will set the local node address according to the setting of the rotary switches. An I/O size of eight bytes will be assigned to the local node address. The baud rate set on DIP switch pins DR0 and DR1 will be used.

7.2 Connection Specifications

7.2.1 Connector Specifications

This section provides the connector specifications for the 260IF-01 Module.

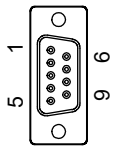
(1) Connector Specifications

Connector Shape	Name	Connector Name	No. of Pins	Connector Model		
				Module	Cable	Manufacturer
PORT 	RS-232C	PORT	9	17LE-13090-27(D2BC) 9-bin D-sub female connector	17JE-23090-02 (D8B) 9-pin D-sub male connector (Shell included.)	DDK Ltd.
DeviceNet 	DeviceNet	DeviceNet	5	MSTB2.5/5-GF-5.08AU	–	PHOENIX CONTACT

(2) Connector Pin Arrangement

■ PORT Connector

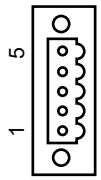
The PORT connector is used to connect the MP2000 Series Machine Controller to computers and HMI devices via an RS-232C connection.



Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	FG	Frame ground	6	–	–
2	SD	Send data	7	SG	Signal ground (0V)
3	RD	Receive data	8	–	–
4	RS	Ready to send	9	ER	Data terminal ready
5	CS	Clear to send	–	–	–

■ DeviceNet Connector

The DeviceNet connector is used to connect the MP2000 Series Machine Controller to computers and peripheral devices via a DeviceNet connection.



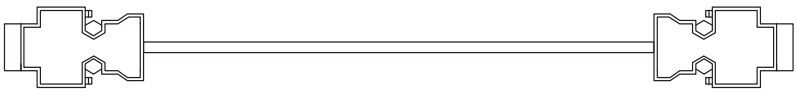
Pin Number	Signal Name	Description
5	V+	24-V external power supply for communication
4*	CAN-H	CAN bus line dominant H
3	SHIELD	–
2*	CAN-L	CAN bus line dominant L
1	V-	0-V external power supply for communication

* The terminator is connected to pin numbers 2 and 4.

7.2.2 Cable Specifications

This section provides the cable specifications for the 260IF-01 Module.

(1) RS-232C Cable (for MPE720 Engineering)

Model Number	Length	External Appearance (JEPMC-W5311-□□-E)
JEPMC-W5311-03-E	2.5 m	
JEPMC-W5311-15-E	15 m	

(2) DeviceNet Cable

A standard cable is not provided. Use a commercially available DeviceNet cable.

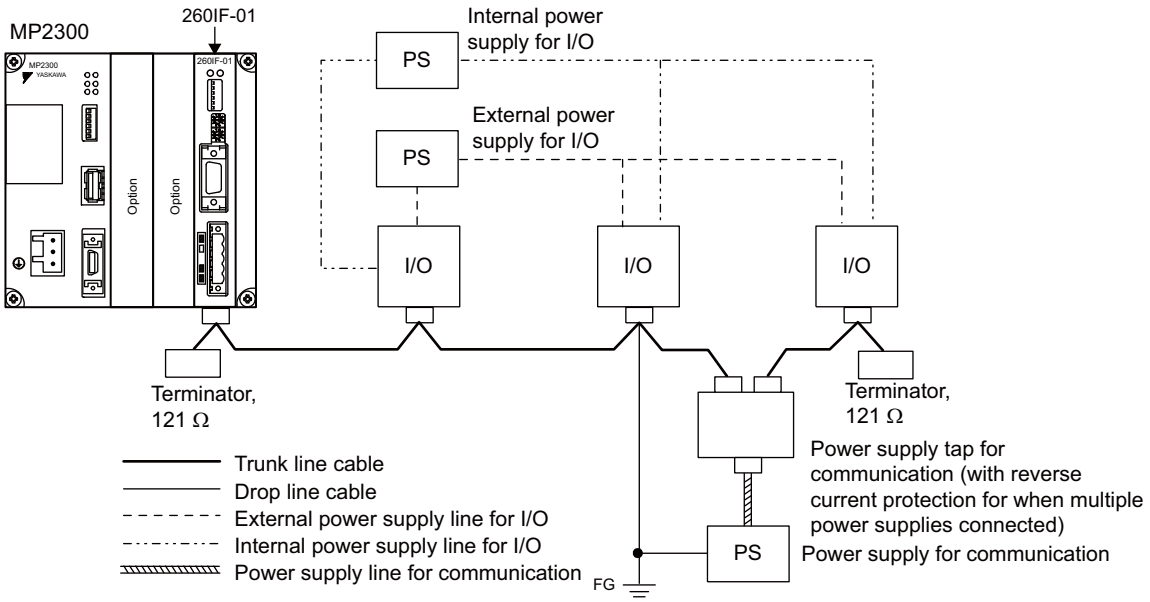
- Refer to the ODVA website (http://www.odva.org/) as required.

7.2.3 DeviceNet System Configuration Example

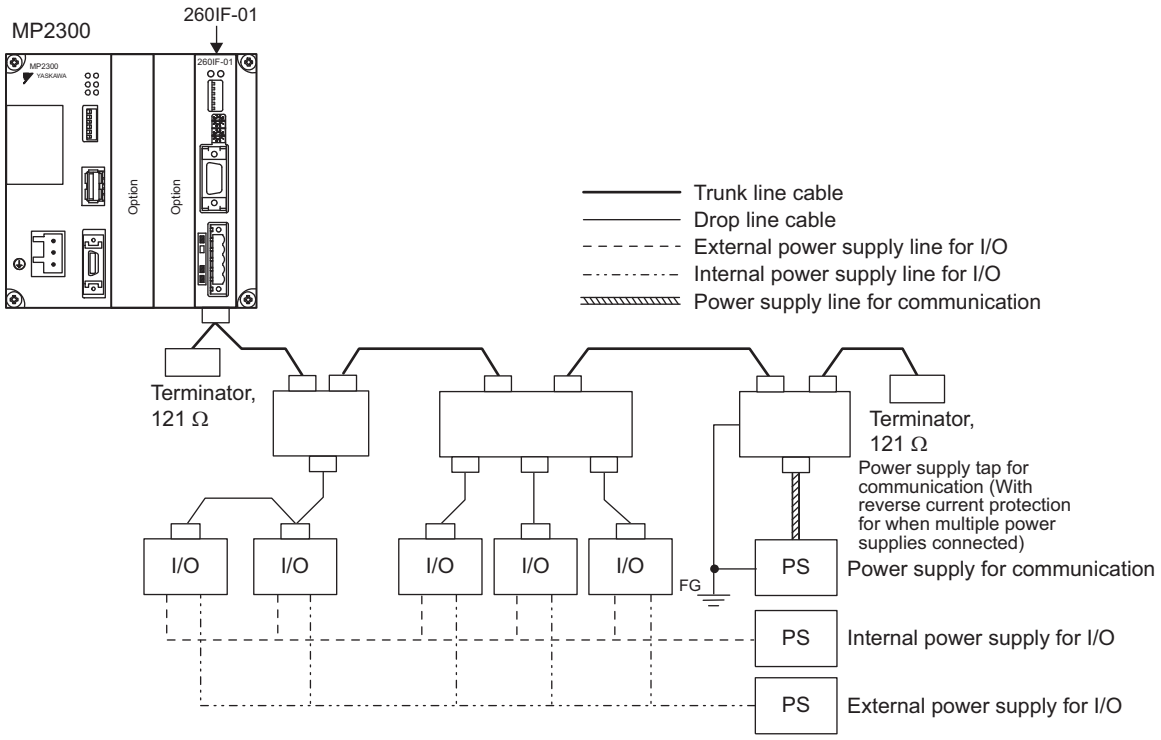
(1) Master Mode

Two system configurations are supported in Master Mode: multidrop and branching (T-branch, multi-branch, and drop-line connections).
The following sections provide examples of each configuration.

[a] Multi-drop Connections

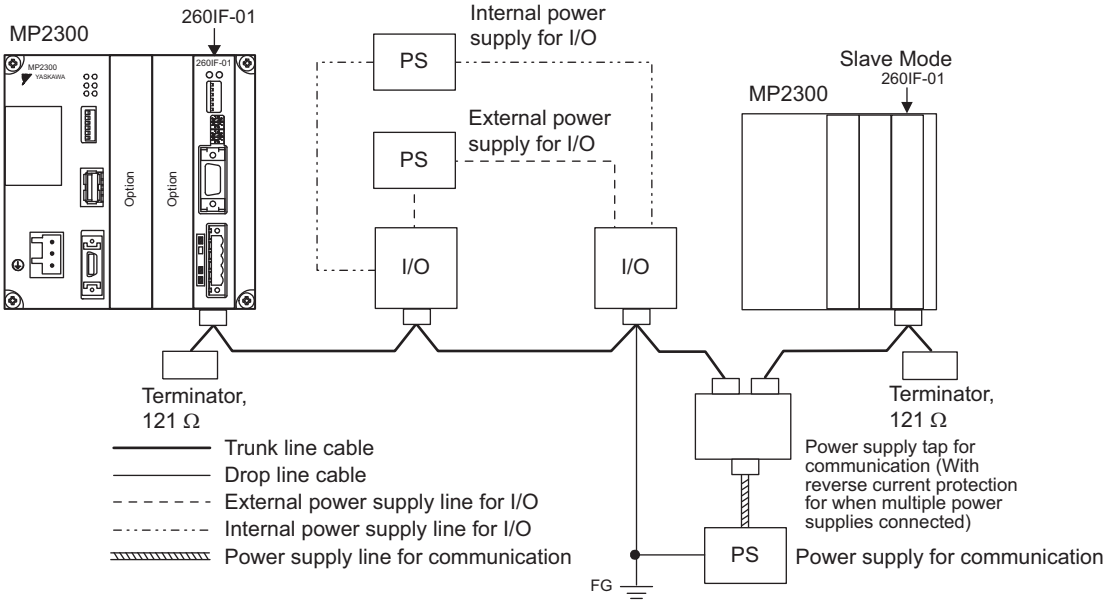


[b] T-branch, Multi-branch, and Drop-line Connections



(2) Slave Mode

The following diagram shows the system configuration for Slave mode.



7.3 RS-232C Communication

Refer to *Chapter 3 Serial Communication* for information on the transmission definitions, connection examples, programming examples, the Message Send function, the Message Receive function, and other details on communication through the RS-232C port on the 260IF-01 Module.

7.4 260IF Transmission Definitions

To perform DeviceNet communication with the MP2000 Series Machine Controller, the 260IF transmission (DeviceNet communication) definition file must be created.
This section describes how to define 260IF transmissions using MPE720 windows.

7.4.1 260IF Transmission Configuration Window

To define the 260IF transmissions, first the **Module Configuration** Window is opened on the MPE720, and then the **260IF Transmission Configuration** Window is opened.

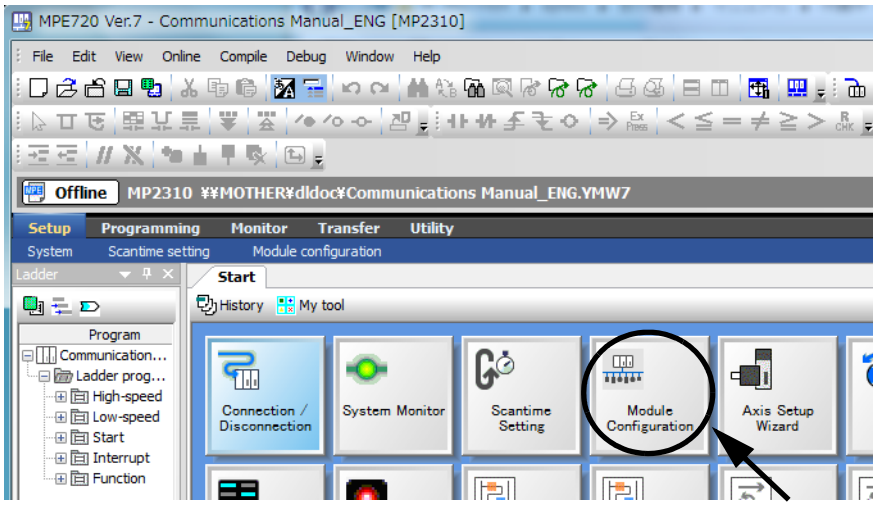
(1) Displaying the Module Configuration Window

Use the following procedure to display the **Module Configuration** Window.

1. Start the MPE720 on the personal computer connected to the Machine Controller and open the project file.

- For information on starting the MPE720, refer to *MP2000/MP3000 Series Engineering Tool MPE720 Version 7 User's Manual (Manual No.: SIEP C880761 03)*.

2. Click the **Module Configuration** Button on My Tool View.



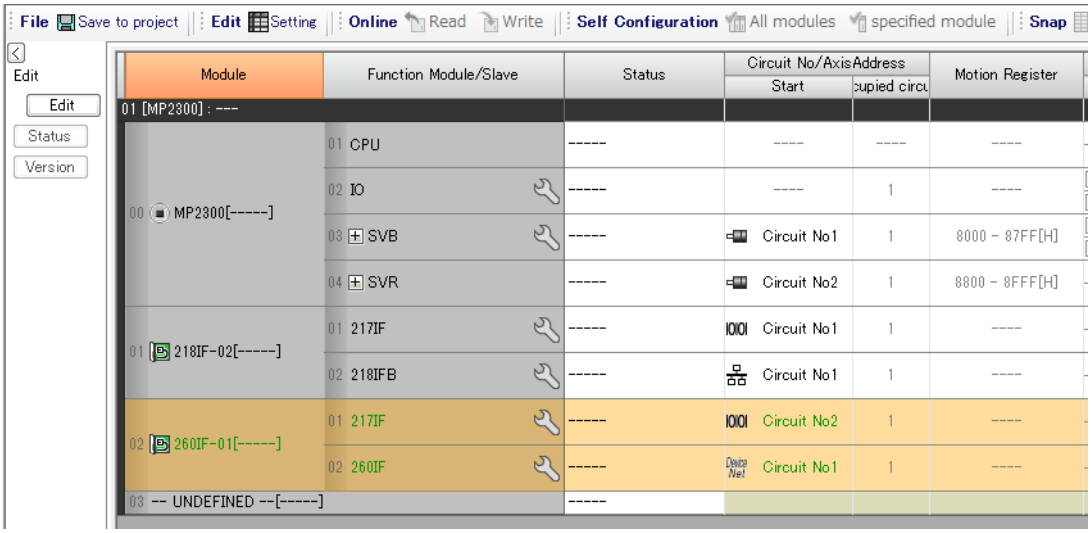
The **Module Configuration** Window will be displayed.

(2) Displaying the 260IF Transmission Configuration Window from the Module Configuration Window

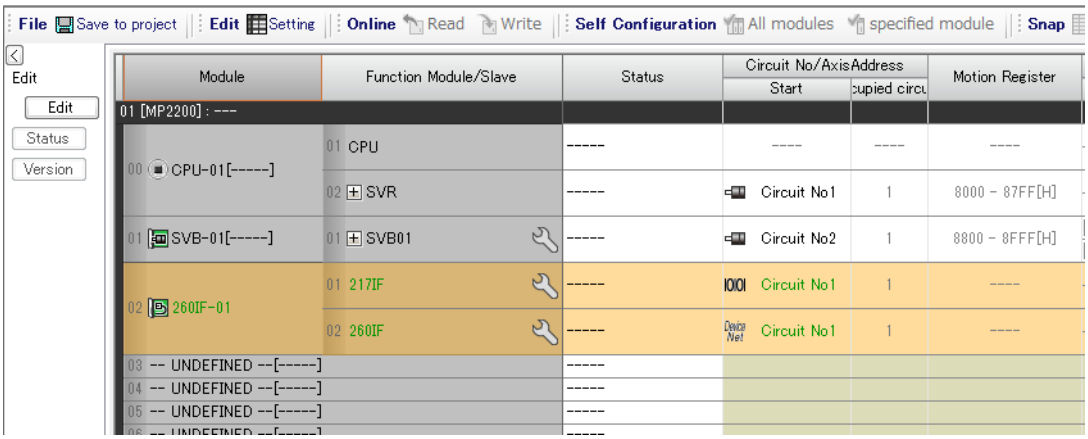
As the result of executing self-configuration, all Option Modules connected to the Machine Controller will be displayed in the **Controller** Area of the **Module Configuration** Window. (Refer to *1.5 Executing Self-configuration* on page 1-26.)

In the **Controller** Area, select the cell for 260IF-01, and then double-click the slot number cell for the 260IF in the **Module Details** Area. The **260IF Transmission Configuration** Window (see next page) will be displayed.

<MP2300/MP2300S/MP2310 Module Configuration Window>



<MP2100M/MP2200 Module Configuration Window>



In Online Mode, the 260IF transmission definitions data saved in the Machine Controller are displayed. In Offline Mode, the definitions data saved in the personal computer on which MPE720 is running is displayed.

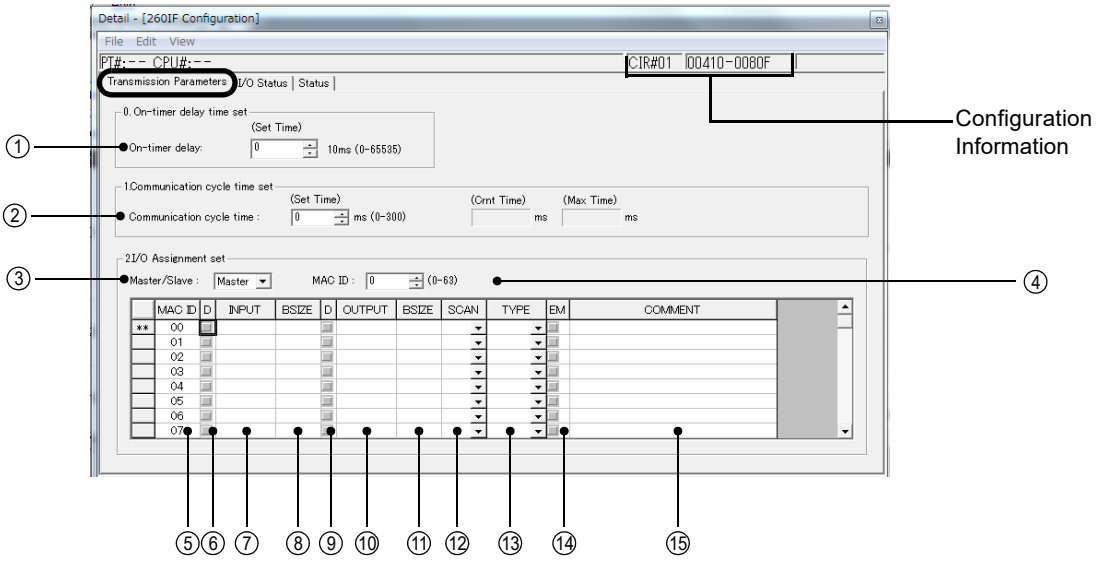
The **260IF Transmission Configuration** Window is composed of three tab pages: **Transmission Parameters**, **I/O Status**, and **Status**. These tab pages are used to set the definitions and monitor the settings.

- If the **Transmission Configuration** Window is being opened for the first time, a "new file" message box will be displayed and the **260IF Transmission Configuration** Window will be displayed when the **OK** Button is clicked.
- If the cell for the 217IF in the **Module Details** Area is double-clicked, the **217IF Transmission (serial communication) Configuration** Window will be displayed. Refer to *Chapter 3 Serial Communication* for details on serial communication.

7.4.2 Transmission Parameters Tab Page

Set the parameters required for using the 260IF transmission system. Setting these parameters enables data communication from using the MSG-SND and MSG-RCV functions

- Refer to *7.4.6 Parameter Setting Examples* on page 7-24 for a concrete example of parameter settings.

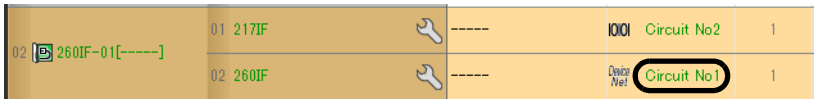


■ Configuration Information

The configuration information is the same as the information displayed in the **Module Details** Area in the **Module Configuration** Window.

RACK#: The rack number of the rack in which the 260IF-01 is defined.
Slot#: The slot number of the slot in which the 260IF-01 is defined.
CIR#: The circuit number of the 260IF-01.
Register number: The leading and ending I/O register number of the 260IF-01.

- The configuration information can also be changed in the **Module Details** Area of the **Module Configuration** Window. Do not use a circuit number that is already being used by another 260IF.



- When the MSG-SND function is used, set the circuit number displayed here for Circuit Number.

■ On-timer Delay Time Set (ON-time Delay Set Time)

① On-timer Delay

The 260IF-01 will be initialized when the time set here elapses after the power turns ON. This function is used to delay the startup of the 260IF-01.

■ Communication Cycle Time Set

② Communication Cycle Time

Set the communication cycle time when a 260IF-01 Module is used as the Master. Setting is not necessary if the Module is used as a Slave.

- Be sure to set the communication cycle time if the 260IF-01 Module is used as the Master. When the 260IF-01 Module is used as the Master with the **Communication cycle time** set to the default value (0), the time to update the data of the Slaves will be delayed.
- Refer to *7.4.7 Communication Cycle Time* on page 7-26 for information on how to estimate the communication cycle time.

■ I/O Assignment Set (I/O Assignments Settings)

- When the ③ **Master/Slave** parameter is set to *Master*, only the Slave stations can be set. The Slave stations will have "**" displayed at the left.
- When the ③ **Master/Slave** parameter is set to *Slave*, only the local station can be set. The local station will have "**" displayed at the left.

③ Master/Slave

Set the operating mode of the 260IF-01 Module to either Slave or Master. The setting here must be the same as the setting on the setting switch (SW4) on the 260IF-01 Module.

④ MAC ID

Set the MAC ID of the 260IF-01 Module. Set the same value as the value set on the rotary switches on the 260IF-01 Module.

⑤ MAC ID Column

This column displays the DeviceNet MAC ID of the devices to which I/O is being assigned. The ID numbers are automatically set in ascending order starting from 00. The symbol "**" is displayed in the left column for the 260IF-01 Module.

③ D

This setting enables or disables receiving input data to the 260IF-01 Module from a connected device. Click the checkbox to select it and disable I/O data communication. Click the checkbox again to clear the selection.

⑦ INPUT (Leading Input Register)

Set the leading address of the input area (input register IWxxxx) to be assigned to the corresponding device. Set a hexadecimal word address.
If nothing is set here, no input data will be sent and received.

⑧ BSIZE (Input Area Size)

Set the size of the input area (input register IWxxxx) assigned to the corresponding device in bytes. Set the byte size in decimal between 1 and 256 bytes per Slave.
The byte arrangement is little-endian.

⑨ D

This setting enables or disables receiving output data to the 260IF-01 Module from a connected device. Click the checkbox to select it and disable I/O data communication. Click the checkbox again to clear the selection.

⑩ OUTPUT (Leading Output Register)

Set the leading address of the output area (output register OWxxxx) to be assigned to the corresponding device. Set a hexadecimal word address.
If nothing is set here, no output data will be sent and received.

⑪ BSIZE (Output Area Size)

Set the size of output area (output register OWxxxx) to be assigned to the corresponding device in bytes. Set the size in decimal between 1 and 256 words per Slave.
The byte arrangement is little-endian.

⑫ SCAN

The data refresh cycle (SCAN) is the timing at which I/O data is refreshed between the connected device and the 260IF-01 Module.
The data refresh cycle is not synchronous with I/O data communication.
If HIGH is set, I/O data is refreshed during the high-speed scan of the MP2000 Series Machine Controller.
If LOW is set, I/O data is refreshed during the low-speed scan of the MP2000 Series Machine Controller.

⑬ TYPE

Set **Polled** or **Strobed** for the I/O communication type.

Polled: This setting is possible for any DeviceNet Slave device.

Strobed: This setting is possible for an input DeviceNet Slave that handles the data of 8 bytes or less.

- For details on polled and strobed transmissions, refer to the DeviceNet specifications.

⑭ EM (Explicit Messages)

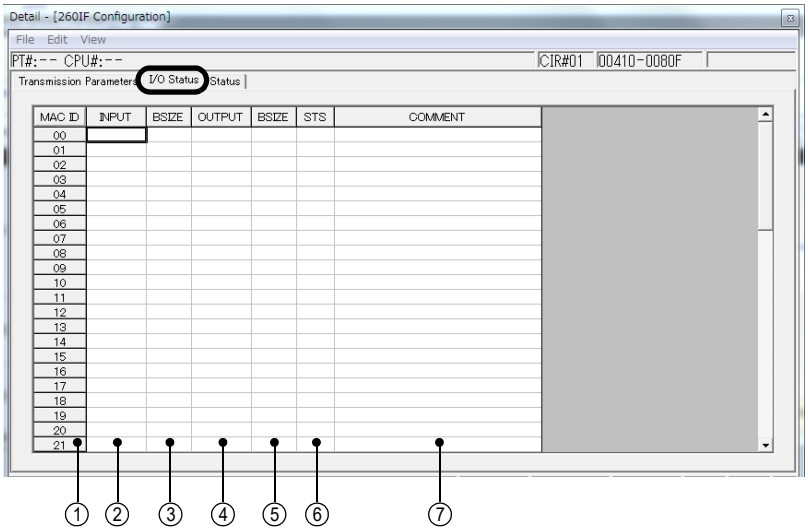
Set whether explicit message communication (explicit transmissions) is performed between the Master and Slaves. Click the checkbox to select it and enable explicit transmissions. Click the checkbox again to clear the selection. This parameter can be set only when the Communication Module is set as the Master.

⑮ COMMENT

A character string, such as a device name or type, can be entered as a comment. Up to 32 characters can be entered.

7.4.3 I/O Status Tab Page

The I/O Status Tab Page displays the 260IF transmission status. The settings cannot be changed.



① MAC ID

This is the MAC ID (DeviceNet address) of the DeviceNet devices connected to the DeviceNet network.

② INPUT (Leading Input Register)

Displays the leading address of the input area (input register IWxxxx) assigned to the corresponding device.

③ BSIZE (Input Area Size)

Displays the size of the input area (input register IWxxxx) assigned to the corresponding device in bytes.

④ OUTPUT (Leading Output Register)

Displays the leading address of the output area (output register OWxxxx) assigned to the corresponding device.

⑤ BSIZE (Output Area Size)

Displays the size of the output area (output register OWxxxx) assigned to the corresponding device in bytes.

⑥ STS

Displays the I/O status as a code. The displayed I/O status is as listed below.

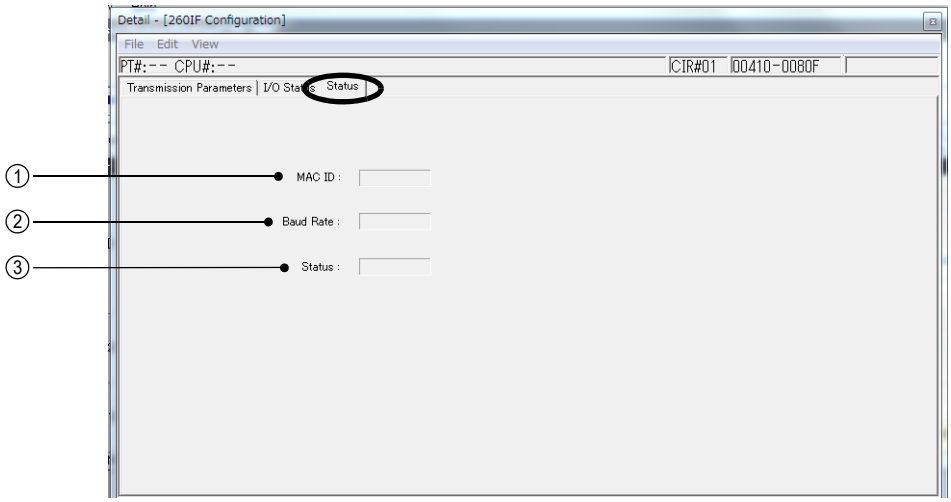
0000H: Communication is not being performed.
8000H: I/O communication is normal.
4048H: I/O communication error; communication is stopped.
4049H: I/O communication error; ID information on the Slave device differs from the set value.
404DH: I/O communication error; I/O size of the Slave device differs from the set value.
404EH: I/O communication error; no response was received from the Slave device.
4053H: I/O communication error; initialization is rejected by the Slave device.
4054H: I/O communication error; the Slave device has not been initialized yet.
4056H: I/O communication error; the Slave device is idle.

⑦ COMMENT

Displays the comment set for each device in the **I/O Assignment Set** Area on the **Transmission Parameters** Tab Page.

7.4.4 Status Tab Page

The **Status** Tab Page displays the MAC ID, baud rate, and communication status of the 260IF-01 Module. The settings cannot be changed.



① MAC ID

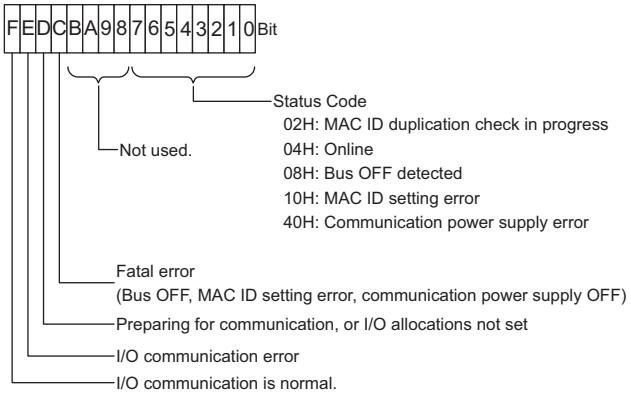
This is the MAC ID (DeviceNet address) that is set for the 260IF-01 Module.

② Baud Rate

Displays the baud rate set for the 260IF-01 Module.

③ Status

Displays the status of the 260IF-01 Module as a code. The displayed status is as listed below.



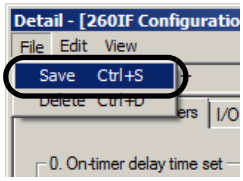
7.4.5 Saving and Deleting 260IF Transmission Definitions

- When changing, saving, or deleting 260IF transmission definitions data, save the data to flash memory in the Machine Controller, and then turn the power to the Machine Controller OFF and then turn it ON again.

(1) Saving the 260IF Transmission Definitions Data

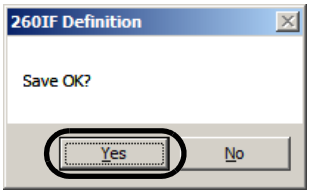
Use the following procedure to save the 260IF transmission definitions data if it has been set or changed. In Online Mode, the definitions data is saved in the Machine Controller and in personal computer running the MPE720. In Offline Mode, the definitions data is saved in the personal computer running the MPE720.

1. Select ***File - Save*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

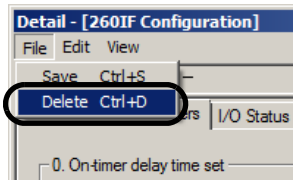
2. Click the **Yes** Button.



(2) Deleting the 260IF Transmission Definitions Data

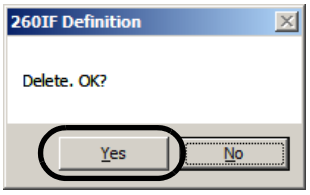
Use the following procedure to delete all of the 260IF transmission definitions data. In Online Mode, the definitions data is deleted from the MP2000 Series Machine Controller and the personal computer running the MPE720. In Offline Mode, the definitions data is deleted from the personal computer running the MPE720.

1. Select ***File - Delete*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

2. Click the **Yes** Button.

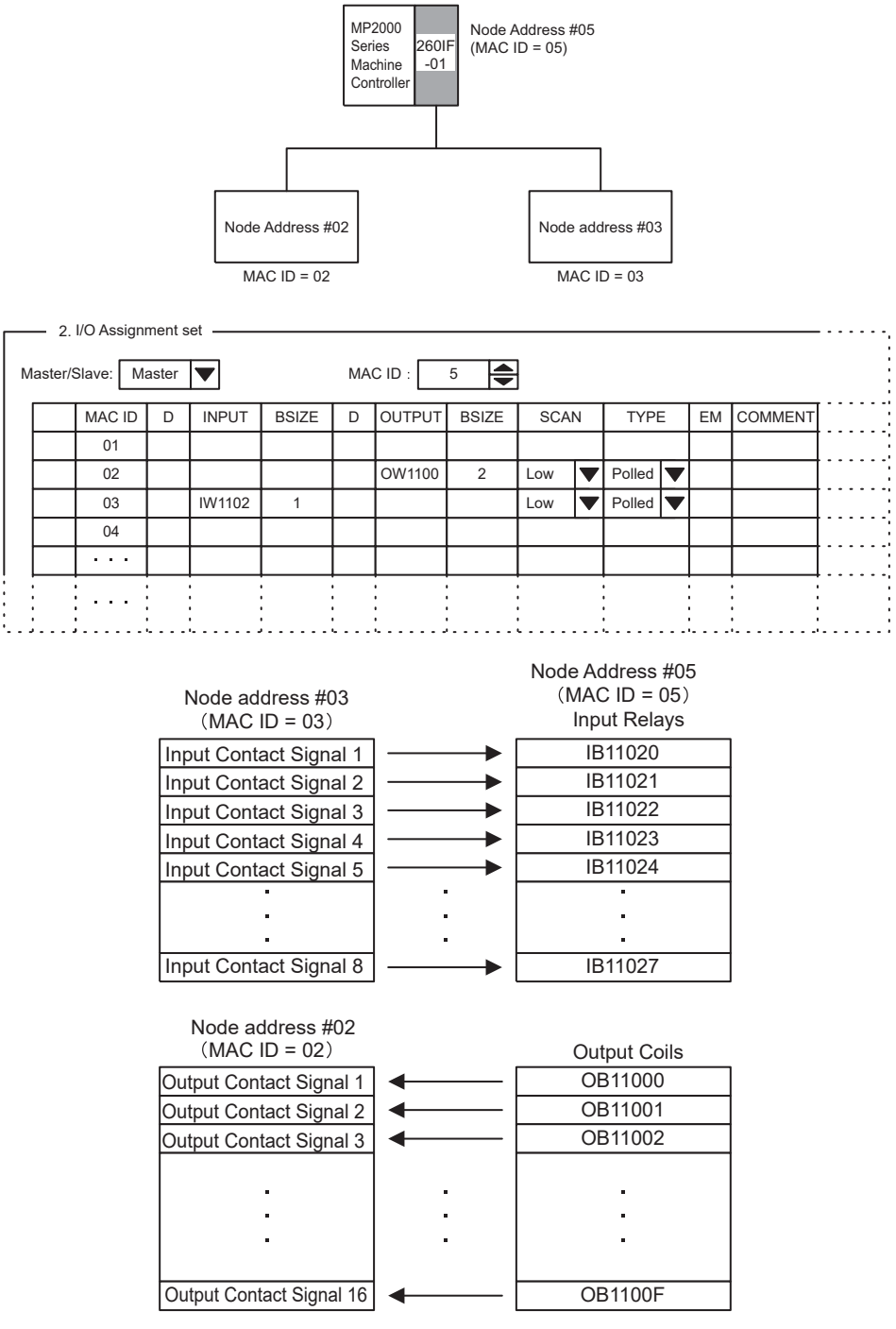


7.4.6 Parameter Setting Examples

This section describes examples of setting the 260 transmission configuration for using the 260IF-01 both as the Master and as a Slave.

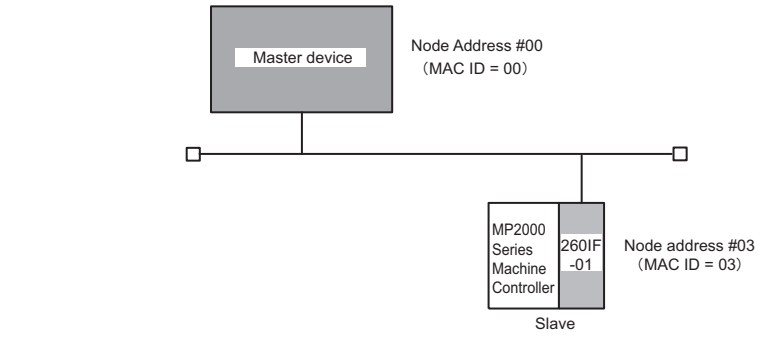
(1) Master I/O Assignment Example

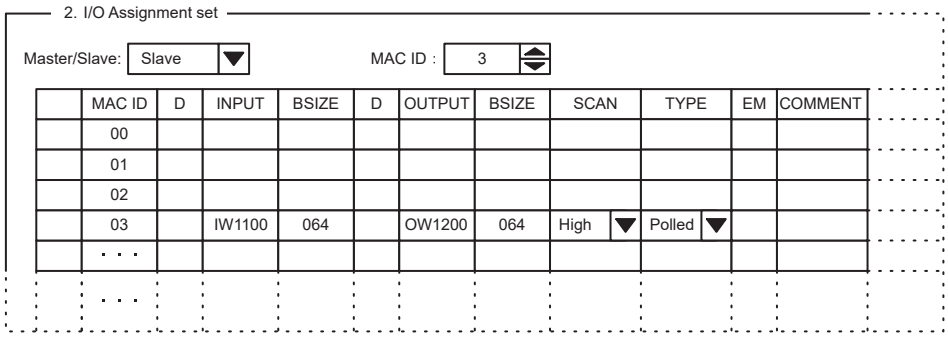
The settings shown below are required to send and receive I/O data between the MP2000 Series Machine Controller, set as the DeviceNet Master (MAC ID = 5), and a 2-byte Output Module (MAC ID = 2) and a 1-byte Input Module (MAC ID = 3).

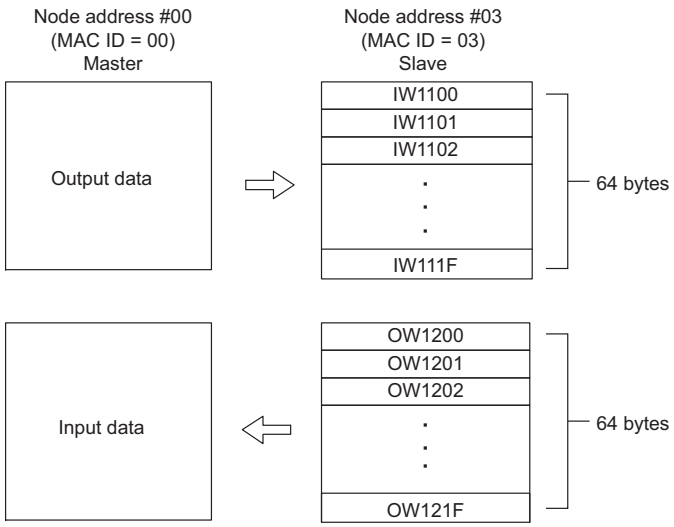


(2) Slave Input/Output Assignment Example

The settings shown below are required to send and receive I/O data between the MP2000 Series Machine Controller and a DeviceNet Master with an I/O data size of 64 bytes each using the MP2000 Series Machine Controller as a DeviceNet Slave (MAC ID = 3).



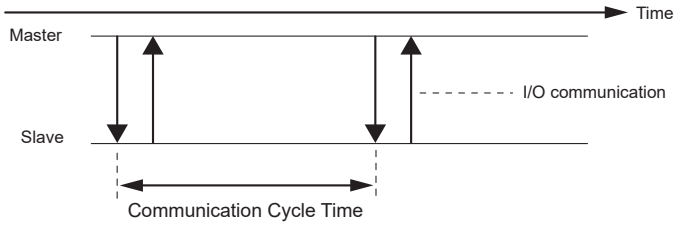




7.4.7 Communication Cycle Time

(1) Overview

The Communication Cycle Time is the time from when the Master transfers I/O data to the DeviceNet Slave device on the transmission path to the next I/O transfer. When the 260IF-01 Module is used as the DeviceNet Master, the Communication Cycle Time must be set. This setting is not required if the 260IF-01 Module is used as a Slave.



(2) Calculation of Communication Cycle Time

The Communication Cycle Time can be calculated as a total value of the communication time with each Slave device.

Communication Cycle Time (ms) = Σ (Communication time with a Slave device)

- The actual communication cycle time differs from the calculation result depending on the length of transmission path and the number of connected devices.

■ Calculation of Communication Time with Each Slave Device

The calculation of communication time of each Slave device can be classified into the following eight cases according to the number of transmission bytes.

- The symbols used in the equations represent the following:

No:	Number of output data bytes
Ni:	Number of input data bytes
TRUNC(No/7):	Quotient of the number of output data bytes (No) divided by 7 (truncated below the decimal point)
TRUNC(Ni/7):	Quotient of the number of input data bytes (Ni) divided by 7 (truncated below the decimal point)
MOD(No/7):	Remainder of No divided by 7
MOD(Ni/7):	Remainder of Ni divided by 7
Tb:	Baud rate of 125 kbps: 8 Baud rate of 250 kbps: 4 Baud rate or 500 kbps: 2

1. Slave device with 8 bytes or less of output data

$(94 + 8 \times No) \times Tb + 600$ [μs]

2. Slave device with 8 bytes or less of input data

$(94 + 8 \times Ni) \times Tb + 600$ [μs]

3. Slave device with 8 bytes or less of I/O data

$\{94 + 8 \times (No + Ni)\} \times Tb + 600$ [μs]

4. Slave device with more than 8 bytes of output data

$\{111 \times TRUNC(No/7)\} \times Tb + \{94 + 8 \times MOD(No/7)\} \times Tb + \underline{300 \times \{TRUNC(No/7) + 1\}}$ [μs]

- MOD(No/7) = 0, the underlined section is replaced with 300 × {TRUNC(No/7)}.

5. Slave device with more than 8 bytes of input data

$\{111 \times TRUNC(Ni/7)\} \times Tb + \{94 + 8 \times MOD(Ni/7)\} \times Tb + \underline{\{300 \times \{TRUNC(Ni/7) + 1\}}$ [μs]

- MOD(Ni/7) = 0, the underlined section is replaced with 300 × {TRUNC(Ni/7)}.

6. Slave device with more than 8 bytes of I/O data

$$\{111 \times TRUNC(No/7)\} \times Tb + \{94 + 8 \times MOD(No/7)\} \times Tb + \underline{300 \times \{TRUNC(No/7) + 1\}}$$
$$+ \{111 \times TRUNC(Ni/7)\} \times Tb + \{94 + 8 \times MOD(Ni/7)\} \times Tb + \underline{300 \times \{TRUNC(Ni/7) + 1\}}\ [\mu s]$$

- If MOD(No/7) = 0, the first underlined section is replaced with $300 \times \{TRUNC\ (No/7)\}$.
- If MOD(Ni/7) = 0, the second underlined section is replaced with $300 \times \{TRUNC(Ni/7)\}$.

7. Slave device with 8 bytes or less of input data, and more than 8 bytes of output data

$$(47 + 8 \times Ni) \times Tb + 300 + \{111 \times TRUNC(No/7)\} \times Tb + \{94 + 8 \times MOD(No/7)\} \times Tb +$$
$$\underline{300 \times \{TRUNC(No/7) + 1\}}\ [\mu s]$$

- If MOD(No/7) = 0, the underlined section is replaced with $300 \times \{TRUNC(No/7)\}$.

8. Slave device with more than 8 bytes of input data, and 8 bytes or less of output data

$$\{111 \times TRUNC(Ni/7)\} \times Tb + \{94 + 8 \times MOD(Ni/7)\} \times Tb + \{\underline{300 \times \{TRUNC(Ni/7) + 1\}}$$
$$+ (47 + 8 \times No) \times Tb + 300\ [\mu s]$$

- If MOD(Ni/7) = 0, the underlined section is replaced with $300 \times \{TRUNC(No/7)\}$.

(3) Precautions on Setting the Communication Cycle Time

If the set **Communication cycle time (Set Time)** is shorter than the actual communication cycle time, the 260IF-01 ignores the set value until the I/O communication command is sent to all the Slave devices. Therefore, the data at the Slave cannot be updated within the set communication cycle time, resulting in delay in the data update.
To avoid delay, calculate the required communication cycle time using the equations given in *7.4.7 (2) Calculation of Communication Cycle Time* on page 7-26 and set the time on the **Transmission Parameters** Tab Page in the **260IF Configuration** Window.
If an appropriate value is set for the **Communication cycle time (Set time)**, the set value in **Set Time** will be the same as the value displayed in **Max. Time**. If the value displayed in **Max. Time** becomes longer than the set value in **Set Time**, adjust the set value so that it is the same as the value displayed in **Max. Time**.

- The **Communication cycle time (Set Time)** is set to 0 by default. If the 260IF-01 is used with the communication cycle time set to the default value, the maximum time (**Max. Time**) will exceed the set value (0). Be sure to set the communication cycle time correctly.

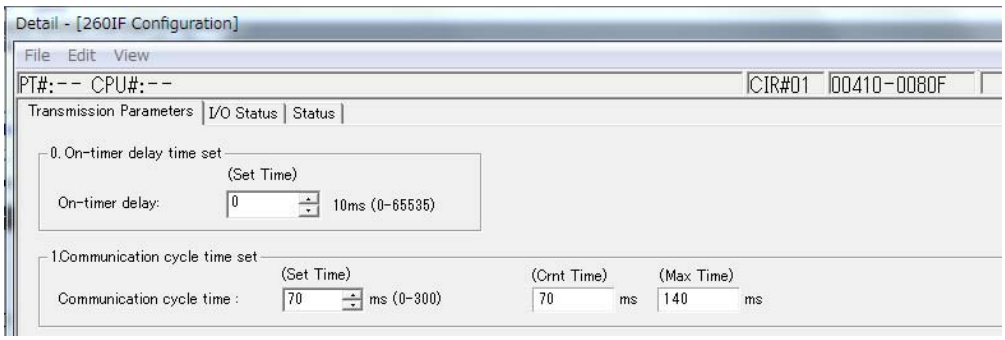


Fig. 7.2 Example of Display When the **Max. Time** Exceeds the **Set Time**

7.4.8 Troubleshooting

If the 260IF-01 Module detects an error in DeviceNet communication, it reports it to the LED indicators, I/O Status, and Status on the **260IF Module Configuration** Window. The following tables show the causes of errors and the corrective actions to be taken.

(1) Errors at the Master

The status of the Master when it is operating normally is shown below.

LED Indicators		Status	I/O Status
MS	NS		
● (Green)	● (Green)	8004	8000

The following table shows the status and corrective actions for when an error occurs with the Master.

- ●: Lit, –: Not lit, ★: Blinking

Status	Check Points				Cause	Corrective Action
	LED Indicators		Status	I/O Status		
	MS	NS				
No DeviceNet communication is performed at all	●	×	–	–	Power is not supplied to the Module.	Check the connections between the 260IF-01 Module and the Mount Base.
	● (Red)	×	–	–	Hardware malfunction	Replace the Module.
	● (Red)	● (Red)	–	–	Hardware malfunction	Replace the Module.
	● (Green)	● (Red)	1010	0000	The DeviceNet device assigned the same MAC ID address exists on the transmission path.	Take either of the following corrective actions: • Change the MAC ID address of this Module and then turn the power to the Module OFF and turn it ON again. • Change the MAC ID address of other DeviceNet device and then turn the power to the Module OFF and turn it ON again.
			1008	–	Bus OFF occurred.	• Check the wiring of the DeviceNet cables and connections of the cable connectors. • Check the source voltage of communication power supply and the connection of the power cable. • Check the source voltage (11 to 24 V) of communication power supply for the DeviceNet connector. • Check the baud rate set at each DeviceNet device on the transmission path. • Check the resistance (121 Ω) and installation conditions of the terminator. • Check the transmission path length. • Replace the Module. • After taking the specified corrective action, turn the power to the Module OFF and then turn it ON again or remove and then insert the DeviceNet connector.
	● (Green)	×	1040	–	Communication power supply error	• Check the wiring of DeviceNet cables and connections of cable connectors. • Check the source voltage of communication power supply and the connection of the power cable. • Check the source voltage (11 to 24 V) of communication power supply for the DeviceNet connector.
	● (Green)	×	0002	0000	DeviceNet line error	• Check the wiring of DeviceNet cables and connections of cable connectors. • Check the baud rate set at each DeviceNet device on the transmission path. • Check the resistance (121 Ω) and installation conditions of the terminator. • Check if the power supply to the DeviceNet device on the transmission path is ON. • Replace the Module.

Status	Check Points				Cause	Corrective Action
	LED Indicators		Status	I/O Status		
	MS	NS				
No DeviceNet communication is performed at all (cont'd)	● (Green)	● (Green)	2004	0000	Connection with a DeviceNet device is not established.	Set the I/O assignments.
	● (Green)	★ (Red)	4004	404D	The I/O data size of the DeviceNet differs from the set value.	Take either of the following corrective actions: • Change the I/O data size in the I/O assignments. • Change the I/O data size of the DeviceNet device to the set value.
			4004	404E	There is no response from the DeviceNet device.	• Check the wiring of DeviceNet cables and connections of cable connectors. • Check the baud rate set at each DeviceNet device on the transmission path. • Check if the power supply to the DeviceNet device on the transmission path is ON.
			4004	4056	The DeviceNet device is idle.	Eliminate the cause that made the DeviceNet device go idle.
Although communication is being performed, the maximum value of the communication cycle time exceeds the set value.	● (Green)	● (Green)	8004	8000	DeviceNet communication load is too heavy (communication cycle time is too short in comparison to the I/O command transmission time).	Increase the setting for the communication cycle time.
Although communication is being performed, there is a delay in refreshing the received data.	● (Green)	● (Green)	8004	8000	DeviceNet communication load is too heavy (communication cycle time is too small in comparison to the I/O response receiving time). Data refresh processing load is too heavy.	• Increase the setting for the communication cycle time of the DeviceNet master device. • Decrease the baud rate. • Increase the setting of MP2000 Series Machine Controller scan time that corresponds to scan set in the I/O assignments.
The MSG-SND function ends in error (the status shows DeviceNet communication is continuing).	● (Green)	● (Green)	8004	8000	Parameter setting error	Check if the MSG-SND function parameters are set correctly. Data address, Data size
	● (Green)	★ (Green)	2004	–	Explicit messages have not been enabled.	Check and set the explicit message settings in the I/O assignment settings.
	● (Green)	● (Red)	4004	40XX	I/O communication error	Eliminate the cause of I/O communication error.
The MSG-SND function stays busy and does not complete.	● (Green)	● (Green)	8004	8000	MSG-SND function parameter setting error	Check if the MSG-SND function parameters are set correctly. Remote station number
					DeviceNet communication load is too heavy (sending an explicit request message is not possible).	• Increase the setting for the communication cycle time. • In the multitask configuration, increase the communication cycle time setting of other Master.

(2) Errors at the Slave

The status of the Slave when it is operating normally is shown below.

LED Indicators		Status	I/O Status
MS	NS		
● (Green)	● (Green)	8004	8003 (Polled) 8005 (Strobed)

The following table shows the status and corrective actions for when an error occurs with the Slave.

- ●: Lit, –: Not lit, ★: Blinking

Status	Check Points				Cause	Corrective Action
	LED Indicators		Status	I/O Status		
	MS	NS				
No DeviceNet communication is performed at all	●	×	–	–	Power is not supplied to the Module.	Check the connections between the 260IF-01 Module and the Mount Base.
	● (Red)	×	–	–	Hardware malfunction	Replace the Module.
	● (Red)	● (Red)	–	–	Hardware malfunction	Replace the Module.
	● (Green)	● (Red)	1010	0000	Duplicated MAC ID setting	Take either of the following corrective actions: • Change the MAC ID address of this Module and then turn the power to the Module OFF and turn it ON again. • Change the MAC ID address of other DeviceNet device and then turn the power to the Module OFF and turn it ON again.
	● (Green)	● (Red)	1008	–	Bus OFF occurred.	• Check the wiring of DeviceNet cables and connections of cable connectors. • Check the source voltage of communication power supply and the connection of the power cable. • Check the source voltage (11 to 24 V) of communication power supply for the DeviceNet connector. • Check the baud rate set at each DeviceNet device on the transmission path. • Check the resistance (121 Ω) and installation conditions of the terminator. • Check the transmission path length. • Replace the Module. After taking the specified corrective action, turn the power to the Module OFF and then turn it ON again or remove and then insert the DeviceNet connector.
	● (Green)	×	1040	–	Communication power supply error DeviceNet line error	• Check the wiring of DeviceNet cables and connections of cable connectors. • Check the source voltage of communication power supply and the connection of the power cable. • Check the source voltage (11 to 24 V) of communication power supply for the DeviceNet connector. • Check the resistance (121 Ω) and installation conditions of the terminator.
	● (Green)	×	0002	0000	DeviceNet line error	• Check the wiring of DeviceNet cables and connections of cable connectors. • Check the setting of the baud rate of each device. • Check the resistance (121 Ω) and installation conditions of the terminator. • Check the operating status of the DeviceNet Master. • Replace the Module.

Status	Check Points				Cause	Corrective Action
	LED Indicators		Status	I/O Status		
	MS	NS				
No DeviceNet communication is performed at all (cont'd)	● (Green)	★ (Green)	2004	0000	Connection with a DeviceNet device is not established.	• Check the setting of scan list in the DeviceNet Master. • Check the wiring of DeviceNet cables and connections of cable connectors. • Check the setting of the baud rate of each device. • Check the operating status of the DeviceNet Master.
					The I/O data size of the DeviceNet device differs from the set value.	• Change the I/O data size in the I/O assignments. • Change the I/O data size of the DeviceNet Master.
	● (Green)	★ (Green)	4004	4000	There is no response from the DeviceNet Master.	• Check the wiring of DeviceNet cables and connections of cable connectors. • Check the setting of the baud rate of each device. • Check the resistance (121 Ω) and installation conditions of the terminator. • Check the operating status of the DeviceNet Master. • Check the source voltage of communication power supply and the connection of the power cable. • Check the source voltage (11 to 24 V) of communication power supply for the DeviceNet connector.
	● (Green)	● (Red)	0000	0004	The MAC ID setting of the switch and the MAC ID setting in the I/O assignments differ from each other.	Take either of the following corrective actions: • Change the MAC ID setting in the I/O assignments. • Change the MAC ID setting of the switch and then turn the power OFF and turn it ON again.
Although communication is performed, the received data is not refreshed.	● (Green)	● (Green)	4004	4003	The DeviceNet Master is idle.	Eliminate the cause that made the DeviceNet Master go idle.
Although communication is being performed, there is a delay in refreshing the received data.	● (Green)	● (Green)	8004	8003	DeviceNet communication load is too heavy (communication cycle time is too short in comparison to the I/O response receiving time). Data refresh processing load is too heavy.	• Increase the setting for the communication cycle time of the DeviceNet master device. • Decrease the baud rate. • Increase the setting of MP2000 Series Machine Controller scan time that corresponds to scan set in the I/O assignments.

7.5 Connection Examples for DeviceNet Communication

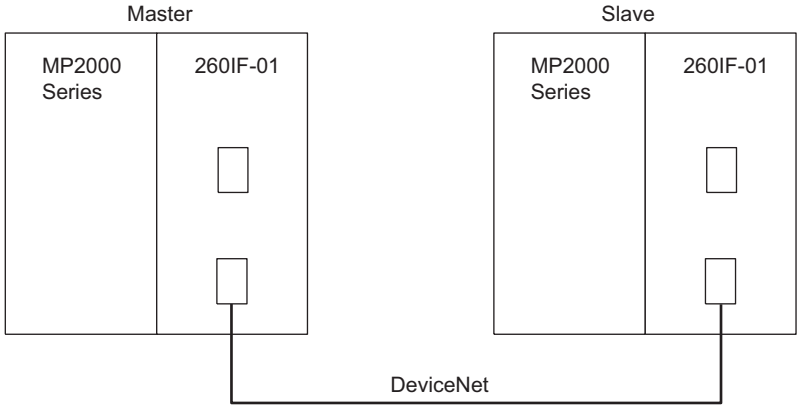
This section shows setting and programming examples for I/O communication and explicit message communication on a DeviceNet network.

7.5.1 I/O Communication Example

This section describes an example of I/O communication between 260IF-01 Modules.

(1) System Configuration

I/O communication is performed using the DeviceNet by installing 260IF-01 Modules on the MP2000 Series Machine Controller.



In this example, two MP2000 Series Machine Controllers are set as shown in the following table to send 2 bytes of I/O data.

Setting Item	Master	Slave
Baud Rate	500 kbps	500 kbps
Master/Slave	Master	Slave
MAC ID	1	2

(2) Master Settings

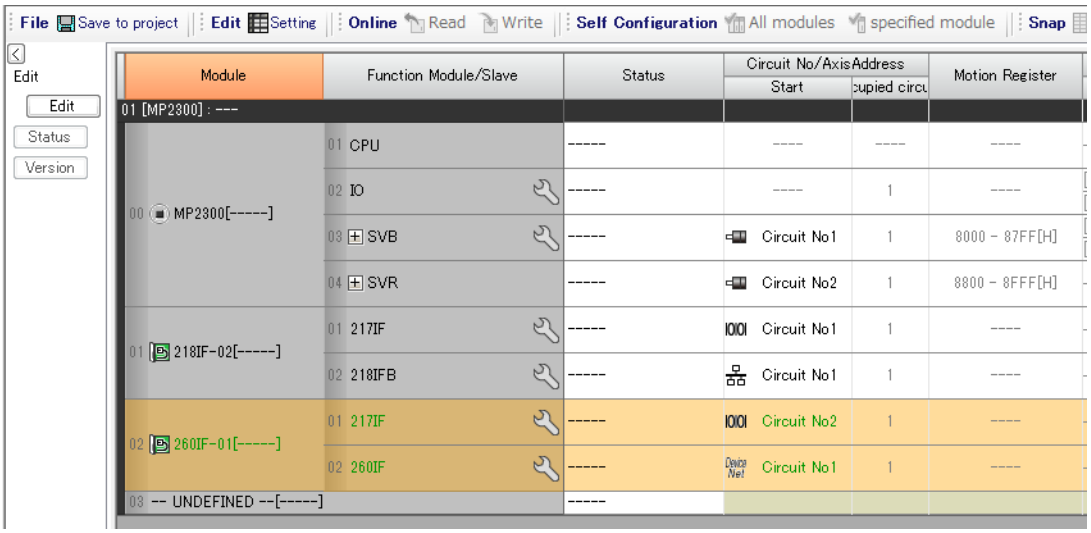
The settings for the 260IF-01 Module that operates as the Master are shown below.

■ Switch Settings

Name	Function	Setting	Remarks
DR0	Baud Rate Setting 0	OFF	Set the baud rate to 500 kbps.
DR1	Baud Rate Setting 1	ON	
X1	Master/Slave mode	ON	Set to Master.
×10	Node Address, 10s Digit Setting	0	Set the MAC ID to 1.
×1	Node Address, 1s Digit Setting	1	

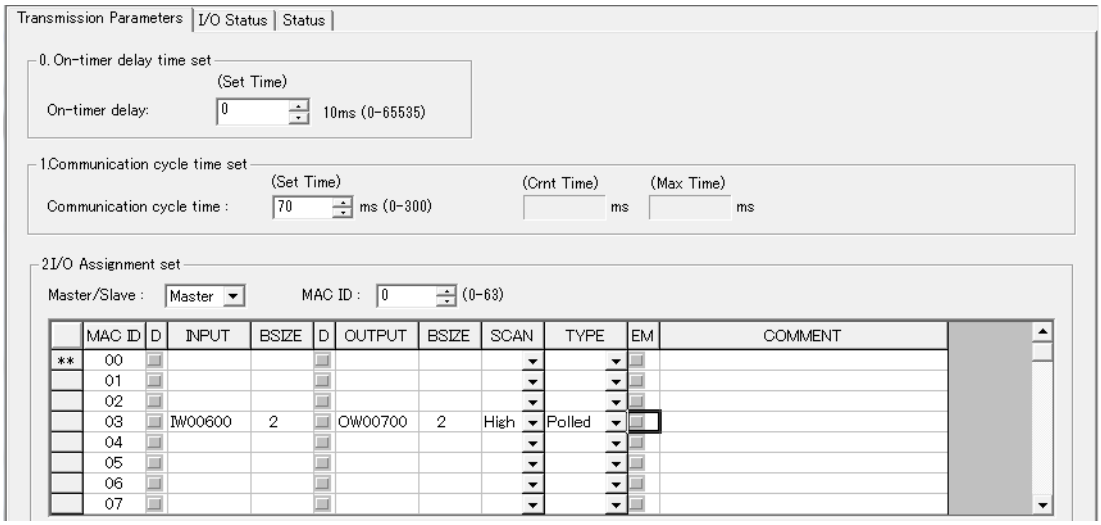
■ Module Configuration Definitions

The following figure shows the settings for the 260IF transmission system.



■ 260IF Transmission Definitions

To perform I/O communication with a Slave 260IF-01 Module, the settings should be as shown below.



After making the required settings, save the parameters to flash memory. The settings will be valid after the power is turned OFF and then ON.

(3) Slave Settings

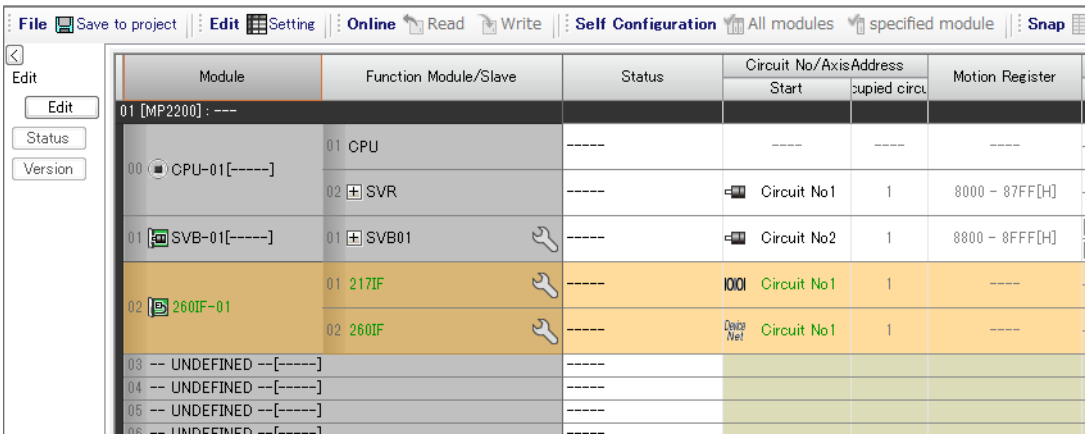
The settings for the 260IF-01 Module that operates as a Slave are shown below.

■ Switch Settings

Name	Function	Setting	Remarks
DR0	Baud Rate Setting 0	OFF	Set the baud rate to 500 kbps.
DR1	Baud Rate Setting 1	ON	
X1	Master/Slave mode	OFF	Set as a Slave.
×10	Node Address, 10s Digit Setting	0	Set the MAC ID to 2.
×1	Node Address, 1s Digit Setting	2	

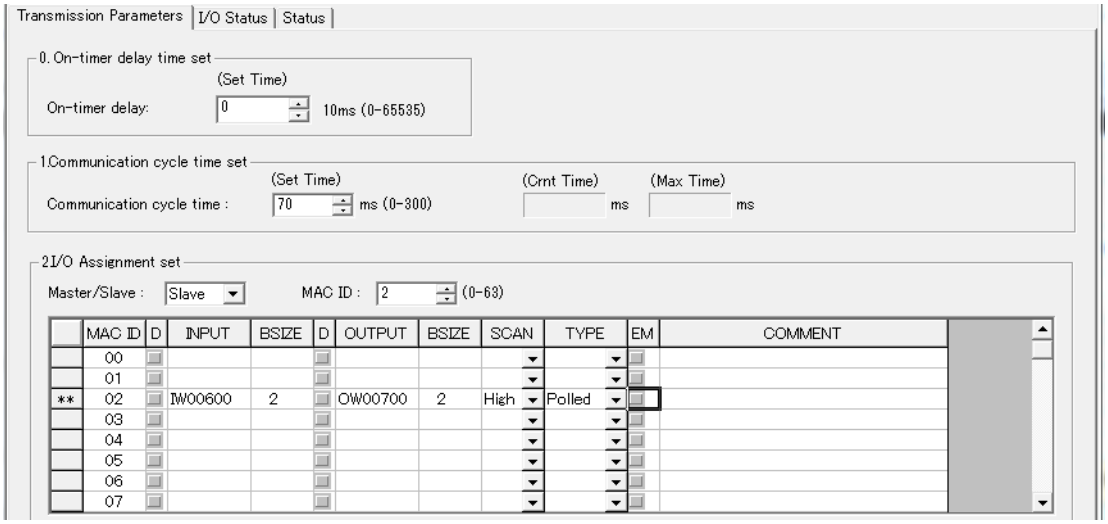
■ Module Configuration Definitions

The following figure shows the settings for the 260IF transmission system.



■ 260IF Transmission Definitions

To perform I/O communication with a Master 260IF-01 Module, the settings should be as shown below.



After making the required settings, save the parameters to flash memory. The settings will be valid after the power is turned OFF and then ON.

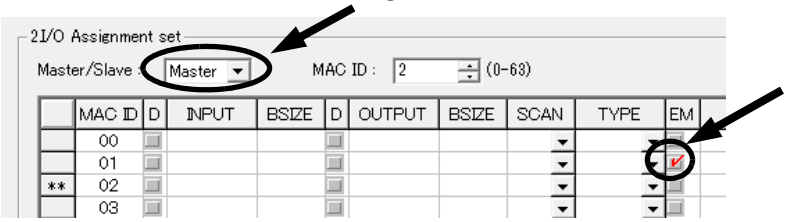
7.5.2 Setting Example for Explicit Message Communication

The MSG-SND function is used to send and receive explicit messages. It sends a request message and receives a response message in one execution.

(1) Precautions on Using Explicit Messages

Observe the following precautions when using explicit messages.

- The Module must be set as the Master.
- **EM** must be selected in the **260IF Transmission Configuration** Window.



(2) Setting the MSG-SND Function

To use explicit messages, the MSG-SND function must be set as shown below. For details on the MSG-SND function, refer to *7.6 Message Send Function* on page 7-40.

[a] Parameter

PARAM No.	IN/OUT	Contents	Remarks
02	IN	Remote station MAC ID	Set the MAC ID of the remote station where the explicit message is to be sent.
04	IN	Function code	Set 3* when sending an explicit message.
12	SYS	System use	The channel number of the current channel is stored. Always set this parameter to 0 from the user program in the first scan after turning ON the power.

* Usually, function code 3 is used to read contents of holding registers. With explicit messages, however, is does not have any meaning.

[b] Inputs

No.	Name	I/O Designation	Contents	Remarks
03	DEV-TYPE	I-REG	Transmission device type	Set 11 to use a 260IF-01 Module.
04	PRO-TYPE	I-REG	Communication protocol	Set 1* to use a 260IF-01 Module.

* Usually, protocol type 1 is MEMOBUS. With a 260IF transmission, however, it does not have any meaning.

(3) Explicit Request Message

[a] Explicit Request Message Format

Write an explicit request message in the data area specified by the data address explained in *7.6.4 (2) [f] PARAM05: Data Address* on page 7-49 on page 7-49 using the following format.

(Offset word addresses)

00000: Request service code (Upper-place byte: Reserved by system, Lower-place byte: Request service code)
00001: Class ID
00002: Instance ID
00003: Attribute ID
00004: (Message data for a write service code)
:

Request service codes include Read (0Eh) and Write (10h).
To use Read request service code, set 8 in the data size* of the parameter list.
To use Write request service code, set 8 plus the write data size in the data size* of the parameter list. The number set in the data size indicates the number of bytes to be written.

* For details on the data size in the parameter list, refer to *7.6.4 (2) [g] PARAM06: Data Size* on page 7-49.

- Refer to the DeviceNet specifications for details on service codes.
- The request service code, class ID, instance ID, and attribute ID are specific to the DeviceNet devices. Contact the manufacturer of the DeviceNet device for any required information.
- The maximum write data size is 248 bytes.

[b] Example of the Explicit Request Message for Reading Vendor ID

If 1000 is specified for the data address in the parameter list (see page 7-45), the explicit request message is written in the following format.

(Offset word addresses)

MW01000: 000Eh (Get_Attribute_Single service code)
MW01001: 0001h (Class ID)
MW01002: 0001h (Instance ID)
MW01003: 0001h (Attribute ID)

(4) Explicit Response Message

[a] Explicit Response Message Format

An explicit response message is written in the data area specified by the data address in the parameter list (see page 7-49) in the following format.

(Offset word addresses)

00000: Response service code (Upper-place byte: Reserved by system, Lower-place byte: Response service code)
00001: Response message data
:

[b] Example of the Explicit Response Message for Reading Vendor ID

If 1000 is specified for data address in the parameter list (see page 7-49), the explicit response message is written in the following format.

(Offset word addresses)

MW01000: 018Eh (Upper byte: Reserved by system, Lower byte (8Eh): Response service code)
MW01001: 002Ch (Vendor ID)

At the same time, 4 is written as the data size in the parameter list (see page 7-49), including the 2 bytes of the response service code.

(5) Procedure to Start Communication

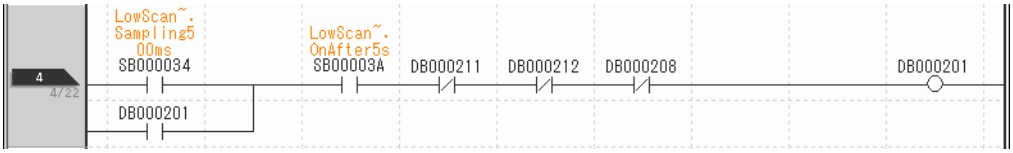
Use the following procedure to start communication.

1. Start the DeviceNet Slave device.
2. Turn ON the **Execute** parameter in the MSG-SND function in the DeviceNet Master 260IF Module to send a message.

Example ladder programming using the MSG-SND function is shown in *(6) Programming Example* on page 7-38. In this programming example, a message is sent when the register set for **Execute** in the MSG-SND function (DB000201) turns ON to start communication with the Slave.

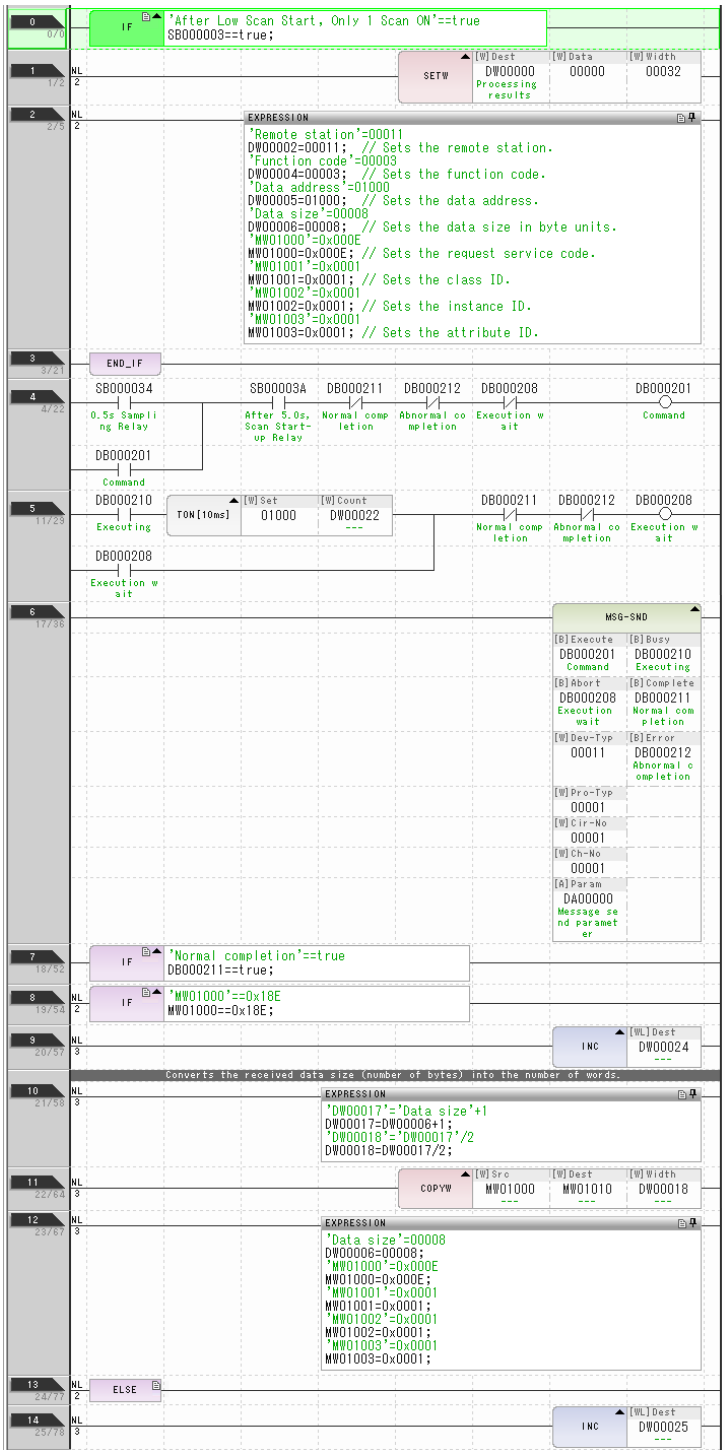
I/O Definitions	No.	Name	Setting Example	Contents
Inputs	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

- In *(6) Programming Example* on page 7-38, the programming is written so that a message will be sent after five seconds has expired after the low-speed (or high-speed) scan has started. When the MSG-SND function has been completed normally, execution of the MSG-SND function will be started again by the SB000034, 0.5-s sampling relay, repeatedly sending a message 0.5 s after sending the previous message has been completed normally.



(6) Programming Example

A programming example for the 260IF-01 Explicit messages is shown below.



Sets parameters in the first scan after startup.
(Low-speed scan (DWG.L): SB000003,
High-speed scan (DWG.H): SB000001)

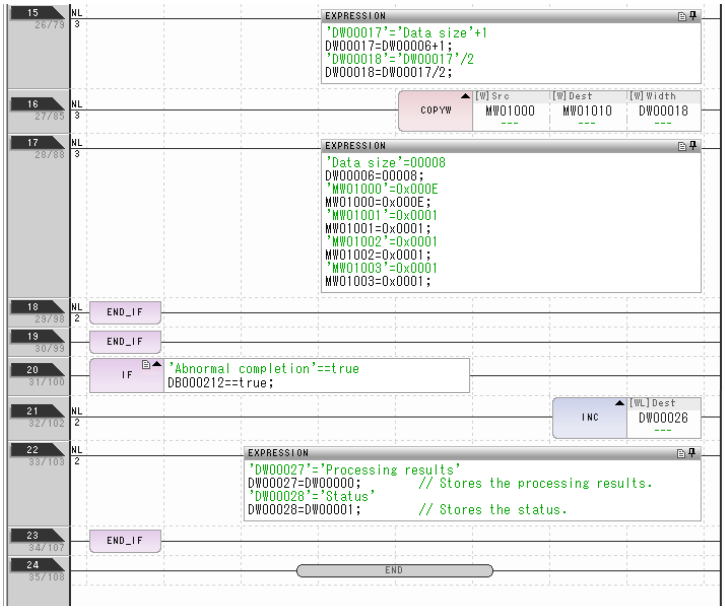
Clears DW register to zero.

Forced stop if processing is not completed within 10 seconds after sending the Send command.

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 1
Cir-No. (circuit number): 1 (connection port)
Ch-No. (channel number): 1 (always 1 for the 260IF-01)
Parameter address: DA00000

The normal completion coil turns ON at normal completion.
Ended with transmission errors.

Increments the normal pass counter.



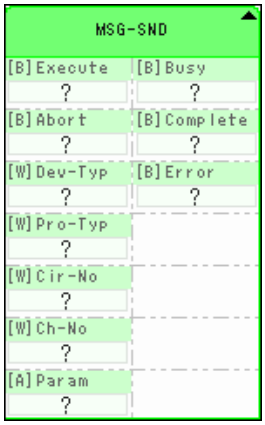
Ended with transmission errors.

Increments the abnormal pass counter.

Stores the processing results and the status.

7.6 Message Send Function

7.6.1 Basic Specifications

Function Name	MSG-SND			
Function	Sends a message to a remote station on the line specified by the Transmission Device Type parameter (DEV-TYP). The function supports multiple protocols. The Execute command must be held ON until the Complete or Error output turns ON.			
Function Definition				
I/O Definitions	No.	Name	I/O Designation*1	Description
Inputs	1	Execute	B-VAL	Send Message command
	2	Abort	B-VAL	Send Message Abort command
	3	Dev-Typ	I-REG	Transmission Device Type MPLINK/CP-215 (MPLINK) = 1, RS-232C/RS-422/RS-485 (217IF) = 5, Ethernet (218IF) = 6, DeviceNet (260IF) = 11, Ethernet (218IFA, 218IFB, 218IFC) = 16
	4	Pro-Typ	I-REG	Communication Protocol MEMOBUS = 1*2, Non-procedure 1*3 = 2, Non-procedure 2*3 = 3
	5	Cir-No	I-REG	Circuit Number MPLINK/CP-215 (MPLINK) = 1 to 8, RS-232C/RS-422/RS-485 (217IF) = 1 to 16, Ethernet (218IF) = 1 to 8, DeviceNet (260IF) = 1 to 8, Ethernet (218IFB) = 1 to 8
	6	Ch-No	I-REG	Transmission Buffer Channel Number MPLINK/CP-215 (MPLINK) = 1 to 12, RS-232C/RS-422/RS-485 (217IF) = 1, Ethernet (218IF) = 1 to 10, DeviceNet (260IF) = 1 to 4, Ethernet (218IFB) = 1 to 10
	7	Param	Address input	Parameter List Leading Address (MA, DA)
Outputs	8	Busy	B-VAL	Processing in progress.
	9	Complete	B-VAL	Processing completed.
	10	Error	B-VAL	Error has occurred.

* 1. The I/O designations are as follows:
B-VAL: I/O is specified as bit data.
I-REG: I/O is specified as integer data. Specify the number of an integer register.
Constants (immediate data) can also be specified for inputs.
Address input: The address of the specified register (any integer register) is passed to the function.

* 2. To use the MEMOBUS, Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, select the MEMOBUS protocol by setting the Communication Protocol parameter (PRO-TYP) to 1. The transmission device will automatically convert to the proper protocol.

* 3. Non-procedure 1: Data is sent in word units.
Non-procedure 2: Data is sent in byte units.

7.6.2 Inputs and Outputs for the Message Send Function

(1) Inputs

The following table gives the registers that can be used for the inputs.

Inputs	I/O Designation	Applicable Registers
Execute Abort	B-VAL	Any bit registers (including those with subscripts) except for # and C registers
Dev-Typ Pro-Typ Cir-No Ch-No	I-REG	Any integer registers (including those with subscripts) Constants
Param	Address input	Any register addresses (including those with subscripts) except for # and C registers

The following sections describe the inputs in more detail.

[a] Execute (Send Message Execute Command)

Specify the bit that will be used to control execution of the Message Send function.
Message send processing is started when the Execute command turns ON. To execute processing, this bit must be turned ON and OFF, e.g., from the ladder program.

- The Execute command must be held at ON until the Complete or Error output turns ON. The message is sent when the Execute command turns ON. To send another message, always turn OFF the Execute command for at least one scan.

[b] Abort (Send Message Abort Command)

Specify the bit that will be used to abort the Message Send function.
Sending the message will be aborted when the Abort command turns ON. The Abort command takes priority over the Execute command.
To abort processing, this bit must be turned ON and OFF, e.g., from the ladder program.

[c] Dev-Typ (Transmission Device Type)

The Dev-Typ input specifies the transmission device type as shown in the following table.

Device	Type Code
MPLINK/CP-215 (MPLINK)	1
RS-232C/422/485 (217IF)	5
Ethernet (218IF)	6
DeviceNet (260IF)	11
Ethernet (218IFA, 218IFB, 218IFC)	16

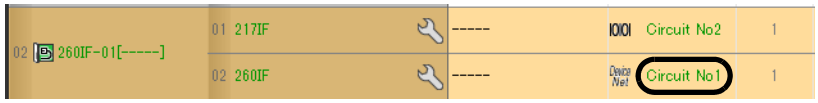
[d] Pro-Typ (Communication Protocol)

The Pro-Typ input specifies the communication protocol as shown in the following table.

Type Code	Communication Protocol	Remarks
1	MEMOBUS	To use the Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, set the type code to 1. The transmission device will automatically convert to the proper protocol.
2	Non-procedure 1 (word units)	Data will be sent in word units using the non-procedure protocol. No response is received from the remote station.
3	Non-procedure 2 (byte units)	Data will be sent in byte units using the non-procedure protocol. No response is received from the remote station.

[e] Cir-No (Circuit Number)

The Cir-No input specifies the circuit number of the transmission device.
Set the circuit number to the value displayed in the **MPE720 Module Configuration** Window.



The valid circuit number ranges are given in the following table.

Device	Valid Circuit Numbers
MPLINK/CP-215 (MPLINK)	1 to 8
RS-232C/422/485 (217IF)	1 to 16
Ethernet (218IF)	1 to 8
DeviceNet (260IF)	1 to 8
Ethernet (218IFB)	1 to 8

[f] Ch-No (Channel Number)

The Ch-No input specifies the channel number of the transmission buffer.

Any channel number can be specified as long as it is within the valid range. If more than one function is being executed at the same time, do not specify the same channel number more than once for the same modem number. (The same channel number can be used as long as the functions are not executed at the same time.)

The valid channel number ranges are given in the following table.

Device	Valid Channel Numbers
MPLINK/CP-215 (MPLINK)	1 to 12
RS-232C/422/485 (217IF)	1
Ethernet (218IF)	1 to 10
DeviceNet (260IF)	1 to 4
Ethernet (218IFB)	1 to 10

<Example>

If a DeviceNet (260IF) transmission device is used, there are four channels of transmission buffers from 1 to 4 for sending and receiving combined, so up to four messages can be sent and received at the same time.

- One MSG-SND function must be programmed for each circuit being used at the same time.
- Refer to *A.3 Transmission Buffer Channels* on page A-9 for information on transmission buffer channels.

[g] Param (Parameter List Leading Address)

The PARAM input specifies the leading address of the parameter list. A parameter list will be automatically created from the 17 words starting with the specified address. Use the parameter list to input the function code and other related parameters. The processing results and status are also output to the parameter list.

- Refer to *7.6.3 Parameter List for DeviceNet Communication (260IF)* on page 7-45 for information on the parameter list.

Example: The following parameter list will be created when the Parameter List Leading Address is set to DA000000.

Register	Parameter F・・・・・・0
DW000000	PARAM00
DW000001	PARAM01
DW000002	PARAM02
DW000003	PARAM03
DW000004	PARAM04
DW000005	PARAM05
DW000006	PARAM06
DW000007	PARAM07
DW000008	PARAM08
DW000009	PARAM09
DW000010	PARAM10
DW000011	PARAM11
DW000012	PARAM12
DW000013	PARAM13
DW000014	PARAM14
DW000015	PARAM15
DW000016	PARAM16

(2) Outputs

The following table gives the registers that can be used for the outputs.

Inputs	I/O Designation	Applicable Registers
Busy Complete Error	B-VAL	Any bit registers (including those with subscripts) except for # and C registers

The following sections describe the outputs in more detail.

[a] Busy (Processing in Progress)

Specify the bit that will report when sending the message is being processed.
The Busy output will be ON while message send processing or abort processing is in progress.
Keep the Execute command or Abort command ON while the Busy output is ON.

[b] Complete (Processing Completed)

Specify the bit that will report when processing to send the message has been completed.
The Complete output will be ON for only one scan after message send processing or abort processing has been completed normally.

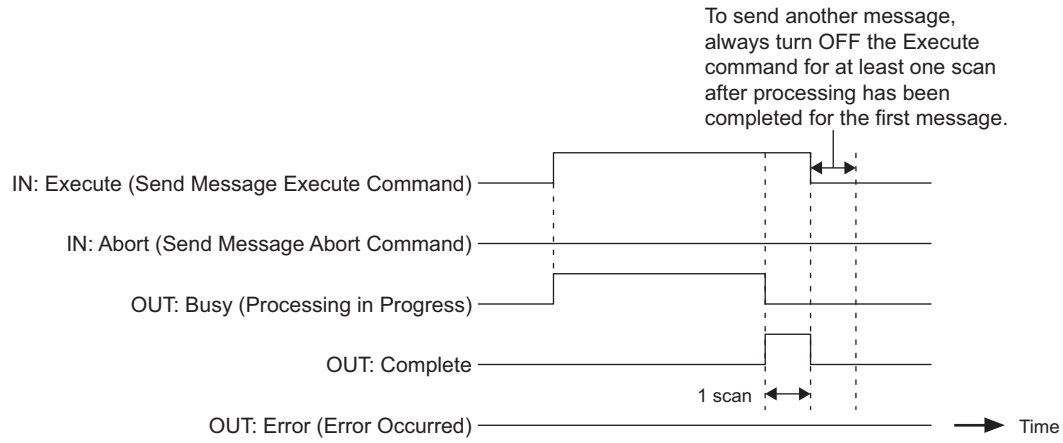
[c] ERROR (Error Occurred)

Specify the bit that will report when an error has occurred while sending the message.
The Error output will turn ON for only one scan when an error occurs.

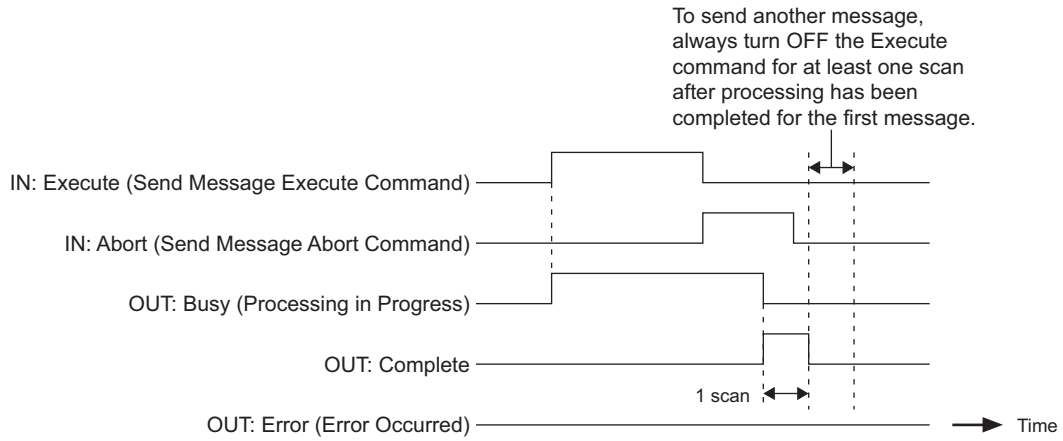
(3) I/O Timing Chart

The following timing charts show the bit inputs and outputs used with the MSG-SND function.

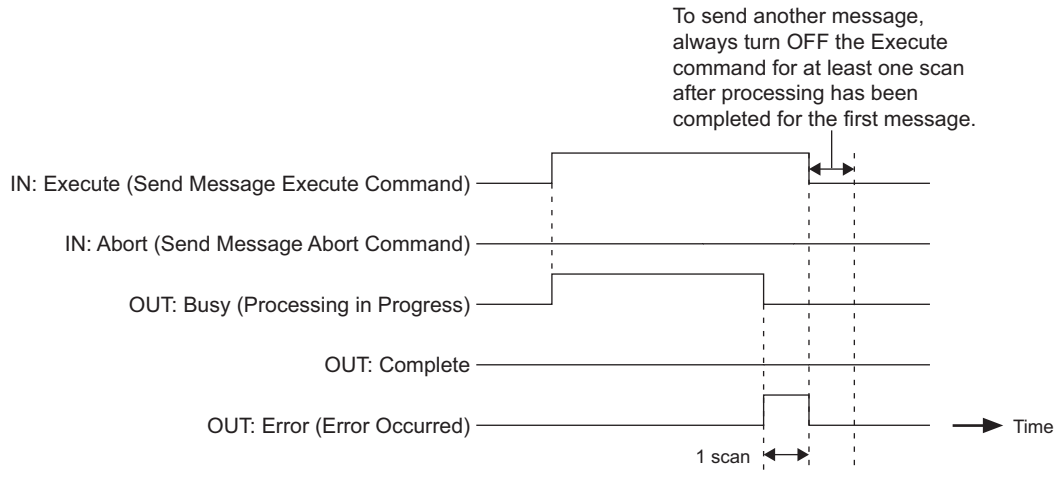
<Normal Processing>



<Abort Processing>



<Error Processing>



7.6.3 Parameter List for DeviceNet Communication (260IF)

The Param input to the MSG-SND function is a parameter list structure consisting of 17 words. The value of the Param input is the leading address (MA or DA) of the parameter list.
Use the parameter list to input the connection number, function code, and other related parameters. The processing results and status are also output to the parameter list.
The parameter list for the MEMOBUS communication protocol is given below.

- The non-procedure protocol cannot be used with DeviceNet communication (260IF).

■ Parameter List for MEMOBUS Protocol

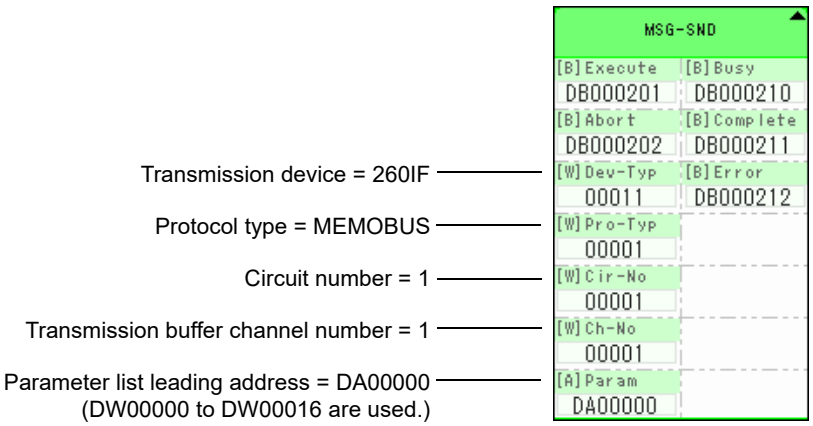
PARAM No.	IN/OUT	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	IN	Remote station number	Specifies the destination.
03	IN	Option	Sets options specific to each transmission device.
04	IN	Function code	Sets the function code to be sent.
05	IN	Data address	Specifies the leading address of the data.
06	IN	Data size	Sets the data size to be read or written.
07	IN	Remote CPU number	Sets the remote CPU number.
08	IN	Coil offset	Sets the offset word address of the coils.
09	IN	Input relay offset	Sets the offset word address of the input relays.
10	IN	Input register offset	Sets the offset word address of the input registers.
11	IN	Holding register offset	Sets the offset word address of the holding registers.
12	SYS	Reserved by the system (1).	
13 to 16	SYS	Reserved by the system (2).	

- IN: Input, OUT: Output, SYS: Used by the system.

7.6.4 Function Setting Example and Parameter Details for Explicit Messages

(1) MSG-SND Function Setting Example

This example shows the settings for using explicit messages with DeviceNet communication (260IF).



The protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 260IF transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *7.6.2 Inputs and Outputs for the Message Send Function* on page 7-41.

(2) Parameter Details for Explicit Messages

This section describes the parameter list in detail for using explicit messages with DeviceNet communication.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARAM00. The lower-place byte is for system analysis.

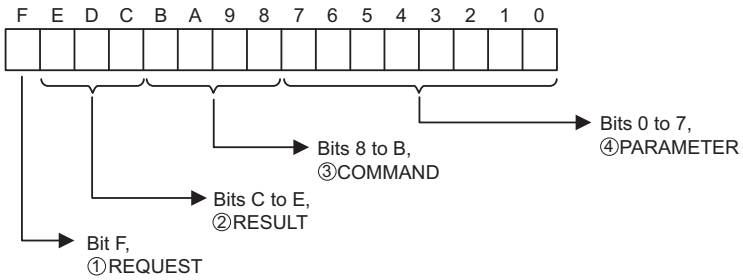
Value of Processing Result	Meaning
00xxH	Processing in progress (Busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	Function code error	An attempt was made to send an undefined function code. Check PARAM04 (function code).
82xxH	Address setting error	One of the following settings is not within the valid range. Check the settings. PARAM05 (data address) PARAM11 (holding register offset)
83xxH	Data size error	The size of the sent or received data is not within the allowable range. Check PARAM06 (data size).
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-SND function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-SND function.
86xxH	MAC address error	The MAC address is not within the allowable range. Check PARAM02 (remote MAC ID).
87xxH	–	Reserved by the system.
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-SND function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-SND function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-SND function is output to these bits.

Code	Abbreviation	Meaning
1	SEND_OK	Sending has been completed normally.
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-SND function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send General-purpose Message with Non-procedure Protocol
2	U_REC	Receive General-purpose Message with Non-procedure Protocol
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response.
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response.
C	MR_SEND	Send MEMOBUS response

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the remote MAC ID is output.

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Remote MAC address outside the allowable range
	02	MEMOBUS response receive monitor time error
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error
	10H	Function code error, or format conversion error
Other values of RESULT	XX	Remote MAC ID

[c] PARAM02: Remote MAC ID

PARAM02 specifies the remote MAC ID.
The valid setting range is given in the following table.

Transmission Device	Remote MAC ID	Remarks
DeviceNet (260IF)	1 to 63	The message is sent to the remote station with the designated MAC ID.

- If the remote station is the 260IF, the remote MAC ID is set to the MAC ID of the Slave.

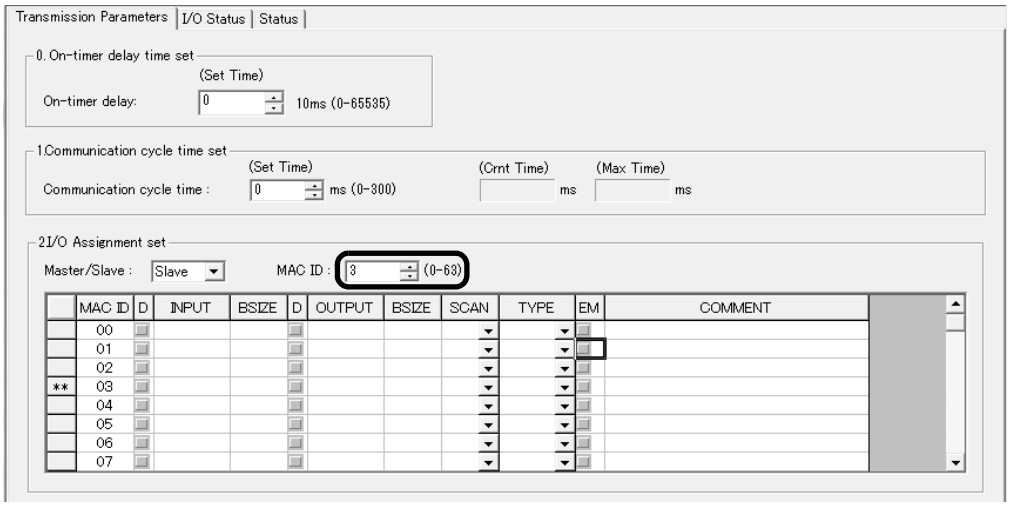


Fig. 7.3 The **260IF Transmission Parameters** Tab Page of the **MPE720 Module Configuration** Window

[d] PARAM03: Option

PARAM03 sets options specific to each transmission device.
PARAM03 is not used for explicit messages. Nothing needs to be set.

[e] PARAM04: Function Code

PARAM04 sets the function code to be sent.
Only function code 03H is used for explicit messages.
The function codes used for explicit messages are listed in the following table.

<Function Codes for Explicit Messages>

Function Code	Applicable Data Type	Function	Protocol Type
03H	–	Explicit Messages	○
Others	–	–	×

- ○: Can be set. , ×: Cannot be set.
- Setting the function code to 03H for DeviceNet (260IF) does not indicate the MEMOBUS command for reading holding registers.

[f] PARAM05: Data Address

PARAM05 sets the leading address of the M registers storing the explicit request message. The explicit response message that is received is also stored in the same area.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

The data addresses used for explicit messages are listed in the following table.

<Data Address Setting Ranges for Explicit Messages>

Function Code	Applicable Data Type	Function	Data Address Setting Range
03H	–	Explicit Messages	0 to 65534 (0 to FFFEH)

[g] PARAM06: Data Size

PARAM06 sets the data size (number of bytes) for explicit request messages. When an explicit response message is received, the data size (number of bytes) of the response message is displayed.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size depends on the request service code, class ID, instance ID, and attribute ID used in the explicit message. These values are specific to the DeviceNet device. For details, ask the manufacturer of the DeviceNet device.
The data size setting ranges used for explicit messages are listed in the following table.

<Data Size Setting Ranges for Explicit Messages>

Function Code	Applicable Data Type	Function	Data Address Setting Range
03H	–	Explicit Messages	1 to 509 bytes

- The data sizes listed in the table are decimal values.
- The data size setting ranges are restricted by the 260IF. They differ from DeviceNet specifications.

[h] PARAM07: Remote CPU Number

PARAM07 sets the remote CPU number.
PARAM07 is not used for explicit messages. Nothing needs to be set.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the offsets for the data read or write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following offset parameters are used for explicit messages.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	(Not used.)
PARAM09	Input relay offset	(Not used.)
PARAM10	Input register offset	(Not used.)
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

[j] PARAM12: Reserved by the System (1)

PARAM12 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[k] PARAM13 to PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

8

261IF-01 Module

This chapter describes the 261IF-01 Module in detail, including the transmission definition methods.

- Before reading this chapter, be sure to read *Chapter 1 Starting Communication Modules*.

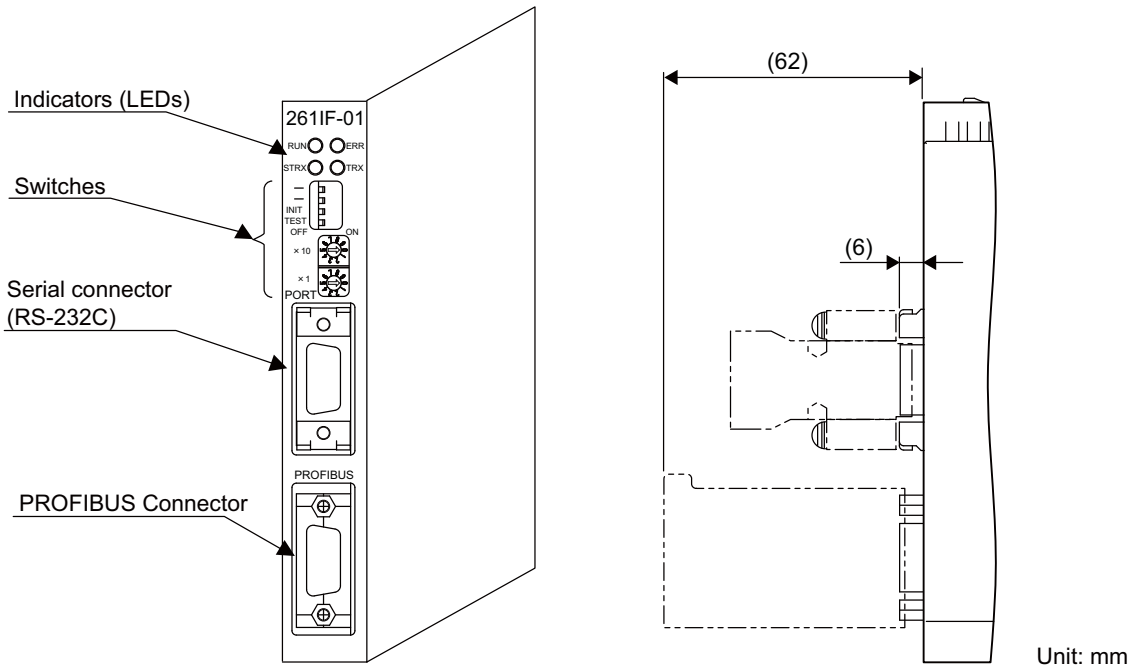
8.1 Overview ---- 8-2
8.1.1 Appearance and Connectors ---- 8-2
8.1.2 System Configuration Example ---- 8-3
8.1.3 Module Specifications ---- 8-4
8.1.4 Indicators ---- 8-6
8.1.5 Switch Settings ---- 8-7
8.1.6 Default Parameters at Initial Startup ---- 8-7
8.2 Connection Specifications ---- 8-8
8.2.1 Connector Specifications ---- 8-8
8.2.2 Cable Specifications ---- 8-9
8.3 RS-232C Communication ---- 8-9
8.4 261IF Transmission Definitions ---- 8-10
8.4.1 261IF Transmission Configuration Window ---- 8-10
8.4.2 I/O Assignment Tab Page ---- 8-12
8.4.3 I/O Map Tab Page ---- 8-14
8.4.4 I/O Status Tab Page ---- 8-15
8.4.5 Status Tab Page ---- 8-16
8.4.6 Saving and Deleting 261IF Transmission Definitions ---- 8-17

8.1 Overview

The 261IF-01 Module is a Communication Module that has an RS-232C serial interface and a PROFIBUS interface mounted in it. Personal computers, HMI devices, and controllers manufactured by other companies can be connected to the 261IF-01 Module via the PORT (RS-232C) or PROFIBUS connectors. The 261IF-01 Module can be connected to the PROFIBUS Class 1 Master as a PROFIBUS-DP (DPV0) slave.

8.1.1 Appearance and Connectors

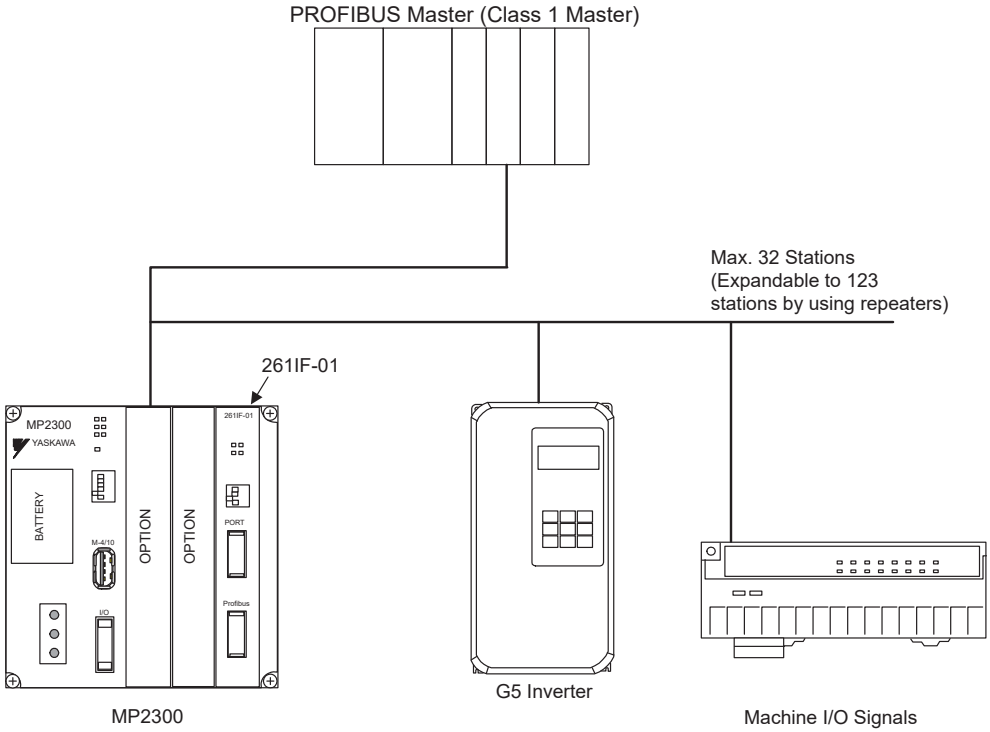
The following diagram shows the appearance of the 261IF-01 Module and gives the external dimensions of the connectors.



8.1.2 System Configuration Example

■ Example of PROFIBUS Connection

The following example shows a system connected to a PROFIBUS network. An MP2000 Series Machine Controller and other devices are connected to the PROFIBUS-DP network with the bus connections.



- The 261IF-01 Module only supports Slave mode. The Slave address can be set between 1 and 64.

8.1.3 Module Specifications

This section provides the specifications of the 261IF-01 Module.

(1) Hardware Specifications

Item	Specification
Name	261IF-01
Model Number	JAPMC-CM2330
Communication Ports	RS-232C: 1 port (PORT)
	PROFIBUS: 1 port (PROFIBUS)
Indicators	Module status LED indicators RUN (green) ERR (red) STRX (green) TRX (green)
Setting Switches	INIT TEST × 10 × 1
Dimensions (mm)	125 x 95 mm (H x D)
Mass	90 g

(2) Transmission Specifications

■ RS-232C Transmission Specifications

Item	Specification
Connector	9-pin D-sub (female)
Transmission Distance	15 m max.
Baud Rate	9600 or 19200 bps
Access Mode	Asynchronous (start-stop synchronization)
Communication Type	Half duplex
Communication Modes	Message communication and engineering communication
Communication Protocols	MEMOBUS, MELSEC, OMRON, and non-procedure
Media Access Control Method	1:1
Transmission Format (Can be set.)	Data bit length: 7 or 8 bits Stop bits: 1 or 2 bits Parity: Odd, even, or none

■ PROFIBUS Transmission Specifications

Item	Specification
Mounted Functions	DP Slave function (Cannot be used as a Master) • Cyclic Communication (DPV0)
Baud Rate	12 M/6 M/3 M/1.5 M/500 K/187.5 K/93.75 K/45.45 K/31.25 K/19.2 K/9.6 kbps (Auto detect)
Configuration	Implemented by the PROFIBUS Master *1
Slave address	1 to 64 *2
I/O Processing	• Total I/O register area: Max. 122 words • I/O assignments: 61 words max. each for inputs and outputs
Diagnostic Functions	• Status and Slave status display using MPE720 • I/O error display using system register

* 1. The PROFIBUS ID is 05C1.
The GSD file MP2000 261IF.gsd is provided for Master configuration.
GSD file: Defines Slave information. This file is stored in the MPE720 installation disk.

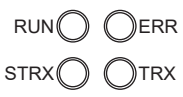
* 2. The PROFIBUS ID can be set between 0 and 125, but the 261IF-01 Module can be set only between 1 and 64.

(3) Operating Environment Specifications

Item		Specification
Environmental Conditions	Ambient Operating Temperature	0 to 55°C
	Ambient Storage Temperature	–25 to 85°C
	Ambient Operating Humidity	30% to 95% (with no condensation)
	Ambient Storage Humidity	5% to 95% (with no condensation)
	Pollution Level	Pollution level 2 (conforming to JIS B 3502)
	Corrosive Gas	There must be no combustible or corrosive gas.
	Operating Altitude	2,000 m above sea level or lower
Mechanical Operating Conditions	Vibration Resistance	Conforming to JIS B 3502 Vibration and Acceleration: 10 to 57 Hz with single-amplitude of 0.075 mm 57 to 150 Hz with fixed acceleration of 9.8 m/s^2 10 sweeps each in X, Y, and Z directions (sweep time: 1 octave/min)
	Shock Resistance	Conforming to JIS B 3502 Peak acceleration of 147 m/s^2 (15 G) twice for 11 ms each in the X, Y, and Z directions
Electrical Operating Conditions	Noise Resistance	Conforming to EN 61000-6-2, EN 61000-6-4, EN 55011 (Group 1 Class A)
Installation Requirements	Ground	Ground to 100 Ω max.
	Cooling Method	Natural cooling

8.1.4 Indicators

The following table shows the status of the 261IF-01 Module shown by the LED indicators.



Indicator	Color	Meaning When Lit	Meaning When Not Lit
RUN	Green	Operating normally	Error occurred or being reset
ERR	Red	A failure has occurred (lit or blinking). Resetting in progress (lit).	Normal
STRX	Green	Sending or receiving RS-232C data	No RS-232C data communication
TRX	Green	PROFIBUS I/O communication in progress	No PROFIBUS I/O communication

■ Status Indication by LED Indicators

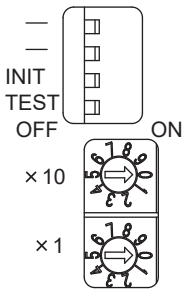
The following table shows the LED indications and their meanings for operating status and errors.

Status	Indicators				Meaning	Description
	RUN	ERR	STRX	TRX		
Power ON	○	●	○	○	Initializing	This is the status immediately after the power to the 261IF-01 is turned ON (initializing). When initialization has been completed, the ERR indicator goes out and the 261IF-01 enters initialized status.
Initialized	●	○	–	–	Waiting for initial settings	The 261IF-01 starts normally and waits for initial settings from the controller (CPU). The STRX and TRX indicators may be lit depending on the connection status.
Normal	●	○	–	–	Operating normally	This is the normal 261IF-01 operating status. The STRX indicator blinks during serial communication. The TRX indicator is lit during PROFIBUS communication.
Error	○	★	–	○	Hardware error (number of blinks:) 1: ROM diagnosis error 2: RAM diagnosis error 3: Shared memory diagnosis error 15: Watchdog timeout error	The 261IF-01 hardware failed. The Module must be replaced.
	●	★	–	○	Processor error (number of blinks:) 2: Slot illegal instruction error 3: CPU address error 5: User break error 6: Trap instruction error	The 261IF-01 processor failed. The Module must be replaced.

- ● : Lit
  ○ : Not lit
  ★ : Blinking
  – : Not relevant

8.1.5 Switch Settings

The following table shows the 261IF-01 Module switch settings.



Label	Name	Status	Function	Factory Setting
–	Reserved	–	Always leave set to OFF.	OFF
–	Reserved	–		
INIT	Initial Startup	ON	For engineering communication. Starts the RS-232C (PORT) port using the default parameters except settings of automatic reception function. The interface is not valid. Given higher priority than the MP2000 Series Machine Controller flash startup and self-configuration startup.	OFF
		OFF	Set to OFF for the MP2000 Series Machine Controller flash startup and self-configuration startup.	
TEST	TEST	ON	System use.	OFF
		OFF	Normal operation (Always leave set to OFF.)	
×10	Node Address, 10s Digit Setting	–	Sets the node address. (Rotary decimal switch)	0
×1	Node Address, 1s Digit Setting	–	Sets the node address. (Rotary decimal switch)	0

8.1.6 Default Parameters at Initial Startup

At initial startup (INIT switch set to ON), the following parameter settings will be made for RS-232C interface.

■ RS-232C Interface (The setting of INIT switch to ON is valid while the power is being supplied.)

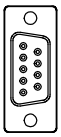
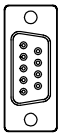
Item	Default Setting		
Communication Protocol	MEMOBUS		
Master/Slave	Slave		
Device Address	1		
Serial I/F	RS-232C		
Communication Mode	RTU		
Data Length	8 bits		
Parity	Even		
Stop Bits	1 bit		
Baud Rate	19.2 kbps		
Send Delay	Disable		
Receive Monitor Time	Disable		
Automatic Reception	Enable		
Interface Register Settings in the Slave		Leading Register	Number of Words
	Read Input Relays	IW0000	32768
	Read Input Registers	IW0000	32768
	Read/Write Coils	MW00000	65535
	Read/Write Holding Registers	MW00000	65535
	Coil/Holding Register Write Range	LO MW00000	
		HI MW65534	

8.2 Connection Specifications

8.2.1 Connector Specifications

This section provides the connector specifications for the 261IF-01 Module.

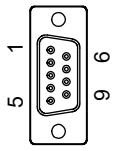
(1) Connector Specifications

Connector Shape	Name	Connector Name	No. of Pins	Connector Model		
				Module	Cable	Manufacturer
PORT 	RS-232C	PORT	9	17LE-13090-27 (D2BC) 9-pin D-sub female connector	17JE-23090-02 (D8B) 9-pin D-sub male connector (Shell included.)	DDK Ltd.
PROFIBUS 	PROFIBUS	PROFIBUS	9	17LE-13090-27 (D33C) 9-pin D-sub female connector	–	DDK Ltd.

(2) Connector Pin Arrangement

■ PORT Connector

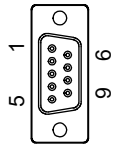
The PORT connector is used to connect the MP2000 Series Machine Controller to computers and HMI devices via an RS-232C connection.



Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	FG	Frame ground	6	–	–
2	SD	Send data	7	SG	Signal ground (0V)
3	RD	Receive data	8	–	–
4	RS	Ready to send	9	ER	Data terminal ready
5	CS	Clear to send	–	–	–

■ PROFIBUS Connector

The PROFIBUS connector is used to connect the MP2000 Series Machine Controller to Masters via a PROFIBUS connection.

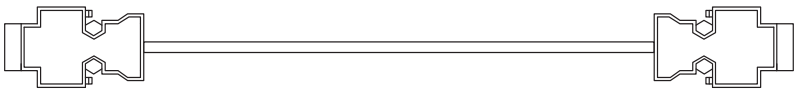


Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	–	–	6	+5V	External power supply
2	–	–	7	–	–
3	TXD/RDX+	Send and receive data +	8	TXD/RDX-	Send and receive data -
4	RTS	Ready to send	9	–	–
5	GND	Ground	–	–	–

8.2.2 Cable Specifications

This section provides the cable specifications for the 261IF-01 Module.

(1) RS-232C Cable (for MPE720 Engineering)

Model Number	Length	External Appearance (JEPMC-W5311-□□-E)
JEPMC-W5311-03-E	2.5 m	
JEPMC-W5311-15-E	15 m	

(2) PROFIBUS-DP Cables

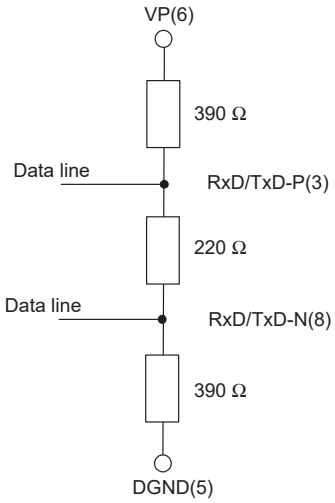
A standard cable is not provided. Make your own cable using commercially available products.
Refer to the product catalog provided on the PROFIBUS User Organization's website (http://www.profibus.com/).
When purchasing a connector, confirm the direction the cable connects to the connector so that it does not interfere with connecting to the RS-232C connector.
Type A cable is recommended.
The maximum cable lengths are shown in the following table. These values assume that a Type A cable is used.

Max. Baud Rate (Kbit/s)	9.6	19.2	93.75	187.5	500	1500	12000
Max. Cable Length (m)	1200	1200	1200	1000	400	200	100

The specifications of a Type A cable are given in the following table.

Parameter	Specification
Impedance (Ω)	135 to 165
Capacitance per unit (pF/m)	30 max.
Loop resistance (Ω/km)	110
Core diameter (mm)	0.64
Core cross-sectional area (mm^2)	0.34 min.

■ Termination



Termination is required on each end of the bus line as shown in the following diagram.

- The 261IF-01 Module does not provide a terminator for termination. Use the terminator provided at the cable side bus, plug or connector.

8.3 RS-232C Communication

Refer to *Chapter 3 Serial Communication* for information on the transmission definitions, connection examples, programming examples, the Message Send function, the Message Receive function, and other details on communication through the RS-232C port on the 261IF-01 Module.

8.4 261IF Transmission Definitions

To perform PROFIBUS communication with the MP2000 Series Machine Controller, the 261IF transmission definition file must be created.
This section describes how to define 261IF transmissions using MPE720 windows.

8.4.1 261IF Transmission Configuration Window

To define the 261IF transmissions, first the **Module Configuration** Window is opened on the MPE720, and then the **261IF Transmission Configuration** Window is opened.

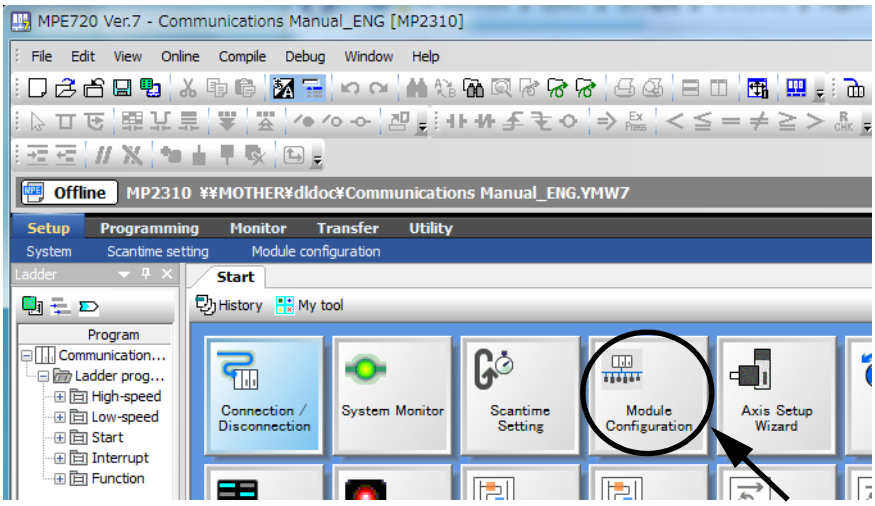
(1) Displaying the Module Configuration Window

Use the following procedure to display the **Module Configuration** Window.

1. Start the MPE720 on the personal computer connected to the Machine Controller and open the project file.

- For information on starting the MPE720, refer to *Machine Controller MP2000/MP3000 Series Engineering Tool MPE720 Version 7 User's Manual (Manual No.: SIEP C880761 03)*.

2. Click the **Module Configuration** Button on My Tool View.



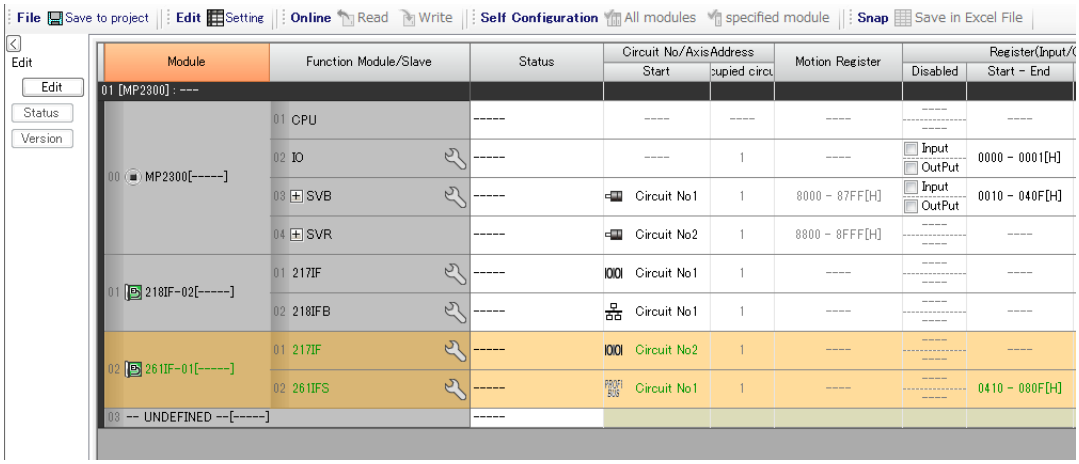
The **Module Configuration** Window will be displayed.

(2) Displaying the 261IF Transmission Configuration Window from the Module Configuration Window

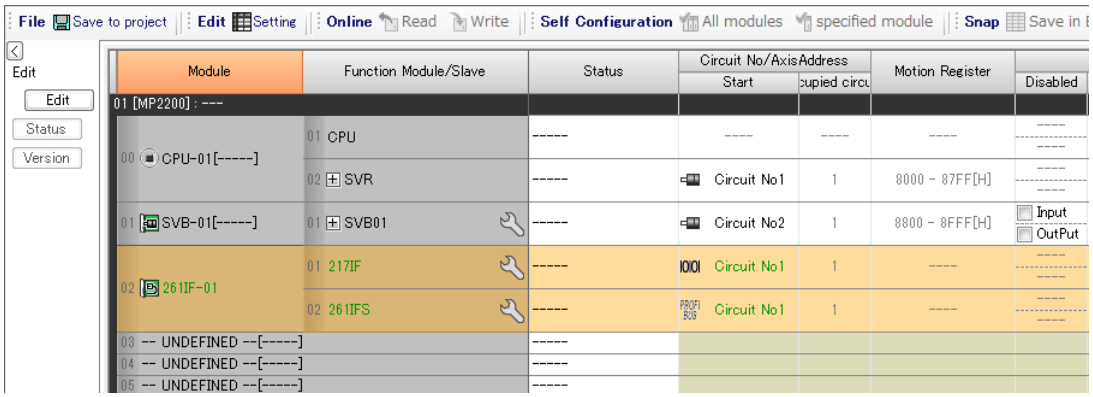
As the result of executing self-configuration, all Option Modules connected to the Machine Controller will be displayed in the **Controller** Area of the **Module Configuration** Window. (Refer to *1.5 Executing Self-configuration* on page 1-26.)

In the **Controller** Area, select the cell for 261IF-01, and then double-click the slot number cell for the 261IF in the **Module Details** Area. The **261IF Transmission Configuration** Window (see next page) will be displayed.

<MP2300/MP2300S/MP2310 Module Configuration Window>



<MP2100M/MP2200 Module Configuration Window>



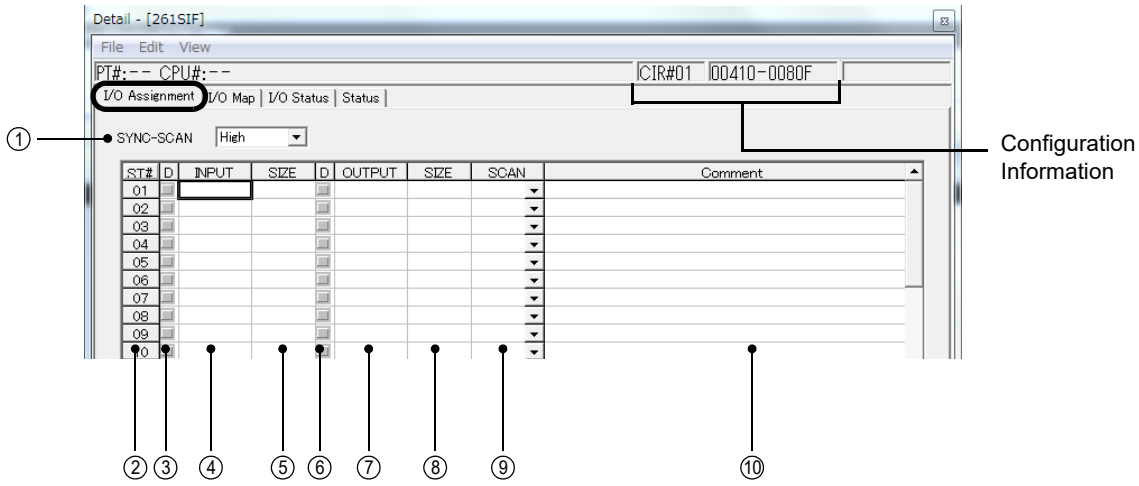
In Online Mode, the 261IF transmission definitions data saved in the Machine Controller are displayed. In Offline Mode, the definitions data saved in the personal computer on which MPE720 is running is displayed.

The **261IF Transmission Configuration** Window is composed of four tab pages: **I/O Assignment**, **I/O Map**, **I/O Status**, and **Status**. These tab pages are used to set the definitions and monitor the settings.

- If the **Transmission Configuration** Window is being opened for the first time, a "new file" message box will be displayed and the **261IF Transmission Configuration** Window will be displayed when the **OK** Button is clicked.
- If the cell for the 217IF in the **Module Details** Area is double-clicked, the **217IF Transmission (serial communication) Configuration** Window will be displayed. Refer to *Chapter 3 Serial Communication* for details on serial communication.

8.4.2 I/O Assignment Tab Page

Set the parameters required for using PROFIBUS communication.



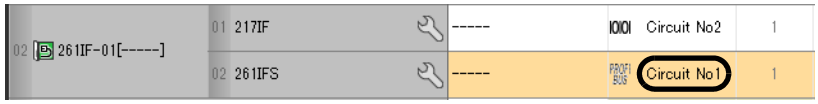
- Do not set the same address for inputs and outputs. If either the input or output field is empty, an error will occur.

■ Configuration Information

Displays the 261IF-01 Module configuration information. The configuration information is the same as the information displayed in the **Module Details** Area in the **Module Configuration** Window.

RACK#: The rack number of the rack in which the 261IF-01 is defined.
Slot#: The slot number of the slot in which the 261IF-01 is defined.
CIR#: The circuit number of the 261IF-01.
Register number: The leading and ending I/O register numbers of the 261IF-01.

- The configuration information can also be changed in the **Module Details** Area of the **Module Configuration** Window. Do not use a circuit number that is already being used by another 261IF.



- When the MSG-SND or MSG-RCV function is used, set the circuit number displayed here for Circuit Number.

■ Setting the Transmission Parameters

① SYNC-SCAN

Select the MP2000 Series Machine Controller scan with which the data refresh cycle will be synchronized for the 261IF transmission system.

High: High-speed scan
Low: Low-speed scan

② ST# (Station Number)

This is the PROFIBUS station number of the station for which I/O is being assigned. Although PROFIBUS addresses can be set between 1 and 125, set a number between 1 and 64 for a 261IF-01 Module.

③ D

This setting enables or disables receiving input data to the 261IF-01 Module from a connected device. Click the checkbox to select it and disable I/O data communication. Click the checkbox again to clear the selection.

④ INPUT (Leading Input Register)

Set the leading address of the input area (input register IWxxxx) to be assigned to the corresponding device. Set a hexadecimal word address. The byte arrangement is little-endian.

- Do not set the same address as for the output.

⑤ SIZE

Set the size of the input area (input register IWxxxx) assigned to the corresponding device in words. Set the size in decimal between 1 and 61 words per Slave.
If the **SIZE** is set to 0, no input data will be sent and received.

⑥ D

This setting enables or disables receiving output data to the 261IF-01 Module from a connected device. Click the checkbox to select it and disable I/O data communication. Click the checkbox again to clear the selection.

⑦ OUTPUT (Leading Output Register)

Set the leading address of the output area (output register OWxxxx) to be assigned to the corresponding device. Set a hexadecimal word address.

- Do not set the same address as for the input.

⑧ SIZE

Set the size of the output area (output register OWxxxx) assigned to the corresponding device in words. Set the size in decimal between 1 and 61 words per Slave. The word arrangement is the same as for the input registers.
If the **SIZE** is set to 0, no output data will be sent and received.

⑨ SCAN

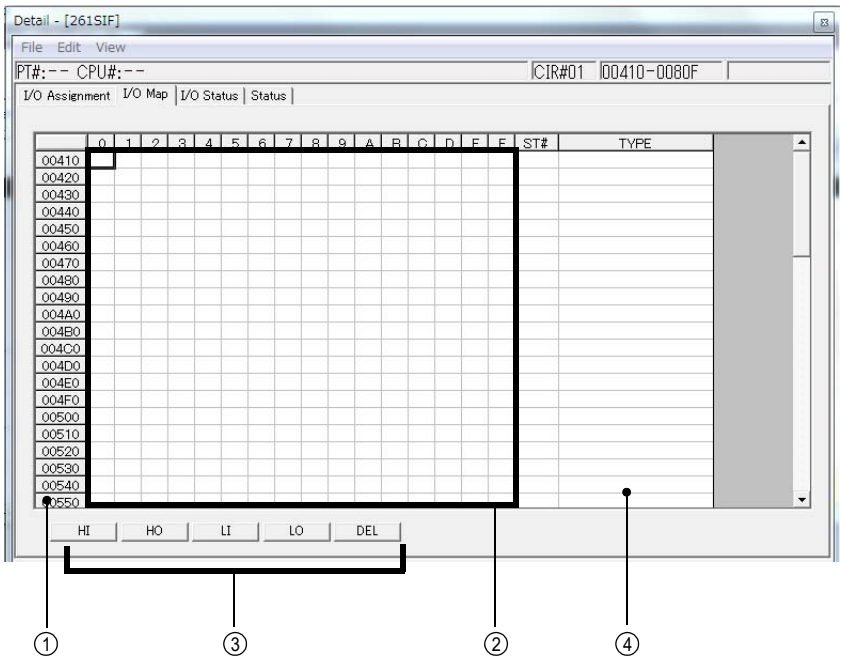
The data refresh cycle (SCAN) is the timing at which I/O data is refreshed between the connected device and the 261IF-01 Module.
The data refresh cycle is not synchronous with I/O data communication.
If **HIGH** is set, I/O data is refreshed during the high-speed scan of the MP2000 Series Machine Controller.
If **LOW** is set, I/O data is refreshed during the low-speed scan of the MP2000 Series Machine Controller.

⑩ COMMENT

A character string, such as a device name or type, can be entered as a comment. Up to 32 characters can be entered.

8.4.3 I/O Map Tab Page

Set the I/O assignments for the Machine Controller.
Setting is possible only for the registers for which I/O assignments have been made on the **I/O Assignment** Tab Page.



① Register No.

The register numbers from the leading I/O register number to the ending I/O register number that have been set for the 261IF-01 Module in the **Module Configuration** Window are displayed.

② I/O Assignment Area (0 to F)

0 to F indicate the 1st digit of each register number. The area enclosed with the rows with register numbers and 0 to F columns is the I/O map assignment area. The I/O assignments set on the **I/O Assignment** Tab Page will be displayed.

HI: Indicates a high-speed scan input.
HO: Indicates a high-speed scan output.
LI: Indicates a low-speed scan input.
LO: Indicates a low-speed scan output.

③ Assignment Setting Buttons

The Assignment Setting Buttons assign I/O to the registers in the I/O map area. In this operation, assignment is possible only for the registers that are set on the **I/O Assignment** Tab Page. Assignments for output settings are not allowed for the registers assigned as the input registers in the I/O assignments. Similarly, it is not possible to assign the input settings to registers assigned for output registers in the I/O assignments.
The function of each button is given below.

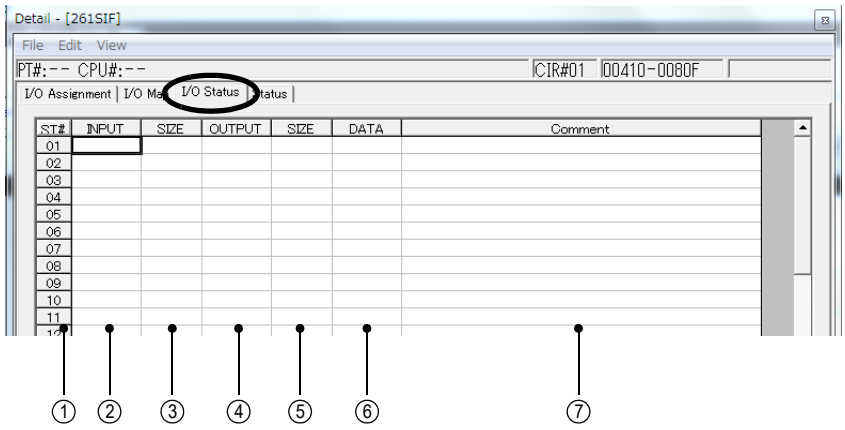
HI: Sets a high-speed scan input.
HO: Sets a high-speed scan output.
LI: Sets a low-speed scan input.
LO: Sets a low-speed scan output.
DEL: Deletes the assignment.

④ ST# (Station Number)

Displays the station number of the I/O registers set on the **I/O Assignment** Tab Page.

8.4.4 I/O Status Tab Page

The **I/O Status** Tab Page displays the 261IF transmission status. The settings cannot be changed.



① ST# (Station Number)

Displays the station number on the PROFIBUS network.

② INPUT (Leading Input Register)

Displays the leading address of the input area (input registers IWxxxx) assigned to the corresponding device.

③ SIZE

Displays the size of the input area (input registers IWxxxx) assigned to the corresponding device in words.

④ OUTPUT (Leading Output Register)

Displays the leading address of the output area (output register OWxxxx) assigned to the corresponding device.

⑤ SIZE

Displays the size of the output area (output register OWxxxx) assigned to the corresponding device in words.

⑥ DATA (I/O Status)

Displays the I/O status code. The meaning of each code is given in the following table.

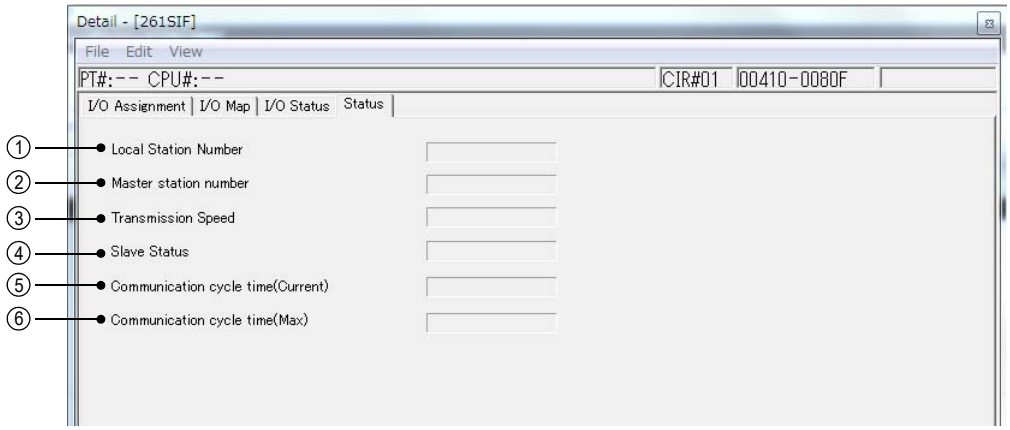
0000H: I/O communication is not being performed.
0001H: I/O communication is normal.
0002H: I/O communication error; communication parameter error
0004H: I/O communication error; configuration error
0008H: I/O communication error; I/O command from the Master is lost.

⑦ COMMENT

Displays the comment set for each device on the **I/O Assignment** Tab Page.

8.4.5 Status Tab Page

The **Status** Tab Page displays the station number, baud rate, and communication status of the 261IF-01 Module. The settings cannot be changed.



① Local Station Number

Displays the station number set for the 261IF-01 Module.

② Master Station Number

Displays the station number of the Master station connected on the PROFIBUS network.

③ Transmission Speed

Displays the current baud rate. Although the 261IF-01 Module is able to operate up to a baud rate of 12 Mbps, it operates at the baud rate selected by the Master when the transmission system is started.

④ Slave Status

Displays the status of the 261IF-01 Module using a message. Codes and their meanings are given below.

WAIT_PRM: Waiting for the communication parameters to be sent from the Master. *1, *2
WAIT_CFG: Waiting for the configuration information to be sent from the Master.
DATA_EXCH: I/O communication in progress

* 1. If writing output data stops during the DATA_EXCH status due to an MP2000 Series Machine Controller error, the 261IF-01 Module stops cyclic communication after 800 ms from the stopping refreshing outputs of the MP2000 Series Machine Controller and then it enters the WAIT_PRM status. Even after moving to WAIT_PRM status, the 261IF-01 Module restarts cyclic communication if it normally receives the information necessary for initializing communication from the Master.
* 2. If, during the DATA_EXCH status, data is not received for the time set at the initialization of communication by the parameters sent from the Master, the 261IF-01 Module stops cyclic communication and enters WAIT_PRM status (waiting for re-initialization of communication). It restarts cyclic communication when it normally receives the information necessary for initializing communication from the Master.

⑤ Communication Cycle Time (Current)

Displays the current communication cycle time of the 261IF-01 Module.

⑥ Communication Cycle Time (Max.)

Displays the maximum communication cycle time of the 261IF-01 Module.

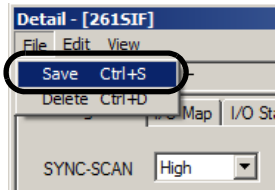
8.4.6 Saving and Deleting 261IF Transmission Definitions

- When changing, saving, or deleting 261IF transmission definitions data, save the data to flash memory in the Machine Controller, and then turn the power to the Machine Controller OFF and then turn it ON again.

(1) Saving the 261IF Transmission Definitions Data

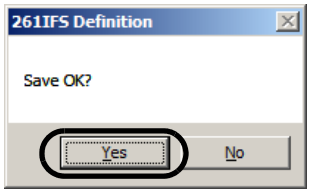
Use the following procedure to save the 261IF transmission definitions data if it has been set or changed. In Online Mode, the definitions data is saved in the Machine Controller and in personal computer running the MPE720. In Offline Mode, the definitions data is saved in the personal computer running the MPE720.

1. Select ***File - Save*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

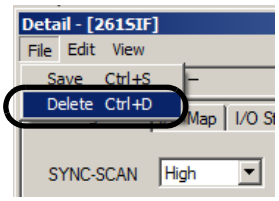
2. Click the **Yes** Button.



(2) Deleting the 261IF Transmission Definitions Data

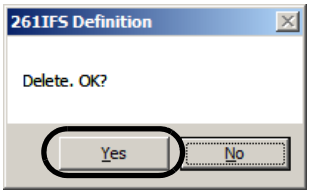
Use the following procedure to delete all of the 261IF transmission definitions data. In Online Mode, the definitions data is deleted from the MP2000 Series Machine Controller and the personal computer running the MPE720. In Offline Mode, the definitions data is deleted from the personal computer running the MPE720.

1. Select ***File - Delete*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

2. Click the **Yes** Button.



9

215AIF-01 Module

This chapter describes the 215AIF-01 Module in detail, including the transmission definition methods and the transmission methods for MPLINK/CP-215 communication.

- Before reading this chapter, be sure to read *Chapter 1 Starting Communication Modules*.

9.1 Overview ---- 9-3
9.1.1 Appearance and Connectors ---- 9-3
9.1.2 System Configuration Examples ---- 9-4
9.1.3 Module Specifications ---- 9-6
9.1.4 Indicators ---- 9-8
9.1.5 Switch Settings ---- 9-9
9.1.6 Default Parameters at Initial Startup ---- 9-10
9.2 Connection Specifications ---- 9-11
9.2.1 Connector Specifications ---- 9-11
9.2.2 Cable Specifications ---- 9-13
9.3 RS-232C Communication ---- 9-15
9.4 MPLINK Transmission Definitions ---- 9-16
9.4.1 MPLINK Transmission Configuration Window ---- 9-16
9.4.2 Transmission Parameters Tab Page ---- 9-18
9.4.3 Link Assignment Tab Page ---- 9-20
9.4.4 I/O Map Tab Page ---- 9-21
9.4.5 Status Tab Page ---- 9-22
9.4.6 Saving and Deleting MPLINK Transmission Definitions ---- 9-24
9.5 Transmission Methods in MPLINK/CP-215 Communication ---- 9-25
9.5.1 Link Communication ---- 9-25
9.5.2 Engineering communication ---- 9-29
9.5.3 Message Communication ---- 9-30
9.5.4 Token Passing ---- 9-32
9.5.5 Relay Function ---- 9-34
9.6 Connection Examples for MPLINK/CP-215 Communication ---- 9-38
9.6.1 Link Communication Example ---- 9-38
9.6.2 Message Communication Example ---- 9-42

9.7 Message Send Function ----------9-46
9.7.1 Basic Specifications ----------9-46
9.7.2 Inputs and Outputs for the Message Send Function ----------9-47
9.7.3 Parameter List for MPLINK/CP-215 Transmissions (MPLINK) ----------9-51
9.7.4 Function Setting Example and Parameter Details for the Extended MEMOBUS Protocol ----------9-52
9.7.5 Function Setting Example and Parameter Details for the Non-procedure Protocol ---9-59
9.8 Message Receive Function ----------9-63
9.8.1 Basic Specifications ----------9-63
9.8.2 Inputs and Outputs for the Message Receive Function ----------9-64
9.8.3 Parameter List for MPLINK/CP-215 Transmissions (MPLINK) ----------9-68
9.8.4 Function Setting Example and Parameter Details for the Extended MEMOBUS Protocol ----------9-69
9.8.5 Function Setting Example and Parameter Details for the Non-procedure Protocol ---9-75

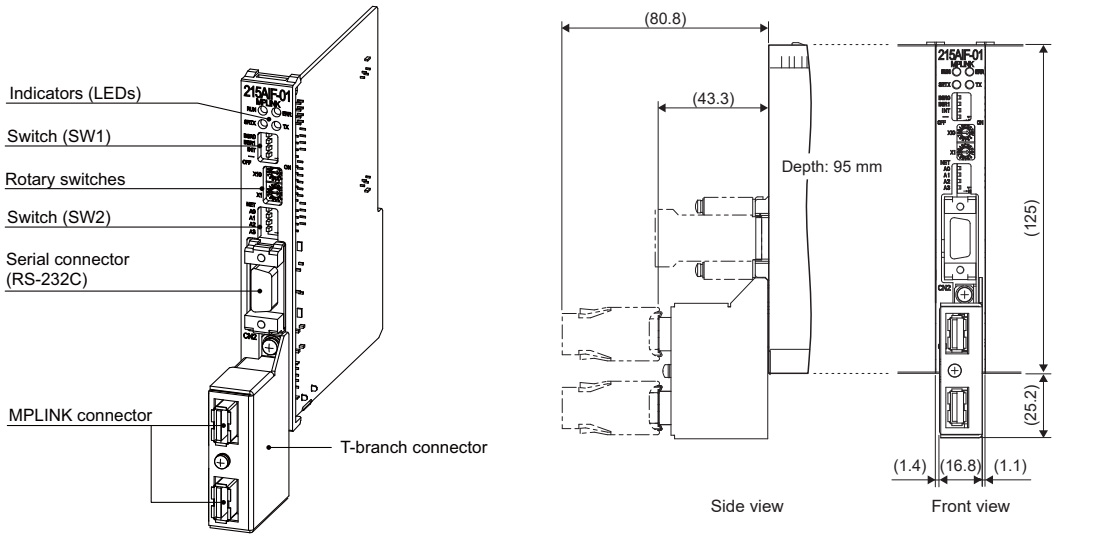
9.1 Overview

- In this chapter, "215AIF-01 Module" indicates both the "215AIF-01 MPLINK Module" and the "215AIF-01 CP-215 Module."

The 215AIF-01 MPLINK Module (model: JAPMC-CM2360) is a Communication Module equipped with a serial interface (RS-232C) and Yaskawa's unique real-time core network MPLINK interface. The MPLINK interface allows communication between 215AIF-01 MPLINK Modules at a baud rate of 10 Mbps.
The 215AIF-01 CP-215 Module (model: JAPMC-CM2361) is a Communication Module equipped with a serial interface (RS-232C) and Yaskawa's unique real-time core network CP-215 interface. The CP-215 interface allows communication with 215IF devices, such as the 215AIF-01 CP-215 Module and 215IF Module, at a baud rate of 4 Mbps or 2 Mbps.
Link, engineering, and message (extended MEMOBUS and non-procedure protocol) communication is possible with both the MPLINK and CP-215 interfaces.

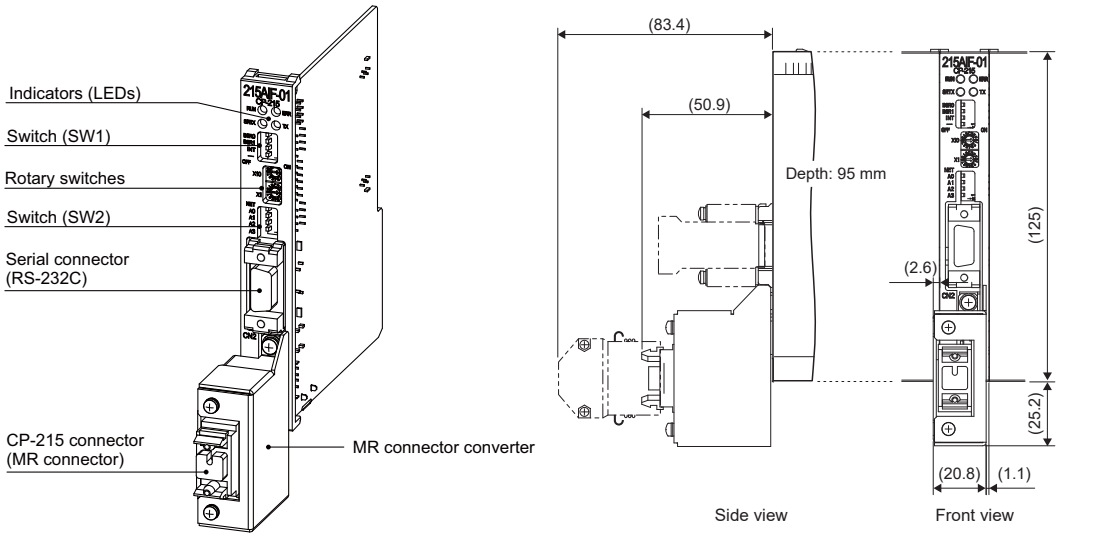
9.1.1 Appearance and Connectors

The following diagram shows the appearance of the 215AIF-01 Module and gives the external dimensions of the connectors.



215AIF-01 MPLINK Module (Model: JAPMC-CM2360)

Unit: mm



215AIF-01 CP-215 Module (Model: JAPMC-CM2361)

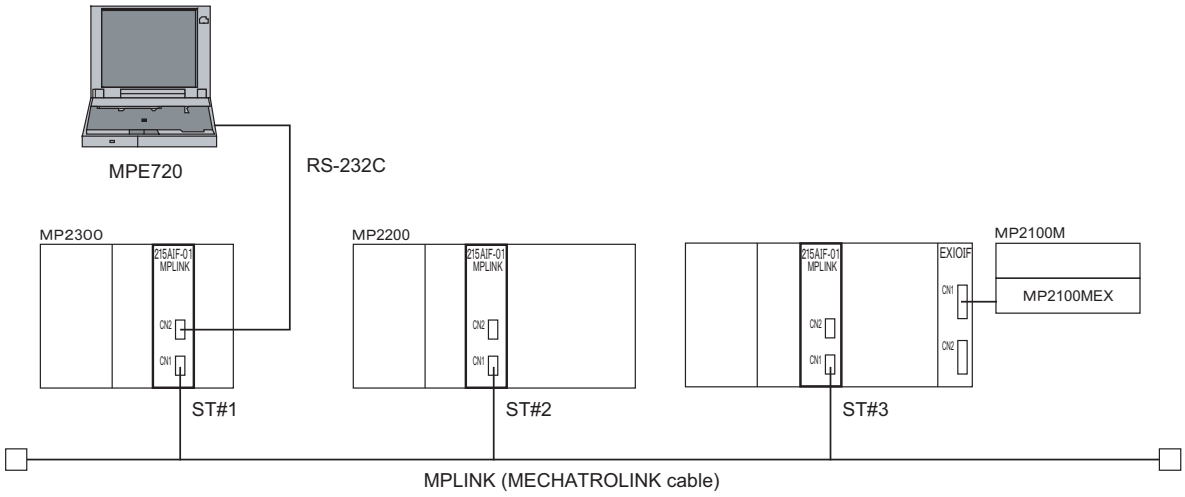
Unit: mm

- Refer to *9.2.1 (3) (3) Mounting a T-branch Connector/MR Connector Converter* on page 9-12 for information on how to mount a T-branch connector or a MR connector converter.
- Two or more 215AIF-01 CP-215 Modules cannot be mounted side by side because a MR-connector converter is wider than the 215AIF-01 CP-215 Module faceplate for 2.6 mm as shown in the diagram. Mount a Module in the slot left side of the slot where a 215AIF-01 CP-215 Module is to be mounted before mounting the 215AIF-01 CP-215 Module.
- T-branch connector (model: JEPMC-OP2310) and MR connector converter (model: JEPMC-OP2320) can be ordered separately.

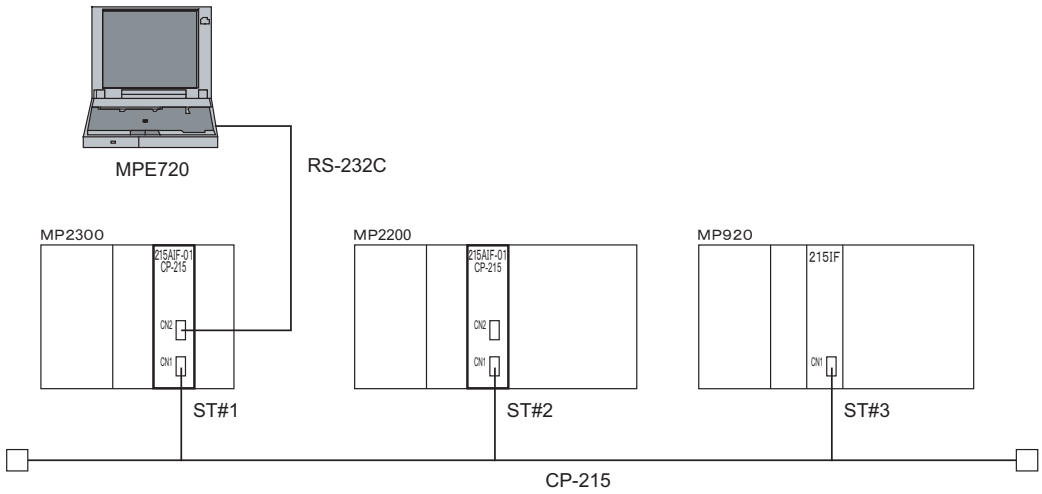
9.1.2 System Configuration Examples

The following diagrams show examples of system configurations using 215AIF-01 Modules.

(1) Example of 215AIF-01 MPLINK Connections



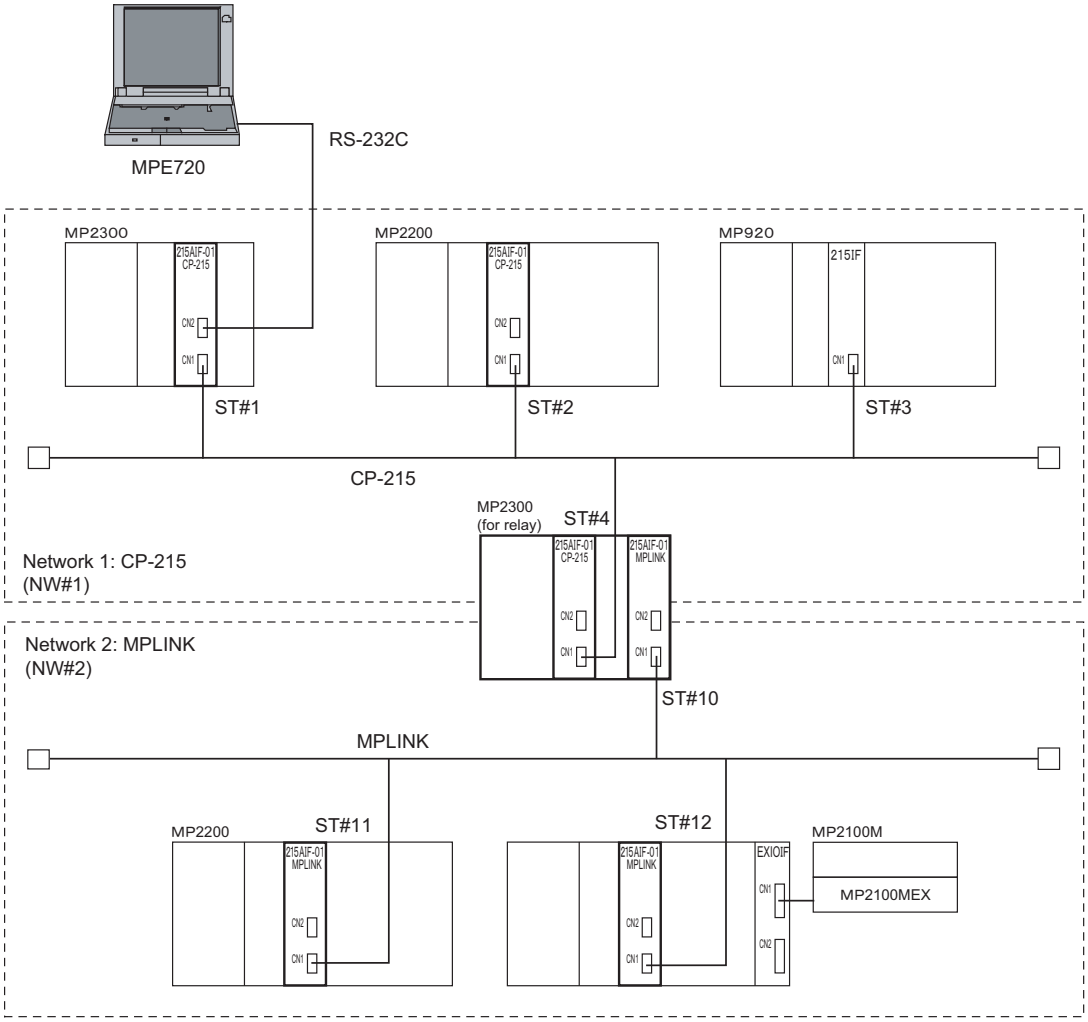
(2) Example of 215AIF-01 CP-215 Connections



- A different station number must be assigned to each 215AIF-01 CP-215 and 215IF Module. Communication will be disabled if the same station number is assigned to more than one Module.

(3) Example of System Configuration Using the Relay Function

Using a relay MP2000 Series Machine Controller with two or more 215AIF-01 Modules mounted, the MP2000 Series Machine Controller can transmit data with control devices connected to different networks and manage and control the entire system using the MPE720.



9.1.3 Module Specifications

This section provides the specifications of 215AIF-01 Module.

(1) Hardware Specifications

Item	Specification	
Name	215AIF-01 MPLINK	215AIF-01 CP-215
Model Number	JAPMC-CM2360	JAPMC-CM2361
Communication Ports	One RS-232C port (CN2)	One RS-232C port (CN2)
	One MPLINK port (CN1)	One CP-215 port (CN1)
Indicators	Module status LED indicators RUN (green) ERR (red) STRX (green) TX (green)	
Setting Switches	BSR0, BSR1, INIT, TEST NET A0 to NET A8 ADRS x 1, ADRS x 10	
Dimensions (mm)	125 x 95 mm (H x D)	
Mass	130 g	

(2) Transmission Specifications

■ RS-232C Transmission Specifications

Item	Specification
Connector	9-pin D-sub (female)
Transmission Distance	15 m max.
Baud Rate	9600 or 19200 bps
Access Mode	Asynchronous (start-stop synchronization)
Communication Type	Half duplex
Communication Modes	Message communication and engineering communication
Communication Protocols	MEMOBUS, MELSEC, OMRON, and non-procedure
Media Access Control Method	1:1
Transmission Format (Can be set.)	Data bit length: 7 or 8 bits Stop bits: 1 or 2 bits Parity: Odd, even, or none

■ MPLINK/CP-215 Transmission Specifications

Item	MPLINK Transmission Specifications	CP-215 Transmission Specifications
Connectors	USB connector: 1 port Converted to MPLINK connector using an externally mounted T-branch connector (JEPMC-OP2310)	USB connector: 1 port Converted to MR connector using an externally mounted MR connector converter (JEPMC-OP2320)
Terminator	Terminators of 130 Ω must be externally installed.	Terminators of 75 Ω must be externally installed.
Cables	MECHATROLINK standard cable	CP-215 Cables (See 9.2.2 (3).)
Baud Rate	10Mbps	2Mbps/4Mbps
Max. Transmission Distance	50 m: 16 stations 100 m: 32 stations (when using a repeater*)	At 4 Mbps: 170 m At 2 Mbps: 270 m
No. of Words for Link Communication	4096 words max. per line, 1024 words max. per station	2048 words max. per line, 512 words max. per station
No. of Words for Message Communication	512 words	512 words
Transmission Control Method	Token passing	Token passing
Data Exchange	N:N	N:N
Error Detection	CRC check, data length check	CRC check, data length check
Max. No. of Connectable Stations	16 (32 when repeaters are used)	32 (64 when repeaters are used)

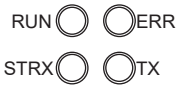
* The repeater (JEPMC-REP2000) for MECHATROLINK-II is used.

(3) Operating Environment Specifications

Item		Specification
Environmental Conditions	Ambient Operating Temperature	0 to 55°C
	Ambient Storage Temperature	–25 to 85°C
	Ambient Operating Humidity	30% to 95% (with no condensation)
	Ambient Storage Humidity	5% to 95% (with no condensation)
	Pollution Level	Pollution level 2 (conforming to JIS B 3502)
	Corrosive Gas	There must be no combustible or corrosive gas.
	Operating Altitude	2,000 m above sea level or lower
Mechanical Operating Conditions	Vibration Resistance	Conforming to JIS B 3502 Vibration and acceleration: 10 to 57 Hz with single-amplitude of 0.075 mm 57 to 150 Hz with fixed acceleration of 9.8 m/s^2 10 sweeps each in X, Y, and Z directions (sweep time: 1 octave/min)
	Shock Resistance	Conforming to JIS B 3502 Peak acceleration of 147 m/s^2 (15 G) twice for 11 ms each in the X, Y, and Z directions
Electrical Operating Conditions	Noise Resistance	Conforming to EN 61000-6-2, EN 61000-6-4, EN 55011 (Group 1 Class A)
Installation Requirements	Ground	Ground to 100 Ω max.
	Cooling Method	Natural cooling

9.1.4 Indicators

The following table shows the status of the 215AIF-01 Module shown by the LED indicators.



Indicator	Color	Meaning When Lit/Blinking	Meaning When Not Lit
RUN	Green	When lit: The micro processor for control is operating normally. Or a transmission error occurs. When blinking: An error has occurred.	An error occurs, or the Module is being initialized.
ERR	Red	A failure has occurred (lit or blinking).	Operating normally
STRX	Green	Sending or receiving RS-232C data	No data being sent or received
TX	Green	MPLINK or CP-215 sending data	No data being sent

■ Status Indication by LED Indicators

Status	Indicators				Meaning	Description
	RUN	ERR	TX	STRX		
Power ON	○	●	○	●	Initializing	This is the status immediately after the power to the 215AIF-01 is turned ON (initializing). When initialization has been completed, the ERR indicator goes out and the 215AIF-01 enters initialized status.
Initialized	●	○	○	–	Waiting for initial setting	The 215AIF-01 starts normally and waits for initial setting from the controller (CPU). The STRX indicator may be lit depending on the connection status.
Normal	●	○	●	–	Operating normally	The 215AIF-01 is performing MPLINK/CP-215 communication normally. The STRX indicator is lit during serial communication.
Error	●	●	●	–	Transmission error	A MPLINK/CP215 transmission error was detected. Out of the token ring. The STRX indicator is lit during serial communication.
	★	★	○	○	Hardware error or switch setting error (number of blinks:) 1: ROM diagnosis error 2: RAM diagnosis error 3: Shared memory diagnosis error 4: CPU diagnosis error 5: FPU diagnosis error 6: SW setting error	When the number of blinks is 1 to 5 times: 215AIF-01 hardware failure. The Module must be replaced. When the number of blinks is 6 times: Check and correct the switch settings.
	○	★	–	–	Software error (number of blinks:) 1: - 2: - 3: Address error (read) exception 4: Address error (write) exception 5: FPU exception 6: General illegal instruction exception 7: Slot illegal instruction exception 8: General FPU prohibition exception 9: Slot FPU prohibition exception 10: Watchdog timeout error	215AIF-01 processor error. The Module must be replaced.

- ● : Lit
  ○ : Not lit
  ★ : Blinking
  – : Not relevant

9.1.5 Switch Settings

(1) SW1 Settings

Set the operation mode such as baud rate. All the switches are set by default to OFF.

Label	Name	Status	Function	Factory Setting
BSR0	MPLINK/ CP-215 Baud Rate Setting	Set the MPLINK/CP-215 communication mode by combination of ON/OFF settings of two switches. • Refer to the following table, *Baud Rate Settings*. • Valid when the INIT switch is ON.		OFF
BSR1				OFF
INIT	Transmission Parameters Startup Selection	ON	For engineering communication. Starts using the default parameters except setting for automatic reception. Given higher priority than the MP2000 Series Machine Controller flash startup and self-configuration startup.	OFF
		OFF	Set to OFF for MP2000 series machine controller flash startup and self-configuration startup.	
–	–	Always leave set to OFF.		OFF

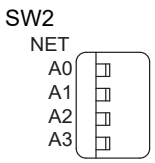
- To perform engineering with MPLINK/CP-215 connection using the MPE720 while the module configuration has not been defined, start the 215AIF-01 Module with the INIT switch set to ON. Communication with the MPE720 can be enabled only by setting the Communication Platform.
- Baud Rate Setting

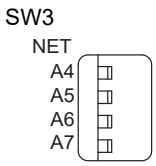
Transmission Method/ Baud Rate / Switch	MPLINK (10 Mbps)	CP-215 (4 Mbps)	CP-215 (2 Mbps)	Not used.
BSR0	OFF	ON	OFF	ON
BSR1	OFF	ON	ON	OFF

(2) SW2 and SW3 Settings

Set the network number (1 to 254) of the MPLINK/CP-215 transmission by setting each bit as shown in the following table.

- The SW3 is mounted on the board and it cannot be viewed on the front face.
- These rotary switches are valid only when the INIT switch is ON.





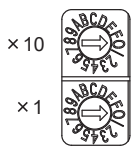
Label	Device Code	Name	Status	Function	Factory Setting
NET A0	S5-4	NETWORK ADDRESS 0	ON/OFF	Sets the least significant 4 bits.	ON
NET A1	S5-3	NETWORK ADDRESS 1	ON/OFF		OFF
NET A2	S5-2	NETWORK ADDRESS 2	ON/OFF		OFF
NET A3	S5-1	NETWORK ADDRESS 3	ON/OFF		OFF
NET A4	S6-4	NETWORK ADDRESS 4	ON/OFF	Sets the most significant 4 bits.	OFF
NET A5	S6-3	NETWORK ADDRESS 5	ON/OFF		OFF
NET A6	S6-2	NETWORK ADDRESS 6	ON/OFF		OFF
NET A7	S6-1	NETWORK ADDRESS 7	ON/OFF		OFF

- A switch will be ON when the switch is pressed to the right side.

(3) Rotary Switches

Two rotary switches are provided to set the station address for the CP-215 transmissions.

- These rotary switches are valid only when the INIT switch is ON.



Label	Name	Setting Range	Function	Factory Setting	Remarks
×10	Node Address, 10s Digit Setting	0 to 6	Sets the node address.	0	Set an address in the range between 1 and 64.
×1	Node Address, 1s Digit Setting	0 to 9		1	

9.1.6 Default Parameters at Initial Startup

At initial startup (INIT switch set to ON), the following parameter settings will be made.

■ MPLINK/CP-215 Interface (The setting of INIT switch to ON is valid only when the power is turned ON.)

Item	Default Setting
Local Station Number	Depends on the address setting of the rotary switches.*
Network No.	Depends on the network number setting of the DIP switches SW1 and SW2.*
Token Cycle Time Setting	100 ms
MEMOBUS Response Watch Time	0 second
Station Search Interval	Every 10 time
Baud Rate	Depends on the setting of the BSR0 and BSR1 of SW1.*
Max. No. of Stations Connected	64
Message Relay Function	None
No. of Relayed Network 1	0 to 0
No. of Relayed Network 2	0 to 0
Maintenance Switch	Not set
Link Assignment	Without assignment

* If the station address setting of the rotary switches, baud rate setting of the BSR0 and BSR1 of DIP switch SW1, or network address setting of the DIP switches SW2 and 3 is out of the setting range, a switch setting error occurs and the 215AIF-01 Module will not start. Set the switch settings correctly and turn ON the power again.

■ RS-232C Interface (The setting of INIT switch to ON is valid while the power is being supplied.)

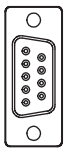
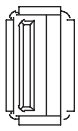
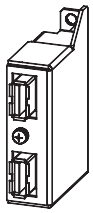
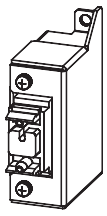
Item	Default Setting		
Communication Protocol	MEMOBUS		
Master/Slave	Slave		
Device Address	1		
Serial I/F	RS-232C		
Communication Mode	RTU		
Data Length	8 bits		
Parity	Even		
Stop Bits	1 bit		
Baud Rate	19.2 kbps		
Send Delay	Disable		
Receive Monitor Time	Disable		
Automatic Reception	Enable		
Interface Register Settings in the Slave		Leading Register	Number of Words
	Read Input Relay Status	IW0000	32768
	Read Input Register Status	IW0000	32768
	Read/Write Coil Status	MW00000	65535
	Read/Write Holding Register Status	MW00000	65535
	Coil/Holding Register Write Range	LO MW00000	
		HI MW65534	

9.2 Connection Specifications

9.2.1 Connector Specifications

The following table shows the specifications of 215AIF-01 Module connectors including externally mounted connectors.

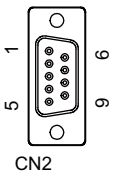
(1) Connector Specifications

Connector	Name	Connector Name	No. of Pins	Connector Model		
				Module	Cable	Manufacturer
	RS-232C	CN2	9	17LE-13090-27 (D2BC) 9-pin D-sub female connector	17JE-23090-02 (D8B) 9-pin D-sub male connector (Shell included.)	DDK Ltd.
	MPLINK/CP-215	CN1	4	DUSB-ARA81-T11	–	DDK Ltd.
	T-branch Connector for MPLINK Communication (JEPMC-OP2310)	(To be mounted on CN1 externally)	4	DUSB-ARA81-T11	DUSB-APA42-B1-C50	DDK Ltd.
	MR-connector Converter for CP-215 Communication (JEPMC-OP2320)	(To be mounted on CN1 externally)	8	MR-8RFA4 (G)	MR-8M (G) (Case: MR-8L)	Honda Tsushin Kogyo Co., Ltd.

(2) Connector Pin Arrangement

■ RS-232C Connector (CN2)

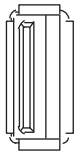
The CN2 connector is used to connect the MP2000 Series Machine Controller to computers and HMI devices via an RS-232C connection.



Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	FG	Frame ground	6	–	–
2	SD	Send data	7	SG	Signal ground (0V)
3	RD	Receive data	8	–	–
4	RS	Ready to send	9	–	–
5	CS	Clear to send	–	–	–

■ MPLINK Connectors

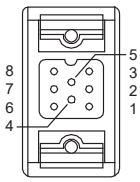
There are two MPLINK connectors on a T-branch connector. Both connectors have the same pin arrangement. For the last station, connect a MECHATROLINK cable to one connector, and a terminator to the other.



Pin Number	Signal Name	Description
1	(NC)	Not used.
2	/DATA	Signal –
3	DATA	Signal +
4	SH	Not used.
Shell	Shield	Connect the shield wire.

■ MR Conversion Connector for CP-215 Communication

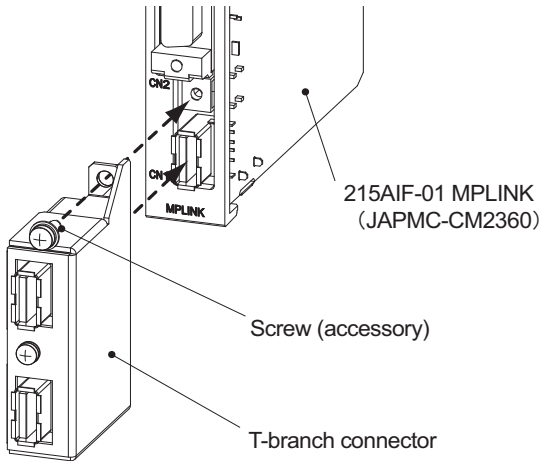
The MR conversion connector is used to connect the MP2000 Series Machine Controller to a CP-215 transmission system via CP-215 connection.



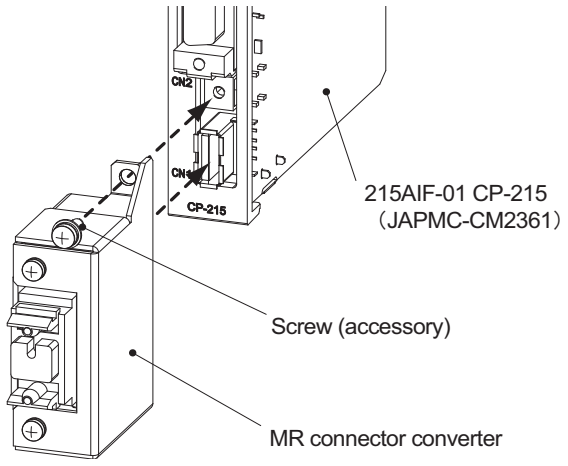
Pin Number	Signal Name	Description	Pin Number	Signal Name	Description
1	SRD–	Send/receive data (–)	5	N.C.	No connection
2	N.C.	No connection	6	N.C.	No connection
3	N.C.	No connection	7	N.C.	No connection
4	N.C.	No connection	8	SRD+	Send/receive data (+)

(3) Mounting a T-branch Connector/MR Connector Converter

Mount a T-branch connector or MR-connector converter on the MPLINK/CP-215 connector CN1 on the 215AIF-01 Module as shown in the diagram below.



Mounting a T-branch Connector
on JAPMC-CM2360



Mounting a MR-connector Converter
on JAPMC-CM2361

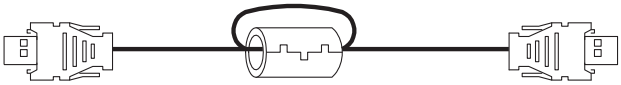
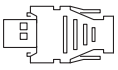
9.2.2 Cable Specifications

(1) RS-232C Cable (for MPE720 Engineering)

Model Number	Length	External Appearance (JEPMC-W5311-□□-E)
JEPMC-W5311-03-E	2.5 m	
JEPMC-W5311-15-E	15 m	

(2) MPLINK Cables

The following MECHATROLINK standard cables can be used for the 215AIF-01 Module with MPLINK specifications.

Name, Specifications, External Appearance	Model Number	Length
MECHATROLINK Cable MECHATROLINK connector-MECHATROLINK connector 	JEPMC-W6002-A5-E	0.5 m
	JEPMC-W6002-01-E	1 m
	JEPMC-W6002-03-E	3 m
	JEPMC-W6002-05-E	5 m
	JEPMC-W6002-10-E	10 m
	JEPMC-W6002-20-E	20 m
	JEPMC-W6002-30-E	30 m
	JEPMC-W6002-40-E	40 m
	JEPMC-W6002-50-E	50 m
MECHATROLINK Cable MECHATROLINK connector-MECHATROLINK connector with ferrite core 	JEPMC-W6003-A5-E	0.5 m
	JEPMC-W6003-01-E	1 m
	JEPMC-W6003-03-E	3 m
	JEPMC-W6003-05-E	5 m
	JEPMC-W6003-10-E	10 m
	JEPMC-W6003-20-E	20 m
	JEPMC-W6003-30-E	30 m
	JEPMC-W6003-40-E	40 m
	JEPMC-W6002-50-E	50 m
Terminator 	JEPMC-W6022-E	–

(3) CP-215 Cables

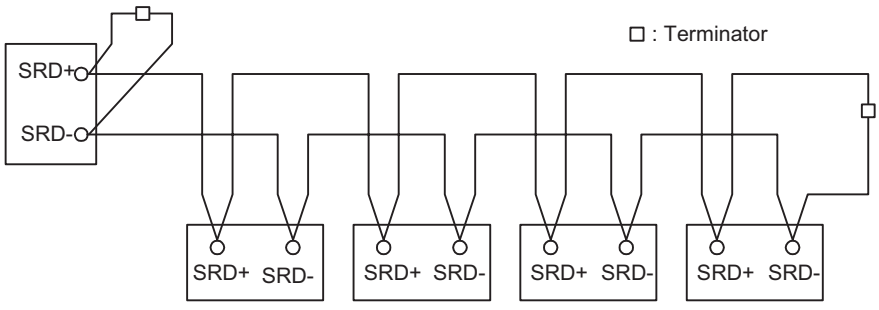
No standard cable is available for CP-215 transmission specifications. Refer to the following table and assemble the cable.

■ CP-215 Cable Components

Name	Model	Electrical Product Code	Specification	Application	Manufacturer
Twisted-pair Cable*1	YS-IPEV-SB *2 1P × 0.3 mm^2	–	Pas4*3: 60 dB/km Z4*4: 75 Ω	For in-panel low-voltage ducts	Fujikura Ltd.
	YS-IPEV-SB*2 3P × 0.3 mm^2	–	Pas4*3: 58 dB/km Z4*4: 75 Ω	For in-panel low-voltage ducts JC215-02 dedicated cable	Fujikura Ltd.
	YS-IPEV-S(Cu) *2 1P × 1.25 mm^2	–	Pas4*3: 23 dB/km Z4*4: 77 Ω	For panel-to-panel low-voltage ducts	Fujikura Ltd.
MR-8 Connector	MR-8LM(G)	–	8-pin, male, case included	For connection and branching of 215IF Module. One connector required for each Module.	HONDA TSUSHIN KOGYO CO., LTD.
Junction Box*5	JC215-01	87215-8100x	For cable size conversion	For conversion of in-panel or panel-to-panel cable size. Two junction boxes required for each control panel.	YASKAWA Electric Corporation
	JC215-02	87215-8200x	For connection of CP-215IF/AT Card Cable length: 3 m	For connection of ACGC4000 and MPE720 Programming Device	YASKAWA Electric Corporation
Terminator*6	EROS1PKF75R0	R002849	75 Ω ± 1%, 1/2W, 100PPm/°C	Mounted at both ends of transmission path. Two resistors required for each transmission path.	YASKAWA Electric Corporation
JC215-02 Dedicated Cable (in-panel cable with connectors)	–	87215-9900x	Twisted-pair cable with MR-8LF(G) and D-sub 9-pin connectors at both ends. Cable length: 3 m	For connection between ACGC4000, MPE720 Programming Device, and JC215-02	YASKAWA Electric Corporation

* 1. Specify the desired model and cable length (in 500-m units) when ordering.
* 2. Refer to the Manufacturing Specifications No. II-95J6015.
* 3. Pas4 refers to cable signal dB loss at 4 MHz.
* 4. Z4 refers to cable characteristic impedance at 4 MHz.
* 5. The Junction Box is used to connect the in-panel cables with the cables outside the panel. For details, refer to *Appendix G Junction Box Connection Example for CP-215 Communications* on page A-48.
* 6. Use junction terminal blocks to install terminating resistors.

■ CP-215 Connections



- Connect terminators on both ends of a transmission path.

■ Calculating the Transmission Distance

The panel-to-panel transmission distance for the CP-215 communications system depends on the baud rate, the number of stations connected, the number of JC215-01 and JC215-02 Junction Boxes connected, and the length of the transmission cables inside the control panels. In general, the maximum transmission distances of the cables between control panels are as follows:

- Maximum transmission distance with 4 Mbps = 520 - 4.5N - 3.0L_1 - 5.0M (m)
- Maximum transmission distance with 2 Mbps = 727 - 8.48N - 2.58L_1 - 6.06M (m)
- Maximum transmission distance with 1 Mbps = 1041 - 14.0N - 2.08L_1 - 8.33M (m)

  - N: Number of stations and Repeaters connected
    L1: Cable length inside the control panels (m)
    M: Number of JC215-01 and JC215-02 Junction Boxes connected.
    (The input side and output side of the JC215-01 may be counted together as one Junction Box.)

9.3 RS-232C Communication

Refer to *Chapter 3 Serial Communication* for information on the transmission definitions, connection examples, programming examples, the Message Send function, the Message Receive function, and other details on communication through the RS-232C port on the 215AIF-01 Module.

9.4 MPLINK Transmission Definitions

To perform MPLINK/CP-215 communication with the MP2000 Series Machine Controller, the MPLINK transmission definition file must be created.
This section describes how to define MPLINK transmissions using MPE720 windows.

9.4.1 MPLINK Transmission Configuration Window

To define the MPLINK transmissions, first the **Module Configuration** Window is opened on the MPE720, and then the **MPLINK Transmission Configuration** Window is opened.

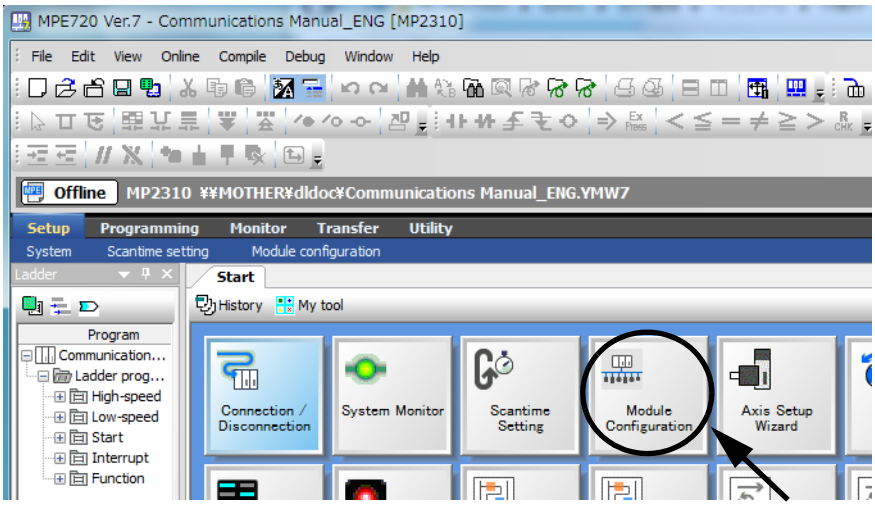
(1) Displaying the Module Configuration Window

Use the following procedure to display the **Module Configuration** Window.

1. Start the MPE720 on the personal computer connected to the Machine Controller and open the project file.

- For information on starting the MPE720, refer to *Machine Controller MP2000/MP3000 Series Engineering Tool MPE720 Version 7 User's Manual (Manual No.: SIEP C880761 03)*.

2. Click the **Module Configuration** Button on My Tool View.



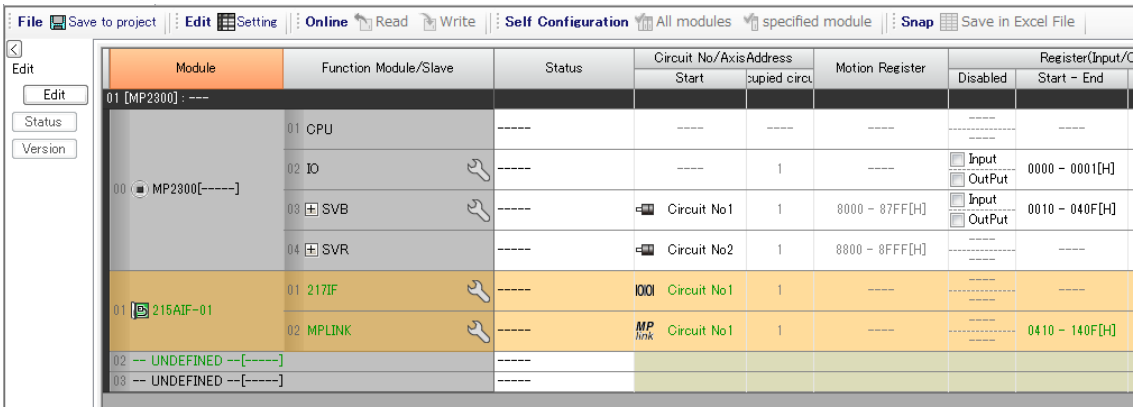
The **Module Configuration** Window will be displayed.

(2) Displaying the MPLINK Transmission Configuration Window from the Module Configuration Window

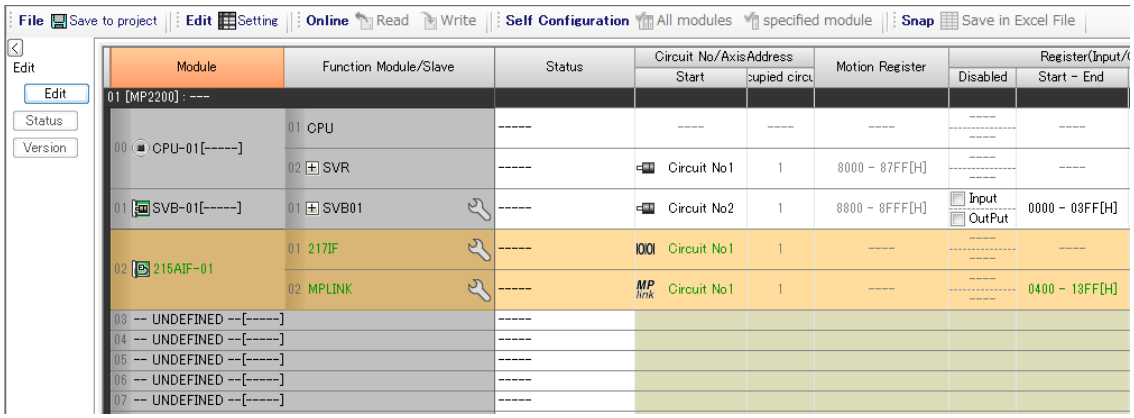
As the result of executing self-configuration, all Option Modules connected to the Machine Controller will be displayed in the **Controller** Area of the **Module Configuration** Window. (Refer to *1.5 Executing Self-configuration* on page 1-26.)

In the **Controller** Area, select the cell for 215AIF-01, and then double-click the slot number cell for the MPLINK in the **Module Details** Area. The **MPLINK Transmission Configuration** Window (see next page) will be displayed.

<MP2300/MP2300S/MP2310 Module Configuration Window>



<MP2100M/MP2200 Module Configuration Window>



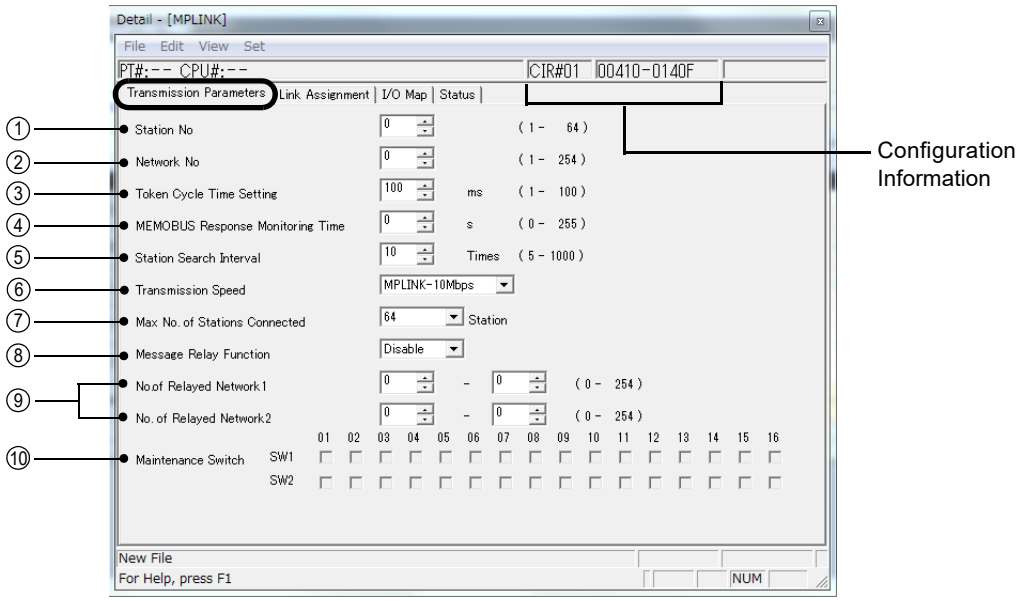
In Online Mode, the MPLINK transmission definitions data saved in the Machine Controller are displayed. In Offline Mode, the definitions data saved in the personal computer on which MPE720 is running is displayed.

The **MPLINK Transmission Configuration** Window is composed of four tab pages: **Transmission Parameters**, **Link Assignment**, **I/O Map**, and **Status**. These tab pages are used to set the definitions and monitor the settings.

- If the **Transmission Configuration** Window is being opened for the first time, a "new file" message box will be displayed and the **MPLINK Transmission Configuration** Window will be displayed when the **OK** Button is clicked.
- If the cell for the 217IF in the **Module Details** Area is double-clicked, the **217IF Transmission (serial communication) Configuration** Window will be displayed. Refer to *Chapter 3 Serial Communication* for details on serial communication.

9.4.2 Transmission Parameters Tab Page

Set the parameters required to use the MPLINK/CP-215 transmission system.



■ Configuration Information

Displays the 215AIF-01 Module configuration information. The configuration information is the same as the information displayed in the **Module Details** Area in the **Module Configuration** Window.

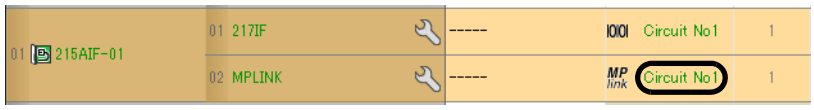
RACK#: The rack number of the rack in which the 215AIF-01 is defined.

Slot#: The slot number of the slot in which the 215AIF-01 is defined.

CIR#: The circuit number of the 215AIF-01.

Register number: The leading and ending I/O register numbers of the 215AIF-01.

- The configuration information can also be changed in the **Module Details** Area of the **Module Configuration** Window. Do not use a circuit number that is already being used by another 215AIF.



- When the MSG-SND or MSG-RCV function is used, set the circuit number displayed here for **CIR#** (circuit number).

■ Setting the Transmission Parameters

① Station No.

Set the local station number of the MPLINK/CP-215 in the range between 1 and 64.

② Network No.

Set the network number of the network in which the MPLINK/CP-215 is connected in the range between 1 and 254.

③ Token Cycle Time Setting

Set the target time from when a station receives a token* until it receives the next token in the range between 1 and 100 ms. The difference between the token cycle time set value and the token cycle time current value (displayed in the Status Tab Page) is the token holding time. During such time, link data or message data can be sent. If there is no data to be sent, the token will be sent to the next station before the token cycle time set value elapses.

If the token cycle time set value is shorter than the current value, only the link communication will be performed.

* Also, refer to *9.5.4 Token Passing* on page 9-32 for details on token passing.

④ MEMOBUS Response Monitoring Time

Set the time to wait for a response after sending a MEMOBUS command by executing the MSG-SND function.

0: The MSG-SND function will wait indefinitely for a response.

1 to 255: A timeout will occur after the time setting × 2 (s) and only one retry of sending the MEMOBUS command will be made.

⑤ Station Search Interval

Set the number of cycles between each search for a new station in the range between 5 and 1000. A station search will be performed every set number of times of token reception to determine whether a new station has been started.

⑥ Transmission Speed

Select the baud rate of MPLINK/CP-215 transmission from MPLINK-10Mbps, CP-215-2Mbps, and CP-215-4Mbps.

⑦ Max. No. of Stations Connected

Select the upper limit of number of stations for the station search among 16, 32, 48, and 64. Set a number bigger than the value set in **Station No**.

- The Max. No. of Stations Connected can be estimated by the total number of words for link communication, the value set in Token Cycle Time Setting, and Transmission Speed. Refer to *9.5.4 (4) Estimation of the Maximum Number of Stations Connected* on page 9-33 for details.

⑧ Message Relay Function

Set the message relay function, i.e., whether the 215AIF-01 will relay the message received from another transmission interface to another station through the MPLINK/CP-215 interface.

⑨ No. of Relayed Network 1 and No. of Relayed Network 2

When the **Message Relay Function** is enabled, set the range of network number that can be relay destinations (setting range: 1 to 254). The network number set in the **No. of Relayed Network 1** is ORed with the network number set in the **No. of Relayed Network 2**.

The **Message Relay Function** will be disabled if the settings of both **No. of Relayed Network 1** and **2** are 0.

- Refer to *9.5.5 (3) Examples of Network Configuration and Parameter Settings* on page 9-35 for information on network configuration and the relationship between the network configuration and the relayed network number.

⑩ Maintenance Switch SW1 and SW2

These switches are used for Yaskawa maintenance. Do not change the settings.

■ Reverting to Defaults

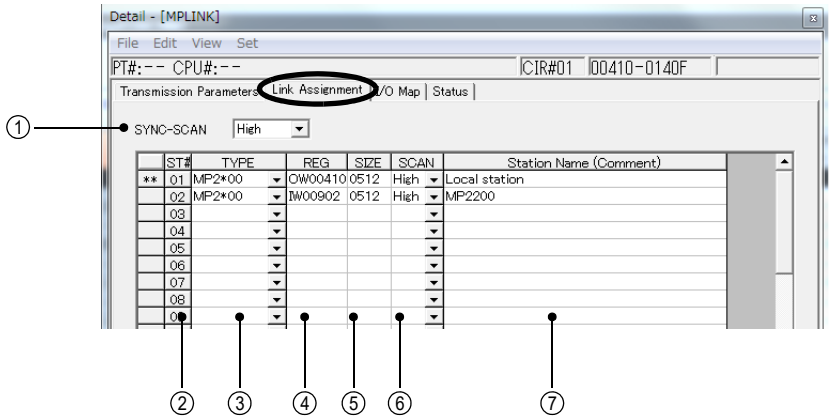
Select ***Edit - Default*** from the Main Menu in the **Module Configuration** Window. The parameters displayed on the **Transmission Parameters** Tab Page will be reset to their defaults as shown in the following table.

Parameter	Default Setting	Parameter	Default Setting
Station No.	0	Transmission Speed	MPLINK–10Mbps
Network No.	0	Max. No. of Stations Connected	64
Token Cycle Time Setting	100 (ms)	Message Relay Function	None
MEMOBUS Response Monitoring Time	0 (s)	No. of Relayed Network 1	0 to 0
Station Search Interval	10 (every cycles)	No. of Relayed Network 2	0 to 0

- Selecting ***Edit - Default*** will initialize the settings in the **Link Assignment** Tab Page as well.

9.4.3 Link Assignment Tab Page

Assign the I/O registers to be linked between the stations connected to the MPLINK/CP-215 network.



① SYNC-SCAN

Select the **High** or **Low** scan cycle as the cycle to which the MPLINK/CP-215 transmission is synchronized. The 215AIF-01 will be synchronized with the set scan cycle to process link data transmission to the transmission path.

- Data is actually sent to the transmission path after receiving a token.

② ST# (Station Number)

Set the station number. Displays the station numbers for the number set in Max. No. of Stations Connected in the Transmission Parameters Tab Page. Two asterisks (**) are displayed for the local station number.

③ TYPE

Select an I/O device name to be connected to each station.
For the MPLINK transmission, MP2*00 (MP2000 series controller) and MP3*00 (MP3000 series controller) can be selected.
For the CP-215 transmission (2 Mbps or 4 Mbps), the following device names can be selected.
CP-316, CP-916A, CP-3550, CP-9200SH, CP-317, CP-316H, CP-916G, ACGC4000, CP-517, CP-902, VS-580B, VS-590, EXCITER, 215IF/INV, 215IF/G5, 215IF/H5, RIO-05, RIO-2000, RIO-120, RIO2000D, RIO2000QD, CP316HQ, RIO2000Q, MP2*00(MP2000 series controller), MP3*00 (MP3000 series controller)

④ REG (Register Number)

Set the leading register number to be used for data input/output with each station. Set an O register number for the local station, and I register number for remote stations.

- Be sure to set different register numbers for each station.

⑤ SIZE

Set the number of I/O data in units of word. For a MPLINK transmission (10 Mbps), up to 1024 words can be set. For a CP-215 transmission (2 Mbps or 4 Mbps), up to 512 words can be set.

- For the local station to receive data, set the same size as for the sending station. If a size different from that of the sending station is set, data cannot be received.

⑥ SCAN

Select the scan in which to execute I/O services between the CPU Module (or Basic Module) and the 215AIF-01 Module. Either High or Low can be selected. In the selected scan, the local station outputs the send data and the remote station inputs the receive data.

- Select the same scan for the local station as for SYNC-SCAN. If the settings are different, consistency of output data will not be maintained within scans.

⑦ Station Name (Comment)

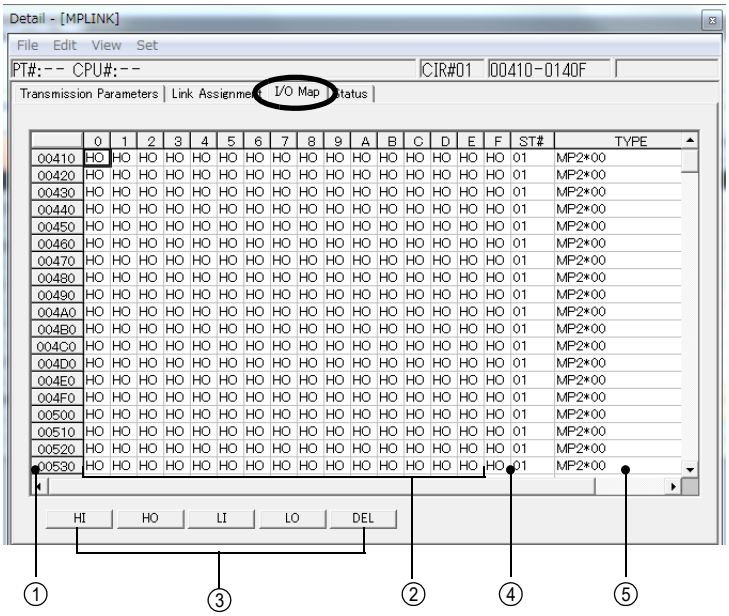
Comments up to 32 characters long can be entered for each station.

■ Deleting Assignment Data

Move the cursor to the row of the station to be deleted. Select ***Edit - Assignment Delete*** form the Main Menu. The assignment data of the selected row will be deleted.

9.4.4 I/O Map Tab Page

In the **I/O Map** Tab Page, the I/O assignment area from the leading I/O register number to the ending I/O register number that have been set in the MPLINK/CP-215 slot in the **Module Configuration** Window are displayed in units of word. Also, the **ST#** (station number) and **TYPE** (device name) that have been set in the **Link Assignment** Tab Page are displayed. Only the amount of scan information that have been set in the **SIZE** of the **Link Assignment** Tab Page are displayed in the I/O map assignment area. The remaining parts are left blank.



① Register No.

The register numbers from the leading I/O register number to the ending I/O register number that have been set for the 215AIF-01 Module in the **Module Configuration** Window are displayed.

② I/O Assignment Area (0 to F)

0 to F indicate the 1st digit of each register number. The area enclosed with the rows with register numbers and 0 to F columns is the I/O map assignment area. The I/O assignments set on the **Link Assignment** Tab Page will be displayed.

HI: Indicates a high-speed scan input.
HO: Indicates a high-speed scan output.
LI: Indicates a low-speed scan input.
LO: Indicates a low-speed scan output.

③ Assignment Setting Buttons

The Assignment Setting Buttons assign I/O to the registers in the I/O map area. In this operation, assignment is possible only for the registers that are set on the **Link Assignment** Tab Page. Assignments for output settings are not allowed for the registers assigned as the input registers in the I/O assignments. Similarly, it is not possible to assign the input settings to registers assigned for output registers in the I/O assignments.

The function of each button is given below.

HI: Sets a high-speed scan input.
HO: Sets a high-speed scan output.
LI: Sets a low-speed scan input.
LO: Sets a low-speed scan output.
DEL: Deletes the assignment.

④ ST# (Station Number)

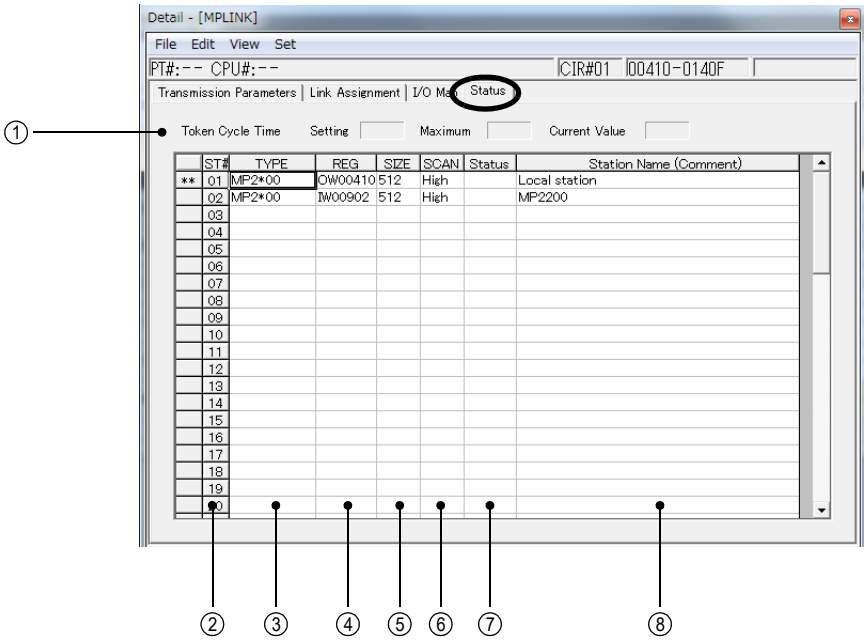
Displays the station numbers (ST#) that have been set in the **Link Assignment** Tab Page.

⑤ TYPE

Displays the I/O device names (**TYPE**) that have been set in the **Link Assignment** Tab Page.

9.4.5 Status Tab Page

The current status of MPLINK/CP-215 link communication is displayed. The settings cannot be changed.



① Token Cycle Time

The token cycle time value that has been set in the **Transmission Parameters** Tab Page, the maximum value, and current value of the actual token cycle time are displayed in units of ms.

- The values in **Maximum** and **Current Value** are displayed with the first decimal point rounded off (values under 0.5 ms are displayed as 0 ms).
- The token cycle time largely differs depending on the timing of power ON and connection to remote stations. To reset the value in **Maximum**, execute the save operation without changing the settings in the **Transmission Parameters** or **Link Assignment** Tab Page.

② ST# (Station Number)

The station number is displayed. Displays the station numbers for the number set in Max. No. of Stations Connected in the **Transmission Parameters** Tab Page. Two asterisks (**) are displayed for the local station number.

③ TYPE

Displays the I/O device names that have been set for **TYPE** in the **Link Assignment** Tab Page.

④ REG (Register Number)

Displays the register numbers that have been set in the **Link Assignment** Tab Page.

⑤ SIZE

Displays the number of I/O data that has been set for **SIZE** in the **Link Assignment** Tab Page.

⑥ SCAN

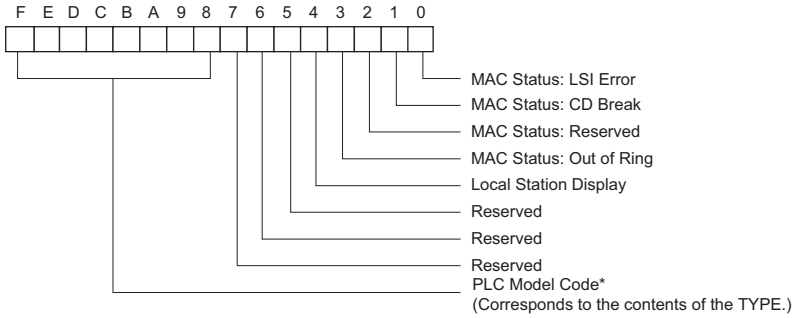
Displays the transmission processing scan that has been set for **SCAN** in the **Link Assignment** Tab Page.

⑦ Status

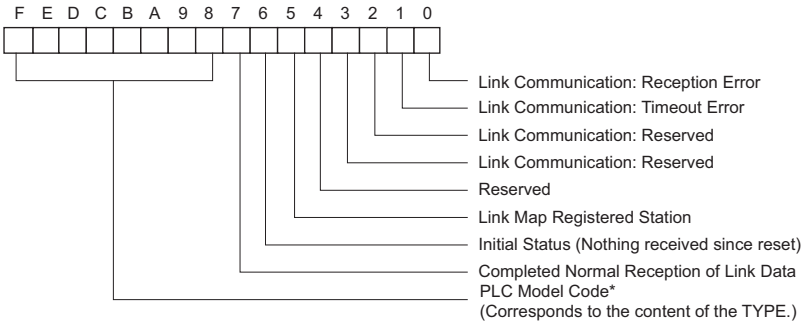
In Online Mode, the MPLINK/CP-215 link status data is displayed in hexadecimal.

- Nothing will be displayed in Offline Mode.

<Link Status Data of Local Station (Normal status: xx10H)>



<Link Status Data for Remote Stations (Normal status: xxA0H)>



* The following table shows the device name for each PLC Model Code.

Model Code	Device Name	Model Code	Device Name	Model Code	Device Name	Model Code	Device Name
05H	CP-316	11H	ACGC4000	62H	215IF/INV	CCH	RIO2000QD
06H	CP-916A	12H	CP-517	64H	215IF/G5	1AH	CP316HQ
0BH	CP-3550	14H	MP920	65H	215IF/H5	C9H	RIO2000Q
0CH	CP-9200SH	16H	CP-902	C2H	RIO-05	D0H	MP2∗00
0DH	CP-317	1CH	VS-580B	C5H	RIO-2000		
0EH	CP-316H	1DH	VS-590	C6H	RIO-120		
0FH	CP-916G	1FH	EXCITER	CBH	RIO2000D		

⑧ Station Name (Comment)

Displays the station name that has been set in the **Link Assignment** Tab Page.

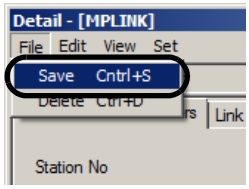
9.4.6 Saving and Deleting MPLINK Transmission Definitions

- When changing, saving, or deleting MPLINK transmission definitions data, save the data to flash memory in the Machine Controller, and then turn the power to the Machine Controller OFF and then turn it ON again.

(1) Saving the MPLINK Transmission Definitions Data

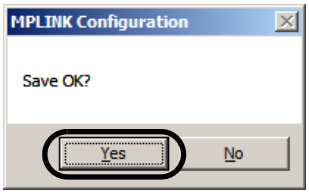
Use the following procedure to save the MPLINK transmission definitions data if it has been set or changed. In Online Mode, the definitions data is saved in the Machine Controller and in personal computer running the MPE720. In Offline Mode, the definitions data is saved in the personal computer running the MPE720.

1. Select ***File - Save*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

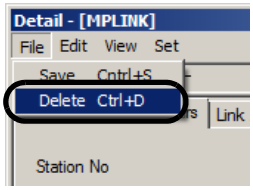
2. Click the **Yes** Button.



(2) Deleting the MPLINK Transmission Definitions Data

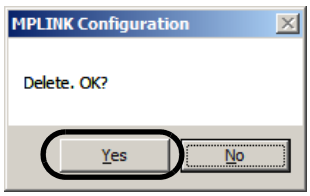
Use the following procedure to delete all of the MPLINK transmission definitions data. In Online Mode, the definitions data is deleted from the MP2000 Series Machine Controller and the personal computer running the MPE720. In Offline Mode, the definitions data is deleted from the personal computer running the MPE720.

1. Select ***File - Delete*** from the Main Menu in the **Module Configuration** Window.



A confirmation message box will be displayed.

2. Click the **Yes** Button.



9.5 Transmission Methods in MPLINK/CP-215 Communication

There are three transmission methods in the MPLINK/CP-215 communication using the 215AIF-01 Module: link communication, engineering communication, and message communication.

- Link communication is used to send and receive I/O data such as relays, coils, and registers between Machine Controllers, and operates only between 215AIF-01 Modules.
- Engineering communication is used to communicate with MPE720 via MPLINK/CP-215 connection.
- Message communication is used to send and receive messages between Machine Controllers one event at a time using the message functions (MSG-SND and MSG-RCV).

This section describes the details on these three transmission methods.

9.5.1 Link Communication

Simple assignment operation in the **Link Assignment** Tab Page in the **MPLINK Transmission Configuration** Window allows data to be sent or received between stations connected to a MPLINK or CP-215 transmission path. Machine Controllers send and receive data at regular intervals by a transmission mechanism called token passing according to the link assignment map that has been set in advance.

- Also, refer to *9.5.4 Token Passing* on page 9-32 for details on token passing.

(1) Link Assignment

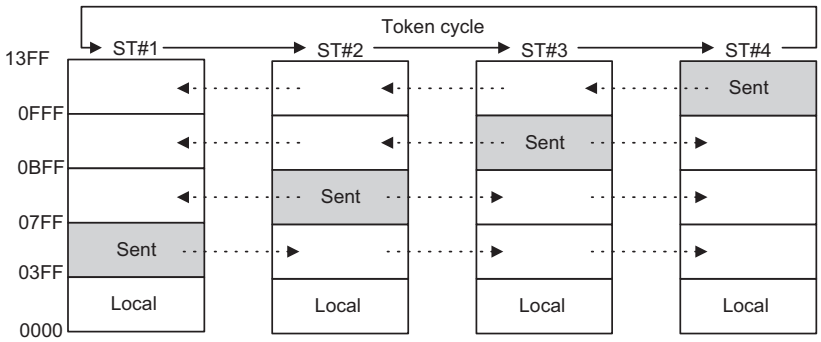
The MP2000 Series Machine Controller has the following I/O register area.

Input Registers (IB, IW, IL, LFhhhh)	32767 words (0000 to 7FFF)
Output Registers (OB, OW OL, OFhhhh)	32767 words (0000 to 7FFF)

Settings are made to map registers in the above I/O register areas between the Machine Controllers to show how data is to be transferred. This is called Link Assignment. Refer to *9.4.3 Link Assignment Tab Page* on page 9-20 for details on link communication.

(2) Conceptual Diagram of Link Communication

If we assume, for example, that I/O registers 0000 to 03FF are in a local area for each station and that registers 03ff to 13FF are used for shared areas, then the link communication would conceptually appear as shown below.



The I/O registers other than the sent registers at each local station will be the receive areas from remote stations.

(3) Transmission Overview

When each station has written send data to its own send area and has received the token, the send area information and data will be sent to all stations. When this data is received, all stations will write the receive data to their own I/O registers, according to the area information. In this way, the contents of the I/O areas of all stations are periodically updated, according to the token cycle.

The following block diagram illustrates the link data updating process between a Machine Controller CPU and the 215AIF-01 Module when the settings are made in the Link Assignment Tab Page.

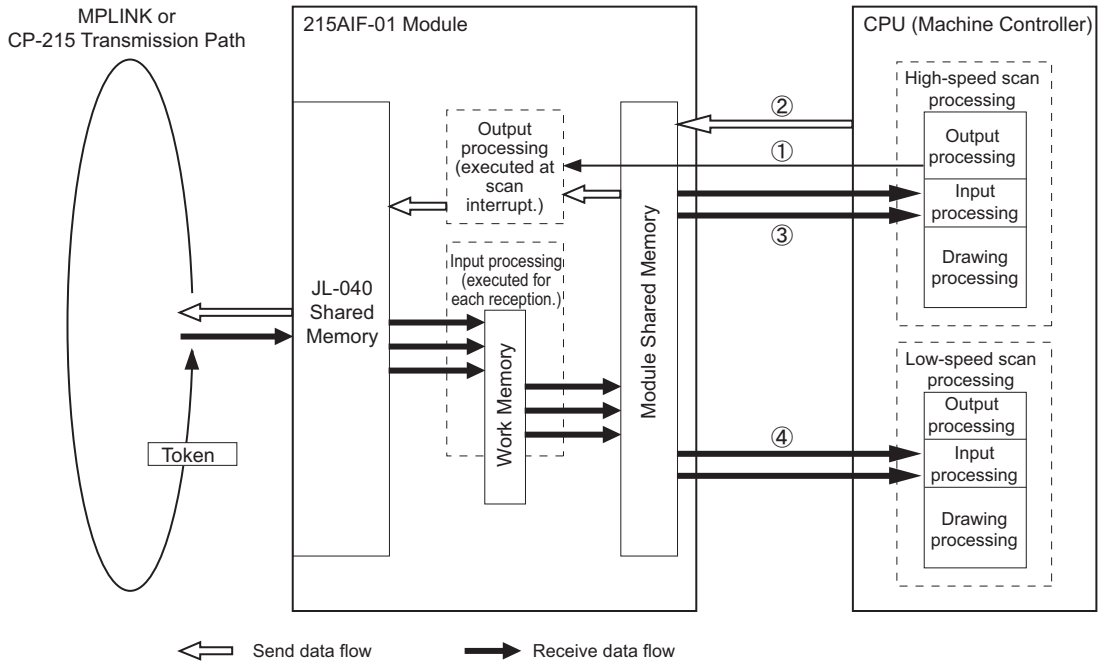


Fig. 9.1 Block Diagram for Link Data Updating Process Between CPU and 215AIF-01

① : An interrupt is sent to the 215AIF-01 Module in the scan cycle set in **SYNC-SCAN** on the **Link Assignment** Tab Page. The 215AIF-01 Module will execute the output processing at the reception of the interrupt.

② : The 215AIF-01 will output the data in the scan cycle set in **SCAN** of the local station (marked with **). If the setting in **SCAN** is the same as that in **SYNC-SCAN**, an interrupt will be sent after outputting the data.

③ : The 215AIF-01 will input data in the high-speed scan cycle set in **SCAN** of the remote stations.

④ : The 215AIF-01 will input data in the low-speed scan cycle set in **SCAN** of the remote stations.

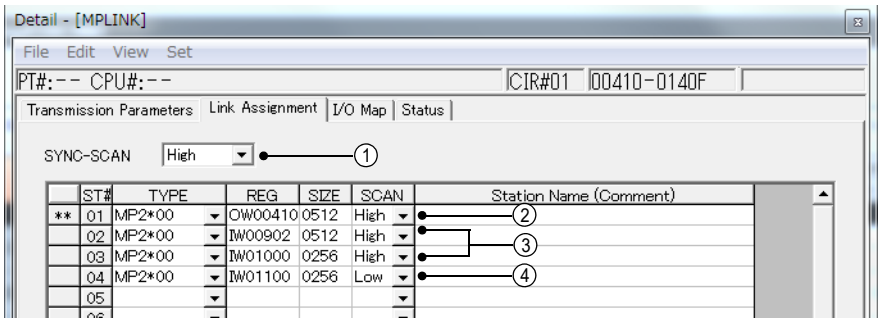
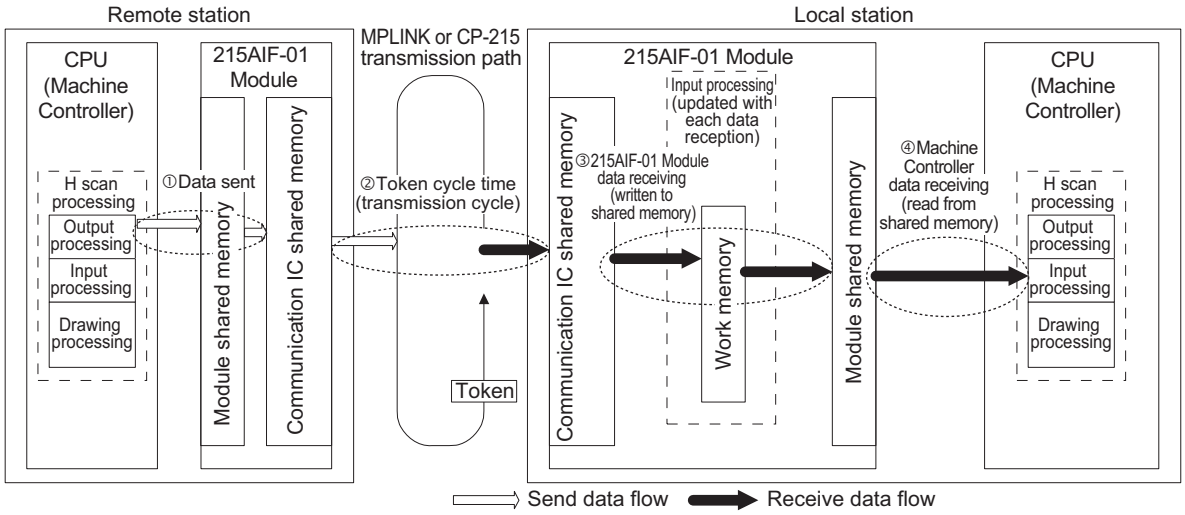


Fig. 9.2 **Link Assignment** Tab Page in the **MPLINK Transmission Configuration** Window

(4) Precautions on Updating Data

Updating the data may be delayed at the Machine Controller depending on the timing of the scan cycle and token cycle time for link assignment in CP-215 transmissions and MPLINK transmissions using the 215AIF-01 Module.
The following figure shows the relations for data send or receive timing between stations in CP-215 transmissions and MPLINK transmissions.

■ Timing Relations for Sending and Receiving Data



① Data Sending

The timing at which the Machine Controller of the remote stations updates send data to the 215AIF-01 Module depends on the scan that is assigned to the local station in the **SCAN** setting on the **Link Assignment** Tab Page. The figure above is an example of assignment to H scans. Data is updated during the output processing in H scan processing. Updating is performed at the same time up to the communication IC shared memory before the transmission path.

② Token Cycle Time

Data is sent on the transmission path when the token is received after data is updated in communication IC shared memory of the 215AIF-01 Module.

- The token (the right to send data) is passed between physically connected 215AIF-01 Modules in order starting from the lowest station number. If there is no data to send to the stations, only the token is passed on the transmission path. Therefore, the token cycle time is almost constant for a short cycle. If there is data to send to the remote stations, link data flows on the transmission path in addition to the token. There is variation in the token cycle time because the data size and update timing on the transmission path depend on the link assignment size and link assignment scan settings.

③ 215AIF-01 Module Data Reception

The link data is written via the work memory to the module shared memory one station at a time when the 215AIF-01 Module receives link data from the transmission path.

- The module shared memory uses a single buffer. Therefore, if data is received by the 215AIF-01 Module while the Machine Controller is reading data, the received data is not written to the module shared memory and it remains in work memory until the Machine Controller completes reading data. Thus, using module shared memory causes access competition between the 215AIF-01 Module and Machine Controller.

④ Machine Controller Data Reception

The timing at which the Machine Controller of the local station updates receive data from the 215AIF-01 Module depends on the scan that is assigned to the remote stations in the **SCAN** setting on the **Link Assignment** Tab Page. The figure above is an example of assignment to H scans. Data is input during the input processing in H scan processing.

- The module shared memory uses a single buffer. Therefore, the Machine Controller cannot read data if the 215AIF-01 Module is writing receive data. In that case, the Machine Controller attempts to obtain the access right twice for reading. If the access right cannot be obtained, the Machine Controller will stop updating data to minimize the scan time extension. (The scan time will be extended slightly, however, due to processing to obtain the access right.)

■ Delay in Updating Receive Data at the Machine Controller

A delay in updating receive data by the Machine Controller occurs due to access competition between the 215AIF-01 Module and Machine Controller in the module shared memory of the 215AIF-01 Module. Writing data received by the 215AIF-01 Module and reading data received by the Machine Controller are performed asynchronously, but both operations may be performed synchronously due to the link assignment settings or scan time settings. In that case, continuous access competition will occur and cause a delay in updating data.
The following is a list of possible causes and corrective actions.

Status	Cause	Corrective Action
The access time to the module shared memory is extended.	Data has been set with an assignment size larger than required.	Delete unnecessary data to reduce the assignment size.*1
	The 215AIF-01 Module is assigned to an Expansion Rack.	Assign the 215AIF-01 Module to the first Rack if that Rack has sufficient capacity.*2
The module shared memory is accessed at the same time from the 215AIF-01 Module and Machine Controller.	The output timing cycle for remote stations and the input timing cycle for the local station are the same.	Adjust the scan settings or scan time settings for the output timing of remote stations and the input timing of the local station so that the timings do not overlap.*3

* 1. The larger the data size, the easier it is for competition to occur due to the longer time to access the module shared memory.
* 2. Accessing the module shared memory from a second or higher Expansion Rack is slower than accessing it from the first Expansion Rack.
* 3. The greater the lowest common multiple scan set value of remote stations and scan set value of the local station, the more effective the scan time set value is. Let A be the scan setting for the data cycle of remote stations set for **SCAN** in the **Link Assignment** Tab Page, and let B be the scan setting for the data receive cycle of the local station. In the following case, accessing the module shared memory is synchronous, and competition occurs easily as shown by the flow of updating link data in the figure on the previous page.

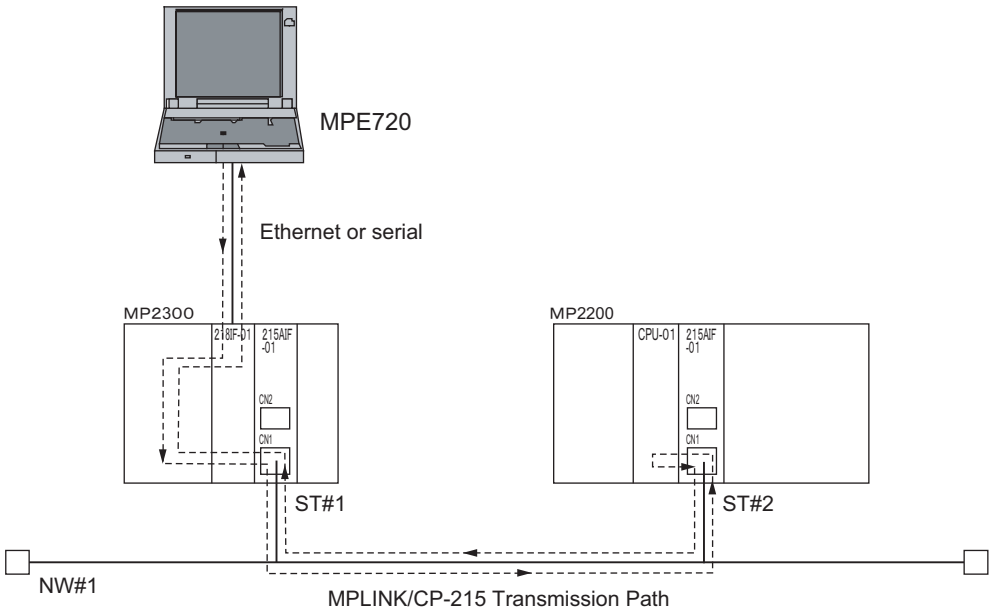
  A and B are the same and the scan time settings for A and B are the same, or the scan time setting for A is a multiple of the scan time setting for B, or the scan time setting for B is a multiple of the scan time setting for A.

9.5.2 Engineering communication

Engineering communication is used to connect a Machine Controller to the MPE720 via MPLINK/CP-215 connection. With engineering communication, the MPLINK/CP-215 transmission channel can operate not only as the Master channel[*1] but also as a Slave channel[*2]. Therefore, it can relay[*3] engineering messages between remote station Machine Controllers.

* 1. The channel that can send engineering messages and receive the responses.
* 2. The channel that can receive engineering messages and send the responses.
* 3. The settings for Relay Function is required to relay messages. Refer to *9.5.5 Relay Function* on page 9-34 for details.

The following diagram illustrates the message flow in engineering communication.



9.5.3 Message Communication

Message communication using MPLINK/CP-215 transmissions support Extended MEMOBUS and non-procedure protocols.

(1) Overview

A message communication using either the Extended MEMOBUS or non-procedure protocol can be performed between MPLINK or CP-215 transmission devices connected to a single network. The Extended MEMOBUS and non-procedure protocol transmission is performed between a Master and Slaves. The Master station sends messages to Slaves and the Slaves that receives messages return the response messages. The automatic reception function is not provided.

Twelve channels[*1] are provided as the message channels for Extended MEMOBUS and non-procedure protocol transmissions, so that 12 message communication can be performed at a time. Master or Slave can be set to each channel[*2].

* 1. A total of 12 channels for Extended MEMOBUS and non-procedure protocol communication.
* 2. The channel that uses MSG-SND function is the Master, and the channel that uses MSG-RCV function is a Slave.

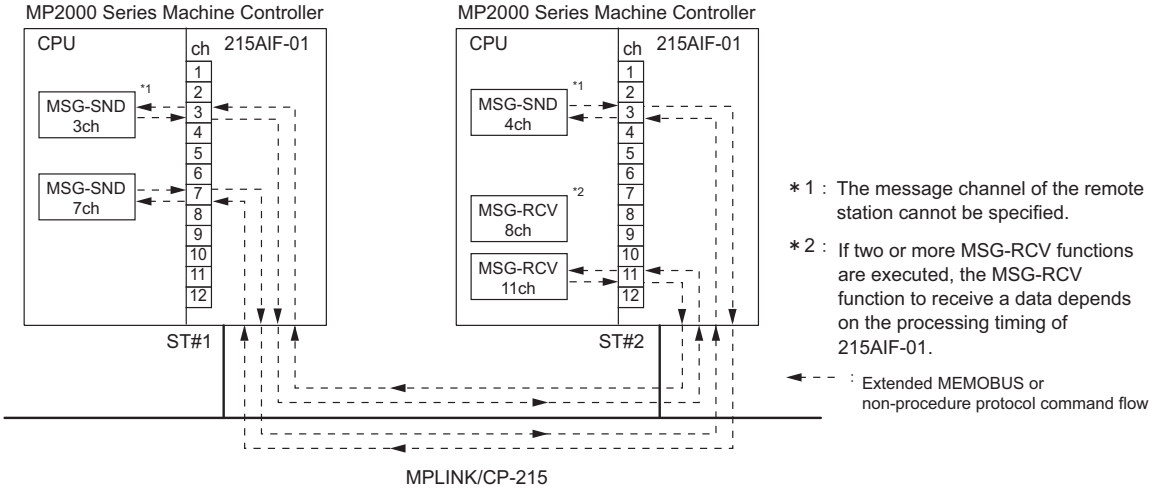


Fig. 9.3 Message Flow Using the Extended MEMOBUS Protocol

(2) Extended MEMOBUS Commands

The Extended MEMOBUS commands that can be used are as follows:

Function Code	Function	Data Address Setting Range	Data Size
01H	Read Coil Status	0 to 65535	1 to 2000 bits
02H	Read Input Relay Status	0 to 65535	1 to 2000 bits
03H	Read Holding Register Contents	0 to 65534	1 to 125 words
04H	Read Input Register Contents	0 to 32767	1 to 125 words
05H	Modify Status of a Single Coil	0 to 65535	Invalid
06H	Write to a Single Holding Register	0 to 65534	Invalid
08H	Loopback Test	Invalid	Invalid
09H	Read Holding Register Contents (Extended)	0 to 65534	1 to 508 words
0AH	Read Input Register Contents (Extended)	0 to 32767	1 to 508 words
0BH	Write to Holding Register (Extended)	0 to 65534	1 to 507 words
0DH	Read Discontinuous Holding Register Contents (Extended)	0 to 65534	1 to 508 words
0EH	Write to Discontinuous Holding Registers (Extended)	0 to 65534	1 to 254 words
0FH	Modify Status of Multiple Coils	0 to 65535	1 to 800 bits
10H	Write to Multiple Holding Registers	0 to 65534	1 to 100 words

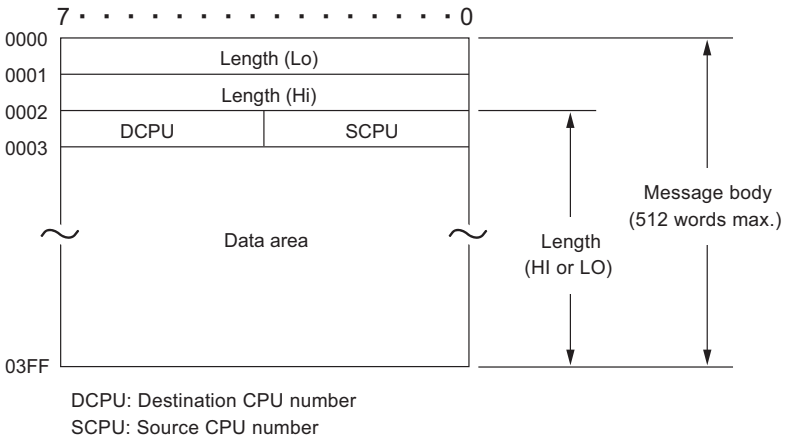
- Maximum length of an Extended MEMOBUS command (MEMOBUS header + data size) is 512 words.
- The message format for transmissions is the same as for the 218IF transmissions.

(3) Message Size and Format Using Non-procedure Protocol

The message sizes that can be used are as follows.

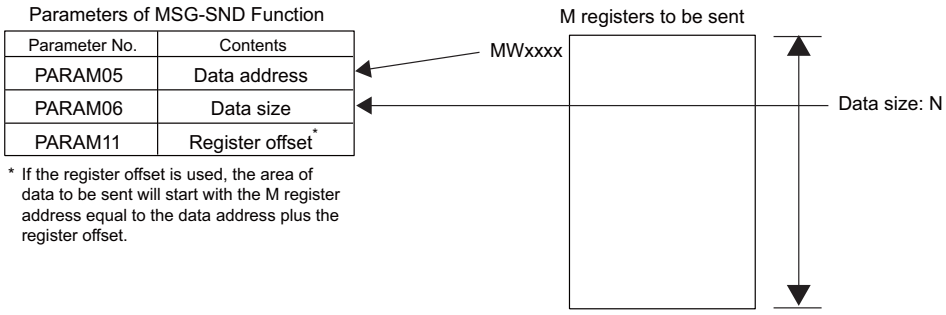
	Data Size
Non-procedure 1	1 to 510 words
Non-procedure 2	1 to 1020 bytes

The data format for message communication using non-protocol is as follows.



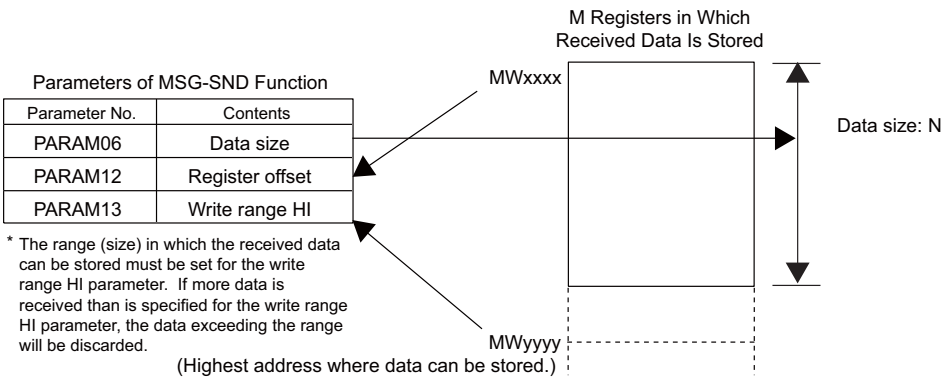
- Communication may stop or data may be corrupted if the quantity of messages received exceeds the processing capacity for message reception. Particularly when a non-procedure protocol is used to send data one way, be sure that the send processing cycle is not shorter than the reception processing cycle.

With the non-procedure protocol, the address of the first M register and the data size are specified in the MSG-SND function to enable sending the contents of M registers.



- Set the remote station number in PARAM02 of the MSG-SND function and set the remote CPU number in PARAM07.

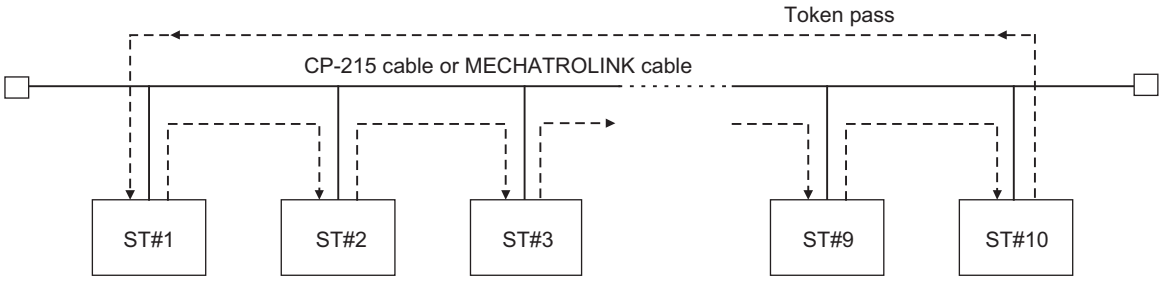
With the MSG-RCV function, the register offset and write range HI can be set to store the received data in the specified M registers.



9.5.4 Token Passing

(1) Token Passing

MPLINK/CP-215 communication uses the token passing as the transmission access method. In the token passing, data of a specific pattern, called a token, circulates continuously on the transmission path in order of station numbers*. The station that has this token (the right to send data) can send data.
The following diagram illustrates the token passing.

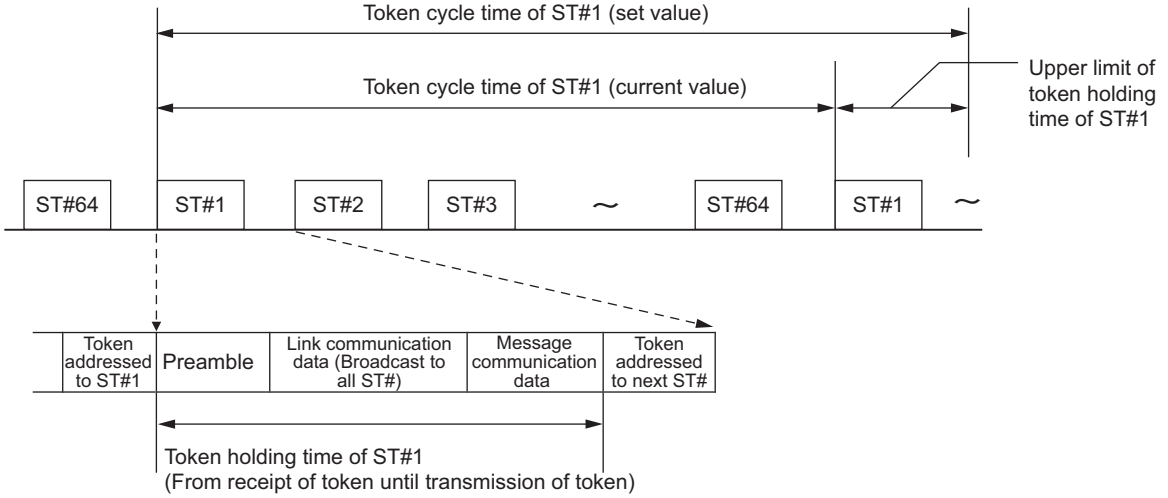


Each time a station with send data obtains the token, it transmits the data within the token holding time (see the section below). A station with no send data immediately passes the token to the next station.

* The stations need not be physically installed in numerical order.

(2) Token Cycle Time and Holding Time

The time from when a token is received until the next time the token is received is called the Token Cycle Time. The time from when the token is received until the token is transmitted is called the Token Holding Time. The following diagram illustrates the relationship between the Token Cycle Time and the Token Holding Time.



The MPLINK/CP-215 transmission operates with the upper limit of the token holding time being the difference in time between the token cycle time (set value) set in the transmission parameter and the token cycle time (current value) measured at each receipt of the token.
Link communication data is transmitted even when this upper limit is exceeded*, while message communication data is not transmitted but is held if there is any possibility that the upper limit may be exceeded.

* Link communication data can be sent even if the token cycle time (set value) is shorter than the token cycle time (current value).

(3) Estimation of the Token Cycle Time

The token cycle time at each station can be estimated using the equations given below.
In each equation, the message communication margin refers to the time margin for message communication from a station included in the token cycle time. A minimum message communication time margin of 1.2 ms for 4 Mbps or 2.4 ms for 2 Mbps must be guaranteed.
If the message communication time margin is increased, the message communication efficiency will increase, but link communication data exchanges will become slower.

- Baud rate: 10 Mbps
  Token cycle time (ms)
  = 0.10 × Number of stations + 0.0016 × Total number of link communication words + Message communication time margin (0.48 ms min.)
- Baud rate: 4 Mbps
  Token cycle time (ms)
  = 0.16 × Number of stations + 0.004 × Total number of link communication words + Message communication time margin (1.2 ms min.)
- Baud rate: 2 Mbps
  Token cycle time (ms)
  = 0.23 × Number of stations + 0.008 × Total number of link communication words + Message communication time margin (2.4 ms min.)

- The token cycle time (set value) does not guarantee data transmission in any specific cycle or a fixed cycle.
- Do not set the token cycle time (set value) to a value less than the value calculated using the above equations. Engineering from the MPE720 may no longer be possible.
- Each parameter in the above equations depends on the type and software version number of the transmission interface board, the system configuration, and other factors in the system. The values should be used only as a guide for system design.

(4) Estimation of the Maximum Number of Stations Connected

By modifying the above equations, the maximum number of stations that can be connected can be estimated for the token cycle time (set value).
The results are shown in the following table.

Number of Link Communication Words	Token Cycle Time (ms)	Baud Rate		
		10 Mbps	4 Mbps	2 Mbps
1024	3	8 stations	–	–
	5	28 stations	–	–
	8	32 stations	16 stations	–
	10	32 stations	30 stations	–
	20	32 stations	64 stations	40 stations
	30	32 stations	64 stations	64 stations
	50	32 stations	64 stations	64 stations
2048	5	12 stations	–	–
	10	32 stations	–	–
	20	32 stations	64 stations	5 stations
	30	32 stations	64 stations	48 stations
	50	32 stations	64 stations	64 stations
4096	5	–	–	–
	10	29 stations	–	–
	20	32 stations	–	–
	30	32 stations	–	–
	50	32 stations	–	–
	100	32 stations	–	–

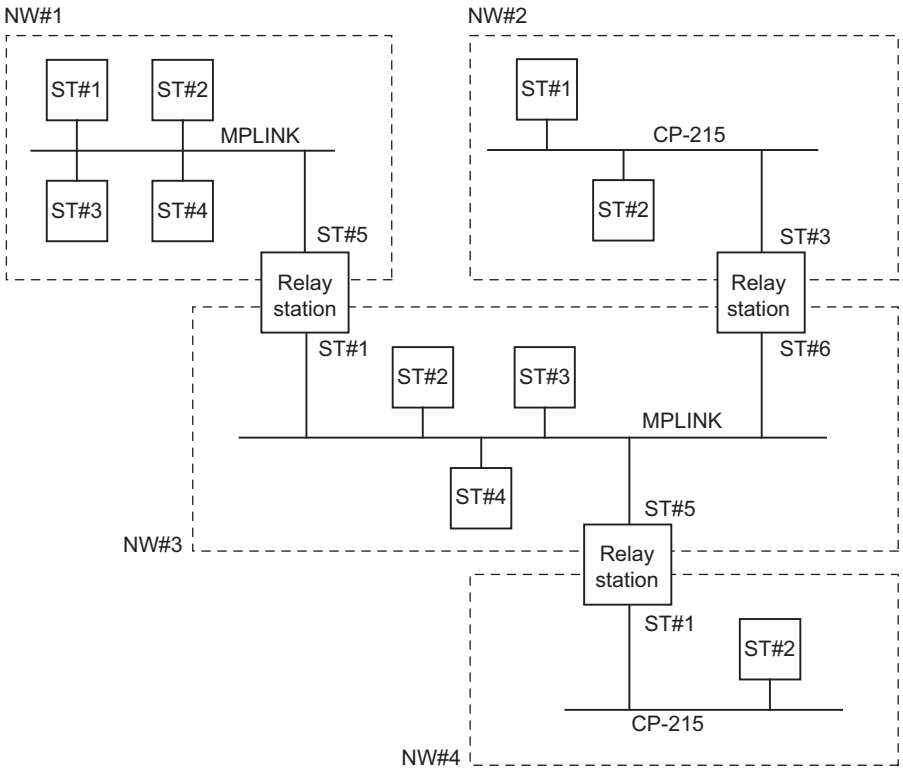
- The number of stations connected is restricted not only by the transmission performance, but also by electrical conditions, such as the number of stations connected and the transmission distance.

9.5.5 Relay Function

The 215AIF-01 Module is provided with the Relay Function that allows message communication among multiple 215AIF-01 networks.

(1) Network Relay Function

Using a relay Machine Controller with two or more 215AIF-01 Modules mounted, the MP2000 Series Machine Controller can transmit data with control devices connected to different networks and manage and control the entire system using the MPE720. A unique network number (NW#nn) is assigned to each network segment in the system and a unique station number (ST#nn) is also assigned to the 215AIF-01 Module in each network segment.
All the units connected to the entire network can be uniquely identified or specified by identifying the combinations of these network numbers (NW#nn) and station numbers (ST#nn) as network addresses.
The Relay Function facilitates message communication* between such networks.
The following diagram illustrates an example of network configuration that uses the Relay Function.

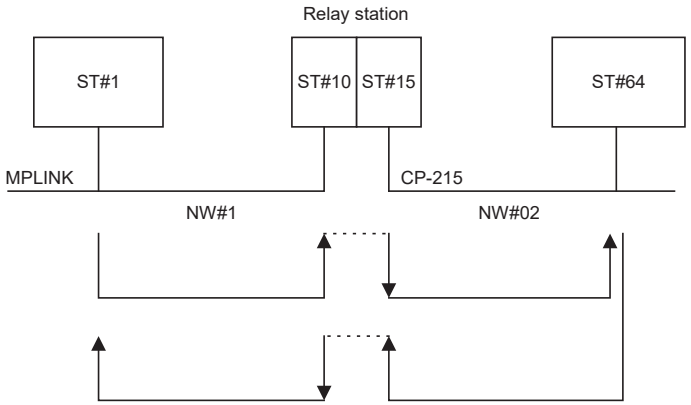


- Relay station in the above diagram indicates a MP2000 Series Machine Controller with two or more 215AIF-01 Modules, and the MPLINK connector or MR connector mounted on each of these Modules is connected to a different network with MPLINK/CP-215 cable to transfer data between the networks.
- To use the Relay Function, set the parameter **Message Relay Function** to **Enable** and specify the relay destination network number (**No. of Relayed Network**) in the **Transmission Parameters** Tab Page of the **215AIF-01 Module Configuration** Window. Refer to *9.4.2 Transmission Parameters Tab Page* on page 9-18 for details.

* The Relay Function can be used only for engineering message communication.

(2) Relay Processing Flow

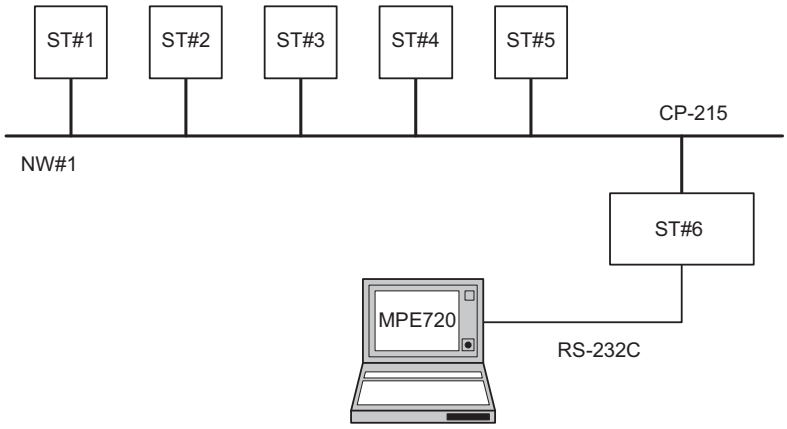
When sending a message, the relay destination address is automatically assigned by the system, and the message is sent to the relay destination via the relay station. The response message is also sent to the station via the relay station.
The following diagram shows the processing flow to send a massage from the ST#1 to ST#64 via the relay station (ST#10/ST#15), and ST#64 returns a response to ST#1 via the relay station.



(3) Examples of Network Configuration and Parameter Settings

The network system using the Relay Function basically consists of a MPE720 that is connected to a station and a system configuration to relay data over MPLINK/CP215 lines.
This section describes how to set the transmission definitions parameters to control the networks for two network configuration examples: with one network, and connection to three networks via a Machine Controller with three 215AIF-01 Modules mounted (relay station).

[a] Network Configuration 1: One Network



■ Setting the Transmission Definition Parameters (Set on the Transmission Parameters Tab Page)

The following 215AIF-01 Relay Function settings are used for all the stations ST#1 to ST#6.

- Message Relay Function: Enable
- No. of Relayed Network 1: 1 - 1
- No. of Relayed Network 2: Not set

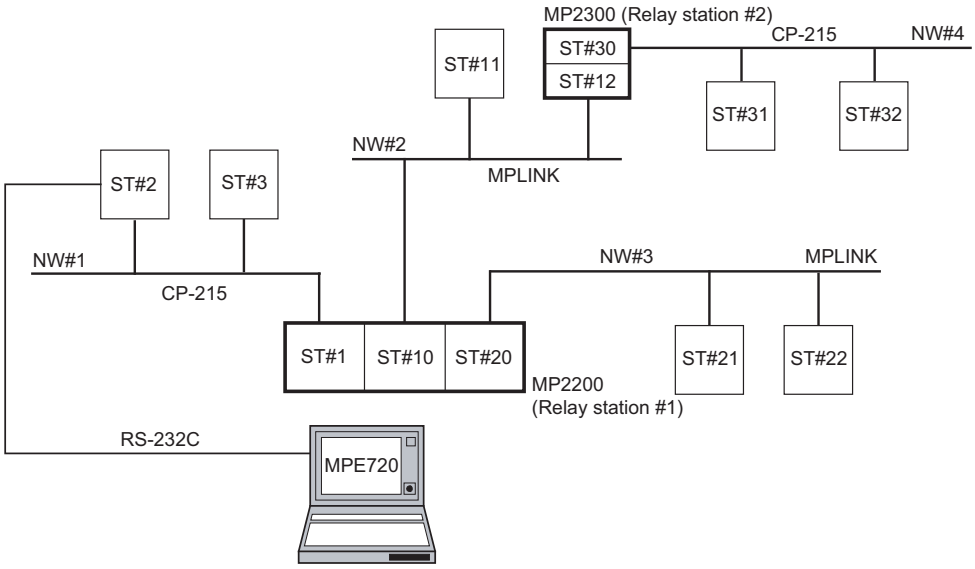
With the above settings, all the Machine Controllers at ST#1 to ST#6 can be accessed from the MPE720.

■ Remarks

- The Relay Function must be set in order to access a Machine Controller (ST#1 to ST#5 in the above diagram) that is not directly connected to the MPE720 via a MPLINK/CP-215.
- With a single network configuration, set the Relay Function for all stations. Set the local network number as the relayed network number.

[b] Network Configuration 2: Connection to Three Networks via One Relay Station

In the following example, three 215AIF-01 Modules are mounted on one MP2200.



■ Setting the Transmission Definition Parameters (Set on the Transmission Parameters Tab Page)

1. NW#1 ST#1

- Message Relay Function: Enable
- No. of Relayed Network 1: 1 - 1
- No. of Relayed Network 2: Not set

2. NW#2 ST#10

- Message Relay Function: Enable
- No. of Relayed Network 1: 2 - 2
- No. of Relayed Network 2: 4 - 4

3. NW#3 ST#20

- Message Relay Function: Enable
- No. of Relayed Network 1: 3 - 3
- No. of Relayed Network 2: Not set

4. NW#2 ST#12

- Message Relay Function: Enable
- No. of Relayed Network 1: 1 - 3
- No. of Relayed Network 2: Not set

5. NW#4 ST#30

- Message Relay Function: Enable
- No. of Relayed Network 1: 4 - 4
- No. of Relayed Network 2: Not set

6. NW#1 ST#2 to ST#3
NW#2 ST#11
NW#3 ST#21 to ST#22
NW#4 ST#31 to ST#32

- Message Relay Function: Enable
- No. of Relayed Network 1: 1 - 4
- No. of Relayed Network 2: Not set

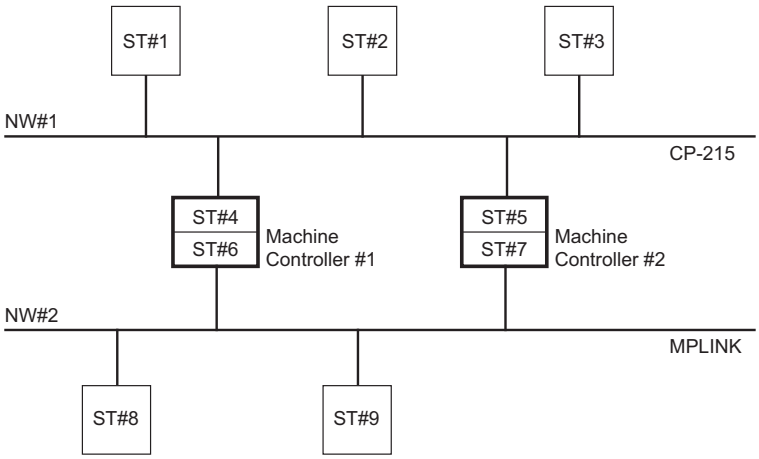
■ Remarks

With a configuration of two or more networks, be sure that only one Machine Controller (Relay station) performs relaying between two networks, for example networks A and B.*

In the example above, the Relay stations 1 and 2 relay each network as follows.

- Relay station between NW#1 and NW#2: Relay station 1 (ST#1/ST#10)
- Relay station between NW#1 and NW#3: Relay station 1 (ST#1/ST#20)
- Relay station between NW#2 and NW#3: Relay station 1 (ST#10/ST#20)
- Relay station between NW#2 and NW#4: Relay station 2 (ST#12/ST#30)

* The diagram below shows an example of the connection that must be avoided. If such a connection cannot be avoided for some reasons, set the **Message Relay Function** to **Enable** for either of the Machine Controller #1 or #2 so that only one Machine Controller is the relay station.



9.6 Connection Examples for MPLINK/CP-215 Communication

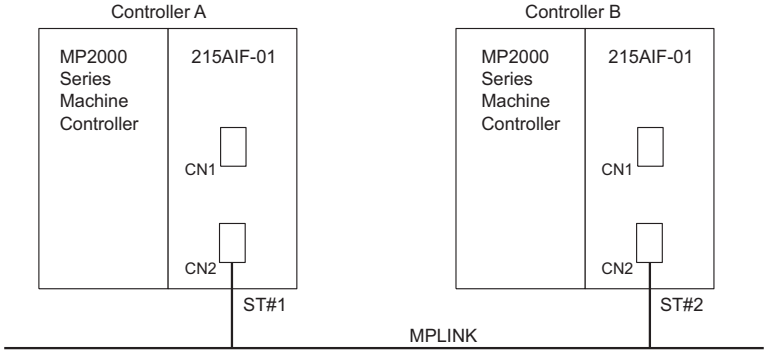
This section shows setting and programming examples for link communication and MEMOBUS message communication using MPLINK/CP-215 communications.

9.6.1 Link Communication Example

This section describes a link communication example between two MP2000 Series Machine Controllers.

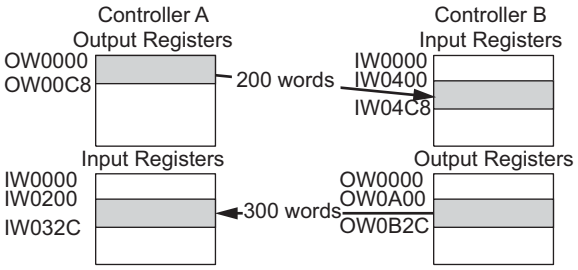
(1) System Configuration Example

Link communication is performed between ST#1 and ST#2 MP2000 Series Machine Controllers using the MPLINK transmissions by installing 215AIF-01 Modules on the MP2000 Series Machine Controllers.



In this example, the 215AIF-01 Modules on two MP2000 Series Machine Controllers are set as given below to perform link communication to send 200 words from Controller A to Controller B and 300 words from Controller B to Controller A.

Setting Item	Controller A	Controller B
Baud Rate	MPLINK–10M	MPLINK–10M
Local Network Address	1	1
Station Number	1	2
Output Address	OW0000	OW0A00
Output Data Size	200 words	300 words
Input Address	IW0200	IW0400
Input Data Size	300 words	200 words



- The input data size will be the output data size for the remote station.

(2) Controller A Settings

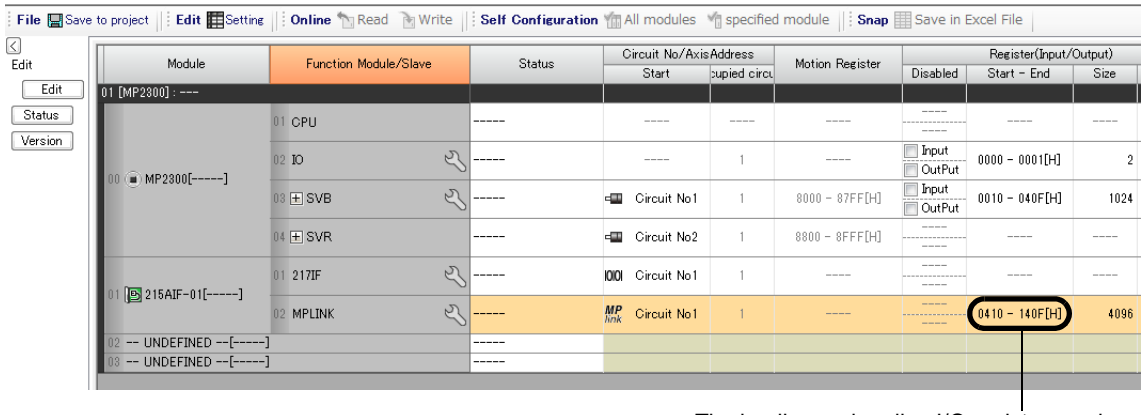
The 215AIF-01 settings for Controller A (station 1) are shown below.

■ Switch Settings

Name		Function	Setting	Meaning
SW1	BSR0	Communication Mode Setting	OFF	Sets MPLINK 10 Mbps.
	BSR1		OFF	
	INIT	Transmission Parameters Startup Selection	OFF	
SW2	A0	Network Number, 4 lower bits	ON	Sets the network address to 1.
	A1		OFF	
	A2		OFF	
	A3		OFF	
SW3	A4	Network Number, 4 upper bits	OFF	
	A5		OFF	
	A6		OFF	
	A7		OFF	
Rotary Switches	×10	10s digit of station number	0	Sets station number to 1.
	×1	1s digit of station number	1	

■ Module Configuration Definitions

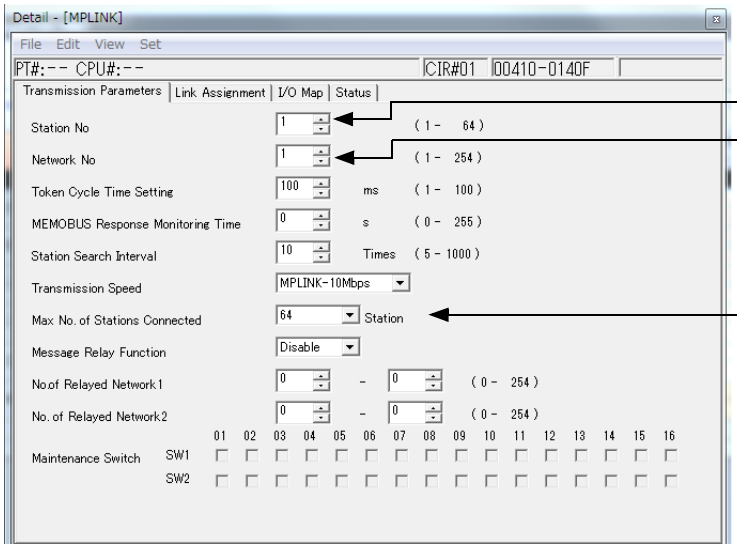
The following figure shows the settings for the 215AIF-01 transmission system.



The leading and ending I/O register numbers are automatically assigned when assignments are made to the 215AIF-01 Module.

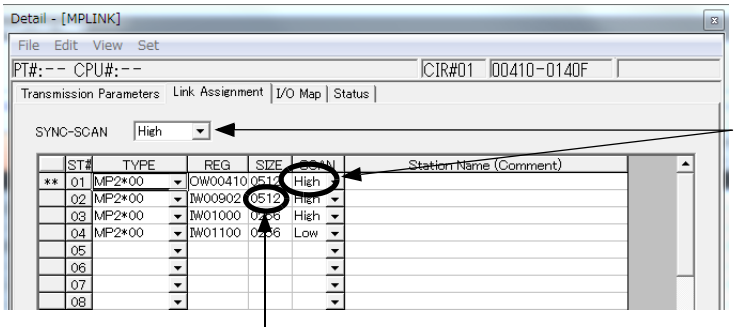
■ MPLINK Parameter Settings

The station number is set to 1 as shown in the following figure.



■ MPLINK Link Assignment

As shown in the following figure, ST#1 (Controller A) is set to output 200 words of data starting from OW0000 in the low-speed scan and ST#2 (Controller B) is set to output 300 words of data starting from IW0200 in the low-speed scan.



When setting the assignment for the remote station, always set the same size as the size output by the remote station. Data will not be input if the sizes are different.

(3) Controller B Settings

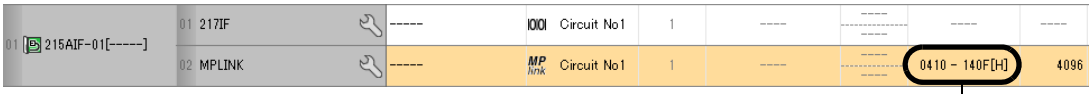
The 215AIF-01 settings for Controller B (station 2) are shown below.

■ Switch Settings

Name		Function	Setting	Meaning
SW1	BSR0	Communication mode setting	OFF	Sets MPLINK 10 Mbps.
	BSR1		OFF	
	INIT	Transmission Parameters Startup Selection	OFF	
SW2	A0	Network Number, 4 lower bits	ON	Sets the network address to 1.
	A1		OFF	
	A2		OFF	
	A3		OFF	
SW3	A4	Network Number, 4 upper bits	OFF	
	A5		OFF	
	A6		OFF	
	A7		OFF	
Rotary Switches	×10	10s digit of station number	0	Sets station number to 2.
	×1	1s digit of station number	2	

■ Module Configuration Definitions

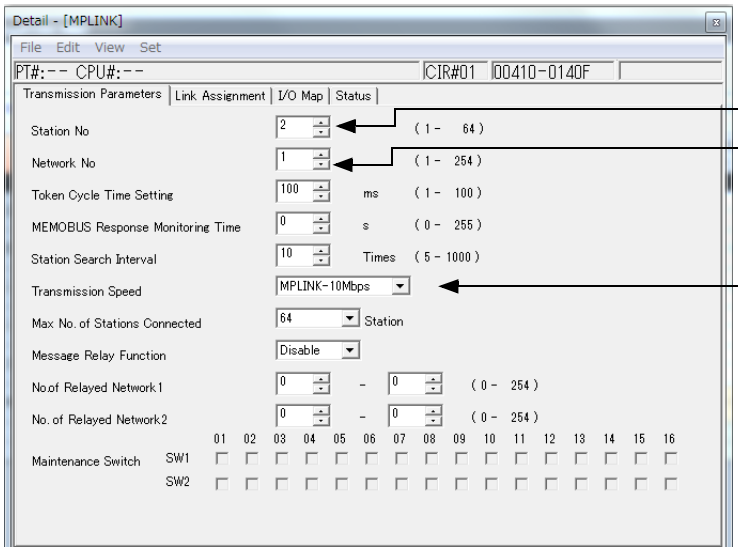
The following figure shows the settings for the 215AIF-01 transmission system.



The leading and ending I/O register numbers are automatically assigned when assignments are made to the 215AIF-01 Module.

■ MPLINK Parameter Settings

The station number is set to 2 as shown in the following figure.

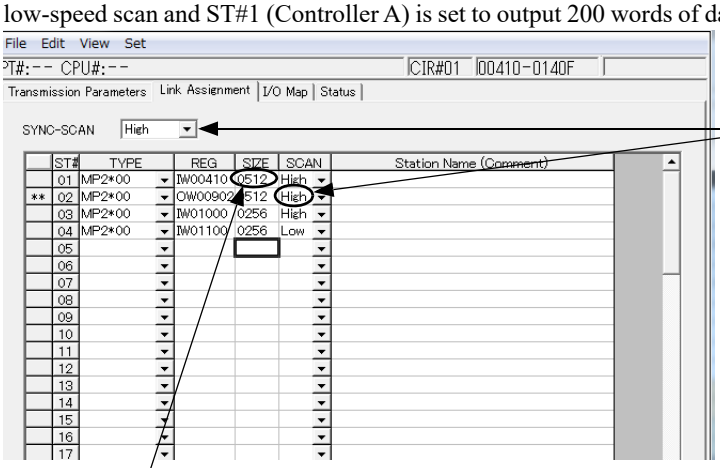


Station number: 2
Network number: 1

MPLINK-10Mbps

■ MPLINK Link Assignment

As shown in the following figure, ST#2 (Controller B) is set to output 300 words of data starting from OW0A00 in the low-speed scan and ST#1 (Controller A) is set to output 200 words of data starting from IW0400 in the low-speed scan.



(4) Programming Example

There in no programming required to perform link communication. Link assignments can be performed to exchange data using the assigned I/O registers. As long as the CPUs are running, I/O data will be constantly exchanged.

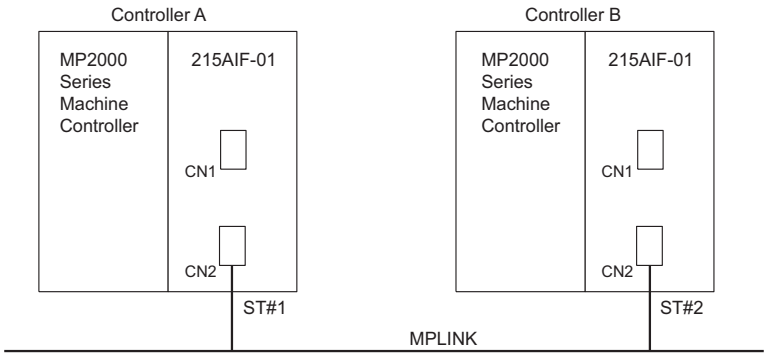
- The scan set for the link assignments (low/high) indicates the data refresh timing between the CPU Module and the 215AIF-01 Module. For the local Module, the data will be output to the transmission path not according to the scan setting, but rather when the 215AIF-01 Module receives the token on the transmission path. Data output is thus not synchronized with the scan setting.

9.6.2 Message Communication Example

This section describes a message communication example between two MP2000 Series Machine Controllers.

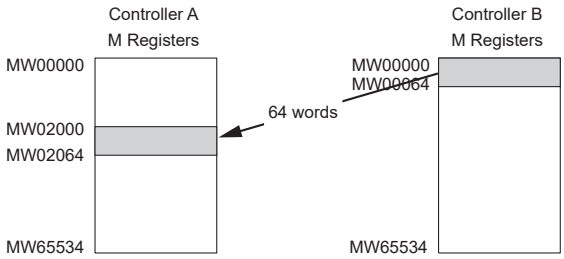
(1) System Configuration Example

Message communication is performed between ST#1 and ST#2 MP2000 Series Machine Controllers using the MPLINK transmissions by installing 215AIF-01 Modules on the MP2000 Series Machine Controllers.



In this example, the 215AIF-01 Modules on two MP2000 Series Machine Controllers are set as given below, and Controller A reads 64 words of data starting from MW00000 of Controller B. The data that is read is stored in MW02000 in Controller A. Here, Controller A is the Master for message communication, and Controller B is the Slave.

Setting Item	Controller A (Master)	Controller B (Slave)
Baud Rate	MPLINK–10M	MPLINK–10M
Local Network Address	1	1
Station Number	1	2
Read data size	64 words	
Read address	–	MW00000
Data storage address	MW02000	–



(2) Controller A Settings

Refer to *9.6.1 (2) Controller A Settings* on page 9-38 for the 215AIF-01 Module settings for station 1 (Controller A), including the switch settings, Module configuration, and MPLINK parameter settings.
Only message communication is performed in this example, so link assignment is not required.

(3) Controller B Settings

Refer to *9.6.1 (3) Controller B Settings* on page 9-40 for the 215AIF-01 Module settings for station 2 (Controller B), including the switch settings, Module configuration, and MPLINK parameter settings.
Only message communication is performed in this example, so link assignment is not required.

(4) Procedure to Start Communication

Communication is started using the following procedure. Ladder applications must be created in advance, with MSG-SND programmed in Controller A (Master) and MSG-RCV programmed in Controller B (Slave).

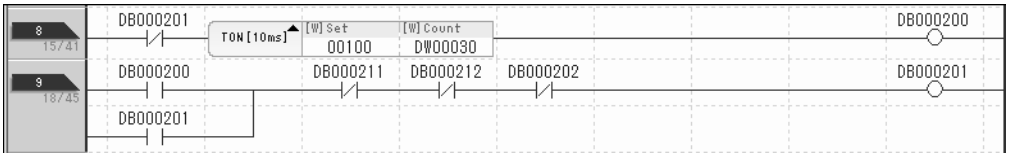
1. Start executing the MSG-RCV function in Controller B (Slave).

2. Turn ON the **Execute** parameter in the MSG-SND function in Controller A (Master).

Example ladder programming using the MSG-SND and MSG-RCV functions is shown in *9.6.2 (5) Programming Example* on page 9-44.
In *9.6.2 (5) [b] Ladder Programming Example Using the Message Receive Function (MSG-RCV)* on page 9-45, MSG-RCV function execution is started with SB000004 (Always ON Coil). Execution of the MSG-RCV function will thus start as soon as the power is turned ON to Controller B.
In *9.6.2 (1) [a] Ladder Programming Example Using the Message Send Function (MSG-SND)* on page 9-44, a message will be sent as soon as DB000201 (the register set for Execute in the MSG-SND function) turns ON, thus starting communication with Controller A.

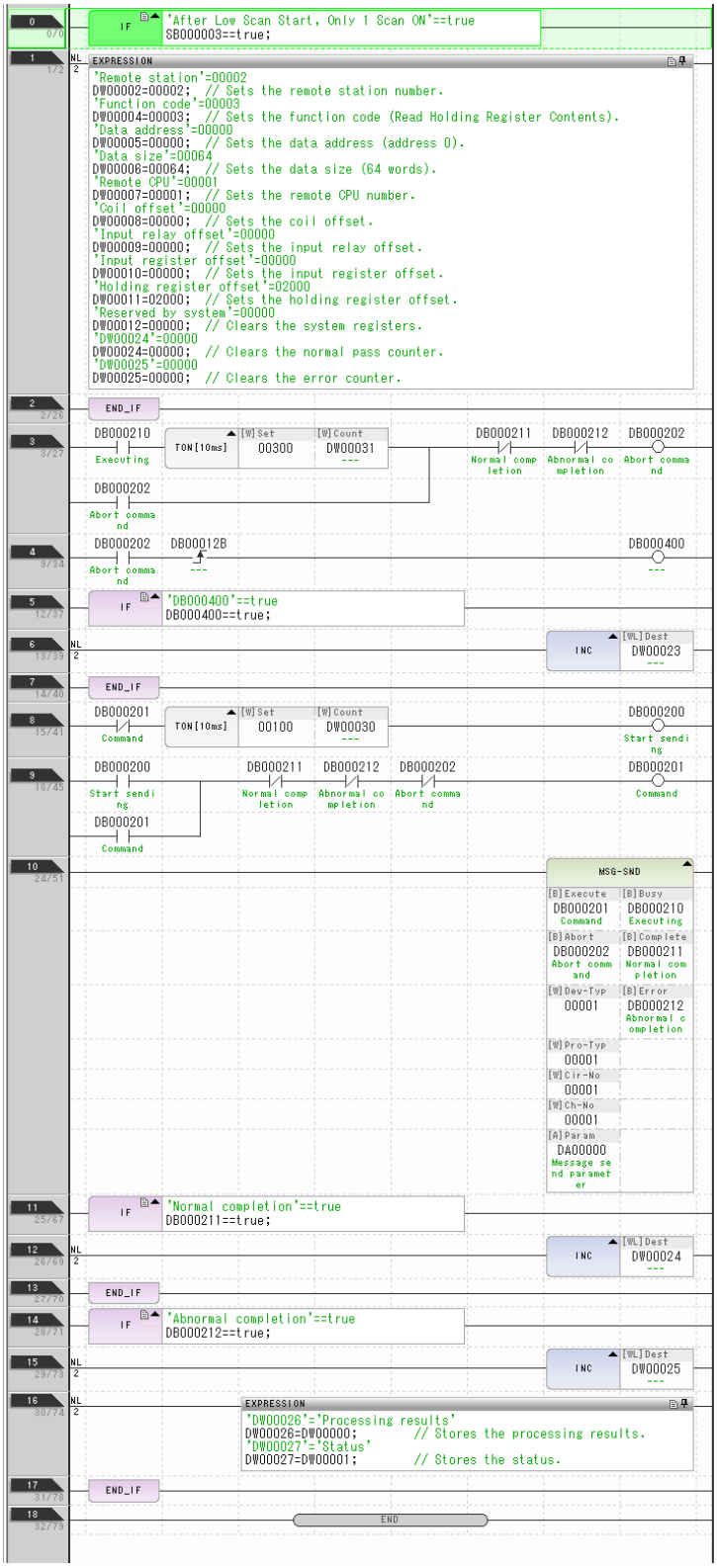
I/O Definitions	No.	Name	Setting Example	Description
Inputs	1	Execute	DB000201	Message send processing is started when the send command (Execute) turns ON.

- In *9.6.2 (1) [a] Ladder Programming Example Using the Message Send Function (MSG-SND)* on page 9-44, the programming is written so that a message will be sent every second after one second has expired after the low-speed (or high-speed) scan has started. To change the interval at which messages are sent, change the set value of the timer shown below.



(5) Programming Example

[a] Ladder Programming Example Using the Message Send Function (MSG-SND)



Sets parameters in the first scan after startup. (Low-speed scan (DWG.L): SB000003, High-speed scan (DWG.H): SB000001)

Sets PARAM02 to PARAM11, clears the system registers, and clears the error counter.

Forced stop if processing is not completed within 3 seconds after sending the Send command.

Increments the stop counter when the Abort command is sent.

Starts sending every 1 second.

Function Send command

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: Always 1 (MEMOBUS communication)
Cir-No. (circuit number): 2 (connection port)
Ch-No. (channel number): 1 (always 1 for the 215AIF-01)

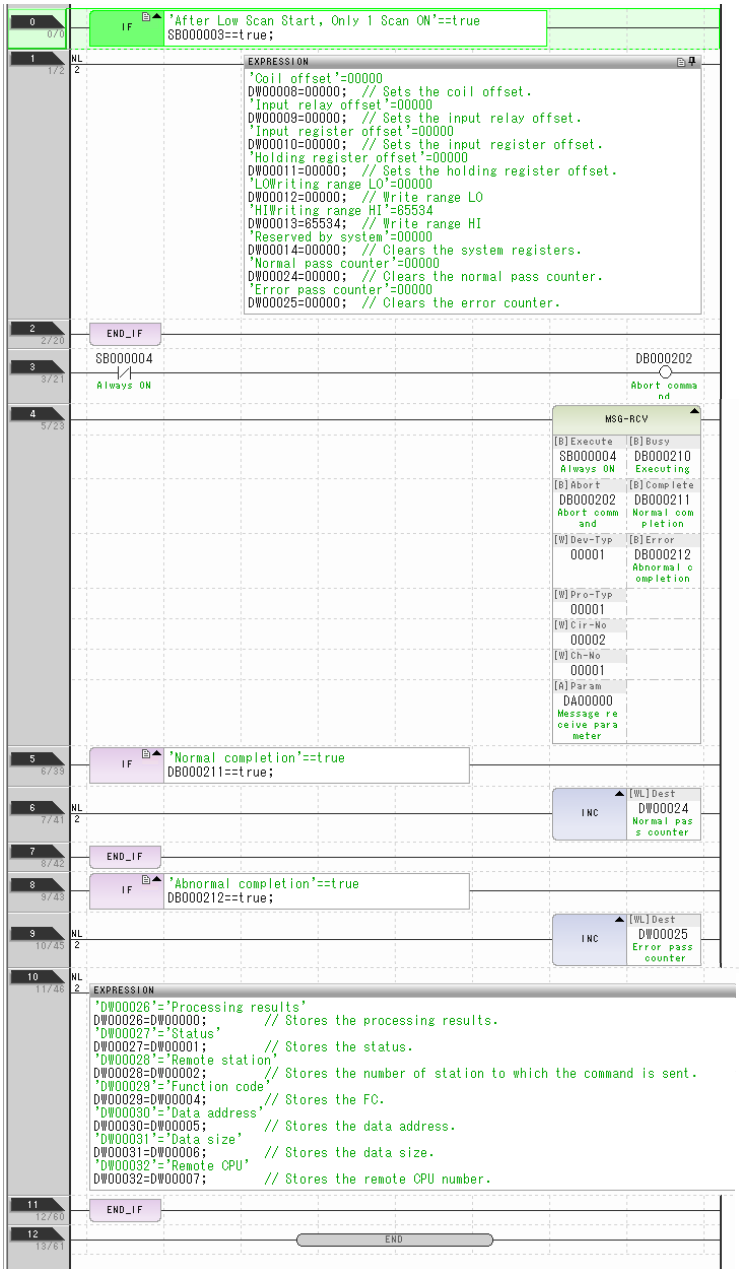
Turns ON after the command message is sent and the response message is received.
Increments the normal pass counter.

Ended with transmission errors.

Increments the abnormal pass counter.

Stores the processing results and the status.

[b] Ladder Programming Example Using the Message Receive Function (MSG-RCV)



Sets parameters in the first scan after startup.
(Low-speed scan (DWG.L): SB000003,
High-speed scan (DWG.H): SB000001)

Sets PARAM08 to PARAM13, clears the system registers, and clears the error counter.

Always turns the Abort command OFF.

The executing coil turns ON when the CPU receives the command. The normal completion or abnormal completion coil turns ON at the completion of processing.
Pro-Type: 1 (MEMOBUS communication)
Cir-No. (circuit number): 3 (connection port)
Ch-No. (channel number): 1 (always 1 for the 215AIF-01)

The normal completion coil turns ON at normal completion.

Increments the normal pass counter.

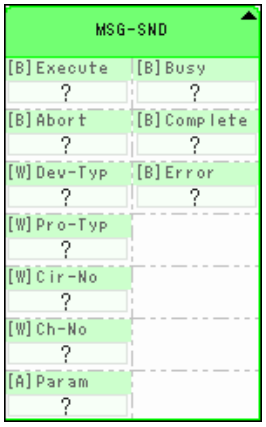
The abnormal completion coil turns ON at abnormal completion.

Increments the abnormal pass counter.

Stores the processing results, the status, the number of station to which the command is sent, the FC, the data address, the data size, and the remote CPU number.

9.7 Message Send Function

9.7.1 Basic Specifications

Function Name	MSG-SND			
Function	Sends a message to a remote station on the line specified by the Transmission Device Type parameter (DEV-TYP). The function supports multiple protocols. The Execute command must be held ON until the Complete or Error output turns ON.			
Function Definition				
I/O Definitions	No.	Name	I/O Designation*1	Description
Inputs	1	Execute	B-VAL	Send Message command
	2	Abort	B-VAL	Send Message Abort command
	3	Dev-Typ	I-REG	Transmission device type MPLINK/CP-215 (MPLINK) = 1, RS-232C/RS-422/RS-485 (217IF) = 5, Ethernet (218IF) = 6, DeviceNet (260IF) = 11, Ethernet (218IFA, 218IFB, 218IFC) = 16
	4	Pro-Typ	I-REG	Communication Protocol MEMOBUS = 1*2, Non-procedure 1*3 = 2, Non-procedure 2*3 = 3
	5	Cir-No	I-REG	Circuit Number MPLINK/CP-215 (MPLINK) = 1 to 8, RS-232C/RS-422/RS-485 (217IF) = 1 to 16, Ethernet (218IF) = 1 to 8, DeviceNet (260IF) = 1 to 8, Ethernet (218IFB) = 1 to 8
	6	Ch-No	I-REG	Transmission Buffer Channel Number MPLINK/CP-215 (MPLINK) = 1 to 12, RS-232C/RS-422/RS-485 (217IF) = 1, Ethernet (218IF) = 1 to 10, DeviceNet (260IF) = 1 to 4, Ethernet (218IFB) = 1 to 10
	7	Param	Address input	Parameter List Leading Address (MA, DA)
Outputs	8	Busy	B-VAL	Processing in progress.
	9	Complete	B-VAL	Processing completed.
	10	Error	B-VAL	Error has occurred.

* 1. The I/O designations are as follows:
B-VAL: I/O is specified as bit data.
I-REG: I/O is specified as integer data. Specify the number of an integer register.
Constants (immediate data) can also be specified for inputs.
Address input: The address of the specified register (any integer register) is passed to the function.

* 2. To use the MEMOBUS, Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, select the MEMOBUS protocol by setting the Communication Protocol parameter (PRO-TYP) to 1. The transmission device will automatically convert to the proper protocol.

* 3. Non-procedure 1: Data is sent in word units.
Non-procedure 2: Data is sent in byte units.

9.7.2 Inputs and Outputs for the Message Send Function

(1) Inputs

The following table gives the registers that can be used for the inputs.

Inputs	I/O Designation	Applicable Registers
Execute Abort	B-VAL	Any bit registers (including those with subscripts) except for # and C registers
Dev-Typ Pro-Typ Cir-No Ch-No	I-REG	Any integer registers (including those with subscripts), Constants
Param	Address input	Any register addresses (including those with subscripts) except for # and C registers

The following sections describe the inputs in more detail.

[a] Execute (Send Message Execute Command)

Specify the bit that will be used to control execution of the Message Send function.
Message send processing is started when the Execute command turns ON. To execute processing, this bit must be turned ON and OFF, e.g., from the ladder program.

- The Execute command must be held at ON until the Complete or Error output turns ON. The message is sent when the Execute command turns ON. To send another message, always turn OFF the Execute command for at least one scan.

[b] Abort (Send Message Abort Command)

Specify the bit that will be used to abort the Message Send function.
Sending the message will be aborted when the Abort command turns ON. The Abort command takes priority over the Execute command.
To abort processing, this bit must be turned ON and OFF, e.g., from the ladder program.

[c] Dev-Typ (Transmission Device Type)

The Dev-Typ input specifies the transmission device type as shown in the following table.

Device	Type Code
MPLINK/CP-215 (MPLINK)	1
RS-232C/422/485 (217IF)	5
Ethernet (218IF)	6
DeviceNet (260IF)	11
Ethernet (218IFA, 218IFB, 218IFC)	16

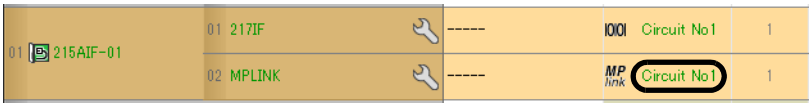
[d] Pro-Typ (Communication Protocol)

The Pro-Typ input specifies the communication protocol as shown in the following table.

Type Code	Communication Protocol	Remarks
1	MEMOBUS	To use the Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, set the type code to 1. The transmission device will automatically convert to the proper protocol.
2	Non-procedure 1 (word units)	Data will be sent in word units using the non-procedure protocol. No response is received from the remote station.
3	Non-procedure 2 (byte units)	Data will be sent in byte units using the non-procedure protocol. No response is received from the remote station.

[e] Cir-No (Circuit Number)

The Cir-No input specifies the circuit number of the transmission device.
Set the circuit number to the value displayed in the **MPE720 Module Configuration** Window.



The valid circuit number ranges are given in the following table.

Device	Valid Circuit Numbers
MPLINK/CP-215 (MPLINK)	1 to 8
RS-232C/422/485 (217IF)	1 to 16
Ethernet (218IF)	1 to 8
DeviceNet (260IF)	1 to 8
Ethernet (218IFB)	1 to 8

[f] Ch-No (Channel Number)

The Ch-No input specifies the channel number of the transmission buffer.

Any channel number can be specified as long as it is within the valid range. If more than one function is being executed at the same time, do not specify the same channel number more than once for the same modem number. (The same channel number can be used as long as the functions are not executed at the same time.)

The valid channel number ranges are given in the following table.

Device	Valid Channel Numbers
MPLINK/CP-215 (MPLINK)	1 to 12
RS-232C/422/485 (217IF)	1
Ethernet (218IF)	1 to 10
DeviceNet (260IF)	1 to 4
Ethernet (218IFB)	1 to 10

<Example>

If an MPLINK/CP-215 (MPLINK) transmission device is used, there are 12 channels of transmission buffers from 1 to 12 for sending and receiving combined, so up to 12 messages can be sent and received at the same time.

- One MSG-SND (or MSG-RCV) function must be programmed for each circuit being used at the same time.
- Refer to *A.3 Transmission Buffer Channels* on page A-9 for information on transmission buffer channels.

[g] Param (Parameter List Leading Address)

The PARAM input specifies the leading address of the parameter list. A parameter list will be automatically created from the 17 words starting with the specified address. Use the parameter list to input the function code and other related parameters. The processing results and status are also output to the parameter list.

- Refer to *9.7.3 Parameter List for MPLINK/CP-215 Transmissions (MPLINK)* on page 9-51 for information on the parameter list.

Example: The following parameter list will be created when the Parameter List Leading Address is set to DA000000.

Register	Parameter F·····0
DW000000	PARAM00
DW000001	PARAM01
DW000002	PARAM02
DW000003	PARAM03
DW000004	PARAM04
DW000005	PARAM05
DW000006	PARAM06
DW000007	PARAM07
DW000008	PARAM08
DW000009	PARAM09
DW000010	PARAM10
DW000011	PARAM11
DW000012	PARAM12
DW000013	PARAM13
DW000014	PARAM14
DW000015	PARAM15
DW000016	PARAM16

(2) Outputs

The following table gives the registers that can be used for the outputs.

Inputs	I/O Designation	Applicable Registers
Busy Complete Error	B-VAL	Any bit registers (including those with subscripts) except for # and C registers

The following sections describe the outputs in more detail.

[a] Busy (Processing in Progress)

Specify the bit that will report when sending the message is being processed.
The Busy output will be ON while message send processing or abort processing is in progress.
Keep the Execute command or Abort command ON while the Busy output is ON.

[b] Complete (Processing Completed)

Specify the bit that will report when processing to send the message has been completed.
The Complete output will be ON for only one scan after message send processing or abort processing has been completed normally.

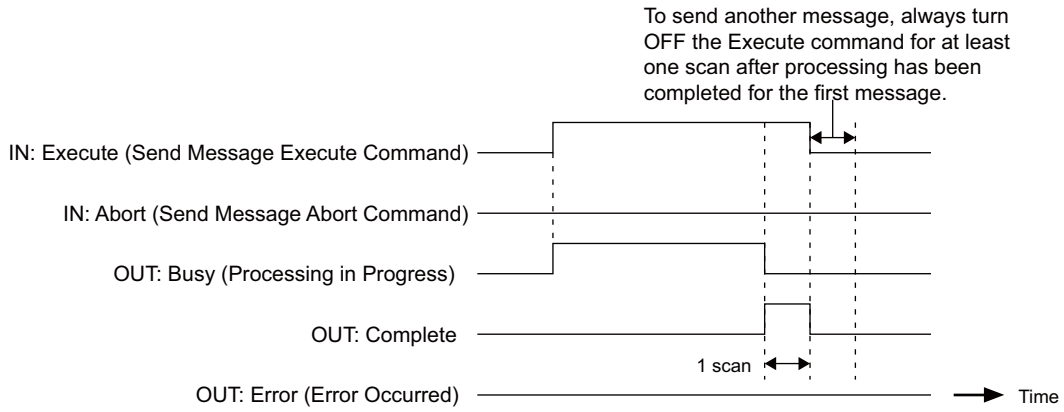
[c] ERROR (Error Occurred)

Specify the bit that will report when an error has occurred while sending the message.
The Error output will turn ON for only one scan when an error occurs.

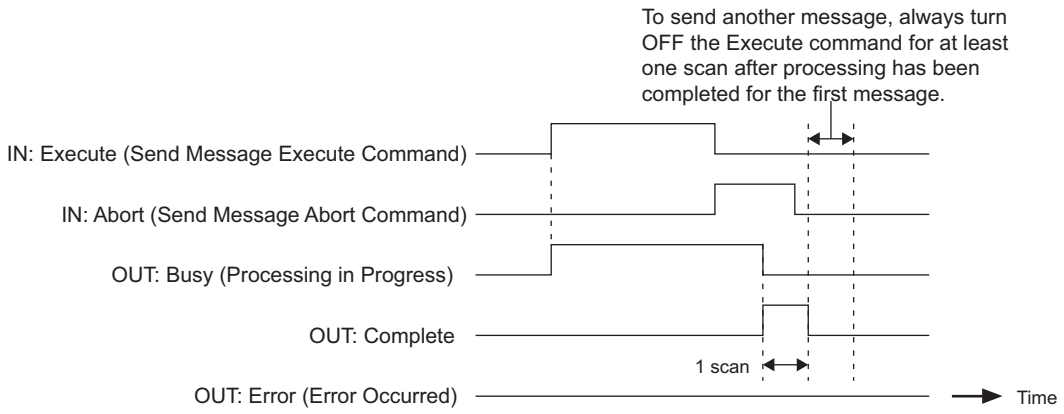
(3) I/O Timing Chart

The following timing charts show the bit inputs and outputs used with the MSG-SND function.

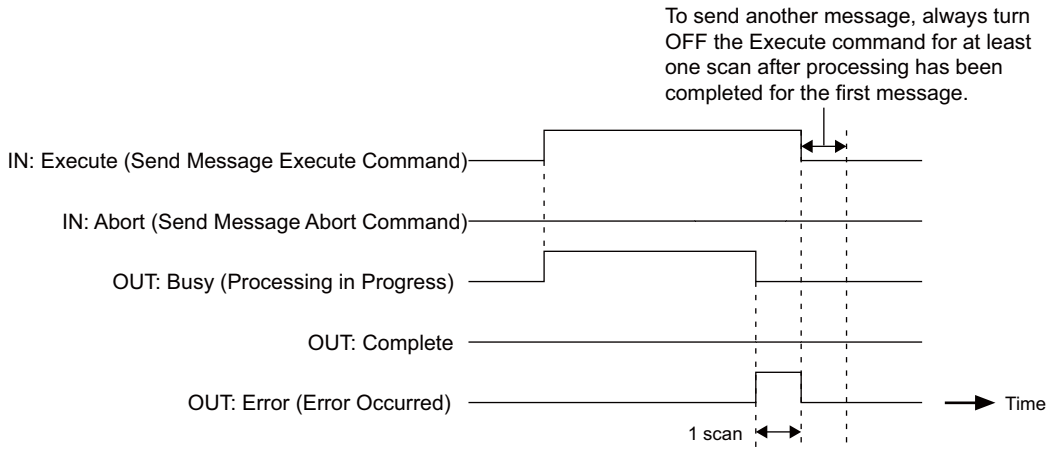
<Normal Processing>



<Abort Processing>



<Error Processing>



9.7.3 Parameter List for MPLINK/CP-215 Transmissions (MPLINK)

The Param input to the MSG-SND function is a parameter list structure consisting of 17 words. The value of the Param input is the leading address (MA or DA) of the parameter list.
Use the parameter list to input the connection number, function code, and other related parameters. The processing results and status are also output to the parameter list.
The parameter lists for the MEMOBUS and non-procedure communication protocols are given below.

■ Parameter List for MEMOBUS Protocol

PARAM No.	IN/OUT	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	IN	Remote station number	Specifies the remote station number.
03	IN	Option	Sets options specific to each transmission device.
04	IN	Function code	Sets the function code to be sent.
05	IN	Data address	Specifies the leading address of the data.
06	IN	Data size	Sets the data size to be read or written.
07	IN	Remote CPU number	Sets the remote CPU number.
08	IN	Coil offset	Sets the offset word address of the coils.
09	IN	Input relay offset	Sets the offset word address of the input relays.
10	IN	Input register offset	Sets the offset word address of the input registers.
11	IN	Holding register offset	Sets the offset word address of the holding registers.
12	SYS	Reserved by the system (1).	
13 to 16	SYS	Reserved by the system (2).	

- IN: Input, OUT: Output, SYS: Used by the system.
- Refer to *9.7.4 Function Setting Example and Parameter Details for the Extended MEMOBUS Protocol* on page 9-52 for details on parameters.

■ Parameter List for Non-procedure Protocol

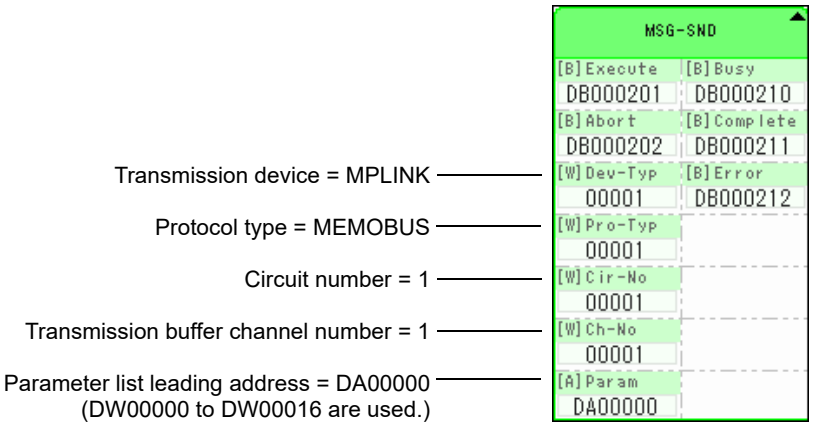
PARAM No.	IN/OUT	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	IN	Remote station number	Specifies the remote station number.
03	IN	(Not used.)	
04	IN	(Not used.)	
05	IN	Data address	Specifies the leading address of the data.
06	IN	Data size	Sets the data size to be read or written.
07	IN	Remote CPU number	Sets the remote CPU number.
08	IN	(Not used.)	
09	IN	(Not used.)	
10	IN	(Not used.)	
11	IN	Register offset	Sets the offset word address of the registers.
12	SYS	Reserved by the system (1).	
13 to 16	SYS	Reserved by the system (2).	

- IN: Input, OUT: Output, SYS: Used by the system.
- Refer to *9.7.5 Function Setting Example and Parameter Details for the Non-procedure Protocol* on page 9-59 for details on parameters.

9.7.4 Function Setting Example and Parameter Details for the Extended MEMOBUS Protocol

(1) MSG-SND Function Setting Example

This example shows the settings for using the MPLINK as the transmission device.



To use the Extended MEMOBUS protocol, the protocol type is set to MEMOBUS.

The circuit number is set to the circuit number assigned to the 215AIF-01 transmission device.

Transmission buffer channel numbers in the same line must all be unique.

Refer to *9.7.3 Parameter List for MPLINK/CP-215 Transmissions (MPLINK)* on page 9-51 for information on the settings.

(2) Parameter Details for Extended MEMOBUS Protocol

This section describes the parameter list in detail for using the Extended MEMOBUS protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARAM00. The lower-place byte is for system analysis.

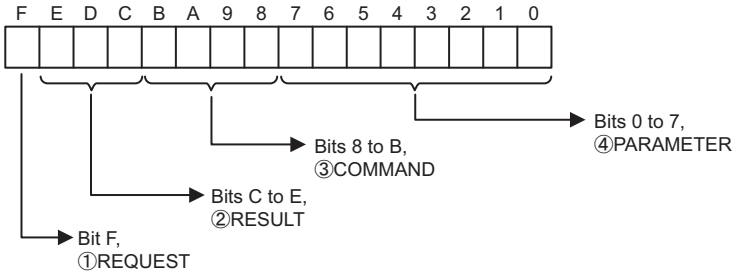
Value of Processing Result	Meaning
00xxH	Processing in progress (Busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	Function code error	An undefined function code was sent or received. Check PARAM04 (function code).
82xxH	Address setting error	One of the following settings is not within the valid range. Check the settings. PARAM05 (data address) PARAM08 (coil offset) PARAM09 (input relay offset) PARAM10 (input register offset) PARAM11 (holding register offset)
83xxH	Data size error	The size of the sent or received data is not within the allowable range. Check PARAM06 (data size).
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-SND function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-SND function.
86xxH	Station number error	The station number is not within the allowable range. Check PARAM02 (connection number).
87xxH	–	Reserved by the system.
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-SND function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-SND function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-SND function is output to these bits.

Code	Abbreviation	Meaning
1	SEND_OK	Sending has been completed normally.
	SEND_NG	Send error (The processing results (PARAM00) indicate an error.)
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-SND function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send General-purpose Message with Non-procedure Protocol
2	U_REC	Receive General-purpose Message with Non-procedure Protocol
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response.
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response.
C	MR_SEND	Send MEMOBUS response

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the remote station number is output.

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Station number outside the allowable range
	02	MEMOBUS response receive monitor time error
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error
Other values of RESULT	XX	Remote station number

[c] PARAM02: Remote Station Number

PARAM02 specifies the remote station number.
When using an MPLINK transmission device, set the station number of the remote station. The valid setting ranges are given in the following table.

Transmission Device	Connection Number	Remarks
MPLINK/CP-215 (MPLINK)	1 to 64	The message is sent to the remote station with the designated station number.
	FFH	The message is sent to all the stations. (Batch processing)

- If the remote station is the 215AI-01, the remote station number is set to the station number of the remote station.

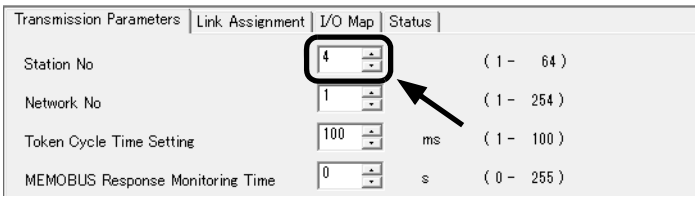


Fig. 9.4 The **MPLINK Transmission Parameters** Tab Page of the **MPE720 Module Configuration** Window

[d] PARAM03: Option

PARAM03 sets options specific to each transmission device.
PARAM03 is not used for the Extended MEMOBUS protocol. Nothing needs to be set.

[e] PARAM04: Function Code

PARAM04 sets the function code to be sent.
The function registered to the function code is used by specifying the function code here. Functions include reading coil and input relay status and writing holding registers.
The function codes used for the Extended MEMOBUS protocol are listed in the following table.

<Function Codes>

Function Code	Applicable Data Type	Function	Setting
00H	–	Not used.	–
01H	B	Read Coil Status	○
02H	B	Read Input Relay Status	○
03H	W	Read Holding Register Contents	○
04H	W	Read Input Register Contents	○
05H	B	Modify Status of a Single Coil	○
06H	W	Write to a Single Holding Register	○
07H	–	Not used.	–
08H	–	Loopback Test	○
09H	W	Read Holding Register Contents (Extended)	○
0AH	W	Read Input Register Contents (Extended)	○
0BH	W	Write to Holding Register (Extended)	○
0CH	–	Not used.	–
0DH	W	Read Discontinuous Holding Register Contents (Extended)	○
0EH	W	Write to Discontinuous Holding Registers (Extended)	○
0FH	B	Modify Status of Multiple Coils	○
10H	W	Write to Multiple Holding Registers	○

- B: Bit data, W: Word data
- ○: Can be set. ×: Cannot be set.
- Only MW (MB) registers may be used for sending and receiving messages when operating as the Master.
- MB, MW, IB, and IW registers can be used as coils, holding registers, input relays, and input registers, respectively, when operating as a Slave.

[f] PARAM05: Data Address

PARAM05 specifies the leading address of the data.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

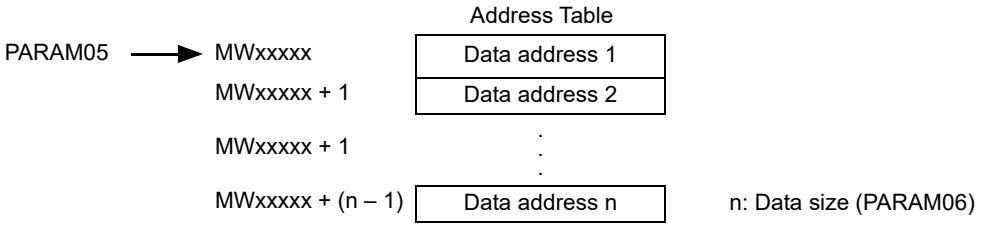
The data address setting depends on function code.
The data address setting ranges used for the Extended MEMOBUS protocol are listed in the following table.

<Data Address Setting Ranges for Extended MEMOBUS>

Function Code	Applicable Data Type	Function	Data Address Setting Range
00H	–	Not used.	Invalid
01H	B	Read Coil Status*1	0 to 65535 (0 to FFFFH)
02H	B	Read Input Relay Status*1	0 to 65535 (0 to FFFFH)
03H	W	Read Holding Register Contents*2	0 to 65534 (0 to FFFEH)
04H	W	Read Input Register Contents*2	0 to 32767 (0 to 7FFFH)
05H	B	Modify Status of a Single Coil*1	0 to 65535 (0 to FFFFH)
06H	W	Write to a Single Holding Register*2	0 to 65534 (0 to FFFEH)
07H	–	Not used.	Invalid
08H	–	Loopback Test	Invalid
09H	W	Read Holding Register Contents (Extended)*2	0 to 65534 (0 to FFFEH)
0AH	W	Read Input Register Contents (Extended)*2	0 to 32767 (0 to 7FFFH)
0BH	W	Write to Holding Register (Extended)*2	0 to 65534 (0 to FFFEH)
0CH	–	Not used.	Invalid
0DH	W	Read Discontinuous Holding Register Contents (Extended)*3	0 to 65534 (0 to FFFEH)
0EH	W	Write to Discontinuous Holding Registers (Extended)*3	0 to 65534 (0 to FFFEH)
0FH	B	Modify Status of Multiple Coils*1	0 to 65535 (0 to FFFFH)
10H	W	Write to Multiple Holding Registers*2	0 to 65534 (0 to FFFEH)

* 1. Read/write request for coils and relays: Set the leading bit address of the data.
* 2. Continuous read/write request for registers: Set the leading word address of the data.
* 3. Read/write request for discontinuous registers: Set the leading M register number of the address table.

- Address Table
  An address table specifies indirect addressing for discontinuous data. The M register specified by PARAM05 (data address) specifies the start of the address table and PARAM06 (data size) specifies the size of the address table.
  When reading data, set data addresses 1 to n to the address to be read at the remote station. The read data will be stored in the local station according to data addresses 1 to n.
  When writing data, the data stored in data addresses 1 to n at the local station will be read and then written to data addresses 1 to n at the remote station.
  The contents of an address table used to read/write discontinuous registers is illustrated below.



[g] PARAM06: Data Size

PARAM06 specifies the data size to be read or written as the number of bits or words.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size setting depends on function code and the transmission device.
The data size setting ranges used for the Extended MEMOBUS protocol are listed in the following table.

<Data Size Setting Ranges for Extended MEMOBUS>

Function Code	Applicable Data Type	Function	Data Address Setting Range
00H	–	Not used.	Invalid
01H	B	Read Coil Status*1	1 to 2000
02H	B	Read Input Relay Status*1	1 to 2000
03H	W	Read Holding Register Contents*2	1 to 125
04H	W	Read Input Register Contents*2	1 to 125
05H	B	Modify Status of a Single Coil	Invalid
06H	W	Write to a Single Holding Register	Invalid
07H	–	Not used.	Invalid
08H	–	Loopback Test	Invalid
09H	W	Read Holding Register Contents (Extended)*2	1 to 508
0AH	W	Read Input Register Contents (Extended)*2	1 to 508
0BH	W	Write to Holding Register (Extended)*2	1 to 507
0CH	–	Not used.	Invalid
0DH	W	Read Discontinuous Holding Register Contents (Extended)*2	1 to 508
0EH	W	Write to Discontinuous Holding Registers (Extended)*2	1 to 254
0FH	B	Modify Status of Multiple Coils*1	1 to 800
10H	W	Write to Multiple Holding Registers*2	1 to 100

* 1. Set the number of bits.
* 2. Set the number of words.
- The data sizes listed in the table are decimal values.

[h] PARAM07: Remote CPU Number

PARAM07 sets the remote CPU number.
If the remote device is an MP2000 Series Machine Controller, set the remote CPU number to 1.
If the remote device is a Yaskawa Controller, but not in the MP2000 Series and it consists of more than one CPU Module, set the destination CPU number.
In all other cases, set 0.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the offsets for the data read or write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter	Protocol Type
			Extended MEMOBUS
01H	Read Coil Status	PARAM08	○
02H	Read Input Relay Status	PARAM09	○
03H	Read Holding Register Contents	PARAM11	○
04H	Read Input Register Contents	PARAM10	○
05H	Modify Status of a Single Coil	PARAM08	○
06H	Write to a Single Holding Register	PARAM11	○
09H	Read Holding Register Contents (Extended)	PARAM11	○
0AH	Read Input Register Contents (Extended)	PARAM10	○
0BH	Write to Holding Register (Extended)	PARAM11	○
0DH	Read Discontinuous Holding Register Contents (Extended)	PARAM11	○
0EH	Write to Discontinuous Holding Registers (Extended)	PARAM11	○
0FH	Modify Status of Multiple Coils	PARAM08	○
10H	Write to Multiple Holding Registers	PARAM11	○

- ○: Can be set. ×: Cannot be set.

[j] PARAM12: Reserved by the System (1)

PARAM12 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

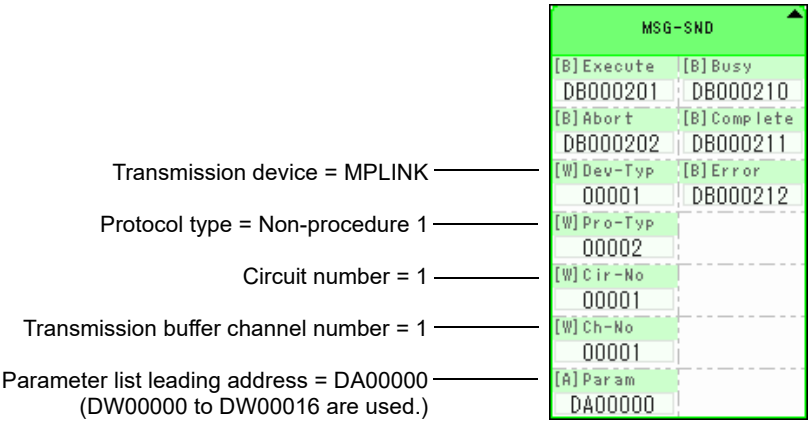
[k] PARAM13 to PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

9.7.5 Function Setting Example and Parameter Details for the Non-procedure Protocol

(1) MSG-SND Function Setting Example

This example shows the settings for using the MPLINK as the transmission device.



The protocol type (Pro-Typ) is set to 2 to use non-procedure protocol 1 and to 3 to use non-procedure protocol 2.
The circuit number is set to the circuit number assigned to the 215AIF-01 transmission device.
Transmission buffer channel numbers in the same line must all be unique.
Refer to *9.7.3 Parameter List for MPLINK/CP-215 Transmissions (MPLINK)* on page 9-51 for information on the settings.

(2) Parameter Details for Non-procedure Protocol

This section describes the parameter list in detail for using the non-procedure protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARAM00. The lower-place byte is for system analysis.

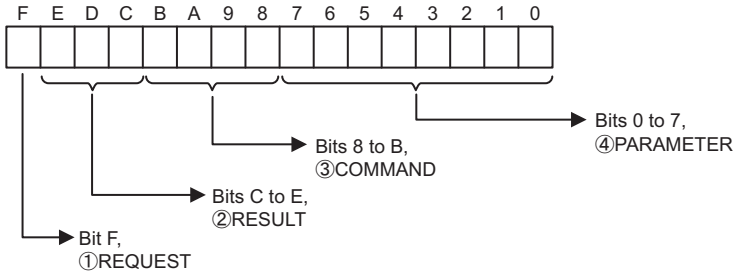
Value of Processing Result	Meaning
00xxH	Processing in progress (Busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	–	–
82xxH	Address setting error	One of the following settings is not within the valid range. Check the settings. PARAM05 (data address) PARAM11 (holding register offset)
83xxH	Data size error	The size of the sent or received data is not within the allowable range. Check PARAM06 (data size).
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-SND function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-SND function.
86xxH	Station number error	The station number is not within the allowable range. Check PARAM02 (connection number).
87xxH	–	Reserved by the system.
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-SND function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-SND function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-SND function is output to these bits.

Code	Abbreviation	Meaning
1	SEND_OK	Sending has been completed normally.
	SEND_NG	Send error (The processing results (PARAM00) indicate an error.)
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-SND function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send general-purpose message.
2	U_REC	Receive general-purpose message.
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response. (For MEMOBUS Protocol)
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response. (For MEMOBUS Protocol)
C	MR_SEND	Send MEMOBUS response (For MEMOBUS Protocol)

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the remote station number is output.

RESULT	Code (Hex)	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Station number outside the allowable range
	02	MEMOBUS response receive monitor time error (for MEMOBUS Protocol)
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error (For MEMOBUS Protocol)
Other values of RESULT	XX	Remote station number

[c] PARAM02: Remote Station Number

Refer to *9.7.4 (2) [c] PARAM02: Remote Station Number* on page 9-55.

[d] PARAM05: Data Address

PARAM05 specifies the leading address of the data.
The address is input as a decimal or hexadecimal value.

- Example: To set a leading address of MW01000, set 1000 (decimal) or 3E8H (hexadecimal).

The valid data address setting ranges are given in the following table.

<Data Address Setting Ranges for Non-procedure Protocol>

Non-procedure Protocol Type	Applicable Data Type	Function	Data Address Setting Range
Non-procedure 1	W	Sends data in word units.	0 to 65534 (0 to FFFEH)
Non-procedure 2	B	Sends data in byte units.*	0 to 65534 (0 to FFFEH)

* The address setting must be in word units.

[e] PARAM06: Data Size

PARAM06 specifies the data size to be read or written as the number of bits or words.
Do not allow the final address of the data, which is determined by the offset, data address, and data size, to exceed the valid range of addresses.
The data size setting depends on function code and the transmission device.

<Data Size Setting Ranges for Non-procedure Protocol>

Non-procedure Protocol Type	Applicable Data Type	Function	Data Address Setting Range
Non-procedure 1	W	Sends data in word units.*1	1 to 510
Non-procedure 2	B	Sends data in byte units.*2	1 to 1020

* 1. Set the number of words.
* 2. Set the number of bytes.

- The data sizes listed in the table are decimal values.

[f] PARAM07: Remote CPU Number

PARAM07 sets the remote CPU number.
If the remote device is an MP2000 Series Machine Controller, set the remote CPU number to 1.
If the remote device is a Yaskawa Controller, but not in the MP2000 Series and it consists of more than one CPU Module, set the destination CPU number.
In all other cases, set 0.

[g] PARAM11: Register Offset

PARAM11 sets the offset for the data write location at the device sending the message.
The position of the data is shifted backward by the number of words set for the offset at the device sending the message.

Example: PARAM11 is set as follows to offset the register address by 1000 words.

PARAM11 = 1000

- For details, refer to *A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND* on page A-2.
- Negative offsets cannot be set.

[h] PARAM12: Reserved by the System (1)

PARAM12 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

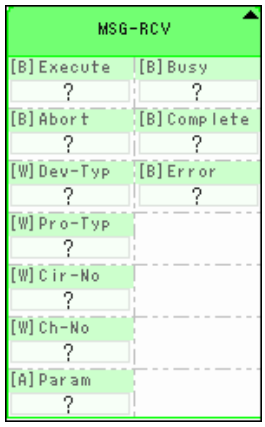
[i] PARAM13 to PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

9.8 Message Receive Function

This section describes the Message Receive function (MSG-RCV) used in a ladder program to receive a message.

9.8.1 Basic Specifications

Function Name	MSG-RCV			
Function	Receives a message from a remote station on the line specified by the Transmission Device Type parameter (DEV-TYP). The function supports multiple protocols. The Execute command must be held ON until the Complete or Error output turns ON.			
Function Definition				
I/O Definitions	No.	Name	I/O Designation*1	Description
Inputs	1	Execute	B-VAL	Receive Message Command
	2	Abort	B-VAL	Reception Abort Command
	3	Dev-Typ	I-REG	Transmission Device Type MPLINK/CP-215 (MPLINK) = 1, RS-232C/RS-422/RS-485 (217IF) = 5, Ethernet (218IF) = 6, Ethernet (218IFA, 218IFB, 218IFC) = 16
	4	Pro-Typ	I-REG	Communication Protocol MEMOBUS = 1*2, Non-procedure 1*3 = 2, Non-procedure 2*3 = 3
	5	Cir-No	I-REG	Circuit Number MPLINK/CP-215 (MPLINK) = 1 to 8, RS-232C/RS-422/RS-485 (217IF) = 1 to 16, Ethernet (218IF) = 1 to 8, Ethernet (218IFB) = 1 to 8
	6	Ch-No	I-REG	Transmission Buffer Channel Number MPLINK/CP-215 (MPLINK) = 1 to 12, RS-232C/RS-422/RS-485 (217IF) = 1, Ethernet (218IF) = 1 to 10, Ethernet (218IFB) = 1 to 10
	7	Param	Address input	Parameter List Leading Address (MA, DA)
Outputs	8	Busy	B-VAL	Processing in progress.
	9	Complete	B-VAL	Processing completed.
	10	Error	B-VAL	Error has occurred.

* 1. The I/O designations are as follows:
B-VAL: I/O is specified as bit data.
I-REG: I/O is specified as integer data. Specify the number of an integer register.
Constants (immediate data) can also be specified for inputs.
Address input: The address of the specified register (any integer register) is passed to the function.

* 2. To use the MEMOBUS, Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, select the MEMOBUS protocol by setting the Communication Protocol parameter (PRO-TYP) to 1. The transmission device will automatically convert to the proper protocol.

* 3. Non-procedure 1: Data is received in word units.
Non-procedure 2: Data is received in byte units.

9.8.2 Inputs and Outputs for the Message Receive Function

(1) Inputs

The following table gives the registers that can be used for the inputs.

Inputs	I/O Designation	Applicable Registers
Execute Abort	B-VAL	Any bit registers (including those with subscripts) except for # and C registers
Dev-Typ Pro-Typ Cir-No Ch-No	I-REG	Any integer registers (including those with subscripts), constants
Param	Address input	Any register addresses (including those with subscripts) except for # and C registers

The following sections describe the inputs in more detail.

[a] Execute (Receive Message Execute Command)

Specify the bit that will be used to control execution of the Message Receive function.
Message receive processing is started when the Execute command turns ON. To execute processing, this bit must be turned ON and OFF, e.g., from the ladder program.

- The Execute command must be held at ON until the Complete or Error output turns ON. A message is received when the Execute command turns ON. To receive another message, always turn OFF the Execute command for at least one scan.

[b] Abort (Receive Message Abort Command)

Specify the bit that will be used to abort the Message Receive function.
Receiving the message will be aborted when the Abort command turns ON. The Abort command takes priority over the Execute command.
To abort processing, this bit must be turned ON and OFF, e.g., from the ladder program.

[c] Dev-Typ (Transmission Device Type)

The Dev-Typ input specifies the transmission device type as shown in the following table.

Device	Type Code
MPLINK/CP-215 (MPLINK)	1
RS-232C/422/485 (217IF)	5
Ethernet (218IF)	6
Ethernet (218IFA, 218IFB, 218IFC)	16

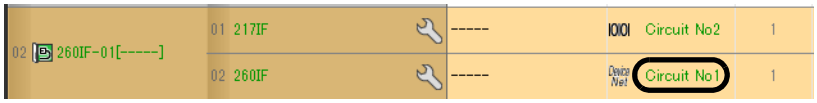
[d] Pro-Typ (Communication Protocol)

The Pro-Typ input specifies the communication protocol as shown in the following table.

Type Code	Communication Protocol	Remarks
1	MEMOBUS	To use the Extended MEMOBUS, MELSEC, OMRON, or MODBUS/TCP protocol to send a message, set the type code to 1. The transmission device will automatically convert to the proper protocol.
2	Non-procedure 1 (word units)	Data will be received in word units using the non-procedure protocol. No response is received from the remote station.
3	Non-procedure 2 (byte units)	Data will be received in byte units using the non-procedure protocol. No response is received from the remote station.

[e] Cir-No (Circuit Number)

The Cir-No input specifies the circuit number of the transmission device.
Set the circuit number to the value displayed in the **MPE720 Module Configuration** Window.



The valid circuit number ranges are given in the following table.

Device	Valid Circuit Numbers
MPLINK/CP-215 (MPLINK)	1 to 8
RS-232C/422/485 (217IF)	1 to 16
Ethernet (218IF)	1 to 8
Ethernet (218IFB)	1 to 8

[f] Ch-No (Channel Number)

The Ch-No input specifies the channel number of the transmission buffer.
Any channel number can be specified as long as it is within the valid range. If more than one function is being executed at the same time, do not specify the same channel number more than once for the same model number. (The same channel number can be used as long as the functions are not executed at the same time.)

The valid channel number ranges are given in the following table.

Device	Valid Channel Numbers
MPLINK/CP-215 (MPLINK)	1 to 12
RS-232C/422/485 (217IF)	1
Ethernet (218IF)	1 to 10
Ethernet (218IFB)	1 to 10

<Example>

If a MPLINK/CP-215 (MPLINK) transmission device is used, there are 12 channels of transmission buffers from 1 to 12 for sending and receiving combined, so up to 12 messages can be sent and received at the same time.

- One MSG-RCV (or MSG-SND) function must be programmed for each circuit being used at the same time.
- Refer to *A.3 Transmission Buffer Channels* on page A-9 for information on transmission buffer channels.

[g] Param (Parameter List Leading Address)

The PARAM input specifies the leading address of the parameter list. A parameter list will be automatically created from the 17 words starting with the specified address. Use the parameter list to input the function code and other related parameters. The processing results and status are also output to the parameter list.

- Refer to *9.8.3 Parameter List for MPLINK/CP-215 Transmissions (MPLINK)* on page 9-68 for information on the parameter list.

Example: The following parameter list will be created when the Parameter List Leading Address is set to DA000000.

Register	Parameter F·····0
DW000000	PARAM00
DW000001	PARAM01
DW000002	PARAM02
DW000003	PARAM03
DW000004	PARAM04
DW000005	PARAM05
DW000006	PARAM06
DW000007	PARAM07
DW000008	PARAM08
DW000009	PARAM09
DW000010	PARAM10
DW000011	PARAM11
DW000012	PARAM12
DW000013	PARAM13
DW000014	PARAM14
DW000015	PARAM15
DW000016	PARAM16

(2) Outputs

The following table gives the registers that can be used for the outputs.

Inputs	I/O Designation	Applicable Registers
Busy Complete Error	B-VAL	Any bit registers (including those with subscripts) except for # and C registers

The following sections describe the outputs in more detail.

[a] Busy (Processing in Progress)

Specify the bit that will report when receiving the message is being processed.
The Busy output will be ON while message receive processing or abort processing is in progress.
Keep the Execute command or Abort command ON while the Busy output is ON.

[b] Complete (Processing Completed)

Specify the bit that will report when processing to receive the message has been completed.
The Complete output will be ON for only one scan after message receive processing or abort processing has been completed normally.

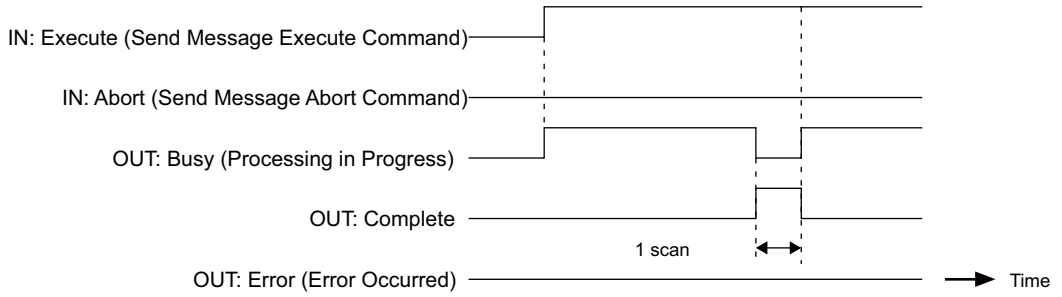
[c] ERROR (Error Occurred)

Specify the bit that will report when an error has occurred while receiving the message.
The Error output will turn ON for only one scan when an error occurs.

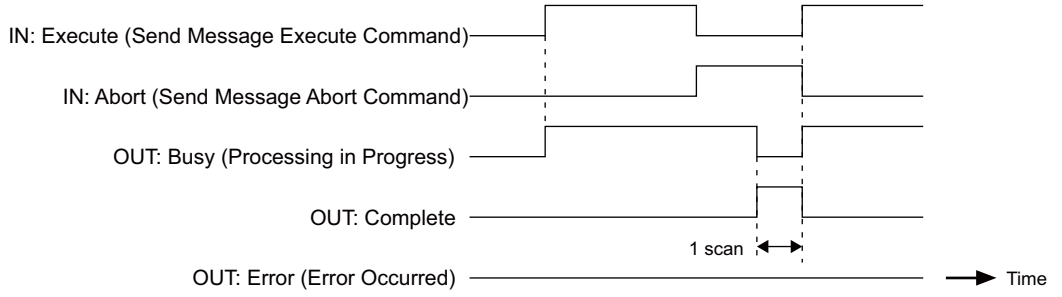
(3) I/O Timing Chart

The following timing charts show the bit inputs and outputs used with the MSG-RCV function.

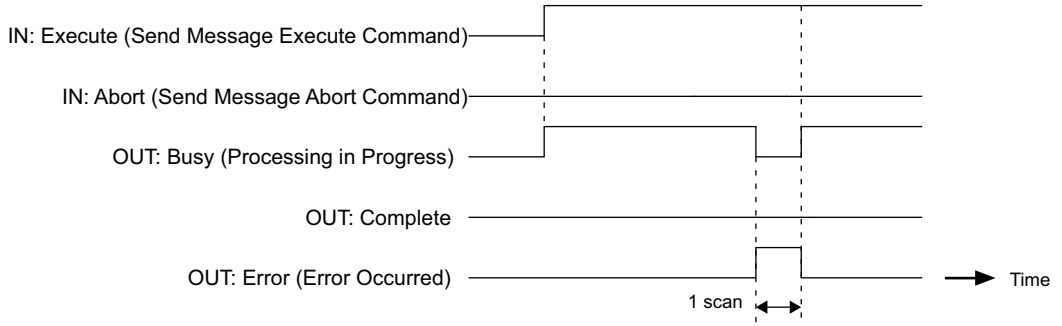
<Normal Processing>



<Abort Processing>



<Error Processing>



9.8.3 Parameter List for MPLINK/CP-215 Transmissions (MPLINK)

The Param input to the MSG-RCV function is a parameter list structure consisting of 17 words. The value of the Param input is the leading address (MA or DA) of the parameter list.
Use the parameter list to input the connection number, function code, and other related parameters. The processing results and status are also output to the parameter list.
The parameter lists for the Extended MEMOBUS and non-procedure communication protocols are given below for MPLINK/CP215 transmissions.

■ Parameter List for Extended MEMOBUS Protocol

PARAM No.	IN/OUT*	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	OUT	Remote station number	The number of the source station is output.
03	OUT	Option	Options specific to each transmission device are output here.
04	OUT	Function code	The function code requested by the sending station is output here.
05	OUT	Data address	The leading address of the data requested by the sending station is output here.
06	OUT	Data size	The read or write data size requested by the sending station is output here.
07	OUT	Remote CPU number	The remote CPU number is output here.
08	IN	Coil offset	Sets the offset word address of the coils.
09	IN	Input relay offset	Sets the offset word address of the input relays.
10	IN	Input register offset	Sets the offset word address of the input registers.
11	IN	Holding register offset	Sets the offset word address of the holding registers.
12	IN	Write range LO	Sets the leading address of the write range.
13	IN	Write range HI	Sets the final address of the write range.
14	SYS	Reserved by the system (1).	
15 and 16	SYS	Reserved by the system (2).	

* IN: Input, OUT: Output, SYS: Used by the system.

- Refer to *9.8.4 Function Setting Example and Parameter Details for the Extended MEMOBUS Protocol* on page 9-69 for details on parameters.

■ Parameter List for Non-procedure Protocol

PARAM No.	IN/OUT*	Contents	Description
00	OUT	Processing result	The processing results are output here.
01	OUT	Status	The status of the transmission device is output here.
02	OUT	Remote station number	The number of the source station is output.
03	OUT	(Not used.)	
04	OUT	(Not used.)	
05	OUT	(Not used.)	
06	OUT	Data size	The data size requested by the sending station is output here.
07	OUT	Remote CPU number	
08 to 11	IN	(Not used.)	
12	IN	Register offset	PARAM12 sets the offset word address of the registers.
13	IN	Write range HI	PARAM13 sets the final address of the write range.
14	SYS	Reserved by the system (1).	
15 and 16	SYS	Reserved by the system (2).	

* IN: Input, OUT: Output, SYS: Used by the system.

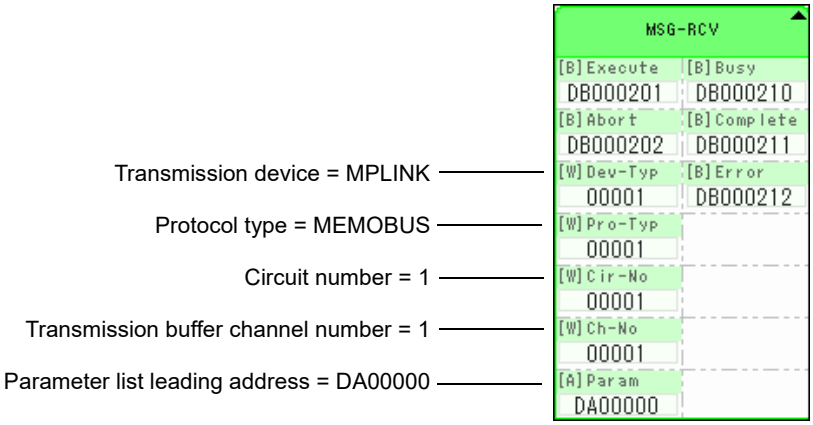
- Refer to *9.8.5 Function Setting Example and Parameter Details for the Non-procedure Protocol* on page 9-75 for details on the parameters.

9.8.4 Function Setting Example and Parameter Details for the Extended MEMOBUS Protocol

This section provides an MSG-RCV function setting example and describes the parameter list in detail for MPLINK/CP-215 transmissions (MPLINK) using the Extended MEMOBUS protocol.

(1) MSG-RCV Function Setting Example

This example shows the settings for using the MPLINK as the transmission device.



To use the Extended MEMOBUS protocol, the protocol type is set to MEMOBUS.
The circuit number is set to the circuit number assigned to the 215AIF-01 transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *9.8.2 Inputs and Outputs for the Message Receive Function* on page 9-64.

(2) Parameter Details for Extended MEMOBUS Protocol

This section describes the parameter list in detail for using the Extended MEMOBUS protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARAM00. The lower-place byte is for system analysis.

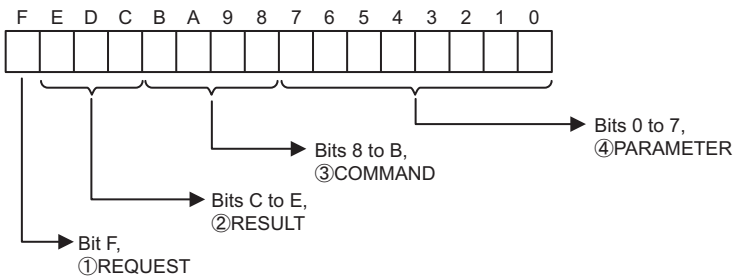
Value of Processing Result	Meaning
00xxH	Processing in progress (Busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	Function code error	An undefined function code was received. Check the function code that was received.
82xxH	Address setting error	One of the following settings is not within the valid range. Check the settings. Data address (in request from sender) PARAM08 (coil offset) PARAM09 (input relay offset) PARAM10 (input register offset) PARAM11 (holding register offset)
83xxH	Data size error	The size of the received data is not within the allowable range. Check the data size from the sending station.
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-RCV function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-RCV function.
86xxH	–	–
87xxH	–	Reserved by the system.
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-RCV function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-RCV function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-RCV function is output to these bits.

Code	Abbreviation	Meaning
1	SEND_OK	Sending has been completed normally.
	SEND_NG	Send error (The processing results (PARAM00) indicate an error.)
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-RCV function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send General-purpose Message with Non-procedure Protocol
2	U_REC	Receive General-purpose Message with Non-procedure Protocol
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response.
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response.
C	MR_SEND	Send MEMOBUS response

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the station number is output.

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Station number outside the allowable range
	02	MEMOBUS response receive monitor time error
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error
Other values of RESULT	XX	Station number

[c] PARAM02: Remote Station Number

The remote station number is output to PARAM02.
When using an MPLINK transmission device, the station number of the sending station is output.
The valid setting ranges are given in the following table.

Transmission Device	Remote Station Number	Remarks
MPLINK/CP-215 (MPLINK)	1 to 64	The number of the sending station is output.

[d] PARAM03: Option

Options specific to each transmission device are output here.
PARAM03 is not used for the Extended MEMOBUS protocol.

[e] PARAM04: Function Code

The function code that was received is output to PARAM04.
The function codes used for the Extended MEMOBUS protocol are listed in the following table.

<Function Codes for Extended MEMOBUS>

Function Code	Applicable Data Type	Function	Setting
00H	–	Not used.	–
01H	B	Read Coil Status	○
02H	B	Read Input Relay Status	○
03H	W	Read Holding Register Contents	○
04H	W	Read Input Register Contents	○
05H	B	Modify Status of a Single Coil	○
06H	W	Write to a Single Holding Register	○
07H	–	Not used.	–
08H	–	Loopback Test	○
09H	W	Read Holding Register Contents (Extended)	○
0AH	W	Read Input Register Contents (Extended)	○
0BH	W	Write to Holding Register (Extended)	○
0CH	–	Not used.	–
0DH	W	Read Discontinuous Holding Register Contents (Extended)	○
0EH	W	Write to Discontinuous Holding Registers (Extended)	○
0FH	B	Modify Status of Multiple Coils	○
10H	W	Write to Multiple Holding Registers	○

- B: Bit data, W: Word data
- ○: Can be set, ×: Cannot be set.
- Only MW (MB) registers may be used for sending and receiving messages when operating as the Master.
- MB, MW, IB, and IW registers can be used as coils, holding registers, input relays, and input registers, respectively, when operating as a Slave.

[f] PARAM05: Data Address

The requested data address is output to PARAM05.

[g] PARAM06: Data Size

The data size requested by the sending station to be read or written as the number of bits or words is output to PARAM06.

[h] PARAM07: Remote CPU Number

The remote CPU number will be output as 1 if the remote device is an MP2000 Series Machine Controller.
If the remote device is a Yaskawa Controller, but not in the MP2000 Series and it consists of more than one CPU Module, the destination CPU number will be output.
In all other cases, 1 will be output.
PARAM07 is not used for the Extended MEMOBUS protocol. A value of 0 will be output.

[i] PARAM08, PARAM09, PARAM10, and PARAM11: Offsets

These parameters set the data address offsets from the sending station.
The position of the data is shifted backward by the number of words set for the offset at the device receiving the message.

- For details, refer to *A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV* on page A-5.
- Negative offsets cannot be set.

There is an offset parameter for each type of data.
The following table describes the offset parameters.

<Offset Parameters>

Parameter	Contents	Description
PARAM08	Coil offset	Sets the offset word address of the coils.
PARAM09	Input relay offset	Sets the offset word address of the input relays.
PARAM10	Input register offset	Sets the offset word address of the input registers.
PARAM11	Holding register offset	Sets the offset word address of the holding registers.

The offset parameter that is valid depends on the function code as shown below.
The following table lists the valid offset parameter for each function code.

<Offset Parameter for Each Function Code>

Function Code	Function	Valid Offset Parameter	Setting
01H	Read Coil Status	PARAM08	○
02H	Read Input Relay Status	PARAM09	○
03H	Read Holding Register Contents	PARAM11	○
04H	Read Input Register Contents	PARAM10	○
05H	Modify Status of a Single Coil	PARAM08	○
06H	Write to a Single Holding Register	PARAM11	○
09H	Read Holding Register Contents (Extended)	PARAM11	○
0AH	Read Input Register Contents (Extended)	PARAM10	○
0BH	Write to Holding Register (Extended)	PARAM11	○
0DH	Read Discontinuous Holding Register Contents (Extended)	PARAM11	○
0EH	Write to Discontinuous Holding Registers (Extended)	PARAM11	○
0FH	Modify Status of Multiple Coils	PARAM08	○
10H	Write to Multiple Holding Registers	PARAM11	○

- ○: Can be set. ×: Cannot be set.

[j] PARAM12 and PARAM13: Write Range

These parameters set the range of addresses that can be written for a write requested from the sending station. Any write request that exceeds this address range will result in an error.
The write ranges in PARAM12 and PARAM13 are set as word addresses.

- The MP3000/MP2000 Series Machine Controllers store all data for write requested from the sending station to M registers.

The write range parameter can be used to set a range of M registers that can be written using messages.
The write range parameters are listed in the following table.

Parameter	Contents	Description
PARAM12	Write range LO	Leading address of the write range
PARAM13	Write range HI	Final address of the write range

Set the write ranges so that the following conditions are satisfied.

$0 \leq$ Write range LO $\leq$ Write range HI $\leq$ Maximum value of M addresses

The write range is valid for the following function codes.

05H (Modify Status of a Single Coil)
06H (Write to a Single Holding Register)
0BH (Write to Holding Register (Extended))
0EH (Write to Discontinuous Holding Registers (Extended))
0FH (Modify Status of Multiple Coils)
10H (Write to Multiple Holding Registers)

Example: The following settings would be used to permit writing to the M registers with addresses 1000 to 1999.

PARAM12=1000
PARAM13=1999

With these settings, the receiving station will output an error and not perform the write if a write request is received for any registers not between MW01000 and MW01999.

[k] PARAM14: Reserved by the System (1)

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[l] PARAM15 and PARAM16: Reserved by the System (2)

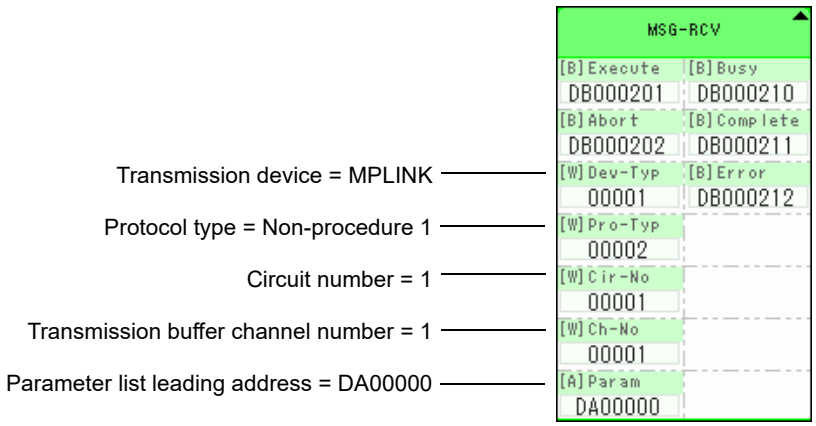
These parameters are used by the system. Do not change the values set for these parameters.

9.8.5 Function Setting Example and Parameter Details for the Non-procedure Protocol

With the non-procedure protocol, the received data is not converted and stored as is in the M registers. This enables receiving data using any protocol of the remote station.
This section provides an MSG-RCV function setting example and describes the parameter list in detail for MPLINK/CP-215 transmissions (MPLINK) using the non-procedure protocol.

(1) MSG-RCV Function Setting Example

This example shows the settings for using the MPLINK as the transmission device.



The protocol type (Pro-Typ) is set to 2 to use non-procedure protocol 1 and to 3 to use non-procedure protocol 2.
The circuit number is set to the circuit number assigned to the 215AIF-01 transmission device.
Transmission buffer channel numbers in the same line must all be unique.
For details on settings, refer to *9.8.2 Inputs and Outputs for the Message Receive Function* on page 9-64.

(2) Parameter Details for Non-procedure Protocol

This section describes the parameter list in detail for using the non-procedure protocol.

[a] PARAM00: Processing Result

The processing result is output to the upper-place byte of PARAM00. The lower-place byte is for system analysis.

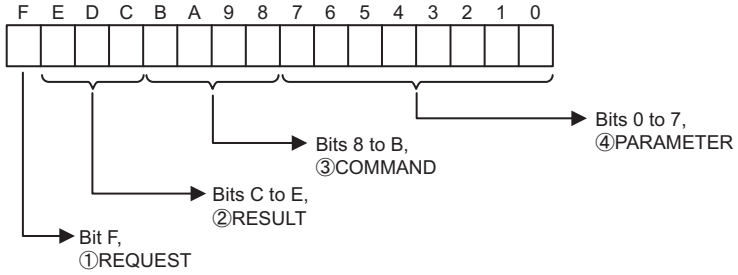
Value of Processing Result	Meaning
00xxH	Processing in progress (Busy)
10xxH	Processing completed (Complete)
8yxxH	Error occurred (Error)

If an error occurs, troubleshoot the problem according to the value of the processing result as listed below.

Error	Error Contents	Description
80xxH	–	Reserved by the system.
81xxH	–	–
82xxH	Address setting error	The following setting is not within the valid range. Check the settings. PARAM11 (holding register offset)
83xxH	Data size error	The size of the received data is not within the allowable range. Check the data size from the sending station.
84xxH	Circuit number setting error	The circuit number is not within the allowable range. Check the circuit number in the MSG-RCV function.
85xxH	Channel number setting error	The transmission buffer channel number is not within the allowable range. Check the transmission buffer channel number in the MSG-RCV function.
86xxH	–	–
87xxH	–	Reserved by the system.
88xxH	Transmission device error	An error response was returned from the transmission device. Check the connection to the equipment. Also, be sure that the remote device can communicate.
89xxH	Device selection error	An unavailable device was set. Check the transmission device type in the MSG-RCV function.

[b] PARAM01: Status

The status of the transmission device is output to PARAM01.
Bit assignments are shown in the following figure and described after the figure.



① REQUEST

The status of the processing request for the MSG-RCV function is output to this bit.

Bit Status	Meaning
1	Processing is being requested.
0	Processing request has been accepted.

② RESULT

The result of executing MSG-RCV function is output to these bits.

Code	Abbreviation	Meaning
1	SEND_OK	Sending has been completed normally.
	SEND_NG	Send error (The processing results (PARAM00) indicate an error.)
2	REC_OK	Receiving has been completed normally.
3	ABORT_OK	Abort completed.
4	FMT_NG	Parameter format error
5	SEQ_NG	Command sequence error
6	RESET_NG	Reset status
7	REC_NG	Data receive error (Error detected by a lower-layer program.)

③ COMMAND

The processing command for the MSG-RCV function is output to these bits.

Code (Hex)	Abbreviation	Meaning
1	U_SEND	Send general-purpose message
2	U_REC	Receive general-purpose message
3	ABORT	Abort
8	M_SEND	Send MEMOBUS command; executing the command is completed upon receiving a response. (For MEMOBUS Protocol)
9	M_REC	Receive MEMOBUS command; executing the command is followed by sending a response. (For MEMOBUS Protocol)
C	MR_SEND	Send MEMOBUS response (For MEMOBUS Protocol)

④ PARAMETER

The following error code is output if RESULT is set to 4 (FMT_NG: parameter format error). In other cases, the station number is output.

RESULT	Code	Meaning
RESULT = 4 (FMT_NG: parameter format error)	00	No errors
	01	Station number outside the allowable range
	02	MEMOBUS response receive monitor time error (for MEMOBUS Protocol)
	03	Retry count setting error
	04	Cyclic area setting error
	05	CPU No. error
	06	Data address error
	07	Data size error
	08	Function code error (for MEMOBUS Protocol)
Other values of RESULT	XX	Station Number

[c] PARAM02: Remote Station Number

Refer to *9.8.4 (2) [c] PARAM02: Remote Station Number* on page 9-72.

[d] PARAM06: Data Size

The size of received data is output to this parameter.
In the non-procedure 1 communication mode, the data size is output as the number of words.
In the non-procedure 2 communication mode, the data size is output as the number of bytes.

[e] PARAM12: Register Offset

This parameter sets the data address offsets from the sending station.
The position of the data is shifted backward by the number of words set for the offset at the device receiving the message.

- For details, refer to *A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV* on page A-5.
- Negative offsets cannot be set.

With the non-procedure protocol, the continuous received data is stored in M registers. The register offset can be set to specify the leading address of the M registers in which to store the data.

Example: With the following setting, the received data would be stored in M registers starting with MW01000.

PARAM12 = 1000

[f] PARAM13: Write Range HI

This parameter sets the range of addresses that can be written for a write requested from the sending station. Any write request that exceeds this address range will result in an error.

The write range HI value in PARAM13 is set as a word address.

The write range parameter can be used to set a range of M registers that can be written using messages.

Set the write ranges so that the following conditions are satisfied.

$0 \leq$ Write range HI $\leq$ Maximum value of M addresses

Example: The following settings would be used to permit writing to the M registers with addresses 1000 to 1999.

PARAM12 = 1000
PARAM13 = 1999

With these settings, the receiving station will output an error and not perform the write if a write request is received for any registers not between MW01000 and MW01999.

[g] PARAM14: Reserved by the System (1)

PARAM14 is used by the system. (The current transmission buffer channel number is held here.)

- Make sure that the user program sets this parameter to 0 in the first scan after the power is turned ON. Thereafter, do not change the value set for this parameter because the system will use it.

[h] PARAM15 and PARAM16: Reserved by the System (2)

These parameters are used by the system. Do not change the values set for these parameters.

Appendices

Appendix A Supplementary Information on Message Send and Receive Functions ----------A-2
A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND ---------- A-2
A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV ---------- A-5
A.3 Transmission Buffer Channels ---------- A-9

Appendix B Details on the Extended MEMOBUS Protocol ----------A-14
B.1 Message Configuration ---------- A-14
B.2 MEMOBUS Binary Mode ---------- A-16
B.3 MEMOBUS ASCII Mode ---------- A-24
B.4 General-purpose Messages in Binary Mode ---------- A-24
B.5 General-purpose Messages in ASCII Mode ---------- A-25

Appendix C Details on the MEMOBUS Protocol ----------A-26
C.1 Message Configuration ---------- A-26
C.2 MEMOBUS RTU Mode ---------- A-27
C.3 MEMOBUS ASCII Mode ---------- A-32

Appendix D Details on Non-procedure Protocol ----------A-33
D.1 Message Configuration ---------- A-33
D.2 General-purpose Messages in Binary Mode ---------- A-34
D.3 General-purpose Messages in ASCII Mode ---------- A-34

Appendix E Sample C Language Programs for the Master ----------A-35
E.1 TCP (When Using Extended MEMOBUS Protocol (SFC = 09)) ---------- A-35
E.2 UDP (When Using Extended MEMOBUS Protocol (SFC = 09)) ---------- A-39

Appendix F Sample C Language Programs for a Slave ----------A-41
F.1 TCP (When Using Extended MEMOBUS Protocol) ---------- A-41
F.2 UDP (When Using Extended MEMOBUS Protocol) ---------- A-46

Appendix G Junction Box Connection Example for CP-215 Communications ----------A-48
G.1 Connections with the 215AIF-01 Module ---------- A-48
G.2 Connection to a Personal Computer ---------- A-49

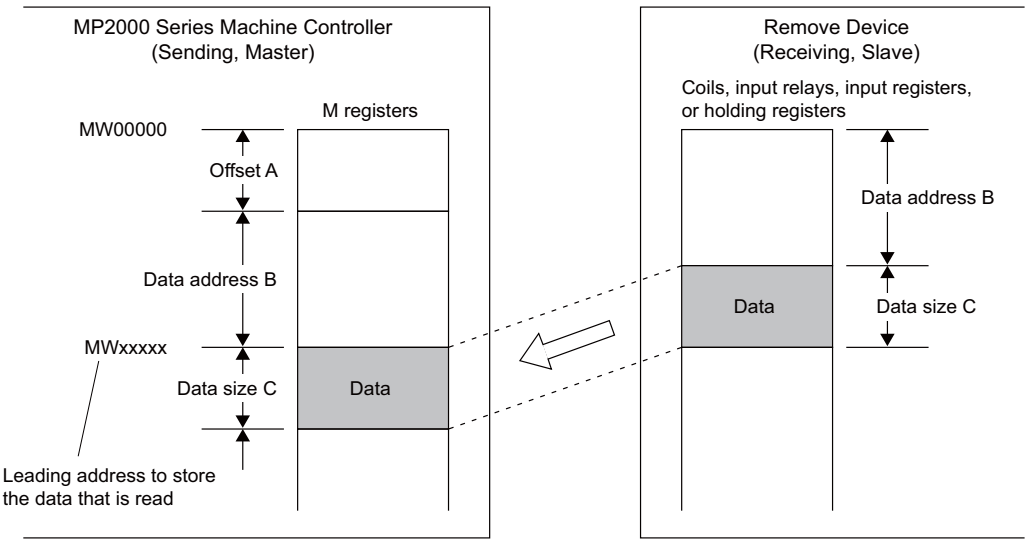
Appendix A Supplementary Information on Message Send and Receive Functions

A.1 Relationship among the Data Address, Data Size, and Offset in MSG-SND

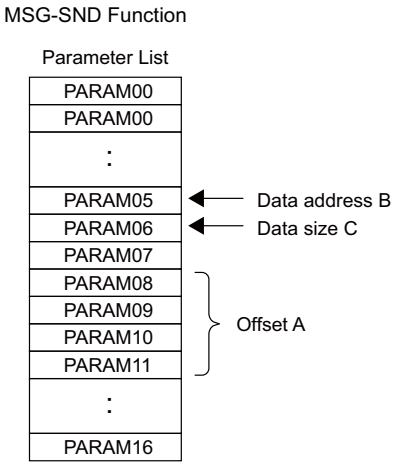
The relationship between the data address, data size, and offset if an offset is used when sending a message is described in the following sections.

(1) Reading Data

■ Data Flow When Reading



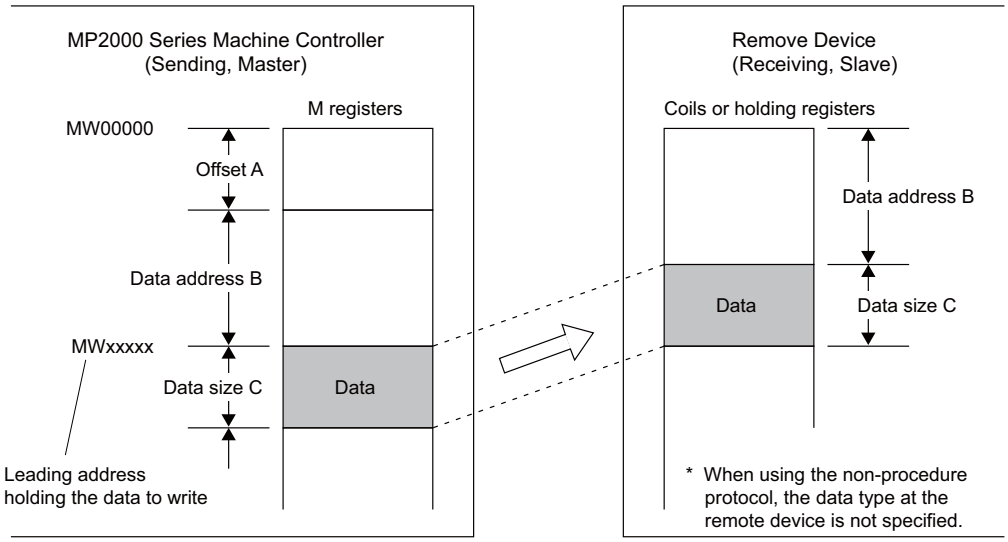
■ Parameter Settings



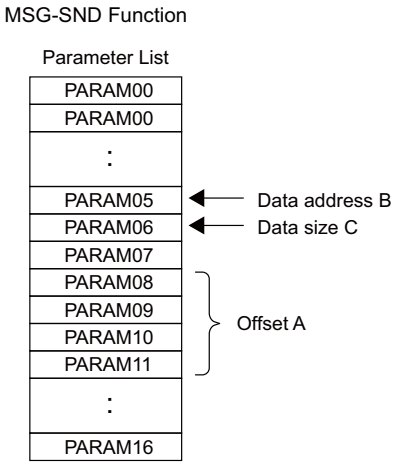
- The parameters that are valid depend on the function code and communication protocol that are being used.

(2) Writing Data

■ Data Flow When Writing



■ Parameter Settings

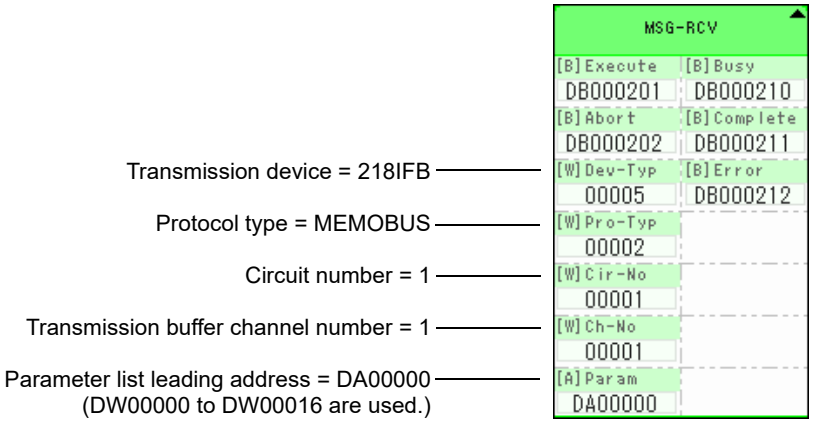


- The parameters that are valid depend on the function code and communication protocol that are being used.

(3) Setting Example When Reading Status Using an Offset

The relationship between the settings and the remote device data when the Read Coil Status message is sent with an offset using the MEMOBUS protocol is described in this section.

■ Entering the MSG-SND Function in the Ladder Program



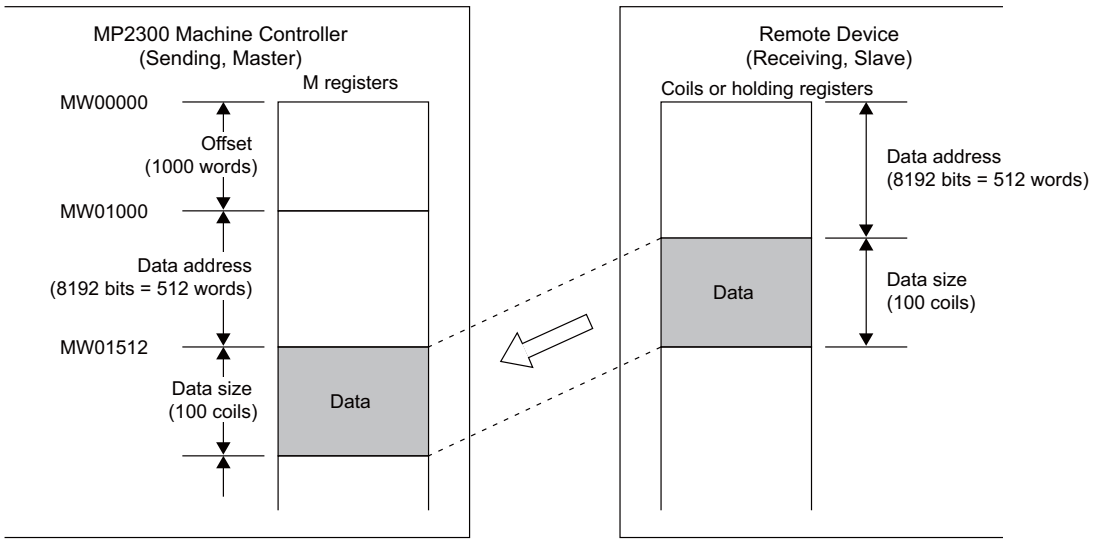
■ Parameter List Settings for the MSG-SND Function

Register No.	Set Value	Parameter No.	IN/OUT	Remarks
DW00000	–	00	OUT	Processing result
DW00001	–	01	OUT	Status
DW00002	00001	02	IN	Connection number = 1
DW00003	–	03	IN	Option (Does not need to be set.)
DW00004	00001	04	IN	Function code = 1 (Read Coil Status)
DW00005	08192	05	IN	Data address = 8192 bits (512 words)
DW00006	00100	06	IN	Data size = 100 coils
DW00007	00001	07	IN	Remote CPU number = 0
DW00008	01000	08	IN	Coil offset = 1000 words
DW00009	00000	09	IN	Input relay offset = 0 words
DW00010	00000	10	IN	Input register offset = 0 words
DW00011	00000	11	IN	Holding register offset = 0 words
DW00012	–	12	SYS	Reserved by the system (cleared to 0 when function execution is started).
DW00013	–	13	SYS	Reserved by the system.
DW00014	–	14	SYS	Reserved by the system.
DW00015	–	15	SYS	Reserved by the system.
DW00016	–	16	SYS	Reserved by the system.

- IN: Input, OUT: Output, SYS: Used by the system.

■ Relationship between Remote Device and Data

The flow of data when the Read Coil Status message is sent with an offset is shown in the following diagram.
If communication is performed normally in the example shown in the following diagram, the status of the coils in the remote device will be stored in the MP2300 starting at MW01512.

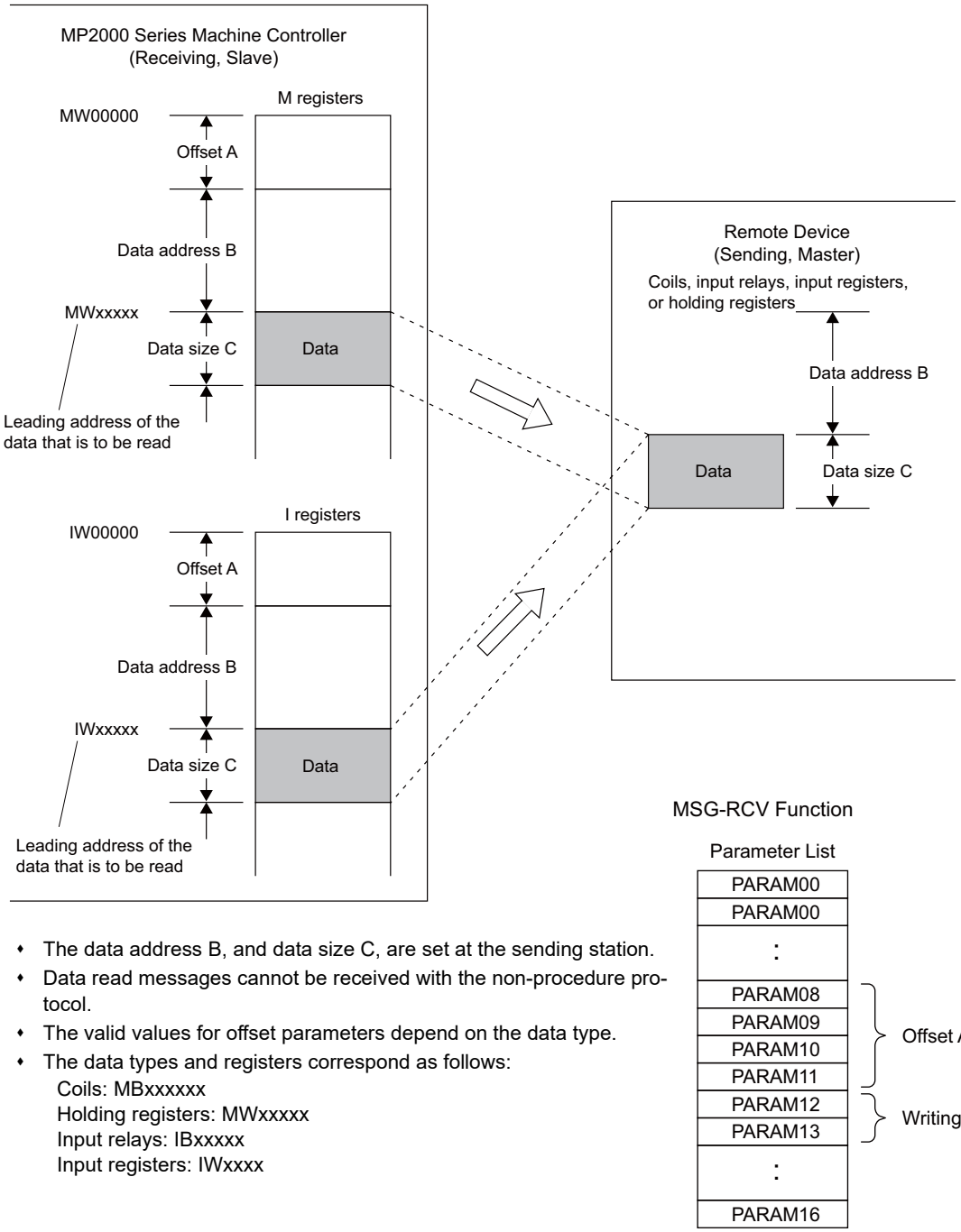


A.2 Relationship among the Data Address, Data Size, and Offset in MSG-RCV

The relationship between the data address, data size, and offset if an offset is used when receiving a message is described in the following sections.

(1) Reading Data

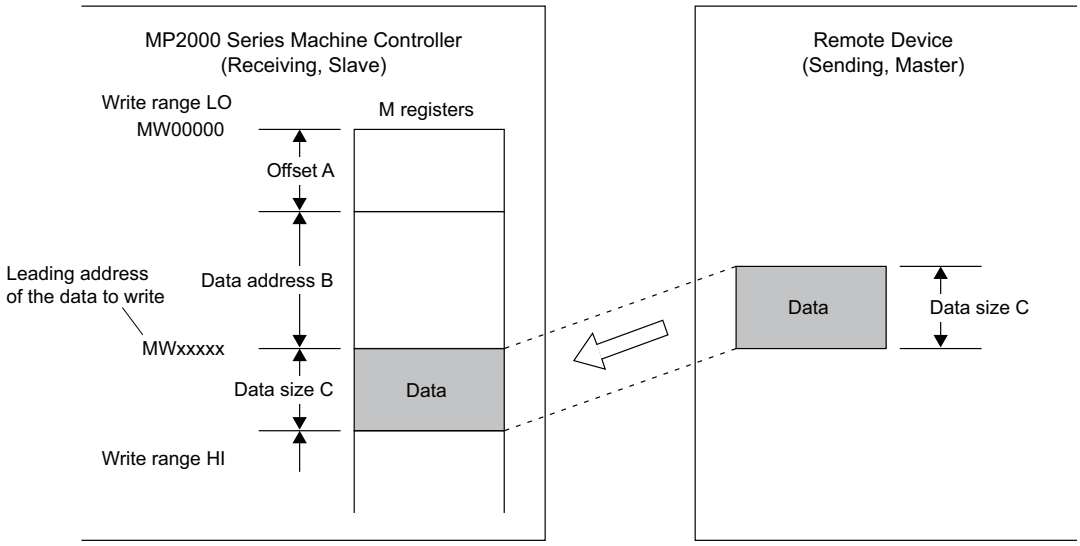
■ Data Flow and Parameter Settings When Reading



- The data address B, and data size C, are set at the sending station.
- Data read messages cannot be received with the non-procedure protocol.
- The valid values for offset parameters depend on the data type.
- The data types and registers correspond as follows:
  Coils: MBxxxxxx
  Holding registers: MWxxxxx
  Input relays: IBxxxxx
  Input registers: IWxxxx

(2) Writing Data

■ Data Flow When Writing



- The data address B, and data size C, are set at the sending station.
- Data read messages cannot be received with the non-procedure protocol.
- The valid values for offset parameters depend on the data type.
- The data types and registers correspond as follows:
  Coils: MBxxxxxx
  Holding registers: MWxxxxx

■ Parameter Settings

<Communication protocol = MEMOBUS>

MSG-RCV Function

Parameter List

PARAM00	
PARAM00	
:	
PARAM08	Offset A
PARAM09	
PARAM10	
PARAM11	
PARAM12	Writing range
PARAM13	
:	
PARAM16	

<Communication protocol = Non-procedure>

MSG-RCV Function

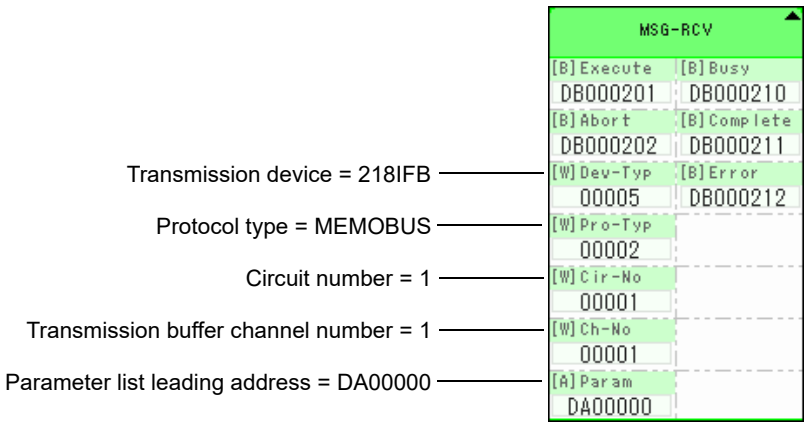
Parameter List

PARAM00	
PARAM00	
:	
PARAM08	
PARAM09	
PARAM10	
PARAM11	← Offset A
PARAM12	Writing range
PARAM13	
:	
PARAM16	

(3) Example for Receiving the Write to Multiple Holding Registers Message Using an Offset

The relationship between the settings and the remote device data when the Write to Multiple Holding Registers message is received with an offset using the MEMOBUS protocol is described in this section.

■ Entering the MSG-RCV Function in the Ladder Program



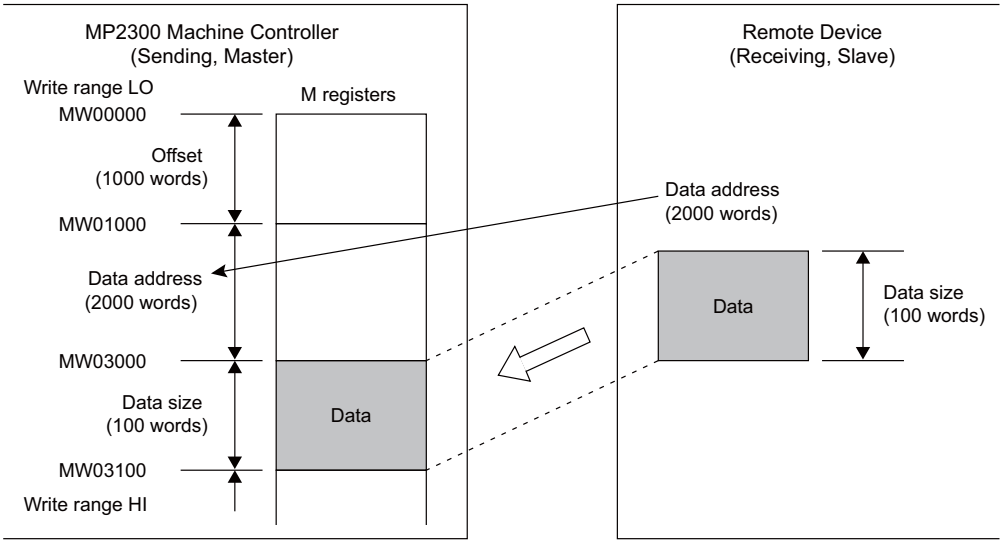
■ Parameter List Settings for the MSG-RCV Function

Register No.	Set Value	Parameter No.	IN/OUT	Remarks
DW00000	–	00	OUT	Processing result
DW00001	–	01	OUT	Status
DW00002	00001	02	IN	Connection number = 1
DW00003	–	03	OUT	Option (Does not need to be set.)
DW00004	–	04	OUT	Function code
DW00005	–	05	OUT	Data address
DW00006	–	06	OUT	Data size
DW00007	–	07	OUT	Remote CPU number
DW00008	00000	08	IN	Coil offset = 0 words
DW00009	00000	09	IN	Input relay offset = 0 words
DW00010	00000	10	IN	Input register offset = 0 words
DW00011	01000	11	IN	Holding register offset = 1000 words
DW00012	00000	12	IN	Write range LO = 0
DW00013	65534	13	IN	Write range HI = 65534
DW00014	–	14	SYS	Reserved by the system (cleared to 0 when function execution is started).
DW00015	–	15	SYS	Reserved by the system.
DW00016	–	16	SYS	Reserved by the system.

- IN: Input, OUT: Output, SYS: Used by the system.

■ Relationship between Remote Device and Data

The flow of data when the Write to Multiple Holding Registers message is received with an offset is shown in the following diagram.
If communication is performed normally in the example shown in the following diagram, the data from the remote device will be stored in the MP2300 starting at MW03000.



A.3 Transmission Buffer Channels

The transmission buffer channels are data buffers used to transfer data between the MSG-SND and MSG-RCV functions and the transmission devices. These data buffers consist of one or more channels. The channels are differentiated using transmission buffer channel numbers.

(1) 218IF/218IFB Transmission Buffer Channels

The relationship between a transmission buffer channel and connection is defined by the channel number (i.e., the transmission buffer channel number) input item and PARAM02 (connection number) in the parameter list of the MSG-SND or MSG-RCV function.
A connection is the setting information required for communication between the local and remote stations, and it is defined in the **Transmission Parameters** Tab Page of the **MPE720 Module Configuration** Window (218IF/218IFB).

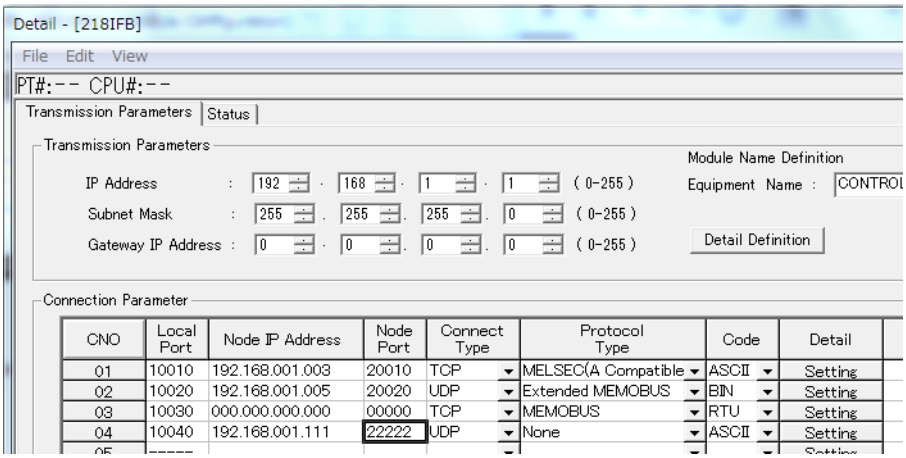


Fig. A.1 **The 218IF Transmission Parameters** Tab Page of the **MPE720 Module Configuration** Window

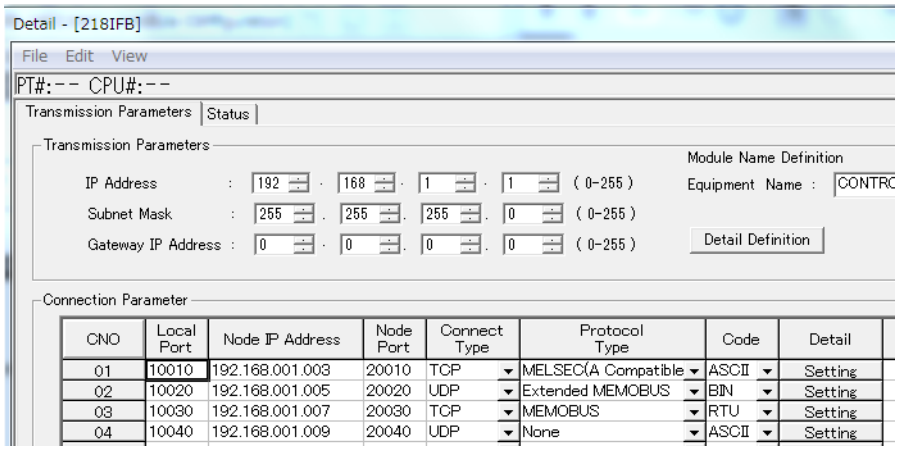
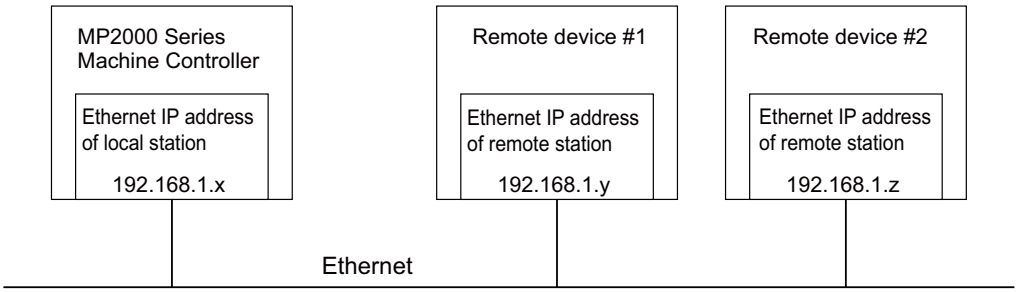


Fig. A.2 **The 218IFB Transmission Parameters** Tab Page of the **MPE720 Module Configuration** Window

A conceptual diagram of the transmission buffer channels is provided on the next page.



Network Configuration Diagram

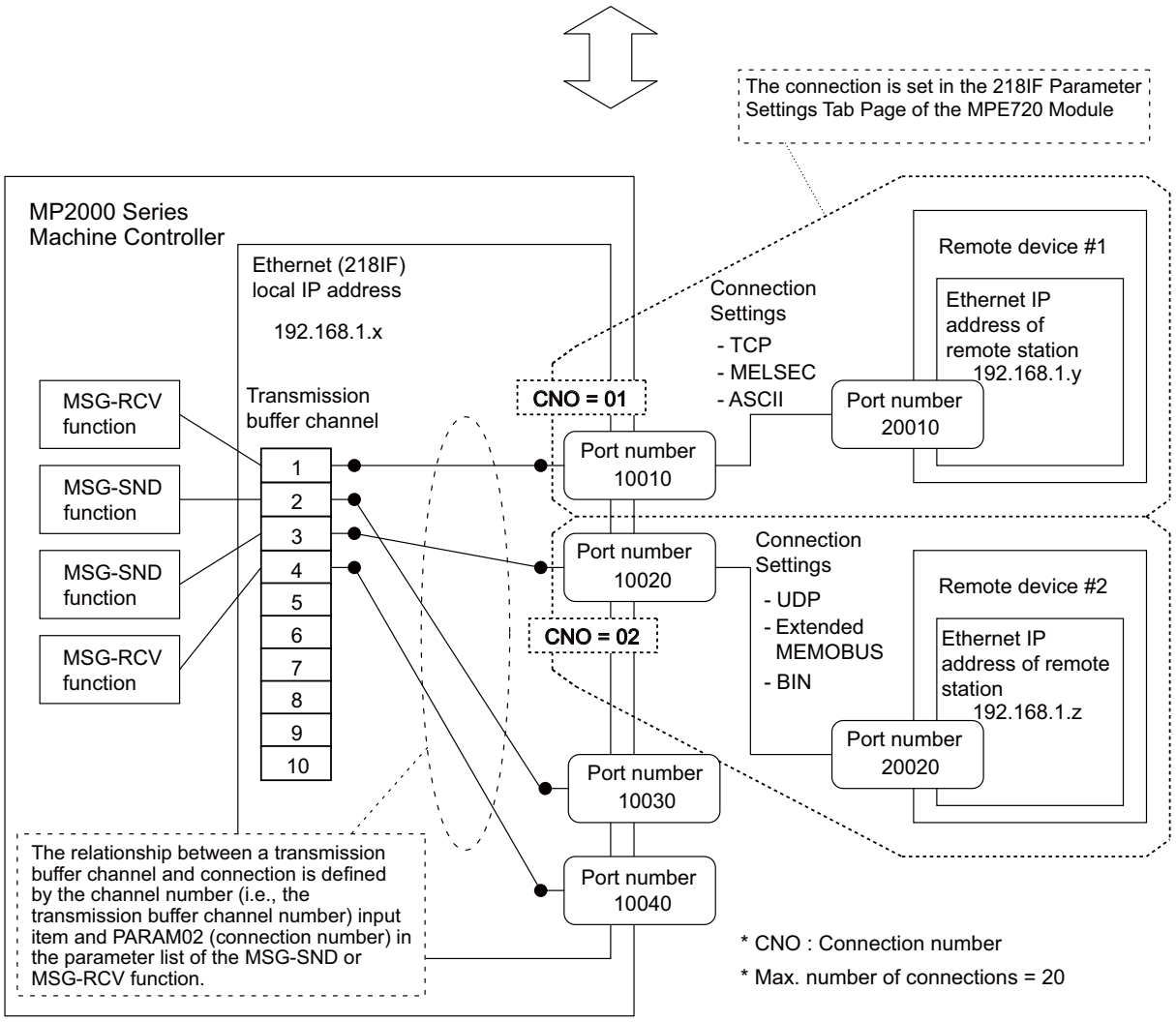
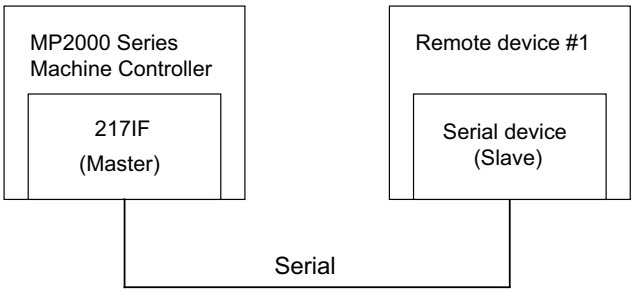
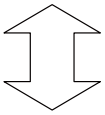


Fig. A.3 218IF/218IFB Transmission Buffer Channels

(2) 217IF Transmission Buffer Channels



Network Configuration Diagram



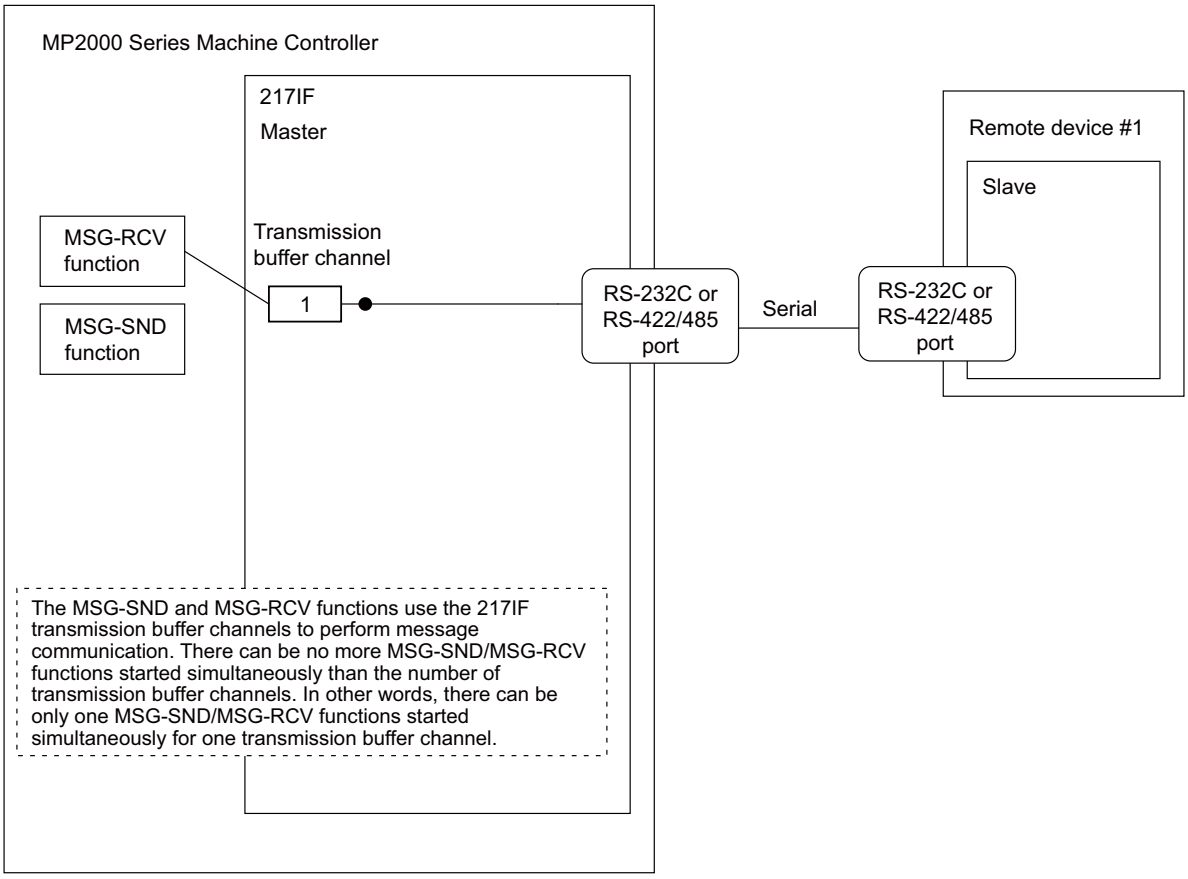
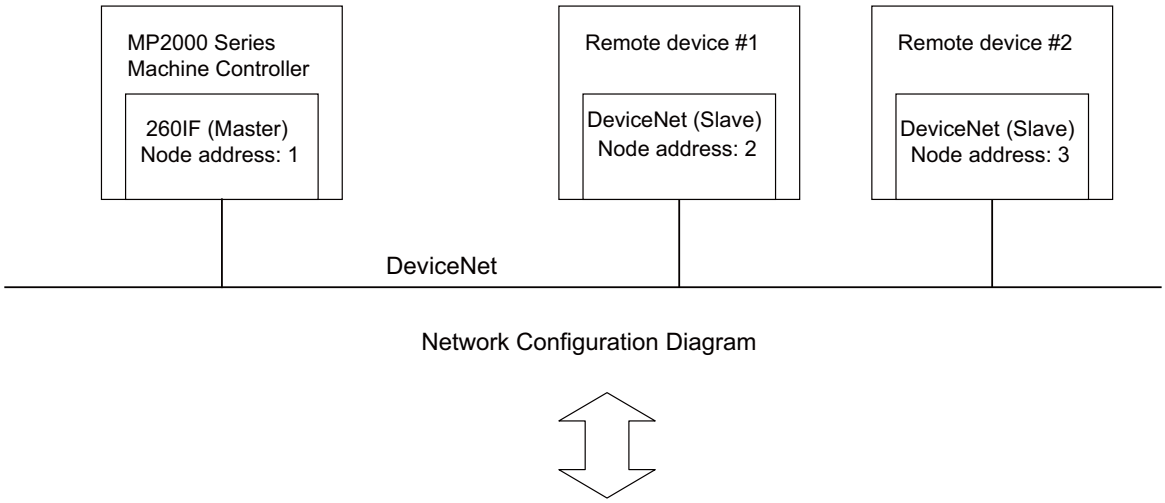


Fig. A.4 217IF Transmission Buffer Channels

(3) 260IF Transmission Buffer Channels



Network Configuration Diagram

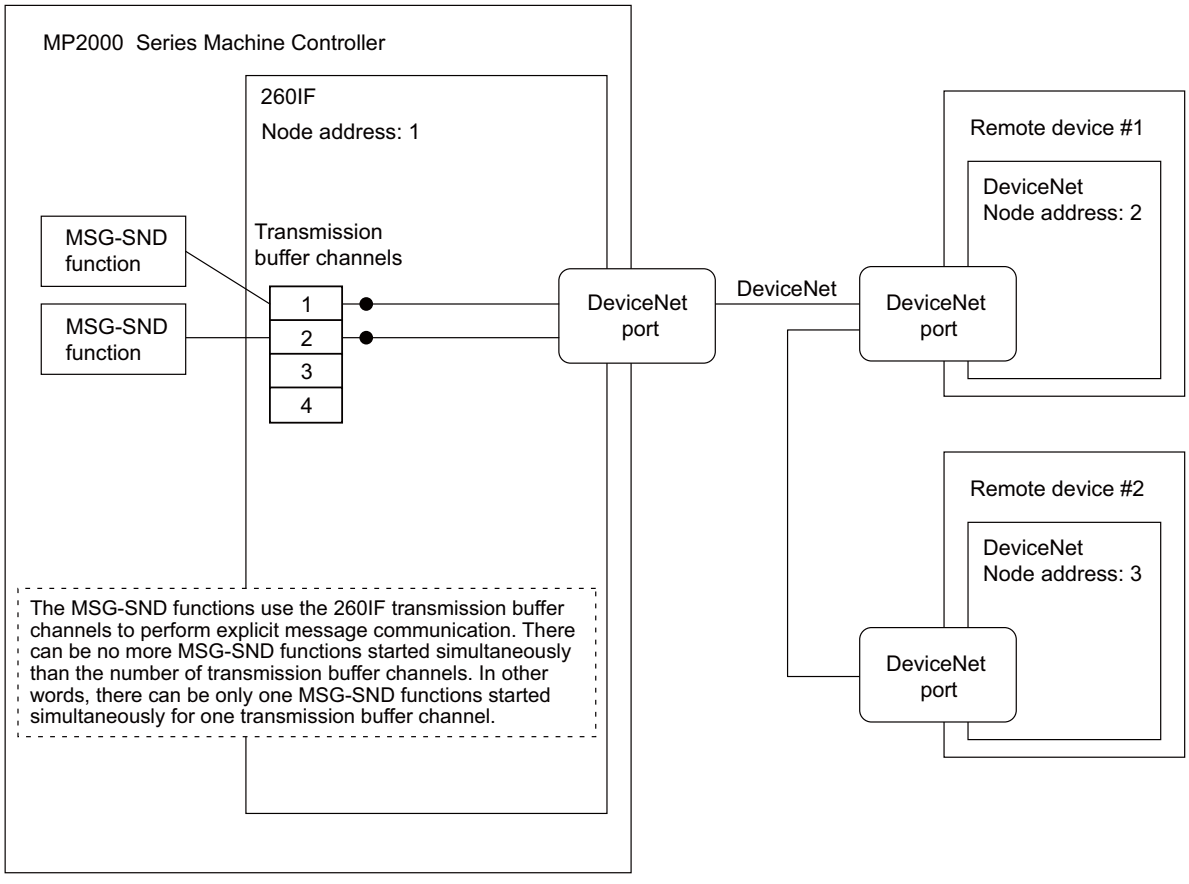
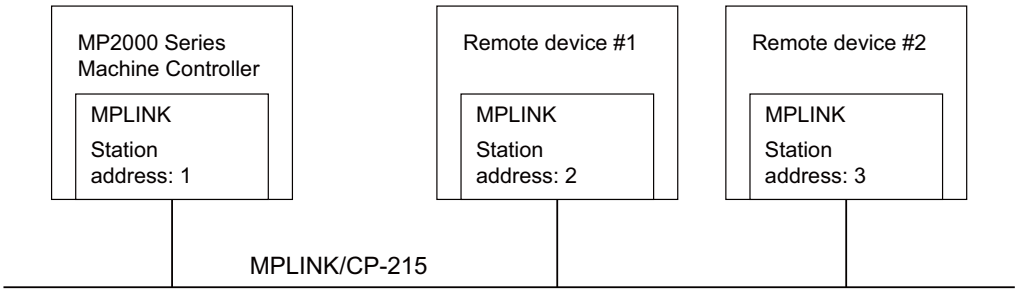


Fig. A.5 260IF Transmission Buffer Channels

(4) MPLINK Transmission Buffer Channels



Network Configuration Diagram

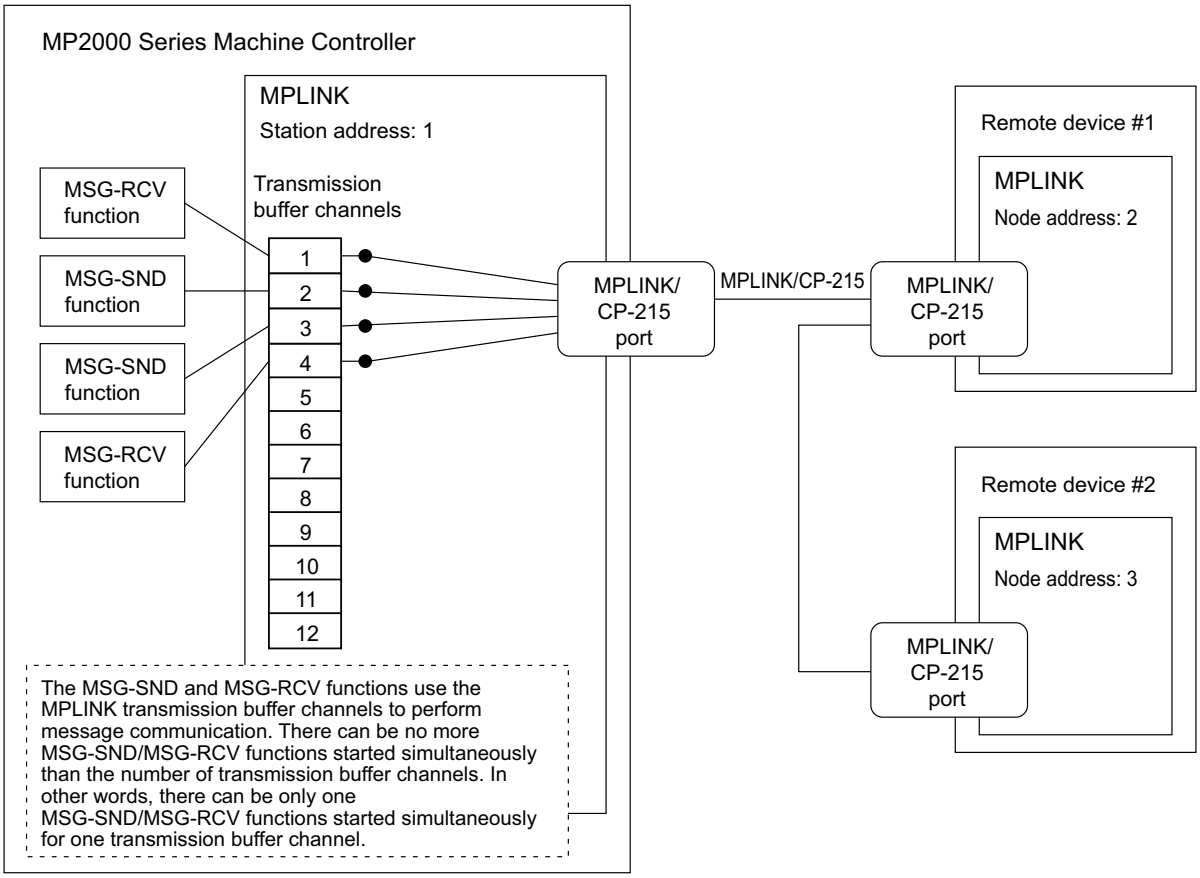
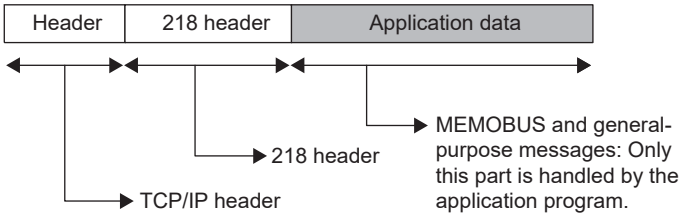


Fig. A.6 MPLINK Transmission Buffer Channels

Appendix B Details on the Extended MEMOBUS Protocol

B.1 Message Configuration

The message configuration used in transmissions in the Ethernet transmission system is shown below. Use this format when creating applications on a personal computer.
If the Extended MEMOBUS protocol is used to send and receive data, each message consists of three elements: A header, a 218 header, and application data.

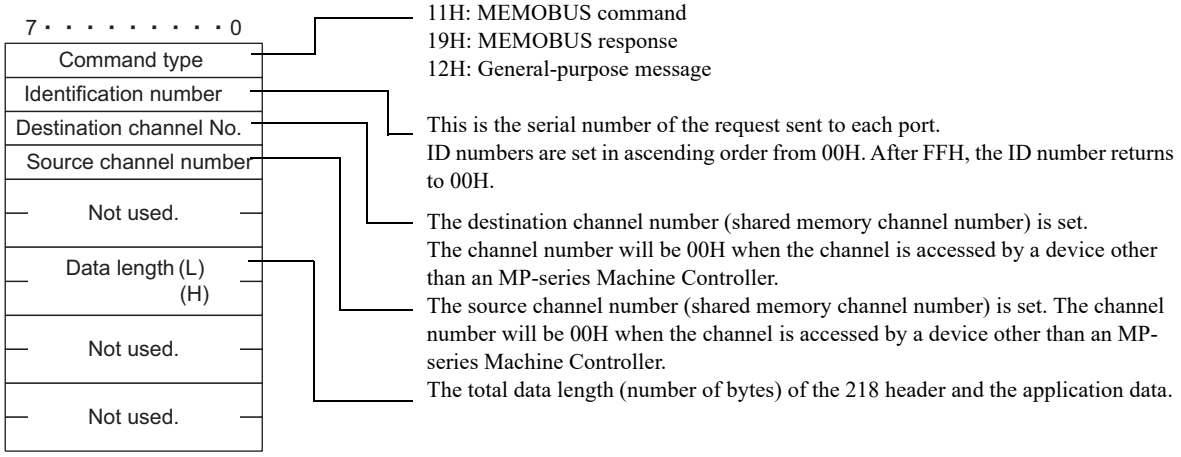


The header is used for TCP/IP and UDP/IP. When creating user programs, it is not necessary to be aware of the header because it is automatically set and deleted in the 218IF/218IFB transmission system.
The 218 header is required when using the Extended MEMOBUS protocol in Ethernet communication. When creating application programs, it is not necessary to be aware of the 218 header because it is automatically set and deleted in the 218IF/218IFB transmission system.
The actual data for the Extended MEMOBUS protocol is stored in the application data field.
When performing communication between a 218IF-01 or 218IF-02 Module and a host computer, the application program in the host computer must set and delete the 218 header.
The application data has the parameter structure shown below according to the communication protocol and the code to be used.

Communication Protocol	Code	Reference
MEMOBUS messages	BIN	*B.2 MEMOBUS Binary Mode* on page A-16
MEMOBUS messages	ASCII	*B.3 MEMOBUS ASCII Mode* on page A-24
General-purpose messages (Non-procedure)	BIN	*B.4 General-purpose Messages in Binary Mode* on page A-24
General-purpose messages (Non-procedure)	ASCII	*B.5 General-purpose Messages in ASCII Mode* on page A-25

(1) The 218 Header

In communication controlled by the Extended MEMOBUS protocol, a 12-byte 218 header is added in front of the application data. The structure of the 218 header is shown below.



(2) Extended MEMOBUS Commands

The commands that make up the Extended MEMOBUS messages are identified by function codes and have the functions shown in the following table.

Function Code	Meaning	Extended MEMOBUS Protocol*	MEMOBUS Protocol*
00H	Not used.	×	×
01H	Read Coil Status	○	○
02H	Read Input Relay Status	○	○
03H	Read Holding Register Contents	○	○
04H	Read Input Register Contents	○	○
05H	Modify Status of a Single Coil	○	○
06H	Write to a Single Holding Register	○	○
07H	Not used.	×	×
08H	Loopback Test	○	○
09H	Read Holding Register Contents (Extended)	○	×
0AH	Read Input Register Contents (Extended)	○	×
0BH	Write to Holding Register (Extended)	○	×
0CH	Not used.	×	×
0DH	Read Discontinuous Holding Register Contents (Extended)	○	×
0EH	Write to Discontinuous Holding Registers (Extended)	○	×
0FH	Modify Status of Multiple Coils	○	○
10H	Write to Multiple Holding Registers	○	○
11H to 20H	Not used.	×	×
21H to 30H	Reserved by the system.	×	×
31H	MELSEC fixed buffer communication	○	×
32H	Read from MELSEC Random-access Buffers	○	×
33H	Write to MELSEC Random-access Buffers	○	×
34H to 3FH	Reserved by the system.	×	×
40H to 4FH	Reserved by the system.	×	×
50H and higher	Not used.	×	×

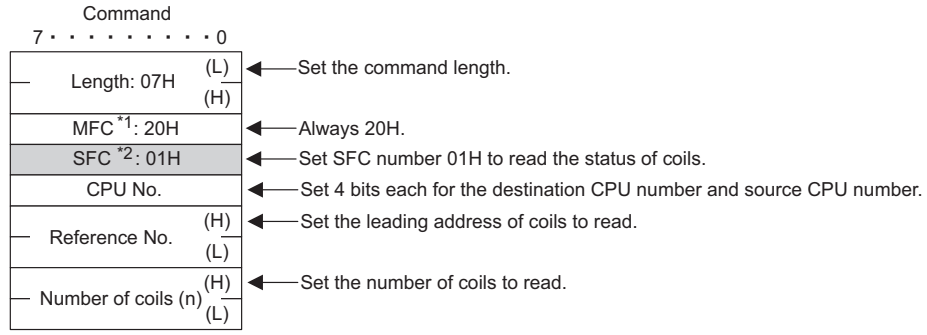
* ○: Can be set. , ×: Cannot be set.

- Only MW (MB) registers may be used for sending and receiving messages when operating as the Master.
- MB, MW, IB, and IW registers can be used as coils, holding registers, input relays, and input registers, respectively, when operating as a Slave.

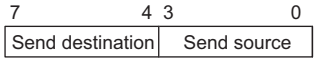
B.2 MEMOBUS Binary Mode

The binary mode formats for MEMOBUS messages are shown below.

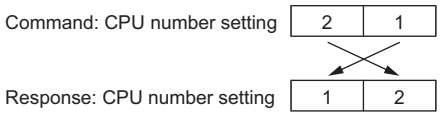
(1) Read Coil Status

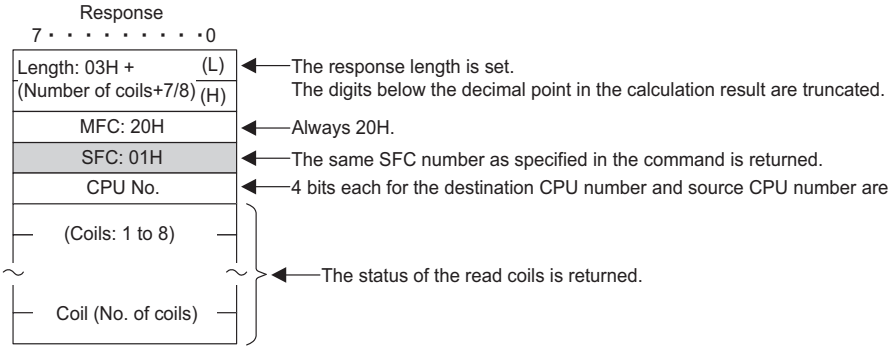


* 1. MFC: Major Function Code
* 2. SFC: Subfunction Code
* 3. The following is the arrangement for CPU numbers.

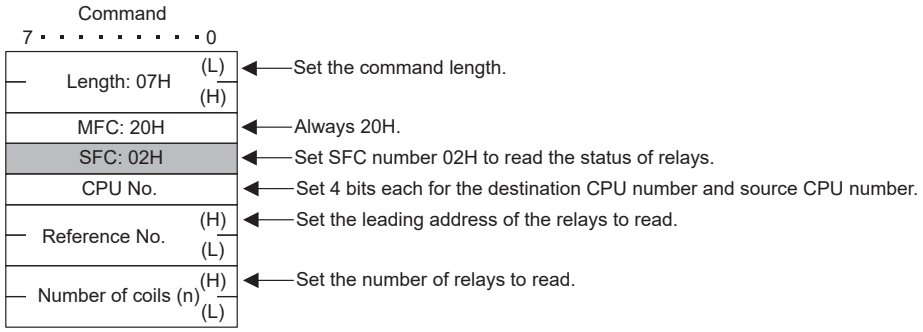


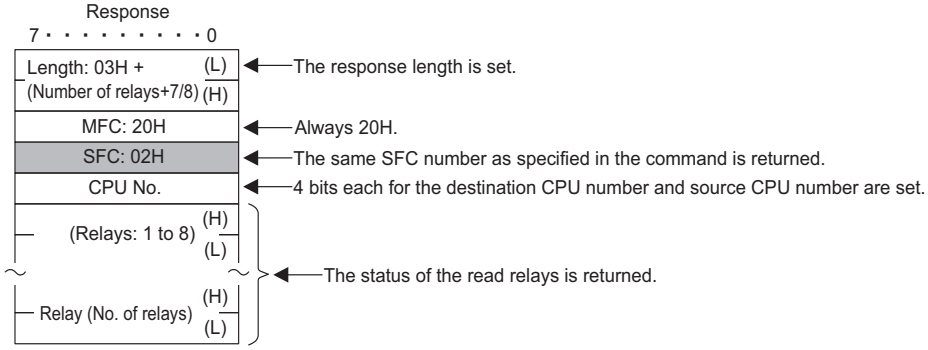
<Example>
Sending a command from send source CPU number 1 to send destination CPU number 2:



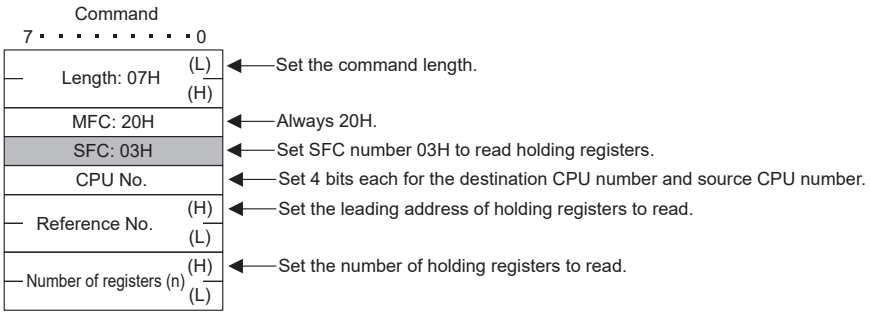


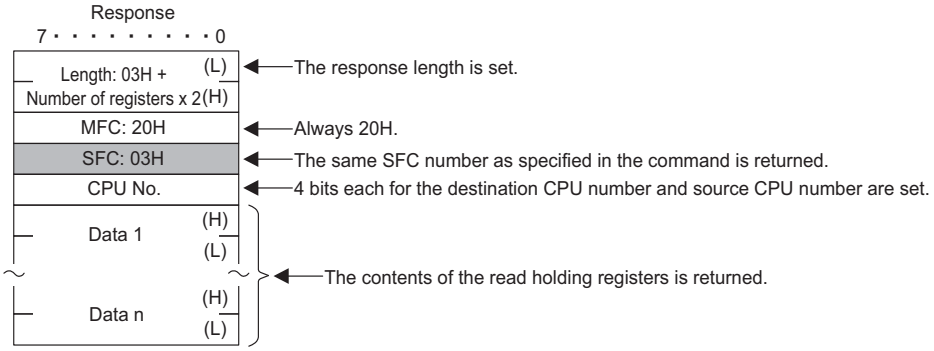
(2) Read Input Relay Status



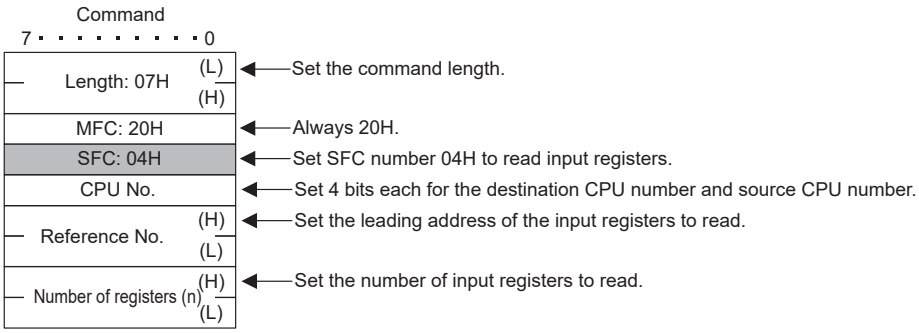


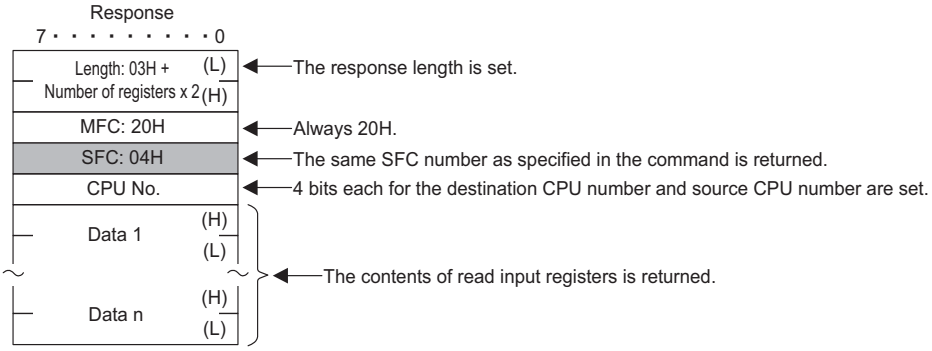
(3) Read Holding Register Contents





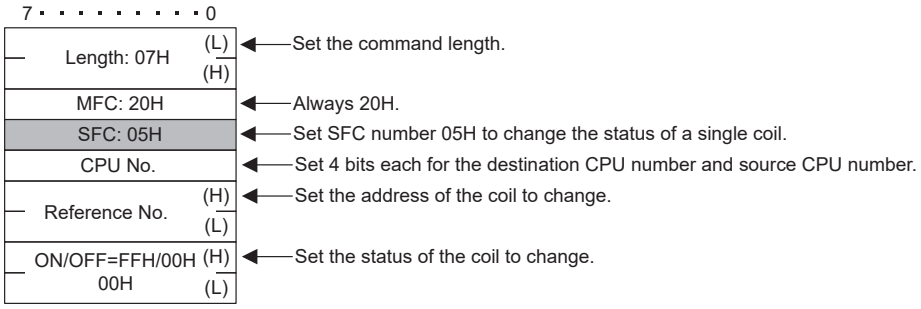
(4) Read Input Register Contents



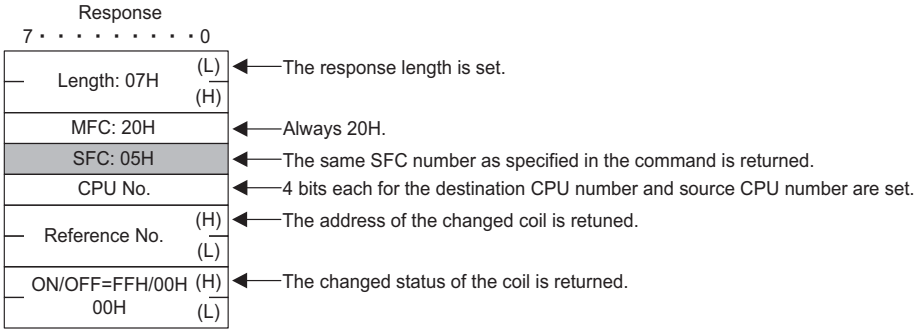


(5) Modify Status of a Single Coil

Command

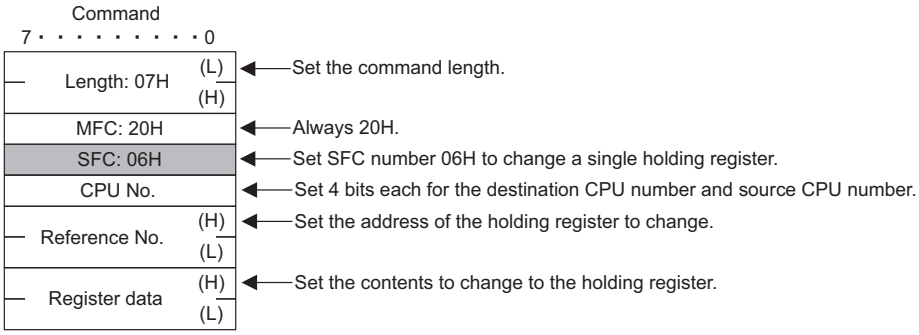


Response

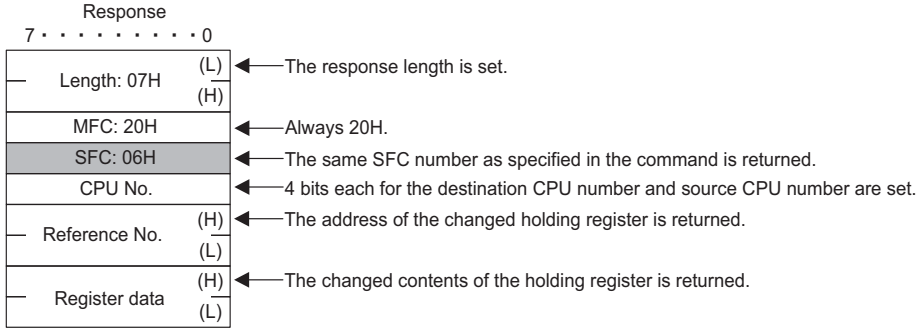


(6) Write to a Single Holding Register

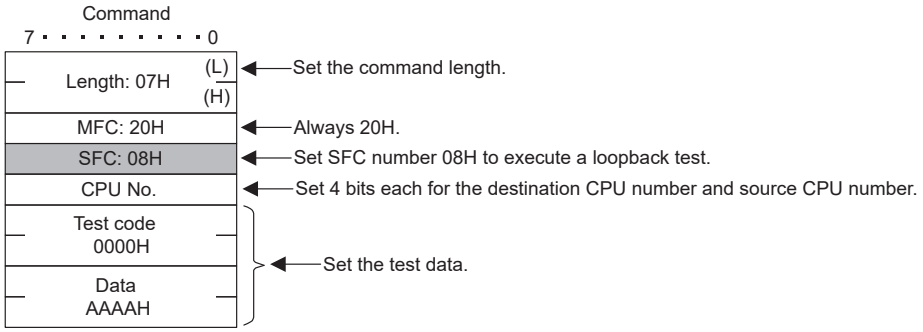
Command

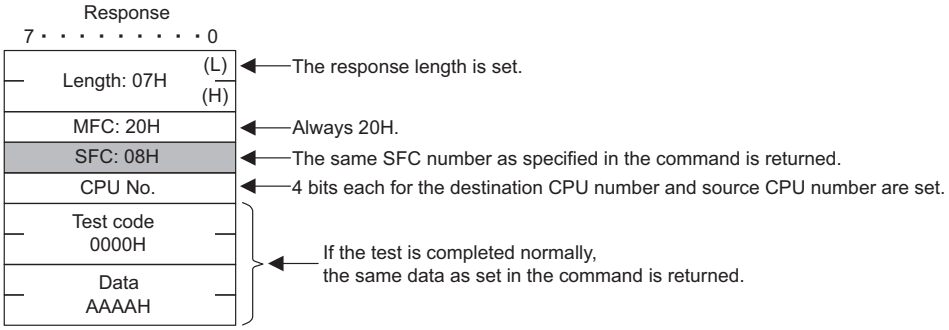


Response

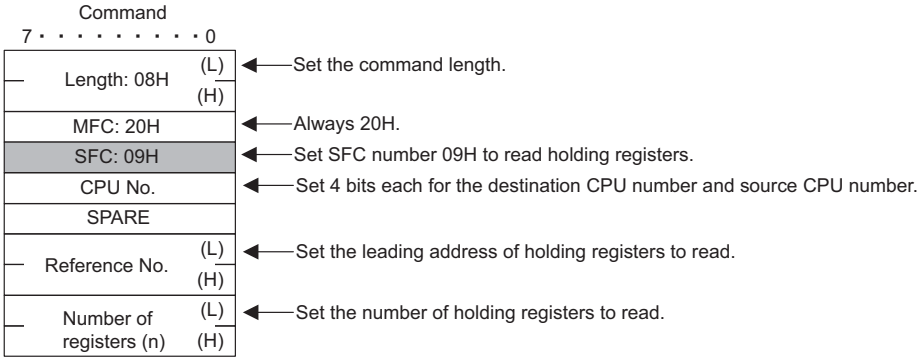


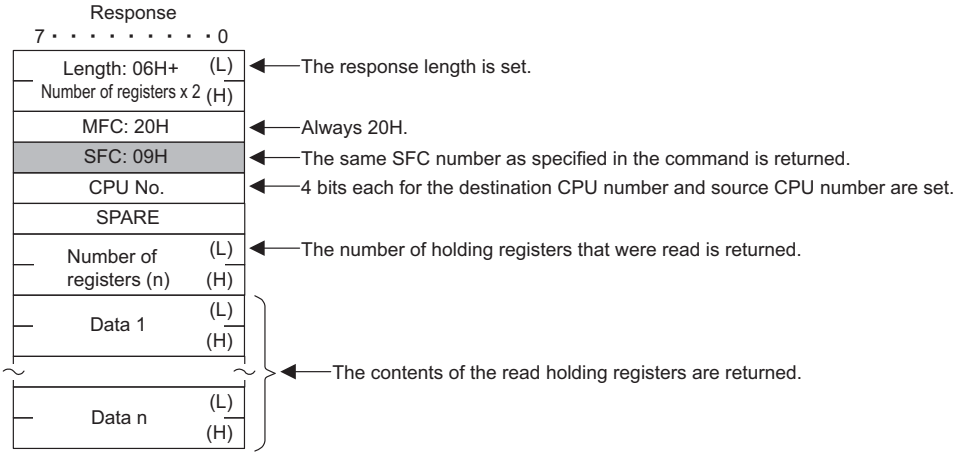
(7) Loopback Test





(8) Read Holding Register Contents (Extended)





(9) Read Input Register Contents (Extended)

Command

7・・・・・・・・・・0		
Length: 08H	(L)	Set the command length.
	(H)	
MFC: 20H		Always 20H.
SFC: 0AH		Set SFC number 0AH to read input registers.
CPU No.		Set 4 bits each for the destination CPU number and source CPU number.
SPARE		
Reference No.	(L)	Set the leading address of the input registers to read.
	(H)	
Number of registers (n)	(L)	Set the number of input registers to read.
	(H)	

Response

7・・・・・・・・・・0		
Length: 06H+ Number of registers x 2	(L)	The response length is set.
	(H)	
MFC: 20H		Always 20H.
SFC: 0AH		The same SFC number as specified in the command is returned.
CPU No.		4 bits each for the destination CPU number and source CPU number are set.
SPARE		
Number of registers (n)	(L)	The number of input registers that were read is returned.
	(H)	
Data 1	(L)	The contents of the read input register are returned.
	(H)	
～	～	
Data n	(L)	
	(H)	

(10) Write to Holding Register (Extended)

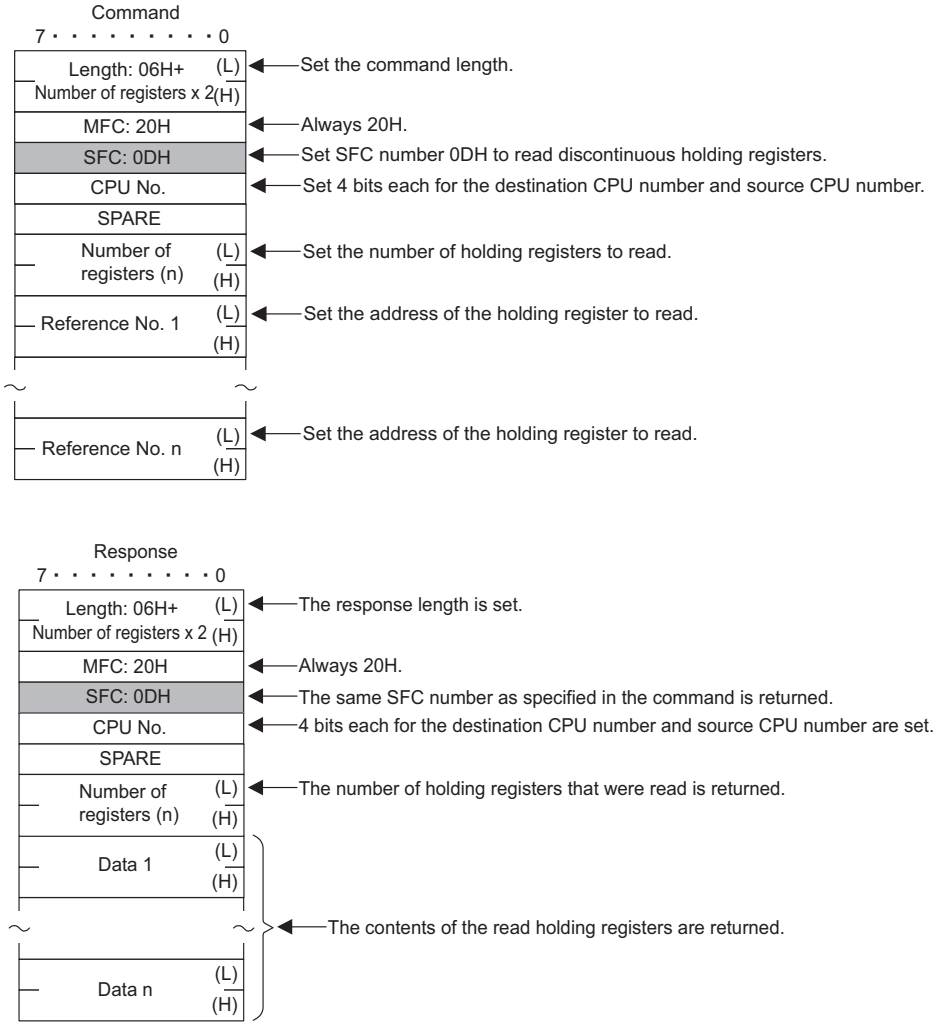
Command

7・・・・・・・・・・0		
Length: 08H+ Number of registers x 2	(L)	Set the command length.
	(H)	
MFC: 20H		Always 20H.
SFC: 0BH		Set SFC number 0BH to write holding registers.
CPU No.		Set 4 bits each for the destination CPU number and source CPU number.
SPARE		
Reference No.	(L)	Set the leading address of the holding registers to write.
	(H)	
Number of registers (n)	(L)	Set the number of the holding registers to write.
	(H)	
Data 1	(L)	Set the contents to write to the holding registers.
	(H)	
～	～	
Data n	(L)	
	(H)	

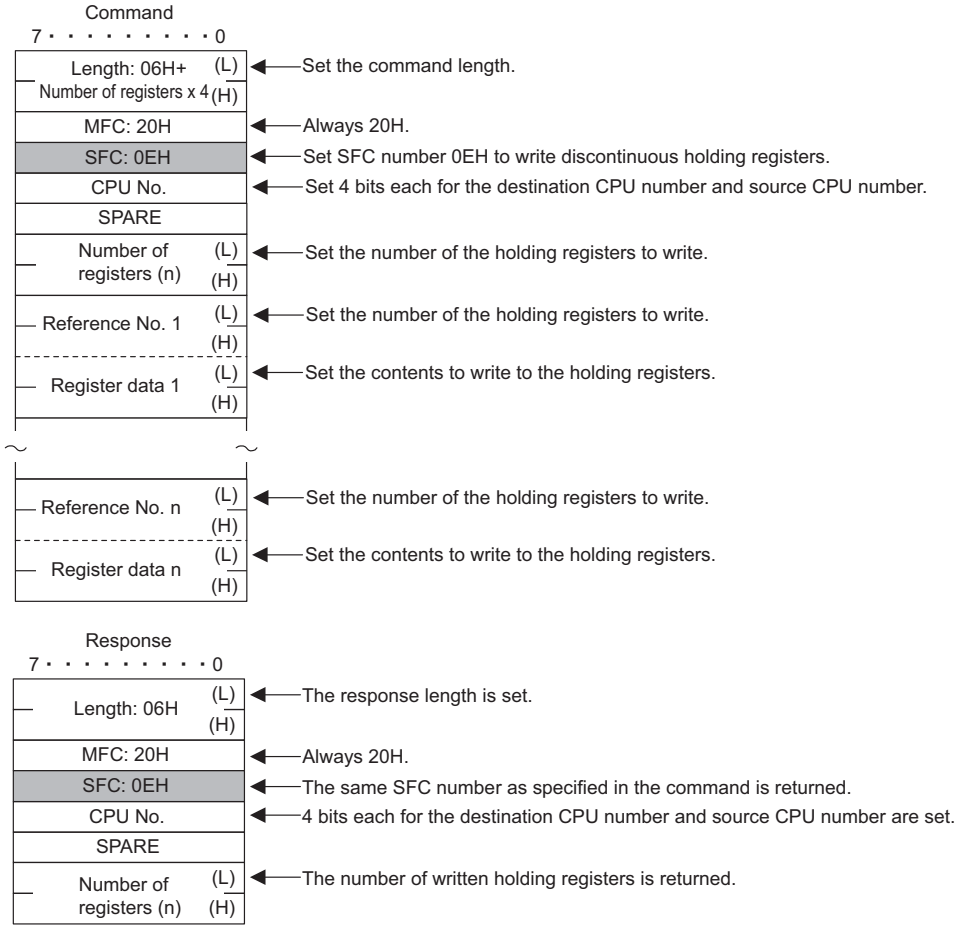
Response

7・・・・・・・・・・0		
Length: 08H	(L)	The response length is set.
	(H)	
MFC: 20H		Always 20H.
SFC: 0BH		The same SFC number as specified in the command is returned.
CPU No.		4 bits each for the destination CPU number and source CPU number are set.
SPARE		
Reference No.	(L)	The leading address of the holding registers that were written is returned.
	(H)	
Number of registers (n)	(L)	The number of written holding registers is returned.
	(H)	

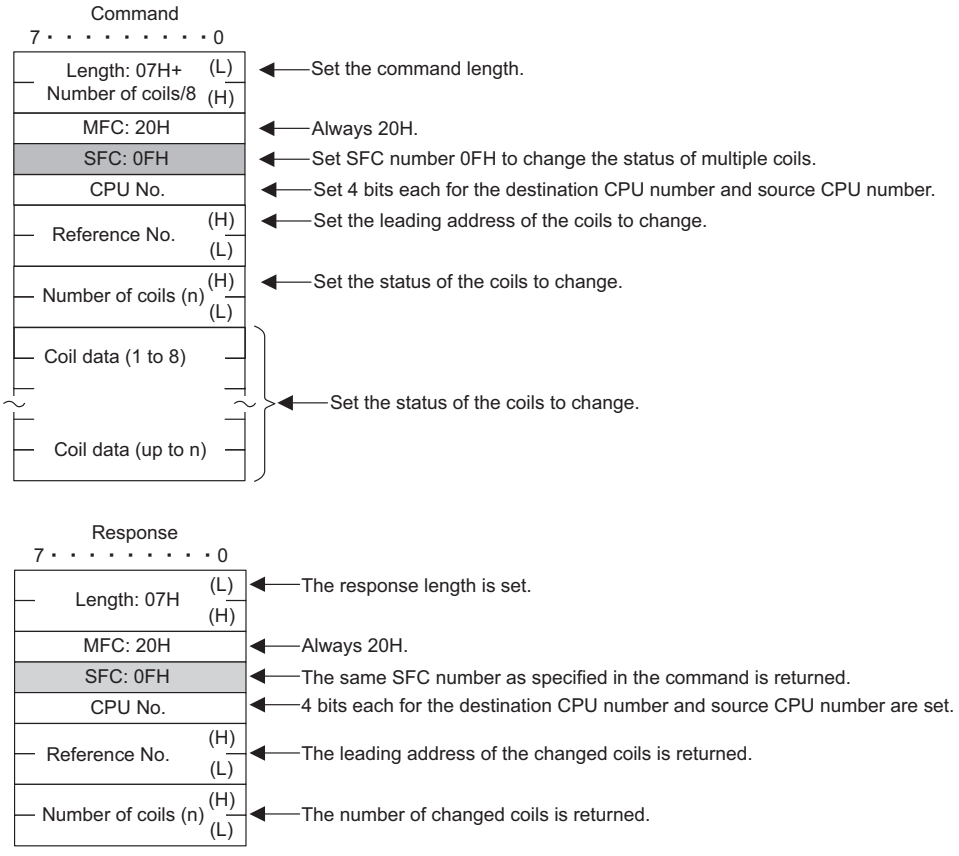
(11) Read Discontinuous Holding Register Contents (Extended)



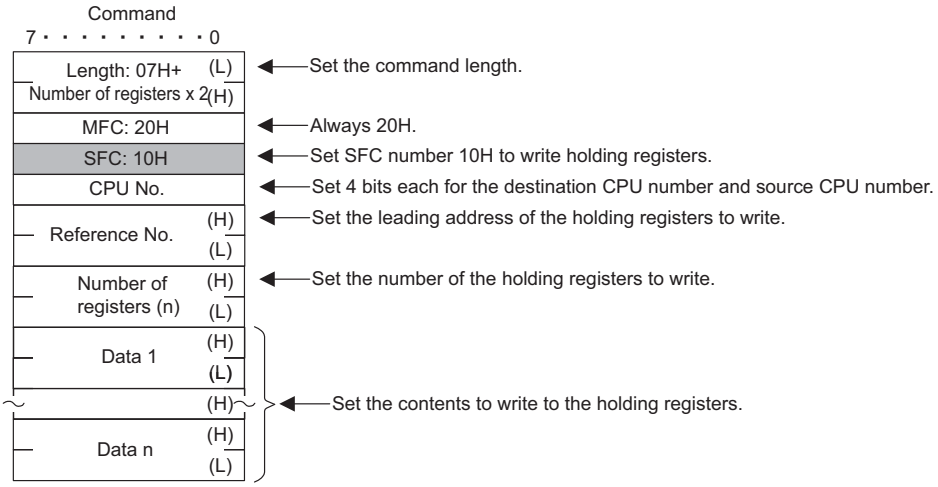
(12) Write to Discontinuous Holding Registers (Extended)

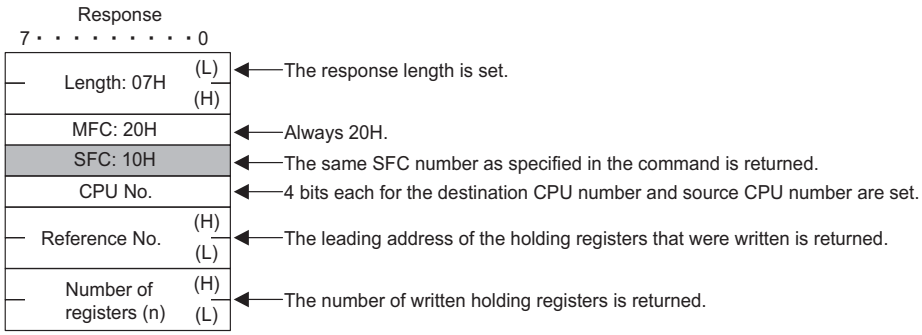


(13) Modify Status of Multiple Coils



(14) Write to Multiple Holding Registers

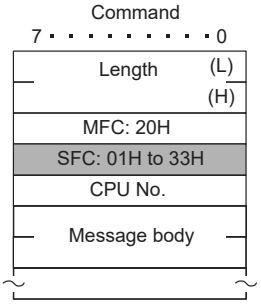


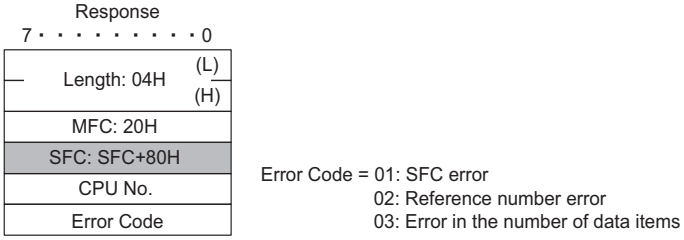


(15) Error Response

The following message is returned.

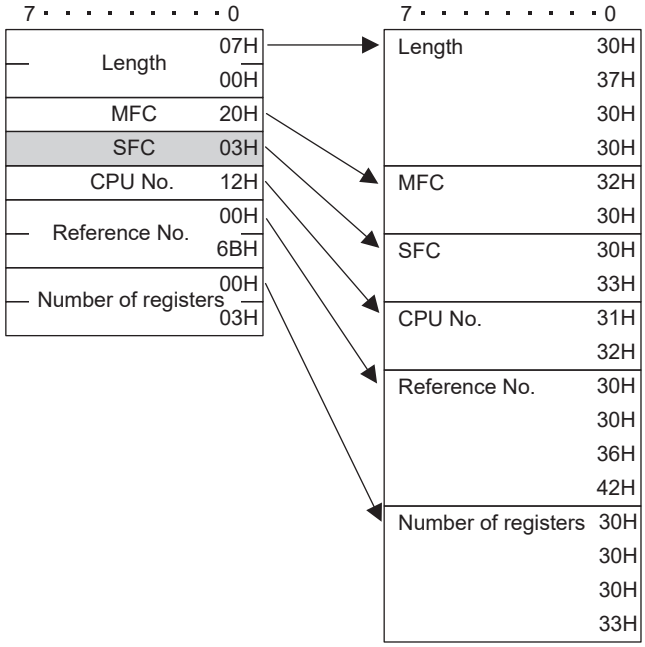
- When the SFC in the command message is incorrect
- When the reference number is incorrect
- When the number of data is incorrect





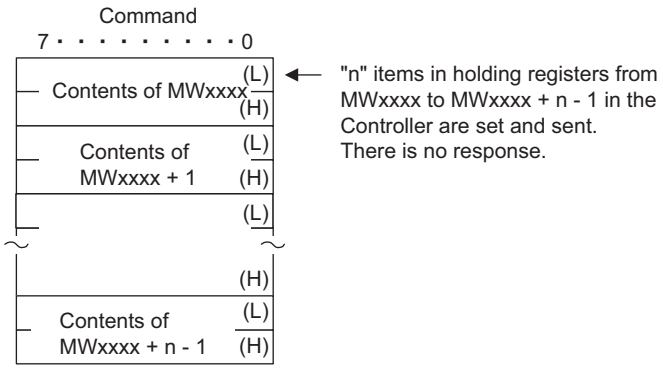
B.3 MEMOBUS ASCII Mode

With ASCII communication, binary data is converted to ASCII before being sent or received.
The following diagram shows an example of binary to ASCII conversion. As shown in the example, 8-bit data is converted into two ASCII characters (7 bits). The example shows the conversion of only the application data. In actual conversion, however, the EIF header is also converted to ASCII.



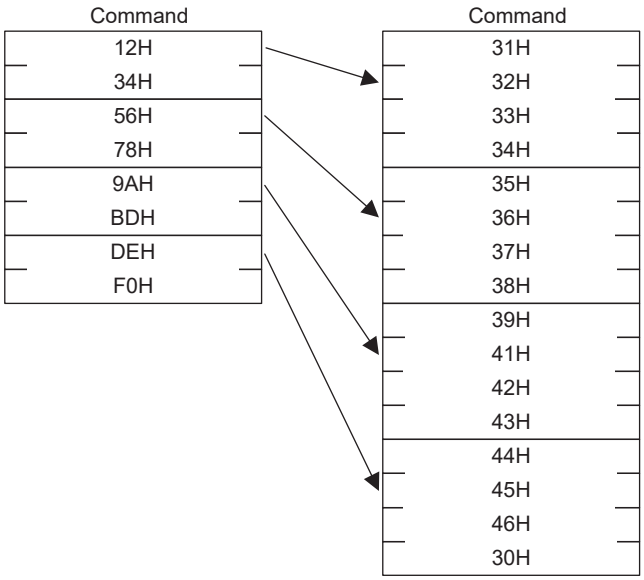
B.4 General-purpose Messages in Binary Mode

In general-purpose mode, the values set for the holding registers (MW registers) of the Controller are set as the application data that follows the EIF header for the command to be sent or received.



B.5 General-purpose Messages in ASCII Mode

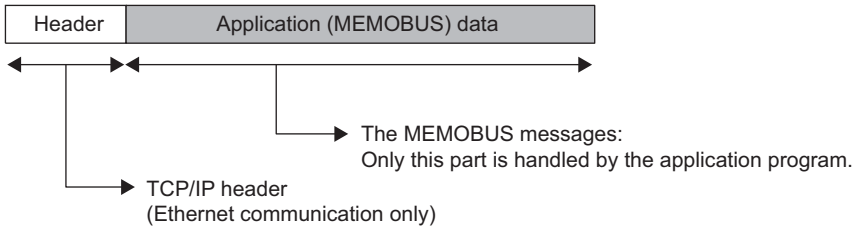
In ASCII mode, binary data is converted into ASCII before being sent or received.
The following diagram shows an example of binary to ASCII conversion. As shown in the example, 8-bit data is converted into two ASCII characters (7 bits). The example shows the conversion of only the application data. In actual conversion, however, the EIF header is also converted to ASCII.



Appendix C Details on the MEMOBUS Protocol

C.1 Message Configuration

The message configuration used in transmissions in the 217IF and 218IF transmission systems is shown below.
When the data is sent and received with the MEMOBUS protocol, messages consist of two elements: The header and application data. Unlike the Extended MEMOBUS protocol, there is no 218 header.



The header is used for only for Ethernet communication using TCP/IP and UDP/IP. When creating user programs, it is not necessary to be aware of the header because it is automatically set and deleted in the 218IF/218IFB transmission system.
In 217IF transmission system, only the application (MEMOBUS) data is sent and received.

The actual data for the MEMOBUS protocol is stored in the application data.

Communication Protocol	Code	Reference
MEMOBUS messages	RTU	*C.2 MEMOBUS RTU Mode* on page A-27
MEMOBUS messages	ASCII	*C.3 MEMOBUS ASCII Mode* on page A-32

- Whether RTU or ASCII code is used for Ethernet communication depends on the code handled by the remote station set in the connection parameters. With 217 transmission, it depends on the communication mode set in the transmission parameters.
- When receiving messages with Ethernet communications, the CRC-16 for RTU mode and the LRC for ASCII mode are not checked. Error checking for receptions is performed using error detection in the TCP, UDP, and IP headers. It is thus not necessary to calculate the CRC-16 or LRC when sending a message.

■ MEMOBUS Commands

The commands that make up the MEMOBUS messages are identified by function codes and have the functions shown in the following table.

Function Code	Function	MEMOBUS (RTU/ASCII)
01H	Read Coil Status	2000 points
02H	Read Input Relay Status	2000 points
03H	Read Holding Register Contents	125 words
04H	Read Input Register Contents	125 words
05H	Modify Status of a Single Coil	1 point
06H	Write to a Single Holding Register	1 word
08H	Loopback Test	–
0FH	Modify Status of Multiple Coils	800 points
10H	Write to Multiple Holding Registers	100 words

C.2 MEMOBUS RTU Mode

- When a message is received using the MEMOBUS protocol for the 218IF or 218IFB, the CRC-16 is not checked for the MEMOBUS protocol.

(1) Read Coil Status

Command

7 ・・・・・・・・・ 0		
Slave address		
Function code: 01H		← Set function code 01H to read the status of coils.
Start No.	(H)	← Set the leading address of the coils to read.
	(L)	
No. of coils	(H)	← Set the number of coils to read.
	(L)	
CRC-16	(H)	← Error check
	(L)	

Response

7 ・・・・・・・・・ 0	
Slave address	
Function code: 01H	← Function code 01H is set when reading the status of coils.
No. of data items	← The byte length of the read data is returned.
Status of the first 8 coils	
Status of the next 8 coils	← The read data contents are returned.
Status of the next 8 coils	
CRC-16 (H)	← Error check
CRC-16 (L)	

(2) Read Input Relay Status

Command

7 ・・・・・・・・・ 0		
Slave address		
Function code: 02H		← Set function code 02H to read the status of relays.
Start No.	(H)	← Set the leading address of the relays to read.
	(L)	
No. of Relays	(H)	← Set the number of relays to read.
	(L)	
CRC-16	(H)	← Error check
	(L)	

Response

7 ・・・・・・・・・ 0		
Slave address		
Function code: 02H		← Function code 02H is set when reading the status of relays.
No. of data items		← The byte length of the read data is returned.
Status of the first 8 relays		
Status of the next 8 relays		← The read data contents are returned.
Status of the next 8 relays		
CRC-16	(H)	← Error check
	(L)	

(3) Read Holding Register Contents

Command

7 · · · · · · · · · 0		
Slave address		
Function code: 03H		← Set function code 03H to read holding registers.
Start No.	(H)	← Set the leading address of the holding registers to read.
	(L)	
No. of registers	(H)	← Set the number of the holding registers to read.
	(L)	
CRC-16	(H)	← Error check
	(L)	

Response

7 · · · · · · · · · 0		
Slave address		
Function code: 03H		← Function code 03H is set when reading holding registers.
No. of data items		← The byte length of the read data is returned.
Contents of the first holding register	(H)	← The read data contents are returned.
	(L)	
Contents of the next holding register	(H)	
	(L)	
Contents of the next holding register	(H)	
	(L)	
CRC-16	(H)	
	(L)	← Error check

(4) Read Input Register Contents

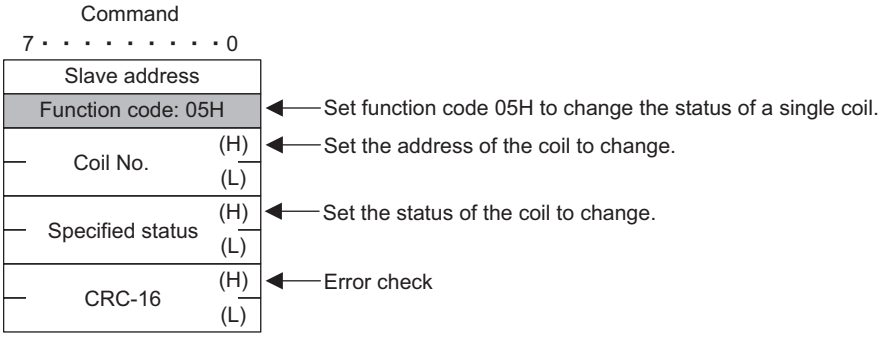
Command

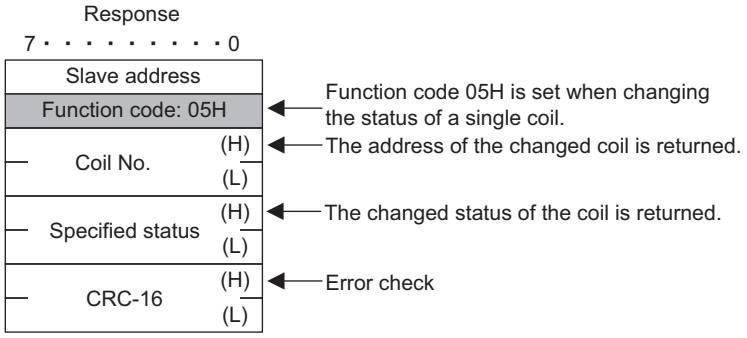
7 · · · · · · · · · 0		
Slave address		
Function code: 04H		← Set function code 04H to read input registers.
Start No.	(H)	← Set the leading address of the input registers to read.
	(L)	
No. of registers	(H)	← Set the number of input registers to read.
	(L)	
CRC-16	(H)	← Error check
	(L)	

Response

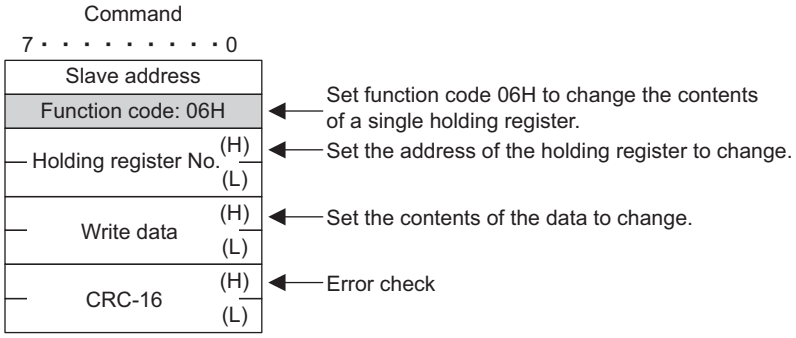
7 · · · · · · · · · 0		
Slave address		
Function code: 04H		← Function code 04H is set when reading input registers.
No. of data items		← The byte length of the read data is returned.
Contents of the first input register	(H)	← The read data contents are returned.
	(L)	
Contents of the next input register	(H)	
	(L)	
Contents of the next input register	(H)	
	(L)	
CRC-16	(H)	
	(L)	← Error check

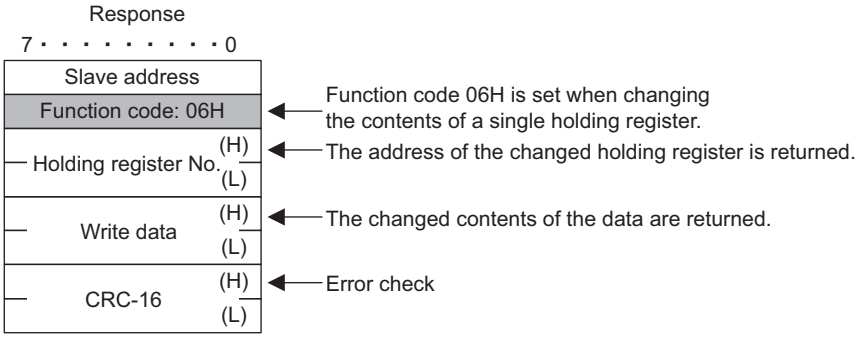
(5) Modify Status of a Single Coil





(6) Write to a Single Holding Register





(7) Loopback Test

Command

7 • • • • • • • • • 0		
Slave address		
Function code: 08H		← Set function code 08H to execute the loopback test.
Test code	(H)	← Set the test code.
	(L)	
Data	(H)	← Set the test data.
	(L)	
CRC-16	(H)	← Error check
	(L)	

Response

7 • • • • • • • • • 0		
Slave address		
Function code: 08H		← Function code 08H is set when executing the loopback test.
Test code	(H)	← If the test is completed normally, the same value as set in the command is returned.
	(L)	
Data	(H)	← If the test is completed normally, the same value as set in the command is returned.
	(L)	
CRC-16	(H)	← Error check
	(L)	

(8) Modify Status of Multiple Coils

Command

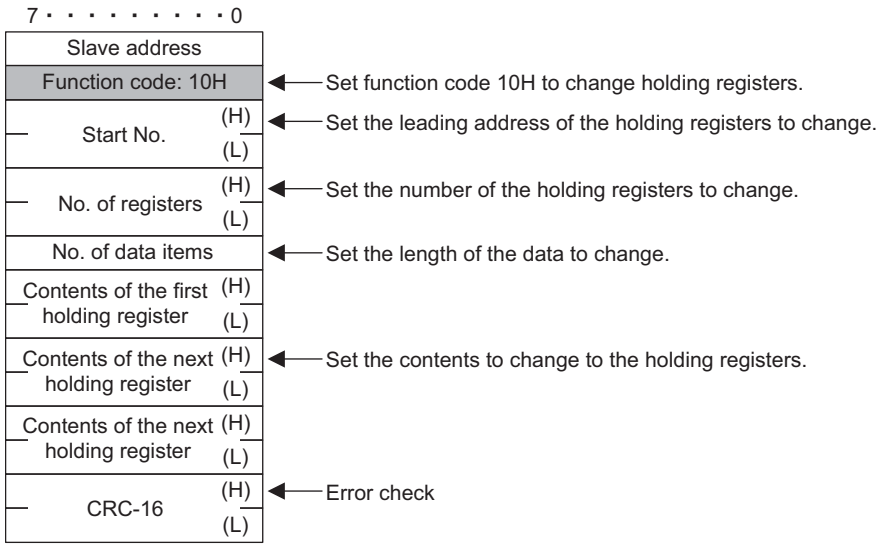
7 • • • • • • • • • 0		
Slave address		
Function code: 0FH		← Set function code 0FH to change the status of multiple coils.
Start No.	(H)	← Set the leading address of the coils to change.
	(L)	
No. of coils	(H)	← Set the number of the coils to change.
	(L)	
No. of data items		← Set the length of the data to change.
Status of the first 8 coils		← Set the status of the coils to change.
Status of the next 8 coils		
Status of the next 8 coils		
CRC-16	(H)	← Error check
	(L)	

Response

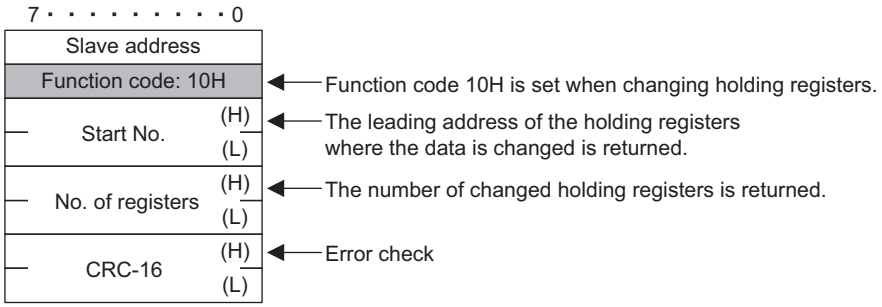
7 • • • • • • • • • 0		
Slave address		
Function code: 0FH		← Function code 0FH is set when changing the status of multiple coils.
Start No.	(H)	← The leading address of the changed coils is returned.
	(L)	
No. of coils	(H)	← The number of the changed coils is returned.
	(L)	
CRC-16	(H)	← Error check
	(L)	

(9) Write to Multiple Holding Registers

Command

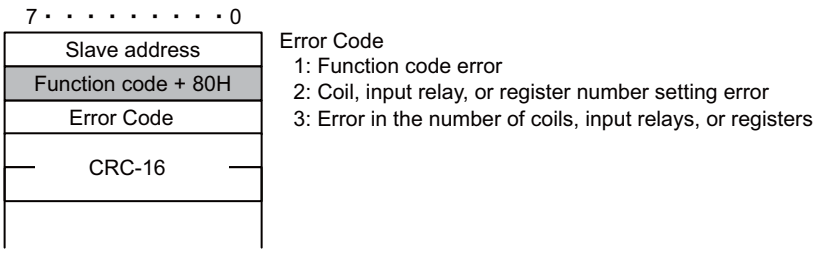


Response



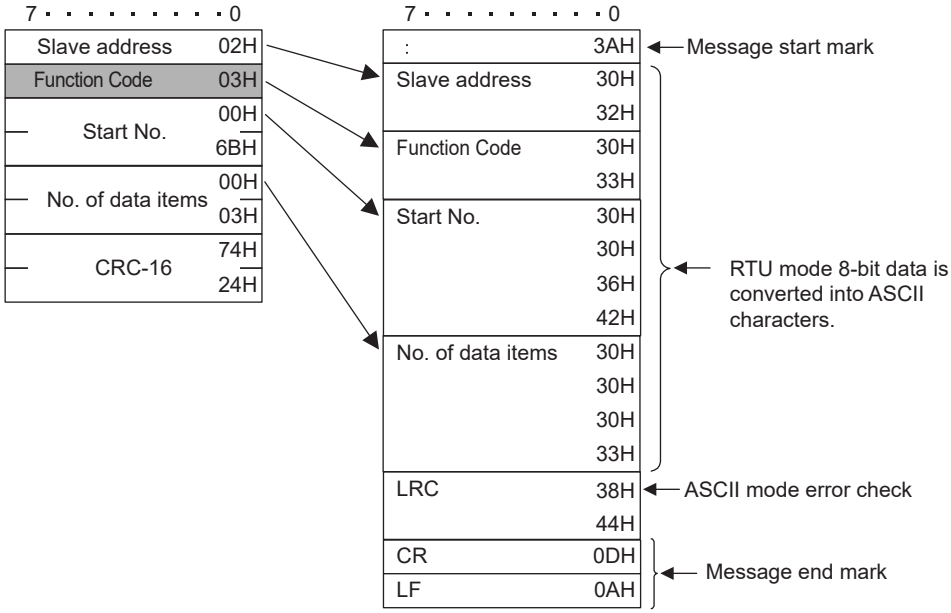
(10) Error Response

If a query message includes an error, the Slave returns an error response without executing the command.



C.3 MEMOBUS ASCII Mode

In ASCII communication, RTU data is converted into ASCII before being sent and received.
The following diagram shows an example of RTU to ASCII conversion. As shown in the example, 8-bit data is converted into two ASCII characters (7 bits). In the MEMOBUS format, the code ":" indicating the start of the data is set at the beginning of the data and the codes CR, LF are set at the end of the data. The error check is done in LRC.

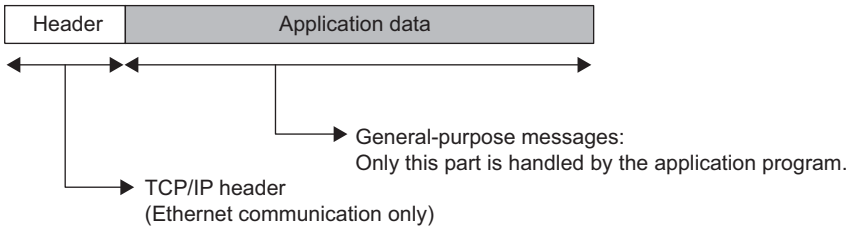


- When a message is received using the MEMOBUS protocol for the 218IF or 218IFB, the LRC is not checked for the MEMOBUS protocol.

Appendix D Details on Non-procedure Protocol

D.1 Message Configuration

When non-procedure is set as the communication protocol, application data is handled as a general-purpose message. When sending and receiving the data, messages consist of two elements: The header and application data.



The header is used for only for Ethernet communication using TCP/IP and UDP/IP. When creating user programs, it is not necessary to be aware of the header because it is automatically set and deleted in the 218IF/218IFB transmission system.
The format of the application data section can be set as required. The application data section has the following message structure.

Communication Protocol	Code	Applicable Transmission Method	Reference
Non-procedure protocol	BIN	217IF, 218IF, 218IFB	*D.2 General-purpose Messages in Binary Mode* on page A-34
Non-procedure protocol	ASCII	218IF, 218IFB	*D.3 General-purpose Messages in ASCII Mode* on page A-34

- The code handles for Ethernet communication (binary or ASCII) is determined by the code set in the connection parameters.
- With the 217IF transmission system, the communication type is "none" (binary).

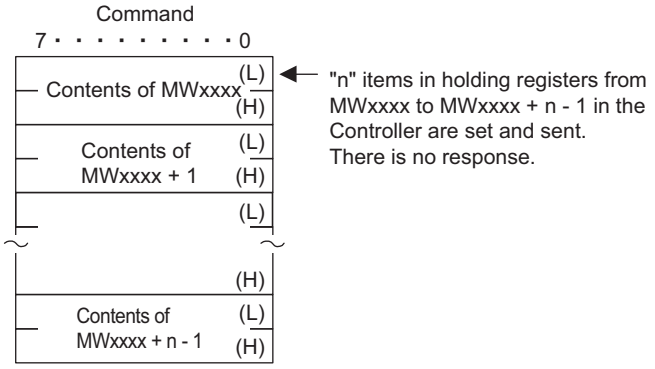
The difference from the general-purpose message mode of the Extended MEMOBUS protocol is that if the 218 header used in the Extended MEMOBUS protocol is not set preceding the application data.

■ General-purpose Message Commands

These commands are set as required by the application.

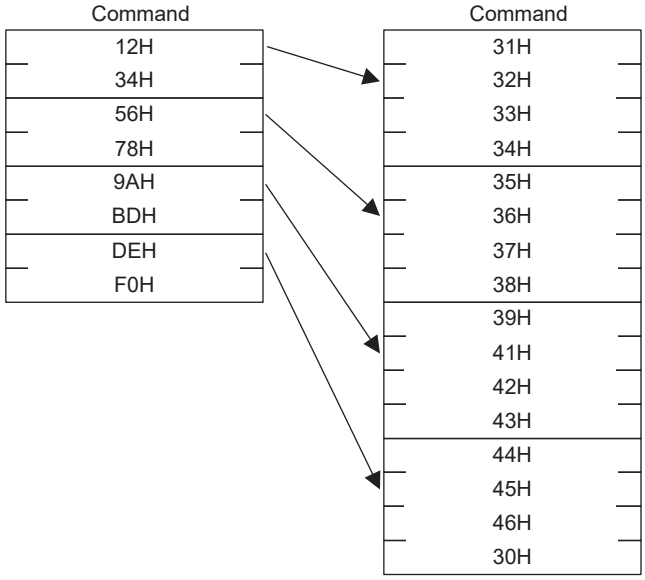
D.2 General-purpose Messages in Binary Mode

In the non-procedure protocol, the values set for the holding registers (MW registers) of the Controller are set in the application data, and the data is sent and received.



D.3 General-purpose Messages in ASCII Mode

In ASCII mode, binary data is converted into ASCII before being sent or received.
The following diagram shows an example of binary to ASCII conversion.
As shown in the example, 8-bit data is converted into two ASCII characters (7 bits).



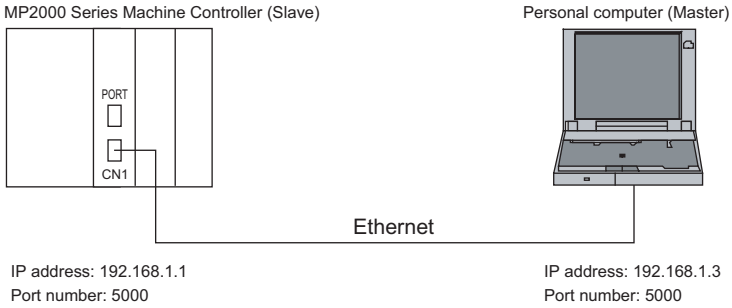
Appendix E Sample C Language Programs for the Master

This appendix provides samples of the C language programs used for Ethernet communication between a Communication Module and a personal computer or a workstation. These programs are for the master.

- The programs listed in this appendix were prepared only for reference and they may not operate correctly in actual applications. To execute the programs, a socket library and C compiler must be installed on the personal computer.
- The brief explanations on the program contents are written in italics after double-slash (//) in the programs.
- The line on the left end of the program indicates the positions of paragraph indentions.

E.1 TCP (When Using Extended MEMOBUS Protocol (SFC = 09))

〈Configuration and Setting Example〉




```
#include <stdio.h>
#include <winsock.h>
#include <winioctl.h>

int sd;             /* Socket Discripter */
struct sockaddr_in my;
struct sockaddr_in dst;

#define MY_IP      0xC0A80103  // Local IP address: 192.168.1.3
#define MY_PORT    5000        // Local port number

#define DST_IP     0xC0A80101  // Remote IP address: 192.168.1.1
#define DST_PORT   5000        // Remote port number

char sbuf[2048];
char rbuf[2048];

void main_tcp(void);
void mk_cmd_data(void);
int chk_rsp_data(int);

void main_tcp(void)
{
        WSADATAwsadata;

        int rc, slen, rlen;

        // Declaration to use Winsock.dll (always required at the beginning of the program)
        rc = WSAStartup( 0x0101, &wsadata );
        if ( rc != 0 )
        {
                exit(0);
        }

        // Clears the sockaddr structure (IP address, port number, etc.) to zero.
        memset( (char *)&my, 0, sizeof(struct sockaddr));
        memset( (char *)&dst, 0, sizeof(struct sockaddr));

        // Declaration of local IP address and port number
        my.sin_family = AF_INET;
        my.sin_addr.s_addr = htonl( MY_IP );
```

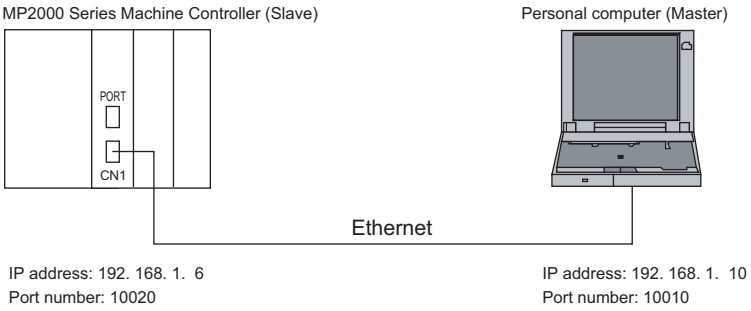
```
my.sin_port = htons( MY_PORT );

// Declaration of remote IP address and port number
dst.sin_family = AF_INET;
dst.sin_addr.s_addr = htonl( DST_IP );
dst.sin_port = htons( DST_PORT );

// Creates TCP socket.
sd = socket( AF_INET, SOCK_STREAM, 0 );
if ( sd <= 0 )
{
        printf( "Error: Socket !!\n" );
        exit(0);
}

// Binds the local port number (assignment).
rc = bind( sd, ( struct sockaddr *)&my, sizeof(struct sockaddr_in));
if ( rc == -1 )
{
        closesocket( sd );
        printf( "Error: bind !!\n" );
        exit(0);
}

// Establishes the connection.
rc = connect( sd, ( struct sockaddr *)&dst, sizeof(struct sockaddr_in));
if ( rc == -1 )
{
        closesocket( sd );
        printf( "Error: Connect !!\n" );
        exit(0);
}

// Prepares command data after the connection is established.
mk_cmd_data();

// Repeats sending the command and receiving the response.
while(1)
{
        // Sends the command data.
        // This processing will not end if the Master cannot send the data.
        slen = send( sd, &sbuf[0], 22, 0  );//Sends the command (22 bytes).
        if ( slen != 22 )// The number of bytes (22) that was sent will be returned if the sending process is successful.
        {
                closesocket(sd);
                printf( "Error: Send !! -> %d\n", slen );
                exit(0);
        }
        // Receives the response data.
        // This processing will not end if the Slave does not send the response data.
        rlen = recv( sd, &rbuf[0], sizeof(rbuf), 0 ); // Receives the response data from the remote station.
        if ( rlen <= 0 )//Value 0 or less will be returned if an error occurs in receiving data.
        {
                closesocket(sd);
                printf( "Error: Recv !! -> %d\n", rlen );
                exit(0);
        }
        // Checks the response data.
        rc = chk_rsp_data( rlen );
        if ( rc != 0 )//Errors in the receive data
        {
                closesocket(sd);
                exit(0);
        }
        sbuf[1] ++;// Increments the serial number of 218 header.
        printf( "Hit Any Key !!\n" );
```

```
        }
}

// Prepares the extended MEMOBUS protocol command, Read Holding Register Contents (SFC = 09).
void mk_cmd_data(void)
{
        // Prepares of 218 Header
        // Sets the data type.
        sbuf[0] = 0x11;// Extended MEMOBUS (reference command)

        // Sets the serial number (The serial number will be incremented every send data.)
        sbuf[1] = 0x00;

        // Sets the destination channel number.
        sbuf[2] = 0x00; // The channel number can be fixed to 0 as the channel of PLC is not specified.

        // Sets the destination channel number.
        sbuf[3] = 0x00; // Fixed to 0 since a personal computer has no channel number.

        sbuf[4] = 0x00; // Reserved
        sbuf[5] = 0x00; // Reserved

        // Sets all the number of data items (from the start of 218 header to the end of MEMOBUS data).
        sbuf[6] = 0x16; // L (22 bytes = 218 header (12 bytes) + MEMOBUS data (10 bytes)
        sbuf[7] = 0x00; // H

        sbuf[8] = 0x00;// Reserved
        sbuf[9] = 0x00;// Reserved
        sbuf[10] = 0x00;// Reserved
        sbuf[11] = 0x00;// Reserved

        // Prepares the MEMOBUS data.
        // Length: From MFC to the end of data
        sbuf[12] = 0x08;    // MEMOBUS data length (L)
        sbuf[13] = 0x00;    // MEMOBUS data length (H)

        // MFC is fixed to 0x20.
        sbuf[14] = 0x20;

        // SFC is 0x09 (Read Holding Register Contents (extended)).
        sbuf[15] = 0x09;

        // Sets the CPU numbers.
        sbuf[16] = 0x10;    // Remote CPU No.: CPU1. Multi CPUs: 1 - 4. Local CPU No.: always 0.

        sbuf[17] = 0x00;    // Spare is fixed to 0.

        // Sets the reference number.
        sbuf[18] = 0x00;    // Adr(L) leading address: MW0
        sbuf[19] = 0x00;    // Adr(H)

        // Sets the number of registers.
        sbuf[20] = 0x0A;    // DataNum(L)  Reads 10 words from the DataNum(L) leading address.
        sbuf[21] = 0x00;    // DataNum(H)

}

// Checks the response data.
int chk_rsp_data( int rlen )
{
        int rc;

        rc = 0;
        // Checks the total data length.
        if ( rlen != 40 )// 40-byte response for 10-word read out
```

```
                    // ((218 header (12 bytes) + MEMOBUS data (28 bytes))
        {
                rc = -1;
                return( rc );
        }
        // Checks the packet type.
        if ( rbuf[0] != 0x19 )// Not the MEMOBUS response
        {
                rc = -2;
                return( rc );
        }

        // Checks the serial number.
        if ( sbuf[1] != rbuf[1] )// Does not agree with the command serial number.
        {
                rc = -3;
                return( rc );
        }

        // Checks the length of the total data in the message.
        if (( rbuf[6] != 0x28 ) && (rbuf[7] !=0x00))// 40 bytes = 218 header (12 bytes) + MEMOBUS data (28 bytes)
        {
                rc = -4;
                return( rc );
        }

        // Checks the MEMOBUS data length.
        if (( rbuf[12] != 0x1A ) || (rbuf[13] != 0x00))// 26 bytes
        {
                rc = -5;
                return( rc );
        }

        // Checks the MFC.
        if ( rbuf[14] != 0x20 )// MFC is fixed to 0x20.
        {
                rc = -6;
                return( rc );
        }

        // Checks the SFC.
        if ( rbuf[15] != 0x09 )//SFC is 0x09 (Read Holding Register Contents).
        {
                rc = -7;
                return( rc );
        }

        // Checks the number of registers.
        if (( rbuf[18] != 0x0A ) || (rbuf[19] != 0x00))//Not 10 words.
        {
                rc = -8;
                return( rc );
        }

        // Reads the register data rbuf[20] and the data that follows.

        return( rc );
}
```

E.2 UDP (When Using Extended MEMOBUS Protocol (SFC = 09))

<Configuration and Setting Example>




```
#include <stdio.h>
#include <winsock.h>
#include <winioctl.h>

int sd;             /* Socket Discripter */
struct sockaddr_in my;
struct sockaddr_in dst;
struct sockaddr_in from;

#define MY_IP      0xC0A8010A  // Local IP address: 192.168.001.010
#define MY_PORT    10010       // Local port number

#define DST_IP     0xC0A80106  // Remote IP address: 192.168.001.006
#define DST_PORT   10020       // Remote port number

unsigned char sbuf[2048];
unsigned char rbuf[2048];

// 218TCP.C
extern  void mk_cmd_data(void);
extern  int chk_rsp_data(int);

void main_udp(void)
{
        WSADATAwsadata;

        int rc, slen, rlen, fromlen;

        // Declaration to use Winsock.dll (always required at the beginning of the program)
        rc = WSAStartup( 0x0101, &wsadata );
        if ( rc != 0 )
        {
                exit(0);
        }

        // Clears the sockaddr structure (IP address, port number, etc.) to zero.
        memset( (char *)&my, 0, sizeof(struct sockaddr));
        memset( (char *)&dst, 0, sizeof(struct sockaddr));

        // Declaration of local IP address and port number
        my.sin_family = AF_INET;
        my.sin_addr.s_addr = htonl( MY_IP );
        my.sin_port = htons( MY_PORT );

        // Declaration of remote IP address and port number
        dst.sin_family = AF_INET;
        dst.sin_addr.s_addr = htonl( DST_IP );
        dst.sin_port = htons( DST_PORT );

        // Creates UDP socket.
        sd = socket( AF_INET, SOCK_DGRAM, 0 );
        if ( sd <= 0 )
```

```
        {
                printf( "Error: Socket !!\n" );
                exit(0);
        }

        // Binds the local port number (assignment).
        rc = bind( sd, ( struct sockaddr *)&my, sizeof(struct sockaddr_in));
        if ( rc == -1 )
        {
                closesocket( sd );
                printf( "Error: bind !!\n" );
                exit(0);
        }

        // Prepares the command data after assigning the local port number.
        mk_cmd_data();

        // Repeats sending the command and receiving the response.
        while(1)
        {
                // Sends the command data.
                // This processing will not end if the Master cannot send the data.
                 slen = sendto( sd, &sbuf[0], 22, 0, (struct sockaddr *)&dst, sizeof(struct sockaddr));
                // Sends the command (22 bytes).
                if ( slen != 22 )// The number of bytes (22) that was sent will be returned if the sending process is successful.
                {
                        closesocket(sd);
                        printf( "Error: Send !! -> %d\n", slen );
                        exit(0);
                }
                // Receives the response data.
                // This processing will not end if the Slave does not send the response data.
                fromlen = sizeof(struct sockaddr);
                 rlen = recvfrom( sd, &rbuf[0], sizeof(rbuf), 0, (struct sockaddr *)&from, &fromlen );
                // Receives data from the remote station.
                if ( rlen <= 0 )//Value 0 or less will be returned if an error occurs in receiving data.
                {
                        closesocket(sd);
                        printf( "Error: Recv !! -> %d\n", rlen );
                        exit(0);
                }
                // Checks the response data.
                rc = chk_rsp_data( rlen );
                if ( rc != 0 )//Errors in the receive data
                {
                        closesocket(sd);
                        exit(0);
                }
                sbuf[1] ++;// Increments the serial number of 218 header.
                printf( "Hit Any Key !!\n" );
        }
}
```

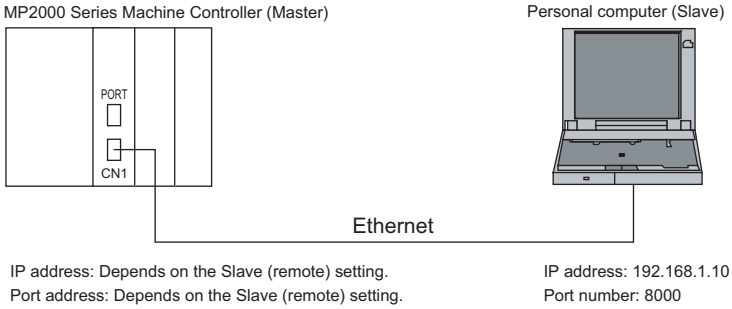
Appendix F Sample C Language Programs for a Slave

This appendix provides samples of the C language programs used for Ethernet communication between a Communication Module and a personal computer or a workstation. These programs are for a slave.

- The programs listed in this appendix were prepared only for reference purpose and they may not operate correctly in actual applications. To execute the programs, a socket library and C compiler must be installed on the personal computer.
- The brief explanations on the program contents are written in italics after double-slash (//) in the programs.
- The line on the left end of the program indicates the positions of paragraph indentions.

F.1 TCP (When Using Extended MEMOBUS Protocol)

<Configuration and Setting Example>




```
#include <stdio.h>
#include <winsock.h>
#include <winioctl.h>

int     sd;                 // Socket to wait for connection
int             new_sd;     // Socket for communication after connected
struct  sockaddr_in my;     // Work to store local IP address and port number
struct  sockaddr_in dst;    // Work to store remote IP address and port number

char    sbuf[2048];                 // Send buffer
char    rbuf[2048];                 // Receive buffer

#define MY_IP     0xC0A8010A        // Local IP address: 192.168.1.10
#define MY_PORT   8000              // Local port number: 8000

void main_tcp_Srv(void);
void mk_rsp_data(int*);
int chk_cmd_data(int);

void main_tcp_Srv(void)
{
        WSADATAwsadata;

        int rc, slen, rlen, send_len, client_addrlen;

        // Declaration to use Winsock.dll (always required at the beginning of the program)
        rc = WSAStartup( 0x0101, &wsadata );
        if ( rc != 0 )
        {
                exit(0);
        }

        // Clears the sockaddr structure (IP address, port number, etc.) to zero.
        memset( (char *)&my, 0, sizeof(struct sockaddr));
        memset( (char *)&dst, 0, sizeof(struct sockaddr));

        // Declaration of local IP address and port number
        my.sin_family = AF_INET;
        my.sin_addr.s_addr = htonl( MY_IP );
        my.sin_port = htons( MY_PORT );
```

```
// Creates TCP socket.
sd = socket( AF_INET, SOCK_STREAM, 0 );
if ( sd <= 0 )  // Value 0 or less will be returned if an error occurs in processing.
{
        printf( "Error: Socket !!\n" );
        exit(0);
}

// Binds the local port number (assignment).
rc = bind( sd, ( struct sockaddr *)&my, sizeof(struct sockaddr_in));
if ( rc == -1 )  // -1 will be returned if an error occurs in processing.
{
        closesocket( sd );
        printf( "Error: bind !!\n" );
        exit(0);
}

// Preparation to wait for connection: Up to 5 connection requests can be held.
// (The number of connection requests is limited depending on the OS.)
rc = listen( sd, 5 );
if( rc == -1 )  // -1 will be returned if an error occurs in processing.
{
        closesocket( sd );
        printf( "Error: listen !!\n" );
        exit(0);
}

client_addrlen = (int)(sizeof(struct sockaddr_in));

// Wait for connection
// This processing will not end if no connection request is sent from the Master.
new_sd = accept( sd, ( struct sockaddr * )&dst, &client_addrlen );
if( new_sd == -1 )  // -1 will be returned if an error occurs in processing.
{
        closesocket( sd );
        printf( "Error: accept !!\n" );
        exit(0);
}

// sd is the socket to wait for connection.  When connecting processing is executed, another socket (new_sd)
// will be created.   The new_sd will be the socket number to be used for communication.

// The socket to wait for connection is closed here so that any more connection request will not be accepted.
closesocket( sd );

// Extended MEMOBUS data sending and receiving processing
// A response must be sent for the received command when using the extended MEMOBUS protocol.
// Repeats receiving the command data and sending the response data.
while(1)
{
        // Receives the command data.
        // This processing will not end if the Master does not send the command data.
        rlen = recv( new_sd, &rbuf[0], sizeof(rbuf), 0 ); // Receives data from the remote station.
        if ( rlen <= 0 )// Value 0 or less will be returned if an error occurs in receiving data.
        {
                closesocket(new_sd);
                printf( "Error: Recv !! -> %d\n", rlen );
                exit(0);
        }

        // Checks the receive data.
        rc = chk_cmd_data( rlen );
        if ( rc != 0 )// Errors in the receive data
        {
                closesocket(new_sd);
                exit(0);
```

```
                }

                // Prepares the response data.
                mk_rsp_data( &send_len );

                // Sends the response data.
                // This processing will not end if the Slave cannot send the response data.
                slen = send( new_sd, &sbuf[0], send_len, 0  );
                if ( slen != send_len )// The number of bytes that was sent will be returned if the sending processing is
                successful.
                {
                        closesocket(new_sd);
                        printf( "Error: Send !! -> %d\n", slen );
                        exit(0);
                }

                printf( "Hit Any Key !!\n" );
        }
}


/*****************************************************************************/
/* Preparation of extended MEMOBUS response data                             */
/*                                                                           */
/* An appropriate response for the function code (SFC=03, 06, 08, 09, 0B, or 10)  */
/* of extended MEMOBUS protocol will be returned.                            */
/* In this sample program, an error response will be returned for other function codes as they  */
/* are not supported. A dummy data will be prepared and returned for Read command.  */
/*****************************************************************************/
void mk_rsp_data( int *slen )
{
int             i;
unsigned shortdata_size, memex_len;

        /************************************/
        /* Preparation of Extended MEMOBUS Data  */
        /************************************/
        // MFC
        sbuf[14] = 0x20;        // Fixed to 20H
        // SFC
        sbuf[15] = rbuf[15];    // Sends the received data.
        // Sets the CPU numbers.
        sbuf[16] = (rbuf[16] << 4) & 0xF0;          // The destination is the source CPU.  Local CPU No.: 0

        // Processing for each function code
        switch( rbuf[15] )
        {
                case    0x03:   // Read Holding Register Contents
                        data_size = (unsigned short)((unsigned char)rbuf[19]) << 8;
                        data_size += (unsigned short)((unsigned char)rbuf[20]);
                        data_size *= 2;// Read data size in bytes
                        memex_len = data_size + 3;                      // Extended MEMOBUS Length for response
                         // data size + 3byte(MFC,SFC,CPU NO)
                        // Prepares a dummy data.
                        // Converts then sends the requested word unit data into byte units.
                        for( i = 0; i < data_size; i++ )
                        {
                                sbuf[17 + i] = i;// Sets data read from sbuf[17].
                        }
                        break;

                case    0x06:   // Modify a Single Holding Register Contents
                case    0x08:   // Loopback Test
                        memex_len = 7;// Extended MEMOBUS Length for Response
                        sbuf[17] = rbuf[17];// Sends the received data
                        sbuf[18] = rbuf[18];// Sends the received data
```

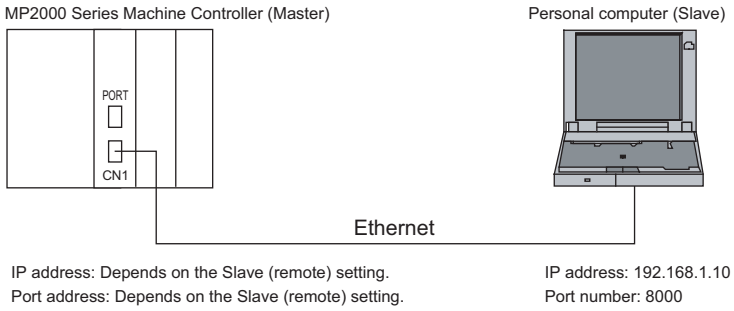
```
                sbuf[19] = rbuf[19];// Sends the received data
                sbuf[20] = rbuf[20];// Sends the received data
                break;

        case    0x09:   // Read Holding Register Contents (Extended)
                data_size = (unsigned short)((unsigned char)rbuf[21]) << 8;
                data_size += (unsigned short)((unsigned char)rbuf[20]);
                data_size *= 2;// Read data size in bytes
                memex_len = data_size + 6;// Extended MEMOBUS Length for response
                // data size + 6byte(MFC,SFC,CPU NO,SPEAR,REG NO)
                sbuf[17] = 0x00;    // Spare is fixed to 0.
                // Sets the number of registers.
                sbuf[18] = rbuf[20];    // DataNum(L)
                sbuf[19] = rbuf[21];    // DataNum(H)
                // Prepares a dummy data.
                // Converts then sends the requested word unit data into byte units.
                for( i = 0; i < (data_size/2); i++ )
                {
                        sbuf[20 + 2 * i] = i;           // Sets data read from sbuf[20].
                }
                break;

        case    0x0B:   // Write to Holding Register (Extended)
                memex_len = 8;// Extended MEMOBUS Length for response
                sbuf[17] = 0x00;    // Spare is fixed to 0.
                sbuf[18] = rbuf[18];// Sends the received data
                sbuf[19] = rbuf[19];// Sends the received data
                sbuf[20] = rbuf[20];// Sends the received data
                sbuf[21] = rbuf[21];// Sends the received data
                break;

        case    0x10:   // Write to Holding Registers
                memex_len = 7;// Extended MEMOBUS Length for response
                sbuf[17] = rbuf[17];// Sends the received data
                sbuf[18] = rbuf[18];// Sends the received data
                sbuf[19] = rbuf[19];// Sends the received data
                sbuf[20] = rbuf[20];// Sends the received data
                break;

        default:// In this sample program, function code other than the above listed are judged as not
                // supported codes and an error response will be returned.
                memex_len = 4;// Extended MEMOBUS Length for response
                sbuf[15] |= 0x80;// Error response (SFC + 80H)
                sbuf[17] = 0x01;// Error response (SFC error)
}

// Length: From MFC to the end of data
sbuf[12] = (char)(memex_len & 0x00FF); // Extended MEMOBUS length L
sbuf[13] = (char)((memex_len >> 8) & 0x00FF); // Extended MEMOBUS length H

/*************************/
/* Preparation of 218 Header  */
/*************************/
// Sets the data type.
sbuf[0] = 0x19;         // Extended MEMOBUS (Response)

// Sets the serial number.
sbuf[1] = rbuf[1];      // Sends the received data

// Sets the destination channel number.
sbuf[2] = rbuf[3];      // Sets the PLC source channel number.

// Sets the source channel number.
sbuf[3] = 0x00;         // Fixed to 0 because a personal computer has no channel number.
```

```
        sbuf[4] = 0x00;         // Reserved
        sbuf[5] = 0x00;         // Reserved

        // Sets all the number of data items (from the start of 218 header to the end of MEMOBUS data).
        sbuf[6] = (char)((12 + 2 + memex_len) & 0x00FF);
        // L (218 header (12 bytes) + MEMOBUS header length (2 bytes) + Extended MEMOBUS length
        (memex_len bytes)
        sbuf[7] = (char)(((12 + 2 + memex_len) >> 8) & 0x00FF); // H

        sbuf[8] = 0x00;         // Reserved
        sbuf[9] = 0x00;         // Reserved
        sbuf[10] = 0x00;// Reserved
        sbuf[11] = 0x00;// Reserved

        // Sets the send data length (218 header + Extended MEMOBUS data).
        *slen = (int)(memex_len + 12 + 2);
}



/*****************************************************/
/* Extended MEMOBUS Command Check                    */
/* Checks the header of the received data.           */
/*****************************************************/
int chk_cmd_data( int rlen )
{
        int rc;
        unsigned shortdata_len, memex_len;

        rc = 0;

        // Total data length
        data_len = (unsigned short)((unsigned char)rbuf[7]) << 8;
        data_len += (unsigned short)((unsigned char)rbuf[6]);


        // Checks the total data length.
        if ( rlen != (int)data_len )
        {
                rc = -1;
                return( rc );
        }

        // Checks the packet type.
        if ( rbuf[0] != 0x11 )// Commands other than MEMOBUS commands are not accepted.
        {
                rc = -2;
                return( rc );
        }

        // Extended MEMOBUS Data Length
        memex_len = (unsigned short)((unsigned char)rbuf[13]) << 8;
        memex_len += (unsigned short)((unsigned char)rbuf[12]);

        // Checks the extended MEMOBUS data length.
        if (( data_len - 14) != memex_len )
        // Extended MEMOBUS data length  is not equal to Total data length - 218 header (12 bytes) - Extended MEMOBUS
        length (2 bytes).
        {
                rc = -3;
                return( rc );
        }

        // Checks the MFC.
        if ( rbuf[14] != 0x20 )// MFC is fixed to 0x20.
        {
                rc = -4;
```

```
                return( rc );
        }

        // Checks the SFC.
        // Checks within mk_rsp_data().

        return( rc );
}
```

F.2 UDP (When Using Extended MEMOBUS Protocol)

<Configuration and Setting Example>




```
#include <stdio.h>
#include <winsock.h>
#include <winioctl.h>

int     sd;                     // Socket
struct  sockaddr_in my;         // Work structure to store local IP address and port number
struct  sockaddr_in from;// Work structure to store the remote IP address and port number

char    sbuf[2048];             // Send buffer
char    rbuf[2048];             // Receive buffer

#define MY_IP     0xC0A8010A // Local IP address: 192.168.1.10
#define MY_PORT   8000       // Local port number: 8000

// 218TCPSrv.C
extern  void mk_rsp_data(int*);
extern  int chk_cmd_data(int);

void main_udp_Srv(void);

void main_udp_Srv(void)
{
        WSADATAwsadata;

        int rc, slen, rlen, send_len, fromlen;

        // Declaration to use Winsock.dll (always required at the beginning of the program)
        rc = WSAStartup( 0x0101, &wsadata );
        if ( rc != 0 )
        {
                exit(0);
        }

        // Clears the sockaddr structure (IP address, port number, etc.) to zero.
        memset( (char *)&my, 0, sizeof(struct sockaddr));
        memset( (char *)&from, 0, sizeof(struct sockaddr));

        // Declaration of local IP address and port number
        my.sin_family = AF_INET;
        my.sin_addr.s_addr = htonl( MY_IP );
        my.sin_port = htons( MY_PORT );
```

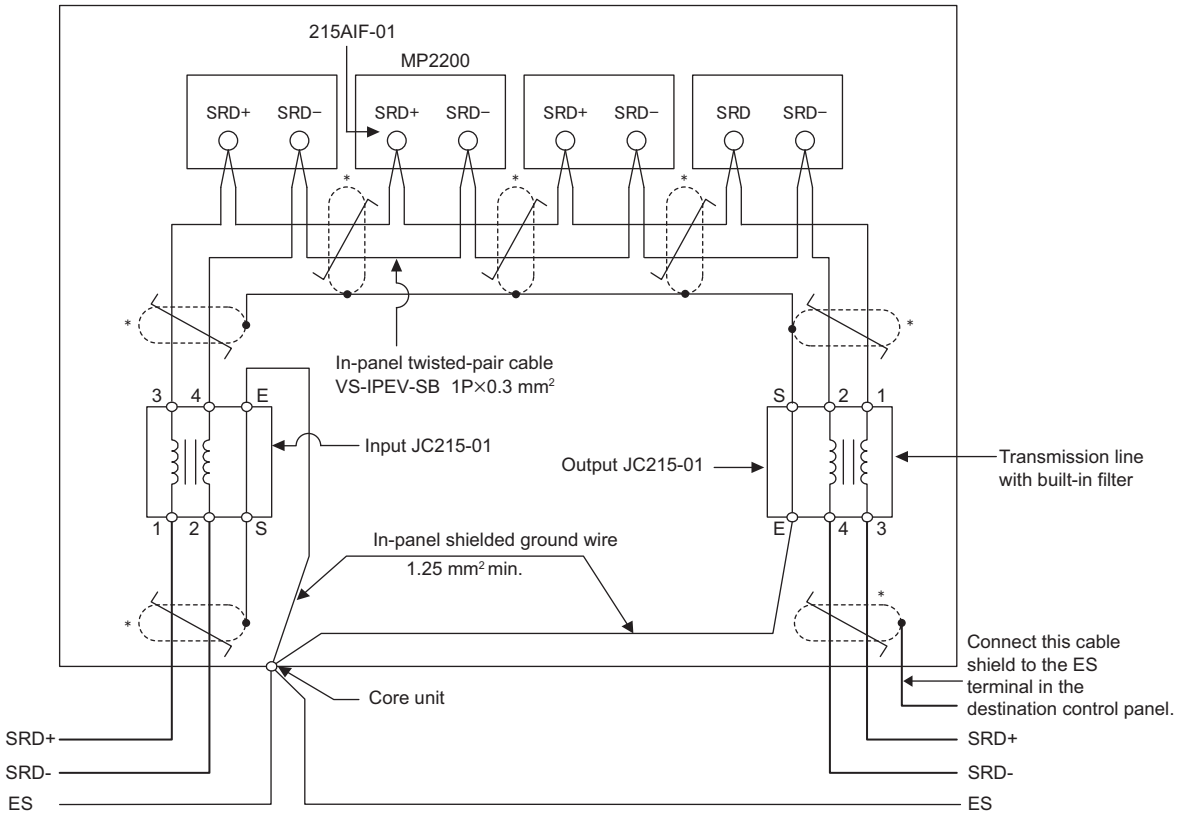
```
        // Creates UDP socket.
        sd = socket( AF_INET, SOCK_DGRAM, 0 );
        if ( sd <= 0 )  // Value 0 or less will be returned if an error occurs in processing.
        {
                printf( "Error: Socket !!\n" );
                exit(0);
        }

        // Binds the local port number (assignment).
        rc = bind( sd, ( struct sockaddr *)&my, sizeof(struct sockaddr_in));
        if ( rc == -1 )  // -1 will be returned if an error occurs in processing.
        {
                closesocket( sd );
                printf( "Error: bind !!\n" );
                exit(0);
        }

        // Immediately waits for receive data since there is no other connection.

        // Extended MEMOBUS data sending and receiving processing
        // A response must be sent for the received command when using the extended MEMOBUS protocol.
        // Repeats receiving the command data and sending the response data.
        while(1)
        {
                // Receives the command data.
                // This processing will not end if the Master does not send the command data.
                fromlen = sizeof(struct sockaddr);
                rlen = recvfrom( sd, &rbuf[0], sizeof(rbuf), 0, (struct sockaddr *)&from, &fromlen );
                if ( rlen <= 0 )//Value 0 or less will be returned if an error occurs in receiving data.
                {
                        closesocket(sd);
                        printf( "Error: Recv !! -> %d\n", rlen );
                        exit(0);
                }

                // Checks the receive data.
                rc = chk_cmd_data( rlen );
                if ( rc != 0 )  // Errors in the receive data
                {
                        closesocket(sd);
                        exit(0);
                }

                // Prepares the response data.
                mk_rsp_data( &send_len );

                // Sends the response data.
                // This processing will not end if the Slave cannot send the response data.
                 slen = sendto( sd, &sbuf[0], send_len, 0, (struct sockaddr *)&from, sizeof(struct sockaddr));
                if ( slen != send_len )  // The number of bytes that was sent will be returned if the sending processing is
                successful.
                {
                        closesocket(sd);
                        printf( "Error: Send !! -> %d\n", slen );
                        exit(0);
                }

                printf( "Hit Any Key !!\n" );
        }
}
```

Appendix G Junction Box Connection Example for CP-215 Communications

The Junction Box is used to connect the in-panel cables with the cables outside the panel.

G.1 Connections with the 215AIF-01 Module

The following figure shows a connection example between the 215AIF-01 Module mounted to an MP2000-series Machine Controller and a JC215-01 Junction Box.

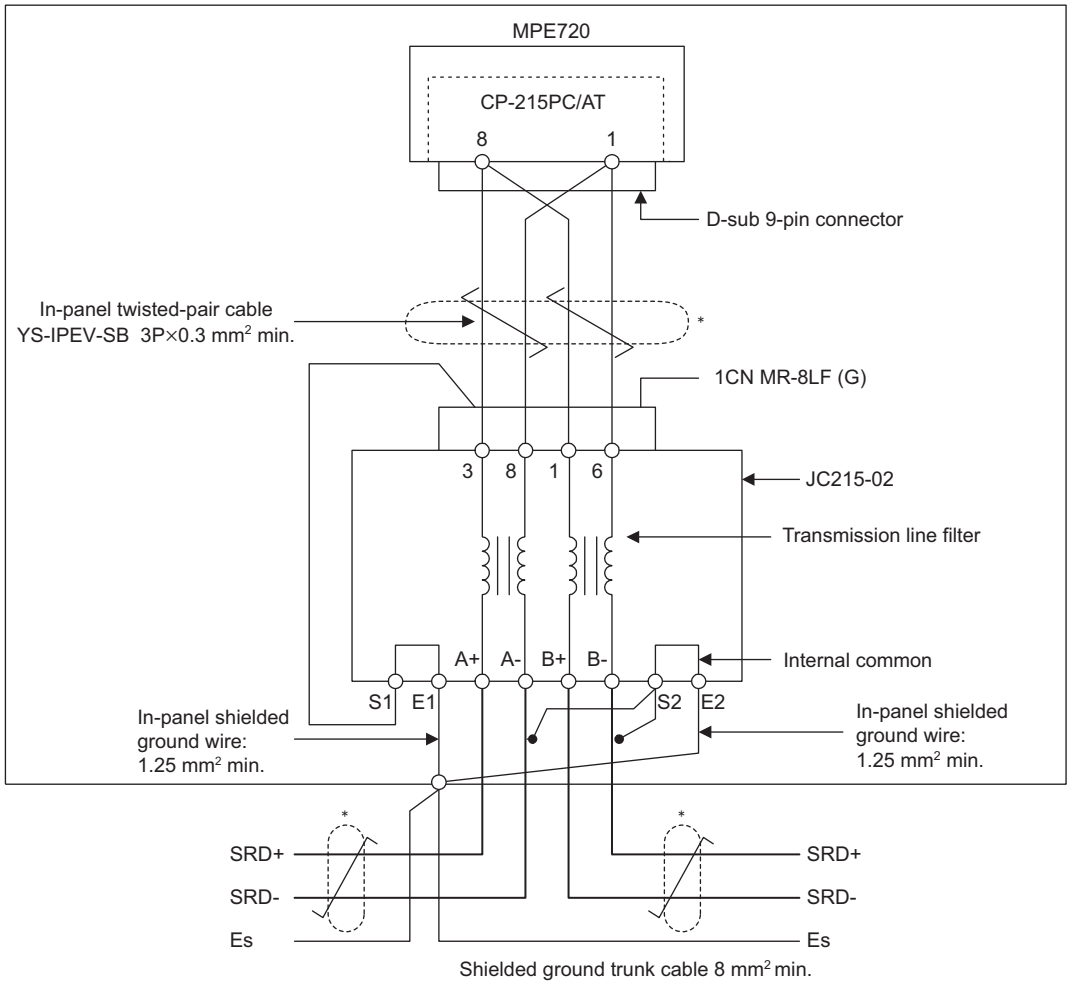


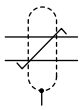
* : represents shielded twisted-pair wires.

- Communications Interface Modules
  Connect the terminals for the same signals with the in-panel cables.
  Connect pin 8 on the MR-8LM(G) to SRD+ and pin 1 to SRD-.
- JC215-01 Junction Box
  Install Junction Boxes on both the input and output sides.
  Signal terminal pairs 1-2 and 3-4 do not have I/O directions.
- Terminating Resistor
  If the Module is installed at the end of the communications line, connect a terminating resistor of 75 Ω between terminals 1 and 2 or between terminals 3 and 4 on the JC215-01.
- Grounding the Shield
  Connect one end only of the shield on the cables both inside and outside the panel to the S terminal on the JC215-01 on the input and output sides. Connect the E terminal to the Es terminal on the panel with a 1.25 mm^2 ground line.

G.2 Connection to a Personal Computer

The following figure shows a connection example between a CP-215PC/AT Board mounted in a IBM PC/AT or compatible and a JC215-02 Junction Box.



*  : represents shielded twisted-pair wires.

- Communications Interface Module
  Connect the CP-215PC/AT to the JC215-02 with an in-panel cable.
- In-panel Cable Signal Lines
  Connect SRD+/SRD- to the A+/A- or B+/B- terminals on the JC215-02.
- Terminating Resistor
  If the Board is at the end of the communications line, connect a terminating resistor to the A+/A- terminals or B+/B- terminals on the JC215-02.
- Grounding the Shield
  Connect one end only of the shield on the cables both inside and outside the panel to the S1 and S2 terminals on the JC215-02. Connect the E1 and E2 terminals to the Es terminal on the panel with a 1.25 mm^2 ground line.

Index

Numerics

215AIF-01 Module
- appearance - 9-3
- connector pin arrangement - 9-11
- connector specifications - 9-11
- default parameters - 9-10
- hardware specifications - 9-6
- indicators - 9-8
- relay function - 9-34
- switch settings - 9-9

217IF parameters
- automatic reception - 3-18
- automatically reception - 3-18
- baud rate - 3-17
- communication mode - 3-17
- communication protocol - 3-16
- data length - 3-17
- device address - 3-17
- master/slave - 3-16
- parity - 3-17
- readout of input registers - 3-18
- readout of input relays - 3-18
- readout/write-in of coils - 3-18
- receive monitor time - 3-17
- send delay - 3-17
- sending - 3-17
- serial I/F - 3-17
- stop bits - 3-17
- transmission mode - 3-17
- transmission protocol - 3-16
- write-in width of coil/holding registers - 3-18

217IF Transmission Configuration Window - 3-15
217IF transmission settings - 3-14
217IF transmissions
- setting parameters - 3-16

217IF-01 Module
- appearance - 6-2
- connecting to another 217IF-01 Module - 3-22
- connector pin arrangement - 6-9
- connector specifications - 6-9
- hardware specifications - 6-4
- indicators - 6-6
- switch settings - 6-7

218IF parameters
- CNO - 4-14, 4-20
- code - 4-15, 4-22
- connect type - 4-15
- connection number - 4-14, 4-20
- connection type - 4-15, 4-22
- count of retry - 4-14
- DIAG. port no. - 4-20
- diagnostic port number - 4-20
- error count - 4-21
- error status - 4-21
- gateway IP address - 4-17, 4-20
- IP address - 4-13
- IP assemble time - 4-18
- local port - 4-14
- maximum packet length - 4-18
- node IP address - 4-14
- node name - 4-15, 4-22
- protocol type - 4-15, 4-22
- receive count - 4-21
- reception pass counter - 4-21
- remote station name - 4-15, 4-22
- remote station's IP address - 4-14, 4-19
- remote station's port number - 4-15
- response time - 4-14, 4-22
- send count - 4-21
- station IP address - 4-20
- subnet mask - 4-17, 4-20
- system port no. - 4-17
- TCP close time - 4-18
- TCP retry time - 4-18
- TCP zero window timer value - 4-18
- trans status - 4-20
- transmission pass counter - 4-21
- transmission status - 4-20

218IF Transmission Configuration Window - 4-12
218IF transmission definitions - 4-11
- local station and TCP/IP settings - 4-17
- Status Tab Page - 4-20
- Transmission Parameters Tab Page - 4-13

218IF-01 Module
- appearance - 4-2
- connecting to another 218IF-01 Module - 4-34
- connector pin arrangement - 4-9
- connector specifications - 4-9
- default parameters - 4-8
- hardware specifications - 4-4
- indicators - 4-6
- switch settings - 4-7

218IF-02 Module
- appearance - 5-3
- connecting to another 218IF-02 Module - 5-35
- connector pin arrangement - 5-11
- connector specifications - 5-11
- default parameters - 5-10
- hardware specifications - 5-6
- indicators - 5-8

218IFB parameters
- CNO - 5-17
- code - 5-19, 5-22
- connect type - 5-18
- connection number - 5-17
- connection type - 5-18, 5-22
- count of retry - 5-17
- detail definition button - 5-16

error count ---- 5-21
error status ---- 5-21
gateway IP address ---- 5-16
IP address ---- 5-15
local port ---- 5-17
module name definition ---- 5-16
node IP address ---- 5-18
node name ---- 5-19, 5-22
node port ---- 5-18
protocol type ---- 5-18, 5-22
receive count ---- 5-21
reception pass counter ---- 5-21
remote station name ---- 5-19, 5-22
remote station's IP address ---- 5-18
remote station's port number ---- 5-18
response time ---- 5-16, 5-22
send count ---- 5-21
subnet mask ---- 5-15
system port number ---- 5-17
trans status ---- 5-20
transmission pass counter ---- 5-21
transmission status ---- 5-20
218IFB Transmission Configuration Window ---- 5-14
218IFB transmission definitions ---- 5-13
Status Tab Page ---- 5-20
Transmission Parameters Tab Page ---- 5-15
260IF parameters
baud rate ---- 7-22
BSIZE ---- 7-20, 7-21
comment ---- 7-21, 7-22
communication cycle time ---- 7-20
D ---- 7-20
EM ---- 7-21
explicit messages ---- 7-21
INPUT ---- 7-20, 7-21
input area size ---- 7-20, 7-21
leading input register ---- 7-20, 7-21
leading output register ---- 7-20, 7-21
MAC ID ---- 7-20, 7-21, 7-22
master/slave ---- 7-20
on-timer delay ---- 7-19
OUTPUT ---- 7-20, 7-21
output area size ---- 7-20, 7-21
SCAN ---- 7-21
status ---- 7-22
STS ---- 7-22
TYPE ---- 7-21
260IF Transmission Configuration Window ---- 7-18
260IF transmission definitions ---- 7-17
I/O Status Tab Page ---- 7-21
parameter setting example ---- 7-24
Status Tab Page ---- 7-22
Transmission Parameters Tab Page ---- 7-19
260IF-01 Module
appearance ---- 7-2
connector pin arrangement ---- 7-14
connector specifications ---- 7-14
default parameters ---- 7-12
hardware specifications ---- 7-9
indicators ---- 7-11
troubleshooting ---- 7-28
261IF parameters
comment ---- 8-13, 8-15
current communication cycle time ---- 8-16
D ---- 8-12, 8-13
DATA ---- 8-15
I/O status ---- 8-15
INPUT ---- 8-13, 8-15
leading input register ---- 8-13, 8-15
leading output register ---- 8-13, 8-15
local station number ---- 8-16
master station number ---- 8-16
maximum communication cycle time ---- 8-16
OUTPUT ---- 8-13, 8-15
SCAN ---- 8-13
SIZE ---- 8-13, 8-15
slave status ---- 8-16
ST# ---- 8-12, 8-14, 8-15
station number ---- 8-12, 8-14, 8-15
SYNC-SCAN ---- 8-12
transmission speed ---- 8-16
261IF Transmission Configuration Window ---- 8-11
261IF transmission definitions ---- 8-10
I/O Assignment Tab Page ---- 8-12
I/O Map Tab Page ---- 8-14
I/O Status Tab Page ---- 8-15
Status Tab Page ---- 8-16
261IF-01 Module
appearance ---- 8-2
connector pin arrangement ---- 8-8
connector specifications ---- 8-8
default parameters ---- 8-7
hardware specifications ---- 8-4
indicators ---- 8-6
switch settings ---- 8-7

A

automatic reception ---- 2-4
execution timing ---- 2-5

C

communication between CPUs ---- 4-27, 5-27
communication cycle time ---- 7-26
communication modes
overview ---- 2-2
Communication Modules
installing ---- 1-12
removing ---- 1-14
Communication Platform ---- 1-17
opening ---- 1-17
setting ---- 1-18
communication protocols ---- 2-6
communication settings (MPE720) ---- 1-28
connecting 218IF-02 Modules ---- 5-35

connecting to a computer
218IF-01 ----- 4-43
connecting to an operator interface
217IF-01 to GP-450 ----- 3-27
connecting to inverter
217IF-01 to VS-616G5 ----- 3-48
connecting to MELSEC
217IF-01 ----- 3-32
218IF-02 ----- 5-40
connecting to OMRON PLC
217IF-01 ----- 3-37
connecting to personal computer
218IF-02 ----- 5-44
connecting to Programming Device
217IF-01 to MPE720 ----- 3-51
connecting to temperature controller
217IF-01 to SRmini ----- 3-42
connection ----- 4-14, 5-18
CP-215 cables ----- 9-14

D

DeviceNet cable ----- 7-15
DeviceNet communication ----- 7-1
DeviceNet connector ----- 7-14
DeviceNet interface ----- 7-2
DeviceNet transmission specifications ----- 7-9
drop-line connections ----- 7-16

E

engineering communication ----- 2-5, 9-29
Ethernet communication ----- 5-24
Ethernet communication port
setting ----- 1-21
Ethernet connector
100Base-TX ----- 5-11
Ethernet connector (10Base-T) ----- 4-9
Ethernet interface ----- 4-2, 5-3
Ethernet transmission specifications ----- 4-4, 5-6
explicit message communication ----- 7-35
explicit request messages ----- 7-36
explicit response messages ----- 7-36
extended MEMOBUS commands ----- 9-30
extended MEMOBUS protocol ----- 2-7, 4-26, 5-26, A-14
function codes ----- 4-26, 5-26

F

fixed buffer communication ----- 4-27, 5-27

G

gateway IP address ----- 4-17, 5-16
general-purpose message communication ----- 2-4
general-purpose message communication mode ----- 2-7

L

link assignment ----- 9-25
link communication ----- 2-5, 9-25
link communication example ----- 9-38

M

maximum number of stations ----- 9-33
MELSEC
connecting to MELSEC ----- 4-39
MELSEC ACPU common commands ----- 3-5, 4-28, 5-28
MELSEC devices ----- 3-6, 4-28, 5-28
MELSEC protocol ----- 2-8, 3-5, 4-27, 5-27
function codes ----- 5-28
MEMOBUS message communication ----- 2-3
MEMOBUS mode ----- 2-7
MEMOBUS protocol ----- 2-6, 3-4, 4-25, 5-25, A-26
function codes ----- 3-4, 4-25, 5-25
message communication ----- 2-3, 9-30
message communication example ----- 9-42
message receive function
Ethernet communication ----- 5-78
MPLINK ----- 9-63
serial communication ----- 3-79
message send function
218IF/218IFB ----- 5-48
260IF ----- 7-40
MPLINK ----- 9-46
serial communication ----- 3-53
MODBUS/TCP protocol ----- 2-9, 4-31, 5-32
function codes ----- 4-31, 5-32
MPLINK cables ----- 9-13
MPLINK connector ----- 9-12
MPLINK parameters
comment ----- 9-20, 9-23
maintenance switch ----- 9-19
maximum number of stations connected ----- 9-19
MEMOBUS response monitoring time ----- 9-19
message relay function ----- 9-19
network no. ----- 9-18
numbers of relayed networks ----- 9-19
REG ----- 9-20, 9-22
register number ----- 9-20, 9-22
SCAN ----- 9-20, 9-22
SIZE ----- 9-20, 9-22
ST# ----- 9-20, 9-22
station name ----- 9-20, 9-23
station no. ----- 9-18
station number ----- 9-20, 9-22
station search interval ----- 9-19
status ----- 9-23
SYNC-SCAN ----- 9-20
token cycle time ----- 9-22
token cycle time setting ----- 9-18
transmission speed ----- 9-19
TYPE ----- 9-20, 9-22
MPLINK transmission definitions
I/O Map Tab Page ----- 9-21
Link Assignment Tab Page ----- 9-20
Transmission Parameters Tab Page ----- 9-18
MPLINK/CP-215
non-procedure protocol ----- 9-31
MPLINK/CP-215 transmission specifications ----- 9-6
MR connector converter ----- 9-12
MR conversion connector for CP-215 communication ----- 9-12
multi-branch connections ----- 7-16
multi-drop connections ----- 7-15

N

non-procedure FD protocol ----- 3-11
non-procedure protocol ----- 2-10, 3-10, 4-32, 5-33, A-33

O

OMRON devices ---- 3-9
OMRON protocol ---- 2-9, 3-8
  function codes ---- 3-9
Optional Modules
  installation ---- 1-12
  replacing and adding ---- 1-14

P

parameters
  DeviceNet communication
    sending ---- 7-45
  Ethernet communication
    receiving ---- 5-83
    sending ---- 5-53
  MPLINK
    receiving ---- 9-68
    sending ---- 9-51
  serial communication
    receiving ---- 3-84
    sending ---- 3-58
PORT connector ---- 4-9, 5-11, 6-9, 7-14, 8-8
port number ---- 4-14, 5-18
PROFIBUS connection example ---- 8-3
PROFIBUS connector ---- 8-8
PROFIBUS interface ---- 8-2
PROFIBUS transmission specifications ---- 8-4
programming examples
  217IF-01 to 217IF-01 ---- 3-25
  217IF-01 to GP-450 ---- 3-31
  217IF-01 to MELSEC ---- 3-36
  217IF-01 to OMRON ---- 3-41
  217IF-01 to SRmini ---- 3-45
  217IF-01 to VS-616G5 ---- 3-50
  218IF-01 to 218IF-01 ---- 4-37
  218IF-01 to MELSEC ---- 4-42
  218IF-02 to 218IF-02 ---- 5-38
  218IF-02 to MELSEC ---- 5-43
  260IF-01 explicit messages ---- 7-38
  MPLINK message communication ---- 9-44

R

random-access buffer communication ---- 4-28, 5-28
register map ---- 3-7, 4-30, 5-30
RS-232C cable ---- 4-9, 5-11, 6-10, 7-15, 9-13
RS-232C connector ---- 9-11
RS-232C transmission specifications - 4-4, 5-6, 6-4, 7-9, 8-4, 9-6
RS-422 wiring ---- 6-11
RS-422/485 ---- 6-2
RS-422/485 cable ---- 6-10
RS-422/485 connector ---- 6-10
RS-422/485 transmission specifications ---- 6-4
RS-485 wiring ---- 6-12

S

self-configuration ---- 1-26
serial communication port
  setting ---- 1-18
socket communication ---- 4-43, 5-44
subnet mask ---- 4-17, 5-16

T

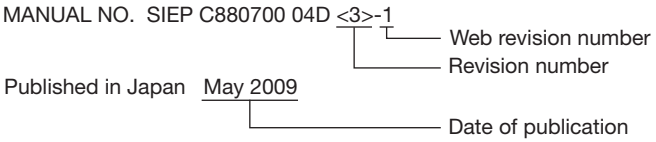
T-branch connections ---- 7-16
T-branch connector ---- 9-12
TCP ---- 4-15, 5-19
token cycle time ---- 9-32
token holding time ---- 9-32
token passing ---- 9-32
transmission
  time required ---- 3-20

U

UDP ---- 4-15, 5-19
unpassive open mode ---- 4-14, 5-18

Revision History

The date of publication, revision number, and web revision number are given at the bottom right of the back cover. Refer to the following example.



Date of Publication	Rev. No.	Web Rev. No.	Section	Revised Content
July 2025	<21>	3	4.1.5	Partly revised
January 2025		2	3.5.5 (6), 5.7.4 (1), 5.7.7 (1), 5.8.7 (1)	Partly revised
June 2024		1	7.4.7 (3), 3.5.5 (5)	Partly revised
			Back cover	Revision: Address
November 2022		0	–	Printed version of the manual that is available on the web (web version: SIEP C880700 04P<20>-2)
			6.2.1 (2)	Addtion: Terminator value (120 Ω)
			6.2.3 (2)	Addition: Notes on wiring with a 2-wire system
			Back cover	Revision: Address
December 2021	<20>	2	C.2 (3), (4)	Revision: Information on “No. of data items” and “Contents of the first holding register”
October 2021		1	7.4.3	Addition: I/O status code
			Back cover	Revision: Address
March 2021		0	–	Printed version of the manual that is available on the web (web version: SIEP C880700 04O<19>-2)
July 2020	<19>	2	3.6.2 (1), 3.7.2 (1)	Partly revised
May 2020		1	1.2.3, 1.2.4, 1.2.5, Chapter 3, 4.2.2, Chapter 5, 6.2.2, 7.2.2, 8.2.2, 9.2.2, 9.8.5	Partly revised
			5.4.2	Addition: Information related to Receive Buffer Selection
			Back cover	Revision: Address
October 2019		0	All chapters	Revision: Operation procedures for MPE720 Ver.7
April 2019	<18>	1	9.2.2 (3)	Revision: Terminator Changed from “75 W ± 1%” to “75 Ω ± 1%”
March 2019		0	–	Printed version of the manual that is available on the web (web version: SIEP C880700 04M<17>-1)
February 2019	<17>	1	5.1.3 (2)	Revision: Ethernet Transmission Specifications Changed from “MEMOBUS/TCP” to “MODBUS/TCP”
			Back cover	Revision: Address
October 2018		0	–	Printed version of the manual that is available on the web (web version: SIEP C880700 04L<16>-1)
			Back cover	Revision: Address
July 2018	<16>	1	6.2.1	Revision: Connector Specifications
February 2018		0	4.1.3 (3), 5.1.3 (3), 6.1.3 (3), 7.1.4 (3), 8.1.3 (3), 9.1.3 (3)	Revision: Noise resistance
January 2018	<15>	0	1.2.2	Revision: Applicable Machine Controllers
			1.2.3	Addition: Supported CPU and MPE720 Versions
			3.7.7	Revision: PARAM11, PARAM12
			4.1.3 (3), 5.1.3 (3), 6.1.3 (3), 7.1.4 (3), 8.1.3 (3), 9.1.3 (3)	Revision: Pollution Level, Noise Resistance
			4.1.6, 5.1.6, 6.1.6, 8.1.6	Deletion: Offline Self-diagnostic Test
			5.1.5, 7.1.6	Revision: Switch Settings
May 2017	<14>	0	–	Printed version of the manual that is available on the web (web version: SIEP C880700 04L<13>-1)
			Back cover	Revision: Address

Date of Publication	Rev. No.	Web Rev. No.	Section	Revised Content
April 2017	<13>	1	6.1.4	Revision: LED indicator (RX → TRX)
			Back cover	Revision: Address
December 2015		0	–	Printed version of the manual that is available on the web (web version: SIEP C880700 04K<12>-1)
			Back cover	Revision: Address
June 2015	<12>	1	Front cover, back cover	Revision: Format
			6.1.6	Addition: Normal performance of offline self-diagnostic test
January 2015		0	–	Printed version of the user's manual that is available on the web (web version: SIEP C880700 04J<11>-1)
			Chapters 2 and 3	Addition: Information related to TOYOPUC and non-procedure FD protocol.
			Chapter 7	Revision: MSTB2-5/5-GF-5.08AM → MSTB2.5/5-GF-5.08AU
September 2014	<11>	1	7.5.1 (2), (3)	Revision: Baud Rate Setting 0 ON → OFF Baud Rate Setting 1 OFF → ON
October 2013		0	Back cover	Revision: Address
November 2012	<10>	0	–	Printed version of the user's manual that is available on the web (web version: SIEP C880700 04J<9>-1)
			–	Based on Japanese user's manual, SIJP C880700 04K<20>-0 published in October 2012.
			Back cover	Revision: Address
July 2012	<9>	1	2.1.1	Revision: Supported communication mode for MEMOBUS and OMRON
May 2012		0	–	Printed version of the user's manual that is available on the web (web version: SIEP C880700 04I<8>-1)
			3.5.1 (2)	Revision: Signal name of MP2000 series machine controller on pin number 6
			4.2.1 (2), 5.2.1 (2), 6.2.1 (2), 7.2.1 (2), 8.2.1 (2)	Revision: Signal name and description of RS-232C on pin number 9
April 2012	<8>	1	9.2.2 (3)	Revision: Manufacturer of twisted-pair cable
January 2012		0	–	SIEP C880700 04H<7>-1, available on the Web.
			Back cover	Revision: Address
September 2011	<7>	1	3.7.1, 3.7.2 (1) [c], 5.7.1, 5.7.2 (1) [c], 5.8.1, 5.8.2 (1) [c], 7.6.1, 7.6.2 (1) [c], 9.7.1, 9.7.2 (1) [c], 9.8.1, 9.8.2 (1) [c]	Revision: Transmission device type Ethernet (218IFB) → Ethernet (218IFA, 218IFB, 218IFC)
June 2011		0	–	SIEP C880700 04G<6>-1, available on the Web.
April 2011	<6>	1	7.4.6 (2)	Revision: Register number of slave
December 2010		0	–	SIEP C880700 04F<5>-4, available on the Web.
October 2010	<5>	4	Front cover, back cover	Revision: Format
			7.1.5	Addition: Indicator and status of MS and NS indicator
September 2010		3	1.2.2	Addition: CPU-03 and CPU-04
			3.6.1	Revision: Description of Inputs No.3, 5 and 6
			3.6.2 [c], [e]	Revision: Ethernet (218IFB) → Ethernet (218IFA, 218IFB, 218IFC)
			3.6.2 [f]	Addition: Ethernet (218IFA) Revision: Ethernet (218IFB) → Ethernet (218IFB, 218IFC)
June 2010		2	3.5.1 (6) [b]	Revision: Description of ladder programming example
			Back cover	Revision: Address
May 2010		1	9.2.2 (3)	Revision: Model of terminator
May 2010		0	7.1.2	Addition: Network configuration
February 2010	<4>	1	4.2.1 (2), 5.2.1 (2), 6.2.1 (2), 7.2.1 (2), 8.2.1 (2), 9.2.1 (2)	Revision: Description on Pin Number 9
January 2010		0	–	Based on Japanese user's manual, SIJP C880700 04F<13>-0 published in November 2009.
			All chapters	Completely revised.

Date of Publication	Rev. No.	Web Rev. No.	Section	Revised Content
November 2009	<3>	3	4.4.2	Addition: Caution to save parameter settings
			7.4.3	Addition: Description of I/O status code 4054H
September 2009		2	Preface	Revision: General precautions Addition: Warranty
			Back cover	Revision: Address
May 2009		1	3.5.3 (2)	Revision: Connection to MELSEC through RS-485 Communication
			8.1.3 (2)	Revision: GSD file name provided for Master configuration
			Back cover	Revision: Address
February 2008		0	All chapters	Completely revised. Addition: Description on 218IF-02 Module
February 2006	<2>	0	All chapters	Addition: Description on related to 215AIF-01 MPLINK Modules
			Chapter 1	Addition: Description related to 215AIF-01
			1.4	Addition: Section on communication process
			2.1.3 (4) [b], 2.2.3 (4), 2.2.6 (4), 2.3.3 (4), 2.4.3 (4), 2.5.3 (3) [b], 3.2.3 (3)	Addition: Description on Receive monitor time
			2.3.4 (3)	Addition: Section on device information
			2.3.6 (5)	Parameters updated
			2.3.7	Addition: Section on communication cycle time
			2.5	Addition: Section on 215AIF-01 Module
			3.2.1 (5)	Revision: To the new ladder programming example
			3.2.2 (5)	Revision: To the new ladder programming example
			3.2.4 (5)	Revision: To the new ladder programming example
			3.2.5 (5)	Revision: To the new ladder programming example
			4.2.1 (4)	Revision: To the new ladder programming example
			5.2.1 (5)	Revision: To the new ladder programming example
			Chapter 6	Addition: Chapter on MPLINK/CP-215 communication
			A.1.3 (3)	Addition: 260IF communication
			Appendix C	Programs completely revised
			Appendix D	Addition: Appendix on cable components
			Others	Slightly revised through all chapters and format updated
			Back cover	Revision: Address
November 2003	<1>	0	1.3.2 (4), 3.1.3	Addition: Section on OMRON protocol
			3.2.3	Addition: Section on connection to a OMRON PLC
			Back cover	Revision: Address
March 2003	–	–	–	First edition

Machine Controller MP2000 Series

Communication Module
USER'S MANUAL

IRUMA BUSINESS CENTER (SOLUTION CENTER)
480, Kamifujisawa, Iruma, Saitama, 358-8555, Japan
Phone: +81-4-2962-5151 Fax: +81-4-2962-6138
www.yaskawa.co.jp

YASKAWA AMERICA, INC.
2121, Norman Drive South, Waukegan, IL 60085, U.S.A.
Phone: +1-800-YASKAWA (927-5292) or +1-847-887-7000 Fax: +1-847-887-7310
www.yaskawa.com

YASKAWA ELÉTRICO DO BRASIL LTDA.
777, Avenida Piraporinha, Diadema, São Paulo, 09950-000, Brasil
Phone: +55-11-3585-1100 Fax: +55-11-3585-1187
www.yaskawa.com.br

YASKAWA EUROPE GmbH
Philipp-Reis-Str. 6, 65795 Hattersheim am Main, Germany
Phone: +49-6196-569-300 Fax: +49-6196-569-398
www.yaskawa.eu.com E-mail: info@yaskawa.eu.com

YASKAWA ELECTRIC KOREA CORPORATION
6F, 112, LS-ro, Dongan-gu, Anyang-si, Gyeonggi-do, Korea
Phone: +82-31-8015-4224 Fax: +82-31-8015-5034
www.yaskawa.co.kr

YASKAWA ASIA PACIFIC PTE. LTD.
30A, Kallang Place, #06-01, 339213, Singapore
Phone: +65-6282-3003 Fax: +65-6289-3003
www.yaskawa.com.sg

YASKAWA ELECTRIC (THAILAND) CO., LTD.
59, 1F-5F, Flourish Building, Soi Ratchadapisek 18, Ratchadapisek Road, Huaykwang, Bangkok, 10310, Thailand
Phone: +66-2-017-0099 Fax: +66-2-017-0799
www.yaskawa.co.th

YASKAWA ELECTRIC (CHINA) CO., LTD.
22F, Link Square 1, No.222, Hubin Road, Shanghai, 200021, China
Phone: +86-21-5385-2200 Fax: +86-21-5385-3299
www.yaskawa.com.cn

YASKAWA ELECTRIC (CHINA) CO., LTD. BEIJING OFFICE
Room 1011, Tower W3 Oriental Plaza, No.1, East Chang An Avenue,
Dong Cheng District, Beijing, 100738, China
Phone: +86-10-8518-4086 Fax: +86-10-8518-4082

YASKAWA ELECTRIC TAIWAN CORPORATION
12F, No. 207, Section 3, Beishin Road, Shindian District, New Taipei City 23143, Taiwan
Phone: +886-2-8913-1333 Fax: +886-2-8913-1513 or +886-2-8913-1519
www.yaskawa.com.tw

YASKAWA

YASKAWA ELECTRIC CORPORATION

In the event that the end user of this product is to be the military and said product is to be employed in any weapons systems or the manufacture thereof, the export will fall under the relevant regulations as stipulated in the Foreign Exchange and Foreign Trade Act. Therefore, be sure to follow all procedures and submit all relevant documentation according to any and all rules, regulations and laws that may apply.

Specifications are subject to change without notice for ongoing product modifications and improvements.

© 2003 YASKAWA ELECTRIC CORPORATION

MANUAL NO. SIEP C880700 04Q <21>-3

Published in Japan July 2025
24-10-20
Original instructions