MELSEC FX Series

Programmable Controllers

User's Manual

FX3U

MODBUS Serial Communication Edition

Art. no: JY997D26201
01042007
Version A

Safety Precautions

(Read these precautions before use.)

Before installing, operating, maintenance or inspecting this product, thoroughly read and understand this manual and the associated manuals. Also pay careful attention to handle the module properly and safety.

This manual classifies the safety precautions into two categories: ⚠DANGER and ⚠CAUTION.

⚠DANGER	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠CAUTION	Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage.

Depending on circumstances, procedures indicated by ⚠CAUTION may also be linked to serious results.
In any case, it is important to follow the directions for usage.
Store this manual in a safe place so that you can take it out and read it whenever necessary. Always forward it to the end user.

1. DESIGN PRECAUTIONS

⚠DANGER

- Provide a safety circuit on the outside of the PLC so that the whole system operates to ensure the safety even when external power supply trouble, PLC failure, or communication error occurs.
  Otherwise, malfunction or output failures may result in an accident.
  1) An emergency stop circuit, a protection circuit, an interlock circuit for opposite movements, such as normal and reverse rotations, and an interlock circuit for preventing damage to the machine at the upper and lower positioning limits should be configured on the outside of the PLC.
  2) When the PLC CPU detects an error, such as a watchdog timer error, during self-diagnosis, all outputs are turned off. When an error that cannot be detected by the PLC CPU occurs in an input/output control block, output control may be disabled.
     Design external circuits and mechanisms to ensure safe operations of the machine in such a case.
  3) The output current of the service power supply for sensor varies depending on the model and the absence/ presence of extension blocks. If overload is applied, the voltage automatically drops, inputs in the PLC are disabled, and all outputs are turned off.
     Design external circuits and mechanisms to ensure safe operations of the machine in such a case.
  4) When some sort of error occurs in a relay, triac or transistor of the output unit, output may be kept on or off.
     For output signals that may lead to serious accidents, design external circuits and mechanisms to ensure safe operations of the machine in such cases.

⚠CAUTION

- Do not bundle the control line together with the main circuit or power line. Do not lay the control line near them.As a rule, lay the control line at least 100mm(3.94") or more away from the main circuit or power line.
  Noise may cause malfunctions.
- Use the product in such a status that excessive force is not applied on the built-in programming board, power connectors, I/O connectors, communication connectors, and communication cables.
  Failure to do so may result in wire breakage or failure of the PLC.

2. WIRING PRECAUTIONS

⚠DANGER

- Cut off all phases of the power source externally before installation or wiring work in order to avoid electric shock or damage of product.
- Make sure to attach the terminal cover offered as an accessory to the product before turning on the power or starting the operation after installation or wiring work.
  Failure to do so may cause electric shock.

⚠CAUTION

- Make sure to observe the precautions below in order to prevent any damage to the machine or any accident which may be caused by abnormal data written to the PLC due to the influence of noise:
  1) Do not lay close or bundle with the main circuit line, high-voltage line, or load line.
     Otherwise, effects of noise or surge induction are likely to take place.
     Keep a safe distance of least 100 mm (3.94") from the above lines during wiring.
  2) Ground the shield wire or shield of a shielded cable at one point on the PLC. However, do not ground at the same point as high voltage lines.
- Observe the following items to wire the lines to the European terminal board. Ignorance of the following items may cause electric shock, short circuit, disconnection, or damage of the product.
  - The disposal size of the cable end should follow the dimensions described in this manual.
  - Tightening torque should follow the torque described in this manual.
  - Twist the end of strand wire and make sure there is no loose wires.
  - Do not solder-plate the electric wire ends.
  - Do not connect electric wires of unspecified size or beyond the specified number of electric wires.
  - Fix the electric wires so that the terminal block and connected parts of electric wires are not directly stressed.

3. STARTUP AND MAINTENANCE PRECAUTIONS

⚠DANGER

- Do not touch any terminal while the PLC's power is on.
  Doing so may cause electrical shock or malfunctions.
- Before cleaning or retightening terminals, externally cut off all phases of the power supply.
  Failure to do so may expose you to shock hazard.
- Before modifying the program under operation or performing operation for forcible output, running or stopping, carefully read the manual, and sufficiently ensure the safety.
  An operation error may damage the machine or cause accidents.
- Do not change programs in the PLC from two or more peripheral equipment (such as the programming tool and GOT) at the same time.
  Such changes may cause destruction or malfunction of programs in the PLC.

⚠CAUTION

- Do not disassemble or modify the PLC.
  Doing so may cause failures, malfunctions or fire.
  For repair, contact your local Mitsubishi Electric distributor.
- Before connecting or disconnecting any extension cable, turn off power.
  Failure to do so may cause unit failure or malfunctions.
- Make sure to turn off the power before attaching or removing the peripheral equipment, expansion board, special adaptor, or function extension memory cassette.
  Failure to do so may cause device failure or malfunctions.

FX3U Series Programmable Controllers

User's Manual

[MODBUS Serial Communication Edition]

Manual number	JY997D26201
Manual revision	A
Date	4/2007

Foreword

This manual explains the "MODBUS serial communication" provided in FX3U Series Programmable Controllers and should be read and understood before attempting to install or use the unit.
Store this manual in a safe place so that you can take it out and read it whenever necessary. Always forward it to the end user.

Outline Precautions

- This manual provides information for the use of the FX3U Series Programmable Controllers. The manual has been written to be used by trained and competent personnel. The definition of such a person or persons is as follows:
  1) Any engineer who is responsible for the planning, design and construction of automatic equipment using the product associated with this manual should be of a competent nature, trained and qualified to the local and national standards required to fulfill that role. These engineers should be fully aware of all aspects of safety with regards to automated equipment.
  2) Any commissioning or service engineer must be of a competent nature, trained and qualified to the local and national standards required to fulfill that job. These engineers should also be trained in the use and maintenance of the completed product. This includes being completely familiar with all associated documentation for the said product. All maintenance should be carried out in accordance with established safety practices.
  3) All operators of the completed equipment should be trained to use that product in a safe and coordinated manner in compliance to established safety practices. The operators should also be familiar with documentation which is connected with the actual operation of the completed equipment.

  Note: The term 'completed equipment' refers to a third party constructed device which contains or uses the product associated with this manual
- This product has been manufactured as a general-purpose part for general industries, and has not been designed or manufactured to be incorporated in a device or system used in purposes related to human life.
- Before using the product for special purposes such as nuclear power, electric power, aerospace, medicine or passenger movement vehicles, consult with Mitsubishi Electric.
- This product has been manufactured under strict quality control. However when installing the product where major accidents or losses could occur if the product fails, install appropriate backup or failsafe functions in the system.
- When combining this product with other products, please confirm the standard and the code, or regulations with which the user should follow. Moreover, please confirm the compatibility of this product to the system, machine and apparatus which a user is using.
- If in doubt at any stage during the installation of the product, always consult a professional electrical engineer who is qualified and trained to the local and national standards. If in doubt about the operation or use, please consult the nearest Mitsubishi Electric distributor.
- Since the examples indicated by this manual, technical bulletin, catalog, etc. are used as a reference, please use it after confirming the function and safety of the equipment and system. Mitsubishi Electric will accept no responsibility for actual use of the product based on these illustrative examples.
- This manual content, specification etc. may be changed without a notice for improvement.
- The information in this manual has been carefully checked and is believed to be accurate; however, if you have noticed a doubtful point, a doubtful error, etc., please contact the nearest Mitsubishi Electric distributor.

Registration

- MODBUS® is a registered trademark of Schneider Electric S.A.
- The company name and the product name to be described in this manual are the registered trademarks or trademarks of each company.

Table of Contents

Related manuals

For a detailed explanation of the MODBUS serial communication network in FX3U PLCs, refer to this manual. For hardware information and instructions on the PLC main unit, other special function units/blocks, etc., refer to the appropriate manuals.
For acquiring required manuals, contact the distributor from where your product was purchased.

⊙Essential manual
○Manual required depending on application
△Manual with additional manual for detailed explanation

		Manual name	Manual number	Contents	Model name code
Manuals for PLC main unit					
■FX3U PLC main unit					
△	Supplied with product	FX3U Series Hardware Manual	JY997D18801	I/O specifications, wiring and installation of the PLC main unit FX3U extracted from the FX3U Series User's Manual - Hardware Edition. For detailed explanation, refer to the FX3U Series User's Manual - Hardware Edition.	–
⊙	Additional Manual	FX3U Series User's Manual - Hardware Edition	JY997D16501	Details about the hardware including I/O specifications, wiring, installation and maintenance of the FX3U PLC main unit.	09R516
■Programming					
⊙	Additional Manual	FX3U/FX3UC Series Programming Manual - Basic & Applied Instruction Edition	JY997D16601	Items related to programming in PLCs including explanation of basic instructions, applied instructions and various devices in FX3U/FX3UC PLCs.	09R517
Manuals for MODBUS serial communication network					
△	Supplied with product	FX3U-232ADP-MB Installation Manual	JY997D26401	Handling procedures of the RS-232C communication special adapter. For MODBUS serial communication network, refer also to the FX3U Series User's Manual - MODBUS Serial Communication Edition. For N:N link, parallel link, computer link or no protocol communication by RS instructions, refer also to the FX Series User's Manual - Data Communication Edition.	–
△	Supplied with product	FX3U-485ADP-MB Installation Manual	JY997D26301	Handling procedures of the RS-485 communication special adapter. For MODBUS serial communication network, refer also to the FX3U Series User's Manual - MODBUS Serial Communication Edition. For N:N link, parallel link, computer link or no protocol communication by RS instructions, refer also to the FX Series User's Manual - Data Communication Edition.	–
⊙	Additional Manual	FX3U Series User's Manual - MODBUS Serial Communication Edition (this manual)	JY997D26201	Explains the MODBUS serial communication network in FX3U PLCs.	–
Manual for N:N link, parallel link, computer link and no protocol communication by RS instructions/FX2N-232IF					
○	Additional Manual	FX Series User's Manual - Data Communication Edition	JY997D16901	Details about simple N:N link, parallel link, computer link and no-protocol communication (RS instruction and FX2N-232IF).	09R715

Generic Names and Abbreviations Used in Manuals

Abbreviation/generic name	Name
Programmable controllers	
FX3U Series	Generic name of FX3U Series PLCs
FX3U PLC or main unit	Generic name of FX3U Series PLC main units
Expansion boards	
Expansion board	Generic name of expansion boards (The models shown below): FX3U-232-BD, FX3U-422-BD, FX3U-485-BD, FX3U-USB-BD, and FX3U-CNV-BD
Special adapters	
Special adapter	Generic name of special high speed I/O adapters, special communication adapters, and special analog adapters Connectable equipment may vary depending on the main unit. For connectable equipment, refer to the User's Manual - Hardware Edition of the main unit.
Special high speed I/O adapter	Generic name of special high speed I/O adapters (The models shown below): FX3U-2HSY-ADP and FX3U-4HSX-ADP
Special communication adapter	Generic name of special communication adapters (The models shown below): FX3U-232ADP-MB, FX3U-485ADP-MB, FX3U-232ADP and FX3U-485ADP
Special analog adapter	Generic name of special analog adapters (The models shown below): FX3U-4AD-ADP, FX3U-4DA-ADP, FX3U-4AD-PT-ADP, and FX3U-4AD-TC-ADP
Peripheral equipment	
Peripheral equipment	Generic name of programming software, handy programming panels, and display units
Programming tools	
Programming tool	Generic name of programming software and handy programming panels
Programming software	Generic name of programming software
GX Developer	Generic name of programming software packages SW□D5C-GPPW-J and SW□D5C-GPPW-E
FX-PCS/WIN(-E)	Generic name of programming software packages FX-PCS/WIN and FX-PCS/WIN-E
Handy programming panel (HPP)	Generic name of programming panels FX-20P(-E) and FX-10P(-E)
RS-232C/RS-422 converter	FX-232AW, FX-232AWC, and FX-232AWC-H
RS-232C/RS-485 converter	FX-485PC-IF-SET and FX-485PC-IF
Manuals	
FX3U Hardware Edition	FX3U Series User's Manual - Hardware Edition
Programming Manual	FX3U/FX3UC Series Programming Manual - Basic & Applied Instruction Edition
Communication Control Edition	FX Series User's Manual - Data Communication Edition
Analog Control Edition	FX3U/FX3UC Series User's Manual - Analog Control Edition
Positioning Control Edition	FX3U/FX3UC Series User's Manual - Positioning Edition

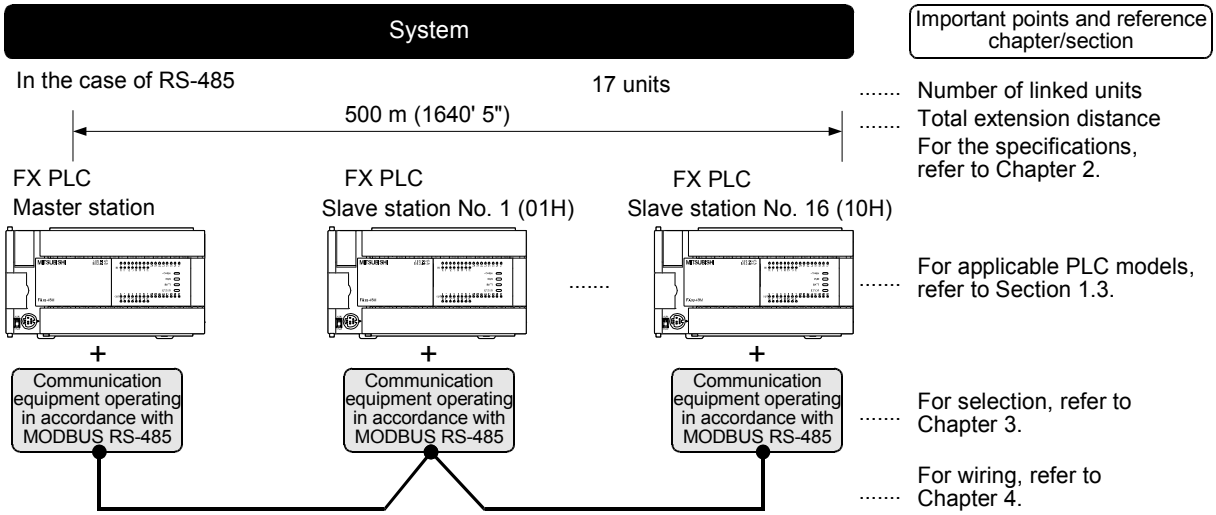
1. Outline

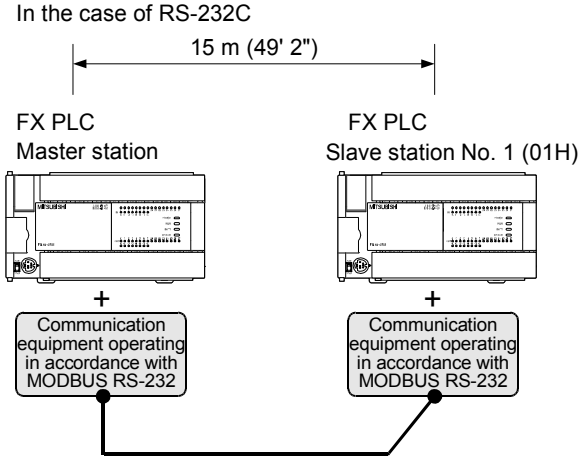
This chapter outlines the MODBUS serial communication network.

1.1 Outline of System

The MODBUS serial communication network allows up to 16 slaves to be controlled by one master in accordance with RS-485 to link devices, or one directly linked slave in accordance with RS-232C.

1) Up to 16 slaves can be controlled by one FX3U MODBUS Master
2) Master and Slave functionality
3) RTU and ASCII mode
4) One channel per PLC can be used for MODBUS serial communication (1 MODBUS Master channel or 1 MODBUS Slave channel)
5) Transmission speed up to 19.2 kbps
6) The MODBUS Master function uses a new PLC command dedicated to MODBUS serial communication



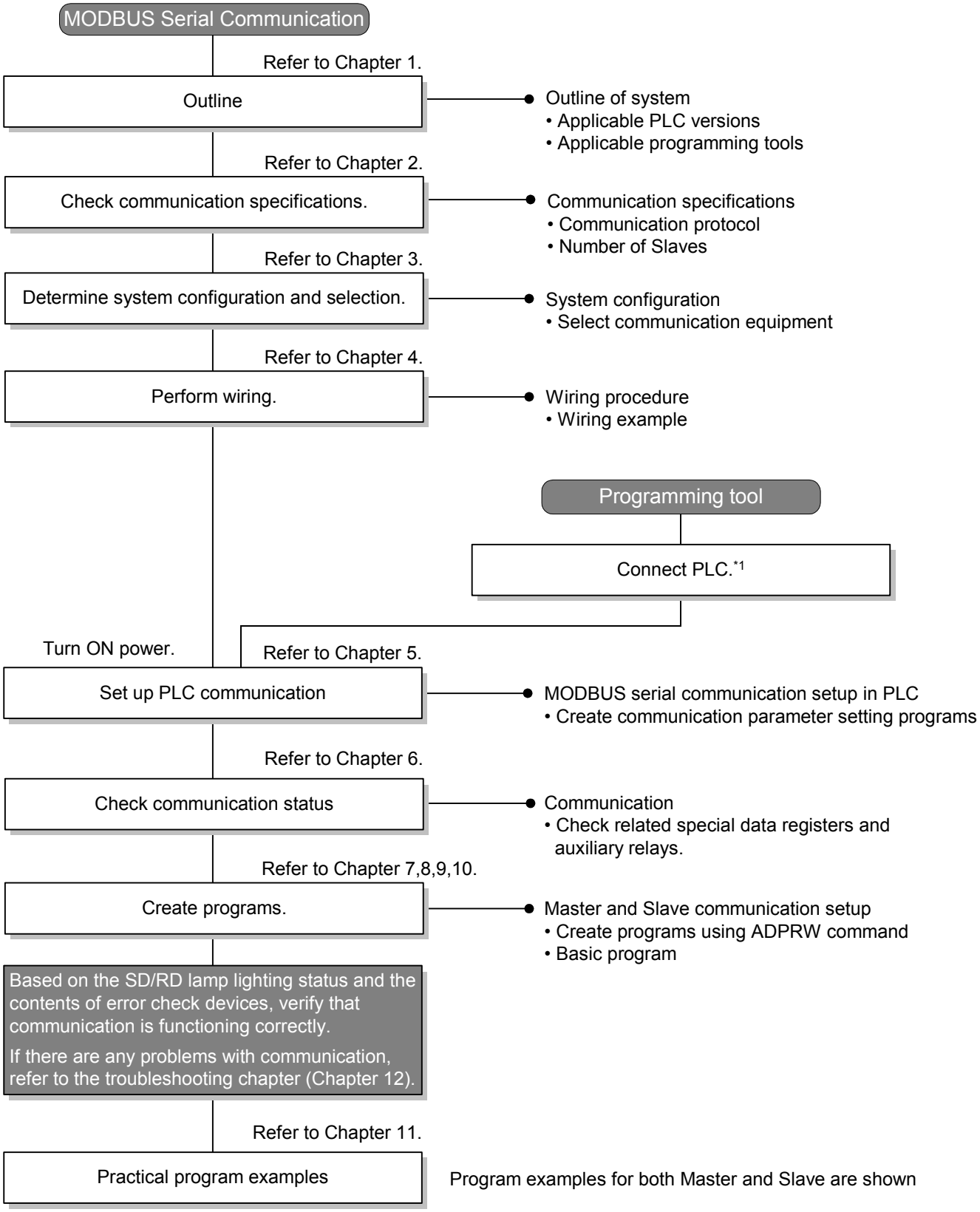


Note

MODBUS Slave Nodes do not need to be numbered in any specific order.

1.2 Major Procedures until Operation

The flow chart below shows the procedure for setting up the MODBUS serial communication network:



*1. For the corresponding programming tool to PLC connection method, refer to the "Programming Communication" Chapter of the FX Series Programmable Controller User's Manual - Data Communication Edition, or the corresponding programming tool manual.
For details on operating procedures, refer to the corresponding programming tool manual.

1.3 PLC Communication Type Applicability

1.3.1 Applicable versions

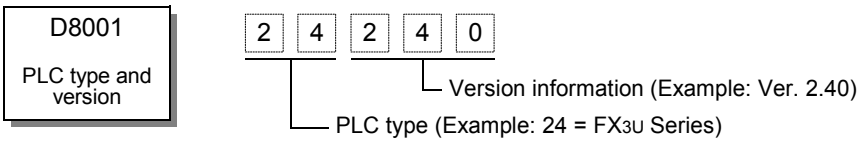
The communication type is applicable in the following versions.

✓: Applicable (If applicable versions are limited, they are described inside ( ).) —: Not applicable

PLC	Applicability (applicable version)	Remarks
FX3U Series	✓ (Ver. 2.40 or later)	

1. Version check

The D8001 (decimal) special data register contains information for determining the PLC version.



1.4 Programming Tool Applicability

1.4.1 For applicable versions

The programming tool is applicable for each FX Series from the following version:

1. Japanese versions

✓: Applicable (If applicable versions are limited, they are described inside ( ).) —: Not applicable

Model name (Media model name is shown below.)	Applicability (applicable version)	Remarks
FX3U PLCs		
GX Developer SW□D5C(F)-GPPW-J	✓ (Ver. SW8 X or later) Ver. 8.45X	Select the model "FX3U(C)".

2. English versions

✓: Applicable (If applicable versions are limited, they are described inside ( ).) —: Not applicable

Model name (Media model name is shown below.)	Applicability (applicable version)	Remarks
FX3U PLCs		
GX Developer SW□D5C(F)-GPPW-E	✓ (Ver. SW8 X or later) Ver. 8.45X	Select the model "FX3U(C)".

2. Specifications

This chapter explains the communication specifications and performance.

2.1 Communication Specifications

MODBUS Serial Communication can be implemented according to the specifications shown in the table below with the corresponding special adapter attached to the PLC, either the FX3U-232ADP-MB or FX3U-485ADP-MB. The communication format, protocol, etc. are determined by the communication setup using a sequence program, as described in Chapter 5.

Item		Specifications		Remarks
		FX3U-232ADP-MB	FX3U-485ADP-MB	
Channels per PLC		1 Channel		Either 1 MODBUS Master or 1 MODBUS Slave channel
Transmission Specifications	Communication Interface	RS-232C	RS-485	
	Transmission Speed	300, 600, 1200, 2400, 4800, 9600, or 19200 bps		
	Data Length	7-bit or 8-bit		
	Stop Bit	1-bit or 2-bit		
	Transmission Distance	Up to 15m (49'2")	Up to 500m (1640'5")	Transmission Distance varies depending on communication equipment type
	Communication Protocol	RTU or ASCII		
Master Function	Number of Slaves	1 Slave	16 Slaves	Number of Slaves varies depending on communication equipment type
	Number of Functions	14 (+14 Diagnostic functions)		
	Number of Simultaneously Executable Instructions	1 Instruction		
	Maximum Write Data	123 words or 1968 coils		
	Maximum Read Data	125 words or 2000 coils		
Slave Function	Number of Functions	14 (+14 Diagnostic functions)		
	Number of Simultaneous Acceptable Request Messages	1 Request Message		
	Station Number	1 to 247		
Special Adapter	External Dimensions	90 (H) x 17.6 (W) 74 (D) [mm]		
	Weight	80g		

Note

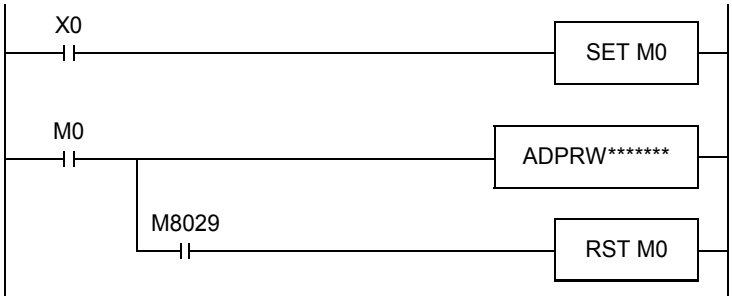
This table details the specifications when using MODBUS communication. The FX3U-485ADP-MB and FX3U-232ADP-MB also features FX3U-485ADP and FX3U-232ADP functionality respectively. Therefore the following communication types are also available when using the following modules:

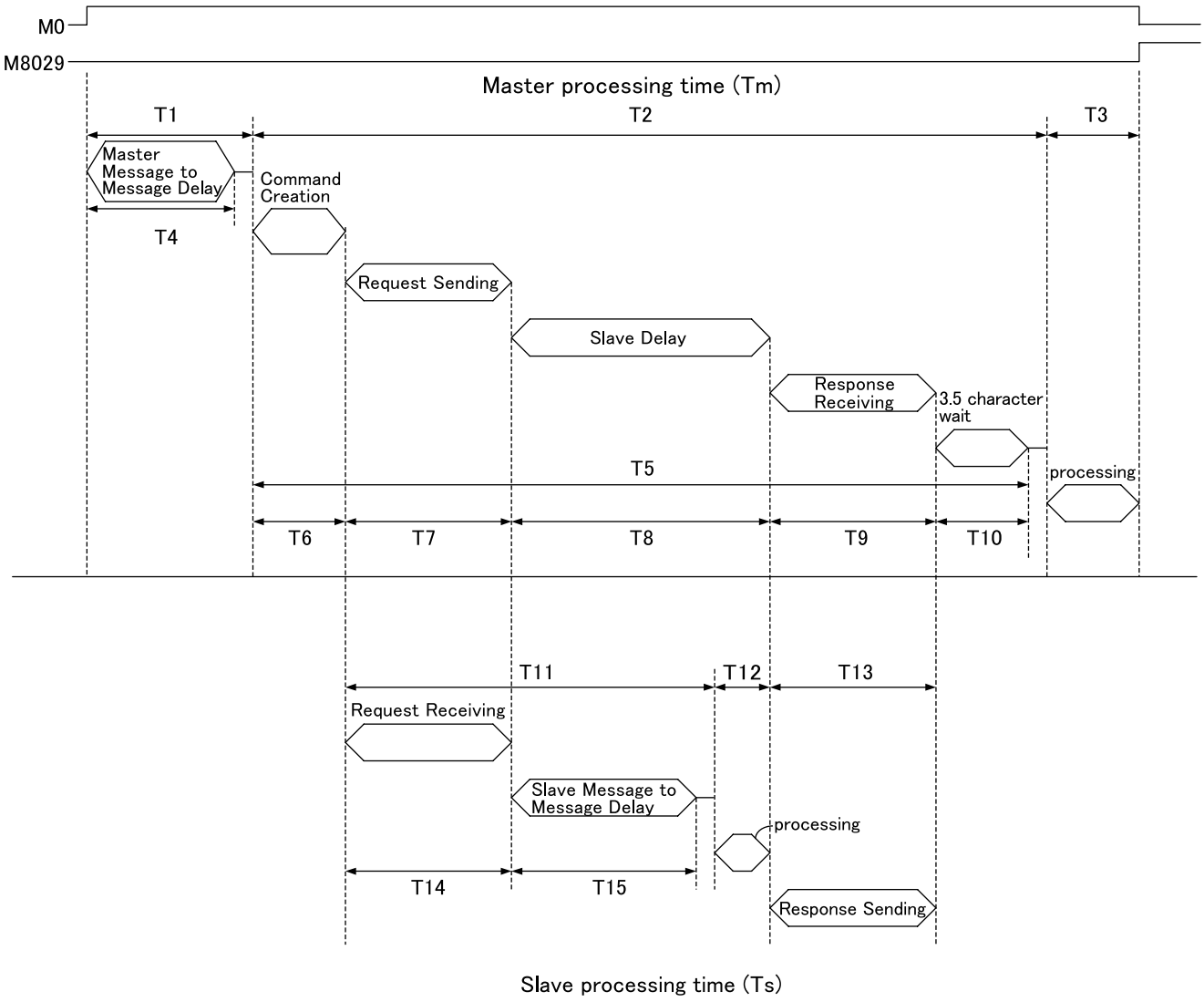
FX3U-485ADP	N:N network, Parallel link, Computer link, Non-protocol communication and Inverter communication.
FX3U-232ADP	Computer link, Non-protocol communication, Programming communication and Remote maintenance.

For further information on the communication types and please refer to the FX Series Programmable Controller User's Manual - Data Communication Edition.

2.2 Link Time

The link time indicates the cycle time in which a Master module completes a single command with a Slave, as illustrated by the diagram below.





The Master processing time (Tm) can be calculated in milliseconds (ms) as follows, where INT (n) indicates the concatenation of any remaining decimal values.

Character Length (bits):
Start bit (1bit) + Data Length (7bit or 8bit) + Parity (0bit or 1bit) + Stop bit (1bit or 2bit)

Tm = T1 + T2 + T3

$$T1 = \left(INT \left(\frac{T4}{Max\ Scan\ Time} \right) + 1 \right) * Max\ Scan\ Time$$

T4 = D8411 (or D8431, depending on the Communication Channel)

$$T2 = \left(INT \left(\frac{T5}{Max\ Scan\ Time} \right) + 1 \right) * Max\ Scan\ Time$$

T5 = T6 + T7 + T8 + T9 + T10

T6 = less than 1ms

$$T7 = \frac{Number\ of\ Bytes\ in\ Request\ *\ Character\ Length\ (bits)}{Baud\ Rate\ (bps)} * 1000\ (ms) + 1ms$$

T8 = Slave Delay Time (depending on the Slave)

$$T9 = \frac{Number\ of\ Bytes\ in\ Response\ *\ Character\ Length\ (bits)}{Baud\ Rate\ (bps)} * 1000\ (ms) + 1ms$$

RTU Mode:

$$T10 = \frac{3.5\ Characters\ *\ Character\ Length\ (bits)}{Baud\ Rate\ (bps)} * 1000\ (ms) + 1ms$$

ASCII Mode:
T10 = 0

T3 = less than 1ms

The Slave processing time (Ts) can be calculated in milliseconds (ms) as follows.

Character Length (bits):
Start bit (1bit) + Data Length (7bit or 8bit) + Parity (0bit or 1bit) + Stop bit (1bit or 2bit)

Ts = T11 + T12 + T13

T11 = T14 + T15 + Max Scan Time

$$T14 = \frac{Number\ of\ Bytes\ in\ Request\ *\ Character\ Length\ (bits)}{Baud\ Rate\ (bps)} * 1000\ (ms) + 1ms$$

T15 = D8411 (or D8431, depending on the Communication Channel)

T12 = less than 1ms

$$T13 = \frac{Number\ of\ Bytes\ in\ Response\ *\ Character\ Length\ (bits)}{Baud\ Rate\ (bps)} * 1000\ (ms) + 1ms$$

Example Link Time Calculations:

Master processing time (Tm)

D8411	= 5ms
Max Scan Time	= 5ms
Command	= Read Holding Registers 0-9 (Command Code 0x03)
Frame Mode	= RTU Mode
Bytes in Request	= 8bytes (1byte Address, 5byte Frame, 2byte CRC)
Bytes in Response	= 25bytes (1byte Address Echo, 22byte Frame, 2byte CRC)
Character Length	= 10bits (1bit Start, 8bit Data Length, 0bit Parity, 1bit Stop)
Baud Rate	= 19.2Kbps
Slave Delay	= 10ms

T4 = 5ms

$$T1 = \left(INT\left(\frac{5ms}{5ms}\right) + 1\right) * 5ms = (1 + 1) * 5ms = 10ms$$

T6 ≈ 1ms

$$T7 = \frac{8\ Bytes\ in\ Request * 10bits}{19200bps} * 1000\ (ms) + 1ms \approx 5.2ms$$

T8 = 10ms

$$T9 = \frac{25\ Bytes\ in\ Response * 10bits}{19200bps} * 1000\ (ms) + 1ms \approx 14.0ms$$

$$T10 = \frac{3.5\ Characters * 10bits}{19200bps} * 1000\ (ms) + 1ms \approx 2.8ms$$

T5 = 1ms + 5.2ms + 10ms + 14.0ms + 2.8ms = 33ms

$$T2 = \left(INT\left(\frac{33ms}{5ms}\right) + 1\right) * 5ms = (6 + 1) * 5ms = 35ms$$

T3 ≈ 1ms

Tm = 5ms + 35ms + 1ms = <u>41ms</u>

Slave processing time (Ts)

Command	= Read Holding Registers 0-9 (Command Code 0x03)
Frame Mode	= RTU Mode
Bytes in Request	= 8bytes (1byte Address, 5byte Frame, 2byte CRC)
Bytes in Response	= 25bytes (1byte Address Echo, 22byte Frame, 2byte CRC)
Character Length	= 10bits (1bit Start, 8bit Data Length, 0bit Parity, 1bit Stop)
Baud Rate	= 19.2Kbps
D8411	= 5ms
Max Scan Time	= 5ms

$$T14 = \frac{8\ Bytes\ in\ Request * 10bits}{19200bps} * 1000\ (ms) + 1ms \approx 5.2ms$$

T15 = 5ms

T11 = 5.2ms + 5ms + 5ms = 15.2ms
T12 ≈ 1ms

$$T13 = \frac{25\ Bytes\ in\ Response * 10bits}{19200bps} * 1000\ (ms) + 1ms \approx 14.0ms$$

Ts = 15.2ms + 1ms + 14.0ms = <u>30.2ms</u>

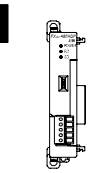
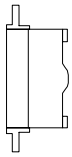
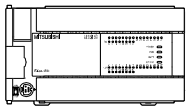
3. System Configuration

This section explains the configuration of communication equipment operating in accordance with RS-485 and RS-232C and the selection of equipment required by FX3U PLCs.

3.1 System Configuration

This section outlines the system configuration required to use MODBUS serial communication.

1 indicates the combination of communication equipment pattern type.

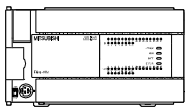
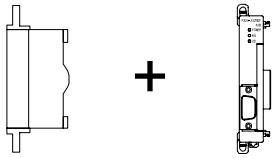
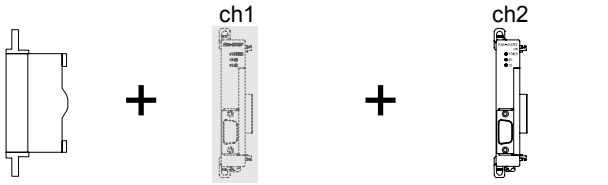
Communication equipment operating in accordance with MODBUS RS-485 or RS-232C	FX3U PLC	Important point in selection	Total extension distance
1  Communication adapter +  Special adapter connection board + 		Attach the special adapter connection board to the main unit, and then attach the communication adapter on the left side of the main unit.	RS-485: 500 m (1640' 5") RS-232C: 15 m (49' 2")

For combinations of communication equipment for each FX3U Series, refer to the next page.

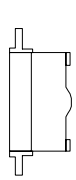
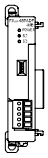
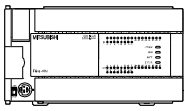
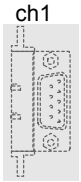
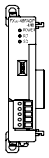
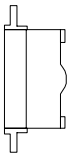
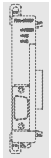
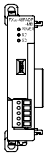
3.2 Applicable FX PLC and Communication Equipment

Select the most suitable combination of (optional) communication equipment from the table below, and put a check mark in the "Check" column of the corresponding equipment.

For communication in accordance with RS-232C

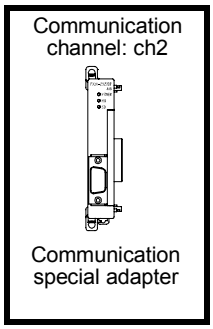
FX Series	Communication equipment (option)	Total extension distance	Check
 FX3U	**When using channel 1 (ch 1)**		
	 FX3U-CNV-BD + FX3U-232ADP-MB (9-pin D-Sub, male)	15 m (49' 2")	
	When using channel 2 (ch 2)		
	 FX3U-☐-BD (Where ☐ represents either 232, 422, 485 or USB) + FX3U-232ADP-MB (9-pin D-Sub, male)	15 m (49' 2")	
	 FX3U-CNV-BD + FX3U-☐ADP (-MB) (Where ☐ represents either 232 or 485) + FX3U-232ADP-MB (9-pin D-Sub, male)	15 m (49' 2")	

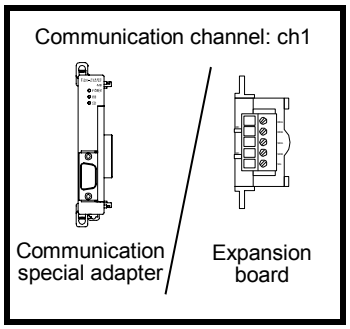
For communication in accordance with RS-485

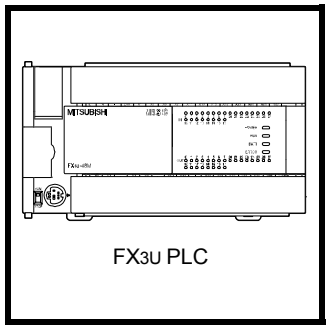
FX Series	Communication equipment (option)	Total extension distance	Check
	When using channel 1 (ch 1)		
	 FX3U-CNV-BD + ch1  FX3U-485ADP-MB (European terminal block)	500 m (1640' 5")	
	When using channel 2 (ch 2)		
 FX3U	ch1  FX3U-□-BD (Where □ represent either 232, 422, 485, or USB). + ch2  FX3U-485ADP-MB (European terminal block)	500 m (1640' 5")	
	ch1 ch2  FX3U-CNV-BD +  FX3U-□ADP (-MB) (Where □ represents either 232 or 485). +  FX3U-485ADP-MB (European terminal block)	500 m (1640' 5")	

3.3 Limitation when Channel 1 and Channel 2 are used at the same time

When using Channel 1 (ch1) and Channel 2 (ch2) at the same time, the available communication type combinations are limited. For more details, refer to the table below.







		Communication Setting on Ch1								
		MODBUS Serial Communication	N:N Networking	Parallel Link	Computer Link	Inverter Communication	Non-protocol Communication (RS instruction)	Non-protocol Communication (RS2 instruction)	Programming Communication	Remote Maintenance
Communication Setting on Ch2*1	MODBUS Serial Communication	- (Example 1)	✓	✓	✓	✓	✓	✓	✓	✓
	N:N Networking	✓ (Example 2)	-	-	✓	✓	✓	✓	✓	✓
	Parallel Link	✓	-	-	✓	✓	✓	✓	✓	✓
	Computer Link	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Inverter Communication	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Non-protocol Communication (RS2 instruction)	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Programming Communication	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Remote Maintenance*2	✓	✓	✓	✓	✓	✓	✓	✓	-

✓: Applicable

-: Not applicable

*1. Ch2 cannot be set for non-protocol communication using the RS instruction.

*2. When using remote maintenance on ch2, use GX Developer Ver. 8.18U or later.

Example1:
When using "MODBUS Serial Communication" on ch1, "MODBUS Serial Communication" can not be set on ch2.

Example2:
When using "MODBUS Serial Communication" on ch1, "N:N Networking" can be used on ch2.

→ **For more details on using N:N Networking and MODBUS Serial Communication, refer to Section 5.3.**

4. Wiring

This chapter explains the wiring.

WIRING PRECAUTIONS **DANGER**

- Cut off all phases of the power supply externally before installation or wiring work in order to avoid damage to the product or electric shock.
- Make sure to attach the terminal cover, offered as an accessory, before turning on the power or initiating operation after installation or wiring work.
  Failure to do so may cause electric shock.

WIRING PRECAUTIONS **CAUTION**

- Make sure to observe the following precautions in order to prevent any damage to the machinery or accidents due to abnormal data written to the PLC under the influence of noise:
  1) Do not bundle the main circuit line together with or lay it close to the main circuit, high-voltage line, or load line.
     Otherwise, noise disturbance and/or surge induction are likely to take place.
     As a guideline, lay the control line at least 100mm (3.94") or more away from the main circuit, high-voltage line, or load line.
  2) Ground the shield wire or shield of the shielded cable at one point on the PLC. However, do not ground them at the same point as the high-voltage lines.
- Make sure to properly wire the FX Series terminal blocks in accordance with the precautions below in order to prevent electric shock, a short-circuit, wire breakage, or damage to the product:
  - Tightening torque should be between 0.5 and 0.8 N•m.
- Observe the following items when wiring to the European terminal board.
  Failure to do so may cause electric shock, a short-circuit, disconnection, or damage to the product.
  - The disposal size of the cable end should follow the dimensions described in this manual.
  - Tightening torque should be between 0.22 and 0.25 N•m.
  - Twist the end of strand wire and make sure that there are no loose wires.
  - Do not solder-plate the electric wire ends.
  - Do not connect more than the specified number of wires or electric wires of unspecified size.
  - Affix the electric wires so that neither the terminal block nor the connected parts are directly stressed.

4.1 Wiring Procedure

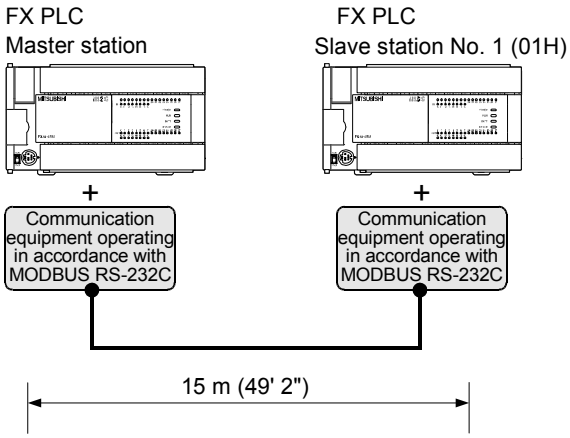
1) Selecting the connection method
   Select the wiring method suitable to the application.
   → **For details, refer to Section 4.2.**
2) Preparing for wiring
   Prepare cables and terminal resistors required for wiring.
   → **For details, refer to Section 4.3.**
3) Turning OFF the power to the PLC
   Before wiring, make sure that the PLC power is OFF.
4) Wiring communication equipment
   Connect communication equipment operating in accordance with MODBUS RS-485 or MODBUS RS-232C.
   → **For communication in accordance with MODBUS RS-232C, refer to Section 4.4.**
   → **For communication in accordance with MODBUS RS-485, refer to Section 4.5.**

4.2 Selecting Connection Method

When using MODBUS serial communication, communication can be achieved in accordance with MODBUS RS-232C or RS-485. For the FX3U, only one channel can be used for MODBUS serial communication, Master or Slave.

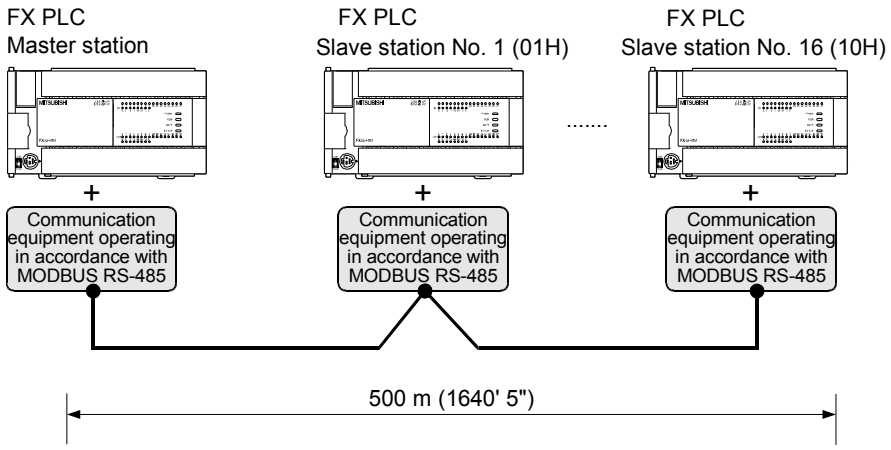
4.2.1 For communication in accordance with MODBUS RS-232C (1-to-1 connection)

For communication in accordance with MODBUS RS-232C, 1-to-1 connection is possible. Make sure that the total extension distance is 15m (49'2") or less.



4.2.2 For communication in accordance with MODBUS RS-485 (1-to-N connection)

For communication in accordance with MODBUS RS-485, up to 16 PLC Slave Nodes can be connected to one Master Node. Make sure that the total extension is 500m (1640'5") or less.



Note

MODBUS Slave Nodes do not need to be numbered in any specific order.

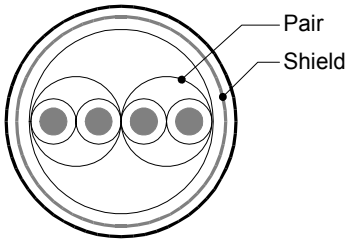
4.3 Selecting Cables and Terminal Resistors (MODBUS RS-485)

Select cables using the procedure described below.

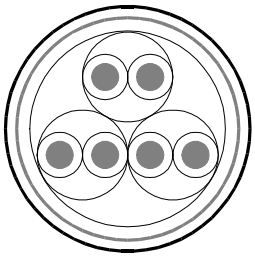
4.3.1 Twisted pair cable

Use shielded twisted pair cables for connecting communication equipment operating in accordance with MODBUS RS-485.

1. Cable structural drawing (reference)



Example of two-pair cable structural drawing



Example of three-pair cable structural drawing

2. RS-485 cable specifications

Item	Description
Cable type	Shielded cable
Number of pairs	2p, 3p
Conductor resistance (20°C)	88.0 Ω/km or less
Insulation resistance	10000 MΩ-km or more
Dielectric withstand voltage	500VDC, 1minute
Electrostatic capacitance (1 kHz)	60nF/km or less by an average
Characteristic impedance (100 kHz)	110±10 Ω

4.3.2 Connecting cables

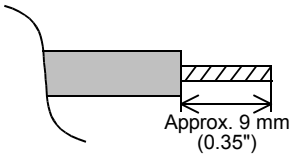
1. European type terminal block

Use shielded twisted pair cables for connecting communication equipment operating in accordance with MODBUS RS-485.
The table below shows applicable cables and tightening torques.

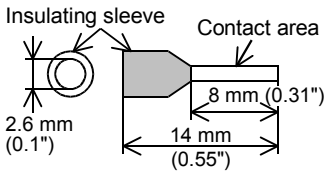
	Cable size when one cable is connected	Cable size when two cables are connected	Cable size for bar terminal with insulating sleeve	Tightening torque	Tool size	
					A	B
FX3U-485ADP-MB	AWG22 to AWG20	AWG22	AWG22 to AWG20	0.22 to 0.25 N·m	0.4 (0.01")	2.5 (0.09")

The cable ends, either stranded or solid, should be left as it is, or used with a bar terminal with insulating sleeve.

- When leaving the cable end as it is
  - Twist the end of stranded cables so that the individual wires do not poke out
  - Do not plate the cable end

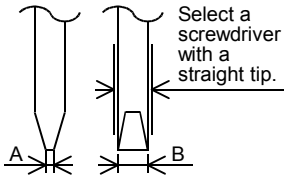


- When using a bar terminal with insulating sleeve
  Because it is difficult to insert a cable into the insulating sleeve depending on the cable sheath thickness, select the proper cable according to the outline drawing.



Manufacturer	Model name	Caulking tool
Phoenix Contact	AI 0.5-8WH	CRIMPFOX UD6

- Tool
  - When tightening a terminal on the European terminal block, use a small straight-shaped commercial screwdriver, as shown in the figure on the right.



For sizes A and B, refer to the table on the left.

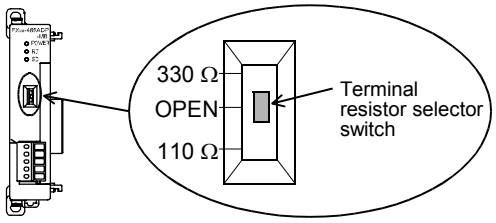
Manufacturer	Model name
Phoenix Contact	SZS 0.4 × 2.5

4.3.3 Connecting terminal resistors

Make sure to provide a terminal resistor at each end of the line.

1. When using the FX3U-485ADP-MB

The FX3U-485ADP-MB has a built-in terminal resistor.
Set the terminal resistor selector switch accordingly.



4.4 Connection Diagram for MODBUS RS-232C

Representative wiring examples are shown in this section. When pin numbers in the counterpart equipment are different, wire the pins as shown below.

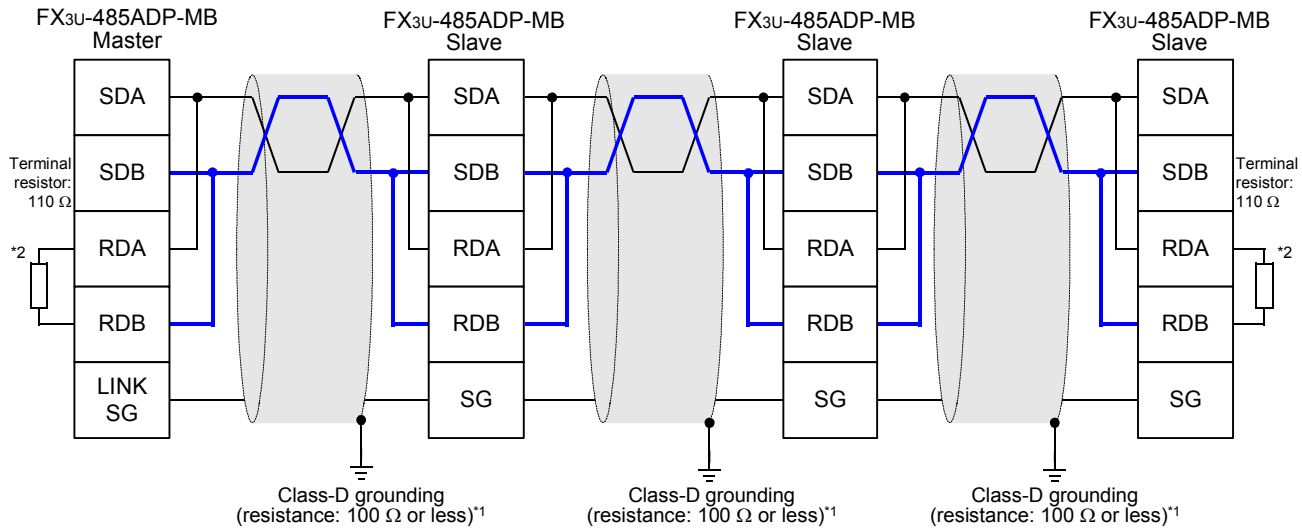
4.4.1 Connection diagram between FX PLC and MODBUS RS-232C equipment

PLC side			External equipment operating in accordance with MODBUS RS-232C					
Name	FX3U-232ADP-MB 9-pin D-Sub		Name	When CS and RS are used		Name	When DR and ER are used	
				9-pin D-Sub	25-pin D-Sub		9-pin D-Sub	25-pin D-Sub
FG	–		FG	–	1	FG	–	1
RD(RXD)	2		RD(RXD)	2	3	RD(RXD)	2	3
SD(TXD)	3		SD(TXD)	3	2	SD(TXD)	3	2
ER(DTR)	4		RS(RTS)	7	4	ER(DTR)	4	20
SG(GND)	5		SG(GND)	5	7	SG(GND)	5	7
DR(DSR)	6	*1	CS(CTS)	8	5	DR(DSR)	6	6

*1. For third-party external equipment requiring the Control Signal, connect these pins. The FX3U-232ADP-MB does not require these pins to be connected.

4.5 Connection Diagram for MODBUS RS-485

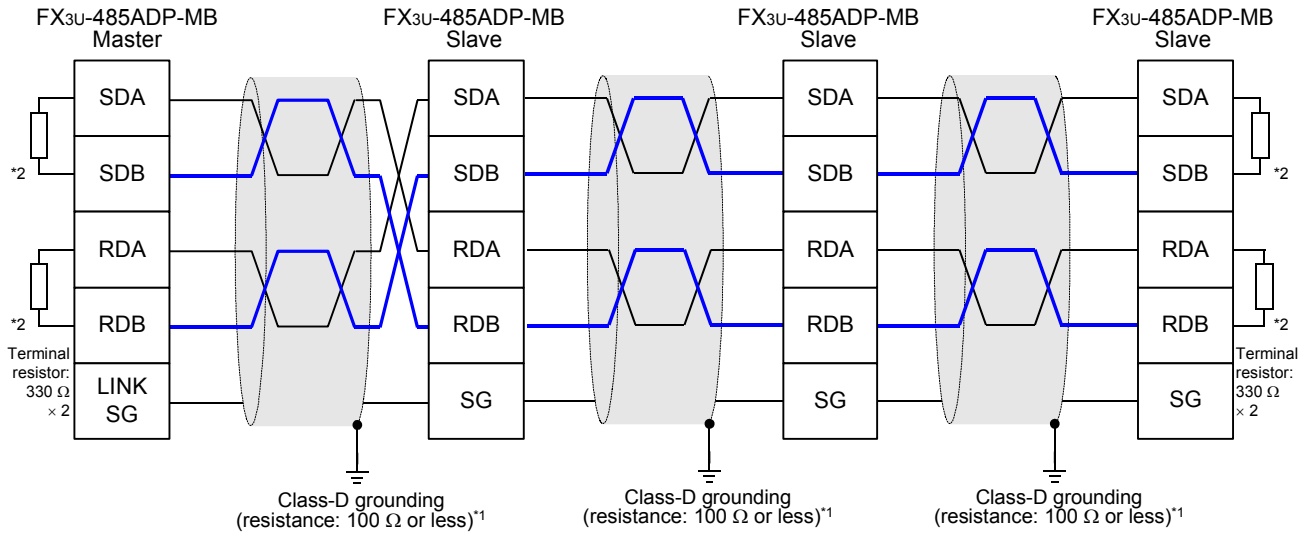
4.5.1 One-pair wiring



*1 Make sure to perform Class-D grounding on the shield of the twisted pair cable connected to the FX3U-485ADP-MB.

*2 Make sure to provide a terminal resistor at each end of a line.
- The FX3U-485ADP-MB has a built-in terminal resistor.
  Set the terminal resistor selector switch accordingly.

4.5.2 Two-pair wiring



*1 Make sure to perform Class-D grounding on the shield of the twisted pair cable connected to the FX3U-485ADP-MB.

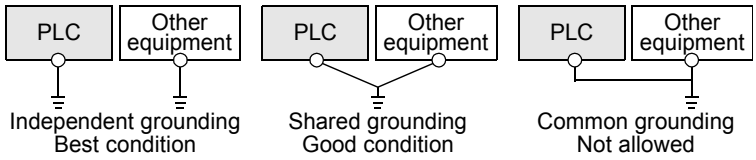
*2 Make sure to provide a terminal resistor at each end of a line.
- The FX3U-485ADP-MB has a built-in terminal resistor.
  Set the terminal resistor selector switch accordingly.

4.6 Grounding

Grounding should be performed as stated below.

- The grounding resistance should be 100 Ω or less.
- Independent grounding should be performed for best results.
  When independent grounding can not be performed, perform "shared grounding" as shown in the following figure

→ For details, refer to the Hardware Edition.



- The grounding wire size should be AWG 14 (2 mm^2) or larger.
- The grounding point should be close to the PLC, and all grounding wires should be as short as possible.

5. Communication Setup

This chapter explains the setup method for using the MODBUS protocol with an FX3U PLC and MODBUS Communication ADP.

5.1 Setup method for MODBUS serial communication

MODBUS communication is setup via the PLC program using GX Developer.

5.2 Example of communication setup

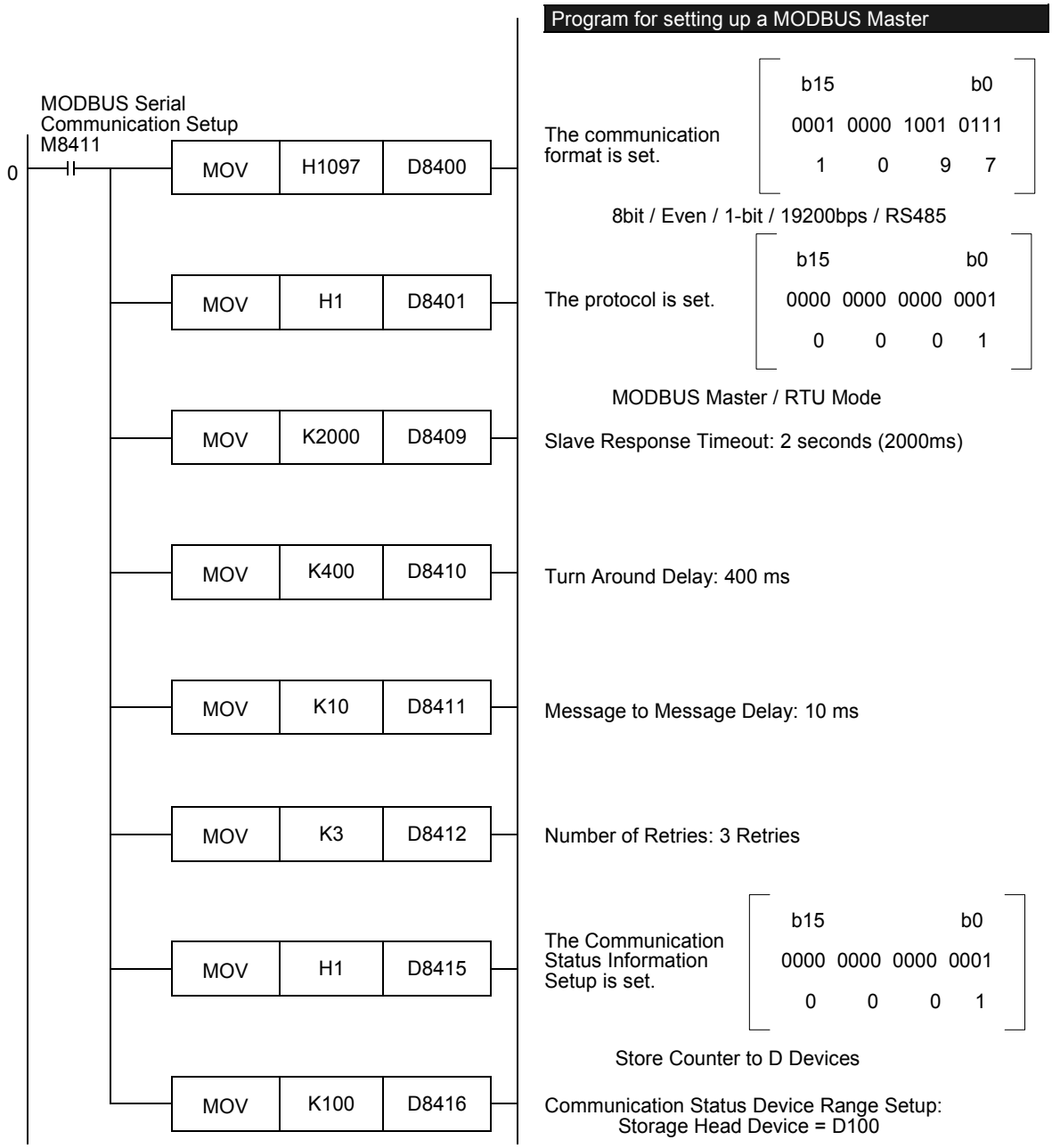
To initiate the setup, the PLC program must use the auxiliary relay M8411 for Channel 1 or Channel 2. When the PLC program contains the "LD M8411" instruction, it is then possible to configure the MODBUS functionality using MOV operations.

The communication parameters for MODBUS communication can be setup using the following ladder code:

Parameters for a Master using Channel 1:

Device	Name	Description
D8400	Communication Format	For Descriptions, refer to Chapter 6.
D8401	Protocol	
D8409	Slave Response Timeout	
D8410	Turn Around Delay	
D8411	Message to Message Delay	
D8412	Number of Retries	
D8415	Communication Status Information Setup	
D8416	Communication Status Device Range Setup	

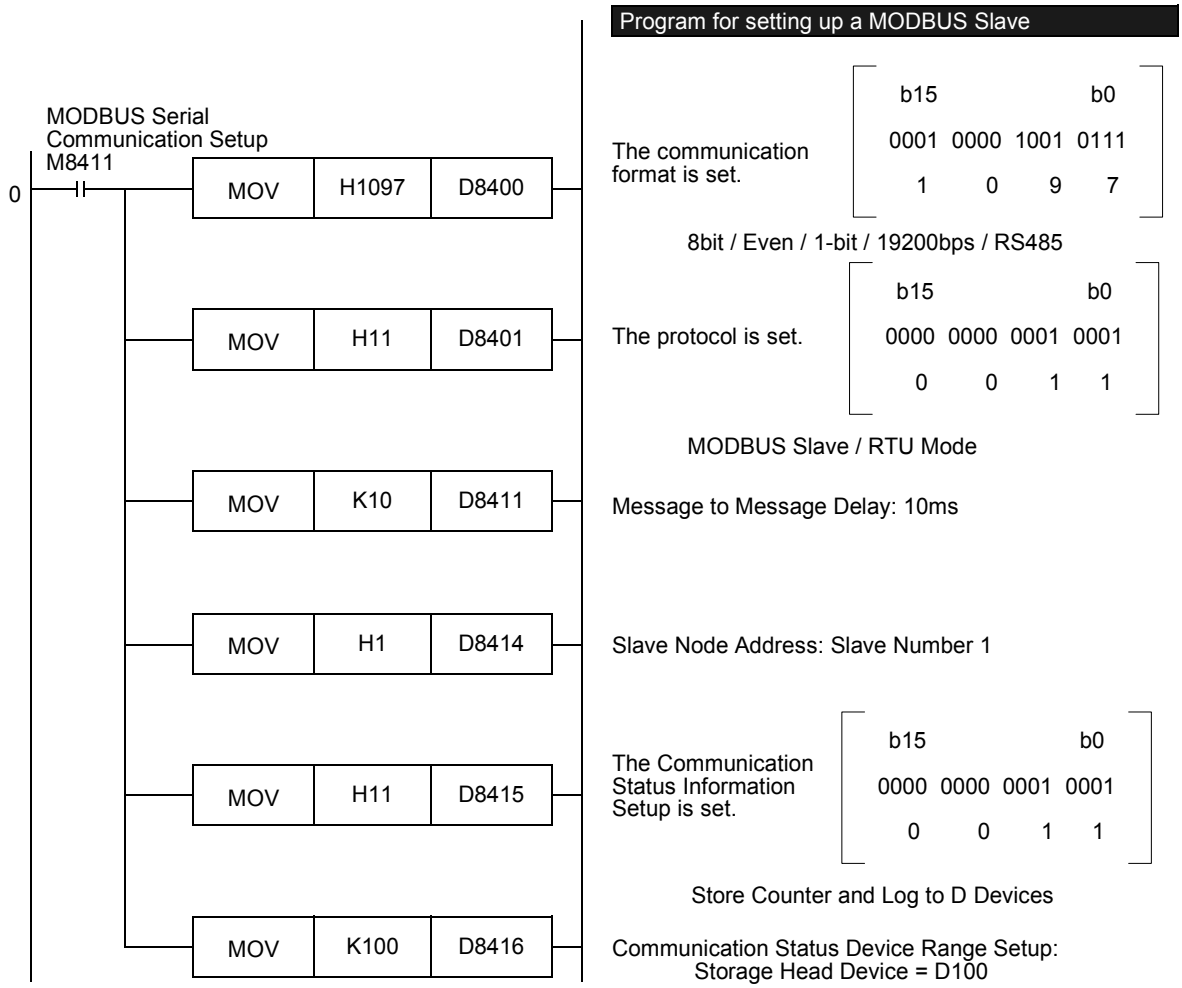
The ladder code for setting the Master parameters can be expressed as follows:



Parameters for a Slave using Channel 1:

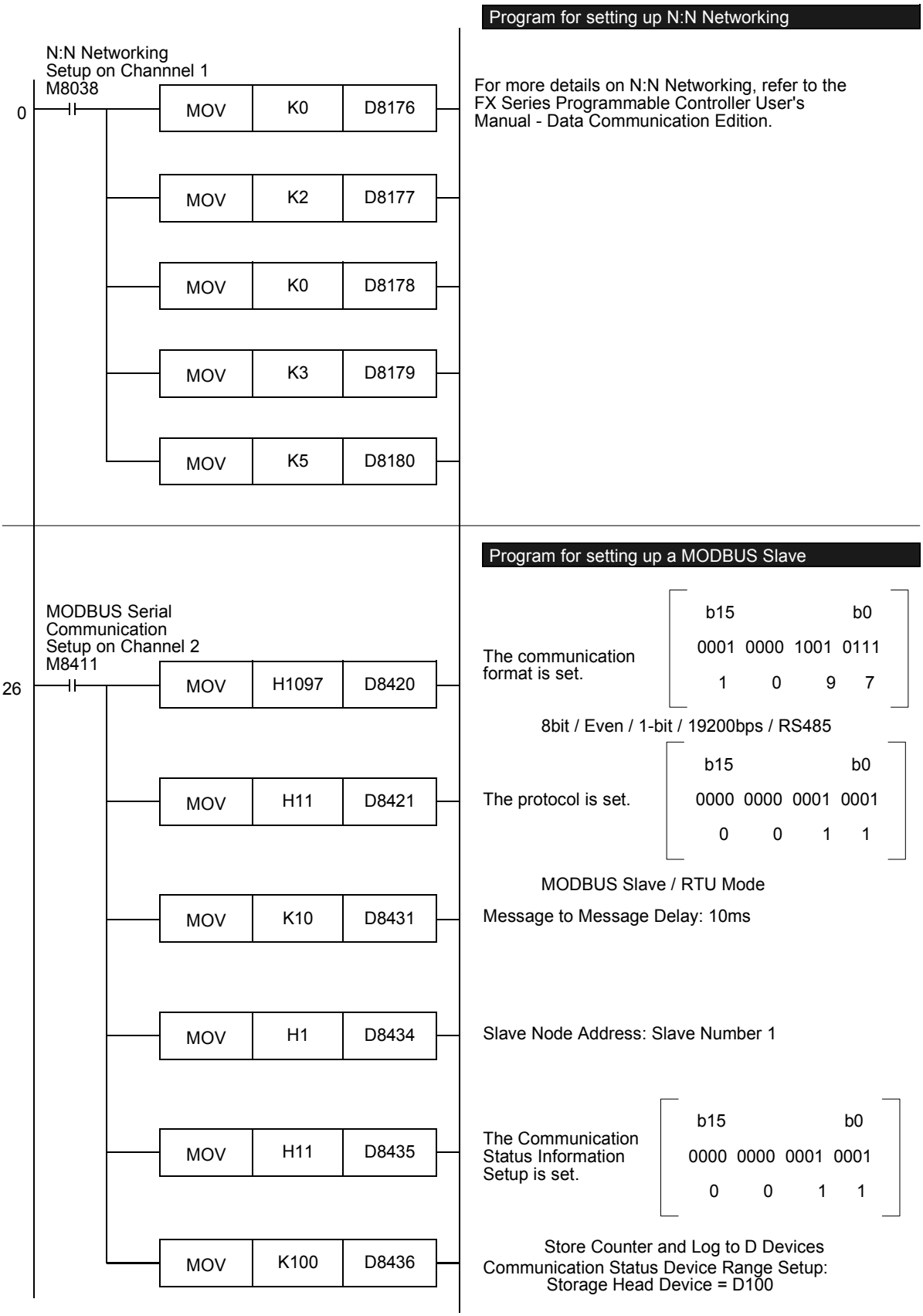
Device	Name	Description
D8400	Communication Format	For Descriptions, refer to Chapter 6.
D8401	Protocol	
D8411	Message to Message Delay	
D8414	Slave Node Address	
D8415	Communication Status Information Setup	
D8416	Communication Status Device Range Setup	

The ladder code for setting the Slave parameters can be expressed as follows:



5.3 Simultaneous N:N Networking and MODBUS Communication

When MODBUS communication and N:N Network are used simultaneously, N:N Networking must be setup first (at program step 0). After which it is possible to program the MODBUS communication setup, as shown below.



5.4 Cautions on Communication Setup

1. Timing of Communication Parameters

Though the MODBUS communication parameters are setup using the PLC program, the parameters become effective only after the PLC power is switched from OFF to ON.

2. N:N Networking and MODBUS Communication

If N:N Networking and MODBUS communication are setup for the same channel, the N:N Networking will operate however the MODBUS settings will be ignored.
If this situation occurs, a "channel double use" error (Error Code #203 in D8402 or D8422) will be displayed for the corresponding channel within the PLC.

→ **For details on MODBUS errors and error devices, refer to Chapter 12.**

3. Using the MODBUS Configuration Request Flag (M8411)

The Auxiliary Relay M8411 is a special relay used for MODBUS Serial Communication setup only. Do not use the PLC ladder program or any other External Devices to set or reset this relay at any time. Do not use any coils or conditionals before the MODBUS Configuration Request Flag or between the MODBUS Configuration Request Flag (M8411) and the MOV commands.
Using other coils or conditionals before or after the MODBUS Configuration Request Flag during Communication Setup will invalidate the MODBUS Communication Parameters and thereby disable MODBUS Serial Communication.

4. Communication Setup Syntax

All MODBUS Communication Parameters must be setup using the MOV command and Constants, meaning a K or H value.
Using non-constant/indirect devices during communication setup will invalidate the MODBUS Communication Parameters and thereby disable MODBUS Serial Communication.

6. Related Devices and Communication Status

In this section the device numbers and functions of the special data registers and special auxiliary relays are described for MODBUS serial communication.

6.1 Special Data Registers

The table shows the Special Data registers used in MODBUS serial communication.

<table>
<tr><th colspan="2">Special Data Register</th><th rowspan="2">Name</th><th rowspan="2">Valid</th><th rowspan="2">Detailed description</th><th rowspan="2">R / W</th></tr>
<tr><th>CH1</th><th>CH2</th></tr>
<tr><td>D8400</td><td>D8420</td><td>Communication Format</td><td>M, S</td><td>This device sets the communication format.

Note: For details on communication format refer to Section 6.2.</td><td>R,W*1</td></tr>
<tr><td>D8401</td><td>D8421</td><td>Protocol</td><td>M, S</td><td>Selection of the channel used, RTU or ASCII mode and Master or Slave identification.
<table>
<tr><th rowspan="2">Bit No.</th><th rowspan="2">Name</th><th colspan="2">Contents</th></tr>
<tr><th>0 (bit = OFF)</th><th>1 (bit = ON)</th></tr>
<tr><td>b0</td><td>Protocol Selection</td><td>Other communication protocol</td><td>MODBUS serial line</td></tr>
<tr><td>b1-3</td><td>Not used</td><td></td><td></td></tr>
<tr><td>b4</td><td>Master/ slave setting</td><td>MODBUS Master</td><td>MODBUS Slave</td></tr>
<tr><td>b5-7</td><td>Not used</td><td></td><td></td></tr>
<tr><td>b8</td><td>RTU/ASCII mode setting</td><td>RTU</td><td>ASCII</td></tr>
<tr><td>b9-15</td><td>Not used</td><td></td><td></td></tr>
</table>
Note: When both flags (b0 of D8401 and b0 of D8421) are turned on, priority will be given to CH1 and CH2 will no longer operate.</td><td>R,W*1</td></tr>
<tr><td>D8402</td><td>D8422</td><td>Communication Error Code</td><td>M, S</td><td>Current error code generated by the MODBUS function.

Special clear conditions:
1) Power on
2) STOP to RUN (master only)</td><td>R,W</td></tr>
<tr><td>D8403</td><td>D8423</td><td>Error Details</td><td>M, S</td><td>Current error details.

Special clear conditions:
1) Power on
2) STOP to RUN (master only)

Note: Refer to the Error table in Chapter 12.</td><td>R,W</td></tr>
<tr><td>D8404</td><td>D8424</td><td>Error step number</td><td>M</td><td>STEP number of the first ADPRW command that caused the original error.

Special clear conditions:
1) Power on
2) STOP to RUN

Note: If the step number is greater than 32767 the value will become a negative number. To see the step numbers above 32767 the user must convert the step number to an unsigned value.</td><td>R,W</td></tr>
</table>

Special Data Register		Name	Valid	Detailed description	R / W
CH1	CH2				
D8405	D8425	Communication Format Display	M, S	This device stores the communication format set in the PLC.	R
D8406	D8426	ASCII Input Delimiter	M, S	Displays the End Of Message character used in ASCII mode. By default this is the LF (0x0A) character. Special clear conditions: 1) Power on Note: In the Master this character can be changed by the user program. For the slave this is a read only device. Refer to the Change Ascii Input Delimiter command in the section 8.2.	R,W
D8407	D8427	Step Number Being Executed	M	Last step number of the MODBUS command that was executed (0 if no command is executed in the program). After the ADPRW command has been executed the last step number will be retained in the device register. Special clear conditions: 1) Power on 2) STOP to RUN Note: If the step number is greater than 32767 the value will become a negative number. To see the step numbers above 32767 the user must convert the step number to an unsigned value.	R
D8408	D8428	Current Retry Value	M	The current value of retries that the master is sending to process the request. Special clear conditions: 1) Power on 2) STOP to RUN 3) Next MODBUS command	R
D8409	D8429	Slave Response Timeout	M	After the master sends a request and no response is received from the slave within the specified time, the master will retry to send the message or terminate the processing of the command with a time out error depending on the setting of the "number of retries" (D8412, D8432). Valid values: 0 to 32767[ms] 0 will default the timeout to 3 seconds Note: This value can also be changed before each command execution.	R,W

Special Data Register		Name	Valid	Detailed description	R / W
CH1	CH2				
D8410	D8430	Turn Around Delay	M	The turnaround delay defines the minimum delay time that the master has to wait after the transmission of a broadcast message before transmitting the next request. This delay allows the slaves to process the broadcast message and prepare the reception of the next request. Valid values: 0 to 32767 [ms] 0 will set the timeout to 400 ms Note 1: This value can also be changed before each command execution. Note 2: If a value less than 3.5 character times (end of message detection time) is selected, the master will wait for at least 3.5 character times. Note 3: The Turn Around delay and Message to Message delay must be set for the slowest slave in the network.	R,W
D8411	D8431	Message to Message delay	M, S	This value defines the minimum waiting time between two messages. This time is used to detect the end of a message. Valid values: 0 to 16382 (ms) 0 will be interpreted as 3.5 character times according to the selected baud rate. If a value less than 3.5 character times is selected, the master will at least wait 3.5 character times. Note 1: This value should be fixed during processing. Note 2: This value can also be changed before each command execution when setting up the network. Note 3: The Turn Around delay and Message to Message delay must be set for the slowest slave in the network.	R,W
D8412	D8432	Number Of Retries	M	In the situation where a slave does not respond within the set time by the Slave Response Timeout the master will try to retransmit the message a set number of retries before it terminates the command processing with a timeout error. Valid values: 0 to 20 [times] If a value of 20 or more is set the number of retries used by the master is set to 20.	R,W*1
D8413	D8433	Not used	-	-	-
D8414	D8434	Slave Node Address	S	Slave node address Valid range: 1 to 247 Note: If during the initialization a value outside the valid range is detected, the configuration is invalid and the slave will not respond to any requests.	R,W*1

<table>
<tr><th colspan="2">Special Data Register</th><th rowspan="2">Name</th><th rowspan="2">Valid</th><th rowspan="2">Detailed description</th><th rowspan="2">R / W</th></tr>
<tr><th>CH1</th><th>CH2</th></tr>
<tr><td>D8415</td><td>D8435</td><td>Communication Status Information Setup</td><td>M,S</td><td>Defines the device range that is used to store the communication state (event & error counter and/or event log).
<table><tr><th rowspan="2">Bit No.</th><th rowspan="2">Name</th><th colspan="2">Contents</th></tr><tr><th>0 (bit = OFF)</th><th>1 (bit = ON)</th></tr><tr><td>b0</td><td>Event and error counter</td><td>Counter values are not stored</td><td>Counter values are stored</td></tr><tr><td>b1-b3</td><td>Not used</td><td></td><td></td></tr><tr><td>b4</td><td>Event log
*Slave Only</td><td>Event log is not stored</td><td>Event log is stored</td></tr><tr><td>b5-7</td><td>Not used</td><td></td><td></td></tr><tr><td>b8</td><td>Communication status storage device type</td><td>D-register</td><td>R-register</td></tr><tr><td>b9-b15</td><td>Not used</td><td></td><td></td></tr></table>
One MODBUS event is one byte so one 16 bit register will hold two events. For further details refer to Subsection 9.5.1.

Note: For Event and Error Counter details refer to Section 6.4.</td><td>R,W*1</td></tr>
<tr><td>D8416</td><td>D8436</td><td>Communication Status Device Range Setup</td><td>M, S</td><td>Defines the PLC head device address of the device block that will store the communication status information.

Counter values occupy 10 devices and the event log requires 33 devices. Therefore if both are displayed a total of 43 devices are required.

According to these rules, the maximum valid range will be:

For D:
Counter only: 0-7990 (i.e. D8415 / D8435 = 01H)
Log only: 0-7967 (i.e. D8415 / D8435 = 010H)
Log and counter: 0-7957 (i.e. D8415 / D8435 = 011H)

For R:
Counter only: 0-32758 (i.e. D8415 / D8435 = 0101H)
Log only: 0-32735 (i.e. D8415 / D8435 = 0110H)
Log and counter: 0-32725 (i.e. D8415 / D8435=0111H)

Note: If the above mentioned rule is violated neither counter nor event log is stored and an error is generated.</td><td>R,W*1</td></tr>
<tr><td>D8417</td><td>D8437</td><td>Not used</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>D8063</td><td>D8438</td><td>Communication Error Code</td><td>M, S</td><td>In the event of a communication error this register holds the error code corresponding to the error occurring during the MODBUS communication.

Special clear conditions:
1) Power on

Note: In the event of a Ch1 error MODBUS communication error '6321' will be stored in D8063. In the event of a Ch2 MODBUS communication error '3821' will be stored in D8438.</td><td>R,W*1</td></tr>
</table>

Special Data Register		Name	Valid	Detailed description	R / W
CH1	CH2				
D8419	D8439	Communication Mode	M, S	Displays the protocol that the serial port is currently using: 0: Programming Port Protocol 1: Programming Port Modem Mode 2: Computer Link 3: N:N Network 4: RS Command 5: RS2 Command 6: Parallel Link 7: Inverter Communication Command 9: MODBUS Serial	R
D8470 D8471		MODBUS Device Mapping 1	S	If a MOV H**** D8470 is triggered by LD M8411 during the MODBUS initialization, the slave device mapping can be changed by the user. Note: For details refer to Section 9.4.	R,W*1
D8472 D8473		MODBUS Device Mapping 2	S	If a MOV H**** D8472 is triggered by LD M8411 during the MODBUS initialization, the slave device mapping can be changed by the user. Note: For details refer to Section 9.4.	R,W*1
D8474 D8475		MODBUS Device Mapping 3	S	If a MOV H**** D8474 is triggered by LD M8411 during the MODBUS initialization, the slave device mapping can be changed by the user. Note: For details refer to Section 9.4.	R,W*1
D8476 D8477		MODBUS Device Mapping 4	S	If a MOV H**** D8476 is triggered by LD M8411 during the MODBUS initialization, the slave device mapping can be changed by the user. Note: For details refer to Section 9.4.	R,W*1
D8478 D8479		MODBUS Device Mapping 5	S	If a MOV H**** D8478 is triggered by LD M8411 during the MODBUS initialization, the slave device mapping can be changed by the user. Note: For details refer to Section 9.4.	R,W*1
D8480 D8481		MODBUS Device Mapping 6	S	If a MOV H**** D8480 is triggered by LD M8411 during the MODBUS initialization, the slave device mapping can be changed by the user. Note: For details refer to Section 9.4.	R,W*1
D8482 D8483		MODBUS Device Mapping 7	S	If a MOV H**** D8482 is triggered by LD M8411 during the MODBUS initialization, the slave device mapping can be changed by the user. Note: For details refer to Section 9.4.	R,W*1
D8484 D8485		MODBUS Device Mapping 8	S	If a MOV H**** D8484 is triggered by LD M8411 during the MODBUS initialization, the slave device mapping can be changed by the user. Note: For details refer to Section 9.4.	R,W*1

M: Master S: Slave R: Read W: Write

*1. Values must be written to these Device registers using the MODBUS configuration block using M8411. For details refer to Section 9.4.

6.2 Communication setting for MODBUS

The following devices are used in the communication setting.
When using the communication port (Ch1), set D8400.
When using the communication port (Ch2), set D8420.

1) D8400 and D8420 (communication format)
By setting values to D8400 or D8420, the data length, parity, baud rate, etc. can be set.
The table below shows the contents of D8400 and D8420.

Bit No.	Name	Contents	
		0 (bit = OFF)	1 (bit = ON)
b0	Data length	7-bit	8-bit
b1 b2	Parity	Parity b2, b1 (0, 0): Not provided (0, 1): Odd (1, 1): Even	
b3	Stop bit	1-bit	2-bit
b4 b5 b6 b7	Baud rate (bps)	b7, b6, b5, b4 (0, 0, 1, 1): 300 (0, 1, 0, 0): 600 (0, 1, 0, 1): 1200 (0, 1, 1, 0): 2400	b7, b6, b5, b4 (0, 1, 1, 1): 4800 (1, 0, 0, 0): 9600 (1, 0, 0, 1): 19200
b8 -11	Reserved		
b12	H/W type	RS232C	RS485
b13-15	Reserved		

6.3 Special Auxiliary Relays

The table shows the Special Auxiliary Relays used for MODBUS serial communication.

Special Device		Name	Valid	Detailed Description	R / W
CH1	CH2				
M8411		MODBUS Configuration Request Flag	M, S	LD M8411 can be used to trigger a set of subsequent MOV commands that initialize the MODBUS function. Note: For details refer to Section 5.2.	R, W
M8029		Command Execution Complete	M	This bit is turned on if the processing of a MODBUS command is completed. Special clear conditions: 1) Power on 2) STOP to RUN 3) If another command using M8029 is triggered (including another MODBUS command) Note: In the event of an error there will be one or more of the error flags set.	R
M8401	M8421	MODBUS Request in Process	M	If the MODBUS stack is processing a command no further commands can be triggered until the current request is completed and the Command Execution Complete Flag is on. Special clear conditions: 1) Power on 2) STOP to RUN	R
M8402	M8422	MODBUS Communication Error	M, S	Set during the processing of the current MODBUS command error. Special clear conditions: 1) Power on 2) STOP to RUN 3) If another MODBUS command is triggered	R
M8403 M8063	M8423 M8438	MODBUS Communication Error (latched)	M, S	Set once a MODBUS command error has been processed. Special clear conditions: 1) Power on 2) STOP to RUN	R
M8404	M8424	Listen Only Mode	S	0: Normal processing 1: Listen only mode Special clear conditions: 1) Power on 2) Reception of a restart command from the master Note: Received messages are evaluated, but no action takes place and no response is sent. The only exception is the "reset communication option" command (diagnosis 0x08 sub command 0x01). If this command is received the slave recovers from listen only mode without sending a response, all subsequent commands to this slave will be answered with a response as usual.	R

Special Device		Name	Valid	Detailed Description	R / W
CH1	CH2				
M8408	M8428	Retry	M	Set while the master sends retries when the slave fails to respond in time. Special clear conditions: 1) Power on 2) STOP to RUN 3) If another MODBUS command is triggered As long as the slave responds on one of the retries the error flag will not be set.	R
M8409	M8429	Timeout	S	Set if a response timeout occurs. Special clear conditions: 1) Power on 2) STOP to RUN 3) If another MODBUS command is triggered Note: If the number of retries is > 0, the error flag is not set until the selected number of retries failed by a timeout (or another failure).	R

M: Master S: Slave R: Read W: Write

6.4 Communication status

The event and error counters will occupy ten devices beginning from the device defined by (D8415 / D8435) and (D8416/D8436). PLC destination devices are latched, therefore the Master's devices will be cleared at power ON and when the PLC is switched from STOP to RUN.

For Slave devices, the event and error counters will be cleared when: the communication is reset, a counter reset command is received, at power ON and when the PLC is switched from STOP to RUN.

The following table shows the communication status of the event and error counters and the communication event log data for the following scenario:

D8415 = 11H - i.e. store event counter and event log into D devices
D8416 = 100 - i.e. head device is set to D100

Device	Description	Valid	Details	R / W
Head Device (D100)*1	Bus Message Counter	M, S	Number of messages that a remote node has detected on the bus. Note: Messages with false CRC/LRC are not taken into account.	R
Head Device + 1 (D101)*1	Bus Communication Error Counter	M, S	This counter is incremented if one of the following errors occur: • CRC/LRC mismatch • Bit-level error (overrun, parity error) • Received telegram length is <= 3 characters (RTU) or <= 8 characters (ASCII)	R
Head Device + 2 (D102)*1	Exception Error Counter	M, S	Master: Number of received exception error responses. Slave: Number of exception conditions detected by the remote node including exceptions caused by broadcast messages (In this case no exception response is sent).	R
Head Device + 3 (D103)*1	Slave Message Counter	S	Number of messages addressed to the slave (including broadcast).	R
Head Device + 4 (D104)*1	Slave No Response Counter	S	Number of received messages for which the slave did not return a response (Number of received broadcasted messages).	R
Head Device + 5 (D105)*1	Slave NAK Counter	S	Number of times the slave responds with a NAK exception (This is always 0 when using FX3U.).	R
Head Device + 6 (D106)*1	Slave Busy Counter	S	Number of times the slave respond with a busy exception (This is always 0 when using FX3U.).	R
Head Device + 7 (D107)*1	Character Overrun Counter	M, S	Master: Number of times the master detected a character overrun condition. Slave: Number of times the slave detected a character overrun condition.	R
Head Device + 8 (D108)*1	Event counter	S	This counter is incremented for each successful message completion.It is not increased in the following cases: • Exception responses • Poll commands • Fetch event counter commands	R
Head Device + 9 (D109)	Not used	-	-	-
Head Device + 10 (D110)*2	Event Log Length	S	Number of events stored in the event log. Note: For details refer to Subsection 9.5.1.	R
Head Device + 11 to 42 (D111-D142)*2	Event Log	S	Up to 64 events (Each D register = 2 events) Note: For details refer to Subsection 9.5.1.	R

M: Master S: Slave R: Read W: Write

*1. Event and Error counters

*2. Communication event log

7. MODBUS Standard Commands

The following chapter explains in detail MODBUS Communication. For standard use of the FX3U MODBUS Serial function please refer to Chapter 8 (Master Specification) or Chapter 9 (Slave Specification).

7.1 MODBUS Standard Commands Support List

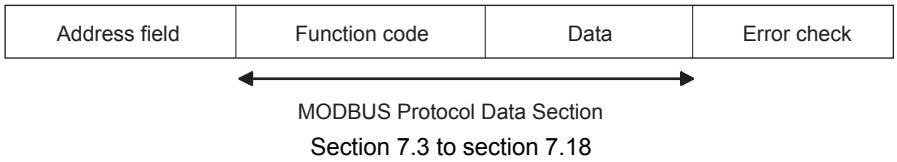
The following table indicates a list of the MODBUS standard functions supported by the FX-Series MODBUS Communication ADP.

Command Code	Subcommand Code	Command Name	Details	Accessible Devices per Message	Broadcast	Reference
0x01		Read Coils	Read binary (R/W) devices	1 to 2000 points	✘	Section 7.4
0x02		Read Discrete Inputs	Read binary (RO) devices	1 to 2000 points	✘	Section 7.5
0x03		Read Holding Registers	Read 16 bit (R/W) register	1 to 125 points	✘	Section 7.6
0x04		Read Input Registers	Read 16 bit (RO) register	1 to 125 points	✘	Section 7.7
0x05		Write Single Coil	Write single binary device	1 point	✔	Section 7.8
0x06		Write Single Register	Write single 16 bit register device	1 point	✔	Section 7.9
0x07		Read Exception Status	Read 1 byte of vendor specified data	-	✘	Section 7.10
0x08 Diagnosis	0x00	Return Query Data	Loop back function	-	✘	Subsection 7.11.1
	0x01	Restart Communication Option	Restart communication/ Remote Communication Reset	-	✔	Subsection 7.11.2
	0x02	Return Diagnostic Register	Read 16 bit register of vendor specified data	-	✘	Subsection 7.11.3
	0x03	Change ASCII Input Delimiter	Change ASCII mode End of Message character	-	✔	Subsection 7.11.4
	0x04	Force Listen Only Mode	Switch slave to Listen Only Mode	-	✔	Subsection 7.11.5
	0x0A	Clear Counters and Diagnostic Register	Clear all counters and the diagnostic registers	-	✔	Subsection 7.11.6
	0x0B	Return Bus Message Count	Read number of detected messages	-	✘	Subsection 7.11.7

Command Code	Subcommand Code	Command Name	Details	Accessible Devices per Message	Broadcast	Reference
0x08 Diagnosis	0x0C	Return Bus Communication Error Count	Read number of detected communication errors	-	✘	Subsection 7.11.8
	0x0D	Return Bus Exception Error Count	Read number of detected exception conditions	-	✘	Subsection 7.11.9
	0x0E	Return Slave Message Count	Read number of received requests	-	✘	Subsection 7.11.10
	0x0F	Return Slave No Response Count	Read "No Response" counter of the slave	-	✘	Subsection 7.11.11
	0x10	Return Slave NAK Count	Read NAK counter of the slave	-	✘	Subsection 7.11.12
	0x11	Return Slave Busy Count	Read "Busy" counter of the slave	-	✘	Subsection 7.11.13
	0x12	Return Bus Character Overrun Count	Read "Bus Character Overrun" counter of the slave	-	✘	Subsection 7.11.14
0x0B		Get Communication Event Counter	Read communication event counter	-	✘	Section 7.12
0x0C		Get Communication Event Log	Read communication event log	-	✘	Section 7.13
0x0F		Write Multiple Coils	Write multiple binary (R/W) devices	1 to 1968 points	✔	Section 7.14
0x10		Write Multiple Registers	Write multiple 16 bit (R/W) registers	1 to 123 points	✔	Section 7.15
0x11		Report Slave ID	Read Slave ID code data	-	✘	Section 7.16
0x16		Mask Write Register	Manipulate slave register with AND Mask / OR Mask	1 point	✔	Section 7.17
0x17		Read/Write Multiple Registers	Read/Write multiple 16 bit (R/W) registers	Read:1 to 125 pointsWrite: 1 to 121 points	✘	Section 7.18

7.2 Frame Specifications

The following shows the frame specifications for the MODBUS protocol.



Section 7.3 to section 7.18

The following table details the frame specification for the MODBUS Protocol.

Area name	Description
Address field	[When the master sends a request message to a slave] 0: Sends a request message to all the slaves. (Broadcast) 1 to 247: Sends a request to a specific Slave number. [When the slave sends a response message to the master] The host station number is stored when sending a response message. Note: 247 is the MODBUS maximum address number. The FX3U MODBUS Master can address from 1 to16 stations within this range.
Function code	[When the master sends a request message to a slave] The master specifies the number of the action to be taken by the slave. [When the slave sends a response message to the master] A requested function code is stored in the case of normal completion. The most significant bit turns ON in the case of error completion.
Data	[When the master sends a request message to a slave] The information needed to execute the action specified by a function code is stored. [When the slave sends a response message to the master] The execution result of the action specified by a function code is stored. An exception code is stored when failed.
Error check *1	The node adds the check code automatically to all transmitted messages and recalculates the check code for any received message. The received message is discarded if it has an error.

*1. The error check method differs depending on the frame mode. See Subsection 7.2.1.

Note

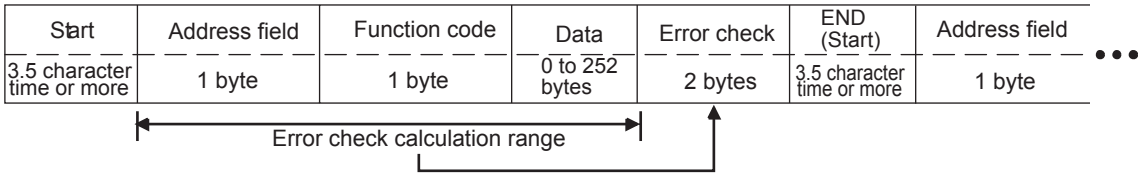
Refer to the Subsection 7.2.1 for the data size of each area.

7.2.1 Frame mode

For the FX-Series MODBUS Communication ADP, the following frame modes are available.

The frame mode of the FX-Series MODBUS Communication ADP must be consistent with that of the target device.

1) Available frame modes
   a) RTU mode
      In this mode, frames are received or sent in binary codes.
      The frame specifications are compliant with the MODBUS protocol specifications.



Note

The Cyclical Redundancy Checking (CRC) field is two bytes, containing a 16-bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The device that receives recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results.

A procedure for generating a CRC is:

1) Load a 16-bit register with FFFF hex (all 1's). Call this the CRC register.
2) Exclusive OR the first 8-bit byte of the message with the low-order byte of the 16-bit CRC register, putting the result in the CRC register.
3) Shift the CRC register one bit to the right (toward the LSB), zero-filling the MSB (Most Significant bit). Extract and examine the LSB (Least Significant bit).
4) (If the LSB was 0): Repeat Step 3 (another shift).
   (If the LSB was 1): Exclusive OR the CRC register with the polynomial value 0xA001 (1010 0000 0000 0001).
5) Repeat Steps 3 and 4 until 8 shifts have been performed. When this is done, a complete 8-bit byte will have been processed.
6) Repeat Steps 2 through 5 for the next 8-bit byte of the message. Continue doing this until all bytes have been processed.
7) The final content of the CRC register is the CRC value.
8) When the CRC is placed into the message, its upper and lower bytes must be switched as described below.

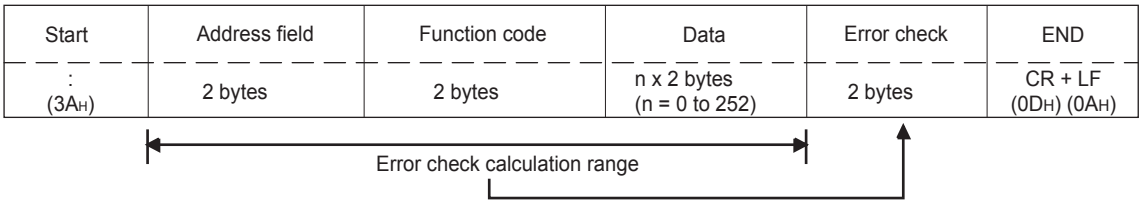
The following is a calculation example in the case where function code 07H is sent to station No. 2.

CRC error check procedure	16-bit register (MSB)				Carry Flag
(Load the register whose 16 bits are all "1")	1111	1111	1111	1111	
02H(Station No.)			0000	0010	
Exclusive OR (XOR)	1111	1111	1111	1101	
Shift 1	0111	1111	1111	1110	1
Generator polynomial	1010	0000	0000	0001	
Exclusive OR (XOR)	1101	1111	1111	1111	
Shift2	0110	1111	1111	1111	1
Generator polynomial	1010	0000	0000	0001	
Exclusive OR (XOR)	1100	1111	1111	1110	
Shift3	0110	0111	1111	1111	0
Shift4	0011	0011	1111	1111	1
Generator polynomial	1010	0000	0000	0001	
Exclusive OR (XOR)	1001	0011	1111	1110	
Shift5	0100	1001	1111	1111	0
Shift6	0010	0100	1111	1111	1
Generator polynomial	1010	0000	0000	0001	
Exclusive OR (XOR)	1000	0100	1111	1110	
Shift7	0100	0010	0111	1111	0
Shift8	0010	0001	0011	1111	1
Generator polynomial	1010	0000	0000	0001	
Exclusive OR (XOR)	1000	0001	0011	1110	
07H(Function)			0000	0111	
Exclusive OR (XOR)	1000	0001	0011	1001	
Shift 1	0100	0000	1001	1100	1
Generator polynomial	1010	0000	0000	0001	
Exclusive OR (XOR)	1110	0000	1001	1101	
Shift2	0111	0000	0100	1110	1
Generator polynomial	1010	0000	0000	0001	
Exclusive OR (XOR)	1101	0000	0100	1111	
Shift3	0110	1000	0010	0111	1
Generator polynomial	1010	0000	0000	0001	
Exclusive OR (XOR)	1100	1000	0010	0110	
Shift4	0110	0100	0001	0011	0
Shift5	0011	0010	0000	1001	1
Generator polynomial	1010	0000	0000	0001	
Exclusive OR (XOR)	1001	0010	0000	1000	
Shift6	0100	1001	0000	0100	0
Shift7	0010	0100	1000	0010	0
Shift8	0001	0010	0100	0001	0
CRC value	12H		41H		

Address field	Function code	CRC (Error check)	
(02H)	(07H)	(41H)	(12H)

b) ASCII mode
In this mode, frames are received or sent in units of 2 characters 2 bytes of ASCII codes.

The frame specifications are compliant with the MODBUS protocol specifications.



Note

The Longitudinal Redundancy Checking (LRC) field is one byte, containing an 8-bit binary value. The LRC value is calculated by the transmitting device, which appends the LRC to the message. The device that receives recalculates an LRC during receipt of the message, and compares the calculated value to the actual value it received in the LRC field. If the two values are not equal, an error results.

A procedure for generating an LRC is:

1) Add all bytes in the message, excluding the starting 'colon' and ending CRLF. Add them into an 8-bit field, so that carries will be discarded.
2) Subtract the final field value from FF hex (all 1's), to produce the ones-complement.
3) Add 1 to produce the twos-complement.
4) For a transmit frame the LRC is calculated before converting to ASCII.

The following are calculation examples in the case where function code 01H is sent to station No. 2.

The following table illustrates the LRC calculation procedure (when sending a request message):

LRC in request message transmission			
Station No. (address field)	02	0000	0010
Function code	01	0000	0001
Head coil number (H)	00	0000	0000
Head coil number(L)	00	0000	0000
Read points (H)	00	0000	0000
Read points (L)	08	+0000	1000
Addition result	0B	0000	1011
Bit reversal 1	F4	1111	0100
+1			1
2's complement	F5	1111	0101
LRC (Error check)	F5	F	5

Start :	Address field (02H)		Function code (01H)		Head input number				Read points				LRC (Error check) (F5H)		"CR"	"LF"
					(00H)		(00H)		(00H)		(00H)					
3AH	30H	32H	30H	31H	30H	30H	30H	30H	30H	30H	30H	38H	46H	35H	0DH	0AH

7.3 Protocol Data Unit Formats by Commands

This section describes MODBUS protocol data unit formats used by the FX-Series MODBUS Communication ADP.

1) Precautions:

   a) When the FX-Series MODBUS Communication ADP receives a broadcast request message:
   Although the processing requested by the request message is performed etc., no response message is sent to the master.

   b) When the FX-Series MODBUS Communication ADP receives a request message in the listen only mode:
   Received messages are evaluated, but no action takes place and no response is sent. The only exception is the "reset communication option" command (diagnosis 0x08 sub command 0x01). If this command is received the slave recovers from listen only mode without sending a response, all subsequent commands to this slave will be answered with a response as usual. For further information refer to Chapter 6.

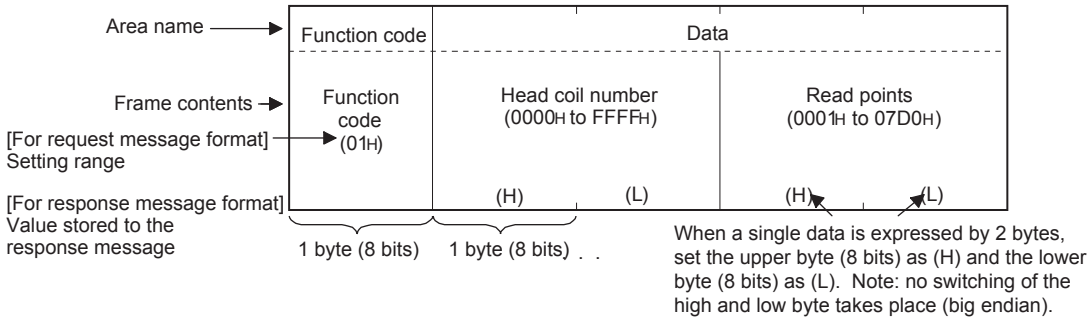
2) When the processing is completed in error at the slave (FX-Series MODBUS Communication ADP):

   When the processing (read/write, diagnostics, etc.) requested by the request message is completed in error, an exception code is sent to the master.
   "Response message formats (when completed with an error)" in Section 7.4 to 7.18.

   a) Storage location of exception code and error code. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

3) How to see the request/response message formats provided in Section 7.4 to 7.18:

a) Request/Response message format diagram
The following shows how to see the request/response message format diagrams provided in Section 7.4 to 7.18.

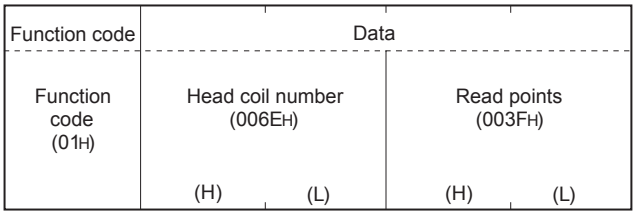


b) Frame mode of the message format
The messages shown in Section 7.4 to 7.18 are displayed in RTU format.

For use in ASCII mode, convert the values into ASCII codes.

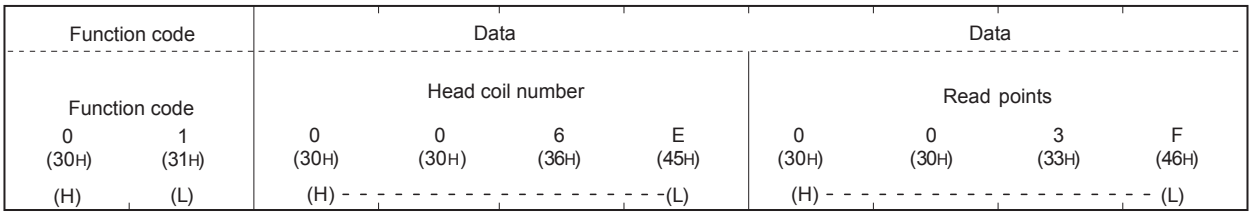
(Conversion example)

(RTU mode)



Convert RTU mode to ASCII mode

(ASCII mode)

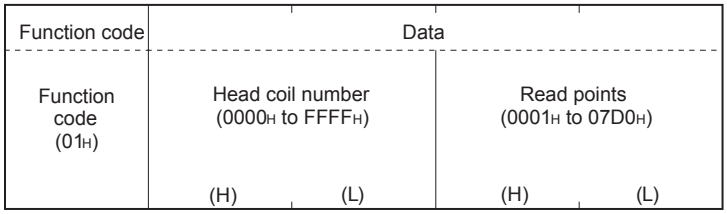


c) Response message format
The response message formats issued from the slave to the master differs depending on whether the slave has normally completed or failed to handle the requested processing (read/write, diagnostics, etc.)
The formats for normal and error completions are shown in Section 7.4 to 7.18.

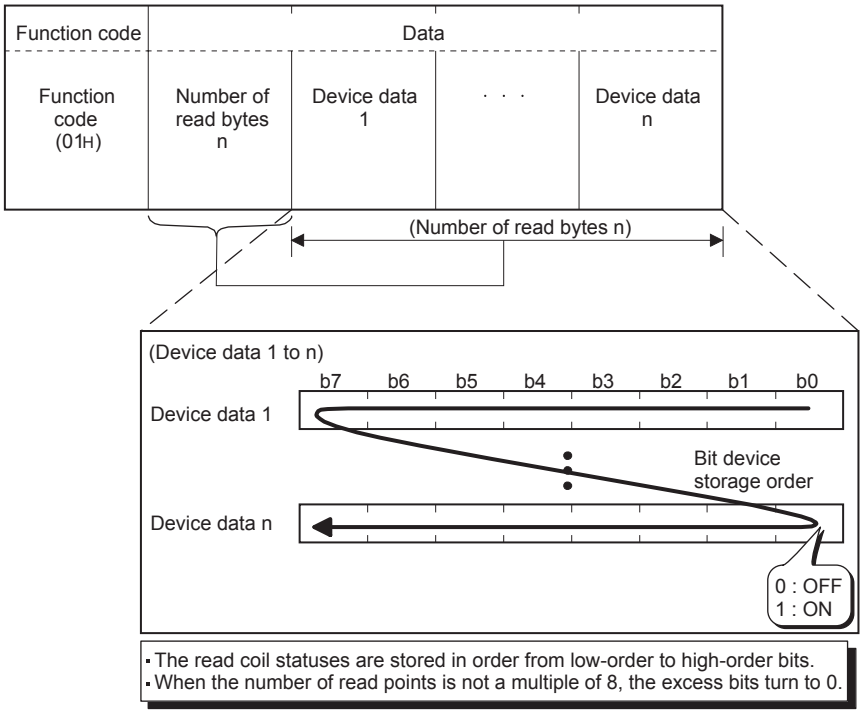
7.4 Read Coils (Command Code: 0x01)

Reads the status (ON/OFF) of one or more coils.

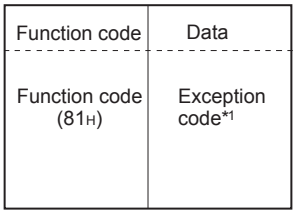
1) Request message format (Master → Slave)



2) Response message format (Slave → Master)
(When completed normally)



(When completed with an error)



*1. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

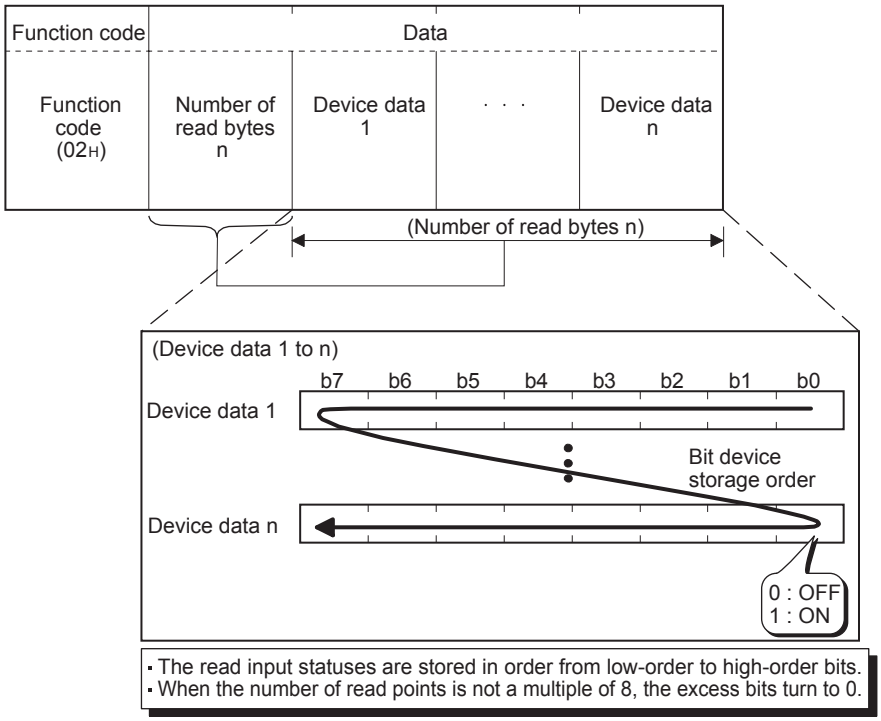
7.5 Read Discrete Inputs (Command Code: 0x02)

Reads the status (ON/OFF) of one or more inputs.

1) Request message format (Master → Slave)

Function code	Data			
Function code (02H)	Head input number (0000H to FFFFH)		Read points (0001H to 07D0H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)



(When completed with an error)

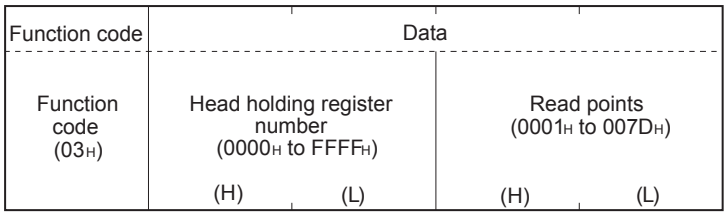
Function code	Data
Function code (82H)	Exception code*1

*1. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

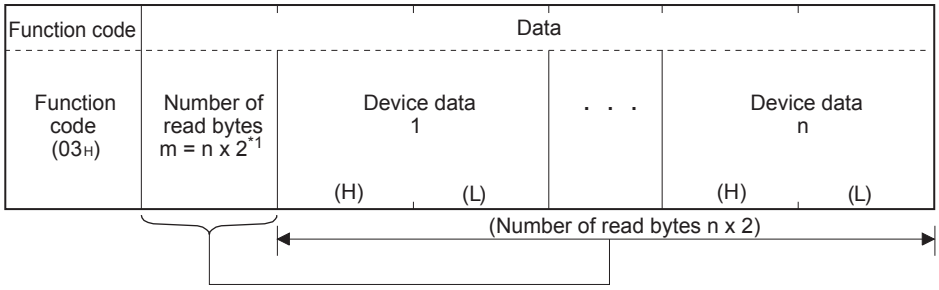
7.6 Read Holding Registers (Command Code: 0x03)

Reads the values of one or more holding registers.

1) Request message format (Master → Slave)

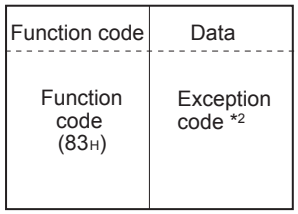


2) Response message format (Slave → Master)
(When completed normally)



*1 For example, if n = 4, the number of bytes is calculated as 4 x 2 = 8 bytes

(When completed with an error)



*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

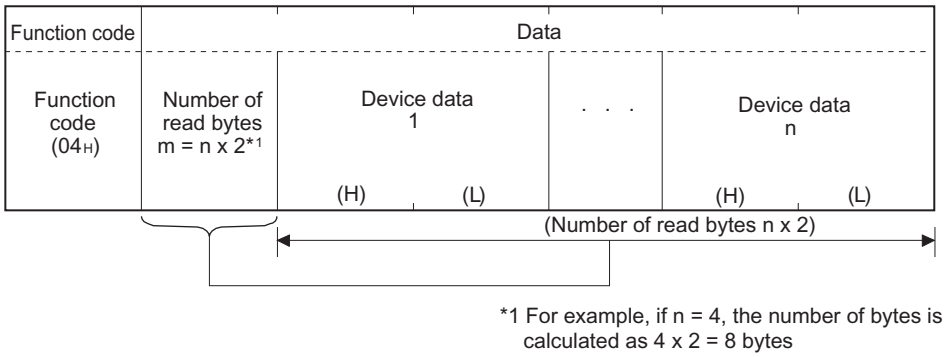
7.7 Read Input Registers (Command Code: 0x04)

Reads the values of one or more input registers.

1) Request message format (Master → Slave)

Function code	Data			
Function code (04H)	Head input register number (0000H to FFFFH)		Read points (0001H to 007DH)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)



*1 For example, if n = 4, the number of bytes is calculated as 4 x 2 = 8 bytes

(When completed with an error)

Function code	Data
Function code (84H)	Exception code *2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.8 Write Single Coil (Command Code: 0x05)

Writes a value (ON/OFF) to one coil.

1) Request message format (Master → Slave)

Function code	Data			
Function code (05_H)	Coil number (0000_H to $FFFF_H$)		ON/OFF specification (0000_H: OFF, $FF00_H$: ON)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)
The slave returns the request message received from the master without change.

(When completed with an error)

Function code	Data
Function code (85_H)	Exception code*1

*1. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.9 Write Single Register (Command Code: 0x06)

Writes a value to one holding register.

1) Request message format (Master → Slave)

Function code	Data			
Function code (06H)	Holding register number (0000H to FFFFH)		Write data (0000H to FFFFH)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)
The slave returns the request message received from the master without change.

(When completed with an error)

Function code	Data
Function code (86H)	Exception code*1

*1. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.10 Read Exception Status (Command Code: 0x07)

Reads error status.

1) Request message format (Master → Slave)

Function code
Function code (07_H)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Data
Function code (07_H)	Error information*1

*1. Vendor specific data.

(When completed with an error)

Function code	Data
Function code (87_H)	Exception code *2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11 Diagnostics (Command Code: 0x08)

Executes the various diagnostics and checks the FX-Series MODBUS Communication ADP.

7.11.1 Return query data (sub-comand code: 0x00)

Returns the contents of the request message without change.
Used to check if the network or the target device is operating normally. (Loopback test)

1) Request message format (Master → Slave)

Function code	Sub-function code	Data
Function code (08_H)	Sub-function code (0000_H) (H) (L)	Arbitrary data

2) Response message format (Slave → Master)
(When completed normally)
The slave returns the request message received from the master without change.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code*1

*1. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.2 Restart communications option (sub-command code: 0x01)

Initializes the communication port of the receiving channel side and restarts the slave function.

Restart is performed after returning the response message corresponding to a request message.

The operation status returns to online mode when it was in the listen only mode.

The following data are cleared when executing the restart communications option.

- Event and Error counter (Refer to Chapter 6)
- Communications event log (Refer to Section 9.5)*1

*1. Clears the data when the communications event log clear is specified in the request message.

1) Request message format (Master → Slave)

Function code	Sub-function code	Data
Function code (08_H)	Sub-function code (0001_H) (H) (L)	Clear setting of Communications event log (0000_H: Do not clear, $FF00_H$: Clear) (H) (L)

2) Response message format (Slave → Master)
(When completed normally)
The slave returns the request message received from the master without change.
However, if a request message is received during listen only mode, the status will only return to online mode and no response message will be returned.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code*2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.3 Return diagnostic register (sub-command code: 0x02)

Reads out the diagnostic register of the remote node to the master.

1) Request message format (Master → Slave)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code (0002_H)		(0000_H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code (0002_H)		Diagnostic register value	
	(H)	(L)	(H)	(L)

(H): 00_H (FX_{3U})*1

(L): M8060-M8067 (When using CH1 for MODBUS communication)
M8060-M8062,M8438, M8064-M8067 (When using CH2 for MODBUS communication)

*1. When using 3rd party products. The high byte (H) does not equal 00H. For further information on the specifications for the slave refer to Chapter 9.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code*2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.4 Change ASCII input delimiter (sub-command code: 0x03)

Changes the 2nd byte (LF(0AH)) of the end code in the ASCII mode to a specified data.
The specified data is stored in D8406/D8426.

Start	Address field	Function code	Data	Error check	END	
: (3AH)	2 characters	2 characters	n x 2 characters (n = 0 to 252)	2 characters	CR + (0DH)	LF (0AH)

Change this into a specified data.

1) Request message format (Master → Slave)

Function code	Sub-function code	Data	
Function code (08H)	Sub-function code (0003H) (H) (L)	Input delimiter setting (00H to FFH)	(00H)

Note

(0x3A), "0"-"9" (0x30-0x39), "A"-"F" (0x41-0x46) and "a"-"f" (0x61-0x66) should not be used as they might occur within the message and cause false end of frame detection.

2) Response message format (Slave → Master)
(When completed normally)
The slave returns the request message received from the master without change.

(When completed with an error)

Function code	Data
Function code (88H)	Exception code*1

*1. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.5 Force listen only mode (sub-command code: 0x04)

Places a slave into the offline mode.
Used when disconnecting a slave from the network.

When FX-Series MODBUS Communication ADP is set in the listen only mode, the status is as follows:

- Ignores all request messages except for those of restart communications option (Refer to Subsection 7.11.2).
- Stops counting of the diagnostic counter (Refer to Chapter 6).
- Continues recording with the communications event log (Refer to Section 9.5).

1) Request message format (Master → Slave)

Function code	Sub-function code	Data
Function code (08_H)	Sub-function code (0004_H) (H) (L)	(0000_H) (H) (L)

2) Response message format (Slave → Master)
(When completed normally)
No response message is returned because the listen only mode (offline status) is active.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code*1

*1. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

Note

1) Whether the FX-Series MODBUS Communication ADP has been switched to listen only mode or not is indicated by M8404/M8424.
2) The listen only mode can be changed to online mode by either of the following:
   - Restart communications option (Refer to Subsection 7.11.2)
   - Power OFF ON

7.11.6 Clear counters and diagnostic register (sub-command code: 0x0A)

Clears counters (e.g. message count).

The following counters will be cleared. (Refer to Chapter 6)

- Bus message count
- Bus communication error count
- Exception error count
- Slave message count
- Slave no-response count
- Slave NAK count
- Slave busy count
- Character overrun error count
- Communications event count (Refer to Section 7.12)

The FX3U bit devices displayed in the diagnostic register are not reset, so the diagnostic register will be overwritten in the next scan by the actual state of the error flags. The error flags can be reset by PLC program or monitoring device.

1) Request message format (Master → Slave)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000A_H$)		(0000_H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave→ Master)
(When completed normally)
The slave returns the request message received from the master without change.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code*1

*1. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.7 Return bus message count (sub-command code: 0x0B)

Reads out the number of messages detected on the line to the master.

1) Request message format (Master → Slave)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000B_H$)		(0000_H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000B_H$)		Bus message count value (0000_H to $FFFF_H$) *1	
	(H)	(L)	(H)	(L)

*1. Refer to Chapter 6 for the relevant counts, count clear methods and precautions.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code *2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.8 Return bus communication error count (sub-command code: 0x0C)

Reads out the number of error messages detected on the line to the master.

1) Request message format (Master → Slave)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000C_H$)		(0000_H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000C_H$)		Bus communication error count value (0000_H to $FFFF_H$) *1	
	(H)	(L)	(H)	(L)

*1. Refer to Chapter 6 for the relevant counts, count clear methods and precautions.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code *2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.9 Return bus exception error count (sub-command code: 0x0D)

Reads out the number of exception errors to the master.

1) Request message format (Master → Slave)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000D_H$)		(0000_H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000D_H$)		Exception error count value (0000_H to $FFFF_H$) *1	
	(H)	(L)	(H)	(L)

*1. Refer to Chapter 6 for the relevant counts, count clear methods and precautions.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code *2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.10 Return slave message count (sub-command code: 0x0E)

Reads out the number of messages processed by the slave to the master. (Including broadcast messages)

1) Request message format (Master → Slave)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000E_H$)		(0000_H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000E_H$)		Slave message count value (0000_H to $FFFF_H$) *1	
	(H)	(L)	(H)	(L)

*1. Refer to Chapter 6 for the relevant counts, count clear methods and precautions.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code *2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.11 Return slave no response count (sub-command code: 0x0F)

Reads to out the number of broadcast request messages received to the master.

1) Request message format (Master → Slave)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000F_H$)		(0000_H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code ($000F_H$)		Slave no-response count value (0000_H to $FFFF_H$) *1	
	(H)	(L)	(H)	(L)

*1. Refer to Chapter 6 for the relevant counts, count clear methods and precautions.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code *2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.12 Return slave NAK count (sub-command code: 0x10)

Reads out the number of NAK responses to the master.

The FX-Series MODBUS Communication ADP always returns "0".

1) Request message format (Master → Slave)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code (0010_H)		(0000_H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code (0010_H)		Slave NAK count value (0000_H) * 1	
	(H)	(L)	(H)	(L)

*1. Refer to Chapter 6 for the relevant counts, count clear methods and precautions.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code *2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.13 Return slave busy count (sub-command code: 0x11)

Reads out the number of busy responses to the master.
The FX-Series MODBUS Communication ADP always returns "0".

1) Request message format (Master → Slave)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code (0011_H)		(0000_H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code (0011_H)		Slave busy count value (0000_H) *1	
	(H)	(L)	(H)	(L)

*1. Refer to Chapter 6 for the relevant counts, count clear methods and precautions.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code *2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.11.14 Return bus character overrun count (sub-command code: 0x12)

Reads out the number of times the request message size exceeds the upper limit to the master.

1) Request message format (Master → Slave)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code (0012_H)		(0000_H)	
	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Sub-function code		Data	
Function code (08_H)	Sub-function code (0012_H)		Bus character overrun count value (0000_H to $FFFF_H$) *1	
	(H)	(L)	(H)	(L)

*1. Refer to Chapter 6 for the relevant counts, count clear methods and precautions.

(When completed with an error)

Function code	Data
Function code (88_H)	Exception code *2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

Note

Refer to Subsection 7.2.1 for details on the size of request messages.

7.12 Get Communications Event Counter (Command Code: 0x0B)

Acquires the number of messages whose requested actions (read/write, diagnostics, etc.) have been normally completed.

Note

Only normally completed messages are counted.

1) Request message format (Master → Slave)

Function code
Function code ($0B_H$)

2) Response message format (Slave → Master)
(When completed normally)

Function code	Data			
Function code ($0B_H$)	Program command status (0000_H) *1		Communications event count value (0000_H to $FFFF_H$) *2	
	(H)	(L)	(H)	(L)

*1. Since the FX-Series MODBUS Communication ADP does not support any program commands, 0000H is stored.

*2. The count is stopped if it has reached FFFFH.
Reset the counter by either of the following methods when restarting the count.
Clearing the counter and diagnostic register (Refer to Subsection 7.11.6).
Restart communications option (Refer to Subsection 7.11.2).
Power OFF → ON, or the PLC state is changed from STOP → RUN.

Note

The communications event counter counts only when the processing (read/write, diagnostics, etc.) has completed normally.
The communications event counter does not count in the case of the following:
The processing has completed with an error.

- When receiving a request message containing a function code that the FX-Series MODBUS Communication ADP does not support
- When receiving the Get communications event counter (FC: 11)
(When completed with an error)

Function code	Data
Function code ($8B_H$)	Exception code *3

*3. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

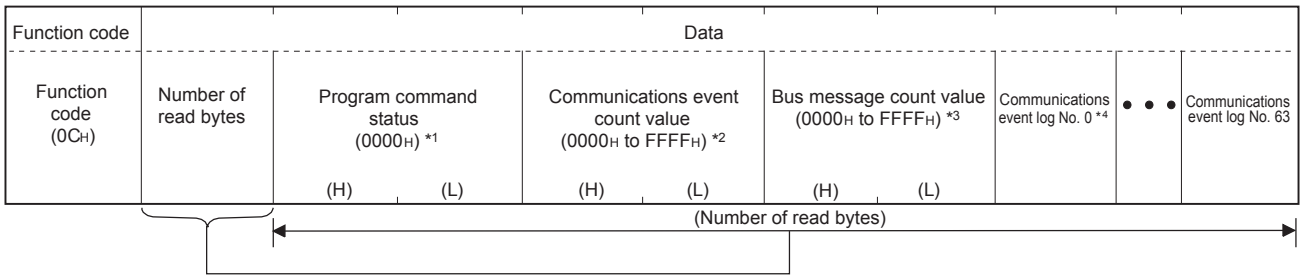
7.13 Get Communications Event Log (Command Code: 0x0C)

Acquires the communications event log of the FX-Series MODBUS Communication ADP into the master.

1) Request message format (Master → Slave)

Function code
Function code ($0C_H$)

2) Response message format (Slave → Master)
(When completed normally)



*1. Since the FX-Series MODBUS Communication ADP does not support any program commands, 0000H is always stored.

*2. Refer to the following for the relevant counts, count clear methods and precautions.
Refer to Section 7.12.

*3. Refer to the following for the relevant counts, count clear methods and precautions.
Refer to Chapter 6.

*4. For information on the Communication Event Log and Communication event log Timing and Communication event log Format refer to Section 9.5.

(When completed with an error)

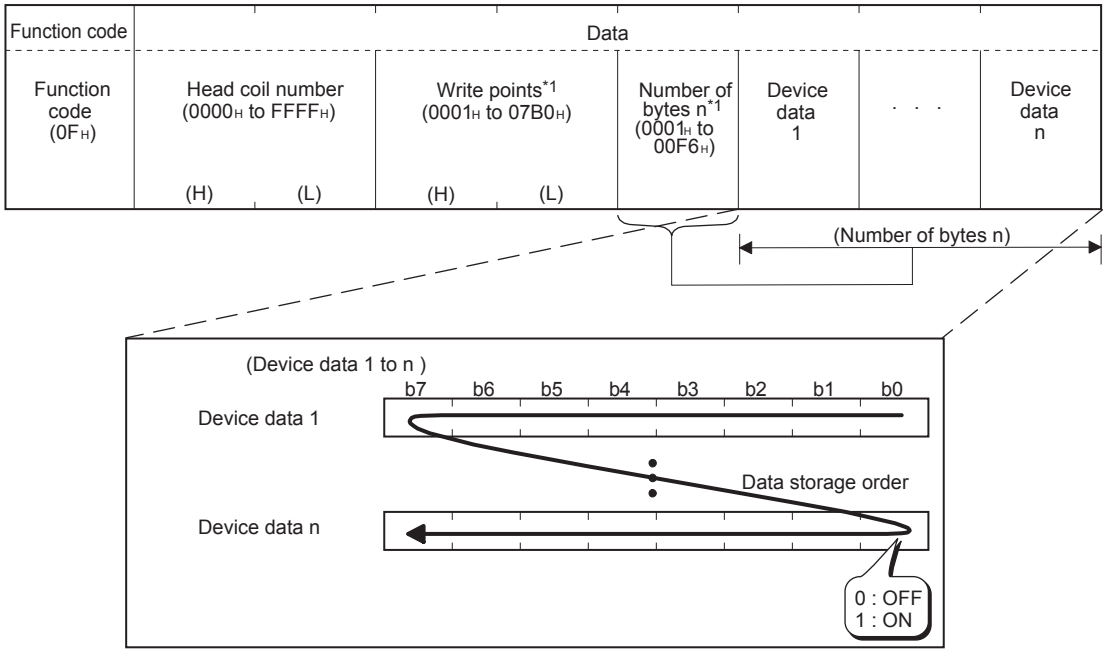
Function code	Data
Function code ($8C_H$)	Exception code *5

*5. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.14 Write Multiple Coils (Command Code: 0x0F)

Writes values (ON/OFF) to multiple coils.

1) Request message format (Master → Slave)



The values (ON/OFF) stored into the device data 1 to n are written to the coils in order from low-order to high-order bits of the device data.

*1. The number of the specified write points must be matched with the number of bits specified as the number of bytes. For example, when the write points are set to 16, set the number of bytes to 2 bytes (= 16 bits).

2) Response message format (Slave → Master)
(When completed normally)

Function code	Data	
Function code (0FH)	Head coil number (The same head coil number value as in the request message is stored.) (H) (L)	Write points (The same write points value as in the request message is stored.) (H) (L)

(When completed with an error)

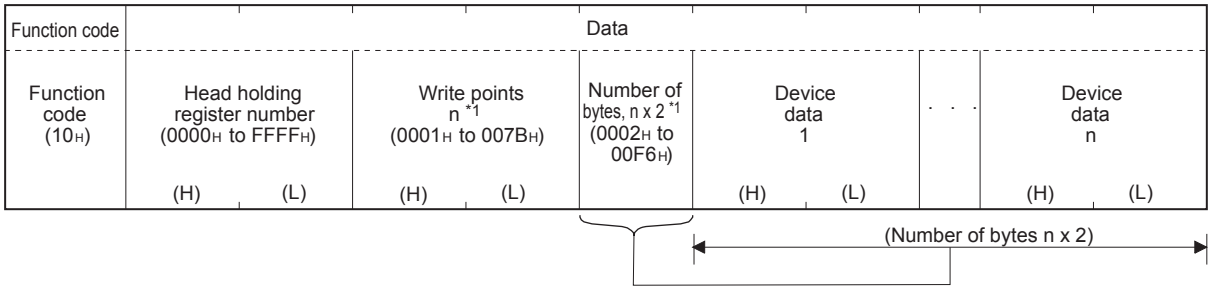
Function code	Data
Function code (8FH)	Exception code*2

*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.15 Write Multiple Registers (Command Code: 0x10)

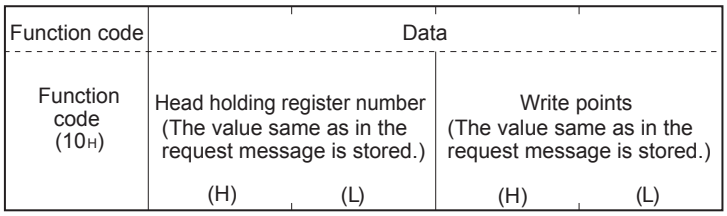
Writes values to multiple holding registers.

1) Request message format (Master → Slave)

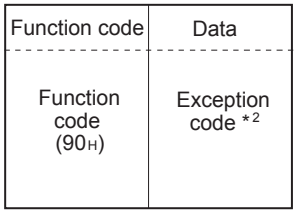


*1. The number of the specified write points must be matched with the number of bytes.

2) Response message format (Slave → Master)
(When completed normally)



(When completed with an error)

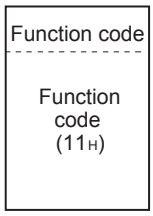


*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

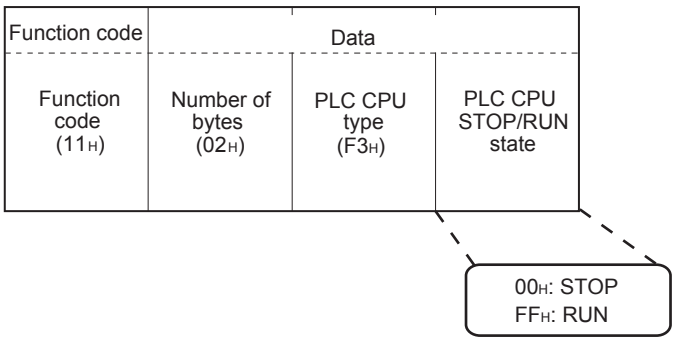
7.16 Report Slave ID (Command Code: 0x11)

Acquires the information of the slave (FX-Series MODBUS Communication ADP) mounted station into the master.

1) Request message format (Master → Slave)

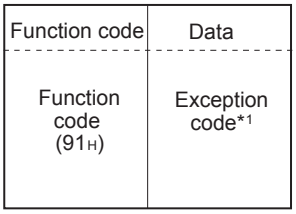


2) Response message format (Slave → Master)
(When completed normally)



The slave (FX-Series MODBUS Communication ADP) will return 'F3' as the PLC CPU type data to the Master:

(When completed with an error)



*1. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.17 Mask Write Register (Command Code: 0x16)

Masks the values stored in a single holding register with AND or OR and writes the value.
The masked values written to the holding register are as shown below.
Result = (Current Contents AND And_Mask) OR (Or_Mask AND (NOT And_Mask)

1) Request Message Format (Master → Slave)

Function code	Data					
Function code (16_H)	Target holding register number (0000_H to $FFFF_H$)		AND mask value (0000_H to $FFFF_H$)		OR mask value (0000_H to $FFFF_H$)	
	(H)	(L)	(H)	(L)	(H)	(L)

2) Response message format (Slave → Master)
(When completed normally)

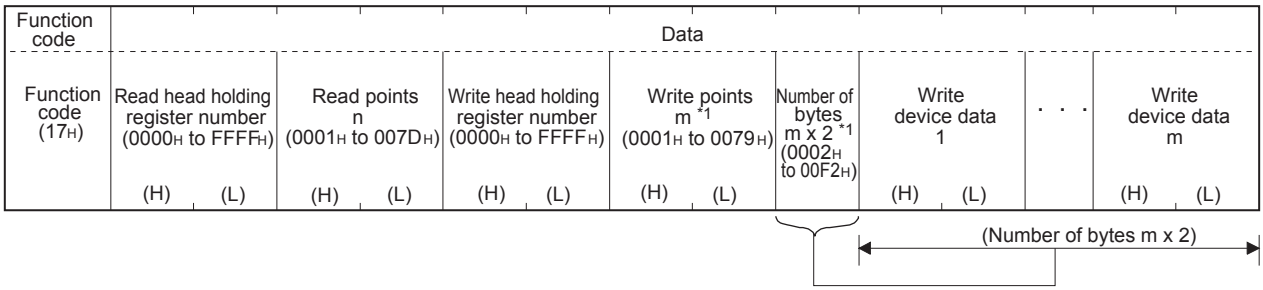
Function code	Data
Function code (96_H)	Exception code*1

*1. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

7.18 Read/Write Multiple Registers (Command Code: 0x17)

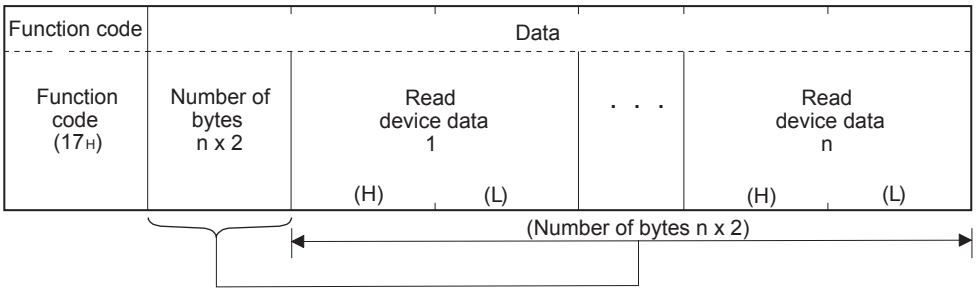
Reads from and writes to multiple holding registers.
Writing is executed first and reading is then executed.

1) Request message format (Master → Slave)

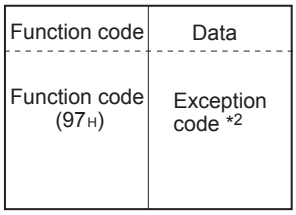


*1. The number of the specified write points must be matched with the number of bytes.

2) Response message format (Slave → Master)
(When completed normally)



(When completed with an error)



*2. Exception and error codes are stored in special data registers and special auxiliary relays in the case of error completion. Refer to Chapter 6 for storage location, confirmation methods, and other detailed contents.

8. Master Specification

This chapter explains the MODBUS Master functions supported by the FX-Series MODBUS Communication ADP.

8.1 MODBUS Master Command List

Command Code	Subcommand Code	Command Name	Details
0x01		Read Coils	Read binary (R/W) devices
0x02		Read Discrete Inputs	Read binary (RO) devices
0x03		Read Holding Registers	Read 16 bit (R/W) register
0x04		Read Input Registers	Read 16 bit (RO) register
0x05		Write Single Coil	Write single binary device
0x06		Write Single Register	Write single 16 bit register device
0x07		Read Exception Status	Read 1 byte of vendor specified data
0x08 Diagnosis	0x00	Return Query Data	Loop back function
	0x01	Restart Communication Option	Restart communication /Remote Communication Reset
	0x02	Return Diagnostic Register	Read 16 bit register of vendor specified data
	0x03	Change ASCII Input Delimiter	Change ASCII mode End of Message character
	0x04	Force Listen Only Mode	Switch slave to Listen Only Mode
	0x0A	Clear Counters and Diagnostic Register	Clear all counters and the diagnostic registers
	0x0B	Return Bus Message Count	Read number of detected messages
	0x0C	Return Bus Communication Error Count	Read number of detected comm. errors
	0x0D	Return Bus Exception Error Count	Read number of detected exception conditions
	0x0E	Return Slave Message Count	Read number of received requests
	0x0F	Return Slave No Response Count	Read "No Response" counter of the slave
	0x10	Return Slave NAK Count	Read NAK counter of the slave
	0x11	Return Slave Busy Count	Read "Busy" counter of the slave
	0x12	Return Bus Character Overrun Count	Read "Bus Character Overrun" counter of the slave
0x0B		Get Comm. Event Counter	Read comm. event counter
0x0C		Get Comm. Event Log	Read comm. event log
0x0F		Write Multiple Coils	Write multiple binary (R/W) devices
0x10		Write Multiple Registers	Write multiple 16 bit (R/W) registers
0x11		Report Slave ID	Read Slave ID code data
0x16		Mask Write Register	Manipulate slave register with AND Mask / OR Mask
0x17		Read/Write Multiple Registers	Read/Write multiple 16 bit (R/W) registers

8.2 FNC 276 - MODBUS Read/Write Instruction

FX3U Ver.2.40

8.2.1 Outline

This instruction allows the MODBUS Master to communicate (read/write data) with its associated Slaves.

1) Instruction Format

FNC 276 ADPRW	16-bit Instruction	Mnemonic	Operation Condition	32-bit Instruction	Mnemonic	Operation Condition
	11 steps ADPRW		Continuous Operation		—	

2) Set Data

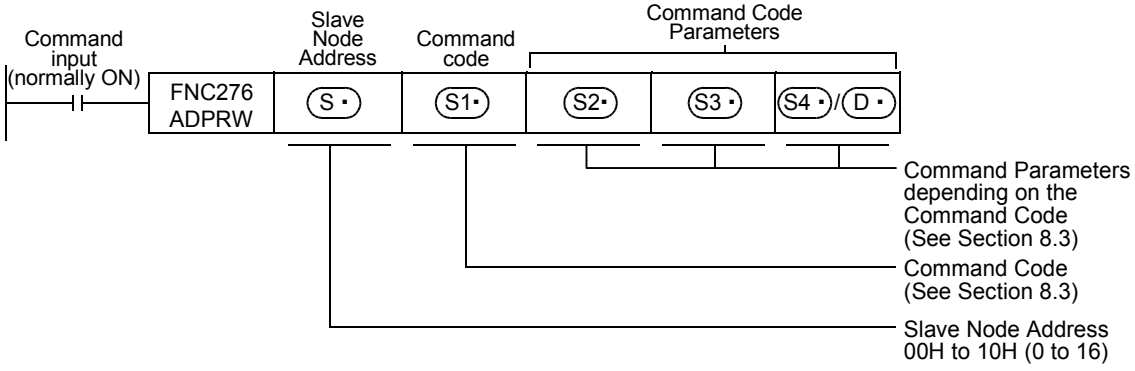
Operand Type	Description	Data Type
(S•)	Slave Node Address	16-bit binary
(S1•)	Command Code	16-bit binary
(S2•)	Command Parameter depending on the Command Code (See Section 8.3)	16-bit binary
(S3•)	Command Parameter depending on the Command Code (See Section 8.3)	16-bit binary
(S4•) / (D•)	Command Parameter depending on the Command Code (See Section 8.3)	Bit or 16-bit binary

3) Applicable Devices

Operand Type	Bit Devices							Word Devices												Others				
	System User						Digit Specification				System User				Special Unit	Index			Constant		Real Number	Character String	Pointer		
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	Modify	K	H	E	"□"	P	
(S•)														✓	✓				✓	✓	✓				
(S1•)														✓	✓				✓	✓	✓				
(S2•)														✓	✓				✓	✓	✓				
(S3•)														✓	✓				✓	✓	✓				
(S4•) / (D•)	✓	✓	✓			✓								✓	✓				✓	✓	✓				

8.2.2 Explanation of function and operation

1) 16-bit operation (ADPRW)
Command Code (S1•) is operated on Slave Node (S•) according to Parameters (S2•), (S3•), and (S4) / (D•).
Use 0 as the Slave Node Address for Broadcast commands.



8.3 ADPRW Command Parameters

The following table shows the required command parameters for each command code.

(S1•) : Command Code	(S2•) : Device Address/ Sub-command Code	(S3•) : Device Count/Sub-command Data/AND Mask	(S4•) / (D•) :Source Data / Destination PLC Device/OR Mask	
	Applicable Devices: D • R • indexing • K • H			
1H Read Coils	MODBUS Address: 0000H~FFFFH	Device Count:1~2000	PLC Destination Device (head address)	
			Applicable Devices	D • R • M • Y • S • indexing
			Block Length	(S3• + 15)÷16
2H Read Discrete Inputs	MODBUS Address: 0000H~FFFFH	Device Count:1~2000	PLC Destination Device (head address)	
			Applicable Devices	D • R • M • Y • S • indexing
			Block Length	(S3• + 15)÷16
3H Read Holding Register	MODBUS Address: 0000H~FFFFH	Device Count:1~125	PLC Destination Device (head address)	
			Applicable Devices	D • R • indexing
			Block Length	S3•
4H Read Input Register	MODBUS Address: 0000H~FFFFH	Device Count:1~125	PLC Destination Device (head address)	
			Applicable Devices	D • R • indexing
			Block Length	S3•
5H Write Single Coil	MODBUS Address: 0000H~FFFFH	0 (fixed)	PLC Source Device (head address)	
			Applicable Devices	D • R • K • H • M • X • Y • S • indexing 0 = bit OFF 1 = bit ON
			Block Length	1 Point
6H Write Single Register	MODBUS Address: 0000H~FFFFH	0 (fixed)	PLC Source Device (head address)	
			Applicable Devices	D • R • K • H • indexing
			Block Length	1 Point
7H Read Exception State	0 (fixed)	0 (fixed)	PLC Destination Device (head address)	
			Applicable Devices	D • R • indexing
			Block Length	1 Point
8H Diagnosis	Sub-command Code: 0H~4H • AH~12H			
	Sub-function: 0H Loop-back Test	Sub-function Data (loop-back data): 0~65535	Loop-back Test Data (Slave response: echo of S3•)	
			Applicable Devices	D • R • indexing
			Block Length	1 Point

(S1•) : Command Code	(S2•) : Device Address/ Sub-command Code	(S3•) : Device Count/Sub-command Data/AND Mask	(S4•) / (D•) :Source Data / Destination PLC Device/OR Mask
	Applicable Devices: D • R • indexing • K • H		
8H Diagnosis	Sub-function: 1H Restart Communication Note: Resets Slave Listen Only Mode	Sub-function Data: 0x0000: Do Not Reset Event Log 0xFF00: Reset Event Log	(Slave response: echo of (S3•)) **Applicable Devices**: D • R • indexing **Block Length**: 1 Point
	Sub-function: 2H Return Diagnostic Register	0 (fixed)	PLC Destination Device (head address) **Applicable Devices**: D • R • indexing **Block Length**: 1 Point
	Sub-function: 3H Change ASCII Input Delimiter	Sub-function Data (ASCII Mode End of Message Character): 00H~FFH	(Slave response: echo of (S3•)) **Applicable Devices**: D • R • indexing **Block Length**: 1 Point
	Sub-function: 4H Force Listen Only Mode Note: Requires the Restart Communication command to reset (1H)	0 (fixed)	0 (fixed) **Applicable Devices**: D • R • indexing **Block Length**: 0
	Sub-function: AH Clear Counter and Diagnostic Register	0 (fixed)	(Slave response: echo of (S3•)) **Applicable Devices**: D • R • indexing **Block Length**: 1 Point
	Sub-function: BH Return Bus Message Counter	0 (fixed)	PLC Destination Device (head address) **Applicable Devices**: D • R • indexing **Block Length**: 1 Point
	Sub-function: CH Return Bus Communication Error Counter	0 (fixed)	PLC Destination Device (head address) **Applicable Devices**: D • R • indexing **Block Length**: 1 Point
	Sub-function: DH Return Bus Exception Error Counter	0 (fixed)	PLC Destination Device (head address) **Applicable Devices**: D • R • indexing **Block Length**: 1 Point
	Sub-function: EH Return Slave Message Counter	0 (fixed)	PLC Destination Device (head address) **Applicable Devices**: D • R • indexing **Block Length**: 1 Point
	Sub-function: FH Return Slave No Response Counter	0 (fixed)	PLC Destination Device (head address) **Applicable Devices**: D • R • indexing **Block Length**: 1 Point

(S1•) : Command Code	(S2•) : Device Address/ Sub-command Code	(S3•) : Device Count/Sub-command Data/AND Mask	(S4•) / (D•) :Source Data / Destination PLC Device/OR Mask	
	Applicable Devices: D • R • indexing • K • H			
8H Diagnosis	Sub-function: 10H Return NAK Counter	0 (fixed)	PLC Destination Device (head address)	
			Applicable Devices	D • R • indexing
			Block Length	1 Point
	Sub-function: 11H Return Slave Busy Counter	0 (fixed)	PLC Destination Device (head address)	
			Applicable Devices	D • R • indexing
			Block Length	1 Point
	Sub-function: 12H Return Character Overrun Counter	0 (fixed)	PLC Destination Device (head address)	
			Applicable Devices	D • R • indexing
			Block Length	1 Point
BH Get Comm. Event Counter	0 (fixed)	0 (fixed)	PLC Destination Device (head address) (D•) : Programming State (D•) +1: Event Counter	
			Applicable Devices	D • R • indexing
			Block Length	2 Point
CH Get Comm. Event Log	0 (fixed)	0 (fixed)	PLC Destination Device (head address) (D•) : Programming State (D•) +1: Event Counter (D•) +2: Bus Message Counter (D•) +3: Log Length (D•) +4~35: Up to 64 Bytes Event Log (2 log bytes/word)	
			Applicable Devices	D • R • indexing
			Block Length	4~36 Point
FH Write Multiple Coils	MODBUS Address: 0000H~FFFFH	Device Count: 1~1968	PLC Source Device (head address)	
			Applicable Devices	D • R • K • H • M • X • Y • S • indexing
			Block Length	((S3•) + 15)÷16
10H Write Multiple Registers	MODBUS Address: 0000H~FFFFH	Device Count:1~123	PLC Source Device (head address)	
			Applicable Devices	D • R • K • H • indexing
			Block Length	(S3•)
11H Report Slave ID	0 (fixed)	0 (fixed)	PLC Destination Device (head address) (D•) : Slave ID (D•) +1: RUN/STOP State (D•) +2: Additional Data Byte Count (D•) +3~127: Up to 249 Bytes Additional Data (Vendor Specified)	
			Applicable Devices	D • R • indexing
			Block Length	3~128 Point

(S1•) : Command Code	(S2•) : Device Address/ Sub-command Code	(S3•) : Device Count/Sub-command Data/AND Mask	(S4•) / (D•) :Source Data / Destination PLC Device/OR Mask	
	Applicable Devices: D • R • indexing • K • H			
16H Mask Write Register	MODBUS Address: 0000H~FFFFH	AND Mask: 0000H~FFFFH	OR Mask: 0000H~FFFFH	
			Applicable Devices	D • R • K • H • indexing
			Block Length	1 Point
17H Read/Write Multiple Registers	MODBUS Address: (S2•) : Write Address 0000H~FFFFH (S2•) +1: Read Address 0000H~FFFFH	Device Count: (S3•) : Write Count 1~121 (S3•) +1: Read Count 1~125	PLC Destination Device (head address) (S4•) : Write Data 1 (S4•) +1: Write Data 2 (S4•) + (Write Count (S3•)) -1: Write Data ((S3•)) (S4•) + (S3•) : Read Data 1 (S4•) +((S3•)) +1: Read Data 2 (S4•) + (S3•) + (Read Count (S3•) +1) -1: Read Data ((S3•) +1)	
			Applicable Devices	D • R • indexing
			Block Length	Write Count (S3•) + Read Count (S3•) +1

9. Slave Specification

This chapter explains the configuration of the slave for MODBUS Serial Communication.

9.1 MODBUS Slave Command Code List

Command Code	Subcommand Code	Command Name	Details
0x01		Read Coils	Read binary (R/W) devices
0x02		Read Discrete Inputs	Read binary (RO) devices
0x03		Read Holding Registers	Read 16 bit (R/W) register
0x04		Read Input Registers	Read 16 bit (RO) register
0x05		Write Single Coil	Write single binary device
0x06		Write Single Register	Write single 16 bit register device
0x07		Read Exception Status	Read 1 byte of vendor specified data CH1: M8060 - M8067 CH2: M8060 - M8062, M8438, M8064 - M8067
0x08 Diagnosis	0x00	Return Query Data	Loop back function
	0x01	Restart Communication Option	Restart communication - Clear all counters - Recover from listen only mode - Reset event log (if requested)
	0x02	Return Diagnostic Register	Read 16 bit register of vendor specified data CH1: M8060 - M8067 CH2: M8060 - M8062, M8438, M8064 - M8067 Note: High byte is unused.
	0x03	Change ASCII Input Delimiter	Change ASCII mode End of Message character. After response is mode slave will change the delimiter.
	0x04	Force Listen Only Mode	Switch slave to Listen Only Mode Note: When the slave enters Listen Only Mode, all active communication controls are turned off. While the device is in this mode, any MODBUS messages addressed to it or broadcast are monitored, but no actions will be taken and no responses will be sent.
	0x0A	Clear Counters and Diagnostic Register	Clear all counters and the diagnostic registers
	0x0B	Return Bus Message Count	Read number of detected messages
	0x0C	Return Bus Communication Error Count	Read number of detected communication errors
	0x0D	Return Bus Exception Error Count	Read number of detected exception conditions
	0x0E	Return Slave Message Count	Read number of received requests
	0x0F	Return Slave No Response Count	Read "No Response" counter of the slave
	0x10	Return Slave NAK Count	Read NAK counter of the slave
	0x11	Return Slave Busy Count	Read "Busy" counter of the slave
	0x12	Return Bus Character Overrun Count	Read "Bus Character Overrun" counter of the slave

Command Code	Subcommand Code	Command Name	Details
0x0B		Get Communication Event Counter	Read communication event counter
0x0C		Get Communication Event Log	Read communication event log
0x0F		Write Multiple Coils	Write multiple binary (R/W) devices
0x10		Write Multiple Registers	Write multiple 16 bit (R/W) registers
0x11		Report Slave ID	Slave ID details: - PLC run/stop state RUN Status: RUN = FFH STOP = 00H - Slave ID F3H (FX3U - same as computer link) Note: No additional data is supported
0x16		Mask Write Register	Manipulate slave register with AND Mask / OR Mask
0x17		Read/Write Multiple Registers	Read/Write multiple 16 bit (R/W) registers

9.2 MODBUS device address allocation

The following information details device allocation when using default values and how to create user defined device address allocation values.

9.3 MODBUS device address allocation (Default Values)

The following table provides the default values for MODBUS address allocation for Bit devices and word devices.

Bit device:

MODBUS Binary Device Address		FX3U Device
Discrete Inputs (Read Only)	Coils (Read / Write)	
0x0000-0x1DFF	0x0000-0x1DFF	M0-M7679
0x1E00-0x1FFF	0x1E00-0x1FFF	M8000-M8511
0x2000-0x2FFF	0x2000-0x2FFF	S0-S4095
0x3000-0x31FF	0x3000-0x31FF	TS0-TS511
0x3200-0x32FF	0x3200-0x32FF	CS0-CS255
0x3300-0x33FF	0x3300-0x33FF	Y0-Y377
0x3400-0x34FF	-	X0-X377

Word device:

MODBUS Word Device Address		FX3U Device
Input-Register (Read Only)	**Holding-Register (Read / Write)**	
0x0000-0x1F3F	0x0000-0x1F3F	D0-D7999
0x1F40-0x213F	0x1F40-0x213F	D8000-D8511
0x2140-0xA13F	0x2140-0xA13F	R0-R32767
0xA140-0xA33F	0xA140-0xA33F	TN0-TN511
0xA340-0xA407	0xA340-0xA407	CN0-CN199
0xA408-0xA477	0xA408-0xA477	CN200-CN255*1
0xA478-0xA657	0xA478-0xA657	M0-M7679
0xA658-0xA677	0xA658-0xA677	M8000-M8511
0xA678-0xA777	0xA678-0xA777	S0-S4095
0xA778-0xA797	0xA778-0xA797	TS0-TS511
0xA798-0xA7A7	0xA798-0xA7A7	CS0-CS255
0xA7A8-0xA7B7	0xA7A8-0xA7B7	Y0-Y377
0xA7B8-0xA7C7	-	X0-X377

*1. CN200-255 are 32-bit counters.

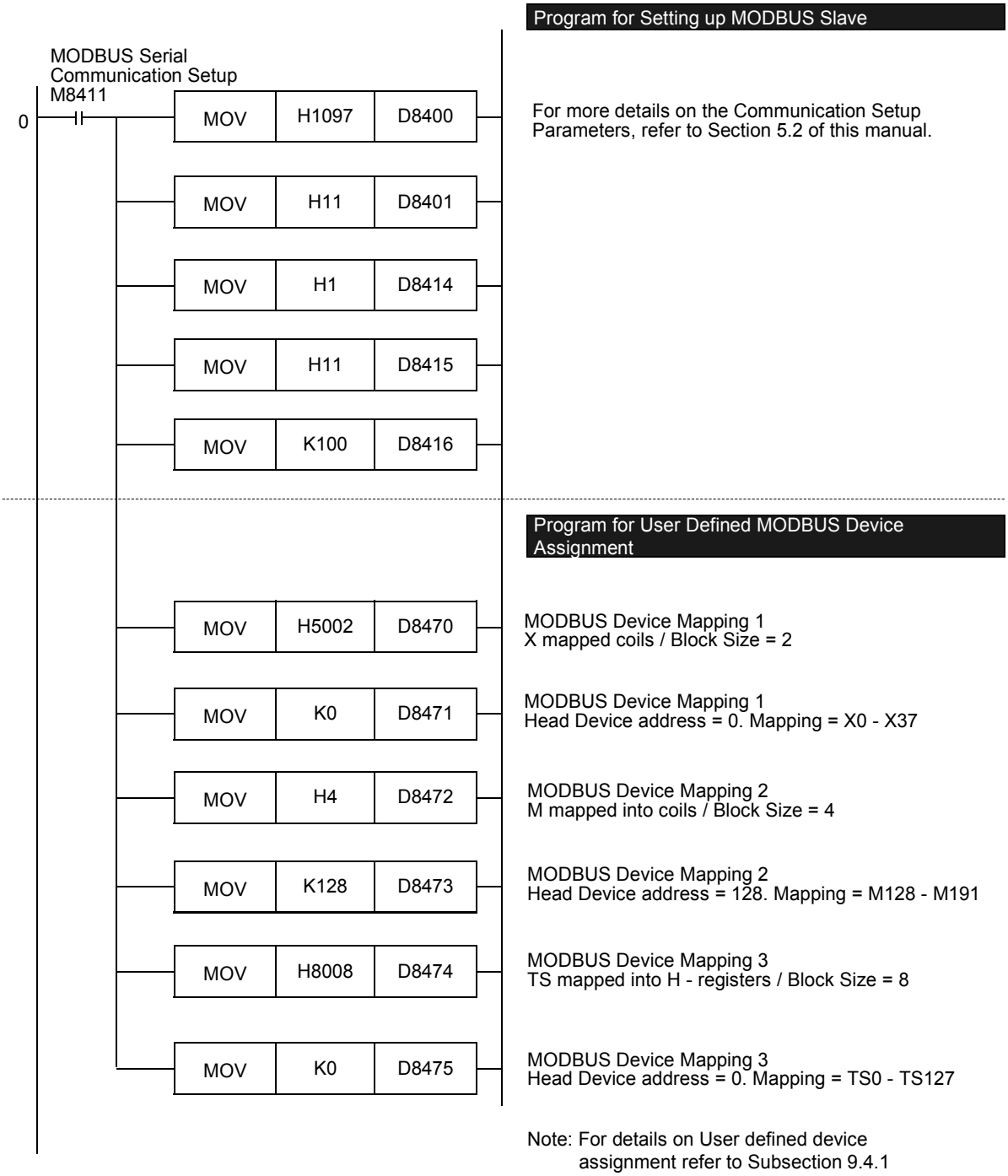
9.4 User defined MODBUS Device Address Assignment

Up to eight PLC device ranges can be mapped to the MODBUS Device Address range in a user defined order. The procedure for creating user defined mapping can be seen in the program example below.

When user defined mapping is set-up by D-registers D8470 to D8485 in the MODBUS Configuration Program, the default MODBUS device assignment becomes invalid and mapping according to the user's program occurs.

Note

When changing the MODBUS configuration the user must reset the power in order that new parameters are recognised.



9.4.1 Format of the user defined device allocation

The user defined device allocation affects only the RW areas - "Coils" and "Holding Register". The mapping of the Read Only (RO) areas "discrete inputs" and "Input Register" is fixed and cannot be changed from the default setting.

One set of configuration information requires two registers where up to eight PLC device areas can be mapped into the MODBUS slave address area. The user defined devices are then mapped to the top of the MODBUS address range of "coils" or "Holding Register". The PLC devices are mapped in the order given by the device allocation data sets 1 to 8 (D8470/D8471 - D8484/D8485).

Note

The values set for D8470 - D8485 by the MOV command are checked at the initialisation phase after power ON. If the values are valid they will be moved into the special data registers D8470 - D8485. In the event that an error is detected, the MOV command will not be executed and the corresponding D-Register and all subsequent registers up to D8485 will be set to 0.

The format of the device allocation data set is as follows:

	MSB D8470		LSB D8471
Device allocation data	① Device code (4bit)	② Size (12bit)	③ PLC head device address (16bit)

Where:
MSB - Most Significant Bit
LSB - Least Significant Bit

① Device code: defines which FX3U device type shall be mapped into "coils" or "Holding Register" (4bit)

0H : M (special M) mapped into "Coils"
1H : S mapped into "Coils"
2H : TS mapped into "Coils"
3H : CS mapped into "Coils"
4H : Y mapped into "Coils"
5H : X mapped into "Coils"
6H : M (special M) mapped into "Holding Registers"
7H : S mapped into "Holding Registers"
8H : TS mapped into "Holding Registers"
9H : CS mapped into "Holding Registers"
AH : Y mapped into "Holding Registers"
BH : X mapped into "Holding Registers"
CH : D (special D) mapped into "Holding Registers"
DH : R mapped into "Holding Registers"
EH : TN mapped into "Holding Registers"
FH : CN mapped into "Holding Registers"

② Size (12bit): 1 to 2048 blocks.

The size of 1 block is defined for the PLC devices as follows:

Bit device (① is 0H to BH):	1 word (16 bit devices)
D and R register (① is CH or DH):	16 word
TN and CN 0~199 (① is EH or FH):	1 word
32 bit counter CN 200~255 (① is FH):	1 double word

Note

If the above range is exceeded, or the selected value exceeds the valid range for the PLC device defined in ① a MODBUS communication error will occur.

③ FX3U PLC head device address (16bit)

Valid values are 0-32767 although this is dependent on the PLC device defined in ①.

Note

For the head device addresses 0H to 5H ① must be a multiple of 8. For the head device 6H to BH ① these values must be multiples of 16. If a device address is selected that is not a multiple of 8 or 16 respectively a MODBUS communication error will occur.

X and Y addressing should always be completed in octal. i.e. 00, 20, 40 etc.

If the selected head device address or its combination with the block length exceeds the valid range for the selected PLC device a MODBUS communication error will occur.

If the setting of ② and ③ is correct and the device mapping is valid, the values will be moved into the special data registers D8470 - D8485. In the event that an error is detected, the MOV will not be executed and the corresponding D-Register and all subsequent registers up to D8485 will be set to 0.

If an error occurs during the device allocation, mapping will be stopped at the first invalid mapping value. However any mapping operations that have been successfully executed before the error occurrence will be effective.

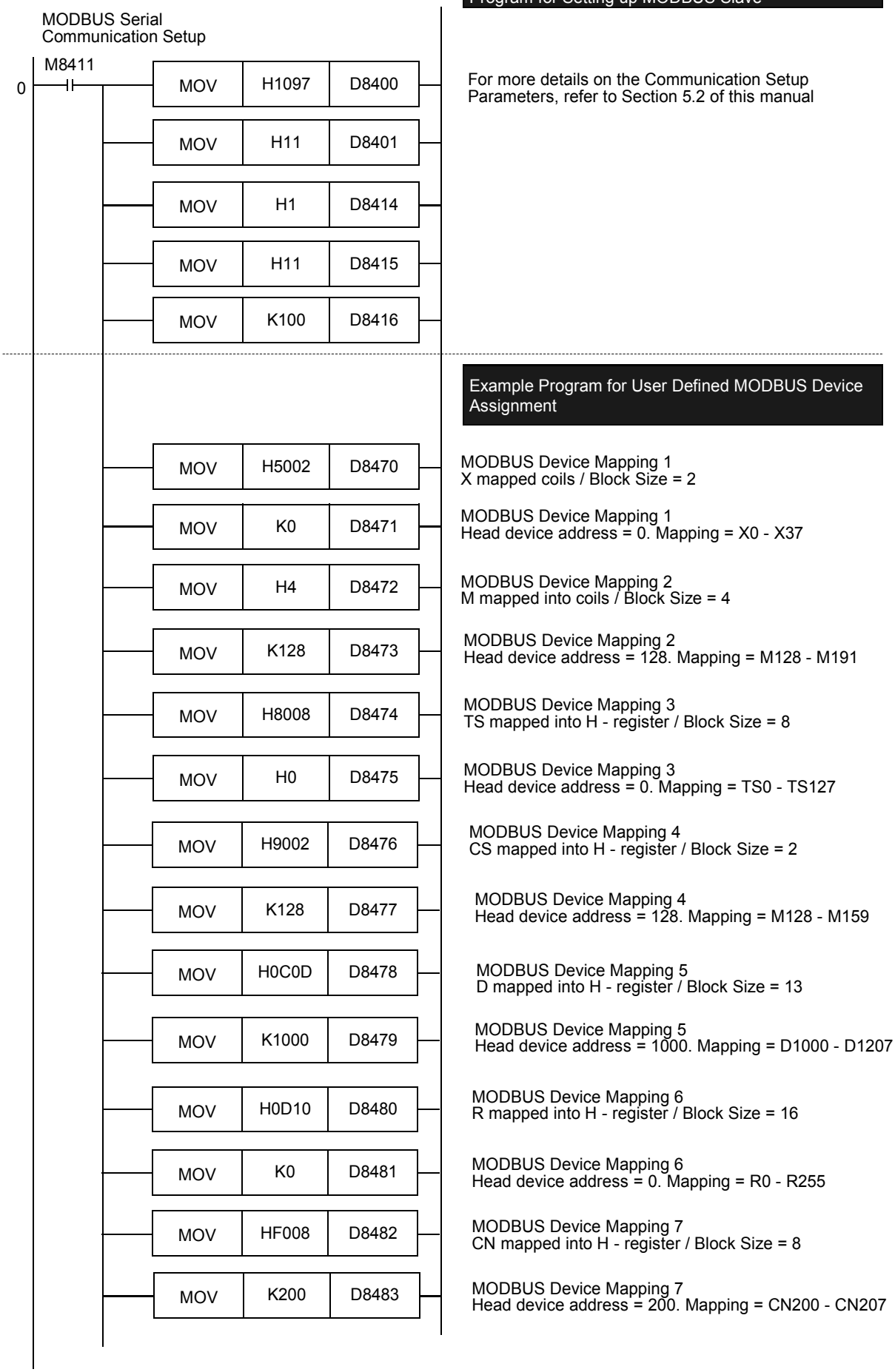
For Auxiliary Relays, Data Registers and Counters it is necessary to separate the mapping for standard and special devices as well as 16 and 32-bit devices. According to this rule it is not possible to map standard Auxiliary Relay (M0-M7679) and Special Auxiliary Relay (M8000-M8511) in the same mapping command. (The same applies for Data Registers & Special Data Registers, 16-bit counters & 32-bit counters).

9.4.2 Example of user defined device allocation

The following table provides an example of a valid user defined device allocation.

Device Allocation Data set	Device Code①		Block Size / Number of Devices②		PLC Head Device Address③		PLC Mapping
1	D8470(4bit)	5H(X)	D8470(12bit)	2	D8471	0	Coil 0-31 → X0-X37
2	D8472(4bit)	0H(M)	D8472(12bit)	4	D8473	128	Coil 32-95 → M128-M191
3	D8474(4bit)	8H(TS)	D8474(12bit)	8	D8475	0	H-Register 0-7 → TS0-TS127
4	D8476(4bit)	9H(CS)	D8476(12bit)	2	D8477	128	H-Register 8-9 → CS128-CS159
5	D8478(4bit)	CH(D)	D8478(12bit)	13	D8479	1000	H-Register 10-217 → D1000-D1207
6	D8480(4bit)	DH(R)	D8480(12bit)	16	D8481	0	H-Register 218-473 → R0-R255
7	D8482(4bit)	FH(CN)	D8482(12bit)	8	D8483	200	H-Register 474-489 → CN200-CN207
8	D8484(4bit)	0	D8484(12bit)	0	D8485	0	Unused

The example program below shows the defined MODBUS defined device allocation for the table on the previous page.


Program for Setting up MODBUS Slave
MODBUS Serial Communication Setup
M8411
0
MOV H1097 D8400
MOV H11 D8401
MOV H1 D8414
MOV H11 D8415
MOV K100 D8416
For more details on the Communication Setup Parameters, refer to Section 5.2 of this manual
Example Program for User Defined MODBUS Device Assignment
MOV H5002 D8470
MODBUS Device Mapping 1
X mapped coils / Block Size = 2
MOV K0 D8471
MODBUS Device Mapping 1
Head device address = 0. Mapping = X0 - X37
MOV H4 D8472
MODBUS Device Mapping 2
M mapped into coils / Block Size = 4
MOV K128 D8473
MODBUS Device Mapping 2
Head device address = 128. Mapping = M128 - M191
MOV H8008 D8474
MODBUS Device Mapping 3
TS mapped into H - register / Block Size = 8
MOV H0 D8475
MODBUS Device Mapping 3
Head device address = 0. Mapping = TS0 - TS127
MOV H9002 D8476
MODBUS Device Mapping 4
CS mapped into H - register / Block Size = 2
MOV K128 D8477
MODBUS Device Mapping 4
Head device address = 128. Mapping = M128 - M159
MOV H0C0D D8478
MODBUS Device Mapping 5
D mapped into H - register / Block Size = 13
MOV K1000 D8479
MODBUS Device Mapping 5
Head device address = 1000. Mapping = D1000 - D1207
MOV H0D10 D8480
MODBUS Device Mapping 6
R mapped into H - register / Block Size = 16
MOV K0 D8481
MODBUS Device Mapping 6
Head device address = 0. Mapping = R0 - R255
MOV HF008 D8482
MODBUS Device Mapping 7
CN mapped into H - register / Block Size = 8
MOV K200 D8483
MODBUS Device Mapping 7
Head device address = 200. Mapping = CN200 - CN207

The following tables provide the values for MODBUS address allocation for Bit devices and word devices for the example stated above:

Bit device:

Coils (Read / Write)	FX3U Device
0x0000 - 0x001F	X0 - X37
0x0020 - 0x005F	M128 - M191

Word device:

Holding-Register (Read / Write)	FX3U Device
0x0000 - 0x0007	TS0 - TS127
0x0008 - 0x0009	CS128 - CS159
0x000A - 0x00D9	D1000 - D1207
0x00DA - 0x01D9	R0 - R255
0x01DA - 0x01E9	CN200 - CN207*1

*1. CN200 - CN207 are 32bit counters.

The following table provides an example of an **invalid** user defined device allocation.

Device Allocation Data set	Device Code		Block size / Number of Devices		PLC Head Device Address		PLC Mapping
1	D8470(4bit)	5H(X)	D8470(12bit)	2	D8471	0	Coil 0-31 →X0-X37
2	D8472(4bit)	0H(M)	D8472(12bit)	4	D8473	128	Coil 32-95 →M128-M191
3	D8474(4bit)	8H(TS)	D8474(12bit)	8	D8475	0	H-Register 0-7 →TS0-TS127
4	D8476(4bit)	9H(CS)→0	D8476(12bit)	2→0	D8477	240→0	NOT MAPPED! CS240-CS271 exceeds the valid range for CS. Error has occurred so the assignment is stopped.
5	D8478(4bit)	CH(D)→0	D8478(12bit)	13→0	D8479	1000→0	NOT MAPPED! Skipped due to error.
6	D8480(4bit)	DH(R)→0	D8480(12bit)	16→0	D8481	0→0	NOT MAPPED! Skipped due to error.
7	D8482(4bit)	FH(CN)→0	D8482(12bit)	16→0	D8483	200→0	NOT MAPPED! Skipped due to error.
8	D8484(4bit)	0	D8484(12bit)	0	D8485	0	Unused

The following tables provide the default values for MODBUS address allocation for Bit devices and word devices for the example stated above:

Bit device:

Coils (Read / Write)	FX3U Device
0x0000 - 0x001F	X0 - X37
0x0020 - 0x005F	M128 - M191

Word device:

Holding-Register (Read / Write)	FX3U Device
0x0000 - 0x0007	TS0 - TS127

9.5 Communication Event log

This section details the communication event log details for the MODBUS slave device.

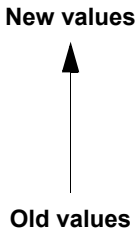
9.5.1 Communications event log

1. Communications event log information in the communication status area

If configured in D8415/D8435 and D8416/D8436 the event log data is displayed in the communication status area. The events of the event log are stored to the communication status area as shown below. For more information on Communication Status refer to Section 6.4.

D8415 = 11H - i.e. store event counter and event log into D devices
D8416 = 100 - i.e. head device is set to D100

	High byte	Low byte
D100- D109	Event and Error counter. For details refer to Section 6.4	
D110	event log length in bytes	
D111	event log byte 1	event log byte 0
D112	event log byte 3	event log byte 2
D113	event log byte 5	event log byte 4
⋮	⋮	⋮
D141	event log byte 61	event log byte 60
D142	event log byte 63	event log byte 62



Note

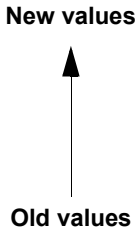
If the number of communications event logs exceeds 64, the oldest log is deleted and the latest log is stored to Communications event log 0.

2. Communications event log response format if ADPRW command is used (For command code 0x0C "get communication event log")

When executing the ADPRW command (command code 0x0C "Get Communication Event Log") the slaves response data is stored in the following format:

Example: Read event log from slave 4
ADPRW K4, H0C, K0, K0, D2000

	High byte	Low byte
D2000	Programming state FFFFH = programming command in process 0000H = no program. Command in process (always 0000H in case of FX3U slave)	
D2001	Event counter	
D2002	Bus message counter	
D2003	event log length in bytes	
D2004	event log byte 1	event log byte 0
D2005	event log byte 3	event log byte 2
D2006	event log byte 5	event log byte 4
⋮	⋮	⋮
D2034	event log byte 61	event log byte 60
D2035	event log byte 63	event log byte 62

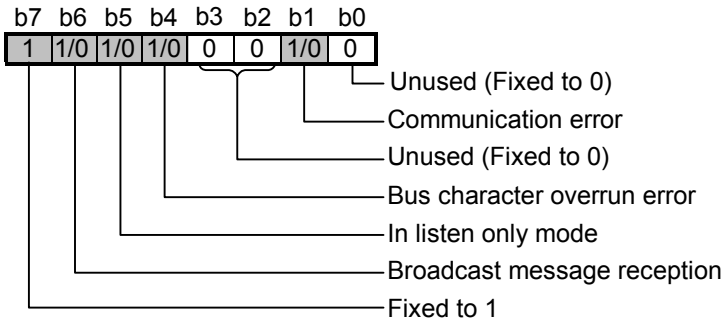


Note

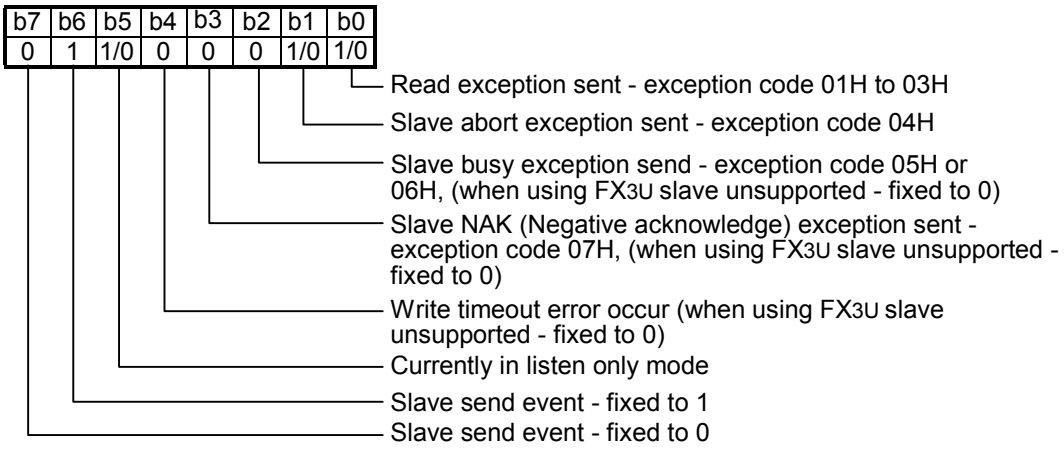
If the number of communications in the event log exceeds 64, the oldest log is deleted and the latest log is stored to Communications event log 0.

9.5.2 Communication event log timing and storage format

1) When receiving a request message:
The slave (FX-Series MODBUS Communication ADP) stores this kind of event to the communications event log before executing the processing of the request message.
For the relevant communications event, "1" is stored.



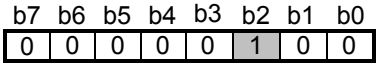
2) When sending a response message:
The slave (FX-Series MODBUS Communication ADP) stores this kind of event to the communications event log after sending the response message.
For the relevant communications event, "1" is stored.



Note

As the FX-Series MODBUS Communication ADP will never be in a state that details exception code 05H to 07H. Therefore the bits 2, 3 & 4 will always read "0" when using a FX-Series MODBUS Communication ADP.

3) When switching to the listen only mode:
The slave (FX-Series MODBUS Communication ADP) stores this kind of event to the communications event log when switching to the listen only mode.
04H is stored to the communications event log.



4) When processing restart communications option:
The slave (FX-Series MODBUS Communication ADP) stores this kind of event to the communications event log when processing the restart communications option.
00H is stored to the communications event log.

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	0

5) Clearing the communications event log
The communications event log can be cleared by either of the following:
Clear setting of the communications event log with the restart communications option (Refer to Subsection 7.11.2).
Power OFF→ ON or PLC state is turned from OFF→ ON.

10. Creating Programs

This chapter explains how to setup FX MODBUS Serial Communication and how to create programs for the Master and Slave stations.

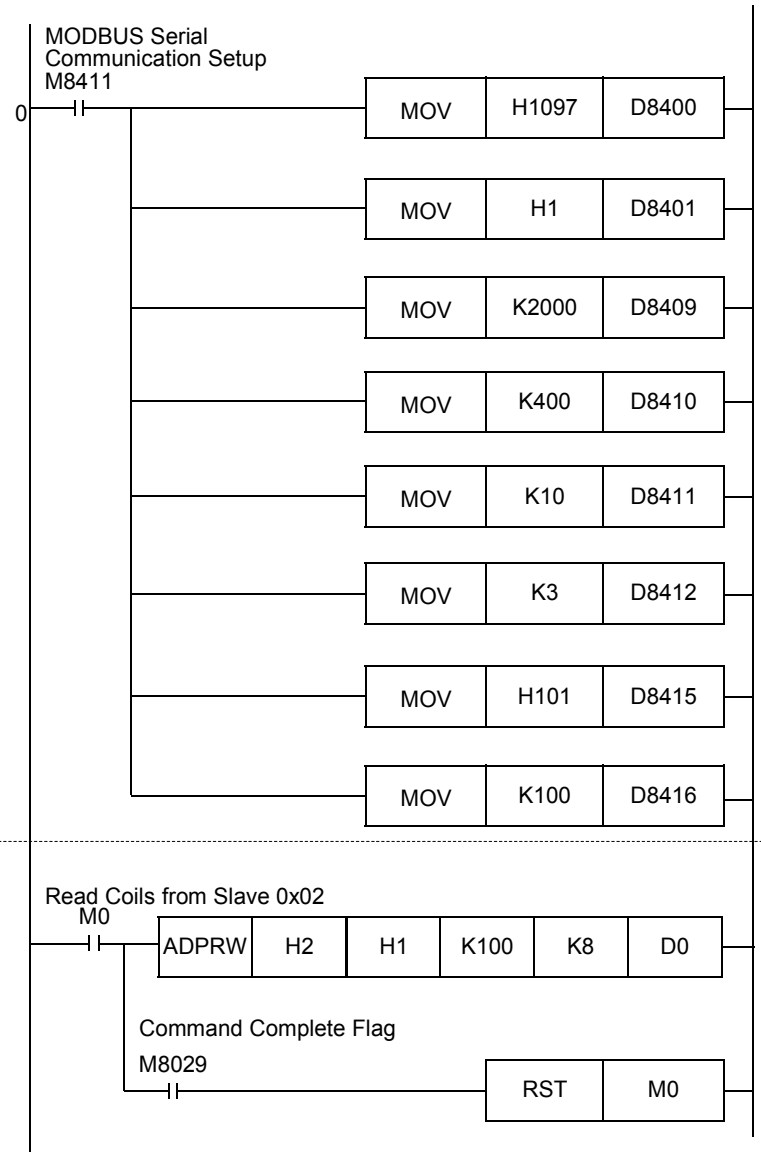
10.1 Checking Contents of Related Devices

For a full description of the PLC devices used by FX MODBUS Serial Communication, please refer to Chapter 6 of this manual.

10.2 Creating Programs for the Master Station

Programs allowing the master station to read and write slave station devices can be created similar to the example below.

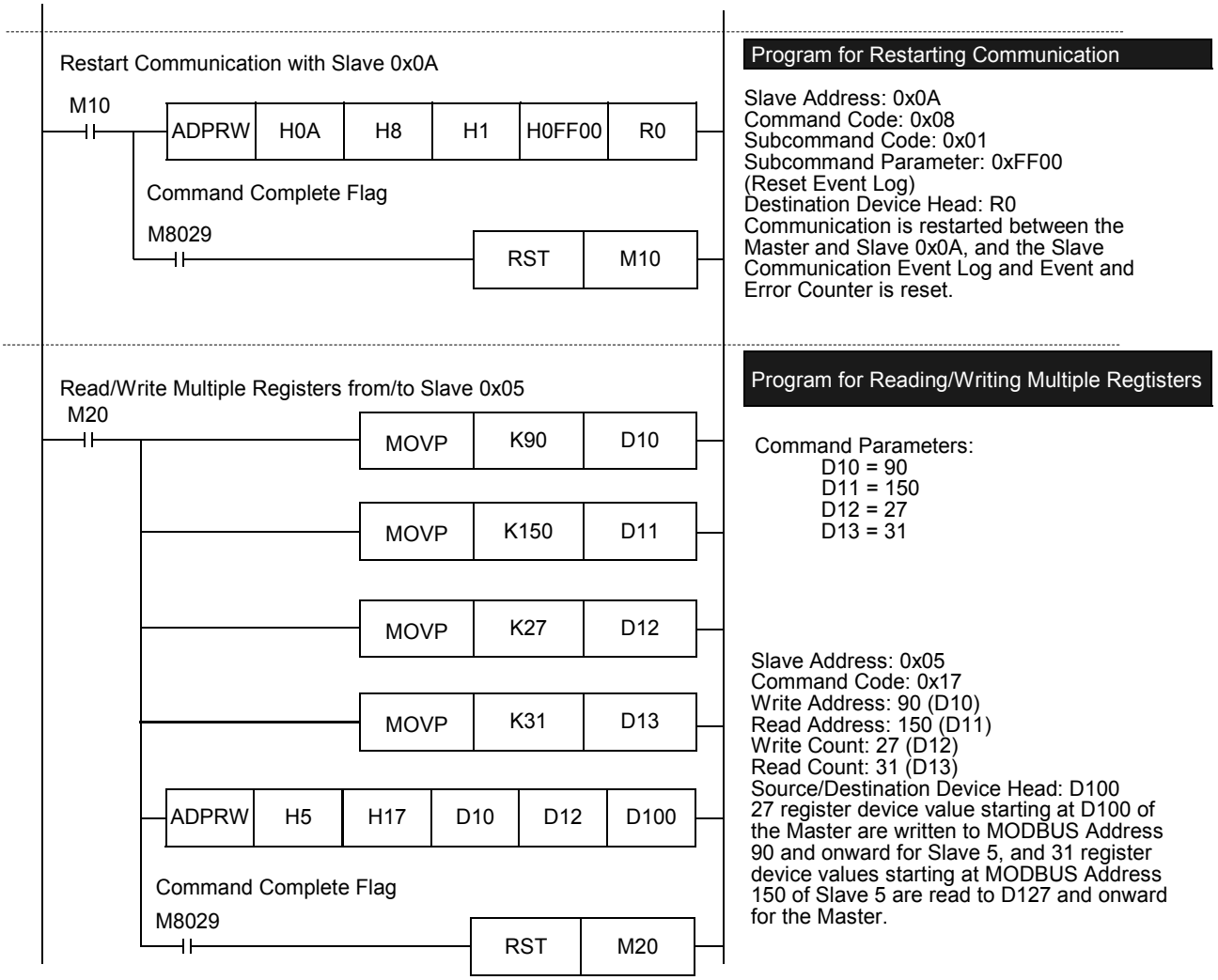
Program for setting up a MODBUS Master



For more details on the Communication Setup Parameters, refer to Section 5.2 of this manual.

Program for Reading Coils

Slave Address: 0x02
Command Code: 0x01
MODBUS Address: 100
Device Count: 8
Destination Device Head: D0
8 coil device values starting at MODBUS Address 100 of Slave 2 are read to the first 8 bits in D0 of the Master.



→ **For more details on Master Station instructions, refer to Chapter 8.**
→ **For cautions on program creation, refer to Section 10.4.**

10.3 Creating Programs for the Slave Station

Create a program for the slave station with user defined MODBUS Device Address Assignment similar to the example program in Section 9.4 of this manual.

10.4 Cautions on Program Creation

1. MODBUS Configuration Request Flag (M8411)

When setting up FX MODBUS Serial Communication on either Channel 1 or 2, be sure to use Special Auxiliary Relay M8411 as shown in Chapter 5 of this manual.

2. Using the ADPRW Command

1) When using the ADPRW command in the FX MODBUS Master, make sure the driving contact of the ADPRW command stays ON until the command has been completed (M8029).
2) When driving multiple ADPRW commands at the same time in the FX MODBUS Master, only one command will be executed at a time. The next ADPRW command in the program is executed after the current command has been completed.

3. Reading Coils

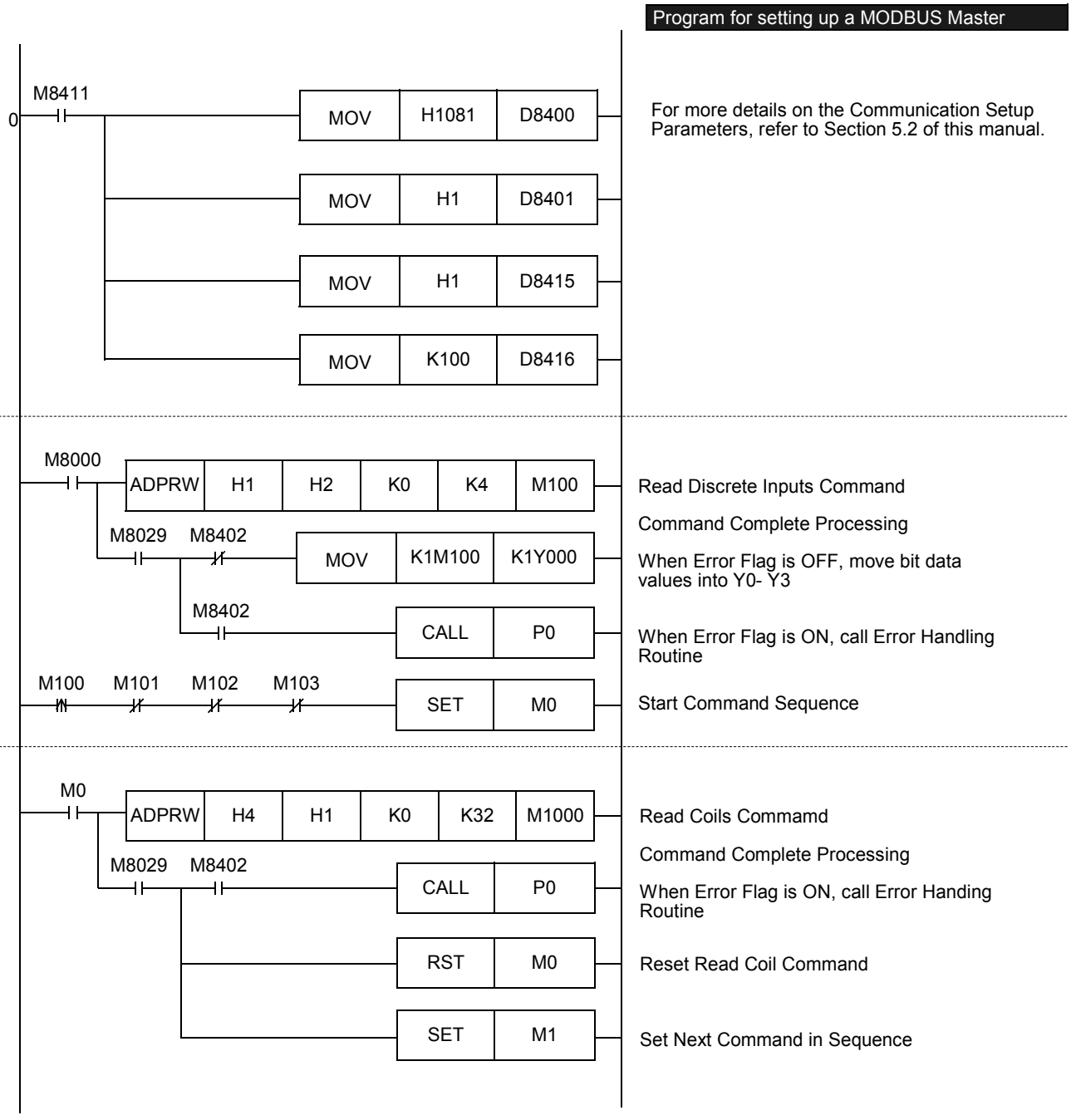
When using the Read Coils command (Command Code 0x01) in the FX MODBUS Master with a word device (i.e. D, or R) as the destination device, only the number of bits assigned in the device count of the ADPRW command will be overwritten. The remaining bits of the word device will not be affected.

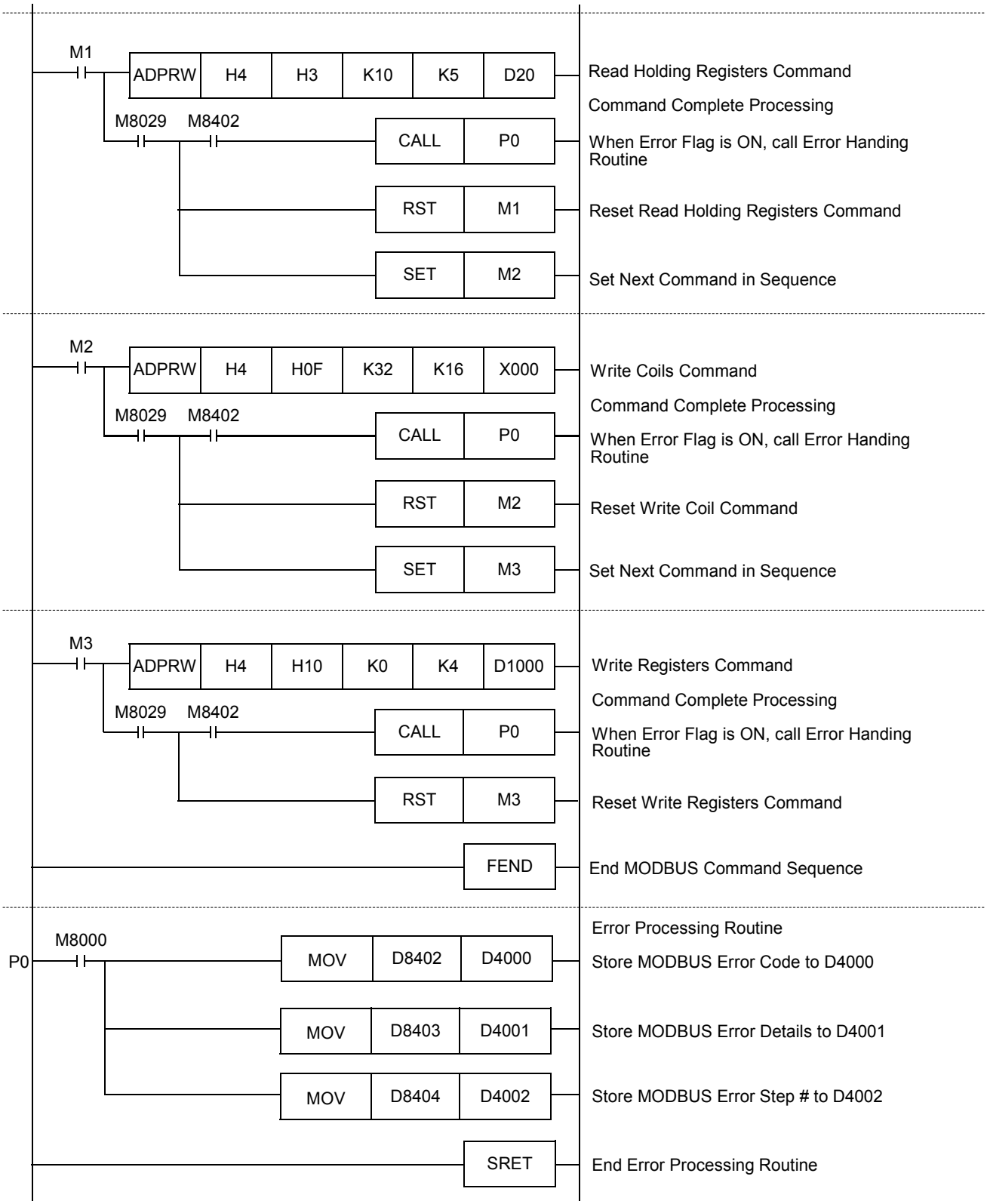
11. Practical Program Examples

This chapter gives practical program examples of how FX MODBUS Serial Communication can be used.

11.1 Setting Program for Master Station

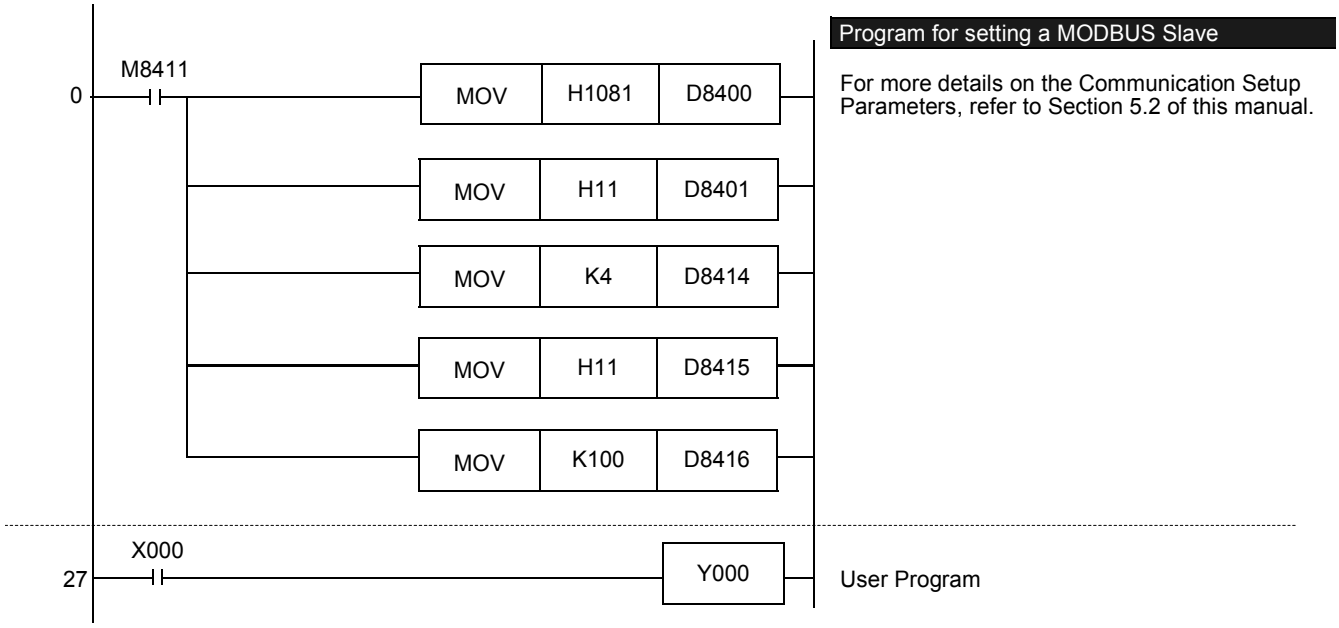
The FX MODBUS Master station can be used to execute a sequence of MODBUS commands in a cycle as shown in the following program. The following program example reads coils, reads holding registers, writes coils, and writes to registers repeatedly while using an Error Handling routine.




M1
ADPRW H4 H3 K10 K5 D20
Read Holding Registers Command
Command Complete Processing
M8029 M8402
CALL P0
When Error Flag is ON, call Error Handing Routine
RST M1
Reset Read Holding Registers Command
SET M2
Set Next Command in Sequence
M2
ADPRW H4 H0F K32 K16 X000
Write Coils Command
Command Complete Processing
M8029 M8402
CALL P0
When Error Flag is ON, call Error Handing Routine
RST M2
Reset Write Coil Command
SET M3
Set Next Command in Sequence
M3
ADPRW H4 H10 K0 K4 D1000
Write Registers Command
Command Complete Processing
M8029 M8402
CALL P0
When Error Flag is ON, call Error Handing Routine
RST M3
Reset Write Registers Command
FEND
End MODBUS Command Sequence
Error Processing Routine
M8000
P0
MOV D8402 D4000
Store MODBUS Error Code to D4000
MOV D8403 D4001
Store MODBUS Error Details to D4001
MOV D8404 D4002
Store MODBUS Error Step # to D4002
SRET
End Error Processing Routine

11.2 Setting Program for Slave Station

After MODBUS Communication Setup, the FX MODBUS Slave station can be used to execute any User Program while the Master station reads and writes devices. An example Slave station program is shown below.



12. Troubleshooting

This chapter explains troubleshooting.

12.1 Checking the FX3U PLC Version Applicability

Verify that the FX3U Series PLC main unit is an applicable version.

→ **For the version applicability check, refer to Section 1.3.**

12.2 Checking the Communication Status Based on LED Indication

Check the status of the "RD" and "SD" indicator LEDs provided in the optional equipment.

LED status		Opeartion status
RD	SD	
Flashing	Flashing	Data is being sent and received.
Flashing	Off	Data is received, but is not sent.
Off	Flashing	Data is sent, but is not received.
Off	Off	Data is not sent nor received.

While MODBUS serial communication is functioning normally, both LEDs flash brightly.
If they are not flashing, check the wiring, communication settings, and error statuses of the master and slave stations.

12.3 Checking the Installation and Wiring

1. Mounting status

Verify that the communication equipment is securely connected with the PLC. If the communication equipment is not securely connected, communication will not function correctly.

→ **For the mounting method, refer to the respective communication equipment manual.**

2. Wiring

Verify that all communication equipment is correctly wired. If the wiring is incorrect, communication will not function correctly.

→ **For the wiring check method, refer to Chapter 4.**

12.4 Checking the Communication Settings and Sequence Program

1. Communication setting using sequence program

Verify that the communication format registers (D8120, D8400 and D8420) are being set correctly. If a communication port is set twice or more, communication is disabled.
After changing any settings, make sure to reboot the PLC's power.

→ **For the MODBUS communication settings, refer to Chapter 5.**

2. Communication setting using parameters

Verify that the communication setting parameters are suitable for use. If the communication setting parameters are not suitable for use, communication will not function correctly. After changing any setting, make sure to reboot the PLC's power.

→ **For the MODBUS communication settings, refer to Chapter 5.**

3. Presence of RS or RS2 instructions

Verify that neither the RS nor RS2 instructions are being used on the same channel as the MODBUS communication.
If either instruction is being used on the same channel, delete it, and then reboot the PLC's power.

4. Presence of IVCK, IVDR, IVRD, IVWR, and IVBWR instructions

Verify that none of the dedicated Inverter communication instructions are being used on the same channel as the MODBUS communication.
If any of the instructions are being used on the same channel, delete it, and then reboot the PLC's power.

12.5 Checking Setting Contents and Errors

1. Checking the setting contents

Each FX PLC has devices for checking the communication settings. Verify that the correct contents are stored in the devices shown in the table below.

Device	Name	Description
D8400	Channel 1 MODBUS Communication Format	For Descriptions, refer to Chapter 6.
D8401	Channel 1 MODBUS Protocol	
D8409	Channel 1 MODBUS Slave ResponseTimeout	
D8410	Channel 1 MODBUS Turn Around Delay	
D8411	Channel 1 MODBUS Message to Message Delay	
D8412	Channel 1 MODBUS Number of Retries	
D8414	Channel 1 MODBUS Slave Node Address	
D8415	Channel 1 MODBUS Communication Status Information Setup	
D8416	Channel 1 MODBUS Communication Status Device Range Setup	

Device	Name	Description
D8420	Channel 2 MODBUS Communication Format	For Descriptions, refer to Chapter 6.
D8421	Channel 2 MODBUS Protocol	
D8429	Channel 2 MODBUS Slave Response Timeout	
D8430	Channel 2 MODBUS Turn Around Delay	
D8431	Channel 2 MODBUS Message to Message Delay	
D8432	Channel 2 MODBUS Number of Retries	
D8434	Channel 2 MODBUS Slave Node Address	
D8435	Channel 2 MODBUS Communication Status Information Setup	
D8436	Channel 2 MODBUS Communication Status Device Range Setup	

If the correct contents are not stored in the above devices, check the sequence program.

2. Checking for setting errors

1) Error flags
If the parameter settings include an error, the serial communication error flag and the MODBUS communication error flag turn ON.
Verify that the devices shown in the table below are OFF.

Device	Name	Description
M8063	Serial Communication Error 1 (ch1)	Turns ON when abnormality occurs using serial communication on ch1.
M8402	MODBUS Communication Error on ch1	Turns ON when a MODBUS command error occurs using ch1.
M8403	MODBUS Communication Error on ch1 (Latched)	Turns ON after a MODBUS command error has occurred using ch1.
M8422	MODBUS Communication Error on ch2	Turns ON when a MODBUS command error occurs using ch2.
M8423	MODBUS Communication Error on ch2 (Latched)	Turns ON after a MODBUS command error has occurred using ch2.
M8438	Serial Communication Error 2 (ch2)	Turns ON when abnormality occurs using serial communication on ch2.

2) Error codes
When a communication error occurs while using MODBUS communication, the corresponding communication error flag turns ON, and the MODBUS error code is stored in the corresponding data register.

Device	Name	Description
D8063	Serial Communication Error Code 1 (ch1)	Set to 6321, representing MODBUS Error on ch1.
D8402	MODBUS Communication Error Code on ch1	Set to corresponding MODBUS Error Code on ch1.
D8403	MODBUS Communication Error Details on ch1	Set to Error Details for MODBUS Error Code in D8402.
D8422	MODBUS Communication Error Code on ch2	Set to corresponding MODBUS Error Code on ch2.
D8423	MODBUS Communication Error Details on ch2	Set to Error Details for MODBUS Error Code in D8422.
D8438	Serial Communication Error Code 2 (ch2)	Set to 3821, representing MODBUS Error on ch2.

→ **For the MODBUS Error Code List, refer to Section 12.6.**

12.6 MODBUS Error Code List

Note

Only one channel can be used for MODBUS serial communication.

MODBUS Error Code	Error Name and Details	Master / Slave	Related Devices:(M & D)	Corrective Action
0201	Invalid Hardware Setup Failed to detect MODBUS communication adapter Details: Channel number 1 or 2	Master / Slave	CH1: M8063 set to ON D8063 set to 6321 M8402 set to ON D8402 set to MODBUS Error Code M8403 set to ON D8403 set to Error Details CH2: M8438 set to ON D8438 set to 3821 M8422 set to ON D8422 set to MODBUS Error Code M8423 set to ON D8423 set to Error Details	Verify that the MODBUS Special Adapters are being used (FX3U-485ADP-MB, or FX3U-232ADP-MB)
0202	Invalid Parameter Setup MODBUS communication parameter settings are invalid Details:Special D register (Device Address) causing the error code e.g. Invalid slave number channel 1: Details (D) 8414	Master / Slave	See above	Invalid Parameter values will not be entered into the corresponding data registers. Check MODBUS configuration program block.
0203	Channel Double Use Single channel used for more than one type of communication (i.e. MODBUS and N:N Networking configured for the same channel)	Master / Slave	See above	Make sure only one Channel is setup for MODBUS Serial Communication.

MODBUS Error Code	Error Name and Details	Master / Slave	Related Devices:(M & D)	Corrective Action
0204	Bit-level Error Parity, overrun (rx register) or framing error	Master / Slave	See above	Check Communication Format data register D8400 or D8420 for errors.
0205	CRC/LRC Error Message CRC/LRC is invalid, or message length ≤ 3 characters (RTU) or ≤ 8 characters (ASCII)	Master / Slave	See above	Check the Communication Format, Turn Around Delay, and Message to Message Delay data registers for errors; D8400 and D8410-D8411, or D8420 and D8430-D8431.
0206	Bus Character Overrun - When more than 256 bytes are received in RTU mode (more than 513 bytes in ASCII mode) - (Slave only) When another telegram is received while the former request is still in process	Master / Slave	See above	Check the Turn Around Delay and Message to Message Delay data registers for errors; D8410-D8411, or D8430-D8431. Also verify that the serial port settings are correct.
0207	Data Length Mismatch The received data does not match the byte count value within the telegram, or the device count exceeds the maximum limit for the command.	Master / Slave	See above	Verify that the Slave is using MODBUS Serial Communication and that the correct command was received. Also verify that the device count of the command is within the limits of the Slave and Master. Protocol error may occur if you don't program correctly.
0208	Unconvertible Character Error When in ASCII mode a byte code can not be converted (any character except '0'-'9' and 'A'-'F' ('a'-'f'))	Master / Slave	See above	See Corrective Actions for Error Code 207.
0209	Unsupported Command Code Error The requested Command Code is invalid or not supported	Slave	See above	Verify that the command used is within the Master and Slave specifications.

<table>
<tr><th>MODBUS Error Code</th><th>Error Name and Details</th><th>Master / Slave</th><th>Related Devices:(M & D)</th><th>Corrective Action</th></tr>
<tr><td>0210</td><td>Invalid Device Address

The selected MODBUS Device Address or the Device Address + Device Count exceeds the supported range of this slave</td><td>Slave</td><td>See above</td><td>Verify that the MODBUS Device Address Allocation of the Slave is set correctly.
Ensure master data is in a valid range for a selected command.
Confirm that the master is accessing valid device ranges.</td></tr>
<tr><td>0211</td><td>Communication Timeout

Timeout occurred after the set number of retries failed</td><td>Master</td><td>See above</td><td>Verify that the Slave Node Address and communication parameters are setup correctly.</td></tr>
<tr><td>0212</td><td>Exception Response Error

Slave answers by exception response
(See Exception Code List at the end of this Section)

Details:
H-Byte: Abnormal function code
L-Byte: Exception code</td><td>Master</td><td>See above</td><td>Verify that the command and command parameters used is within the Master and Slave specifications.</td></tr>
<tr><td>0213</td><td>Slave Node Address Mismatch

The Slave Node address of the response does not match the Slave Node address of the request

Details:
H-Byte: requested Slave Node address
L-Byte: received Slave Node address</td><td>Master</td><td>See above</td><td>See Corrective Actions for Error Code 207.</td></tr>
<tr><td>0214</td><td>Function Code Mismatch

The function code of the response does not match the function code of the request

Details:
H-Byte: requested function code
L-Byte: received function code</td><td>Master</td><td>See above</td><td>See Corrective Actions for Error Code 207.</td></tr>
<tr><td>0215</td><td>Illegal Broadcast Command

Slave receives broadcast request for command unsupported by broadcast function

Details:
<table>
<tr><th></th><th>Non-Diagnosis Commands</th><th>Diagnosis Commands</th></tr>
<tr><td>H-Byte</td><td>0</td><td>Command Code (08H)</td></tr>
<tr><td>L-Byte</td><td>Command Code</td><td>Sub-Command Code</td></tr>
</table></td><td>Slave</td><td>See above</td><td>Verify that the command is within the Slave Specifications and that broadcasting is applicable (Chapter 7).</td></tr>
</table>

MODBUS Error Code	Error Name and Details	Master / Slave	Related Devices:(M & D)	Corrective Action
0216	Illegal Data Value Data value does not match MODBUS specification (i.e. Write Single Coil [5H] value other than OFF [0000H] or ON [FF00H])	Slave	See above	See Corrective Actions for Error Code 207.
0217	Illegal Instruction Use ADPRW command used in Slave mode (D8401 or D8421 bit 4 ON)	Slave	See above	Do not use the ADPRW command in the MODBUS Slave.
0218	ADPRW User Command Error The PLC source/destination device of the ADPRW command is invalid or the occupied PLC device range exceeds the valid area Details: H-Byte: 0 L-Byte: 1-5 according to the invalid parameter of the ADPRW command (S•) to (S4•) / (D•)	Master	See above AND M8067 set to ON D8067 set to 6705 or 6706	Verify that the command is within the Master Specivication and device range.

1. Exception codes supported by FX3U MODBUS Slave

The following table outlines the exception codes supported by the MODBUS slave device.

Exception code	Exception name	Details
01H	Illegal Function	The requested function (code) is unsupported by the slave
02H	Illegal device address	The requested device address or device address + device count exceeds the supported range of this slave
03H	Illegal data value	One of the fields within the request exceeds the allowed value (e.g. the implied length, the device count)
04H	Slave device failure	An unrecoverable error occurred while the slave was processing the request

Warranty

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

If any faults or defects (hereinafter "Failure") found to be the responsibility of Mitsubishi occurs during use of the product within the gratis warranty term, the product shall be repaired at no cost via the sales representative or Mitsubishi Service Company. However, if repairs are required onsite at domestic or overseas location, expenses to send an engineer will be solely at the customer's discretion. Mitsubishi shall not be held responsible for any re-commissioning, maintenance, or testing on-site that involves replacement of the failed module.

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
   a) Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
   b) Failure caused by unapproved modifications, etc., to the product by the user.
   c) When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
   d) Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
   e) Relay failure or output contact failure caused by usage beyond the specified Life of contact (cycles).
   f) Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
   g) Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
   h) Anyother failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued.
   Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

Overseas, repairs shall be accepted by Mitsubishi's local overseas FA Center. Note that the repair conditions at each FA Center may differ.

4. Exclusion of loss in opportunity and secondary loss from warranty liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation of damages caused by any cause found not to be the responsibility of Mitsubishi, loss in opportunity, lost profits incurred to the user or third person by Failures of Mitsubishi products, special damages and secondary damages whether foreseeable or not , compensation for accidents, and compensation for damages to products other than Mitsubishi products, replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

6. Product application

1) In using the Mitsubishi MELSEC programmable logic controller, the usage conditions shall be that the application will not lead to a major accident even if any problem or fault should occur in the programmable logic controller device, and that backup and fail-safe functions are systematically provided outside of the device for any problem or fault.
2) The Mitsubishi programmable logic controller has been designed and manufactured for applications in general industries, etc. Thus, applications in which the public could be affected such as in nuclear power plants and other power plants operated by respective power companies, and applications in which a special quality assurance system is required, such as for Railway companies or Public service purposes shall be excluded from the programmable logic controller applications.
   In addition, applications in which human life or property that could be greatly affected, such as in aircraft, medical applications, incineration and fuel devices, manned transportation, equipment for recreation and amusement, and safety devices, shall also be excluded from the programmable logic controller range of applications.
   However, in certain cases, some applications may be possible, providing the user consults their local Mitsubishi representative outlining the special requirements of the project, and providing that all parties concerned agree to the special circumstances, solely at the users discretion.

Revised History

Date	Revision	Discription
4/2007	A	First Edition

HEADQUARTERS

MITSUBISHI ELECTRIC EUROPE B.V. **EUROPE**
German Branch
Gothaer Straße 8
D-40880 Ratingen
Phone: +49 (0)2102 / 486-0
Fax: +49 (0)2102 / 486-1120

MITSUBISHI ELECTRIC EUROPE B.V. **FRANCE**
French Branch
25, Boulevard des Bouvets
F-92741 Nanterre Cedex
Phone: +33 (0)1 / 55 68 55 68
Fax: +33 (0)1 / 55 68 57 57

MITSUBISHI ELECTRIC EUROPE B.V. **IRELAND**
Irish Branch
Westgate Business Park, Ballymount
IRL-Dublin 24
Phone: +353 (0)1 4198800
Fax: +353 (0)1 4198890

MITSUBISHI ELECTRIC EUROPE B.V. **ITALY**
Italian Branch
Viale Colleoni 7
I-20041 Agrate Brianza (MI)
Phone: +39 039 / 60 53 1
Fax: +39 039 / 60 53 312

MITSUBISHI ELECTRIC CORPORATION **JAPAN**
Office Tower "Z" 14 F
8-12,1 chome, Harumi Chuo-Ku
Tokyo 104-6212
Phone: +81 3 622 160 60
Fax: +81 3 622 160 75

MITSUBISHI ELECTRIC EUROPE B.V. **UK**
UK Branch
Travellers Lane
UK-Hatfield, Herts. AL10 8XB
Phone: +44 (0)1707 / 27 61 00
Fax: +44 (0)1707 / 27 86 95

MITSUBISHI ELECTRIC EUROPE B.V. **SPAIN**
Spanish Branch
Carretera de Rubí 76-80
E-08190 Sant Cugat del Vallés (Barcelona)
Phone: +34 93 / 565 3131
Fax: +34 93 / 589 1579

MITSUBISHI ELECTRIC AUTOMATION **USA**
500 Corporate Woods Parkway
Vernon Hills, IL 60061
Phone: +1 847 478 21 00
Fax: +1 847 478 22 83

EUROPEAN REPRESENTATIVES

GEVA **AUSTRIA**
Wiener Straße 89
AT-2500 Baden
Phone: +43 (0)2252 / 85 55 20
Fax: +43 (0)2252 / 488 60

TEHNIKON **BELARUS**
Oktyabrskaya 16/5, Off. 703-711
BY-220030 Minsk
Phone: +375 (0)17 / 210 46 26
Fax: +375 (0)17 / 210 46 26

Koning & Hartman B.V. **BELGIUM**
Industrial Solutions
Woluwelaan 31
BE-1800 Vilvoorde
Phone: +32 (0)2 / 257 02 40
Fax: +32 (0)2 / 257 02 49

AKHNATON **BULGARIA**
4 Andrej Ljapchev Blvd. Pb 21
BG-1756 Sofia
Phone: +359 (0)2 / 97 44 05 8
Fax: +359 (0)2 / 97 44 06 1

INEA CR d.o.o. **CROATIA**
Losinjska 4 a
HR-10000 Zagreb
Phone: +385 (0)1 / 36 940 - 01/ -02/ -03
Fax: +385 (0)1 / 36 940 - 03

AutoCont Control Systems, s.r.o. **CZECH REPUBLIC**
Jelinkova 59/3
CZ-721 00 Ostrava Svinov
Phone: +420 (0)59 / 5691 150
Fax: +420 (0)59 / 5691 199

AutoCont Control Systems, s.r.o. **CZECH REPUBLIC**
Technologická 374/6
CZ-708 00 Ostrava - Pustkovec
Phone: +420 595 691 150
Fax: +420 595 691 199

B:TECH, a.s. **CZECH REPUBLIC**
Na Ostrove 84
CZ - 58001 Havlickuv Brod
Phone: +420 (0)569 / 408 841
Fax: +420 (0)569 / 408 889

B:TECH, a.s. **CZECH REPUBLIC**
Headoffice
U Borové 69
CZ-580 01 Havlickuv Brod
Phone: +420 569 777 777
Fax: +420 569 777 778

Beijer Electronics A/S **DENMARK**
Lautruphoj 1-3
DK-2750 Ballerup
Phone: +45 (0)70 / 26 46 46
Fax: +45 (0)70 / 26 48 48

Beijer Electronics Eesti OÜ **ESTONIA**
Pärnu mnt.160i
EE-11317 Tallinn
Phone: +372 (0)6 / 51 81 40
Fax: +372 (0)6 / 51 81 49

Beijer Electronics OY **FINLAND**
Jaakonkatu 2
FIN-01620 Vantaa
Phone: +358 (0)207 / 463 500
Fax: +358 (0)207 / 463 501

UTECO A.B.E.E. **GREECE**
5, Mavrogenous Str.
GR-18542 Piraeus
Phone: +30 211 / 1206 900
Fax: +30 211 / 1206 999

MELTRADE Ltd. **HUNGARY**
Fertő utca 14.
HU-1107 Budapest
Phone: +36 (0)1 / 431-9726
Fax: +36 (0)1 / 431-9727

Beijer Electronics SIA **LATVIA**
Vestienas iela 2
LV-1035 Riga
Phone: +371 (0)784 / 2280
Fax: +371 (0)784 / 2281

EUROPEAN REPRESENTATIVES

Beijer Electronics UAB **LITHUANIA**
Savanoriu Pr. 187
LT-02300 Vilnius
Phone: +370 (0)5 / 232 3101
Fax: +370 (0)5 / 232 2980

INTEHSIS srl **MOLDOVA**
bld. Traian 23/1
MD-2060 Kishinev
Phone: +373 (0)22 / 66 4242
Fax: +373 (0)22 / 66 4280

Koning & Hartman B.V. **NETHERLANDS**
Haarlerbergweg 21-23
NL-1101 CH Amsterdam
Phone: +31 (0)20 / 587 76 00
Fax: +31 (0)20 / 587 76 05

Beijer Electronics AS **NORWAY**
Postboks 487
NO-3002 Drammen
Phone: +47 (0)32 / 24 30 00
Fax: +47 (0)32 / 84 85 77

MPL Technology Sp. z o.o. **POLAND**
Ul. Krakowska 50
PL-32-083 Balice
Phone: +48 (0)12 / 630 47 00
Fax: +48 (0)12 / 630 47 01

SIRIUS TRADING & SERVICES SRL **ROMANIA**
Aleea Lacul Morii Nr. 3
RO-060841 Bucuresti, Sector 6
Phone: +40 (0)21 / 430 40 06
Fax: +40 (0)21 / 430 40 02

CRAFT Consulting & Engineering d.o.o. **SERBIA**
Bulevar Svetog Cara Konstantina 80-86
SER-18106 Nis
Phone: +381 (0)18 / 292-24-4/5 , 523 962
Fax: +381 (0)18 / 292-24-4/5 , 523 962

INEA SR d.o.o. **SERBIA**
Karadjordjeva 12/260
SER-113000 Smederevo
Phone: +381 (0)26 / 617 163
Fax: +381 (0)26 / 617 163

CS MTrade Slovensko, s.r.o. **SLOVAKIA**
Vajanskeho 58
SK - 92101 Piestany
Phone: +421 (0)33 / 7742 760
Fax: +421 (0)33 / 7735 144

INEA d.o.o. **SLOVENIA**
Stegne 11
SI-1000 Ljubljana
Phone: +386 (0)1 / 513 8100
Fax: +386 (0)1 / 513 8170

Beijer Electronics Automation AB **SWEDEN**
Box 426
SE-20124 Malmö
Phone: +46 (0)40 / 35 86 00
Fax: +46 (0)40 / 35 86 02

ECONOTEC AG **SWITZERLAND**
Hinterdorfstr. 12
CH-8309 Nürensdorf
Phone: +41 (0)44 / 838 48 11
Fax: +41 (0)44 / 838 48 12

GTS **TURKEY**
Darulaceze Cad. No. 43 KAT. 2
TR-34384 Okmeydani-Istanbul
Phone: +90 (0)212 / 320 1640
Fax: +90 (0)212 / 320 1649

CSC Automation Ltd. **UKRAINE**
15, M. Raskova St., Fl. 10, Office 1010
UA-02002 Kiev
Phone: +380 (0)44 / 494 33 55
Fax: +380 (0)44 / 494-33-66

EURASIAN REPRESENTATIVES

Kazpromautomatics Ltd. **KAZAKHSTAN**
2, Scladskaya str.
KAZ-470046 Karaganda
Phone: +7 3212 / 50 11 50
Fax: +7 3212 / 50 11 50

ELEKTROSTILY **RUSSIA**
Rubzowskaja nab. 4-3, No. 8
RU-105082 Moscow
Phone: +7 495 / 545 3419
Fax: +7 495 / 545 3419

ICOS **RUSSIA**
Industrial Computer Systems ZAO
Ryazanskij Prospekt, 8A, Office 100
RU-109428 Moscow
Phone: +7 495 / 232 0207
Fax: +7 495 / 232 0327

NPP "URALELEKTRA" **RUSSIA**
Sverdlova 11A
RU-620027 Ekaterinburg
Phone: +7 343 / 353 2745
Fax: +7 343 / 353 2461

MIDDLE EAST REPRESENTATIVE

TEXEL ELECTRONICS Ltd. **ISRAEL**
2 Ha´umanut, P.O.B. 6272
IL-42160 Netanya
Phone: +972 (0)9 / 863 08 91
Fax: +972 (0)9 / 885 24 30

AFRICAN REPRESENTATIVE

CBI Ltd. **SOUTH AFRICA**
Private Bag 2016
ZA-1600 Isando
Phone: + 27 (0)11 / 928 2000
Fax: + 27 (0)11 / 392 2354