

Programmable Controller

MELSEC iQ-R
series

MELSEC iQ-R MODBUS and MODBUS/TCP Reference Manual

-RJ71C24
-RJ71C24-R2
-RJ71C24-R4
-RJ71EN71
-R04ENCPU
-R08ENCPU
-R16ENCPU
-R32ENCPU
-R120ENCPU

SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using MELSEC iQ-R series programmable controllers, please read the manuals for the product and the relevant manuals introduced in those manuals carefully, and pay full attention to safety to handle the product correctly.

Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

CONDITIONS OF USE FOR THE PRODUCT

(1) MELSEC programmable controller ("the PRODUCT") shall be used in conditions;

- i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
- ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.

MITSUBISHI ELECTRIC SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI ELECTRIC USER'S, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.

("Prohibited Application")

Prohibited Applications include, but not limited to, the use of the PRODUCT in;

- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
- Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
- Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.

Notwithstanding the above restrictions, Mitsubishi Electric may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi Electric and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTS are required. For details, please contact the Mitsubishi Electric representative in your region.

(3) Mitsubishi Electric shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

INTRODUCTION

Thank you for purchasing the Mitsubishi Electric MELSEC iQ-R series programmable controllers.

This manual describes the frame specifications and MODBUS standard functions to use the MODBUS slave function and MODBUS/TCP slave function of the following modules.

Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the MELSEC iQ-R series programmable controller to handle the product correctly.

Relevant products

RJ71C24, RJ71C24-R2, RJ71C24-R4

RJ71EN71, R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, R120ENCPU

CONTENTS

SAFETY PRECAUTIONS	1
CONDITIONS OF USE FOR THE PRODUCT	1
INTRODUCTION	2
RELEVANT MANUALS	5
TERMS	5
GENERIC TERMS AND ABBREVIATIONS	5
CHAPTER 1 FRAME SPECIFICATIONS	6
1.1 MODBUS Protocol Frame Specifications	6
RTU mode	6
ASCII mode	8
1.2 MODBUS/TCP Protocol Frame Specifications	10
1.3 Protocol Data Unit Formats Grouped by Function	11
CHAPTER 2 MODBUS STANDARD FUNCTION LIST	13
CHAPTER 3 MODBUS STANDARD FUNCTION DETAILS	14
3.1 Read Coils (FC: 01H)	14
3.2 Read Discrete Inputs (FC: 02H)	15
3.3 Read Holding Registers (FC: 03H)	16
3.4 Read Input Registers (FC: 04H)	17
3.5 Write Single Coil (FC: 05H)	18
3.6 Write Single Register (FC: 06H)	19
3.7 Read Exception Status (FC: 07H)	20
3.8 Diagnostics (FC: 08H)	21
Return query data (sub-function code: 00H)	22
Restart communications option (sub-function code: 01H)	23
Return diagnostic register (sub-function code: 02H)	24
Change ASCII input delimiter (sub-function code: 03H)	25
Force listen only mode (sub-function code: 04H)	26
Clear counters and diagnostic register (sub-function code: 0AH)	27
Return bus message count (sub-function code: 0BH)	28
Return bus communication error count (sub-function code: 0CH)	29
Return bus exception error count (sub-function code: 0DH)	30
Return slave message count (sub-function code: 0EH)	31
Return slave no response count (sub-function code: 0FH)	32
Return slave NAK count (sub-function code: 10H)	33
Return slave busy count (sub-function code: 11H)	34
Return bus character overrun count (sub-function code: 12H)	35
Return IOP overrun error count (sub-function code: 13H)	36
Clear overrun counter and flag (sub-function code: 14H)	37
3.9 Get Communications Event Counter (FC: 0BH)	38
3.10 Get Communications Event Log (FC: 0CH)	39
3.11 Write Multiple Coils (FC: 0FH)	41
3.12 Write Multiple Registers (FC: 10H)	42
3.13 Report Slave ID (FC: 11H)	43
3.14 Read File Record (FC: 14H) (SC: 06H)	45
3.15 Write File Record (FC: 15H) (SC: 06H)	47

3.16	Mask Write Register (FC: 16H)	48
3.17	Read/Write Multiple Registers (FC: 17H).....	49

CHAPTER 4 EXCEPTION CODES **50**

INDEX **52**

REVISIONS54
WARRANTY55
TRADEMARKS56

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MELSEC iQ-R MODBUS and MODBUS/TCP Reference Manual [BCN-P5999-1060] (this manual)	Protocol (MODBUS, MODBUS/TCP) that reads and writes data from a target device to a module	e-Manual PDF

Point

e-Manual refers to the Mitsubishi Electric FA electronic book manuals that can be browsed using a dedicated tool.

e-Manual has the following features:

- Required information can be cross-searched in multiple manuals.
- Other manuals can be accessed from the links in the manual.
- The hardware specifications of each part can be found from the product figures.
- Pages that users often browse can be bookmarked.
- Sample programs can be copied to an engineering tool.

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Master	A name for the side that requests function execution
MODBUS device	A device used in communications via the MODBUS protocol
Request message	A message that requests function execution to slaves. In the MODBUS protocol, the master requests function execution to the slaves.
Response message	A message that returns execution results of functions from a slave to the master
Slave	A name for the side that processes execution requests from the master and returns the execution results

GENERIC TERMS AND ABBREVIATIONS

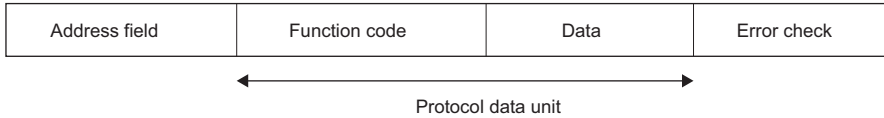
Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Generic term/abbreviation	Description
C24	A MELSEC iQ-R series serial communication module
FC	An abbreviation of function code
SC	An abbreviation of sub code

1 FRAME SPECIFICATIONS

1.1 MODBUS Protocol Frame Specifications

This section describes the frame specifications of the MODBUS protocol.



Area name	Description				
Address field	<ul style="list-style-type: none"> ■ When request message is sent from the master to a slave <ul style="list-style-type: none"> • 0: Sends a request message to all the slaves. (Broadcast) • 1 to 247: Stores the target slave station number. ■ When response message is sent from a slave to the master The host station number is stored when sending a response message. 				
Protocol data unit	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; vertical-align: top;">Function code</td> <td> <ul style="list-style-type: none"> ■ When request message is sent from the master to a slave Specifies the content of the processing instructed from the master to a slave. ■ When response message is sent from a slave 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. </td> </tr> <tr> <td style="vertical-align: top;">Data</td> <td> <ul style="list-style-type: none"> ■ When request message is sent from the master to a slave Stores the request content of the processing. ■ When response message is sent from a slave to the master Stores the execution result of the processing. </td> </tr> </table>	Function code	<ul style="list-style-type: none"> ■ When request message is sent from the master to a slave Specifies the content of the processing instructed from the master to a slave. ■ When response message is sent from a slave 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	<ul style="list-style-type: none"> ■ When request message is sent from the master to a slave Stores the request content of the processing. ■ When response message is sent from a slave to the master Stores the execution result of the processing.
Function code	<ul style="list-style-type: none"> ■ When request message is sent from the master to a slave Specifies the content of the processing instructed from the master to a slave. ■ When response message is sent from a slave 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	<ul style="list-style-type: none"> ■ When request message is sent from the master to a slave Stores the request content of the processing. ■ When response message is sent from a slave to the master Stores the execution result of the processing. 				
Error check	<ul style="list-style-type: none"> The master adds a check code in a request message and transmits the request message. The slave, which received the request message, recalculates the check code in the request message and determines whether the message is correct or not. The message is discarded if it has an error. The error check method differs depending on the frame mode. 				

The following frame modes are available.

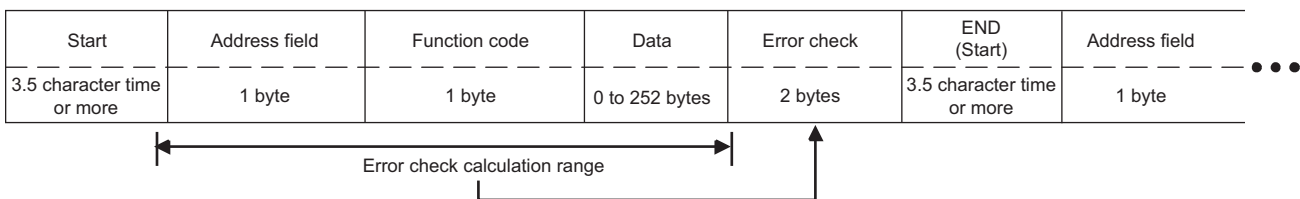
- RTU mode
- ASCII mode

The frame mode of C24 must be consistent with that of the target device.

RTU mode

In this mode, frames are received or sent in binary codes.

The frame specifications are compliant with the MODBUS protocol specifications.



Error check in RTU mode

The error check in the RTU mode is conducted by CRC (Cyclic Redundancy Check).

C24 calculates the CRC by the following steps.

Follow the same steps to calculate the CRC when conducting an error check on the target device.

1. Load the 16-bit register whose bits are all '1'.
2. The CRC is calculated every 8 bits from the upper bit of the frame.
Calculate the exclusive OR (XOR) from the 8 bits of the frame and the bits of the step 1.
3. Shift the result of step 2 by 1 bit to the right.

4. If the least significant bit of the step 2 is '1', calculate the exclusive OR (XOR) from the result of step 3 and the generator polynomial (A001H).

If the least significant bit is '0', do not calculate the exclusive OR (XOR), but shift it by 1 bit to the right.

5. Repeat the step 3 and 4 until the bit is shifted up to 8 times.

6. Calculate the exclusive OR (XOR) from the result of step 5 and the next 8 bits of the frame.

7. Repeat the step 3 to 6.

8. Repeat the above operations until the end of the data unit is reached. The final value is a calculated CRC value.

9. The CRC value is stored in the frame in the order of lower 8 bits to upper 8 bits.

Ex.

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

Error check range is as follows:

- Address field (1 byte): 02H
- Function code (1 byte): 07H
- Data (0 bytes)

CRC error checking procedure		16-bit register (MSB)			Flag ^{*1}
1	Load the 16-bit register whose bits are all '1'.	16-bit register whose bits are all '1'.	1111 1111	1111 1111	—
2	Calculate the exclusive OR (XOR) from the first 8 bits of the frame and the bits in the above.	Address field (02H)	—	0000 0010	
		Exclusive OR (XOR)	1111 1111	1111 1101	
3	Shift by 1 bit to the right.	Shift 1	0111 1111	1111 1110	1
4	Since the flag is '1', calculate the exclusive OR (XOR) from the result in 'Shift 1' and generator polynomial (A001H).	Generator polynomial (A001H)	1010 0000	0000 0001	
		Exclusive OR (XOR)	1101 1111	1111 1111	
5	Shift by 1 bit to the right.	Shift 2	0110 1111	1111 1111	1
		Generator polynomial (A001H)	1010 0000	0000 0001	
	Since the flag is '1', calculate the exclusive OR (XOR) from the result in 'Shift 2' and generator polynomial (A001H).	Exclusive OR (XOR)	1100 1111	1111 1110	
		Shift by 1 bit to the right.	Shift 3	0110 0111	
	Shift by 1 bit to the right.	Shift 4	0011 0011	1111 1111	1
	Since the flag is '1', calculate the exclusive OR (XOR) from the result in 'Shift 4' and generator polynomial (A001H).	Generator polynomial (A001H)	1010 0000	0000 0001	
		Exclusive OR (XOR)	1001 0011	1111 1110	
	Shift by 1 bit to the right.	Shift 5	0100 1001	1111 1111	0
	Shift by 1 bit to the right.	Shift 6	0010 0100	1111 1111	1
	Since the flag is '1', calculate the exclusive OR (XOR) from the result in 'Shift 6' and generator polynomial (A001H).	Generator polynomial (A001H)	1010 0000	0000 0001	
		Exclusive OR (XOR)	1000 0100	1111 1110	
	Shift by 1 bit to the right.	Shift 7	0100 0010	0111 1111	0
	Shift by 1 bit to the right.	Shift 8	0010 0001	0011 1111	1
	Since the flag is '1', calculate the exclusive OR (XOR) from the result in 'Shift 8' and generator polynomial (A001H).	Generator polynomial (A001H)	1010 0000	0000 0001	
		Exclusive OR (XOR)	1000 0001	0011 1110	
6	Calculate the exclusive OR (XOR) from the next 8 bits of the frame and the bits in the above.	Function code (07H)	—	0000 0111	—
		Exclusive OR (XOR)	1000 0001	0011 1001	

CRC error checking procedure		16-bit register (MSB)			Flag ^{*1}
7	Shift by 1 bit to the right.	Shift 1	0100 0000	1001 1100	1
	Since the flag is '1', calculate the exclusive OR (XOR) from the result in 'Shift 1' and generator polynomial (A001H).	Generator polynomial (A001H)	1010 0000	0000 0001	
		Exclusive OR (XOR)	1110 0000	1001 1101	
	Shift by 1 bit to the right.	Shift 2	0111 0000	0100 1110	1
		Since the flag is '1', calculate the exclusive OR (XOR) from the result in 'Shift 2' and generator polynomial (A001H).	Generator polynomial (A001H)	1010 0000	
	Exclusive OR (XOR)		1101 0000	0100 1111	
	Shift by 1 bit to the right.	Shift 3	0110 1000	0010 0111	1
		Since the flag is '1', calculate the exclusive OR (XOR) from the result in 'Shift 3' and generator polynomial (A001H).	Generator polynomial (A001H)	1010 0000	
	Exclusive OR (XOR)		1100 1000	0010 1110	
	Shift by 1 bit to the right.	Shift 4	0110 0100	0001 0011	0
	Shift by 1 bit to the right.	Shift 5	0011 0010	0000 1001	1
	Since the flag is '1', calculate the exclusive OR (XOR) from the result in 'Shift 5' and generator polynomial (A001H).	Generator polynomial (A001H)	1010 0000	0000 0001	
Exclusive OR (XOR)		1001 0010	0000 1000		
Shift by 1 bit to the right.	Shift 6	0100 1001	0000 0100	0	
Shift by 1 bit to the right.	Shift 7	0010 0100	1000 0010	0	
Shift by 1 bit to the right.	Shift 8	0001 0010	0100 0001	0	
8	The final value is a calculated CRC value.	CRC value	12H	41H	—

*1 The least significant bit from the previous exclusive OR (XOR)

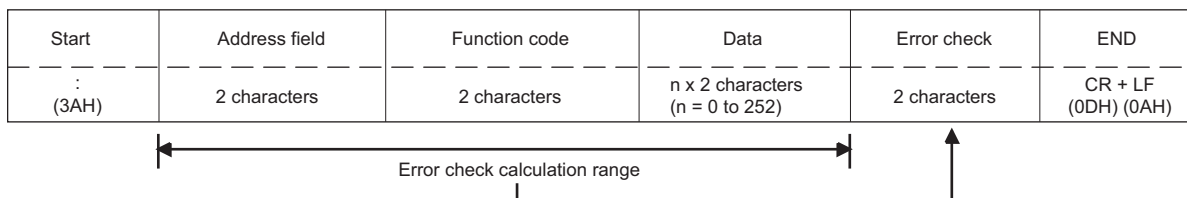
Data is stored as follows:

Address field	Function code	Error check	
		Lower 8 bits of the CRC value	Upper 8 bits of the CRC value
02H	07H	41H	12H

ASCII mode

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

The frame specifications are compliant with the MODBUS protocol specifications.



Error check in ASCII mode

The error check in the ASCII mode is conducted by LRC (Longitudinal Redundancy Check).

C24 calculates the LRC by the following steps.

Follow the same steps to calculate the LRC when conducting an error check on the target device.

1. To calculate the LRC, convert the ASCII codes within the error check range into the RTU format (binary).
2. Add the figures in units of contiguous 8 bits in the frame. (Excluding carries during addition.)
3. Change the result of step 2 to a 2's complement. (Reverse the bits and add 01H.)
4. Convert the result of step 3 to an ASCII code.

Ex.

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

Error check range of the request message is as follows:

- Address field (1 byte): 02H
- Function code (1 byte): 01H

- Data (2 bytes): Start coil number specification (0000H), Number of read points (0008H)

Error check procedure		LRC in request message transmission	
1	Convert the ASCII codes within the error check range into the RTU format (binary).	Address field (02H)	0000 0010
		Function code (01H)	0000 0001
		Start coil number specification (H)	0000 0000
		Start coil number specification (L)	0000 0000
		Read points (H)	0000 0000
		Read points (L)	0000 1000
2	Add the figures. (Excluding carries during addition.)	Addition result	0000 1011
3	Change the result to a 2's complement. (Reverse the bits and add 01H.)	Reversed bit value	1111 0100
		01H	0000 0001
		LRC value	1111 0101
4	Convert the result to an ASCII code.	LRC value (ASCII code)	F 5
			46H 35H

Data is stored in ASCII code as follows:

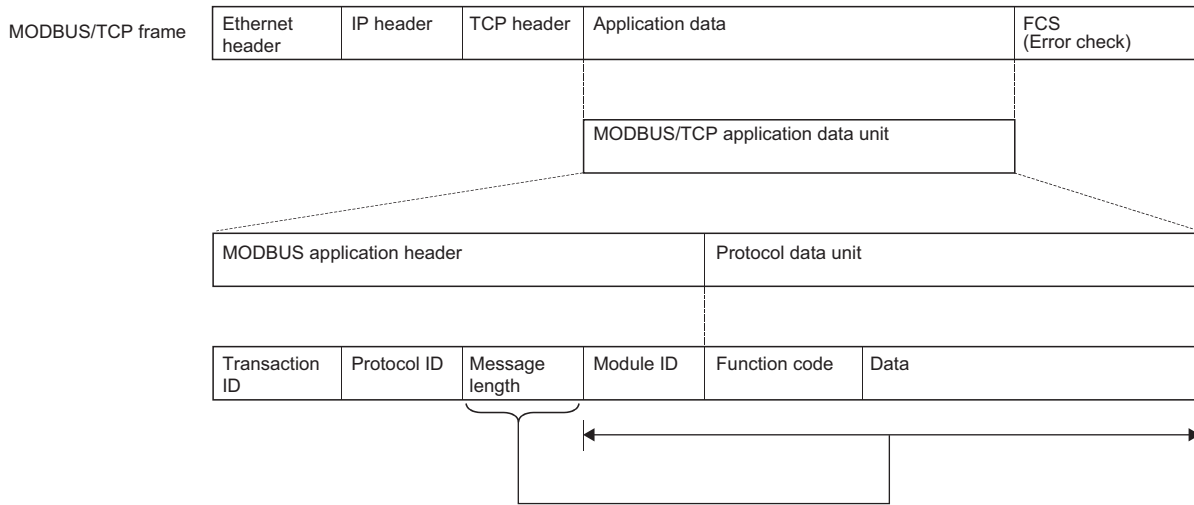
Data category	Address field		Function code		Start coil number specification				Read points				Error check	
Value	02H		01H		0000H				0008H				F5H	
Character	0	2	0	1	0	0	0	0	0	0	0	8	F	5
RTU (binary)	0000	0010	0000	0001	0000	0000	0000	0000	0000	0000	0000	1000	1111	0101
ASCII code	30H	32H	30H	31H	30H	30H	30H	30H	30H	30H	30H	38H	46H	35H

The slave, which received the request message, recalculates the check code in the request message and determines whether the message is correct or not.

Error check procedure		LRC check when receiving a request message	
1	Convert the ASCII codes within the error check range into the RTU format (binary).	Address field (02H)	0000 0010
		Function code (01H)	0000 0001
		Start coil number specification (H)	0000 0000
		Start coil number specification (L)	0000 0000
		Read points (H)	0000 0000
		Read points (L)	0000 1000
2	Add the LRC value. (Excluding carries during addition.) Normally, the addition result is zero.	LRC value	1111 0101
		Addition result	0000 0000

1.2 MODBUS/TCP Protocol Frame Specifications

This section describes the frame specifications of the MODBUS/TCP protocol.



Area name		Area size	Description
MODBUS application header	Communication ID	2 bytes	Used by the master to match response messages from slaves.
	Protocol ID	2 bytes	Indicates the protocol of the protocol data unit. For MODBUS/TCP, 0 is stored.
	Message length	2 bytes	Stores the message size in bytes. The stored message length indicates the length of the message in areas subsequent to this area.
	Module ID	1 byte	Used when specifying slaves connected to other lines, such as with the MODBUS Serial protocol.
Protocol data unit	Function code	1 byte	Specifies the content of the processing instructed from the master to a slave.
	Data	1 to 252 bytes	<ul style="list-style-type: none"> ■When request message is sent from the master to a slave Stores the request content of the processing. ■When response message is sent from a slave to the master Stores the execution result of the processing.

1.3 Protocol Data Unit Formats Grouped by Function

This section describes the protocol data unit formats of the MODBUS standard functions.

When the device number is specified within the message

When specifying the device number within the message, specify "device number - 1".

However, this instruction is not applicable to the file number and the device number specified for reading or writing extended file register.

Ex.

When the status of input 32 (100032) is read by the read discrete inputs (function code: 02H)

Function code	Data	
	Start input number specification	Number of read points
02H	001FH (H) ← (1) (L)	0001H (H) (L)

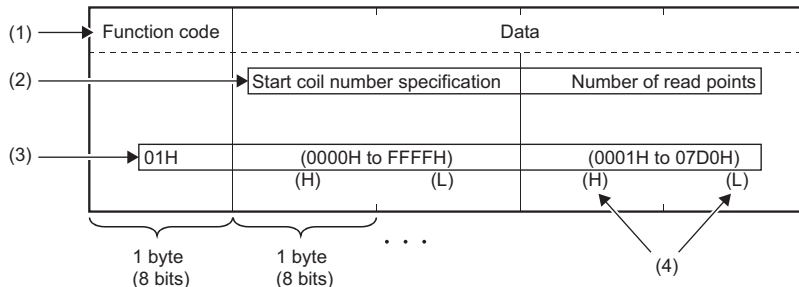
(1) When reading the status of input 32 (100032), specify 31 (001FH) for the start input number.

The device number stored in the response message is "the device number of the device that actually performed reading/writing - 1".

Descriptions of request message and response message formats

This section describes descriptions of the request message and response message formats of the MODBUS standard functions.

■ Message format for MODBUS RTU mode and MODBUS/TCP



(1) Area name

(2) Frame description

(3) For request message format: Setting range

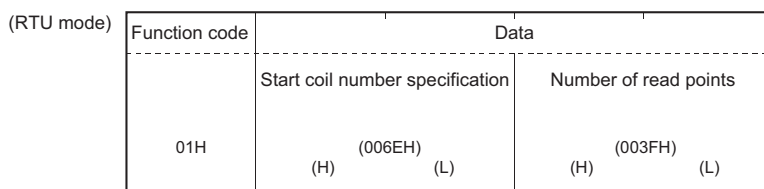
For response message format: Value stored in the response message

(4) When one piece of data consists of two bytes, the upper byte (eight bits) is (H) and the lower byte (eight bits) is (L).

■ Message format for MODBUS ASCII mode

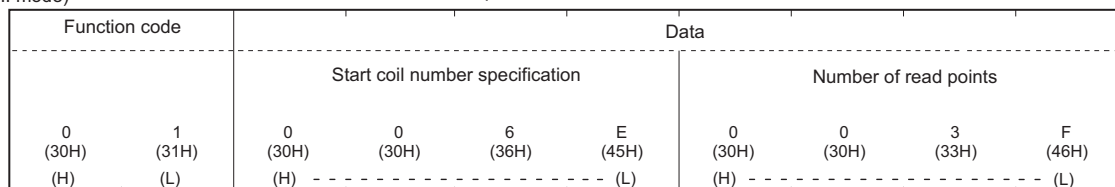
The message format shown in Chapter 3 is for RTU mode or MODBUS/TCP.

In the ASCII mode, read the values shown in Chapter 3 in ASCII code.



Convert RTU mode to ASCII mode

(ASCII mode)



■ Response message format

The format of the response message sent from a slave to the master varies depending on whether the processing performed by the slave completed successfully or completed with an error.

In the MODBUS standard function, the formats when completed successfully and completed with an error are described in the response message format.

Storage locations of exception code and error code

When processing on a slave is completed with an error, an exception code is sent to the master.

For details on exception codes, refer to the following.

Page 50 EXCEPTION CODES

Exception codes are also stored in the buffer memory of modules.

The detailed cause is detected as an error by modules.

For details on the buffer memory and errors of modules, refer to the following.

MELSEC iQ-R Ethernet User's Manual (Application)

MELSEC iQ-R Serial Communication Module User's Manual(Application)


2 MODBUS STANDARD FUNCTION LIST

This chapter lists the MODBUS standard functions and availability for modules when using the slave function.

The modules are abbreviated as follows:

- C24: RJ71C24, RJ71C24-R2, RJ71C24-R4
- E71: RJ71EN71, R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, R120ENCPU

○: Available, ×: Not available

Function code (FC)	Sub code (SC)	Function name	C24	E71	Reference
01H	—	Read coils	○	○	Page 14 Read Coils (FC: 01H)
02H	—	Read discrete inputs	○	○	Page 15 Read Discrete Inputs (FC: 02H)
03H	—	Read holding registers	○	○	Page 16 Read Holding Registers (FC: 03H)
04H	—	Read input registers	○	○	Page 17 Read Input Registers (FC: 04H)
05H	—	Write single coil	○	○	Page 18 Write Single Coil (FC: 05H)
06H	—	Write single register	○	○	Page 19 Write Single Register (FC: 06H)
07H	—	Read exception status	○	×	Page 20 Read Exception Status (FC: 07H)
08H	—	 Page 13 Diagnosis function details	○	×	Page 21 Diagnostics (FC: 08H)
0BH	—	Get communications event counter	○	×	Page 38 Get Communications Event Counter (FC: 0BH)
0CH	—	Get communications event log	○	×	Page 39 Get Communications Event Log (FC: 0CH)
0FH	—	Write multiple coils	○	○	Page 41 Write Multiple Coils (FC: 0FH)
10H	—	Write multiple registers	○	○	Page 42 Write Multiple Registers (FC: 10H)
11H	—	Report slave ID	○	×	Page 43 Report Slave ID (FC: 11H)
14H	06H	Read file record	○	○	Page 45 Read File Record (FC: 14H) (SC: 06H)
15H	06H	Write file record	○	○	Page 47 Write File Record (FC: 15H) (SC: 06H)
16H	—	Mask write register	○	○	Page 48 Mask Write Register (FC: 16H)
17H	—	Read/Write multiple registers	○	○	Page 49 Read/Write Multiple Registers (FC: 17H)

Diagnosis function details

Function code (FC)	Sub-function code	Function name	Reference
08H	0000H	Return query data	Page 22 Return query data (sub-function code: 00H)
	0001H	Restart communications option	Page 23 Restart communications option (sub-function code: 01H)
	0002H	Return diagnostic register	Page 24 Return diagnostic register (sub-function code: 02H)
	0003H	Change ASCII input delimiter	Page 25 Change ASCII input delimiter (sub-function code: 03H)
	0004H	Force listen only mode	Page 26 Force listen only mode (sub-function code: 04H)
	000AH	Clear counters and diagnostic register	Page 27 Clear counters and diagnostic register (sub-function code: 0AH)
	000BH	Return bus message count	Page 28 Return bus message count (sub-function code: 0BH)
	000CH	Return bus communication error count	Page 29 Return bus communication error count (sub-function code: 0CH)
	000DH	Return bus exception error count	Page 30 Return bus exception error count (sub-function code: 0DH)
	000EH	Return slave message count	Page 31 Return slave message count (sub-function code: 0EH)
	000FH	Return slave no response count	Page 32 Return slave no response count (sub-function code: 0FH)
	0010H	Return slave NAK count	Page 33 Return slave NAK count (sub-function code: 10H)
	0011H	Return slave busy count	Page 34 Return slave busy count (sub-function code: 11H)
	0012H	Return bus character overrun count	Page 35 Return bus character overrun count (sub-function code: 12H)
	0013H	Return IOP overrun error count	Page 36 Return IOP overrun error count (sub-function code: 13H)
0014H	Clear overrun counter and flag	Page 37 Clear overrun counter and flag (sub-function code: 14H)	

3 MODBUS STANDARD FUNCTION DETAILS

This chapter describes protocol data unit formats.

3.1 Read Coils (FC: 01H)

Operation description

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

Request message format (from master to slave)

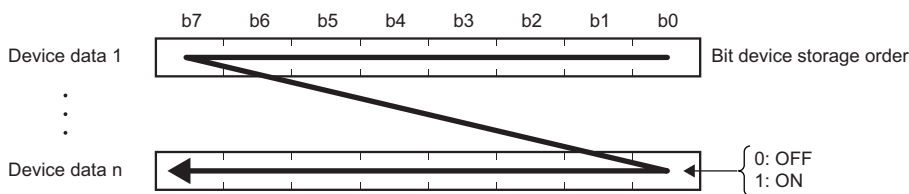
Function code	Data	
01H	Start coil number specification	Number of read points
	(0000H to FFFFH) (H) (L)	(0001H to 07D0H) (H) (L)

Response message format (from slave to master)

■ When completed successfully

Function code	Data			
01H	Number of read bytes n	Device data 1	...	Device data n
	Number of read bytes n			

• Device data 1 to n



- The read coil status is stored in the order of lower bit to higher bit.
- If the number of read points is not a multiple of eight, the remaining bits are all set to 0.

■ When completed with an error

Function code	Data
81H	Exception code

3.2 Read Discrete Inputs (FC: 02H)

Operation description

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

Request message format (from master to slave)

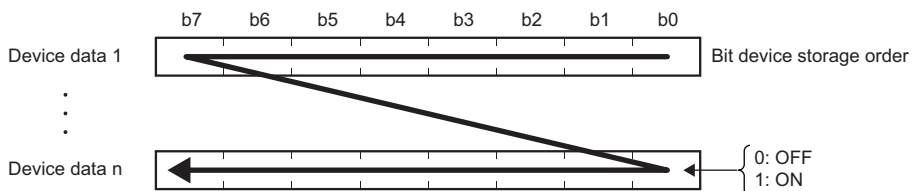
Function code	Data	
02H	Start input number specification	Number of read points
	(0000H to FFFFH) (H) (L)	(0001H to 07D0H) (H) (L)

Response message format (from slave to master)

■ When completed successfully

Function code	Data			
02H	Number of read bytes n	Device data 1	...	Device data n
	Number of read bytes n			

• Device data 1 to n



- The read input status is stored in the order of lower bit to higher bit.
- If the number of read points is not a multiple of eight, the remaining bits are all set to 0.

■ When completed with an error

Function code	Data
82H	Exception code

3.3 Read Holding Registers (FC: 03H)

Operation description

Reads one or multiple holding register values.

Request message format (from master to slave)

Function code	Data	
03H	Start holding register number specification	Number of read points
	(0000H to FFFFH) (H) (L)	(0001H to 007DH) (H) (L)

Response message format (from slave to master)

■ When completed successfully

Function code	Data			
03H	Number of read bytes $n \times 2$	Device data 1	...	Device data n
		(H) (L)		(H) (L)

• For example, when $n = 4$, the number of read bytes is $4 \times 2 = 8$.

■ When completed with an error

Function code	Data
83H	Exception code

3.4 Read Input Registers (FC: 04H)

Operation description

Reads one or multiple input register values.

Request message format (from master to slave)

Function code	Data	
04H	Start input register number specification	Number of read points
	(0000H to FFFFH) (H) (L)	(0001H to 007DH) (H) (L)

3

Response message format (from slave to master)

■ When completed successfully

Function code	Data			
04H	Number of read bytes n×2	Device data 1	...	Device data n
		(H) (L)		(H) (L)

• For example, when n = 4, the number of read bytes is 4 × 2 = 8.

■ When completed with an error

Function code	Data
84H	Exception code

3.5 Write Single Coil (FC: 05H)

Operation description

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

Request message format (from master to slave)

Function code	Data	
05H	Coil number specification	ON/OFF specification
	(0000H to FFFFH) (H) (L)	$\left[\begin{array}{l} \text{FF00H: ON} \\ \text{0000H: OFF} \end{array} \right]$ (H) (L)

Response message format (from slave to master)

■ When completed successfully

The slave returns the request message received from the master as-is.

■ When completed with an error

Function code	Data
85H	Exception code

3.6 Write Single Register (FC: 06H)

Operation description

Writes a value to one holding register area.

Request message format (from master to slave)

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

Response message format (from slave to master)

■ When completed successfully

The slave returns the request message received from the master as-is.

■ When completed with an error

Function code	Data
86H	Exception code

3.7 Read Exception Status (FC: 07H)

Operation description

Reads the error status.

Request message format (from master to slave)

Function code
07H

Response message format (from slave to master)

■ When completed successfully

Function code	Data
07H	Error information

In the error information, the specified device data by "Specifying the error status" of the parameter is stored.

📖 MELSEC iQ-R Serial Communication Module User's Manual(Application)

■ When completed with an error

Function code	Data
87H	Exception code

3.8 Diagnostics (FC: 08H)

Executes the various diagnostics and checks the C24 status and communication status.

In the diagnostics, a sub-function code is specified after a function code with a request message.

Checking communication conditions using a diagnostic counter

When using the MODBUS slave function, the number of error occurrences during communication is stored in a diagnostic counter.

The communication status of MODBUS can be checked by a diagnostic counter of the MODBUS standard function.

Diagnostic counter

Counter		Buffer memory		MODBUS standard function
Type	Description	CH1	CH2	
Bus message count* ¹	Counts the number of messages detected on the line.	32512 (7F00H)	32576 (7F40H)	☞ Page 28 Return bus message count (sub-function code: 0BH)
Bus communication error count* ¹	Counts the number of the following error messages detected on the line. <ul style="list-style-type: none"> • CRC/LRC error message • Overrun/parity error • Short frame (less than 3 bytes) • Character overrun (256 bytes or more) Messages other than the above are counted by the bus message count.	32513 (7F01H)	32577 (7F41H)	☞ Page 29 Return bus communication error count (sub-function code: 0CH)
Exception error count	Counts the number of exception error occurrences. (Including broadcast messages)	32522 (7F0AH)	32586 (7F4AH)	☞ Page 30 Return bus exception error count (sub-function code: 0DH)
Slave message count	Counts the number of times that messages addressed to the host were processed. (Including when broadcast request messages were received)	32518 (7F06H)	32582 (7F46H)	☞ Page 31 Return slave message count (sub-function code: 0EH)
Slave no-response count	Counts the number of times that broadcast request messages were received.	32519 (7F07H)	32583 (7F47H)	☞ Page 32 Return slave no response count (sub-function code: 0FH)
Slave NAK count	Refers to the number of times that NAK responses were received in a master device. It is always '0' in C24s.	32520 (7F08H)	32584 (7F48H)	☞ Page 33 Return slave NAK count (sub-function code: 10H)
Slave busy count	Refers to the number of times that busy responses were received in a master device. It is always '0' in C24s.	32521 (7F09H)	32585 (7F49H)	☞ Page 34 Return slave busy count (sub-function code: 11H)
Character overrun count	Counts the number of times that the request message size exceeded the upper limit.	32514 (7F02H)	32578 (7F42H)	☞ Page 35 Return bus character overrun count (sub-function code: 12H) ☞ Page 36 Return IOP overrun error count (sub-function code: 13H)

*1 The bus message count is in an exclusive relationship with the bus communication error count.

Count range

0000H to FFFFH are counted.

The count is stopped if it has reached FFFFH. To continue, clear the counter.

Methods for clearing a counter

A counter can be cleared by any of the following methods:

- Turning the power OFF and ON
- Resetting a CPU module
- Clearing the buffer memory to '0' by using a sequence program
- Receiving the following MODBUS standard functions

Counter to be cleared	MODBUS standard function
All diagnostic counters shown above	☞ Page 23 Restart communications option (sub-function code: 01H) ☞ Page 27 Clear counters and diagnostic register (sub-function code: 0AH)
Character overrun count	☞ Page 37 Clear overrun counter and flag (sub-function code: 14H)

Return query data (sub-function code: 00H)

Operation description

Returns the contents of the request message without change.

Used to check if the network or the target device is operating normally. (Loopback test)

Request message format (from master to slave)

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

Response message format (from slave to master)

■ When completed successfully

The slave returns the request message received from the master as-is.

■ When completed with an error

Function code	Data
88H	Exception code

Restart communications option (sub-function code: 01H)

Operation description

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.

- Data being received
- 'CH1/2 side LED lighting status and communication error status' of the buffer memory (Un\G513/Un\G514)^{*1}
- 'CH1/2 side exception code storage area' of the buffer memory (Un\G28674/Un\G28676)^{*1}
- Diagnostic counter (☞ Page 21 Checking communication conditions using a diagnostic counter)
- ERR LED and C ERR LED OFF^{*2}
- Communications event count (☞ Page 38 Get Communications Event Counter (FC: 0BH))

*1 Clears only the receiving channel side area.

*2 Clears the errors of the channel that has received the request message.

As the errors of other channels are not cleared, the LED will not turn OFF if an error has occurred on any other channel.

Clears the the following data when "Clear setting of the communications event log" is specified with "FF00H (clear)" in the request message.

- Communications event log (☞ Page 39 Get Communications Event Log (FC: 0CH))

Request message format (from master to slave)

Function code	Sub-function code	Data
08H	(0001H) (H) (L)	Clear setting of Communications event log { FF00H: Clear } { 0000H: Not clear } (H) (L)

Response message format (from slave to master)

■ When completed successfully

The slave returns the request message received from the master as-is.

■ When completed with an error

Function code	Data
88H	Exception code

Return diagnostic register (sub-function code: 02H)

Operation description

Reads out the detailed LED status of C24 to the master.

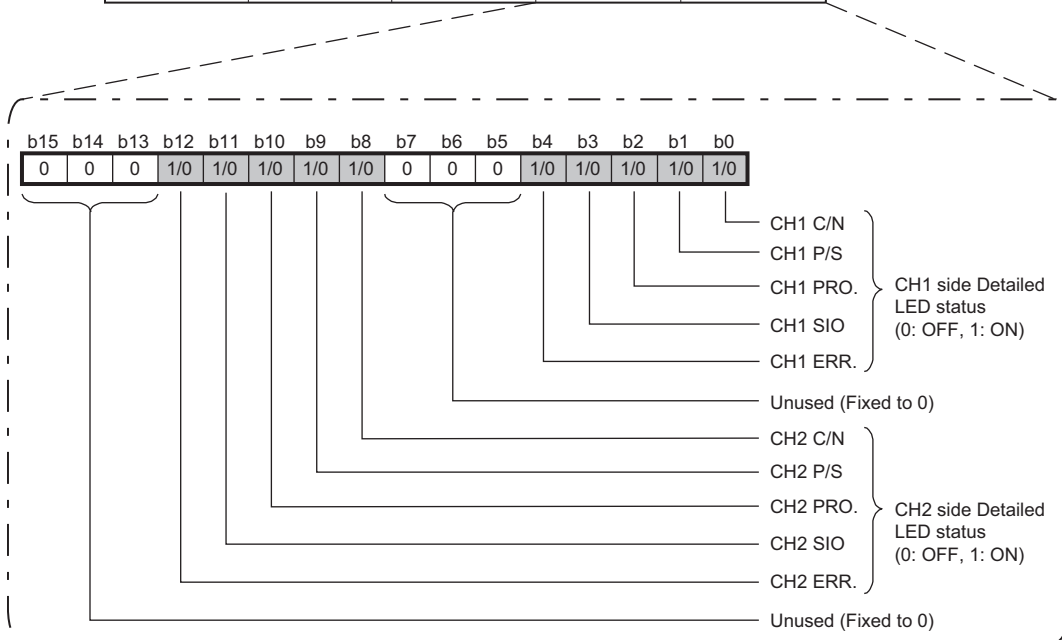
Request message format (from master to slave)

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

Response message format (from slave to master)

■ When completed successfully

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



In the C24, the detailed LED status of 'CH1/2 side LED lighting status and communication error status' of the buffer memory (Un\G513/Un\G514) is stored as the diagnostic register.

■ When completed with an error

Function code	Data
88H	Exception code

Change ASCII input delimiter (sub-function code: 03H)

Operation description

Changes the 2nd byte (LF(0AH)) of the end code in the ASCII mode to a specified data.

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 + LF (0DH) ; (0AH)

Change this into a specified data.

The specified data is stored in '2nd byte of end code' of the buffer memory (Un\G32524/Un\G32588).

Request message format (from master to slave)

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

Response message format (from slave to master)

■ When completed successfully

The slave returns the request message received from the master as-is.

■ When completed with an error

Function code	Data
88H	Exception code

Point

This function is used only for 1:1 connections.
Do not use this function for 1:n connections.




Force listen only mode (sub-function code: 04H)

Operation description

Places a slave into the offline mode.

Used when disconnecting a slave from the network.

When C24 is set in the listen only mode, the status is as follows:

- Ignores all request messages except for those of restart communications option. ( Page 23 Restart communications option (sub-function code: 01H))
- Stops counting of the diagnostic counter. ( Page 21 Checking communication conditions using a diagnostic counter)
- Continues recording with the communications event log. ( Page 39 Get Communications Event Log (FC: 0CH))

Request message format (from master to slave)

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

Response message format (from slave to master)

■ When completed successfully

No response message is returned because the listen only mode (offline status) is active.

■ When completed with an error


Function code	Data
88H	Exception code

Point

Whether the C24 has been switched to listen only mode or not can be checked in the 'Communications mode' of the buffer memory (Un\G32525/32589).

- 0000H: Online mode
- 0001H: Listen only mode

The listen only mode can be changed to online mode by either of the following:

- Restart communications option ( Page 23 Restart communications option (sub-function code: 01H))
- Turn the power OFF and ON, or reset the programmable controller CPU.



Clear counters and diagnostic register (sub-function code: 0AH)

Operation description

Clears counters (e.g. message count).

Also, clears the diagnostic register and the error of the channel where the request message has been received.

The following counters will be cleared.

- Diagnostic counter ( Page 21 Checking communication conditions using a diagnostic counter)
- Communications event count ( Page 38 Get Communications Event Counter (FC: 0BH))

The following diagnostic registers will be cleared.

- 'CH1/2 side LED lighting status and communication error status' of the buffer memory (Un\G513/Un\G514)*¹
- 'CH1/2 side exception code storage area' of the buffer memory (Un\G28674/Un\G28676)*¹

*1 Clears only the receiving channel side area.

Request message format (from master to slave)

Function code	Sub-function code	Data
08H	(000AH) (H) (L)	(0000H) (H) (L)

Response message format (from slave to master)

■ When completed successfully

The slave returns the request message received from the master as-is.

■ When completed with an error

Function code	Data
88H	Exception code

Return bus message count (sub-function code: 0BH)

Operation description

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

Request message format (from master to slave)

Function code	Sub-function code	Data
08H	(000BH) (H) (L)	(0000H) (H) (L)

Response message format (from slave to master)


■ When completed successfully

Function code	Sub-function code	Data
08H	(000BH) (H) (L)	Bus message count value (0000H to FFFFH) (H) (L)

Returns the count value of the buffer memory listed below to the master.

- 'Bus message count' (Un\G32512/Un\G32576)

Refer to the following for the counter descriptions, count range, and count clear methods.

 Page 21 Checking communication conditions using a diagnostic counter

■ When completed with an error

Function code	Data
88H	Exception code

Return bus communication error count (sub-function code: 0CH)

Operation description

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

Request message format (from master to slave)

Function code	Sub-function code	Data
08H	(000CH) (H) (L)	(0000H) (H) (L)

3

Response message format (from slave to master)

■ When completed successfully

Function code	Sub-function code	Data
08H	(000CH) (H) (L)	Bus communication error count value (0000H to FFFFH) (H) (L)

Returns the count value of the buffer memory listed below to the master.

- 'Bus communication error count' (Un\G32513/Un\G32577)

Refer to the following for the counter descriptions, count range, and count clear methods.

☞ Page 21 Checking communication conditions using a diagnostic counter

■ When completed with an error

Function code	Data
88H	Exception code

Return bus exception error count (sub-function code: 0DH)

Operation description

Reads out the frequency of exception errors to the master.

Request message format (from master to slave)

Function code	Sub-function code	Data
08H	(000DH) (H) (L)	(0000H) (H) (L)

Response message format (from slave to master)


■ When completed successfully

Function code	Sub-function code	Data
08H	(000DH) (H) (L)	Exception error count value (0000H to FFFFH) (H) (L)

Returns the count value of the buffer memory listed below to the master.

- 'Exception error count' (Un\G332522/Un\G32586)

Refer to the following for the counter descriptions, count range, and count clear methods.

 Page 21 Checking communication conditions using a diagnostic counter

■ When completed with an error

Function code	Data
88H	Exception code

Return slave message count (sub-function code: 0EH)

Operation description

Reads out the number of the slave message processing to the master (including receive of request messages from broadcast).

Request message format (from master to slave)

Function code	Sub-function code	Data
08H	(000EH) (H) (L)	(0000H) (H) (L)

Response message format (from slave to master)

■ When completed successfully

Function code	Sub-function code	Data
08H	(000EH) (H) (L)	Slave message count value (0000H to FFFFH) (H) (L)

Returns the count value of the buffer memory listed below to the master.

- 'Slave message count' (Un\G32518/Un\G32582)

Refer to the following for the counter descriptions, count range, and count clear methods.

☞ Page 21 Checking communication conditions using a diagnostic counter

■ When completed with an error

Function code	Data
88H	Exception code

Return slave no response count (sub-function code: 0FH)

Operation description

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

Request message format (from master to slave)

Function code	Sub-function code	Data
08H	(000FH) (H) (L)	(0000H) (H) (L)

Response message format (from slave to master)

■ When completed successfully

Function code	Sub-function code	Data
08H	(000FH) (H) (L)	Slave no-response count value (0000H to FFFFH) (H) (L)

Returns the count value of the buffer memory listed below to the master.

- 'Slave no-response count' (Un\G32519/Un\G32583)

Refer to the following for the counter descriptions, count range, and count clear methods.

☞ Page 21 Checking communication conditions using a diagnostic counter

■ When completed with an error

Function code	Data
88H	Exception code

Return slave NAK count (sub-function code: 10H)

Operation description

Reads out the number of NAK responses to the master.

Request message format (from master to slave)

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

Response message format (from slave to master)

■ When completed successfully

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

Returns the count value of the buffer memory listed below to the master.

- 'Slave NAK count' (Un\G32520/Un\G32584)

It is always '0' in C24s.

■ When completed with an error

Function code	Data
88H	Exception code

Return slave busy count (sub-function code: 11H)

Operation description

Reads out the number of busy responses to the master.

Request message format (from master to slave)

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

Response message format (from slave to master)

■ When completed successfully

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

Returns the count value of the buffer memory listed below to the master.

- 'Slave busy count' (Un\G32521/Un\G32585)

It is always '0' in C24s.

■ When completed with an error

Function code	Data
88H	Exception code

Return bus character overrun count (sub-function code: 12H)

Operation description

To the master, reads out the number of times the request message size exceeds the upper limit.

Request message format (from master to slave)

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

3

Response message format (from slave to master)

■ When completed successfully

Function code	Sub-function code	Data
08H	(0012H) (H) (L)	Bus character overrun count value (0000H to FFFFH) (H) (L)

Returns the count value of the buffer memory listed below to the master.

- 'Character overrun count' (Un\G32514/Un\G32578)

Refer to the following for the counter descriptions, count range, and count clear methods.

☞ Page 21 Checking communication conditions using a diagnostic counter

■ When completed with an error

Function code	Data
88H	Exception code

Return IOP overrun error count (sub-function code: 13H)

Operation description

Reads the IOP overrun error count value to the master.

C24 returns to the master the number of times the request message size exceeds the upper limit. (Same as the Return bus character overrun count)

Request message format (from master to slave)

Function code	Sub-function code	Data
08H	(0013H) (H) (L)	(0000H) (H) (L)

Response message format (from slave to master)

■ When completed successfully

Function code	Sub-function code	Data
08H	(0013H) (H) (L)	Bus character overrun count value (0000H to FFFFH) (H) (L)

Returns the count value of the buffer memory listed below to the master.

- 'Character overrun count' (Un\G32514/Un\G32578)

Refer to the following for the counter descriptions, count range, and count clear methods.

☞ Page 21 Checking communication conditions using a diagnostic counter

■ When completed with an error

Function code	Data
88H	Exception code

Clear overrun counter and flag (sub-function code: 14H)

Operation description

Clears the overrun error counter and flag.

C24 clears the character overrun count value.

Request message format (from master to slave)

Function code	Sub-function code	Data
08H	(0014H) (H) (L)	(0000H) (H) (L)

Response message format (from slave to master)

■ When completed successfully

The slave returns the request message received from the master as-is.

■ When completed with an error

Function code	Data
88H	Exception code

3.9 Get Communications Event Counter (FC: 0BH)

Operation description

Acquires the number of messages whose requested actions (read/write, diagnostics, etc.) have been normally completed. Whether the action corresponding to the request message is normally completed or not can be checked.

Request message format (from master to slave)

Function code
0BH

Response message format (from slave to master)

■ When completed successfully

Function code	Data	
0BH	Program command status	Communications event count value
	(0000H) (H) (L)	(0000H to FFFFH) (H) (L)

Since C24 does not support any program commands, 0000H is stored in the program command status.

Returns the count value of the buffer memory listed below to the master.

- 'Communications event count' (Un\G32523 to Un\G32587)

The count is stopped if it has reached FFFFH.

Reset the counter by any of the following methods when restarting the count.

- Restart communications option (☞ Page 23 Restart communications option (sub-function code: 01H))
- Clear counters and diagnostic register (☞ Page 27 Clear counters and diagnostic register (sub-function code: 0AH))
- Turn the power OFF and ON, or reset the programmable controller CPU.

■ When completed with an error

Function code	Data
8BH	Exception code

3.10 Get Communications Event Log (FC: 0CH)

Operation description

Acquires the communications event log into the master.

Request message format (from master to slave)

Function code

0CH

Response message format (from slave to master)

When completed successfully

Function code	Data						
-----	-----						
0CH	Number of read bytes	Program command status	Communications event count value	Bus message count value	Communications event log No. 0	• • •	Communications event log No. 63
		(0000H) (H) (L)	(0000H to FFFFH) (H) (L)	(0000H to FFFFH) (H) (L)			

Since C24 does not support any program commands, 0000H is stored in the program command status. Returns the count values of the buffer memory listed below to the master.

- 'Communications event count' (Un\G32523 to Un\G32587)
- 'Bus message count' (Un\G32512/Un\G32576)
- 'Communications event log 1 to 64' (Un\G32544 to Un\G32575/Un\G32608 to Un\G32639)

Refer to the following for the counter descriptions, count range, and count clear methods.

- ☞ Page 21 Checking communication conditions using a diagnostic counter
- ☞ Page 38 Get Communications Event Counter (FC: 0BH)

For details on the communications event log, refer to the following.

- ☞ Page 40 Communications event log

When completed with an error

Function code	Data
-----	-----
8CH	Exception code

Communications event log

When the get communications event log (FC: 0CH) is received from the master, C24 returns the following data of the buffer memory.

- 'Communications event log 1 to 64' (Un\G32544 to Un\G32575/Un\G32608 to Un\G32639)


The number of communications event logs can be checked in the buffer memory 'Communications event log count' (Un\G32543/Un\G32607).

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.

Communications event logs are stored in the buffer memory at the following timing.

Timing to be stored in the buffer memory	Description																
When receiving a request message	<p>C24 stores the communications event log before executing the processing of the request message. For the relevant communications event, '1' is stored.</p> <table border="1"> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>1</td><td>1/0</td><td>1/0</td><td>1/0</td><td>0</td><td>0</td><td>1/0</td><td>0</td> </tr> </table> <ul style="list-style-type: none"> • b0: Unused (Fixed to 0) • b1: Communication error • b2, b3: Unused (Fixed to 0) • b4: Bus character overrun error • b5: In listen only mode • b6: Broadcast message reception • b7: Fixed to 1 	b7	b6	b5	b4	b3	b2	b1	b0	1	1/0	1/0	1/0	0	0	1/0	0
b7	b6	b5	b4	b3	b2	b1	b0										
1	1/0	1/0	1/0	0	0	1/0	0										
When sending a response message	<p>C24 stores the communications event log after sending the response message. For the relevant communications event, '1' is stored.</p> <table border="1"> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1/0</td><td>1/0</td> </tr> </table> <ul style="list-style-type: none"> • b0: Message error (exception code 01H to 03H) • b1: Processing interruption (exception code 04H) • b2 to b5: Unused (Fixed to 0) <p>While the occurrence of busy status (exception code 05H to 07H) is stored for the MODBUS protocol, '0' is stored for C24 because this kind of events does not occur in it.</p> <ul style="list-style-type: none"> • b6: Fixed to 1 • b7: Unused (Fixed to '0') 	b7	b6	b5	b4	b3	b2	b1	b0	0	1	0	0	0	0	1/0	1/0
b7	b6	b5	b4	b3	b2	b1	b0										
0	1	0	0	0	0	1/0	1/0										
When switching to the listen only mode	<p>C24 stores the communications event log when switching to the listen only mode. 04H is stored to the communications event log.</p>																
When processing restart communications option	<p>C24 stores the communications event log when processing the restart communications option. 00H is stored to the communications event log.</p>																

The communications event can be cleared by either of the following:

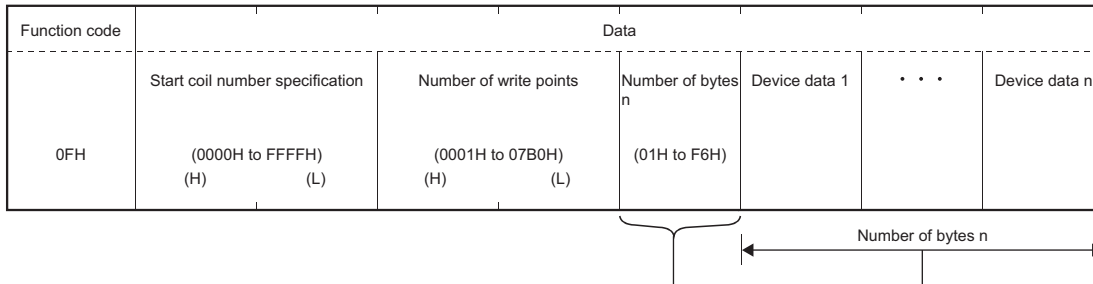
- Clear setting of the communications event log with the restart communications option. ( Page 23 Restart communications option (sub-function code: 01H))
- Turn the power OFF and ON, or reset the programmable controller CPU.

3.11 Write Multiple Coils (FC: 0FH)

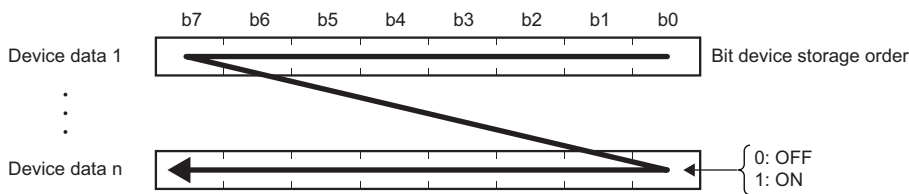
Operation description

Writes values (ON/OFF) to multiple coils.

Request message format (from master to slave)



- Device data 1 to n



- The values (ON/OFF) stored in device data 1 to n are written to the coils in the order of lower bit to higher bit of the device data.

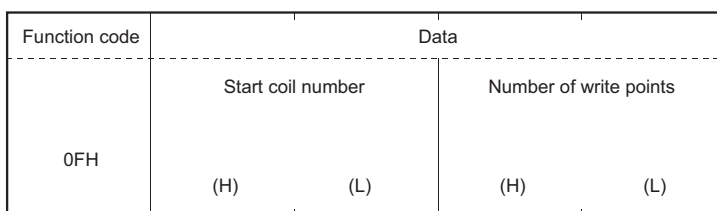
Point

Ensure that the specified number of write points matches the number of bits specified with the number of bytes.

For example, when the specified number of write points is 16, set the number of bytes to 2 bytes (= 16 bits).

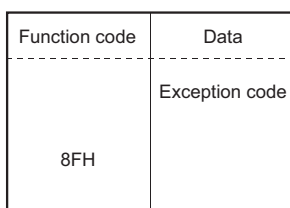
Response message format (from slave to master)

When completed successfully



- Start coil number: Stores the same value as the start coil number of the request message.
- Number of write points: Stores the same value as the number of write points of the request message.

When completed with an error

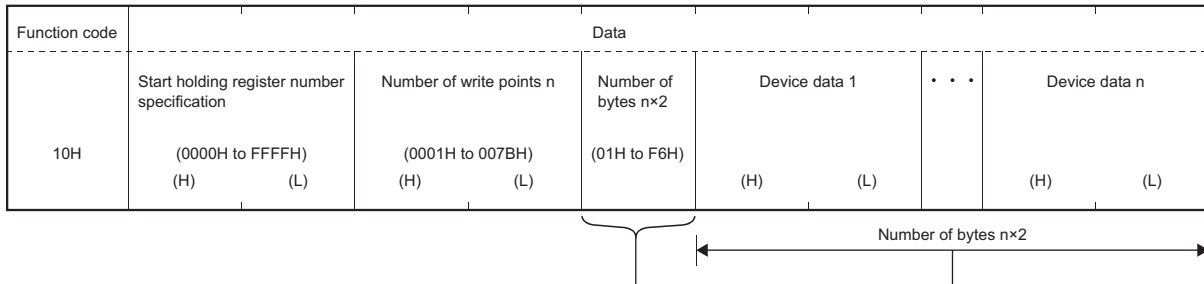


3.12 Write Multiple Registers (FC: 10H)

Operation description

Writes values to multiple holding register areas.

Request message format (from master to slave)

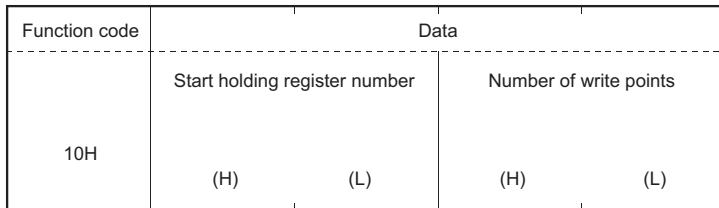


Point

Ensure that the specified number of write points matches the number of bits specified with the number of bytes.

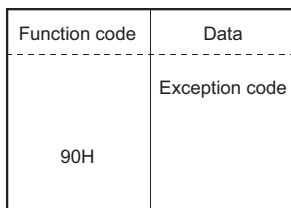
Response message format (from slave to master)

When completed successfully



- Start holding register number: Stores the same value as the start holding register number of the request message.
- Number of write points: Stores the same value as the number of write points of the request message.

When completed with an error



3.13 Report Slave ID (FC: 11H)

Operation description

Acquires the information of C24 mounted station into the master.

Request message format (from master to slave)


Function code

11H

Response message format (from slave to master)

■ When completed successfully

Function code	Data		
-----	-----		
11H	Number of bytes (02H)	Programmable controller CPU type	Programmable controller CPU STOP/RUN state

- Programmable controller CPU type:  Page 43 Programmable controller CPU type
- Programmable controller CPU STOP/RUN: STOP (00H), RUN (FFH)

■ When completed with an error

Function code	Data
-----	-----
91H	Exception code

Programmable controller CPU type

C24 will return any of the following programmable controller CPU type data.

The programmable controller CPU type is the original for C24s.

Module name	Model	Programmable controller CPU type returned to Master
Programmable controller CPU	R00CPU	A0H
	R01CPU	A1H
	R02CPU	A2H
	R04CPU	00H
	R08CPU	01H
	R16CPU	02H
	R32CPU	03H
	R120CPU	04H
	R04ENCPU	05H
	R08ENCPU	06H
	R16ENCPU	07H
	R32ENCPU	08H
	R120ENCPU	09H

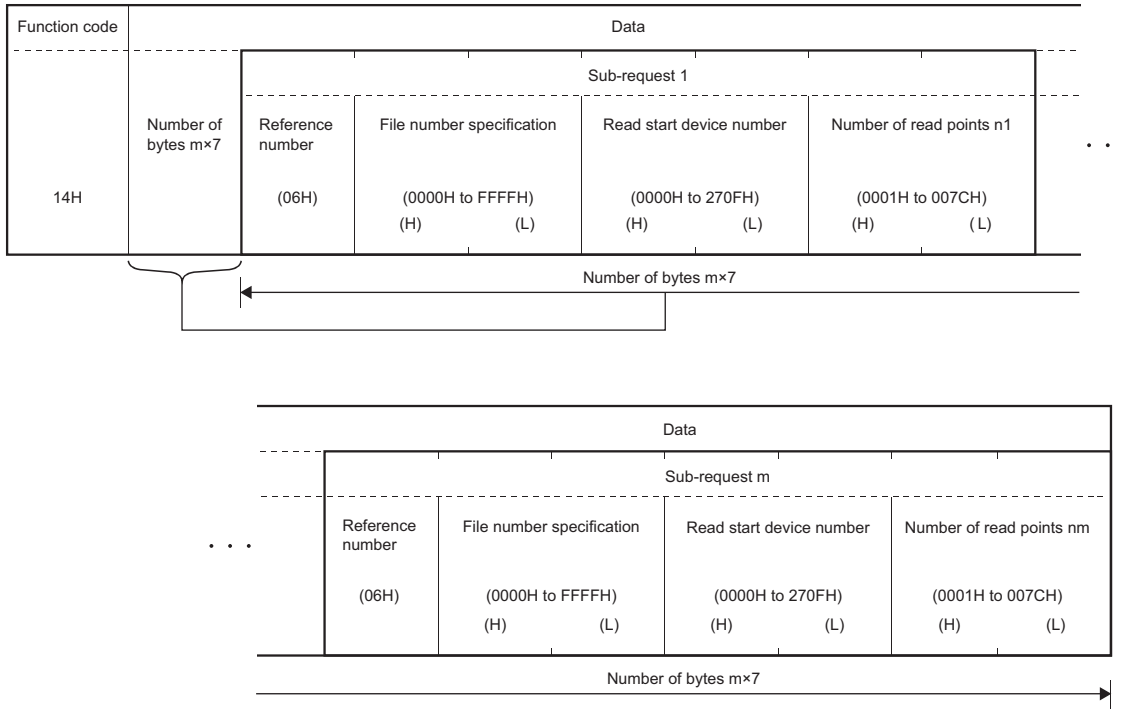
Module name	Model	Programmable controller CPU type returned to Master
Process CPU	R08PCPU	41H
	R16PCPU	42H
	R32PCPU	43H
	R120PCPU	44H
SIL2 process CPU	R08PSFCPU	51H
	R16PSFCPU	52H
	R32PSFCPU	53H
	R120PSFCPU	54H
Safety CPU	R08SFCPU	91H
	R16SFCPU	92H
	R32SFCPU	93H
	R120SFCPU	94H
C Controller module	R12CCPU-V	20H
Remote head module	RJ72GF15-T2	60H
	RJ72GF15-T2(SR)	61H
	RJ72GF15-T2(LR)	62H

3.14 Read File Record (FC: 14H) (SC: 06H)

Operation description

Reads the values of multiple extended file register areas.

Request message format (from master to slave)



- File number specification: The upper limit of the file numbers that can be received by an Ethernet-equipped module is the file register size of the mounted CPU module.

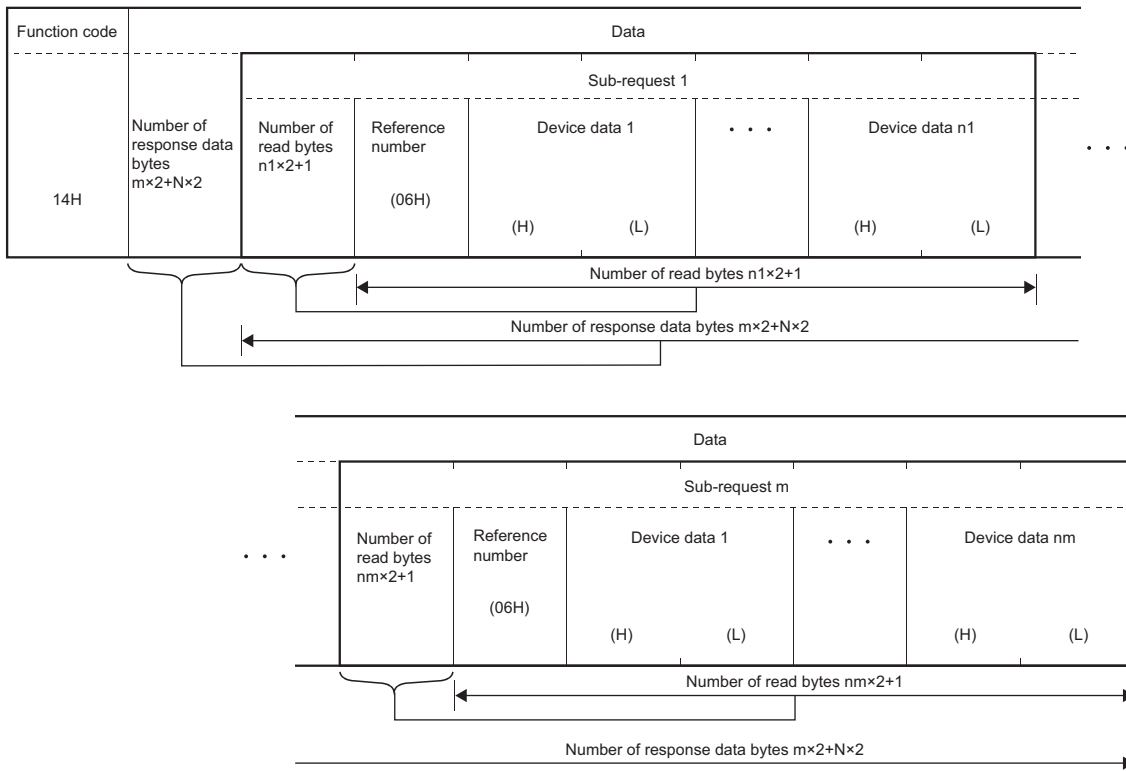
Point

- Specify the number of sub requests, m , so that the size of the protocol data unit of the request message does not exceed 253 bytes. The request message will be discarded if the following condition is not met.
 [Conditional formula] $253 \geq 2 + (m \times 7)$
- Specify the total number of read points of each sub request, N ($n_1 + \dots + n_m$), so that the size of the protocol data unit of the response message does not exceed 253 bytes. The slave will return an abnormal response if the following condition is not met.
 [Conditional formula] $253 \geq 2 + (m \times 2) + (N \times 2)$

Response message format (from slave to master)

■ When completed successfully

N shown below is the total of the device data ($n_1 + \dots + n_m$).



■ When completed with an error

Function code	Data
94H	Exception code

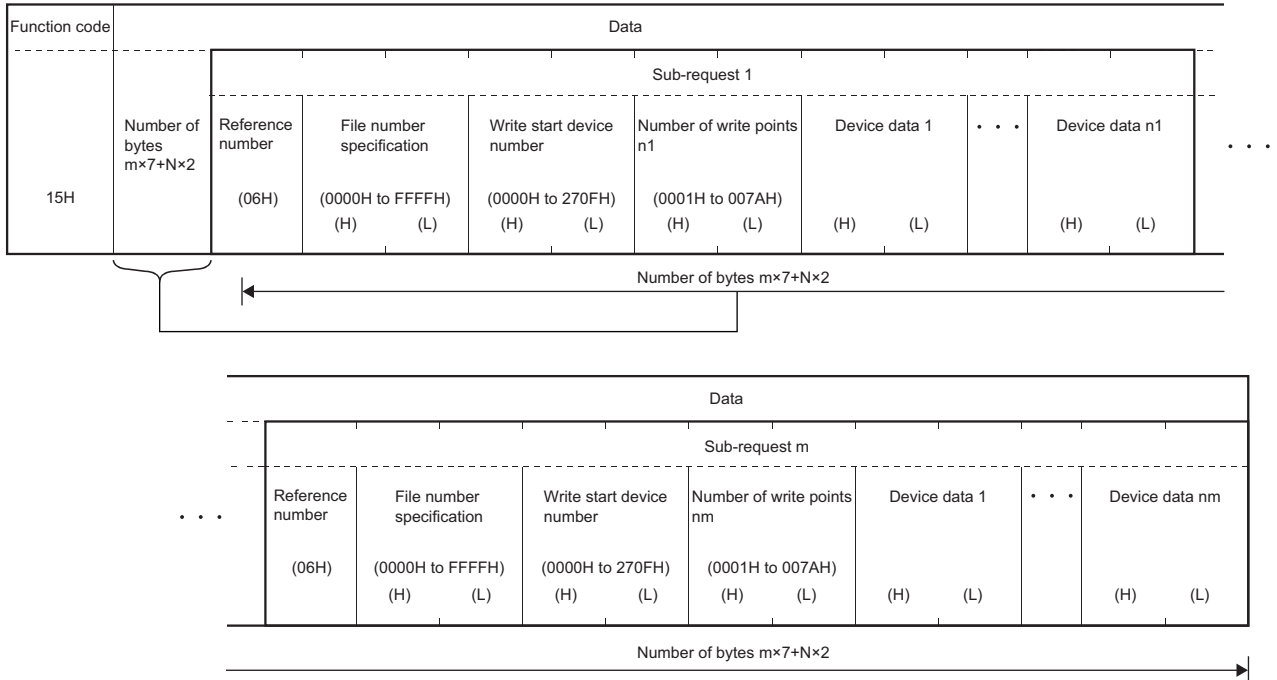
3.15 Write File Record (FC: 15H) (SC: 06H)

Operation description

Writes a value to one extended file register area or values to multiple extended file register areas.

Request message format (from master to slave)

N shown below is the total of the device data (n1 + ... + nm).



- File number specification: The upper limit of the file numbers that can be received by an Ethernet-equipped module is the file register size of the mounted CPU module.

Point

Specify the number of sub requests, m, and the total number of write points of each sub request, N ($n1 + \dots + nm$), so that the size of the protocol data unit of the request message does not exceed 253 bytes. The request message will be discarded if the following condition is not met.

$$[Conditional formula] 253 \geq 2 + (m \times 7) + (N \times 2)$$

Response message format (from slave to master)

■ When completed successfully

The slave returns the request message received from the master as-is.

■ When completed with an error

Function code	Data
95H	Exception code

3.16 Mask Write Register (FC: 16H)

Operation description

Writes the masked value obtained by performing AND and OR operations on the value stored in one holding register area. The value is written to the holding register as shown below.

Write value = (current value of target register \wedge AND mask value) \vee (OR mask value \wedge $\overline{\text{AND mask value}}$)

If the OR mask value is 0000H, only the AND processing of the AND mask value will be performed.

If the AND mask value is 0000H, the OR mask value will be the write value.

Request message format (from master to slave)

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

Response message format (from slave to master)

■ When completed successfully

The slave returns the request message received from the master as-is.

■ When completed with an error

Function code	Data
96H	Exception code

Point

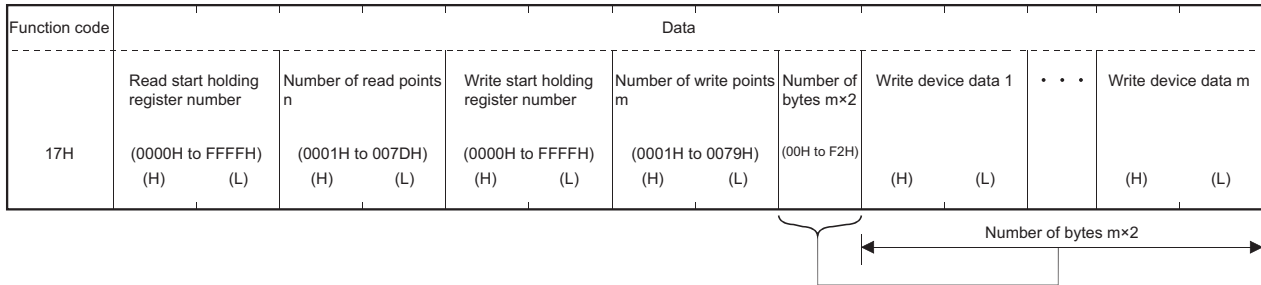
With this function, the value stored in the holding register is read from the slave, the AND/OR processing is performed on the master, and then the mask value is written to the holding register on the slave. Therefore, if the holding register value is changed during AND/OR processing, the changed value will be overwritten.

3.17 Read/Write Multiple Registers (FC: 17H)

Operation description

Reads/writes data from/to multiple holding register areas.
 During the processing, writing is performed first, followed by reading.

Request message format (from master to slave)

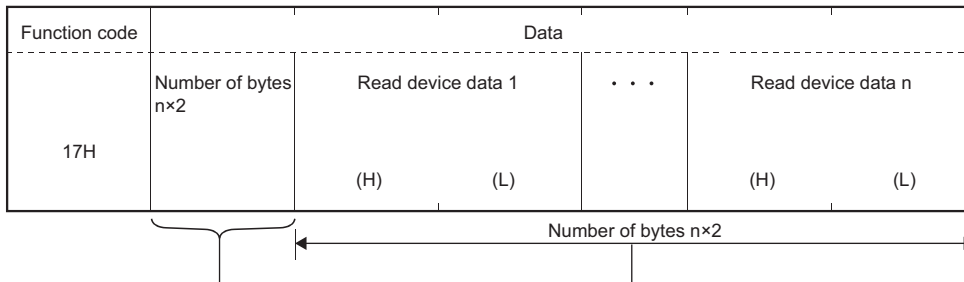


Point

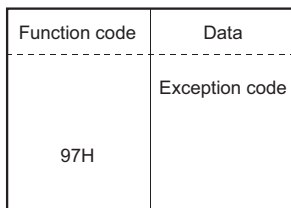
Ensure that the specified number of write points matches the number of bits specified with the number of bytes.

Response message format (from slave to master)

■ When completed successfully



■ When completed with an error

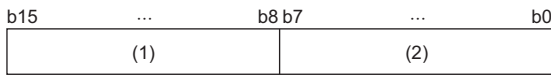


4 EXCEPTION CODES

Exception codes are error codes that are common to the MODBUS protocol and are stored in the response message when a slave returns an abnormal response for the request message from the master.

When the master receives an exception code from a slave, perform actions according to the description in this chapter.

The response message format when processing completed with an error is shown below.




(1) Abnormal response function code

(2) Exception code

Abnormal response function code list

This section lists the abnormal response function codes that are stored in the upper byte of the response message format.

Abnormal response function code	Function name	Details
81H	Read coils	'Read coils' completed with an error.
82H	Read discrete inputs	'Read discrete inputs' completed with an error.
83H	Read holding registers	'Read holding registers' completed with an error.
84H	Read input registers	'Read input registers' completed with an error.
85H	Write single coil	'Write single coil' completed with an error.
86H	Write single register	'Write single holding register' completed with an error.
87H	Read exception status	'Read exception status' completed with an error.
88H	 Page 13 Diagnosis function details	'Diagnostics' completed with an error.
8BH	Get communications event counter	'Get communications event counter' completed with an error.
8CH	Get communications event log	'Get communications event log' completed with an error.
8FH	Write multiple coils	'Write multiple coils' completed with an error.
90H	Write multiple registers	'Write multiple holding registers' completed with an error.
91H	Report slave ID	'Report slave ID' completed with an error.
94H	Read file record	'Read file record' completed with an error.
95H	Write file record	'Write file record' completed with an error.
96H	Mask write register	'Mask write register' completed with an error.
97H	Read/Write multiple registers	'Read/Write multiple registers' completed with an error.

Exception code list

This section lists the exception codes that are stored in the lower byte of the response message format.

Exception code	Error name	Error definition and cause	Action
01H	Function code error	The slave received an unsupported function code.	Check the function codes that are supported by the slave, and then check the request message sent from the master.
02H	Device address error	The specified MODBUS device address is incorrect.	Check the types and sizes of the MODBUS devices that are supported by the slave, and then check the address specification of the request message sent from the master.
03H	Data error	The content of the protocol data unit of the request message is abnormal.	Revise the data unit of the request message sent from the master.
04H	Processing failure	Processing was not possible because of an error that occurred during request message processing on the slave.	Eliminate the error factor that has occurred on the slave. If a module is issuing this code, use the module diagnostics or a similar method to identify the problem from the error that is occurring, and then perform the appropriate actions.
06H	Slave busy	The slave cannot execute the processing of the request message because it is executing other processing.	Retry at a later time. If a module is issuing this code, revise the settings so that the number of request messages that can be received at the same time is 64 or less.

INDEX

A

Abnormal response function code list 50

D

Descriptions of request message and response
message formats 11

E

Exception code list 51

M

Master 5

S

Slave 5
Storage locations of exception code and error code 12

MEMO

REVISIONS

*The manual number is given on the bottom left of the back cover.

Print date	*Manual number	Revision
October 2018	BCN-P5999-1060-A	First edition
October 2020	BCN-P5999-1060-B	■Added models RJ71C24, RJ71C24-R2, RJ71C24-R4 ■Added functions MODBUS slave function ■Added or modified parts CONDITIONS OF USE FOR THE PRODUCT, INTRODUCTION, RELEVANT MANUALS, TERMS, GENERIC TERMS AND ABBREVIATIONS, Chapter 1, Section 1.1, Section 1.2, Section 1.3, Chapter 2, Chapter 3, Section 3.7, Section 3.8, Section 3.9, Section 3.10, Section 3.13, Chapter 4

Japanese manual number: BCN-P5999-1059-B

This manual confers no industrial property rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric cannot be held responsible for any problems involving industrial property rights that may occur as a result of using the contents noted in this manual.

© 2018 MITSUBISHI ELECTRIC CORPORATION

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.
 1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
 2. Failure caused by unapproved modifications, etc., to the product by the user.
 3. 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.
 4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
 5. 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.
 6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
 7. Any other 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 to:

- (1) Damages caused by any cause found not to be the responsibility of Mitsubishi.
- (2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.
- (3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.
- (4) 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.

TRADEMARKS

The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as [™] or [®] are not specified in this manual.

BCN-P5999-1060-B(2010)

mitsubishi electric corporation

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA, JAPAN

When exported from Japan, this manual does not require application to the
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.