SYSMAC
CS/CJ Serie

CS1W-ETN21
CJ1W-ETN21

MODBUS TCP

Quick Start Manual

Warning

This documentation is intended to facilitate the implementation of the material omron. Certain details are voluntarily occulted not to cause confusion. Despite everything the care taken to the realization of this documentation, omron could not be held for person in charge for the errors or omissions and their consequences. This documentation could be modified without notice and does not present any engagement on behalf of omron.

Table of content

1. Specifications

1-1 Supported command list

Code (Hex)	Function	Name in MODBUS
0x01	I/O memory area (CIO) Read Multiple Coils	Read Coils
0x02	I/O memory area (CIO) Read Multiple Coils	Read Discrete Inputs
0x03	I/O memory area (DM) Read Multiple Registers	Read Holding Registers
0x04	I/O memory area (CIO) Read Multiple Registers	Read Input Registers
0x05	I/O memory area Write Single Coil	Write Single Coil
0x06	I/O memory area (DM) Write Single Register	Write Single Register
0x08	Echo back test	Diagnostic
0x0F	***** NOT SUPPORTED ******	Write Multiple Coils
0x10	I/O memory area (DM) Write Multiple Registers	Write Multiple Registers

1-2 Memory map

1-2-0 Used by ModbusTCP PLC program

Modbus process

Type	Memory address	Descriptions
Work Area	W480 -511	Used for counter and calculation
Receive Area	CIO 5800 - 6000	Used to store received bytes
Send Area	CIO 6001 - 6143	Used to prepare bytes to send

Ethernet Unit flag & command switch (Unit n°0)

Type	Memory address	Descriptions
Flag/command Area	CIO1000- CIO1024	For more details see Op. manual W343
Parameter Area	D 30000 - D30099	

1-2-1 PLC area accessible by Modbus TCP request

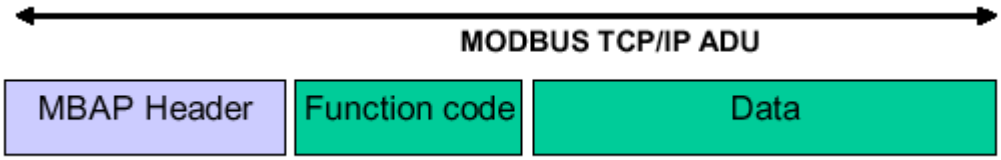
	MODBUS Address	PDU Address	Corresponding CS/CJ's address
Discrete Inputs	1 - 5120	0 - 5119	0 – 5119 (CIO 0.00 - CIO319.15)
Coils	1 - 65536	0 - 65535	0 – 65535 (CIO 0.00 – CIO4095.15)
Input Registers	1 - 5801	0 - 57800	0 – 5800 (CIO0 – CIO5800) *
Holding Registers	1 - 32768	0 - 32767	0 - 32767 (D0 – D32767)

*: area 5801 to 6143 is reserved for the ModbusTCP PLC program (see 1-2-0)

1-3 Modbus TCP frame Format

A dedicated header is used on TCP/IP to identify the MODBUS Application Data Unit. It is called the MBAP header (MODBUS Application Protocol header).
This header provides some differences compared to the MODBUS RTU application data unit used on serial line:

- The MODBUS ‘ slave address’ field usually used on MODBUS Serial Line is replaced by a single byte ‘ Unit Identifier’ within the MBAP Header. The ‘ Unit Identifier’ is used to communicate via devices such as bridges, routers and gateways that use a single IP address to support multiple independent MODBUS end units.
- All MODBUS requests and responses are designed in such a way that the recipient can verify that a message is finished. For function codes where the MODBUS PDU has a fixed length, the function code alone is sufficient. For function codes carrying a variable amount of data in the request or response, the data field includes a byte count.
- When MODBUS is carried over TCP, additional length information is carried in the MBAP header to allow the recipient to recognize message boundaries even if the message has been split into multiple packets for transmission. The existence of explicit and implicit length rules, and use of a CRC-32 error check code (on Ethernet) results in an infinitesimal chance of undetected corruption to a request or response message.



1-3-0 MBAP Header description

The MBAP Header contains the following fields:

Fields	Length	Description	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction	Initialized by the client (request)	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = MODBUS protocol	Initialized by the client (request)	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client (request)	Initialized by the server (Response)
Unit Identifier	1 Byte	Identification of a remote slave connected on a serial line or on other buses	Initialized by the client (request)	Recopied by the server from the received request

The header is 7 bytes long:
• **Transaction Identifier** - It is used for transaction pairing, the MODBUS server copies in the response the transaction identifier of the request.
• **Protocol Identifier** – It is used for intra-system multiplexing. The MODBUS protocol is identified by the value 0.
• **Length** - The length field is a byte count of the following fields, including the Unit Identifier and data fields.
• **Unit Identifier** – This field is used for intra-system routing purpose. It is typically used to communicate to a MODBUS or a MODBUS+ serial line slave through a gateway between an Ethernet TCP-IP network and a MODBUS serial line. This field is set by the MODBUS Client in the request and must be returned with the same value in the response by the server.

All Modbus/TCP ADU are sent via TCP on registered port 502.

1-4 Function Code

1-4-0 I/O memory area (CIO) Read Multiple Coils

[Function]

Reads coils in I/O memory area.

[Request]

	Length	Data
Function Code	1 Byte	**0x01**
Starting Address	2 Bytes	0x0000-0xFFFF
Quantity of Coils	2 Bytes	1-2000(0x7D0)

[Response]

	Length	Data
Function Code	1 Byte	**0x01**
Byte Count	1 Byte	N
Coil Status	n Byte	n = N or N+1

Example: read 19 bits (CIO 0001.04 to 0002.06)

Request			Response	
	Data			Data
Function Code	0x01		Function Code	0x01
Starting Address(H)	0x00		Byte Count	0x03
Starting Address(L)	**0x14**		Coil Status 27-20	0xCD
Quantity of Coils(H)	0x00		Coil Status 35-28	0x6B
Quantity of Coils(L)	0x13		Coil Status 38-36	0x05

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0CH	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1CH	31 *1*	30 *0*	29 *1*	28 *1*	27 *1*	26 *1*	25 *0*	24 *0*	23 *1*	22 *1*	21 *0*	20 *1*	19	18	17	16
2CH	47	46	45	44	43	42	41	40	39	38 *1*	37 *0*	36 *1*	35 *0*	34 *1*	33 *1*	32 *0*
3CH	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

Italic characters show the ON/OFF(1/0) status of its bit condition.

1-4-1 I/O memory area (CIO) Read Multiple Coils

[Function]
Reads coils in I/O memory area

[Request]

	Length	Data
Function Code	1 Byte	**0x02**
Starting Address	2 Bytes	0x0000-0x13FF
Quantity of Coils	2 Bytes	1-2000(0x7D0)

[Response]

	Length	Data
Function Code	1 Byte	**0x02**
Byte Count	1 Byte	N
Coil Status	n Byte	n = N or N+1

Example: read 19 bits (CIO 0001.04 to 0002.06)

Request			Response	
	Data			Data
Function Code	0x02		Function Code	0x02
Starting Address(H)	0x00		Byte Count	0x03
Starting Address(L)	0x13		Coil Status 27-20	0xCD
Quantity of Coils(H)	0x00		Coil Status 35-28	0x6B
Quantity of Coils(L)	0x13		Coil Status 38-36	0x05

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0CH	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1CH	31 *1*	30 *0*	29 *1*	28 *1*	27 *1*	26 *1*	25 *0*	24 *0*	23 *1*	22 *1*	21 *0*	20 *1*	19	18	17	16
2CH	47	46	45	44	43	42	41	40	39	38 *1*	37 *0*	36 *1*	35 *0*	34 *1*	33 *1*	32 *0*
3CH	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

Italic characters show the ON/OFF(1/0) status of its bit condition.

1-4-2 I/O memory area (DM) Read Multiple Registers

[Function]
Reads registers in I/O memory area

[Request]

	Length	Data
Function Code	1 Byte	**0x03**
Starting Address	2 Bytes	0x0000-0x7FFF(*)
Quantity of Registers	2 Bytes	1-125(0x7D)

(*)Useful range of Start Address depends on allocated area.

[Response]

	Length	Data
Function Code	1 Byte	**0x03**
Byte Count	1 Byte	N x 2(*)
Register Value	N x 2 Bytes	

(*)N=Quantity of Registers

Example: read 3 words (DM 1000 to DM 1002)

Request			Response	
	Data			Data
Function Code	0x03		Function Code	0x03
Starting Address(H)	0x03		Byte Count	0x06
Starting Address(L)	0xE8		Register Value(H)DM1000	0xAB
Quantity of Registers(H)	0x00		Register Value(L) DM1000	0x12
Quantity of Registers(L)	0x03		Register Value(H)DM1001	0x56
			Register Value(L) DM1001	0x78
			Register Value(H)DM1002	0x97
			Register Value(L) DM1002	0x13

DM	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
1000	A	B	1	2
1001	5	6	7	8
1002	9	7	1	3

1-4-3 I/O memory area (CIO) Read Multiple Registers

[Function]

Reads registers in I/O memory area

[Request]

	Length	Data
Function Code	1 Byte	**0x04**
Starting Address	2 Bytes	0x0000-0x16A8
Quantity of Registers	2 Bytes	1-125(0x7D)

[Response]

	Length	Data
Function Code	1 Byte	**0x04**
Byte Count	1 Byte	N x 2(*)
Register Value	N x 2 Bytes	

(*)N=Quantity of Registers

Example: read 3 words (CIO 1000 to CIO 1002)

Request			Response	
	Data			Data
Function Code	0x04		Function Code	0x04
Starting Address(H)	0x03		Byte Count	0x06
Starting Address(L)	0xE8		Register Value(H)CIO1000	0xAB
Quantity of Registers(H)	0x00		Register Value(L)CIO1000	0x12
Quantity of Registers(L)	0x03		Register Value(H)CIO1001	0x56
			Register Value(L)CIO1001	0x78
			Register Value(H)CIO1002	0x97
			Register Value(L)CIO1002	0x13

DM	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1000	A				B				1				2			
1001	5				6				7				8			
1002	9				7				1				3			

1-4-4 I/O memory area Write Single Coil

[Function]
Writes single coil.

[Request]

	Length	Data
Function Code	1 Byte	**0x05**
Output Address	2 Bytes	0x0000-0xFFFF
Output Value	2 Bytes	0x0000(OFF) or 0xFF00(ON)

[Response]

	Length	Data
Function Code	1 Byte	**0x05**
Output Address	2 Bytes	0x0000-0xFFFF
Output Value	2 Bytes	0x0000 or 0xFF00

Example: write 1 coil. (CIO 0002.02 ON)

Request			Response	
	Data			Data
Function Code	0x05		Function Code	0x05
Output Address(H)	0x00		Output Address(H)	0x00
Output Address(L)	0x22		Output Address(L)	0x22
Output Value(H)	0xFF		Output Value(H)	0xFF
Output Value(L)	0x00		Output Value(L)	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0CH	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1CH	31 *1*	30 *0*	29 *1*	28 *1*	27 *1*	26 *1*	25 *0*	24 *0*	23 *1*	22 *1*	21 *0*	20 *1*	19	18	17	16
2CH	47	46	45	44	43	42	41	40	39	38 *1*	37 *0*	36 *1*	35 *0*	34 *1*	33 *1*	32 *0*
3CH	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

Italic characters show the ON/OFF(1/0) status of its bit condition.

1-4-5 I/O memory area (DM) Write Single Register

[Function]
Writes single register.

[Request]

	Length	Data
Function Code	1 Byte	**0x06**
Register Address	2 Bytes	0x0000-0x7FFF
Register Value	2 Bytes	0x0000-0xFFFF

[Response]

	Length	Data
Function Code	1 Byte	**0x06**
Register Address	2 Bytes	0x0000-0x7FFF
Register Value	2 Bytes	0x0000-0xFFFF

Example: write &h3AC5 to DM 2000.

Request			Response	
	Data			Data
Function Code	0x06		Function Code	0x06
Register Address(H)	0x07		Register Address(H)	0x07
Register Address(L)	0xD0		Register Address(L)	0xD0
Register Value(H)	0x3A		Register Value(H)	0x3A
Register Value(L)	0xC5		Register Value(L)	0xC5

DM	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
2000	3	A	C	5
2001				
2002				

1-4-6 Echo back test

[Function]
Executes echo back test. Sending data is returned.

[Request]

	Length	Data
Function Code	1 Byte	**0x08**
Sub-function Code	2 Bytes	0x0000
Data	N x 2 Bytes (*)	---

(*)2 to 125.(0x0002 to 0x007D) Bytes

[Response]

	Length	Data
Function Code	1 Byte	**0x08**
Sub-function Code	2 Bytes	0x0000
Data	N x 2 Bytes (*)	---

(*) Same as request data.

Example: sending 0xA537.

Request			Response	
	Data			Data
Function Code	0x06		Function Code	0x06
Sub-function Code(H)	0x00		Sub-function Code(H)	0x00
Sub-function Code(L)	0x00		Sub-function Code(L)	0x00
Data(H)	0xA5		Data(H)	0xA5
Data(L)	0x37		Data(L)	0x37

1-4-7 I/O memory area (DM) Write Multiple Registers

[Function]
Writes registers.

[Request]

	Length	Data
Function Code	1 Byte	**0x10**
Starting Address	2 Bytes	0x0000-0x17FF
Quantity of Registers	2 Bytes	1-123(0x7B)
Byte Count	1 Byte	2 x N(*)
Registers Value	N x 2 Bytes	value

(*)N = Quantity of Registers to write.

[Response]

	Length	Data
Function Code	1 Byte	**0x10**
Starting Address	2 Bytes	0x0000-0x17FF
Quantity of Registers	2 Bytes	1-123(0x7B)

Example: write 2 words into DM1000-1001.

Request			Response	
	Data			Data
Function Code	0x10		Function Code	0x10
Starting Address(H)	0x03		Starting Address(H)	0x03
Starting Address(L)	0xE8		Starting Address(L)	0xE8
Quantity of Registers(H)	0x00		Quantity of Registers(H)	0x00
Quantity of Registers(L)	0x02		Quantity of Registers(L)	0x02
Byte Count	0x04			
Registers Value(H)	0x3A			
Registers Value(L)	0xC5			
Registers Value(H)	0x97			
Registers Value(L)	0x13			

DM	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1000	3				A				C				5			
1001	9				7				1				3			

1-5 Error response

If the request contains error like address or quantity out of range, the PLC program will generate an exception response containing the exception code

[Response]

	Length	Data
Function Code	1 Byte	Function Code + 0x80
Exception Code	1 Byte	01, 02 or 03

Exception Code	Name
01	ILLEGAL FUNCTION
02	ILLEGAL DATA ADDRESS
03	ILLEGAL DATA VALUE

1-6 Status counter

Counter	Channel	Descriptions
Exception_Counter	W491	Keep a record of illegal request
RCV_Counter	W492	Keep a record of response sent
SND_Counter	W493	Keep a record of frame received
ER_RCV_Counter	W494	Keep a record of socket receive error
ER_SND_Counter	W495	Keep a record of socket send error