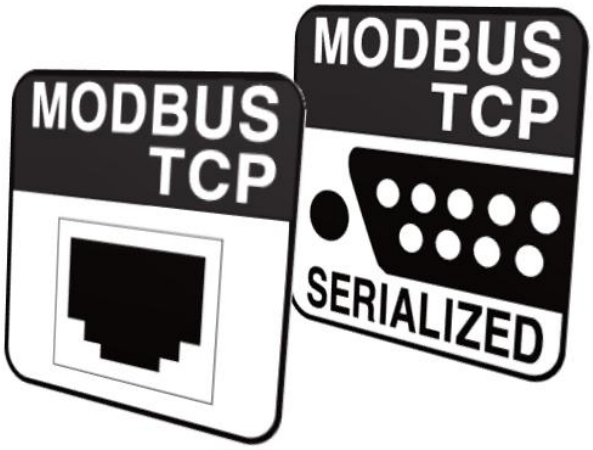
ezTCP Technical Document

Modbus/TCP of ezTCP

Version 1.3



☞ ***Caution: Specifications of this document may be changed without prior notice for improvement.***

Sollae Systems Co., Ltd.

http://www.sollae.co.kr

Contents

1 Overview

Modbus is a serial communication protocol widely used in the world for devices such as Programmable Logic Controllers (PLCs) and etc. It is one of the Modbus' versions which perform on TCP/IP network. I/O controllers of our products named ezTCP have been using of this protocol.

To use the Modbus/TCP, devices have to use its Ethernet port. Sometimes, users need to control or monitor their devices through a serial port (RS232). Serialized Modbus/TCP mode was developed to meet these demands.

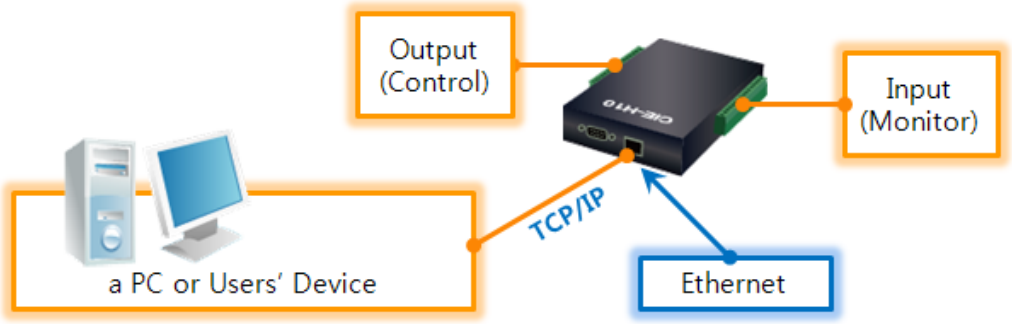


Figure 1-1 a diagram of Modbus/TCP system

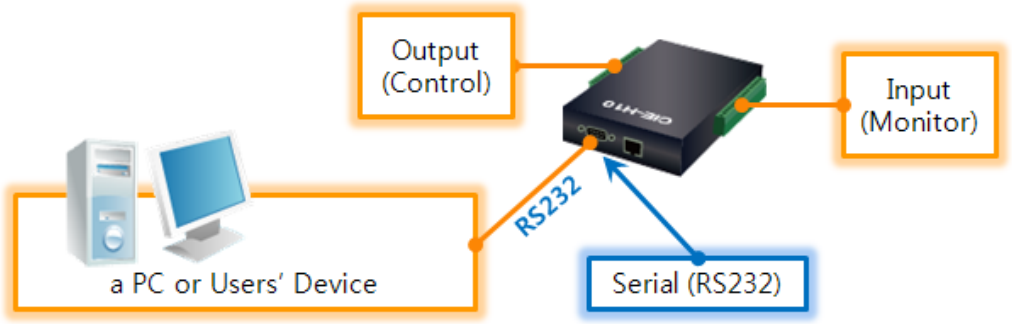


Figure 1-2 a diagram of Serialized Modbus/TCP system

In the serialized Modbus/TCP mode, ezTCP sends and receives the Modbus/TCP data through the RS232 port.

2 Overview of Modbus/TCP

2.1 Features

- TCP/IP version of Modbus/TCP protocol
- Connection process
  In the standard, a slave operates as a TCP server only. However, ezTCP can be performed both TCP server and client. Port number should be TCP 502.
- Master and Slave
  As well as a slave, the ezTCP can be performed as a Modbus/TCP Master by configuration.
- Big-endian
  In the Big-endian system, the Most Significant Byte (MSB) has the lowest address. For example, 0x1234 might be placed with order of 0x12 and 0x34.

☞ ***MSB: the byte in a multiple-byte word with the largest value.***

☞ ***LSB: the byte in a multiple-byte word with the smallest value.***

- Only Class 0 functions are supported.

Table 2-1 functions of ezTCP's Modbus/TCP

Class	Function Code	Name	Description
0	0x03	read multiple registers	read I/O ports with WORD unit
0	0x10	write multiple registers	write output ports with WORD unit
1	0x01	read coils	read output ports with BIT unit
1	0x02	read input discretes	read input ports with BIT unit
1	0x04	read input registers	read I/O ports with WORD unit
1	0x05	write coil	write output ports with BIT unit
1	0x06	write single register	write output ports with WORD unit
1	0x07	read exception status	read MACRO status of output ports
2	0x0f	force multiple coils	multiple write of output ports with BIT unit
-	0x69	write pulse	write output ports with BIT and TIME unit

2.2 Components

2.2.1 Modbus/TCP Master

A master sends request packets called query and it sends queries to a slave in periodically. After then, the master waits responses from slaves.

2.2.2 Modbus/TCP Slave

A slave sends response packets to the master.

☞ ***In the standard, a slave sends packets when it receives queries from the master. However, by using [Notify Input Port change], the ezTCP can sends packets without any queries from the master when its status of input ports has been changed in slave mode.***

2.3 Format of Modbus/TCP frame

Format of Modbus/TCP frame is described in the below figure.

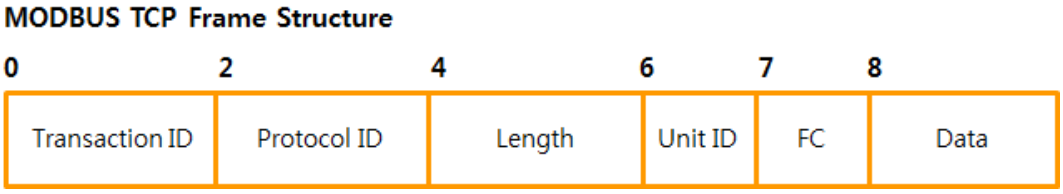


Figure 2-1 the format of Modbus/TCP frame

- byte 0 ~ 1: transaction ID (Transaction Identification)
  This means the sequence number of queries and responses. While operating as a master, ezTCP increases the value one by one in every query. (It is fine to set all the value to 0x0000)

☞ ***HEX: HEX is used as the contraction of Hexadecimal in this document like the notation of 0xABCD.***

- byte 2 ~ 3: protocol ID(Protocol Identification)
  This means the protocol identification and the value is fixed as 0x0000 for Modbus/TCP
- byte 4 ~ 5: length
  The value of this means the number of bytes from next byte of length field to the end of the frame.
- byte 6: unit ID(Unit Identification)
- byte 7: function code
- byte 8~ : data

2.4 Functions of Class 0

This is the minimum useful set of functions, for both a master and a slave.

- Read Multiple Registers
- Write Multiple Registers

2.5 Functions of Class 1

This is the additional set of functions which is commonly implemented and interoperable. Many slaves choose to treat input, output, discrete and register as equivalent.

- Read coils
- Read input discretes
- Read input registers
- Write coil
- Write single register
- Read exception status

2.6 Functions of Class 2

This is transfer set needed for routine operations such as HMI and supervision. ezTCP supports only the below function.

- Force multiple coils

2.7 Additional function

- Write pulse
  This function is designed by us to make pulse type operation in output port and it is not specified in the standard of Modbus/TCP.

3 Class 0 commands detail

3.1 Read multiple registers (FC 3)

3.1.1 Request

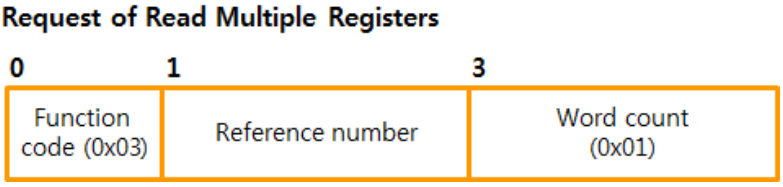


Figure 3-1 request of read multiple registers

- byte 0: function code
  Function code of read multiple register is 0x03.
- byte 1 ~ 2: reference number
  This is the initial address of an input port.
- byte 3 ~ 4: word count
  The value of word count for ezTCP is 0x01.

3.1.2 Response

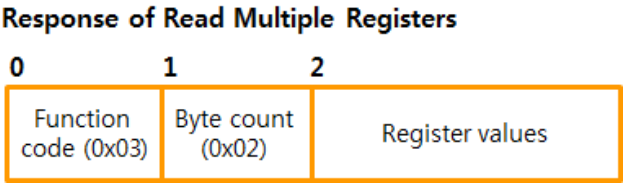


Figure 3-2 response of read multiple registers

- byte 0: function code (0x03)
- byte 1 ~ 2: byte count
  The byte count (B) in this frame is equal to word count × 2. In case of ezTCP, it should be 0x02.
- byte 2 ~ 3: register values
  This value shows status of the input ports. '0' means 'OFF' and '1' means 'ON'.

3.1.3 Exceptions

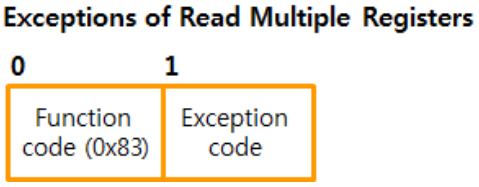


Figure 3-3 exceptions of read multiple registers

- byte 0: function code
  Function code of exception response is 0x83.
- byte 1: exception code
  Exception code can be 0x01 or 0x02.

3.1.4 Examples

- an example of request

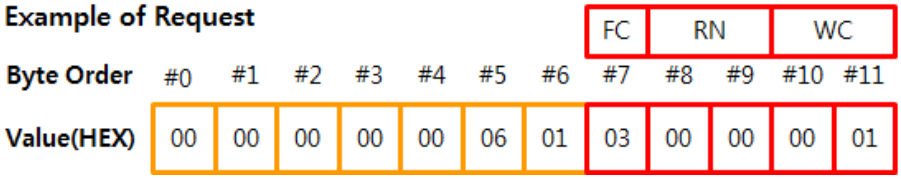


Figure 3-4 example of request

Table 3-1 request data

byte #	value	description
7	0x03	Function code is 3.
8~9	0x0000	Input port base address is 0.
10~11	0x0001	Word count is 1.

- an example of response

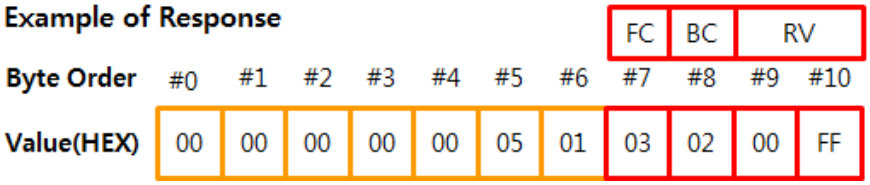


Figure 3-5 example of response

Table 3-2 response data

byte #	value	description
7	0x03	Function code is 3.
8	0x02	Byte count is 2.
9~10	0x00FF	Register value is 0xFF. (All ports are ON)

3.2 Write Multiple Registers (FC 16)

3.2.1 Request

Request of Write Multiple Registers

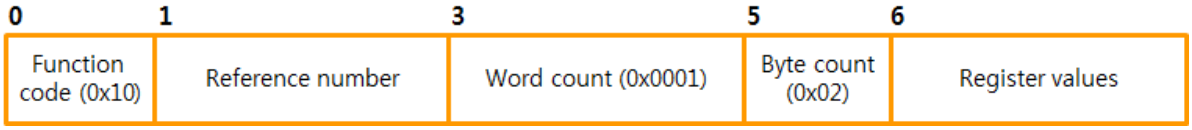


Figure 3-6 request of write multiple registers

- byte 0: function code
  Function code of write multiple registers is 0x10.
- byte 1~2: reference number
  This is output port base address.
- byte 3~4: word count (0x01)
- byte 5: byte count (0x02)
- byte 6~7: register values
  This value represents the status of output ports. '0' means 'OFF' and '1' means 'ON'.

☞ ***Port assignment in the register values***

LSB of the last byte of register values represents the first port (port #0) and the MSB means the last port (port #7).

Figure 3-7 port assignment

3.2.2 Response

Response of Write Multiple Registers

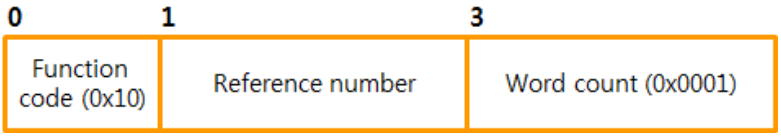


Figure 3-8 response of write multiple registers

- byte 0: function code (0x10)
- byte 1~2: reference number
- byte 3~4: word count (0x01)

3.2.3 Exception

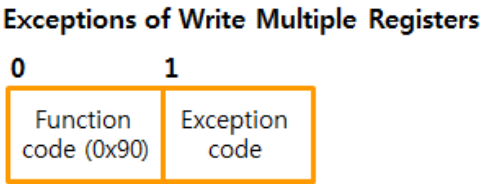


Figure 3-9 exceptions of write multiple registers

- byte 0: function code
  Function code of exception response is 0x90.
- byte 1: exception code
  Exception code can be 0x01 or 0x02.

3.2.4 Examples

- an example of request

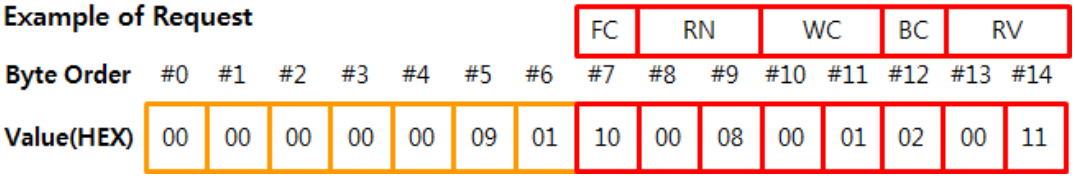


Figure 3-10 an example of request

Table 3-3 request data

byte #	value	description
7	0x10	Function code is 16.
8~9	0x0008	Output port base address is 8.
10~11	0x0001	Word count is 1.
12	0x02	Byte count is 2.
13~14	0x0011	Register value is 0x0011. (0001 0001 – ports #0, 4 ON)

- an example of response

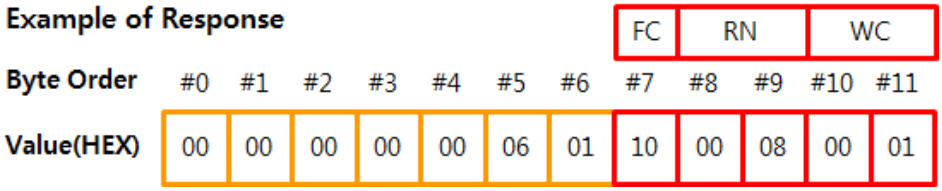


Figure 3-11 an example of response

Table 3-4 response data

byte #	value	description
7	0x03	Function code is 16.
8~9	0x0008	Output port base address is 8.
10~11	0x0001	Word count is 1.

4 Class 1 commands detail

4.1 Read coils (FC 1)

4.1.1 Request

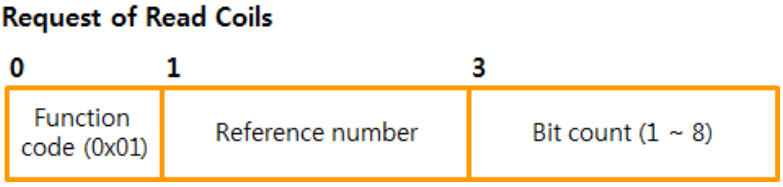


Figure 4-1 request of read coils

- byte 0: function code
  Function code of read coils is 0x01.
- byte 1~2: reference number
  It is the initial address of an output port to read the value.
- byte 3~4: bit count
  You can set the number of output ports you want to read in here. Remember that ezTCP has 8 output ports (maximum). For example, set the reference number to 0x0008 and bit count to 0x0004 if you want to read 4 output ports from the first output port (port 0).

4.1.2 Response

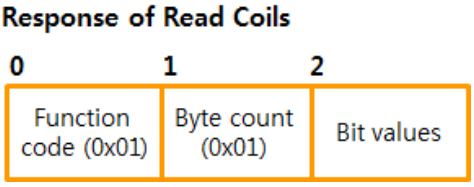


Figure 4-2 response of read coils

- byte 0: function code (0x01)
- byte 1: byte count (0x02)
  In case of this frame, byte count (B) should be 0x01. (B = (bit count + 7) / 8)
- byte 2: bit values
  This value represents the status of output ports. The LSB is assigned to the first port.

4.1.3 Exception

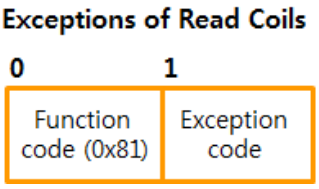


Figure 4-3 exceptions of read coils

- byte 0: function code
  Function code of exception response is 0x81.
- byte 1: exception code
  Exception code can be 0x01 or 0x02.

4.1.4 Examples

- an example of request

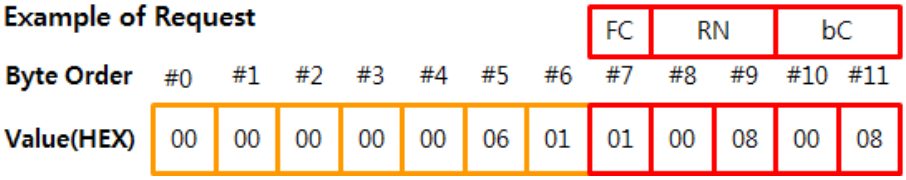


Figure 4-4 an example of request

Table 4-1 request data

byte #	value	description
7	0x01	Function code is 1
8~9	0x0008	Output port base address is 8.
10~11	0x0008	Status of 8 ports is read.

- an example of response

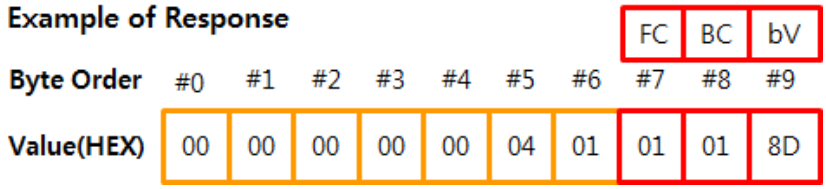


Figure 4-5 an example of response

Table 4-2 response data

byte #	value	description
7	0x01	Function code is 1
8	0x01	Byte count is 1.
9	0x8D	Bit count is 0x8D. (1000 1101 – ports #0, 2, 3, 7 ON)

4.2 Read input discretes (FC 2)

4.2.1 Request

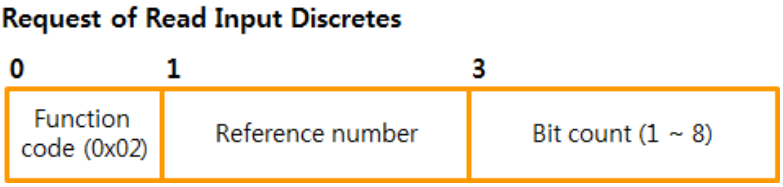


Figure 4-6 request of read input discretes

- byte 0: function code
  Function code of read input discretes is 0x02.
- byte 1~2: reference number
  This value represents the first port you want to read the status.
- byte 3~4: bit count
  You can set the number of input ports you want to read in here. Remember that ezTCP has 8 input ports (maximum). For example, set the reference number to 0x0000 and bit count to 0x0004 if you want to read 4 input ports from the first input port (port 0).

4.2.2 Response

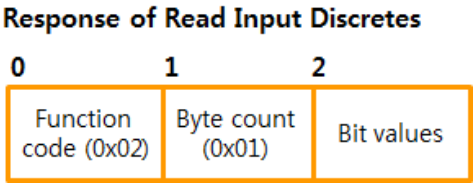


Figure 4-7 response of read input discretes

- byte 0: function code (0x02)
- byte 1: byte count (0x01)
  In this fame, byte count (B) should be 0x01. (B = (bit count + 7) / 8)
- byte 2: bit values
  This value represents the status of input ports. The LSB is assigned to the first port.

4.2.3 Exception

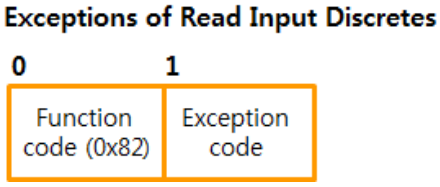


Figure 4-8 exceptions of read input discretes

- byte 0: function code
  Function code of exception response is 0x82.
- byte 1: exception code
  Exception code can be 0x01 or 0x02.

4.2.4 Examples

- an example of request

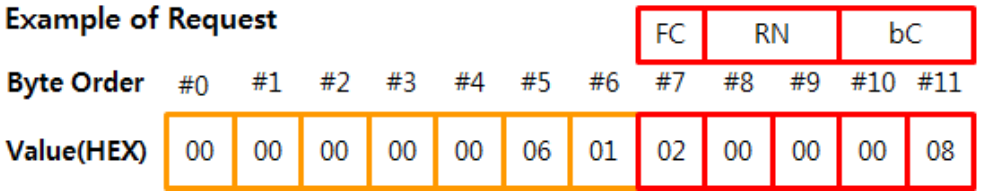


Figure 4-9 an example of request

Table 4-3 request data

byte #	value	description
7	0x02	Function code is 2.
8~9	0x0000	Input port base address is 0.
10~11	0x0008	Status of 8 ports is read.

- an example of response

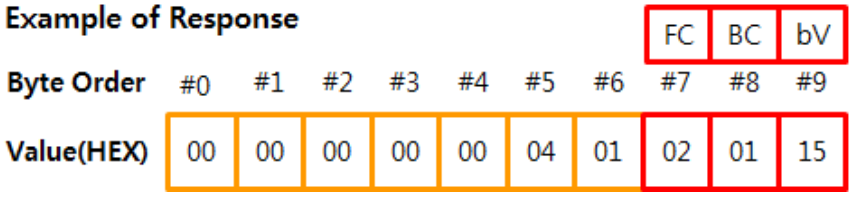


Figure 4-10 an example of response

Table 4-4 response data

byte #	value	description
7	0x02	Function code is 2.
8	0x01	Byte count is 1.
9	0x15	Bit value is 0x15. (0001 0101 – port #0, 2, 4 ON)

4.3 Read input registers (FC 4)

4.3.1 Request

Request of Read Input Registers

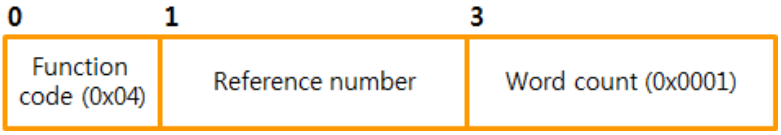


Figure 4-11 request of read input registers

- byte 0: function code
  Function code of read input registers is 0x04.
- byte 1~2: reference number
  The value of this field should be the input port base address.
- byte 3~4: word count
  Word count is fixed to 0x0001.

4.3.2 Response

Response of Read Input Registers

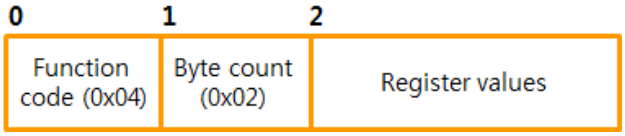


Figure 4-12 response of read input registers

- byte 0: function code (0x04)
- byte 1: byte count (0x02)
  The byte count (B) in this frame is equal to word count × 2. In case of ezTCP, it should be 0x02.
- byte 2~3: register values
  This value means the status of input ports. The LSB is the reference initial address.

4.3.3 Exception

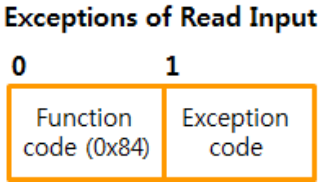


Figure 4-13 exceptions of read input registers

- byte 0: function code
  Function code of exception response is 0x84.
- byte 1: exception code
  Exception code should be 0x01 or 0x02.

4.3.4 Examples

- an example of request

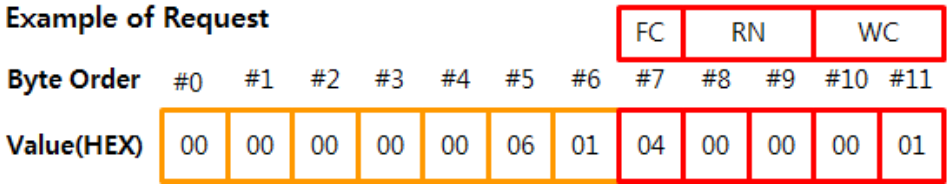


Figure 4-14 an example of request

Table 4-5 request data

byte #	value	description
7	0x04	Function code is 4
8~9	0x0000	Input port base address is 0.
10~11	0x0001	Word count is 1.

- an example of response

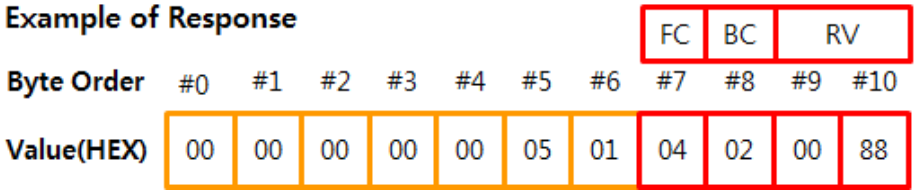


Figure 4-15 an example of response

Table 4-6 response data

byte #	value	description
7	0x04	Function code is 4.
8	0x02	Byte count is 2.
9~10	0x0088	Register value is 0x88. (1000 1000 – port #3, 7 ON)

4.4 Write coil (FC 5)

4.4.1 Request / Response

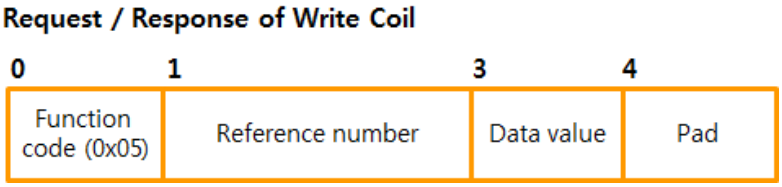


Figure 4-16 request / response of write coil

- byte 0: function code
  Function code of write coil is 0x05.
- byte 1~2: reference number
  This is address of an output port you want to control.
- byte 3: data value
  This value can be 0xFF or 0x00. '0xFF' means 'ON' and '0x00) means 'OFF'.

☞ ***In case of this function, the frame format of request is the same with response.***

4.4.2 Exception

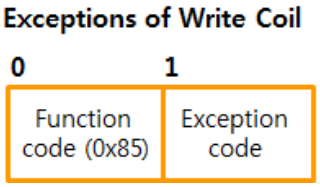


Figure 4-17 exceptions of write coil

- byte 0: function code
  Function code of exception response is 0x85.
- byte 1: exception code
  Exception code can be 0x01 or 0x02.

4.4.3 Examples

- an example of request / response

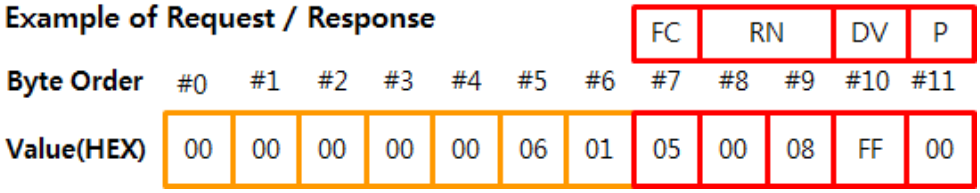


Figure 4-18 an example of request / response

Table 4-7 request / response data

byte #	value	description
7	0x05	Function code is 5.
8~9	0x0008	Output port address is 0x0008.
10	0xFF	Data value is 0xFF. (port ON)
11	0x00	pad (usually set to 0x00)

4.5 Write single register (FC 6)

4.5.1 Request / Response

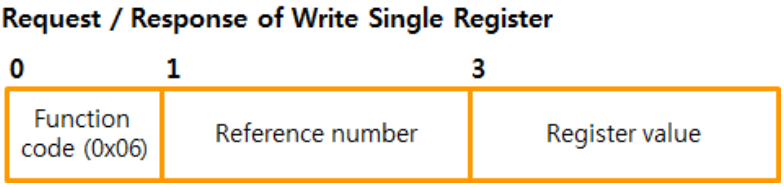


Figure 4-19 request / response of write single register

- byte 0: function code
  Function code of write single register is 0x06.
- byte 1~2: reference number
  This is the output port base address.
- byte 3~4: register value
  This value represents the status of output ports. '0' means 'OFF' and '1' means 'ON'. LSB is assigned to the first port.

☞ ***In case of this function, the frame format of request is the same with response.***

4.5.2 Exception

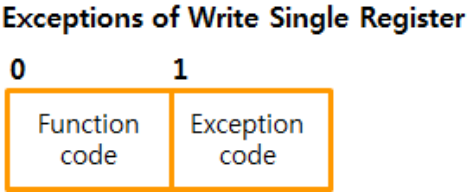


Figure 4-20 exceptions of write single register

- byte 0: function code
  Function code of exception response is 0x86.
- byte 1: exception code
  Exception code can be 0x01 or 0x02.

4.5.3 Examples

- an example of request / response

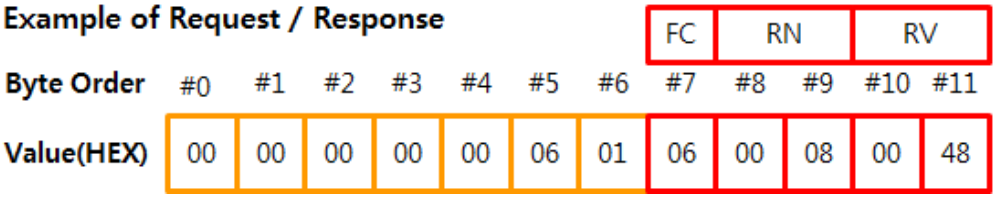


Figure 4-21 an example of request / response

Table 4-8 request / response data

byte #	value	description
7	0x06	Function code is 6.
8~9	0x0008	Output port base address is 8.
10~11	0x0048	Register value is 0x0048. (0100 1000 – port #3, 6 ON)

4.6 Read exception status (FC 7)

Note that ‘exception status’ has nothing to do with ‘exception response’. The ‘read exception status’ has been added to distinguish ports that Macro was set among output ports of the ezTCP.

4.6.1 Request

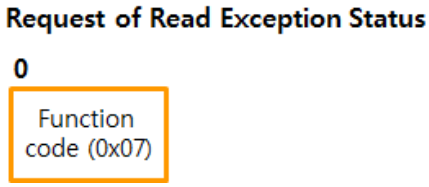


Figure 4-22 request of read exception status

- byte 0: function code
  Function code of read exception status is 0x07.

4.6.2 Response

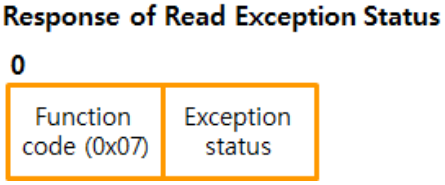


Figure 4-23 response of read exception status

- byte 0: function code (0x07)
- byte 1: exception status
  ‘1’ means the port is set to the macro mode and ‘0’ means the port is not. LSB is assigned to the first port.

4.6.3 Exception

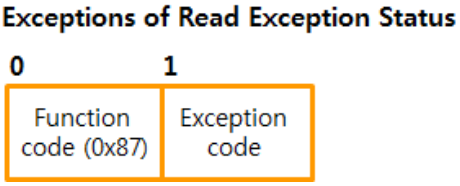


Figure 4-24 exceptions of read exception status

- byte 0: function code
  Function code of exception response is 0x87.
- byte 1: exception code
  Exception code can be 0x01 or 0x02.

4.6.4 Examples

- an example of request

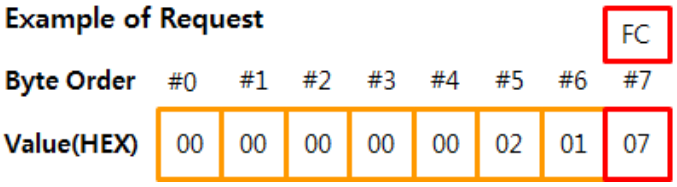


Figure 4-25 an example of request

Table 4-9 request data

byte #	value	description
7	0x07	Function code is 7.

- an example of response

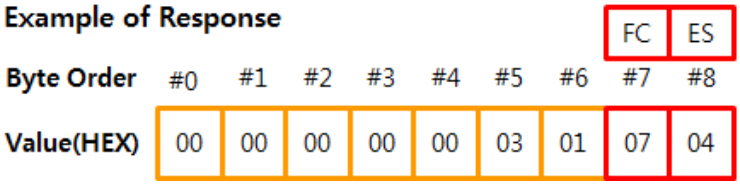


Figure 4-26 an example of response

Table 4-10 response data

byte #	value	description
7	0x07	Function code is 7.
8	0x04	Exception code is 0x04. (0000 1000 – port #3 Macro)

5 Class 2 commands detail

5.1 Force multiple coils (FC 15)

5.1.1 Request

Request of Force Multiple Coils

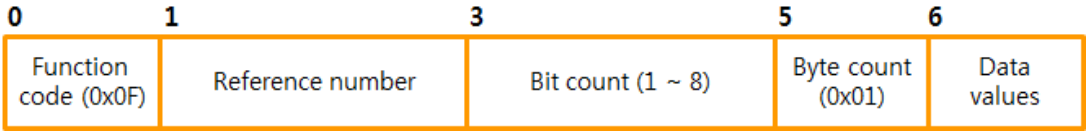


Figure 5-1 request of force multiple coils

- byte 0: function code
  Function code of force multiple coils is 0x0F.
- byte 1~2: reference number
  This is output port base address.
- byte 3~4: bit count
  You can set the number of output ports you want to control in here. Remember that ezTCP has 8 input ports (maximum).
- byte 5: byte count (0x01)
- byte 6: data values
  This value represents the status of output ports. '0' means 'OFF' and '1' means 'ON'. LSB is assigned to the first port.

5.1.2 Response

Response of Force Multiple Coils

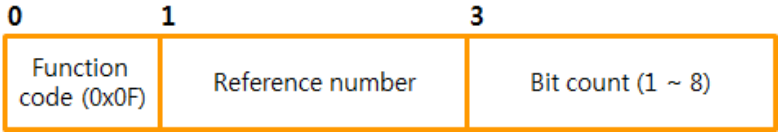


Figure 5-2 response of force multiple coils

- byte 0: function code (0x0F)
- byte 1~2: reference number
- byte 3~4: bit count

5.1.3 Exception

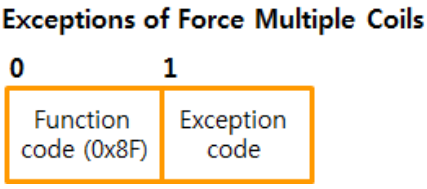


Figure 5-3 exceptions of force multiple coils

- byte 0: function code
  Function code of exception response ix 0x8F.
- byte 1: exception code
  Exception code can be 0x01 or 0x02.

5.1.4 Examples

- an example of request

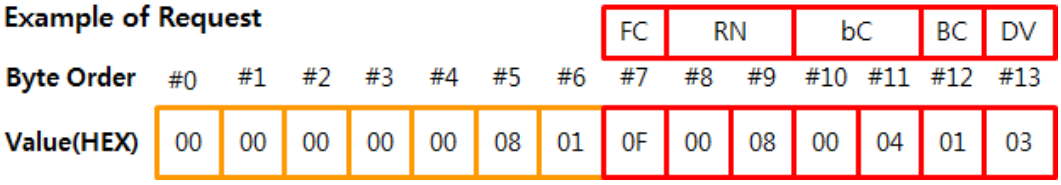


Figure 5-4 an example of request

Table 5-1 request data

byte #	value	description
7	0x0F	Function code is 15.
8~9	0x0008	Output port base address is 8.
10~11	0x0004	4 output ports from the first port are controlled.
12	0x01	Byte count is 1.
13	0x03	Data value is 0x03. (0000 0011 – port #0, 1 ON)

- an example of response

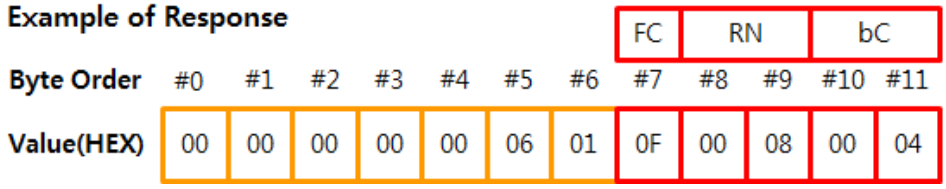


Figure 5-5 an example of response

Table 5-2 response data

byte #	value	description
7	0x0F	Function code is 15.
8~9	0x0008	Output port base address is 8.
10~11	0x0004	4 output ports from the first port are controlled.

6 The other commands detail

6.1 Write Pulse (FC 105)

This function is designed by us to make pulse type operation in output port and it is not specified in the standard of Modbus/TCP.

6.1.1 Request / Response

Request / Response of Write Pulse

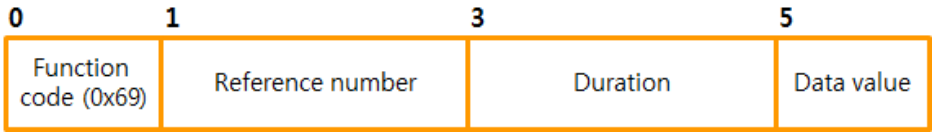


Figure 6-1 request / response of write pulse

- byte 0: function code
  Function code of write pulse is 0x69.
- byte 1~2: reference number
  This is address of an output port you want to control.
- byte 3~4: duration
  The unit is millisecond. You can set this value from 40 to 10000. (0x0028 ~ 0x2710)
- byte 5: On/Off
  Data value can be '0xFF' or '0x00'. '0xFF' means 'ON' and '0xFF' means 'OFF' for the duration.

☞ ***In case of this function, the frame format of request is the same with response.***

6.1.2 Exception

Exceptions of Write Pulse

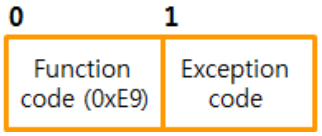


Figure 6-2 exceptions of write pulse

- byte 0: function code
  Function code of exception response is 0xE9.
- byte 1: exception code
  Exception code can be 0x01, 0x02, 0x03 or 0x06.

6.1.3 Examples

- an example of request / response

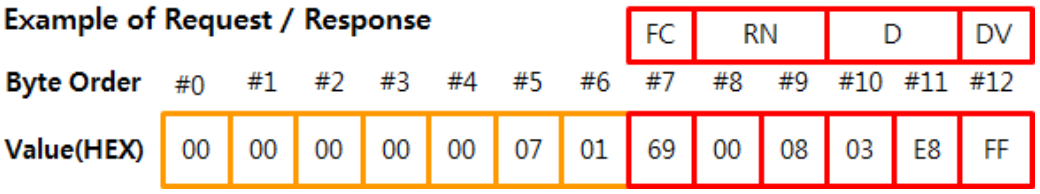


Figure 6-3 an example of request / response

Table 6-1 request / response data

byte #	value	description
7	0x69	Function code is 105.
8~9	0x0008	Output port address is 8.
10~11	0x03E8	Duration is 1 seconds. (1000ms = 0x03E8)
12	0xFF	Data value is 0xFF (ON for the duration)

☞ ***If a port is already being operated by FC 105, the port cannot be controled until the timer is expired.***

7 Additional Instructions

7.1 Exception codes

Table 7-1 exception codes

exception code	description
0x01	Illegal Function
0x02	Illegal Data Address
0x03	Illegal Data Value
0x06	Slave Device Busy

7.2 ADC values

7.2.1 Request

CIE-M10 has an Analog to Digital Conversion (ADC) port. To query this value, you should send a command request of read multiple registers. Note that you should set the reference number to specific address.

ADC values are stored at [Input Port Base Address] + 4. This value is reference number of read multiple registers. For example, if the [Input Port Base Address] is set to zero, you have to set the reference number to 4(0x00 0x04). See the below example.

- an example of request

Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
Value(HEX)	00	00	00	00	00	06	01	03	00	04	00	01

Figure 7-1 an example of request

7.2.2 Response

You can read the value of ADC port from the last two bytes of command response of read multiple registers.

- an example of response

Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Value(HEX)	00	00	00	00	00	05	01	03	02	02	7f

Figure 7-2 an example of response

In the above example, ADC value is 0x02 0x7f in hexadecimal. (639 in decimal)

☞ ***The ADC port of CIE-M10 has 10 bit resolution. (0 ~ 1023)***

7.3 Using

7.3.1 Modbus/TCP Configuration

- CIE-M10/H10/H12
  With ezManager, search the products and configure the values in the box ③.

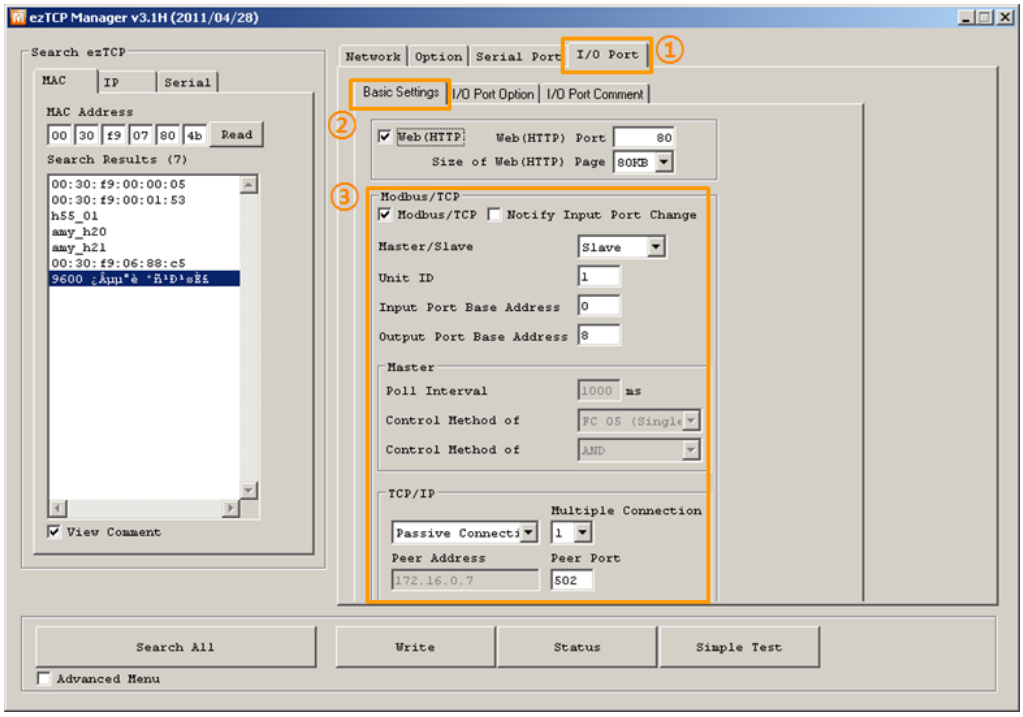


Figure 7-3 configuration for Modbus/TCP on ezManager

- EZI-10
  With ezConfigIO, search the products and configure the values in the orange box.

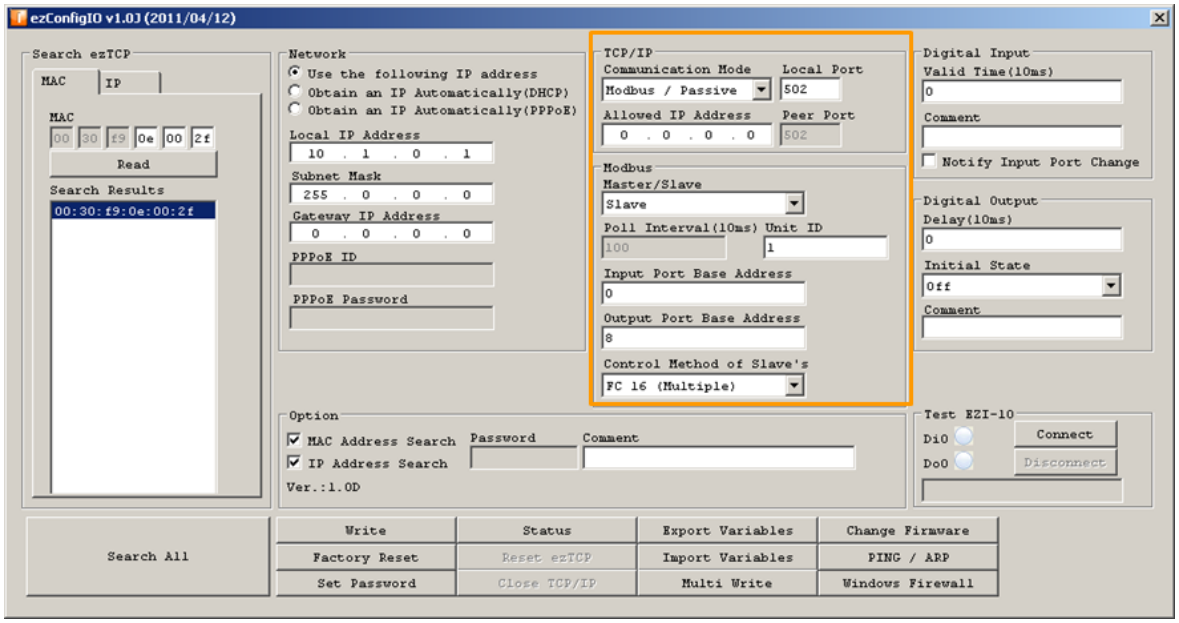


Figure 7-4 configuration for Modbus/TCP on ezConfigIO

- Parameters

Table 7-2 parameters of Modbus/TCP

parameter	description
Modbus/TCP	Whether the Modbus/TCP use or not.
Notify Input Port Change	If input status has been changed, slave sends response although doesn't receive any query from master.
Master/Slave	Operation Type.
Poll Interval	Interval for sending query (Unit: millisecond)
Unit ID	ID for the pair of the master and slave
Input Port Base Address	The address for the first input port
Output Port Base Address	The address for the first output port
Control Method of (FCXX)	Control method for output ports of slave (Single / Multiple)
Control Method of (AND/OR)	Control method for output ports of master (AND/OR)
Active/Passive Connection	Active or Passive connection
Multiple Connection	Activation of multiple connection (1 ~ 8)
Peer Address	Peer's IP address under active connection
Peer Port	Peer's port number

7.3.2 Configuration Example

Table 7-3 an example of configuration

parameter	ezTCP	another ezTCP or Modbus/TCP program
Local IP Address	192.168.0.10	192.168.0.20
Subnet Mask	255.255.255.0	255.255.255.0
Modbus/TCP	Check or Select	-
Master / Slave	Slave	Master
Poll Interval	-	1,000㎳ (1 sec)
Unit ID	1	1
Input Port Base Address	0	0
Output Port Base Address	8	8
Active/Passive Connection	Passive	Active
Multiple Connection	3	-
Peer Address	-	192.168.0.10
Peer Port	-	502
Local Port	502	-

7.4 Sample Codes

We have been offering sample codes for users which will make Modbus/TCP application program to use our I/O controllers.

The codes can be downloaded on the [SUPPORT]>>[Download]>>[Sample Codes] page on our web site. (http://www.eztcp.com/en/support/sample_code.php)

- CModBusEngine
  The class which performs the Modbus/TCP

7.4.1 Functions

- SendReadRequest
  This function is for read input and output ports of ezTCP.

Table 7-4 parameters of SendReadRequest function

parameter	description
Transaction_id	Transaction identification
Unit_id	Unit identification
address	Input and output port address

- SendWriteRequest
  This function is for write output

Table 7-5 parameters of SendWriteRequest function

parameter	description
Transaction_id	Transaction identification
Unit_id	Unit identification
address	Input and output port address
value	Values of output ports

- OnReceive
  This function handles response packets of Modbus/TCP

8 Serialized Modbus/TCP

Serialized Modbus/TCP mode is for only CIE-M10 and H10. The other communication modes are all disabled when using this mode. The RS232 port is used for this.

8.1 Features

- Sending and Receiving the Modbus/TCP data through the RS232
- Controlling digital I/O ports via a serial port.
- No connection processes.
  In this mode, devices (or terminals) sends and receives data without any connection processes. Use the hardware flow control (RTS/CTS) to prevent data loss.

8.2 Using

8.2.1 Configuration

- Serialized Modbus/TCP configuration

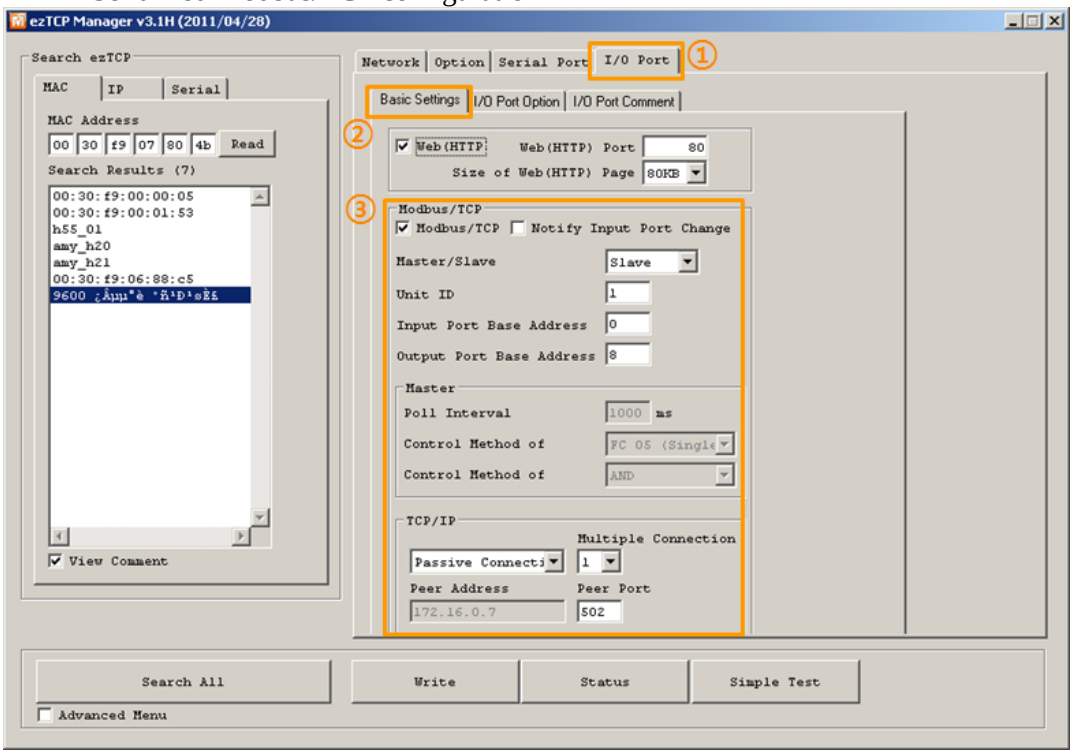


Figure 8-1 configuration of serialized Modbus/TCP

① Moving to the [Serial Port] tab

② Setting and Checking parameters for the serial port

③ Selecting [Serialized Modbus/TCP] on the [Communication Mode]

④ Pressing the [Write] button for the saving

8.3 Trial Run

8.3.1 Preperations for Communication

For testing the serialized Modbus/TCP mode, design the connection as follows:

☞ ***Connection via an Ethernet cable is not required.***

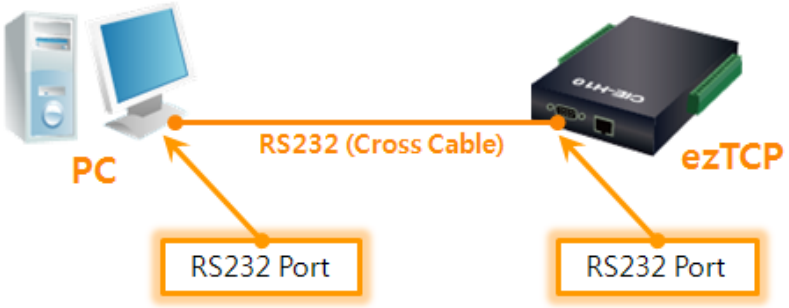


Figure 8-2 connection diagram

Then, for the test, the default value of Modbus/TCP setting has to be kept as follows:

Table 8-1 default values for the parameters

Name	Default Value
Modbus/TCP	Checked
Notify Input Port Change	Unchecked
Master / Slave	Slave
Poll Interval	1,000
Unit ID	1
Input Port	0
Output Port	8

8.3.2 Sending an Example data

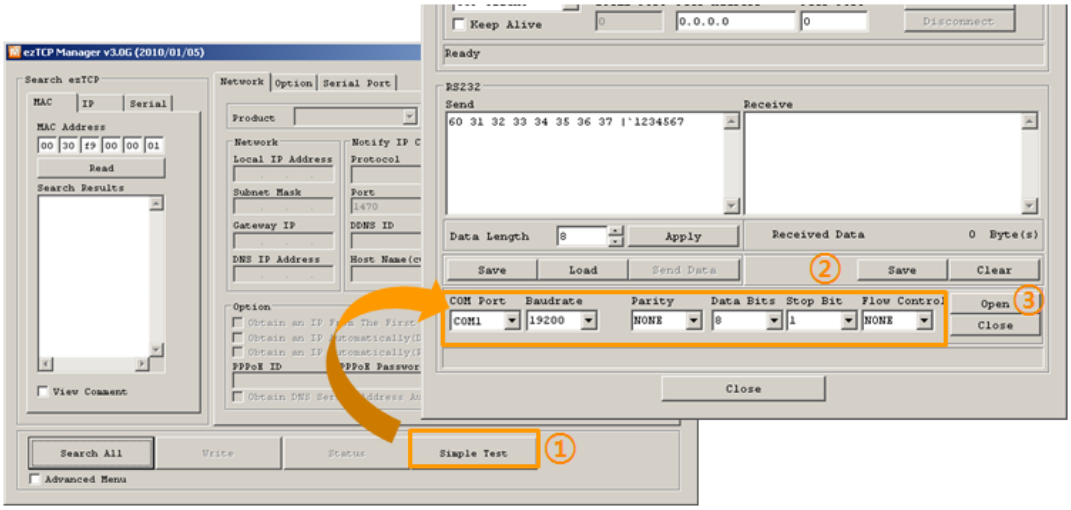


Figure 8-3 running Simple Test Program

① Clicking the [Simple Test] button.

② Selecting and checking all the parameters related with the serial port.

③ Pressing the [Open] button.

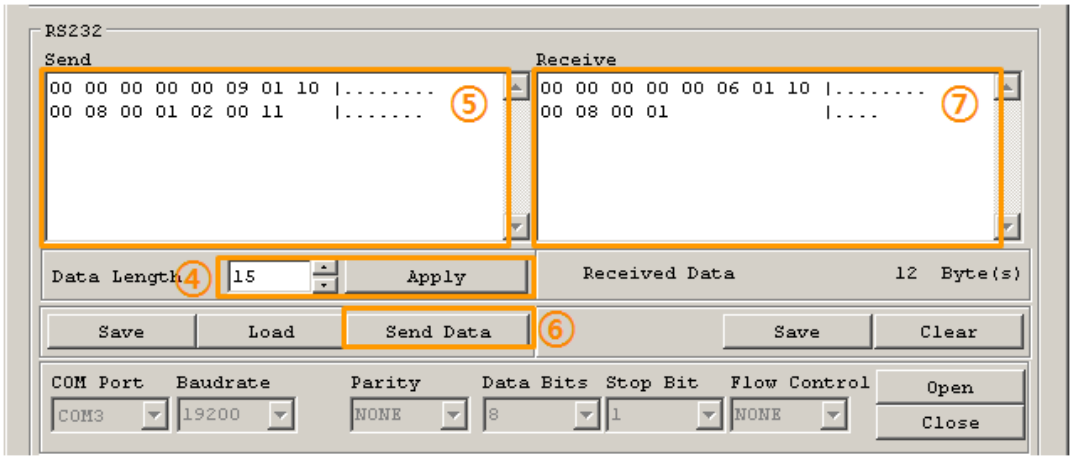


Figure 8-4 sending an example data

④ Click the [Apply] button after setting the [Data Length] to 15 bytes.

⑤ Input the data which is used in the example data of write multiple registers.

⑥ Press the [Send Data] button.

⑦ Check the real output ports of ezTCP and if the response data in the [Receive] box is the same with the above figure.

☞ ***The data sent on the step ⑤ means turning on the output port 0 and 4 (Write Multiple Registers). The slave should response data appeared on the step ⑦.***

9 Revision History

Date	Version	Comments	Author
2010.03.08	1.0	○ Initial Release	Roy LEE
2010.07.20	1.1	○ Name of the document has been changed ○ Contents of the Modbus/TCP document has been included ○ Contents about the EZI-10 has been added	Roy LEE
2010.11.23	1.2	○ Descriptions about querying/response of ADC values have been added ○ Date item on the front page has been removed.	Roy LEE
2011.06.24	1.3	○ Contents about new functions have been added. (FC 1, 2, 4, 5, 6, 7, 15, 105) ○ Most of figures have been updated. ○ Figures about configuration tools have been updated. ○ Document structure has been redesigned. ○ Title of document has been changed.	Roy LEE