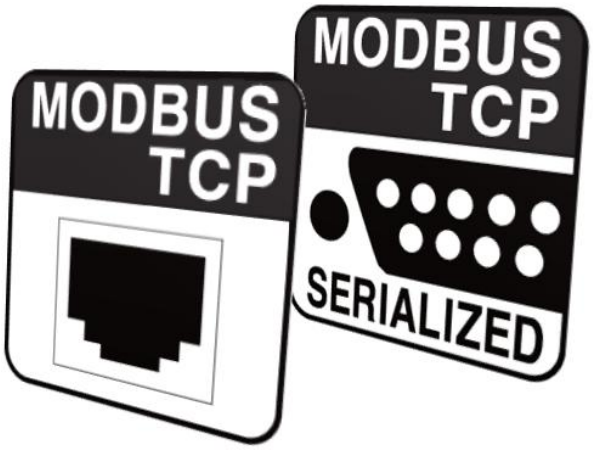
ezTCP Technical Document

Modbus/TCP

Version 1.1

2010-07-20



☞ ***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 Programmable Logic Controllers (PLCs). Modbus/TCP is the one of the Modbus' versions which performs 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 of ezTCP was developed to require these demands.

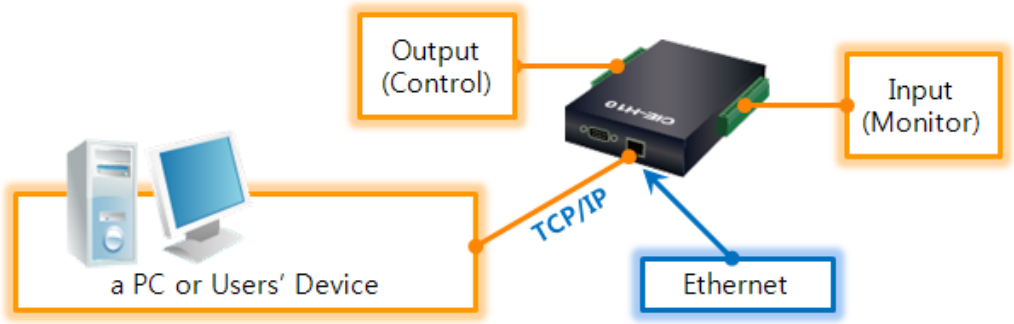


Figure 1-1 a diagram of Modbus/TCP system

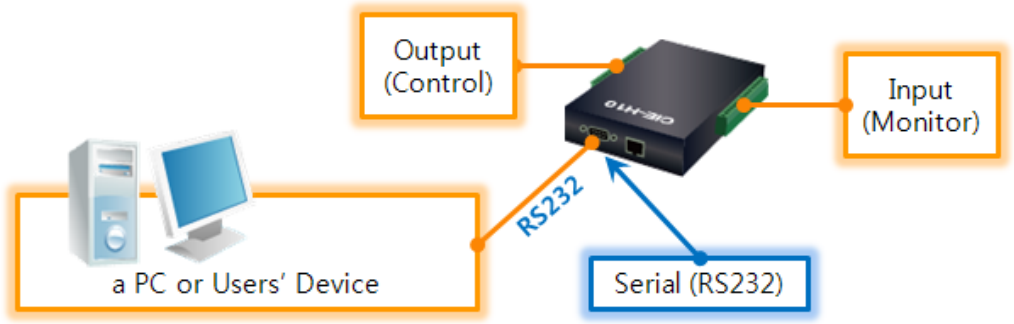


Figure 1-2 a diagram of Serialized Modbus/TCP system

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

2 Modbus/TCP

2.1 Features

- TCP/IP version of Modbus/TCP protocol
- Connection process
  In the standard, a slave operates only TCP client. However, ezTCP can be performed both TCP server and client. Port number should be TCP 502.
- Master and Slave
  In difference with the standard, ezTCP can be not only a Modbus/TCP Master but also Slave 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 Class 0

Function Code	Name	Description
0x03	Read Multiple Registers	Master: Query for states of input ports
		Slave: Response for the read query
0x10	Write Multiple Registers	Master: Command for writing output ports
		Slave: Response for the write command

2.2 Components

2.2.1 Modbus/TCP Master

A master sends request packets and the packet is called query. A master sends the queries to a Slave in periodically and waits the responses.

2.2.2 Modbus/TCP Slave

A slave sends response packets to the master.

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

2.3 Data Format of Modbus/TCP Header

Header of Modbus/TCP is described in the below figure. It always leads all the Modbus/TCP data in front of them.

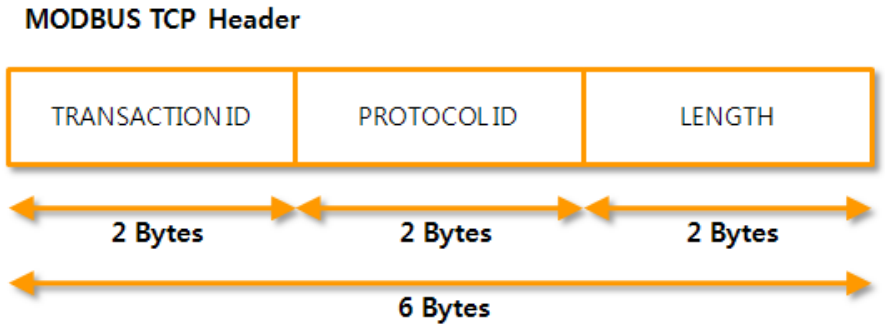


Figure 2-1 the format of Modbus/TCP Header

- TRANSACTION ID (Transaction Identification)
  This means the sequence number related with queries and responses. While operating as a master, ezTCP increases the value one by one in every query. (It is okay to set all the value to 0x0000)

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

- PROTOCOL ID (Protocol Identification)
  This means the protocol identification and the value is fixed as 0x0000 for Modbus/TCP
- LENGTH
  The value of this means the number of bytes from next byte to the end of the frame.

2.4 Functions in Class 0

The below two functions which is in Class 0 are only available on ezTCP.

2.4.1 Read Multiple Registers

- For reading input and output ports
- Function Code: 0x03(HEX)

2.4.2 Write Multiple Registers

- For giving signals to the output ports
- Function Code: 0x10(HEX)

2.5 Types of Communication

Except for the exception codes, the total number of communication data format is 4 because each master and slave use function number 3(0x03) and 16(0x10).

2.5.1 Command Request of Read Multiple Registers

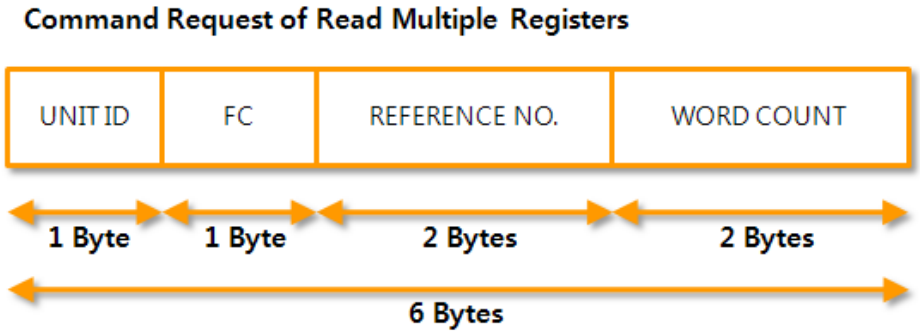


Figure 2-2 Command Request of Read Multiple Registers

- UNIT ID (Unit Identification)
  This is the configured value for the master and slave.
- FC (Function Code)
  The function code of Read Multiple Registers is 0x03 in hexadecimal.
- REFERENCE NO. (Reference Number)
  These 2 bytes mean the start address of the digital input port. You can configure this value by ezManager. (Default: 0x0000)
- WORD COUNT
  ezTCP uses 0x01(HEX) as default value of WORD COUNT.
- An example

Figure 2-3 an example data of read multiple registers from the master

Table 2-2 descriptions of the examples

Byte order	Value(HEX)	Description
0~1	0x0000	Transaction ID is 0x0000 in HEX
2~3	0x0000	Protocol ID is 0x0000 in HEX (Fixed)
4~5	0x0006	The number of rest bytes of the frame is 6. (Fixed)
6	0x01	Unit ID is set to 1.
7	0x03	0x03 is the function code for Read Multiple Registers. (Fixed)
8~9	0x0000	The first address of the input ports is set to 0.
10~11	0x0001	Word Count is 0x0001 in HEX (Fixed)

2.5.2 Command Response of Read Multiple Registers

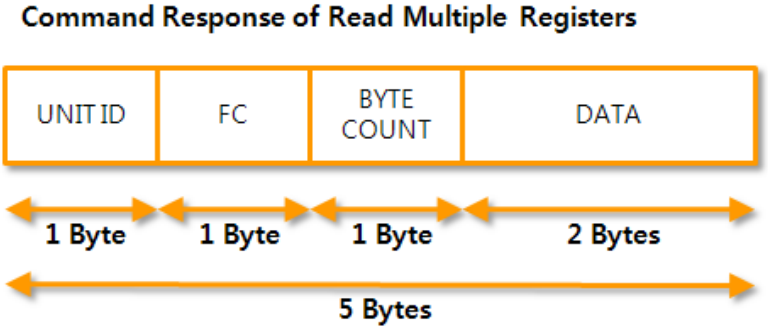


Figure 2-4 Command Response of Read Multiple Registers

- UNIT ID
- FC
  The function code of Read Multiple Registers is 0x03 in hexadecimal.
- BYTE COUNT
  BYTE COUNT has to be double compared to WORD COUNT. (Default: 0x02)
- DATA (Register Values)
  The data presents the status of input ports. Each bit is assigned for the input port and the value 0 means the port is OFF and value 1 means the port is ON.

Table 2-3 bit assignment

Value	Description
0	Input is OFF
1	Input is ON

As you can see in the below figure, the Least Significant Bit represents the state of input number 0.

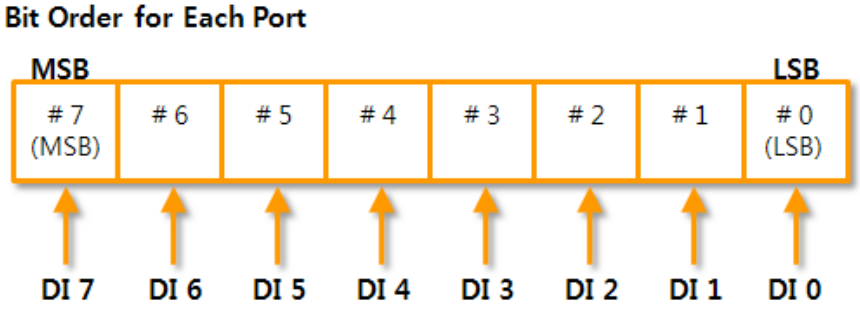


Figure 2-5 bit order for each port

- An example

Figure 2-6 an example data of read multiple registers from the slave

Table 2-4 descriptions of the examples

Byte order	Value(HEX)	Description
0~1	0x0000	Transaction ID is 0x0000 in HEX
2~3	0x0000	Protocol ID is 0x0000 in HEX (Fixed)
4~5	0x0005	The number of rest bytes of the frame is 5. (Fixed)
6	0x01	Unit ID is set to 1.
7	0x03	0x03 is the function code for Read Multiple Registers. (Fixed)
8	0x02	Byte Count = Word Count * 2 (Fixed)
9~10	0x00FF	All the states of port number 0 to 7 are ON. (expression in bit: 0000 0000 1111 1111)

2.5.1 Command Exception of Read Multiple Registers

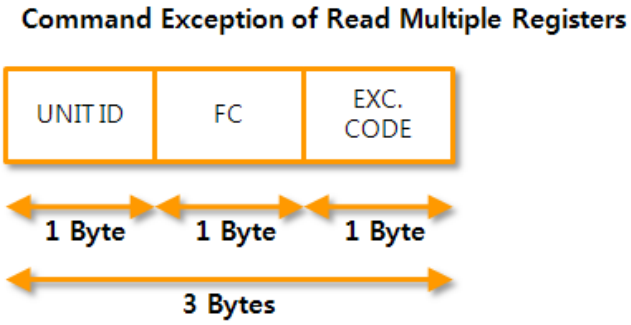


Figure 2-7 Command Exception of Read Multiple Registers

- UNIT ID
- FC
  It should be 0x83 in HEX.
- EXCEPTION CODE
  It could be 0x01 or 02 in HEX

Table 2-5 exception codes

Value	Description
0x01	The function code is not an allowable action for the slave or the slave is in the wrong state.
0x02	The data address is not allowable address for the slave.

2.5.2 Command Requst of Write Multiple Registers

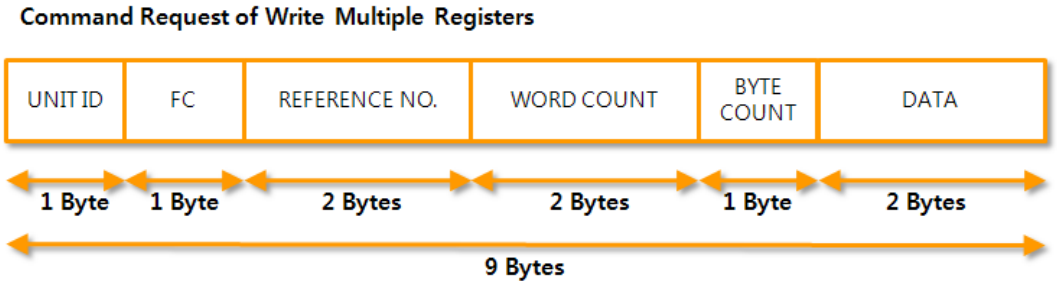


Figure 2-8 command request of write multiple registers

- UNIT ID
- FC(Function Code)
  The function code of Read Multiple Registers is 0x10 in hexadecimal.
- REFERENCE NO. (Reference Number)
  These 2 bytes mean the start address of the digital output port. You can configure this value by ezManager. (Default: 0x0008)
- WORD COUNT
- BYTE COUNT
- DATA (Register Values)
  The data presents turning the output ports ON. Each bit is assigned for the output port and the value 0 means turn the port OFF and value 1 means turn the port ON.

As you can see, the Least Significant Bit represents the state of output number 0.

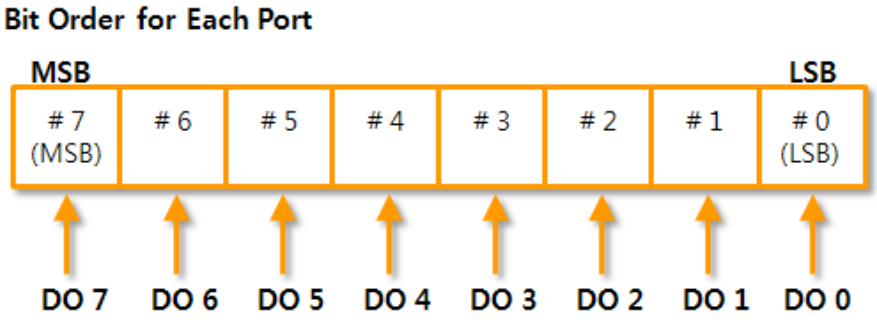


Figure 2-9 bit order for each port

- An example

Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Value(HEX)	00	00	00	00	00	09	01	10	00	08	00	01	02	00	11

Figure 2-10 an example data of write multiple registers from the master

Table 2-6 descriptions of the examples

Byte order	Value(HEX)	Description
0~1	0x0000	Transaction ID is 0x0000 in HEX
2~3	0x0000	Protocol ID is 0x0000 in HEX (Fixed)
4~5	0x0005	The number of rest bytes of the frame is 5. (Fixed)
6	0x01	Unit ID is set to 1.
7	0x10	0x10 is the function code for Write Multiple Registers. (Fixed)
8~9	0x0008	The first address of the output ports is set to 8.
10~11	0x0001	Word Count is 0x0001 in HEX (Fixed)
12	0x02	Byte Count = Word Count * 2 (Fixed)
13~14	0x0011	Turn the number 0 and 4 of the output ports ON. (expression in bit: 0000 0000 0001 0001)

2.5.3 Command Response of Write Multiple Registers

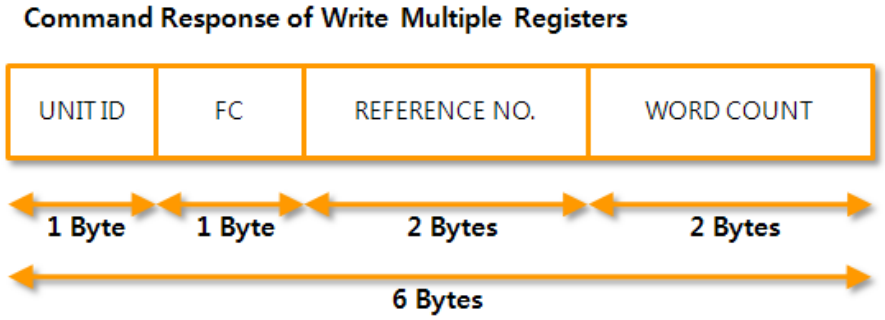


Figure 2-11 command response of write multiple registers

- UNIT ID
- FC
  The function code of Read Multiple Registers is 0x10 in hexadecimal.
- REFERENCE NO. (Reference Number)
  These 2 bytes mean the start address of the digital output port. You can configure this value by ezManager. (Default: 0x0008)
- WORD COUNT
- An example

Figure 2-12 an example data of write multiple registers from the slave

Table 2-7 descriptions of the example

Byte order	Value(HEX)	Description
0~1	0x0000	Transaction ID is 0x0000 in HEX
2~3	0x0000	Protocol ID is 0x0000 in HEX (Fixed)
4~5	0x0006	The number of rest bytes of the frame is 6. (Fixed)
6	0x01	Unit ID is set to 1.
7	0x10	0x10 is the function code for Write Multiple Registers. (Fixed)
8~9	0x0008	The first address of the output ports is set to 8.
10~11	0x0001	Word Count is 0x0001 in HEX (Fixed)

2.5.4 Command Exception of Write Multiple Registers

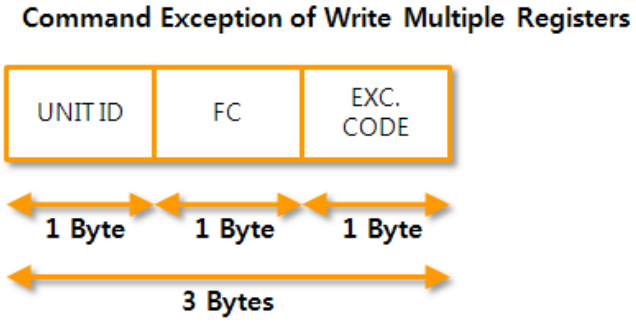


Figure 2-13 Command Exception of Write Multiple Registers

- UNIT ID
- FC
  It should be 0x90 in HEX.
- EXCEPTION CODE
  It could be 0x01 or 02 in HEX

Table 2-8 exception codes

Value	Description
0x01	The function code is not an allowable action for the slave or the slave is in the wrong state.
0x02	The data address is not allowable address for the slave.

2.6 Using

2.6.1 Modbus/TCP Configuration

- CIE-M10/H10

  With ezManager, search the products and configure the values in the box ③.

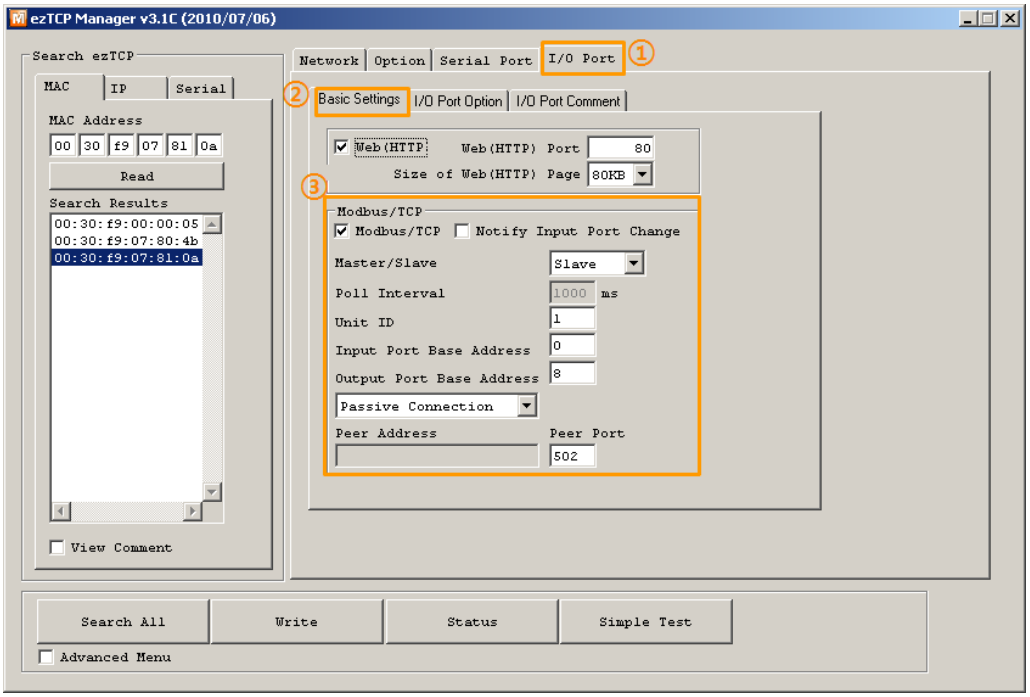


Figure 2-14 configuration for Modbus/TCP on ezManager

- EZI-10

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

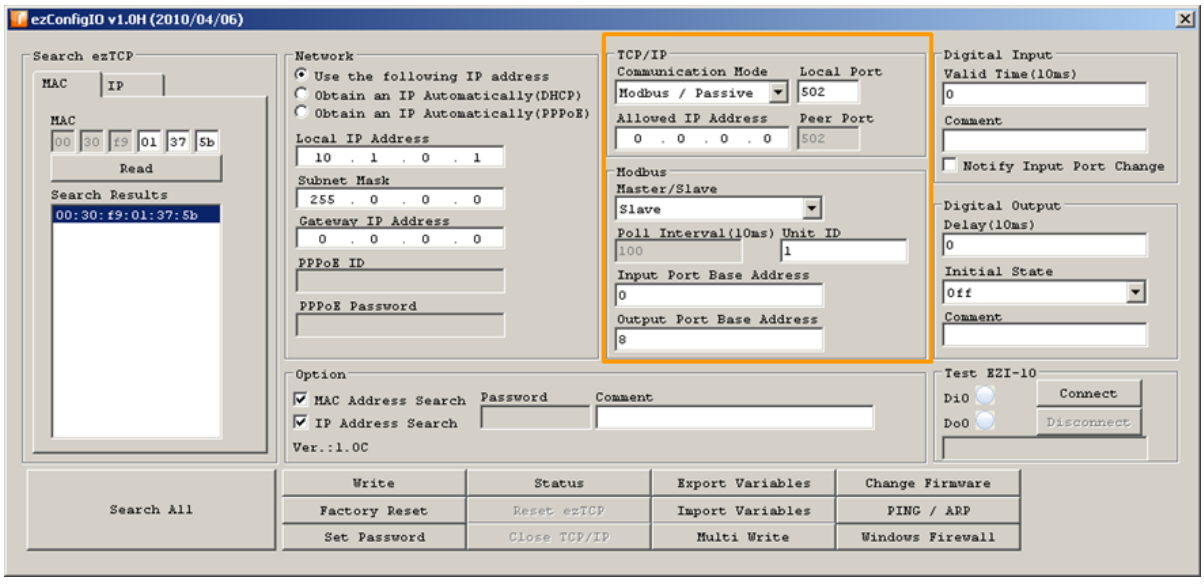


Figure 2-15 configuration for Modbus/TCP on ezConfigIO

- Parameters

Table 2-9 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
Active/Passive Connection	Active or Passive connection
Peer Address	Peer's IP address under active connection
Peer Port	Peer's port number

2.6.2 Configuration Example

Table 2-10 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
Peer Address	-	192.168.0.10
Peer Port	-	502
Local Port	502	-

2.7 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]>>[Technical Data]>>[Sample Codes] page on our web site.

- CModBusEngine
  The class which performs the Modbus/TCP

2.7.1 Functions

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

Table 2-11 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 2-12 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

3 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.

3.1 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.

3.2 Using

3.2.1 Configuration

- Serialized Modbus/TCP configuration

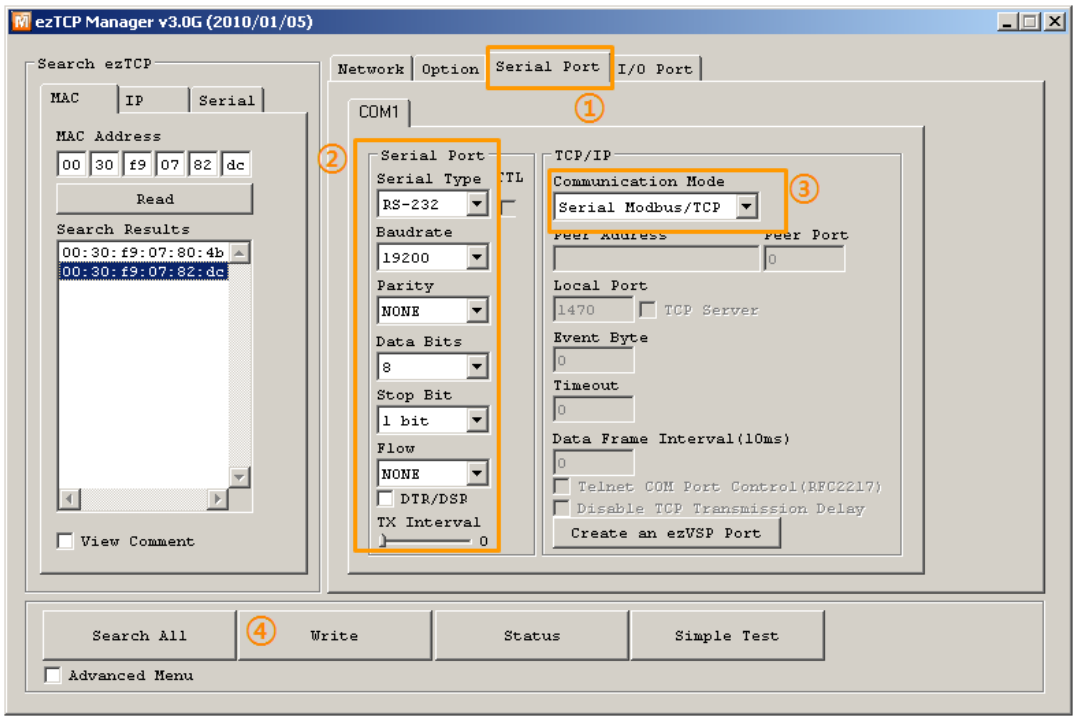


Fig 3-1 configuration of serialized Modbus/TCP

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

3.3 Trial Run

3.3.1 Preperations for Communication

For testing the serialized Modbus/TCP mode, design the connection like the below figure.

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

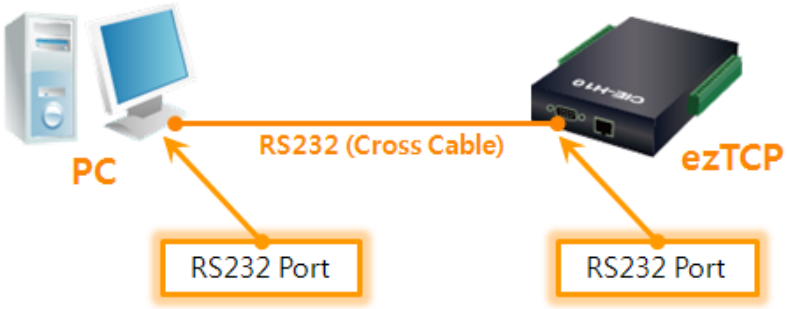


Fig 3-2 connection diagram

Then, keep the parameters as their default values referring to the below table.

Table 3-1 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

3.3.2 Sending an Example data

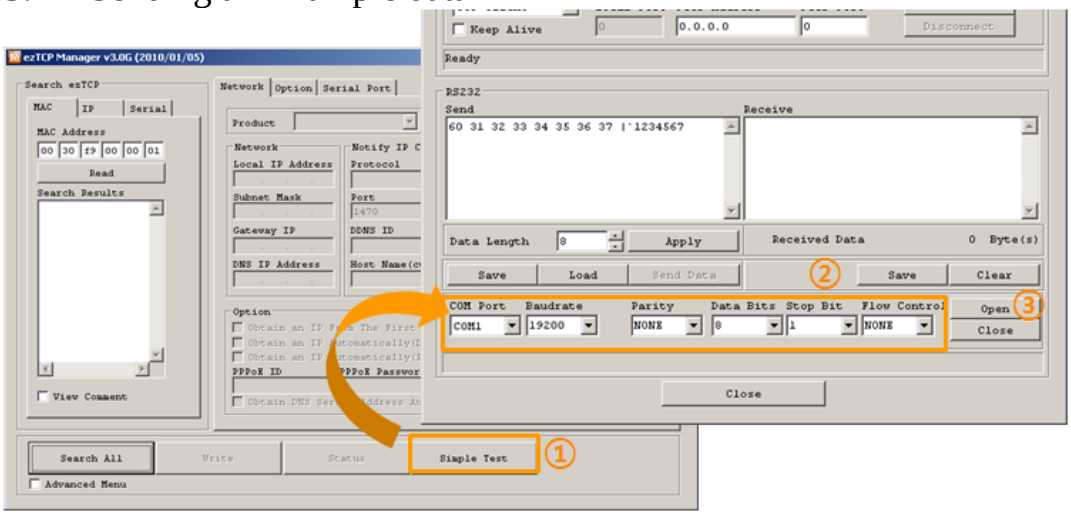


Fig 3-3 running Simple Test Program

① Clicking the [Simple Test] button.

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

③ Pressing the [Open] button.

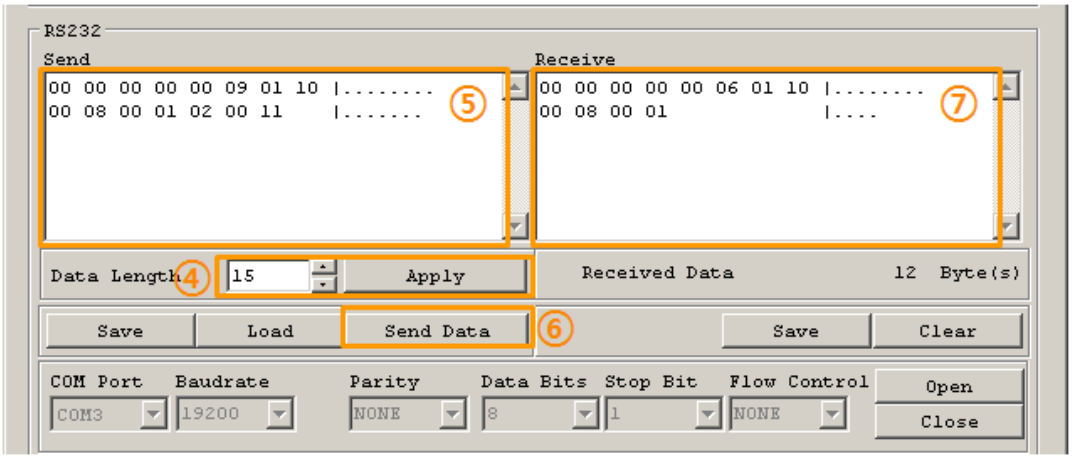


Fig 3-4 sending an example data

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

⑤ Input the data which is used in the Figure 2-10.

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 Figure 2-12.

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

4 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