WEINTEK LABS., INC.

Modbus TCP Server

Using ModbusServer function block to send CODESYS tag info to an external device

Demo Project

Contents

1. Overview

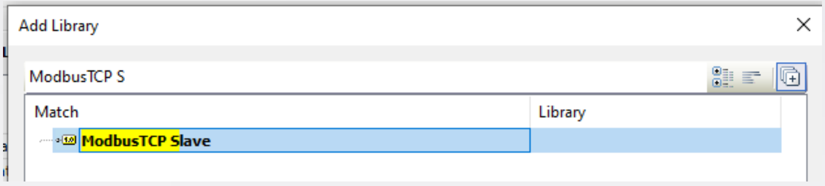
Modbus is an industrial protocol standard widely available for communication among controllers. Almost all controllers nowadays have built-in Modbus, and most of the devices used for simple purposes, such as digital meters, barcode scanners, or remote IO modules…etc., only support Modbus communication protocol.

ModbusTCP is defined as a master/slave protocol. The ModbusTCP Master (Client) can initiate communication and send requests to the ModbusTCP Slave (Server), and the ModbusTCP Slave (Server) must wait to be asked for information.

Weintek built-in CODESYS supports ModbusTCP Master (Client). For applications that require ModbusTCP Slave (Server), please see this demonstration about how to use ModbusServer function block.

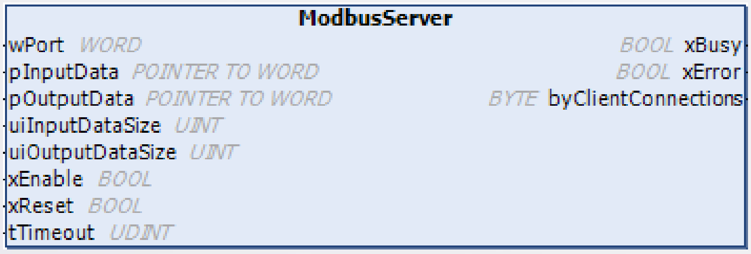
2. Library

Step 1. In CODESYS software add ModbusTCP Slave Library.



Function Block

ModbusServer:



Function: ModbusTCP Server starts processing requests.

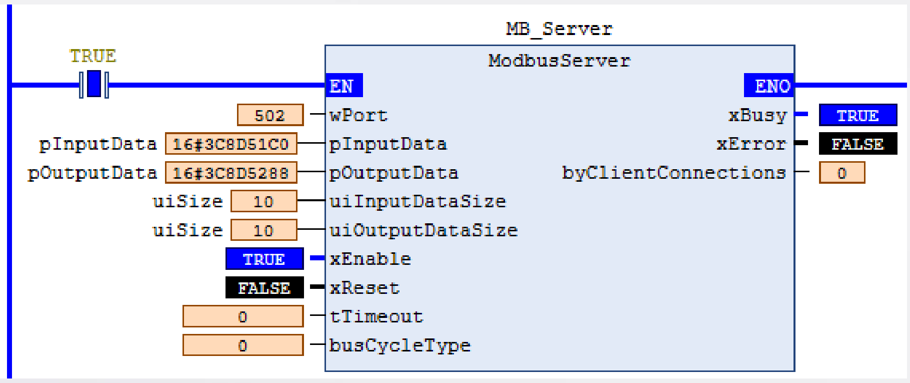
Input/Output	Type	Description
wPort	WORD	The port number for sending and receiving data. (Default = 502)
pInputData	POINTER TO WORD	Points to the map of the Modbus input register.
pOutputData	POINTER TO WORD	Points to the map of the Modbus holding register.
uiInputDataSize	UINT	Length of the input data.
uiOutputDataSize	UINT	Length of the output data.
xEnable	BOOL	Flag for starting request processing.
xReset	BOOL	Reset of status of Modbus server.
udiTimeOut	UDINT	Timeout (in ms). If it is not zero and no Modbus write error has occurred until the timeout, then the output data is set to zero.
xBusy	BOOL	TRUE= the request is in process.
xError	BOOL	Information about errors found.
byClientConnections	BYTE	Number of Modbus client connections.

3. Operation

Run the demo project: CODESYS_ModbusServer.

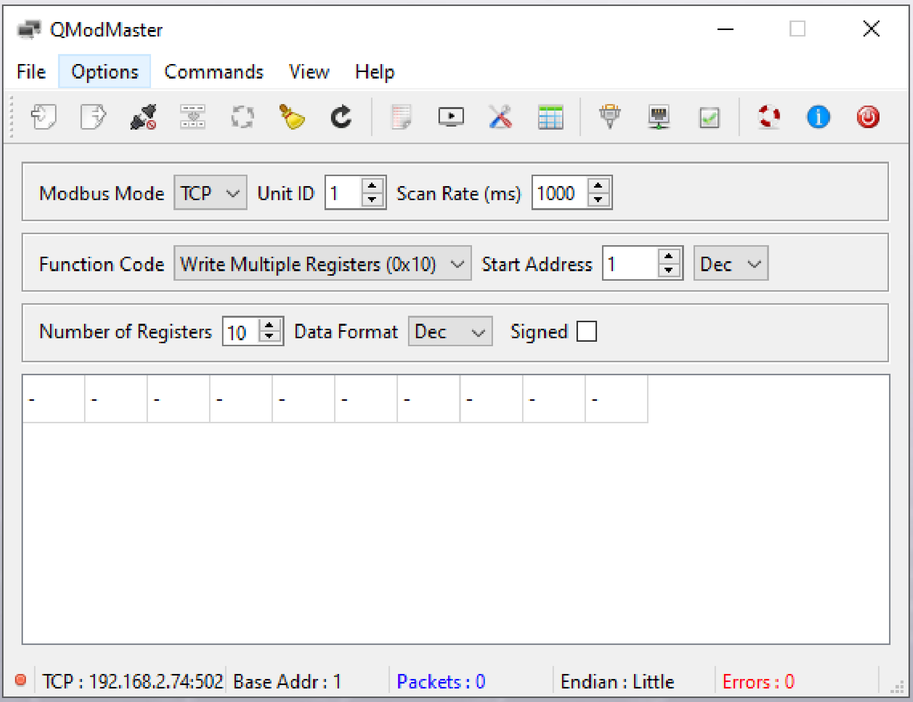
Write to holding register:

Step 1. pOutputData points to the address which returns value, uiSize cannot be 0, and ModbusServer.xEnable=TRUE.



MB_Server
ModbusServer
TRUE — EN
502 — wPort
pInputData 16#3C8D51C0 — pInputData
pOutputData 16#3C8D5288 — pOutputData
uiSize 10 — uiInputDataSize
uiSize 10 — uiOutputDataSize
TRUE — xEnable
FALSE — xReset
0 — tTimeout
0 — busCycleType
ENO
xBusy — TRUE
xError — FALSE
byClientConnections — 0

Step 2. Execute ModbusTCP test tool, set the IP address and port number of Weintek built-in CODESYS (same as ModbusServer.wPort), and then use function code 16. The number of registers should not exceed uiSize.

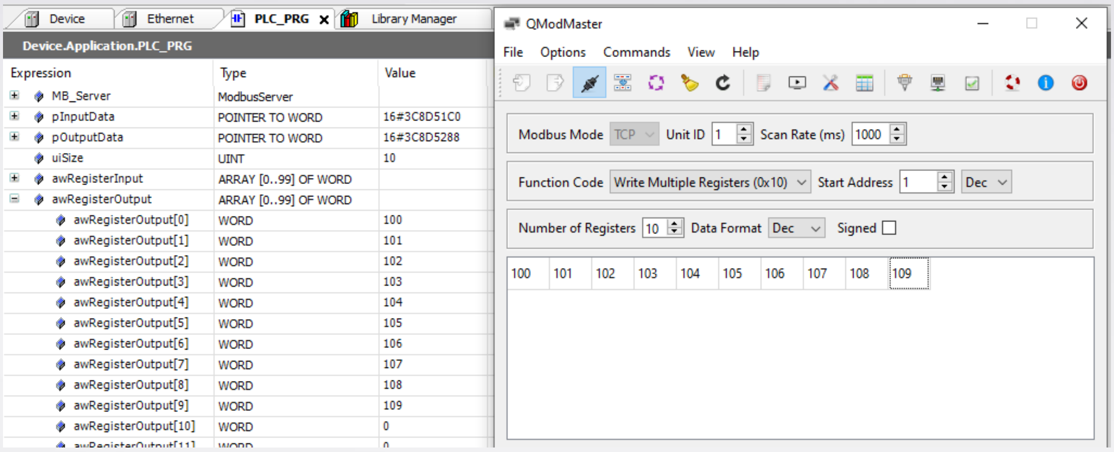


Modbus Mode: TCP
Unit ID: 1
Scan Rate (ms): 1000
Function Code: Write Multiple Registers (0x10)
Start Address: 1
Dec
Number of Registers: 10
Data Format: Dec
Signed ☐

-	-	-	-	-	-	-	-	-	-

TCP : 192.168.2.74:502 Base Addr : 1 Packets : 0 Endian : Little Errors : 0

Step 3. The value written in the test tool is sent to CODESYS's OutputData.



Device.Application.PLC_PRG

Expression	Type	Value
MB_Server	ModbusServer	
pInputData	POINTER TO WORD	16#3C8D51C0
pOutputData	POINTER TO WORD	16#3C8D5288
uiSize	UINT	10
awRegisterInput	ARRAY [0..99] OF WORD	
awRegisterOutput	ARRAY [0..99] OF WORD	
awRegisterOutput[0]	WORD	100
awRegisterOutput[1]	WORD	101
awRegisterOutput[2]	WORD	102
awRegisterOutput[3]	WORD	103
awRegisterOutput[4]	WORD	104
awRegisterOutput[5]	WORD	105
awRegisterOutput[6]	WORD	106
awRegisterOutput[7]	WORD	107
awRegisterOutput[8]	WORD	108
awRegisterOutput[9]	WORD	109
awRegisterOutput[10]	WORD	0

Modbus Mode TCP Unit ID 1 Scan Rate (ms) 1000

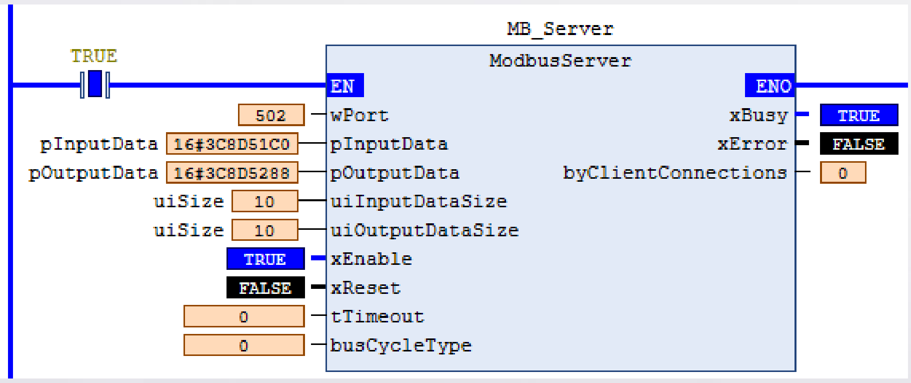
Function Code Write Multiple Registers (0x10) Start Address 1 Dec

Number of Registers 10 Data Format Dec Signed ☐

100	101	102	103	104	105	106	107	108	109

Read input / holding register:

Step 1. pInputData & pOutputData point to the address which returns value, uiSize cannot be 0, and ModbusServer.xEnable=TRUE.



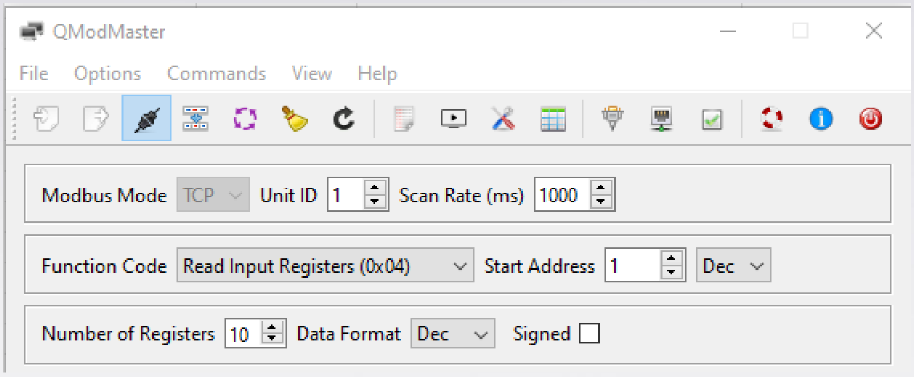
MB_Server
ModbusServer

TRUE — EN / ENO — TRUE

502 — wPort
pInputData 16#3C8D51C0 — pInputData
pOutputData 16#3C8D5288 — pOutputData
uiSize 10 — uiInputDataSize
uiSize 10 — uiOutputDataSize
TRUE — xEnable
FALSE — xReset
0 — tTimeout
0 — busCycleType

xBusy — TRUE
xError — FALSE
byClientConnections — 0

Step 2. Execute ModbusTCP test tool, set the IP address and port number of Weintek built-in CODESYS (same as ModbusServer.wPort), and then use function code 3 or 4. The number of registers should not exceed uiSize.

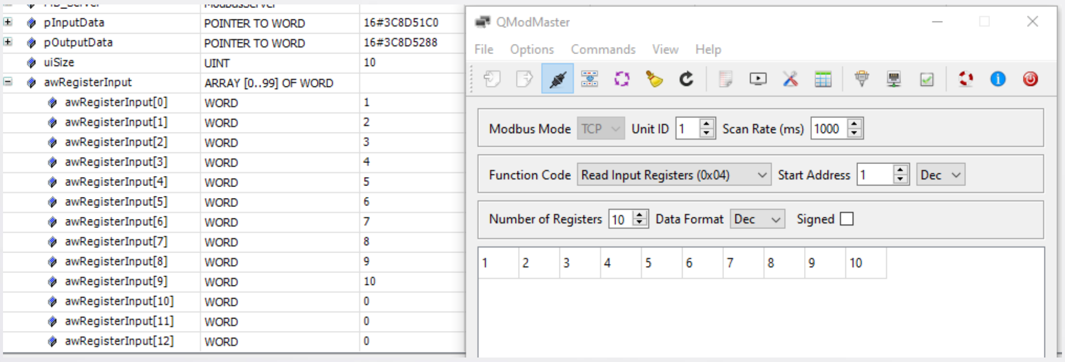


Modbus Mode TCP Unit ID 1 Scan Rate (ms) 1000

Function Code Read Input Registers (0x04) Start Address 1 Dec

Number of Registers 10 Data Format Dec Signed ☐

Step 3. The InputData tag value written in CODESYS can be obtained in the test tool.



pInputData	POINTER TO WORD	16#3C8D51C0
pOutputData	POINTER TO WORD	16#3C8D5288
uiSize	UINT	10
awRegisterInput	ARRAY [0..99] OF WORD	
awRegisterInput[0]	WORD	1
awRegisterInput[1]	WORD	2
awRegisterInput[2]	WORD	3
awRegisterInput[3]	WORD	4
awRegisterInput[4]	WORD	5
awRegisterInput[5]	WORD	6
awRegisterInput[6]	WORD	7
awRegisterInput[7]	WORD	8
awRegisterInput[8]	WORD	9
awRegisterInput[9]	WORD	10
awRegisterInput[10]	WORD	0
awRegisterInput[11]	WORD	0
awRegisterInput[12]	WORD	0

Modbus Mode TCP Unit ID 1 Scan Rate (ms) 1000

Function Code Read Input Registers (0x04) Start Address 1 Dec

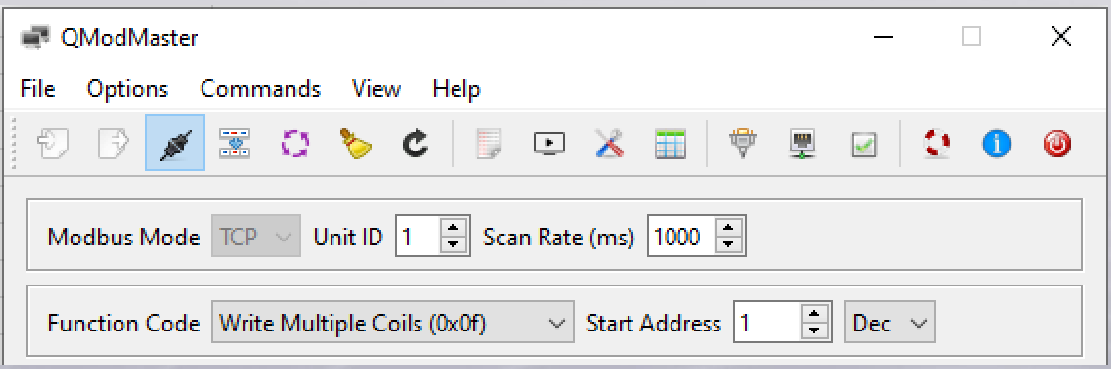
Number of Registers 10 Data Format Dec Signed ☐

1	2	3	4	5	6	7	8	9	10

Write coils:

Step 1. pOutputData points to the address which returns value, uiSize cannot be 0, and ModbusServer.xEnable=TRUE.

Step 2. Execute ModbusTCP test tool, set the IP address and port number of Weintek built-in CODESYS (same as ModbusServer.wPort), and then use function code 15. The number of coils should not exceed uiSize.

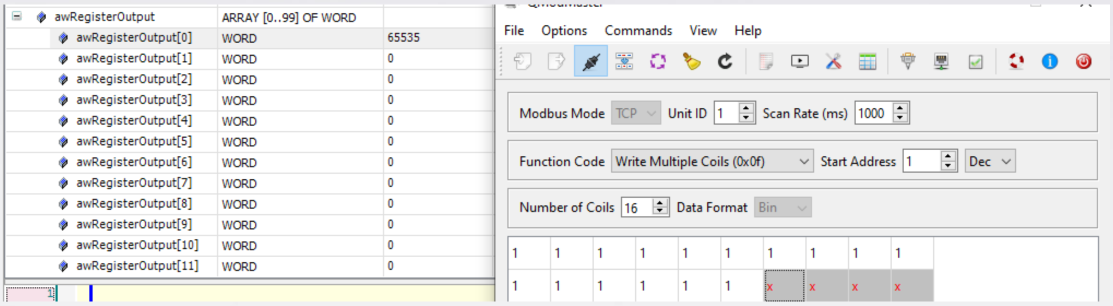


Modbus Mode TCP Unit ID 1 Scan Rate (ms) 1000

Function Code Write Multiple Coils (0x0f) Start Address 1 Dec

Step 3. The value written in the test tool is sent to CODESYS's OutputData.

Only the value in the first register will change accordingly (16bit = 1word).



awRegisterOutput	ARRAY [0..99] OF WORD	
awRegisterOutput[0]	WORD	65535
awRegisterOutput[1]	WORD	0
awRegisterOutput[2]	WORD	0
awRegisterOutput[3]	WORD	0
awRegisterOutput[4]	WORD	0
awRegisterOutput[5]	WORD	0
awRegisterOutput[6]	WORD	0
awRegisterOutput[7]	WORD	0
awRegisterOutput[8]	WORD	0
awRegisterOutput[9]	WORD	0
awRegisterOutput[10]	WORD	0
awRegisterOutput[11]	WORD	0

Modbus Mode TCP Unit ID 1 Scan Rate (ms) 1000

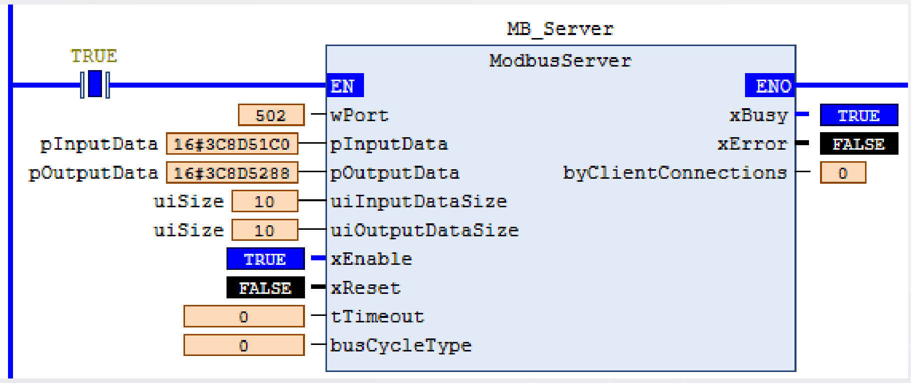
Function Code Write Multiple Coils (0x0f) Start Address 1 Dec

Number of Coils 16 Data Format Bin

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	x	x	x	x

Read discrete inputs / coils:

Step 1. pInputData & pOutputData point to the address which returns value, uiSize cannot be 0, and ModbusServer.xEnable=TRUE.



MB_Server

ModbusServer

TRUE — EN / ENO

502 — wPort

pInputData 16#3C8D51C0 — pInputData

pOutputData 16#3C8D5288 — pOutputData

uiSize 10 — uiInputDataSize

uiSize 10 — uiOutputDataSize

TRUE — xEnable

FALSE — xReset

0 — tTimeout

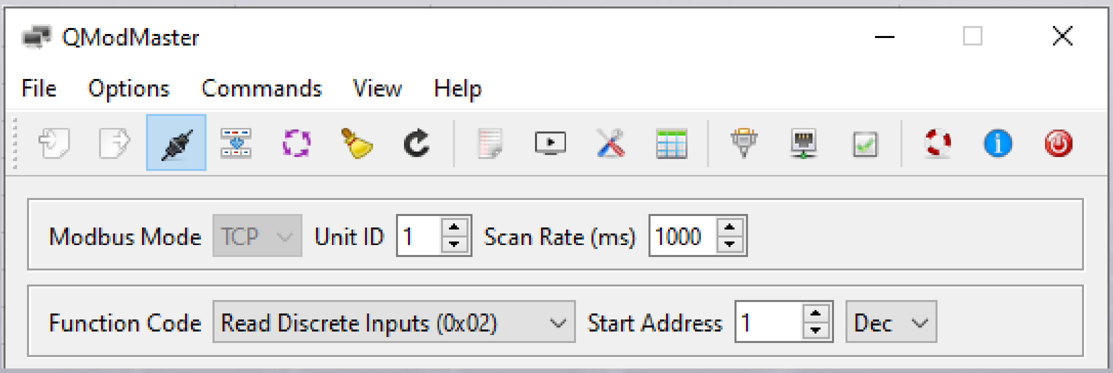
0 — busCycleType

xBusy — TRUE

xError — FALSE

byClientConnections — 0

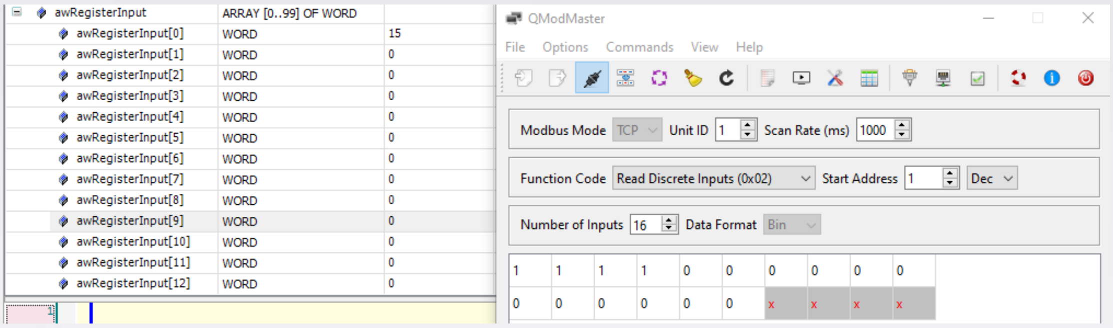
Step 2. Execute ModbusTCP test tool, set the IP address and port number of Weintek built-in CODESYS (same as ModbusServer.wPort), and then use function code 1 or 2. The number of coils should not exceed uiSize.



Modbus Mode TCP Unit ID 1 Scan Rate (ms) 1000

Function Code Read Discrete Inputs (0x02) Start Address 1 Dec

Step 3. The InputData tag value written in CODESYS can be obtained in the test tool.



awRegisterInput	ARRAY [0..99] OF WORD	
awRegisterInput[0]	WORD	15
awRegisterInput[1]	WORD	0
awRegisterInput[2]	WORD	0
awRegisterInput[3]	WORD	0
awRegisterInput[4]	WORD	0
awRegisterInput[5]	WORD	0
awRegisterInput[6]	WORD	0
awRegisterInput[7]	WORD	0
awRegisterInput[8]	WORD	0
awRegisterInput[9]	WORD	0
awRegisterInput[10]	WORD	0
awRegisterInput[11]	WORD	0
awRegisterInput[12]	WORD	0

Modbus Mode: TCP
Unit ID: 1
Scan Rate (ms): 1000
Function Code: Read Discrete Inputs (0x02)
Start Address: 1 Dec
Number of Inputs: 16
Data Format: Bin

1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	x	x	x	x