Modbus POE ETH Relay 30CH

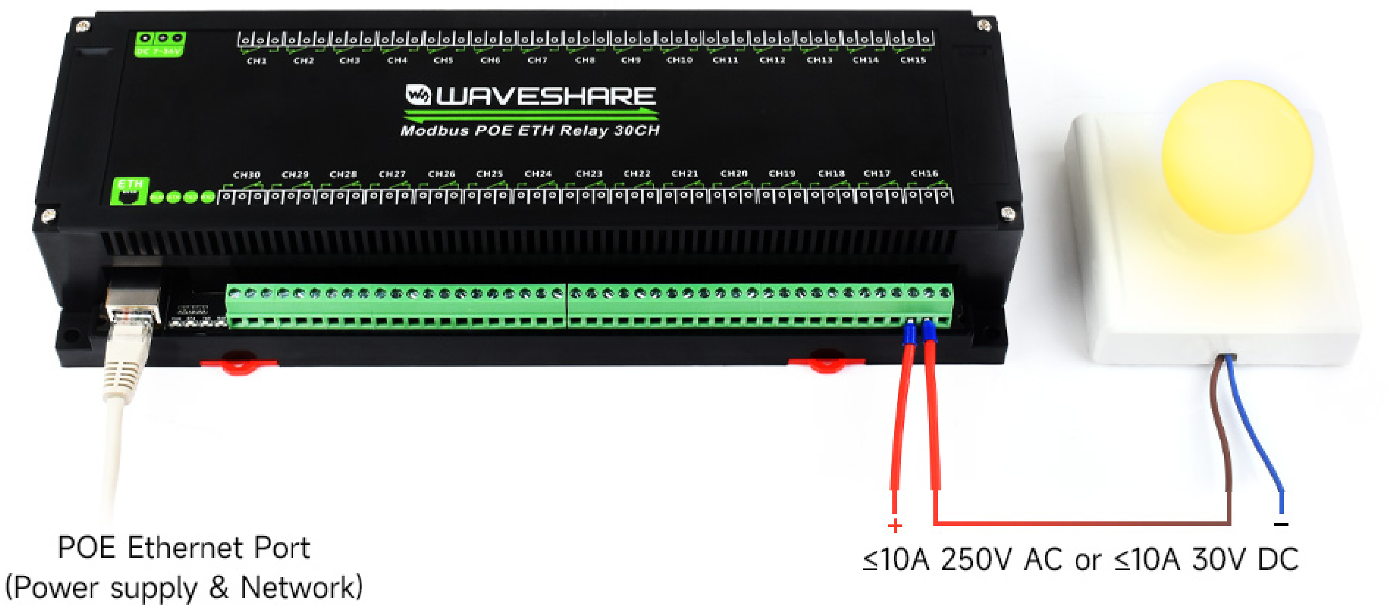
Overview

- This product is an industrial 30-CH relay module controlled via Ethernet port, using Modbus RTU/Modbus TCP protocol, support PoE Ethernet port power supply, and ABS case design. This product is easy to operate, with fast communication speed, stability, reliability, security and other characteristics, can be applied to a variety of high communication requirements of industrial control equipment or applications.

Hardware Description

Hardware Connection

- Connects Modbus POE ETH Relay 30CH to the LAN via a network cable, powered through the power port or via POE.



Note: The above picture is using a PoE port for power supply. If the network cable you used is a normal one, you need to connect an external power adapter to supply power in the range of 7~36V.

Electrical Safety Precautions

- This product must be operated and used by professional electricians or technical personnel. During use, please ensure electrical safety and take measures to prevent leakage and insulation.
- Before installing, maintaining, or replacing relay equipment, be sure to turn off the power and unplug the plug.
- Do not attempt to disassemble relay equipment to avoid damaging the equipment or causing the risk of electric shock.
- Please install and place the relay equipment product properly. Do not use it in damp, overheated, or flammable environments to avoid safety incidents caused by improper installation or use.

Specification

Power Supply	PoE Ethernet Port, DC 5.5*2.1 power port or power supply screw terminal (7~36V)
Communication Interface	PoE Ethernet Port, Support IEEE 802.3af Standard
Relay Channels	8 Channels
Touch Contact	1NO, 1NC
Touch Contact Load	≤10A 250V AC Or ≤10A 30V DC
Modbus Protocol	Modbus RTU Protocol Or Modbus TCP Protocol

Indicator Note

Indicator	Status Description
RUN	Ethernet operation indicator. After the chip is working properly, it will output a square wave with a period of 2 seconds.
STA	MCU indicator, blinking when the MCU is operating normally.
TXD	Transmitting indicator, light is on when sending data.

RXD	Receiving indicator, light is on when receiving data.
Ethernet Port Green Light	The green light is on when a TCP connection is established and can be used to determine if the module has established a communication link with the host computer software.
Ethernet Port Yellow Light	Data transmission light, when there is data transmission on the network port, the yellow light transition state will change, which can be used to determine whether there is data transmission.

Module Parameter Configuration

The module needs to set the module parameters before communication, such as IP address, serial port format, Modbus protocol, etc. There are two modes of setting parameters: Vircom software configuration and web configuration.

Vircom software configuration allows for setting more parameters, but requires software installation. Web configuration does not require installation, but you need to know the IP address first, and the configuration parameters are few. It is recommended to use Virom for configuration.

Note:

1. The configuration can be done in any way, and it is recommended to use Virom software for first test.
2. It is recommended to modify only the IP address for the first configuration, other parameters are not recommended to be modified. The serial port parameters must be the default parameters; modifying the serial port parameters will result in no communication.
3. The module supports both Modbus RTU and Modbus TCP protocols. In the Advanced Settings -> Transfer Protocol, you can choose "None", which means the Modbus RTU protocol. It is not recommended to modify during the first configuration.
4. The selected Modbus TCP protocol must be configured using the Virom software and set to a non-storage Modbus gateway, otherwise the communication will not be normal.

Virom Software Mode Configuration

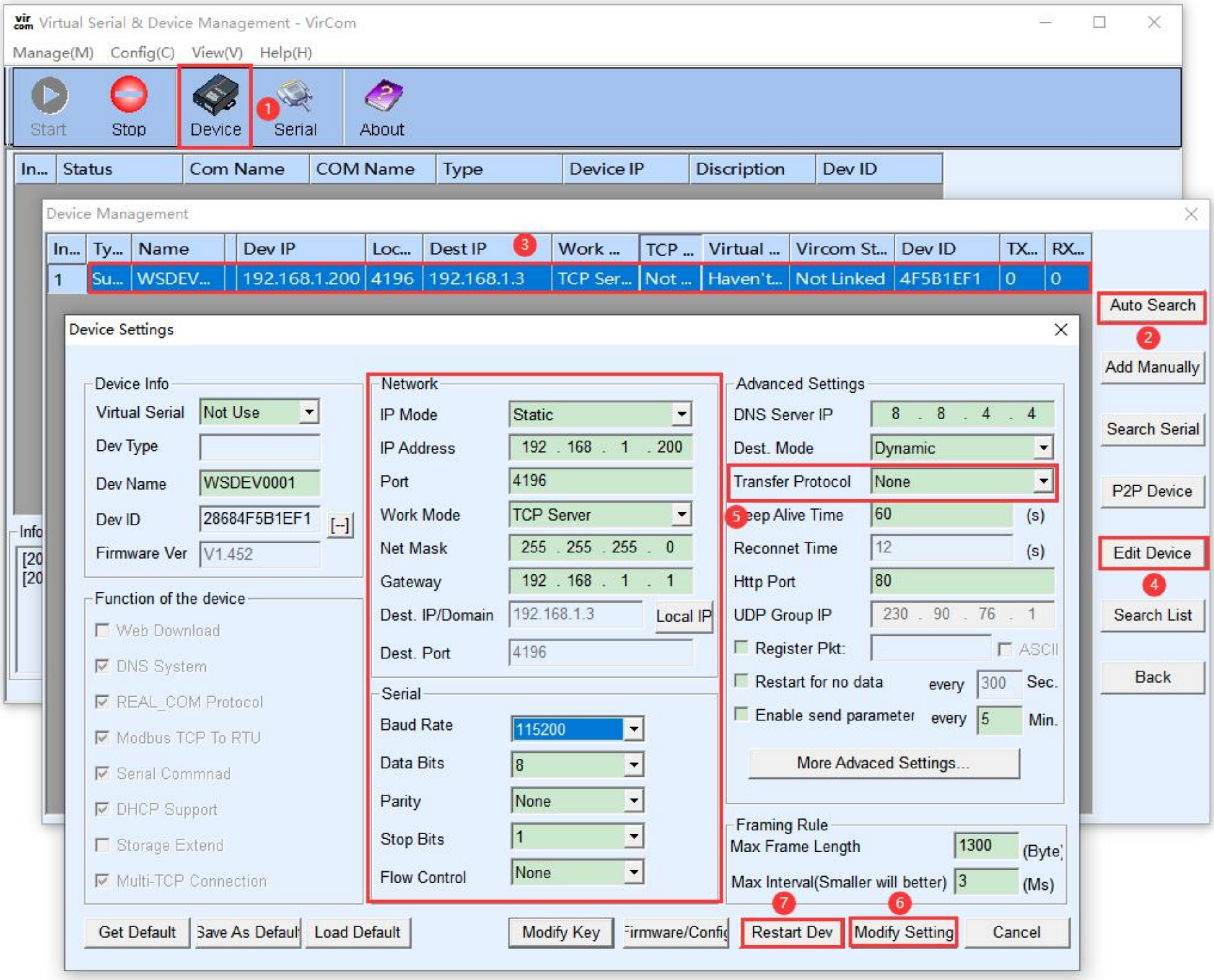
General Settings

Connect the module to the hardware and connect it to the network. Run the VirCom software (the computer on which Vircom is installed must be on the same

LAN as the module).
The operation is as follows:

- 1. Click `Device`
- 2. Click `Auto Search`
- 3. Software search recognizes the device connected to the LAN
- 4. Select the device, and then click `Edit Device` or double-click the searched device directly
- 5. Set up the device parameters:
    - Modify the "IP mode" to a static assigned address, set the IP address, note that the static IP address entered must not be used by other devices, and it needs to be on the same LAN as the computer.
    - The working mode is TCP server. The serial port setting is 115200 by default and cannot be modified.
    - The "Transfer Protocol" in "Advanced Settings" defaults to "None", which means using the Modbus RTU protocol; if you select "Modbus TCP protocol", then use the Modbus TCP communication protocol.
    - Click on "More Advanced Settings..." and select the Modbus Gateway Type as a non-storage Modbus gateway.
- 6. Once the settings are complete, click `Modify Setting`
- 7. Click `Restart Dev`, wait for the module to restart, and the new settings will take effect.

Note: It is recommended to modify only the IP address for the first configuration, and do not modify other parameters.
See the figure below for details:



In...	Ty...	Name	Dev IP	Loc...	Dest IP	Work ...	TCP ...	Virtual ...	Vircom St...	Dev ID	TX...	RX...
1	Su...	WSDEV...	192.168.1.200	4196	192.168.1.3	TCP Ser...	Not ...	Haven't...	Not Linked	4F5B1EF1	0	0

Device Info

- Virtual Serial: Not Use
- Dev Type:
- Dev Name: WSDEV0001
- Dev ID: 28684F5B1EF1
- Firmware Ver: V1.452

Function of the device

- ☐ Web Download
- ☑ DNS System
- ☑ REAL_COM Protocol
- ☑ Modbus TCP To RTU
- ☑ Serial Commnad
- ☑ DHCP Support
- ☐ Storage Extend
- ☑ Multi-TCP Connection

Network

- IP Mode: Static
- IP Address: 192 . 168 . 1 . 200
- Port: 4196
- Work Mode: TCP Server
- Net Mask: 255 . 255 . 255 . 0
- Gateway: 192 . 168 . 1 . 1
- Dest. IP/Domain: 192.168.1.3 Local IP
- Dest. Port: 4196

Serial

- Baud Rate: 115200
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: None

Advanced Settings

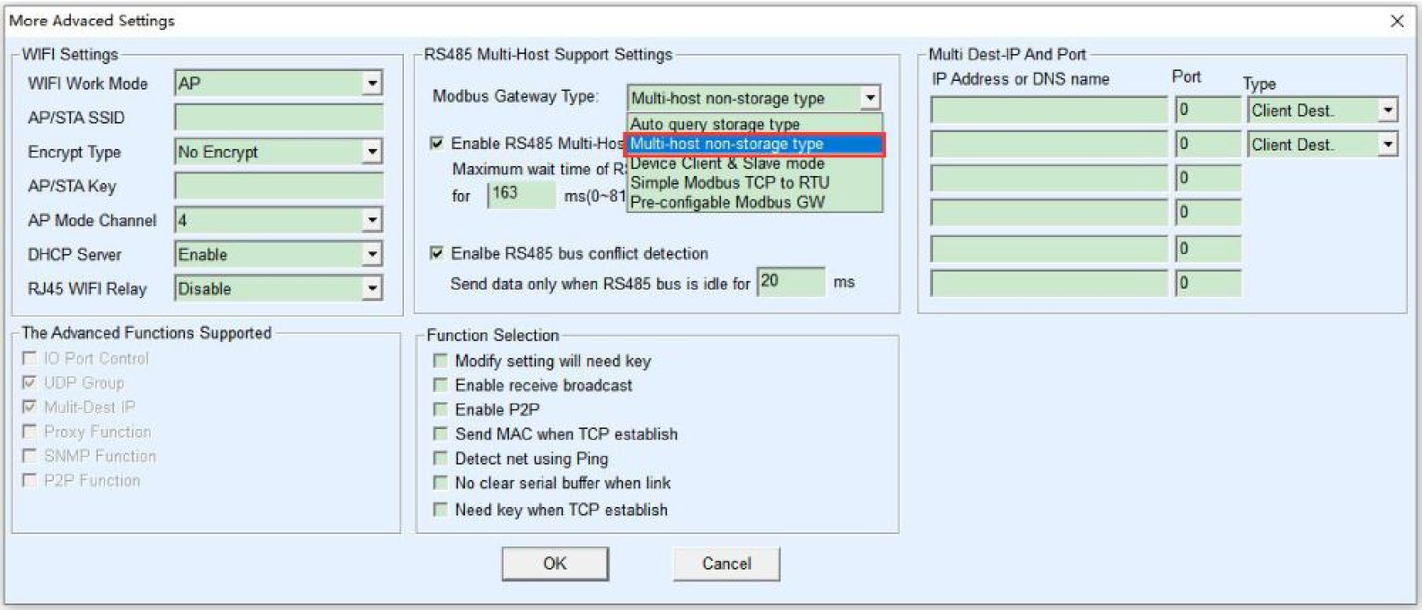
- DNS Server IP: 8 . 8 . 4 . 4
- Dest. Mode: Dynamic
- Transfer Protocol: None
- Keep Alive Time: 60 (s)
- Reconnet Time: 12 (s)
- Http Port: 80
- UDP Group IP: 230 . 90 . 76 . 1
- ☐ Register Pkt: ☐ ASCII
- ☐ Restart for no data every 300 Sec.
- ☐ Enable send parameter every 5 Min.
- More Advaced Settings...

Framing Rule

- Max Frame Length: 1300 (Byte)
- Max Interval(Smaller will better): 3 (Ms)

Note: The default Modbus gateway type is storage type, which will automatically send query commands several times, which may cause the controller chip to fail to respond, resulting in no response to the query commands. Therefore, you need to set it as Multi-host non-storage

type.



WIFI Settings

- WIFI Work Mode: AP
- AP/STA SSID:
- Encrypt Type: No Encrypt
- AP/STA Key:
- AP Mode Channel: 4
- DHCP Server: Enable
- RJ45 WIFI Relay: Disable

RS485 Multi-Host Support Settings

- Modbus Gateway Type: Multi-host non-storage type
  - Auto query storage type
  - Multi-host non-storage type
  - Device Client & Slave mode
  - Simple Modbus TCP to RTU
  - Pre-configable Modbus GW
- ☑ Enable RS485 Multi-Hos
- Maximum wait time of R
- for 163 ms(0~81
- ☑ Enalbe RS485 bus conflict detection
- Send data only when RS485 bus is idle for 20 ms

Multi Dest-IP And Port

IP Address or DNS name	Port	Type
	0	Client Dest.
	0	Client Dest.
	0	
	0	
	0	
	0	

The Advanced Functions Supported

- ☐ IO Port Control
- ☑ UDP Group
- ☑ Mulit-Dest IP
- ☐ Proxy Function
- ☐ SNMP Function
- ☐ P2P Function

Function Selection

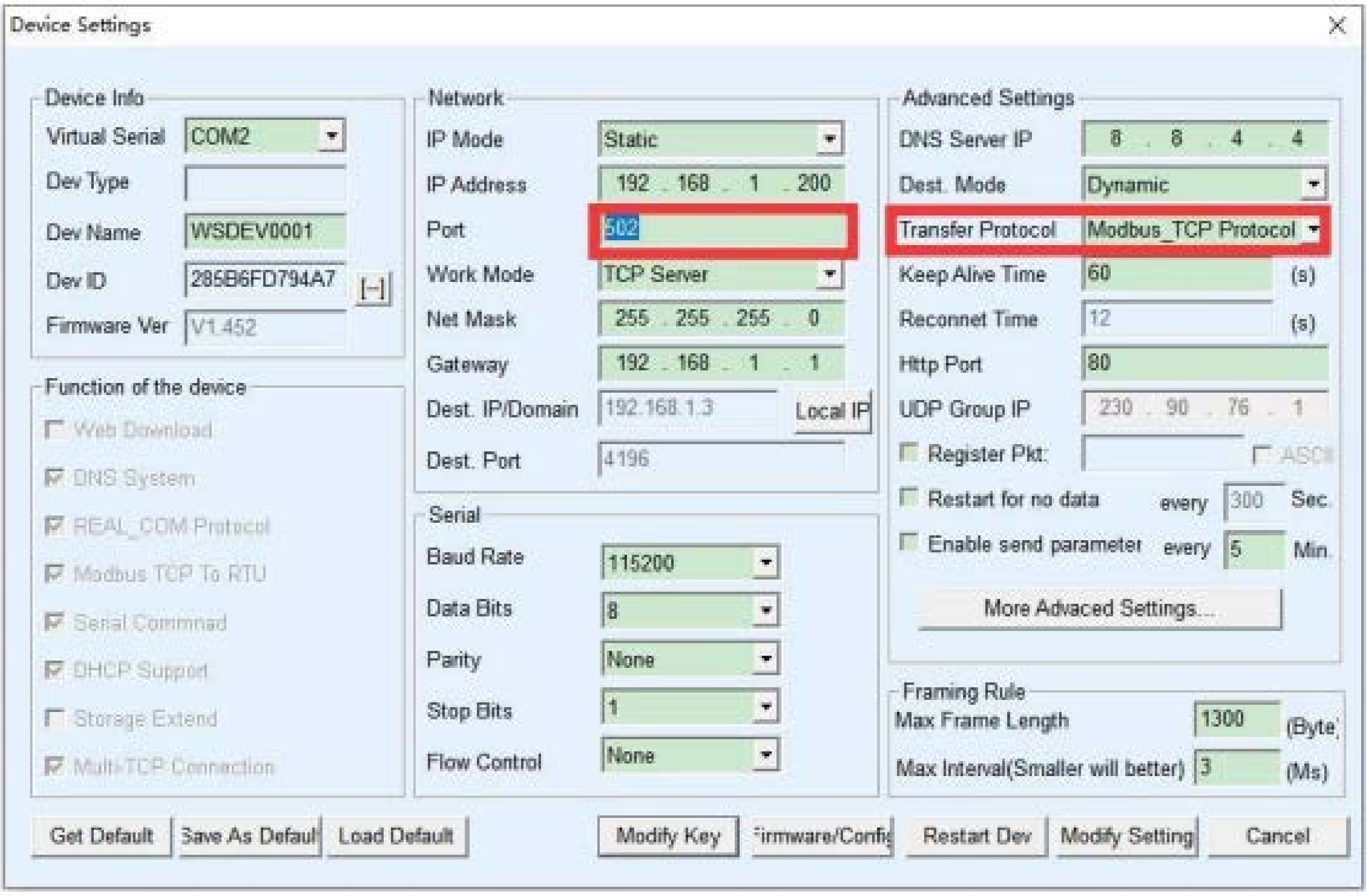
- ☐ Modify setting will need key
- ☐ Enable receive broadcast
- ☐ Enable P2P
- ☐ Send MAC when TCP establish
- ☐ Detect net using Ping
- ☐ No clear serial buffer when link
- ☐ Need key when TCP establish

Protocol Setting

Note: It is recommended to use the default Modbus RTU protocol for the first configuration and no modifications are needed.
Although the module transmits data through the network port, it supports two Modbus protocols: Modbus RTU and Modbus TCP. By default, data is transparently transmitted, i.e. using the Modbus RTU protocol.

Modbus TCP Protocol Settings

- The "Transfer Protocol" in the "Advanced Settings" can be set to "Modbus TCP protocol". In this case, the Modbus RTU protocol of the main controller will be converted to the Modbus TCP protocol and transmitted through the network port.
- In this case, the device port automatically changes to 502. Users can use the Modbus TCP tool to connect to the IP port 502 of the serial port server.
- Click on "More Advanced Settings..." and select the Modbus Gateway Type as a non-storage Modbus gateway.



Device Settings

Device Info
Virtual Serial: COM2
Dev Type:
Dev Name: WSDEV0001
Dev ID: 285B6FD794A7
Firmware Ver: V1.452

Function of the device
□ Web Download
☑ DNS System
☑ REAL_COM Protocol
☑ Modbus TCP To RTU
☑ Serial Commnad
☑ DHCP Support
□ Storage Extend
☑ Multi-TCP Connection

Network
IP Mode: Static
IP Address: 192 . 168 . 1 . 200
Port: 502
Work Mode: TCP Server
Net Mask: 255 . 255 . 255 . 0
Gateway: 192 . 168 . 1 . 1
Dest. IP/Domain: 192.168.1.3
Dest. Port: 4196

Serial
Baud Rate: 115200
Data Bits: 8
Parity: None
Stop Bits: 1
Flow Control: None

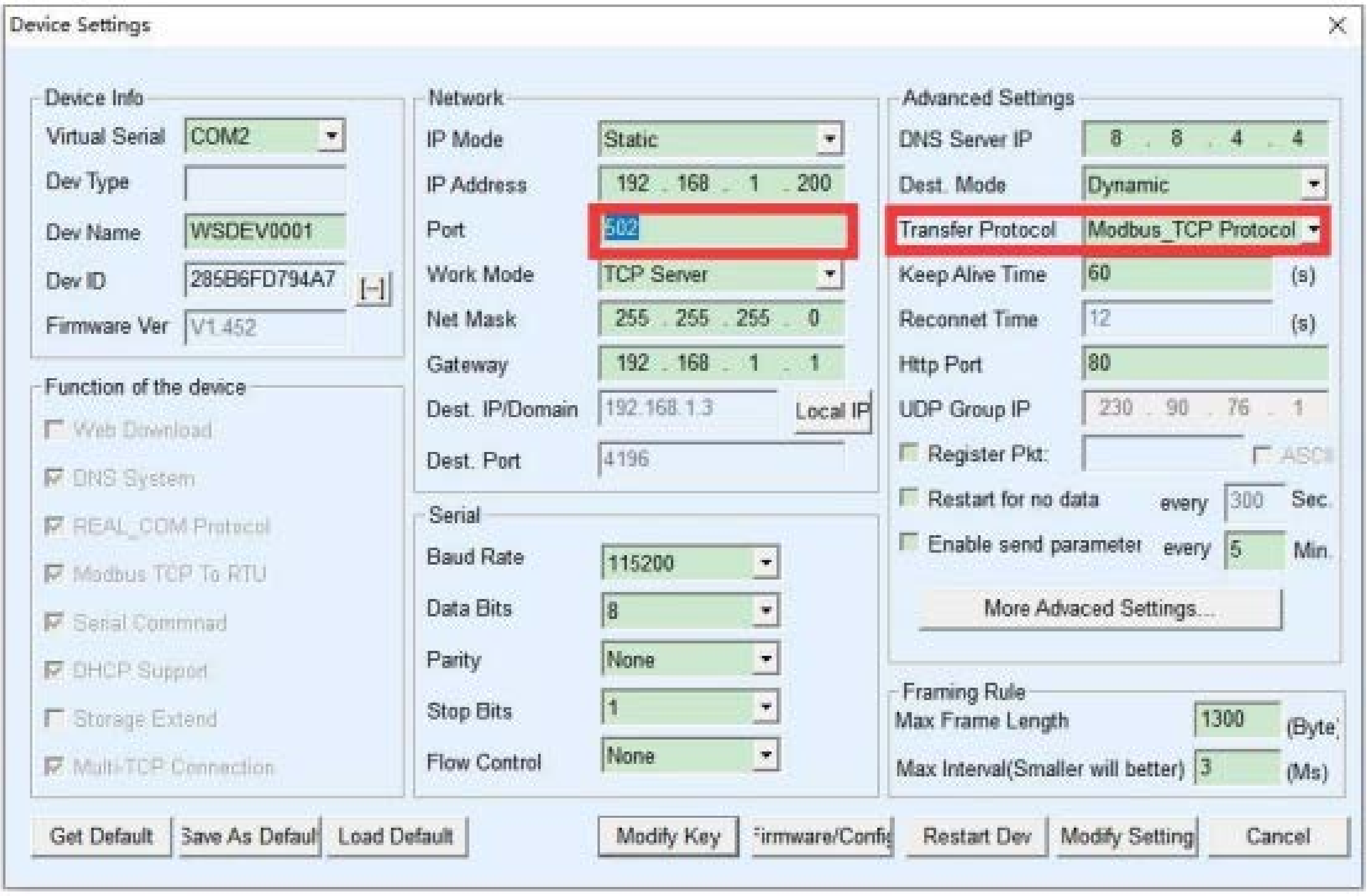
Advanced Settings
DNS Server IP: 8 . 8 . 4 . 4
Dest. Mode: Dynamic
Transfer Protocol: Modbus_TCP Protocol
Keep Alive Time: 60 (s)
Reconnet Time: 12 (s)
Http Port: 80
UDP Group IP: 230 . 90 . 76 . 1
□ Register Pkt: □ ASCII
□ Restart for no data every 300 Sec.
□ Enable send parameter every 5 Min.

Framing Rule
Max Frame Length: 1300 (Byte)
Max Interval(Smaller will better): 3 (Ms)

Modbus RTU Protocol Settings

- Set "Transfer Protocol" in the "Advanced Settings" to "None", and change to use Modbus RTU protocol.
- Click on "More Advanced Settings..." and select the Modbus Gateway Type as a non-storage Modbus gateway.

Note: The default Modbus gateway type is storage type, which will automatically send query commands several times, which may cause the controller chip to fail to respond, resulting in no response to the query commands. Therefore, you need to set it as Multi-host non-storage
type.



Device Settings

Device Info
Virtual Serial: COM2
Dev Type:
Dev Name: WSDEV0001
Dev ID: 285B6FD794A7
Firmware Ver: V1.452

Function of the device
☐ Web Download
☑ DNS System
☑ REAL_COM Protocol
☑ Modbus TCP To RTU
☑ Serial Commnad
☑ DHCP Support
☐ Storage Extend
☑ Multi-TCP Connection

Network
IP Mode: Static
IP Address: 192 . 168 . 1 . 200
Port: 502
Work Mode: TCP Server
Net Mask: 255 . 255 . 255 . 0
Gateway: 192 . 168 . 1 . 1
Dest. IP/Domain: 192.168.1.3
Dest. Port: 4196

Serial
Baud Rate: 115200
Data Bits: 8
Parity: None
Stop Bits: 1
Flow Control: None

Advanced Settings
DNS Server IP: 8 . 8 . 4 . 4
Dest. Mode: Dynamic
Transfer Protocol: Modbus_TCP Protocol
Keep Alive Time: 60 (s)
Reconnect Time: 12 (s)
Http Port: 80
UDP Group IP: 230 . 90 . 76 . 1
☐ Register Pkt: ☐ ASCII
☐ Restart for no data every 300 Sec.
☐ Enable send parameter every 5 Min.

Framing Rule
Max Frame Length: 1300 (Byte)
Max Interval(Smaller will better): 3 (Ms)

Virtual Serial Port Setting

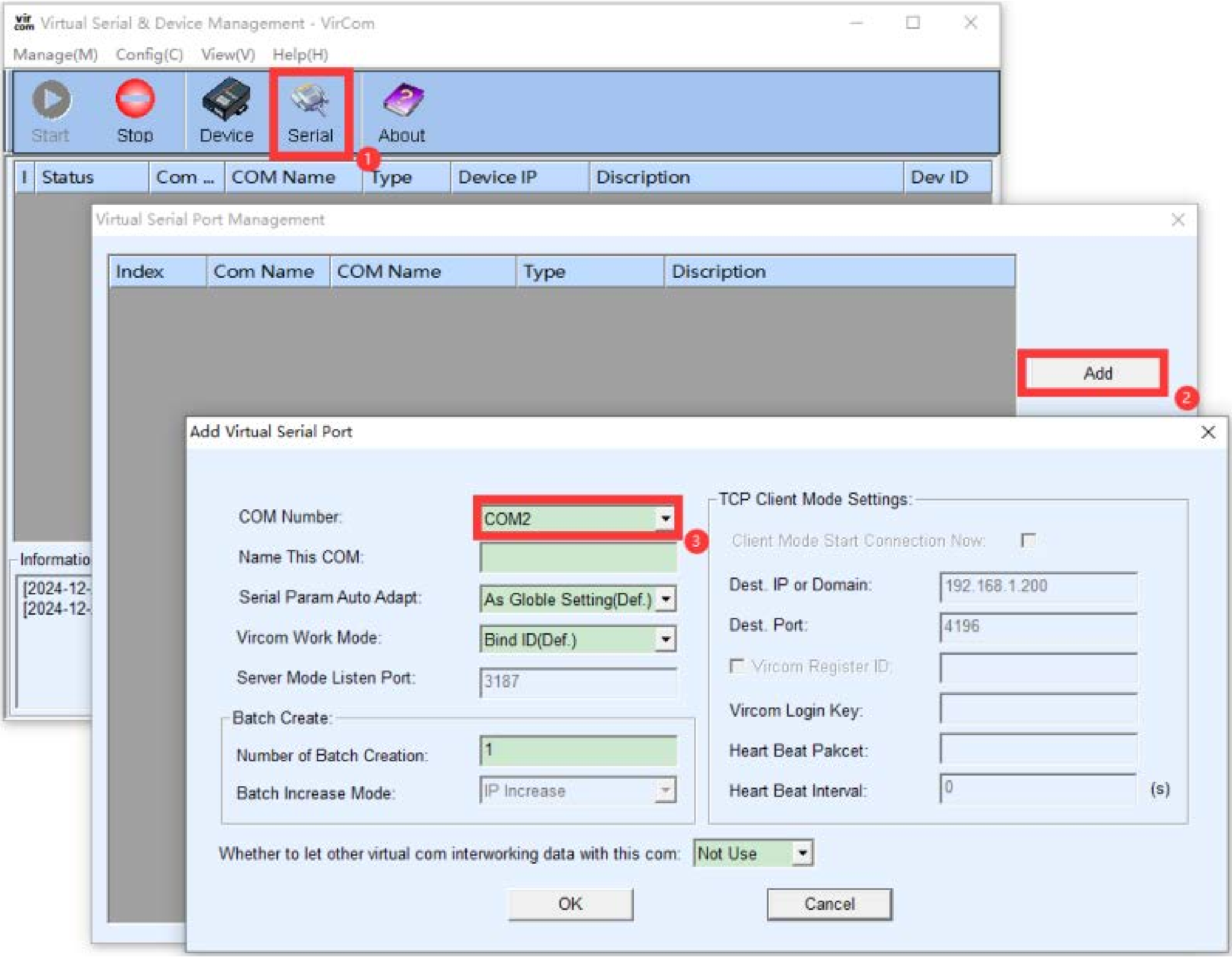
The module transmits data through a network port (TCP/UDP protocol). In order to enable users to use the PoE port communication even with developed serial port software, a virtual serial port needs to be added. If not needed, this part can be skipped.

- First, install the virtual serial driver Virtual serial port driver
- Run Vircom and the user program on the same computer.
- Vircom virtualize a COM port, mapping this COM port to a network port and connecting it to a serial port server. When the user program uses the COM

communication, it can send data to the user's serial port device through the Vircom serial port server.

The following steps demonstrate this operation:

- Click on "Serial Port & Device Management" on the Vircom main interface, then click "Add" and select to add COM2 (Among them, COM2 is the newly emerging COM port on the computer).


Virtual Serial & Device Management - VirCom
Manage(M) Config(C) View(V) Help(H)
Start
Stop
Device
Serial
About
1
I | Status | Com ... | COM Name | Type | Device IP | Discription | Dev ID
Informatio
[2024-12-
[2024-12-
Virtual Serial Port Management
Index | Com Name | COM Name | Type | Discription
Add
2
Add Virtual Serial Port
3
OK
Cancel

COM Number: COM2

Name This COM:

Serial Param Auto Adapt: As Globle Setting(Def.)

Vircom Work Mode: Bind ID(Def.)

Server Mode Listen Port: 3187

Batch Create:

Number of Batch Creation: 1

Batch Increase Mode: IP Increase

TCP Client Mode Settings:

Client Mode Start Connection Now: □

Dest. IP or Domain: 192.168.1.200

Dest. Port: 4196

□ Vircom Register ID:

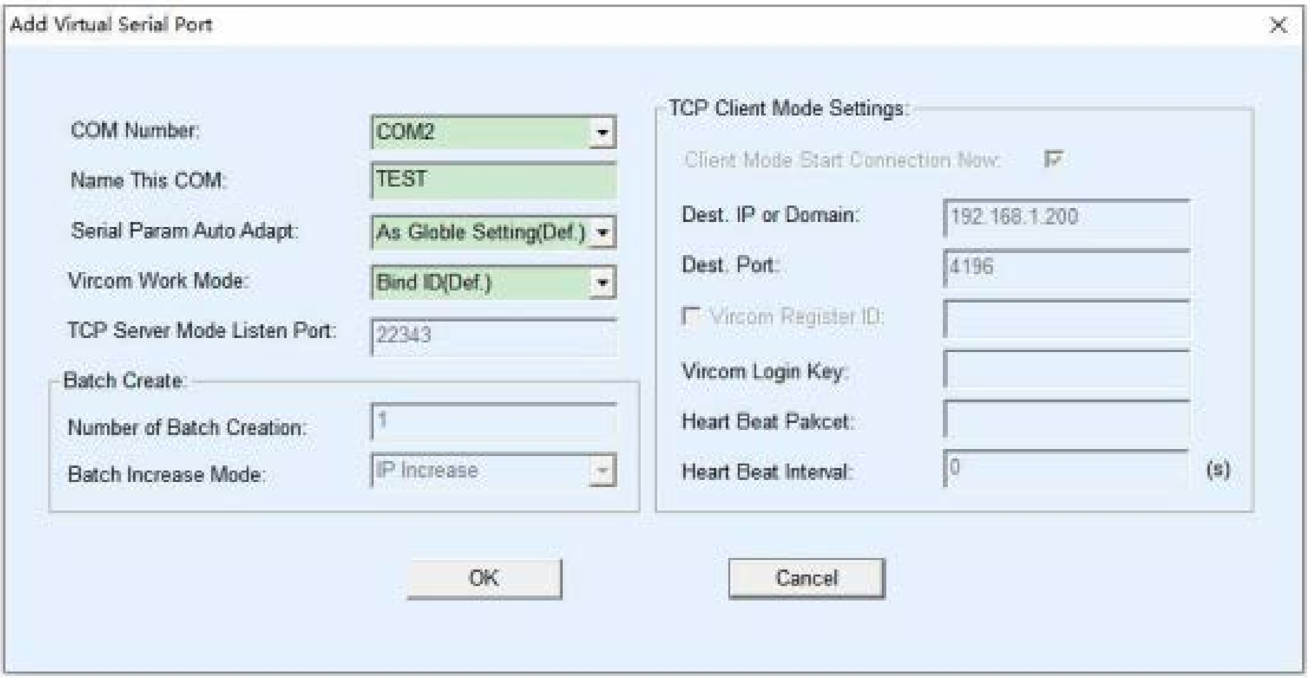
Vircom Login Key:

Heart Beat Pakcet:

Heart Beat Interval: 0 (s)

Whether to let other virtual com interworking data with this com: Not Use

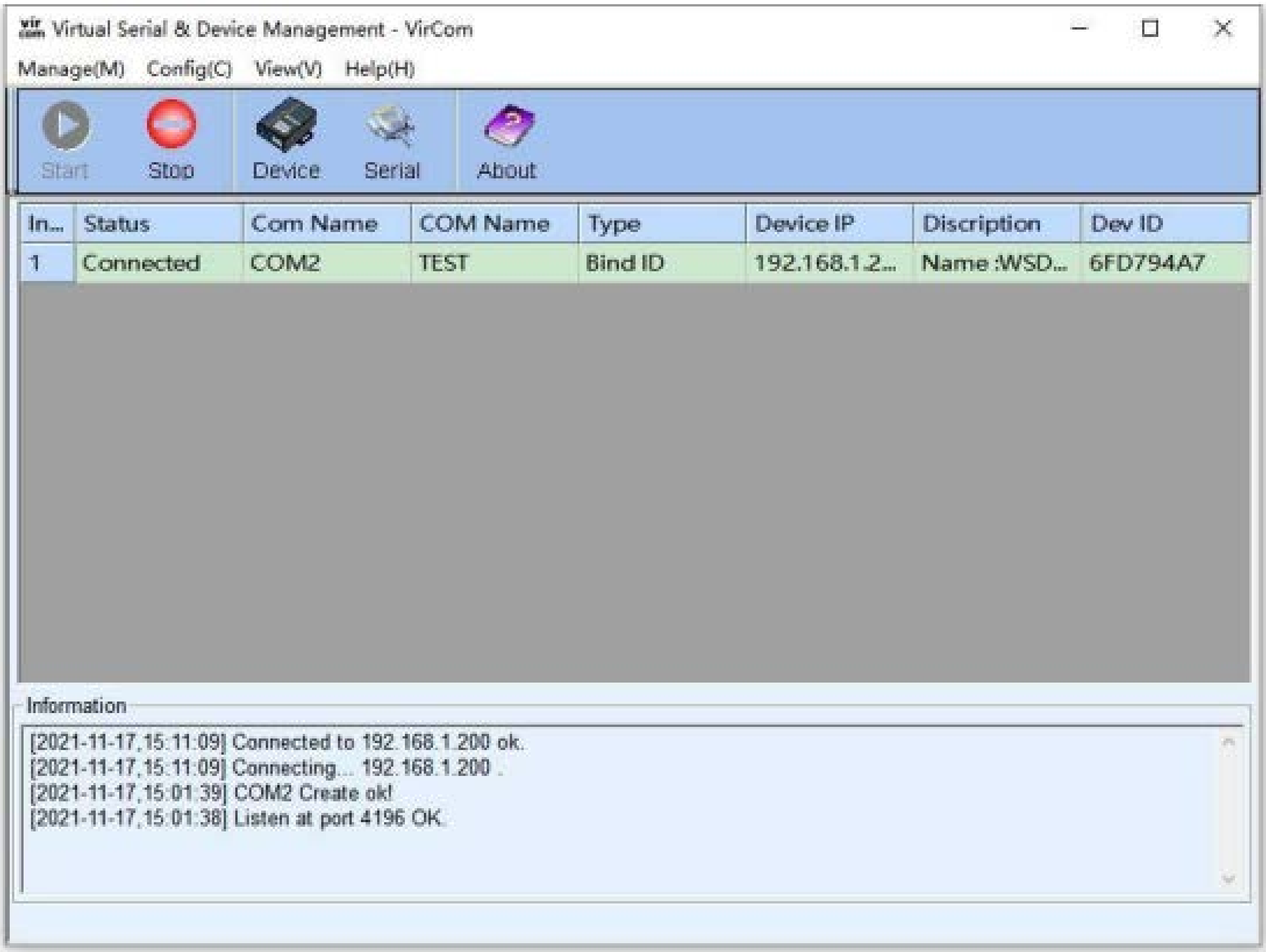
- Then enter the device management and double-click the device that needs to be bound to COM2. As shown in the diagram, select COM2 from the Virtual Serial Port list in the top left corner. Then click on "Modify Setting" and then click on "Restart Device".



COM Number: COM2
Name This COM: TEST
Serial Param Auto Adapt: As Globle Setting(Def.)
Vircom Work Mode: Bind ID(Def.)
TCP Server Mode Listen Port: 22343
Number of Batch Creation: 1
Batch Increase Mode: IP Increase

Client Mode Start Connection Now: ☑
Dest. IP or Domain: 192.168.1.200
Dest. Port: 4196
□ Vircom Register ID:
Vircom Login Key:
Heart Beat Pakcet:
Heart Beat Interval: 0 (s)

- Return to the main interface of Vircom. It can be seen that COM2 has been connected to the device whose IP is 192.168.1.200. In this case, the virtual serial port COM2 can be used instead of the network port for communication.



In...	Status	Com Name	COM Name	Type	Device IP	Discription	Dev ID
1	Connected	COM2	TEST	Bind ID	192.168.1.2...	Name :WSD...	6FD794A7

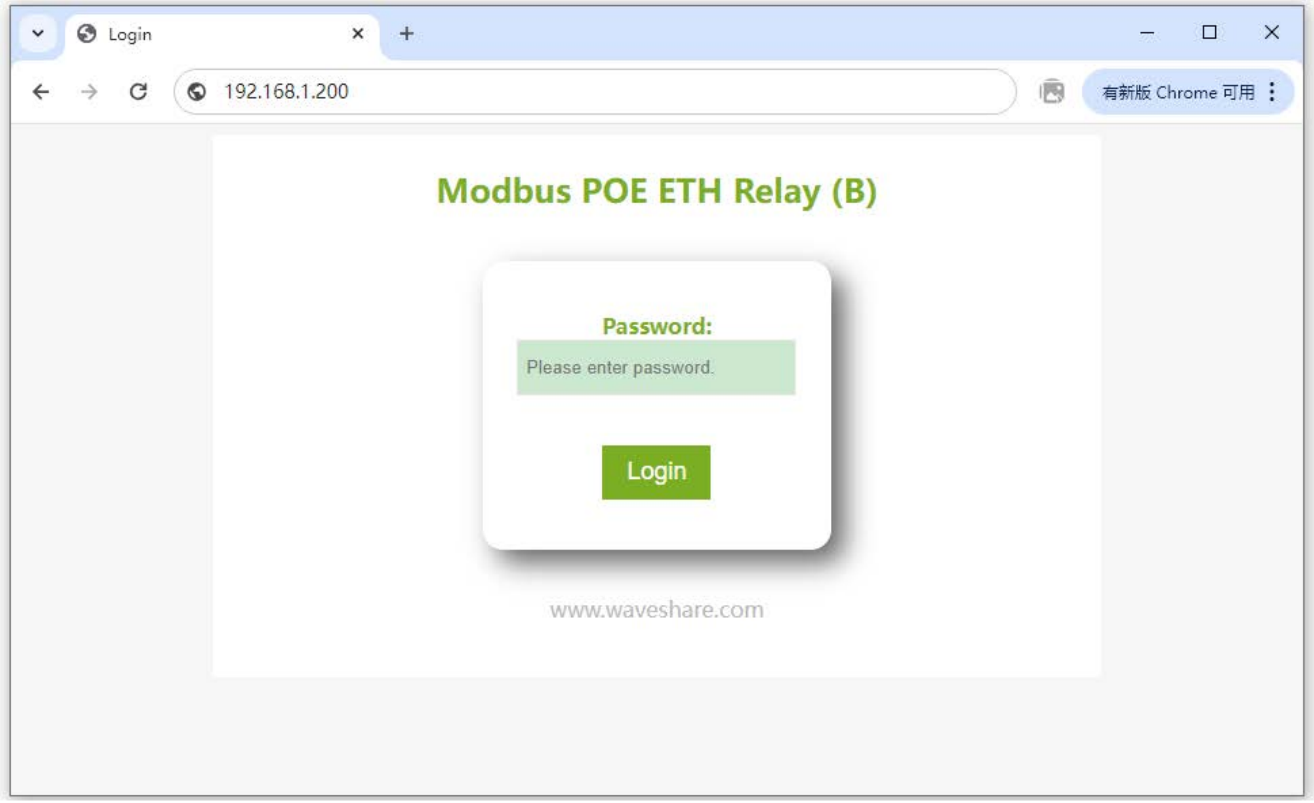
```
[2021-11-17,15:11:09] Connected to 192.168.1.200 ok.
[2021-11-17,15:11:09] Connecting... 192.168.1.200 .
[2021-11-17,15:01:39] COM2 Create ok!
[2021-11-17,15:01:38] Listen at port 4196 OK.
```

WEB Configuration

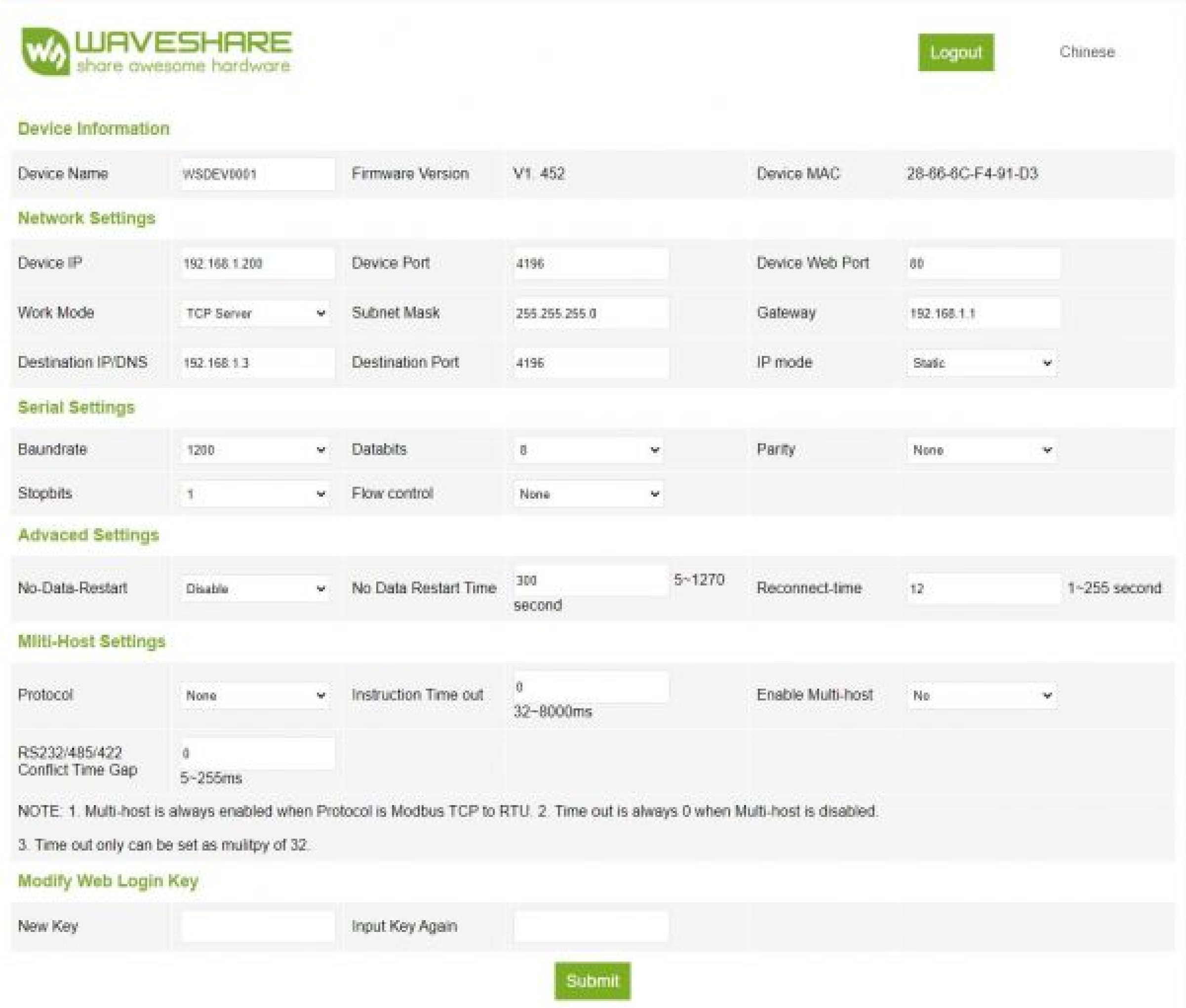
Using Vircom, you can search for and configure device parameters in different network segments. For Web configuration, you must first ensure that the computer and the serial server are in the same IP segment, and you need to know the IP address of the serial server in advance.

But Web configuration can be done on any computer without Vircom. (Different products have different web interfaces, which can be switched between Chinese and English)

1. Enter the IP address of the serial server in the browser, such as http://192.168.1.200 to open the following web page



2. In the Password field, enter your password: The default login password is not set or is set to 123456. If no password is set, you can enter any password and click the Login button to log in. After setting the password to log in, the settings at "Modify Web Login Key" will take effect:



Device Information

Device Name	WSDEV0001	Firmware Version	V1. 452	Device MAC	28-66-8C-F4-91-D3

Network Settings

Device IP	192.168.1.200	Device Port	4196	Device Web Port	80
Work Mode	TCP Server	Subnet Mask	255.255.255.0	Gateway	192.168.1.1
Destination IP/DNS	192.168.1.3	Destination Port	4196	IP mode	Static

Serial Settings

Baundrate	1200	Databits	8	Parity	None
Stopbits	1	Flow control	None		

Advaced Settings

No-Data-Restart	Disable	No Data Restart Time	300 5~1270 second	Reconnect-time	12 1~255 second

Mliti-Host Settings

Protocol	None	Instruction Time out	0 32~8000ms	Enable Multi-host	No
RS232/485/422 Conflict Time Gap	0 5~255ms				

NOTE: 1. Multi-host is always enabled when Protocol is Modbus TCP to RTU. 2. Time out is always 0 when Multi-host is disabled.

3. Time out only can be set as mulitpy of 32.

Modify Web Login Key

New Key		Input Key Again	

3. The serial server parameters can be modified on the web page that appears.

4. After modifying the parameters, click the "Submit" button.

Attention: The system has added webpage settings function by default when it leaves the factory. If the configuration interface page file is overwritten and the webpage cannot be opened, the webpage file needs to be downloaded again.

Please refer to RS485 TO ETH (B) Manual

Example Demonstration

The demo shows how the following two software operate.

SSCOM serial port debugging assistant is more convenient to operate, free of installation, and more convenient for complete display and analysis of instructions, but the disadvantage is that the data is not intuitive.

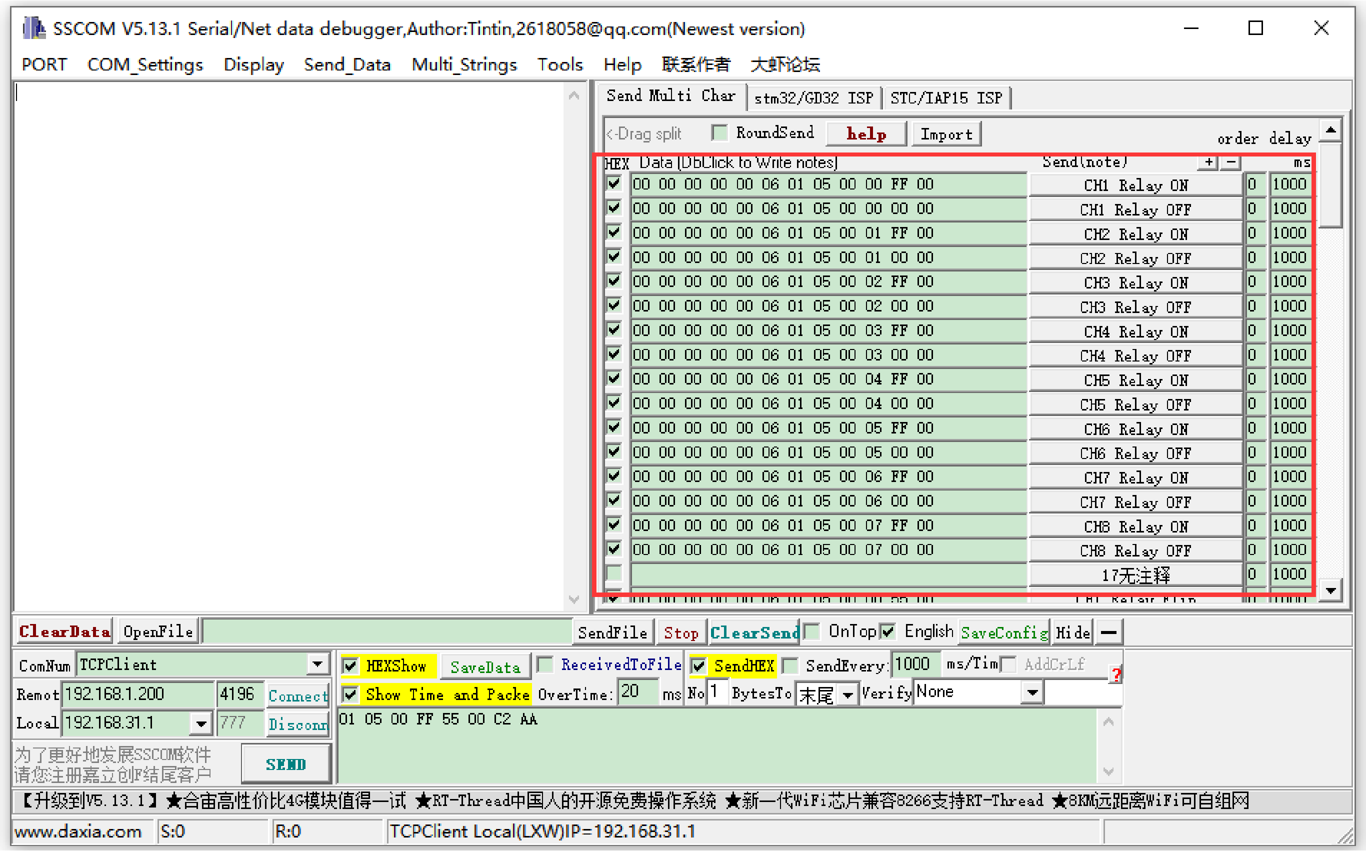
Modbus Poll software is directly operated on the register, and the data display is more convenient to observe, but the disadvantage is that the instruction is not displayed completely, so you need to be familiar with the Modbus register operation.

You can test using any method. It is recommended to use the SSCOM serial port debugging assistant software for the first test.

SSCOM Serial Port Debugging Assistant

Modbus RTU Command: The default configuration is the Modbus RTU command

- 1. Open the serial port debugging assistant window
- 2. Select TCPClient for port number
- 3. Modify the remote IP and port number according to the Vircom settings above
- 4. Click the "Connect" button to connect to the TCP server
- 5. The green light of the network port will light up when the connection is successful
- 6. Click Multi-Char to open the Send Multi-Char window, the default display is the Modbus RTU command, click the corresponding function to send the corresponding command.
- 7. If you use the custom input box below to send the command, you need to set Verify as ModbusCRC16



HEX	Data (DbClick to Write notes)	Send(note)	order	delay ms
☑	00 00 00 00 00 06 01 05 00 00 FF 00	CH1 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 00 00 00	CH1 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 01 FF 00	CH2 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 01 00 00	CH2 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 02 FF 00	CH3 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 02 00 00	CH3 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 03 FF 00	CH4 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 03 00 00	CH4 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 04 FF 00	CH5 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 04 00 00	CH5 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 05 FF 00	CH6 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 05 00 00	CH6 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 06 FF 00	CH7 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 06 00 00	CH7 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 07 FF 00	CH8 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 07 00 00	CH8 Relay OFF	0	1000
□		17无注释	0	1000

ComNum TCPClient
Remot 192.168.1.200 4196
Local 192.168.31.1 777
☑ HEXShow □ ReceivedToFile ☑ SendHEX □ SendEvery: 1000 ms/Tim □ AddCrLf
☑ Show Time and Packe OverTime: 20 ms No 1 BytesTo 末尾 Verify None

```
01 05 00 FF 55 00 C2 AA
```

为了更好地发展SSCOM软件
请您注册嘉立创F结尾客户

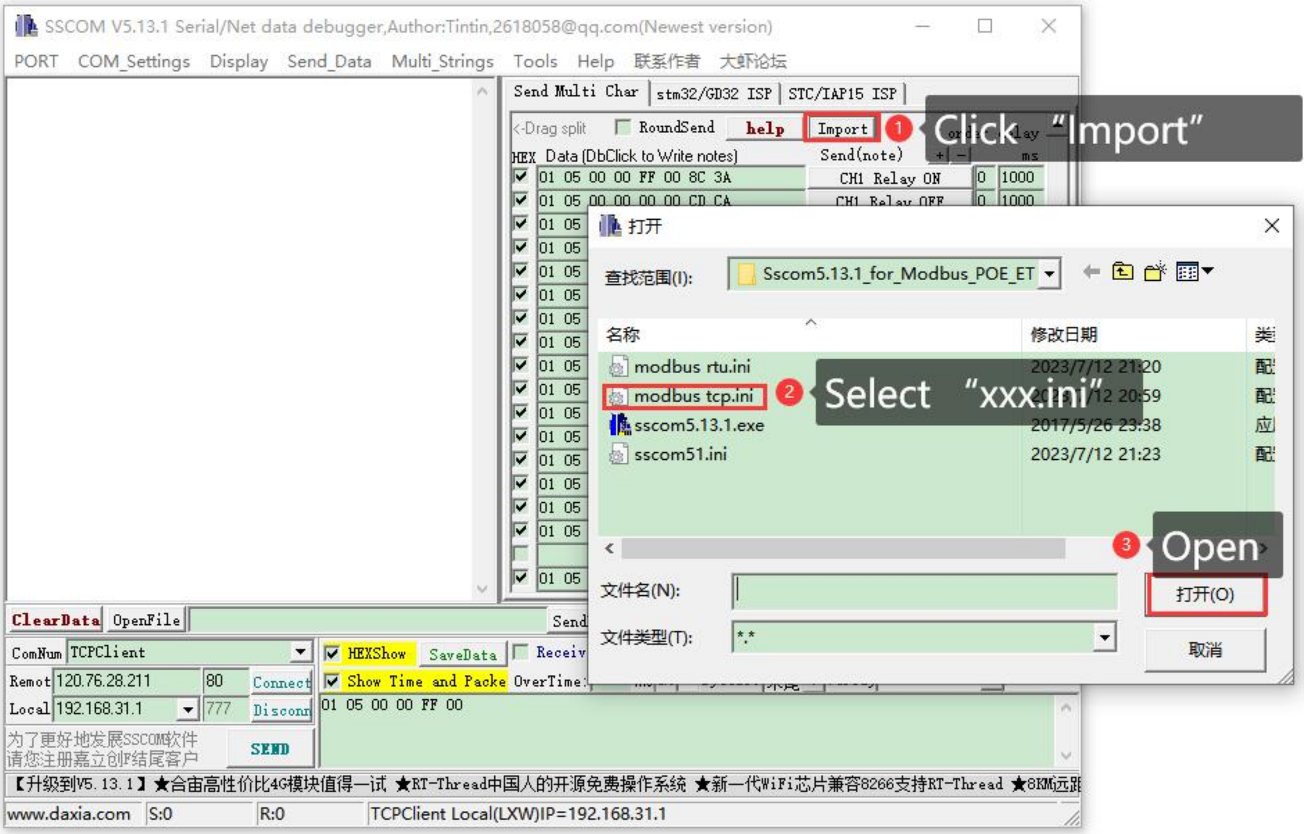
【升级到V5.13.1】★合宙高性价比4G模块值得一试 ★RT-Thread中国人的开源免费操作系统 ★新一代WiFi芯片兼容8266支持RT-Thread ★8KM远距离WiFi可自组网

www.daxia.com S:0 R:0 TCPClient Local(LXW)IP=192.168.31.1

Configure Modbus TCP Directives: If you want to set it as a Modbus TCP Directive, you need to change the commands

- 1. Click on the Import ini button in the Send Multi-Char column
- 2. Select the modbus tcp.ini file to import the Modbus TCP command

Note: If a popup error message says "A component named HEX0 already exists", then you need to close and reopen the software, which will reload the files and refresh the buttons.


SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com(Newest version)
PORT COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾论坛
Send Multi Char | stm32/GD32 ISP | STC/IAP15 ISP
<-Drag split □ RoundSend help Import
1 Click "Import"
2 Select "xxx.ini"
3 Open
ClearData OpenFile
ComNum TCPClient
Remot 120.76.28.211 80 Connect
Local 192.168.31.1 777 Disconn
☑ HEXShow SaveData □ Receiv
☑ Show Time and Packe OverTime
SEND
为了更好地发展SSCOM软件 请您注册嘉立创F结尾客户
www.daxia.com S:0 R:0 TCPClient Local(LXW)IP=192.168.31.1

HEX Data (DbClick to Write notes) — Send(note)

☑	01 05 00 00 FF 00 8C 3A	CH1 Relay ON	0 1000
☑	01 05 00 00 00 00 CD CA	CH1 Relay OFF	0 1000

打开

查找范围(I): Sscom5.13.1_for_Modbus_POE_ET

名称	修改日期	类
modbus rtu.ini	2023/7/12 21:20	配
modbus tcp.ini	2023/7/12 20:59	配
sscom5.13.1.exe	2017/5/26 23:38	应
sscom51.ini	2023/7/12 21:23	配

文件名(N):

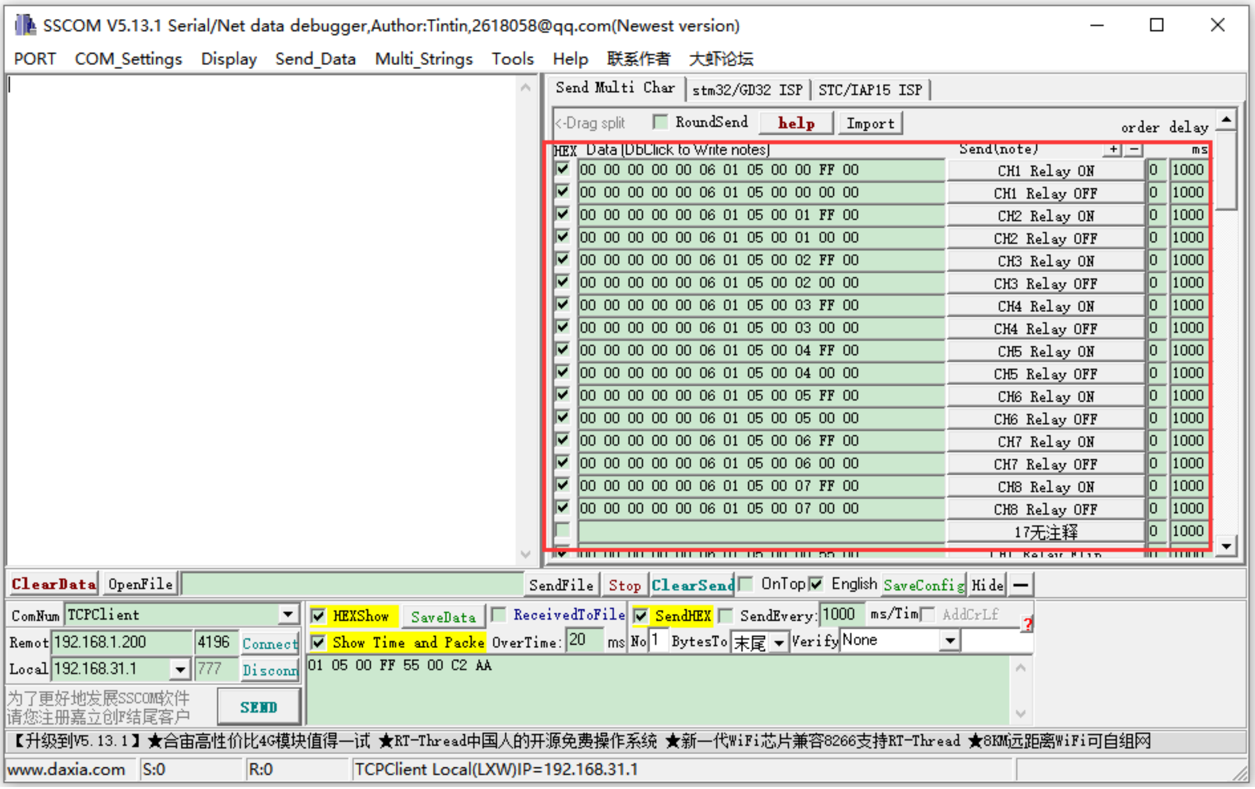
文件类型(T): *.*

打开(O) 取消

01 05 00 00 FF 00

- 3. After successful import, the following is displayed, click on the function to send the corresponding command.

Note: Modbus tcp does not require CRC checksum, select None for verification.


SSCOM V5.13.1 Serial/Net data debugger,Author:Tintin,2618058@qq.com(Newest version)
PORT COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾论坛
Send Multi Char | stm32/GD32 ISP | STC/IAP15 ISP
<-Drag split □ RoundSend help Import
ClearData OpenFile SendFile Stop ClearSend □ OnTop ☑ English SaveConfig Hide
SEND
为了更好地发展SSCOM软件 请您注册嘉立创F结尾客户
【升级到V5.13.1】★合宙高性价比4G模块值得一试 ★RT-Thread中国人的开源免费操作系统 ★新一代WiFi芯片兼容8266支持RT-Thread ★8KM远距离WiFi可自组网
www.daxia.com S:0 R:0 TCPClient Local(LXW)IP=192.168.31.1

HEX	Data (DbClick to Write notes)	Send(note)	order	delay ms
☑	00 00 00 00 00 06 01 05 00 00 FF 00	CH1 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 00 00 00	CH1 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 01 FF 00	CH2 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 01 00 00	CH2 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 02 FF 00	CH3 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 02 00 00	CH3 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 03 FF 00	CH4 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 03 00 00	CH4 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 04 FF 00	CH5 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 04 00 00	CH5 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 05 FF 00	CH6 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 05 00 00	CH6 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 06 FF 00	CH7 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 06 00 00	CH7 Relay OFF	0	1000
☑	00 00 00 00 00 06 01 05 00 07 FF 00	CH8 Relay ON	0	1000
☑	00 00 00 00 00 06 01 05 00 07 00 00	CH8 Relay OFF	0	1000
□		17无注释	0	1000

ComNum TCPClient

Remot 192.168.1.200 4196 Connect

Local 192.168.31.1 777 Disconn

☑ HEXShow SaveData □ ReceivedToFile ☑ SendHEX □ SendEvery: 1000 ms/Tim □ AddCrLf

☑ Show Time and Packe OverTime: 20 ms No 1 BytesTo 末尾 Verify None

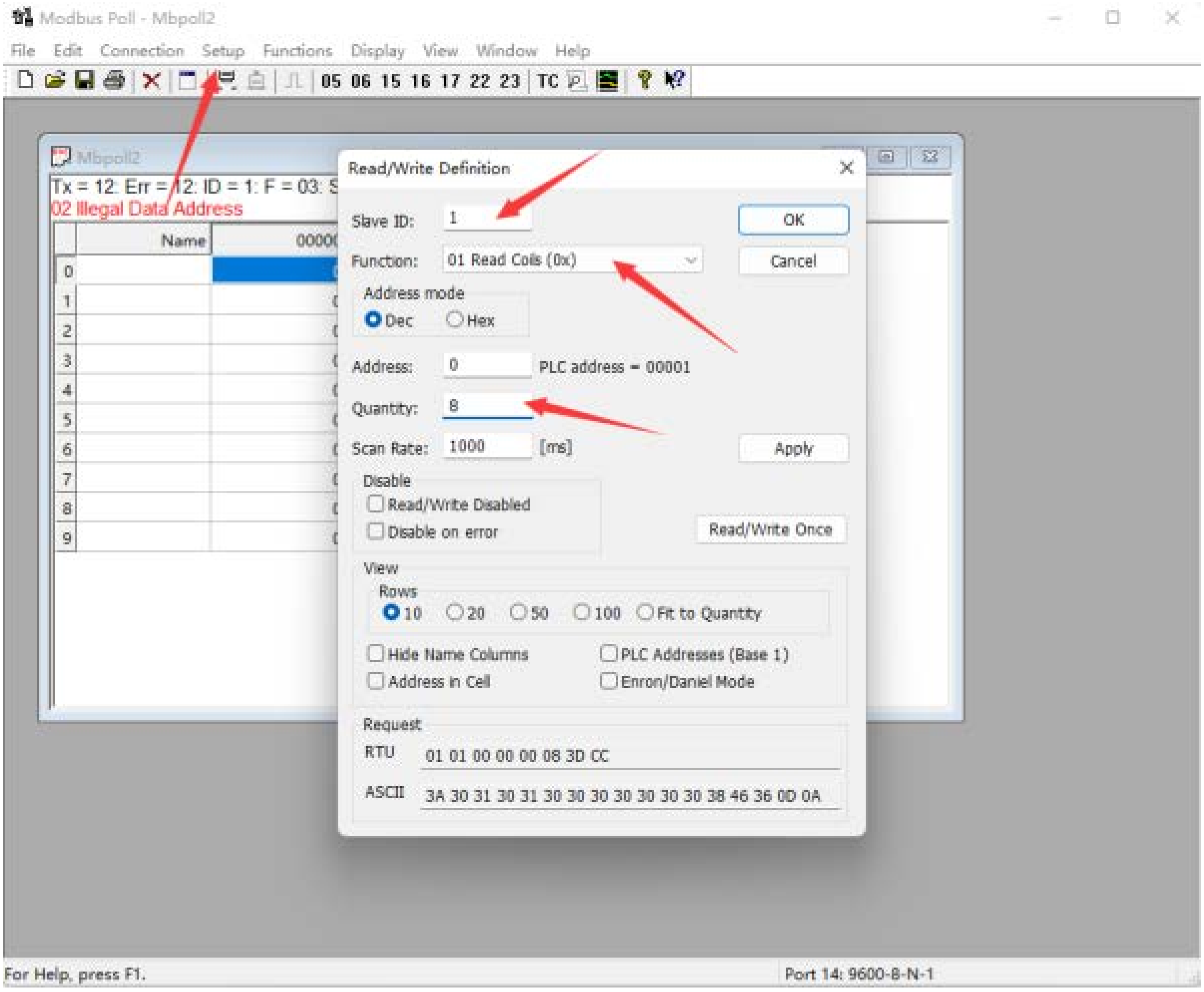
01 05 00 FF 55 00 C2 AA

- For detailed Modubs commands, please see the development protocol.

Modbus Poll Software

It is not convenient to use the SSCOM software for observing the data, you can select Modbus Poll software to read the data. Download and install the Modbus Poll software.

- 1. Open Modbus Poll software
- 2. Select Setup->Read/Write Definition, select the actual device address for Slave ID, select 01 Read Coils function code for Function, and change Quantity to 8 channels. Click OK to confirm.



Modbus Poll - Mbpoll2

Mbpoll2

Tx = 12: Err = 12: ID = 1: F = 03: S

02 Illegal Data Address

Read/Write Definition

Slave ID: 1

Function: 01 Read Coils (0x)

Address mode ● Dec ○ Hex

Address: 0 PLC address = 00001

Quantity: 8

Scan Rate: 1000 [ms]

Disable

□ Read/Write Disabled

□ Disable on error

View

Rows ● 10 ○ 20 ○ 50 ○ 100 ○ Fit to Quantity

□ Hide Name Columns □ PLC Addresses (Base 1)

□ Address in Cell □ Enron/Daniel Mode

Request

RTU 01 01 00 00 00 08 3D CC

ASCII 3A 30 31 30 31 30 30 30 30 30 30 30 38 46 36 0D 0A

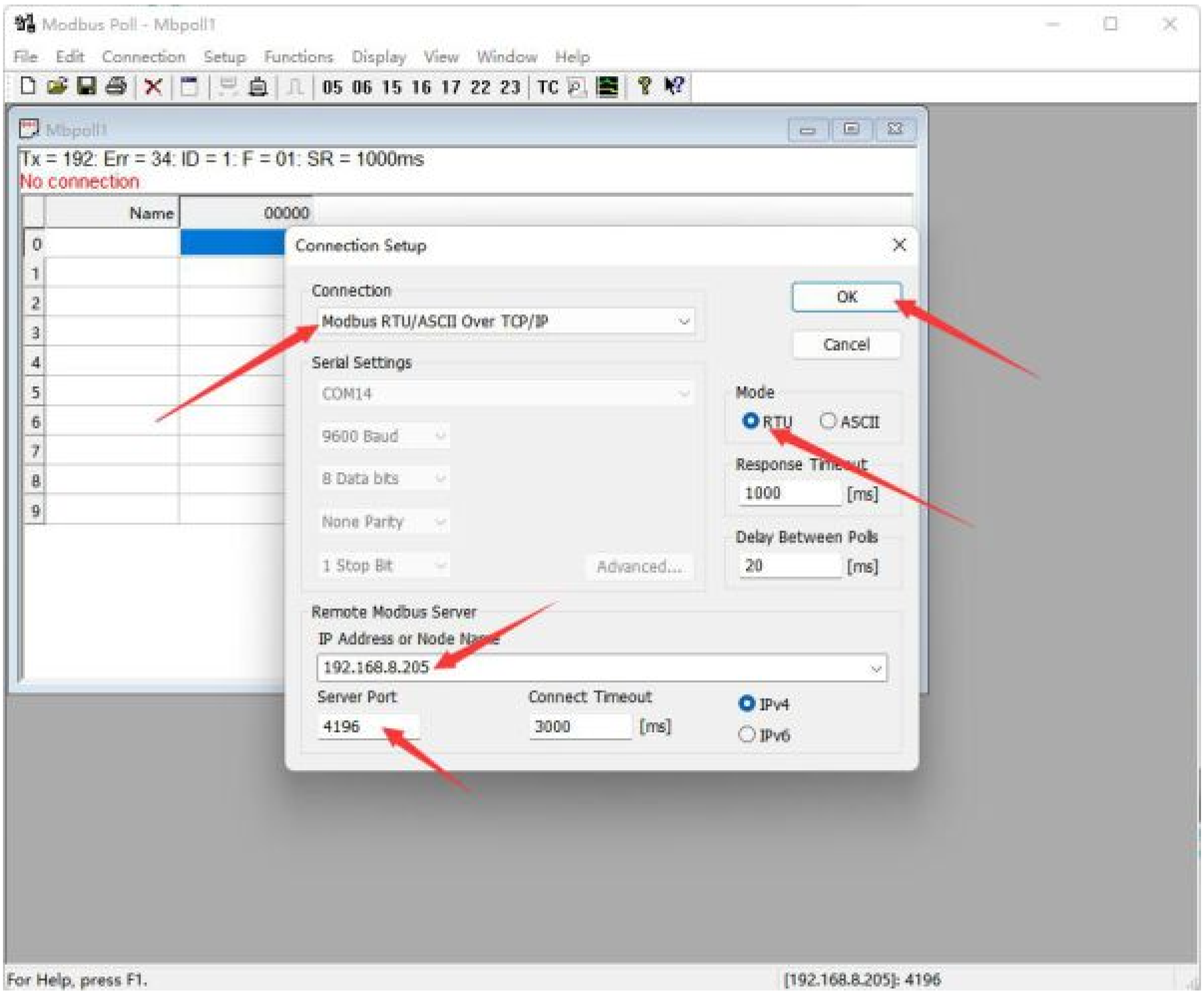
For Help, press F1.

Port 14: 9600-8-N-1

- 3.

--->If you are using the Modbus RTU protocol, select Connection->Connect Setup, select Modbus RTU/ASCII Over TCP/IP for Connection, select RTU for Mode, and enter the correct IP address and port number. Click OK to connect.
--->If you are using the Modbus TCP protocol, select Connection->Connect Setup, select Modbus TCP/IP for Connection, and enter the correct IP address and port

number. Click OK to connect.



Tx = 192: Err = 34: ID = 1: F = 01: SR = 1000ms
No connection

Connection: Modbus RTU/ASCII Over TCP/IP

Serial Settings: COM14, 9600 Baud, 8 Data bits, None Parity, 1 Stop Bit

Mode: ● RTU ○ ASCII

Response Timeout: 1000 [ms]

Delay Between Polls: 20 [ms]

Remote Modbus Server

IP Address or Node Name: 192.168.8.205

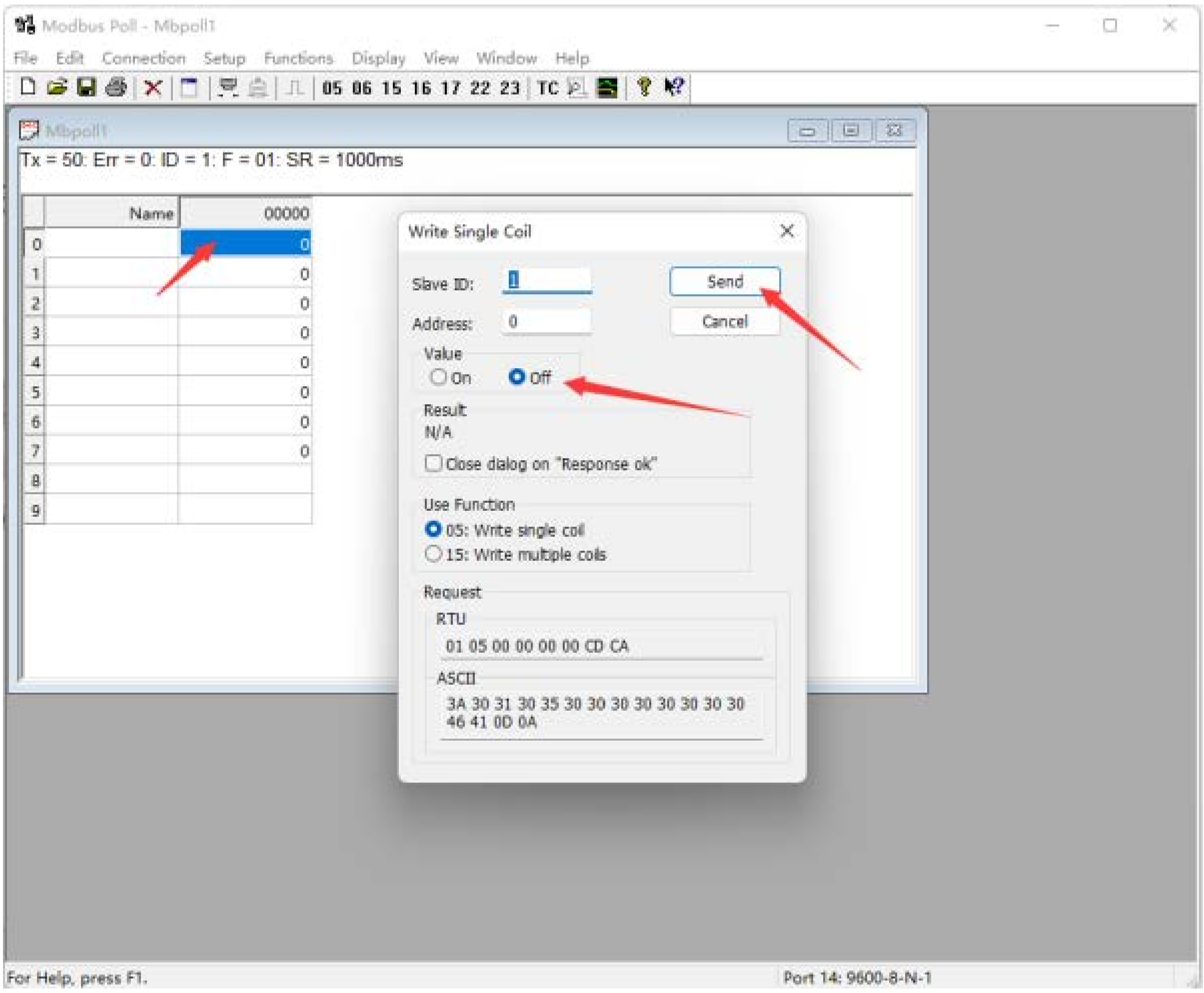
Server Port: 4196

Connect Timeout: 3000 [ms]

● IPv4 ○ IPv6

For Help, press F1.   [192.168.8.205]: 4196

- 4. After the connection is normal, you can check the current relay status. Select the corresponding channel, then double-click the status value to pop up the send page. Choose On or Off, then Click Send to control the relay opening and closing.



Tx = 50: Err = 0: ID = 1: F = 01: SR = 1000ms

	Name	00000
0		0
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		
9		

Slave ID: 1

Address: 0

Value
○ On ● Off

Result
N/A

□ Close dialog on "Response ok"

Use Function
● 05: Write single coil
○ 15: Write multiple coils

Request
RTU
01 05 00 00 00 00 CD CA
ASCII
3A 30 31 30 35 30 30 30 30 30 30 30 30
46 41 0D 0A

For Help, press F1.

Port 14: 9600-8-N-1

Demo

Raspberry Pi

Connect the Raspberry Pi and the ModBus POE ETH Relay module to the same LAN. Open the Raspberry Pi terminal and run the program by entering the following command:

```
sudo apt-get install unzip

wget https://files.waveshare.com/wiki/Modbus-POE-ETH-Relay-30CH/Modbus_POE_ETH_R
elay_30CH_Code.zip

unzip Modbus_POE_ETH_Relay_30CH_Code.zip

cd Modbus_POE_ETH_Relay_30CH_Code


#modbus rtu protocol

vi modbus_rtu.py  #Change the IP address and port number according to the actual
situation

sudo python3 modbus_rtu.py
```

```
#modbus tcp protocol

vi modbus_tcp.py  #Change the IP address and port number according to the actual
situation

sudo python3 modbus_tcp.py
```

Note: To run this demo, you need to modify the demo file to change the IP address and port number to the actual IP address and port number of the ModBus POE ETH Relay.

Modbus RTU Development Protocol V2

Function Code Introduction

Function Code	Description	Note
01	Read coil status	Read relay status
03	Read holding register	Read the address and version
05	Write single coil	Write single relay
06	Write single register	Set the baud rate and address
0F	Write multiple coils	Write all relays

Register Address Introduction

Address (HEX)	Address storage content	Register value	Permission	Modbus Function Code
0x0000 …… 0x001D	Channel 1~30 relay address	0xFF00: relay on 0x0000: relay off 0x5500: relay toggle	Read/Write	0x01, 0x05, 0x0F
0x00FF	Control all relays	0xFF00: all relays on 0x0000: all relays off	Write	0x05

		0x5500: all relays toggle		
0x0100 …… 0x011D	Channel 1~30 relay toggle address	0xFF00: relay toggle 0x0000: relay unchanged	Write	0x05, 0x0F
0x01FF	Control all relays toggle	0xFF00: all relays toggle 0x0000: all relays unchanged	Write	0x05
0x0200 …… 0x021D	Channel 1~30 relay flash on	Interval time: data*100ms Value: 0x0007, Interval time: 7*100MS = 700MS	Write	0x05
0x0400 …… 0x041D	Channel 1~30 relay flash off	Interval time: data*100ms Value: 0x0007, Interval time: 7*100MS = 700MS	Write	0x05
4x4000	Device Address	Directly store Modbus address Device address: 0x0001	Read	0x03
4x8000	Software Version	Converting to decimal and then shifting the decimal point two places to the left will represent the software version 0x0064 = 100 = V1.00	Read	0x03

Modbus RTU Command Introduction

Control Single Relay

Send code: 01 05 00 00 FF 00 8C 3A

Field	Description	Note
01	Device Address	Fixed 0x01
05	05 Command	Relay control
00 00	Address	The register address of the relay to be controlled, 0x0000-0x001D

FF 00	Command	0xFF00: relay on;
		0x0000: relay off;
		0x5500: relay toggle

8C 3A	CRC16	The CRC16 checksum of the first 6 bytes of data

Return code: 01 05 00 00 FF 00 8C 3A

Field	Description	Note
01	Device Address	Fixed 0x01
05	05 Command	Relay control
00 00	Address	The register address of the relay to be controlled, 0x0000-0x001D

FF 00	Command	0xFF00: relay on;
		0x0000: relay off;
		0x5500: relay toggle

8C 3A	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]:

```
Relay 0 on: 01 05 00 00 FF 00 8C 3A
Relay 0 off: 01 05 00 00 00 00 CD CA
Relay 1 on: 01 05 00 01 FF 00 DD FA
Relay 1 off: 01 05 00 01 00 00 9C 0A
Relay 2 on: 01 05 00 02 FF 00 2D FA
Relay 2 off: 01 05 00 02 00 00 6C 0A
Relay 3 on: 01 05 00 03 FF 00 7C 3A
Relay 3 off: 01 05 00 03 00 00 3D CA
Relay 0 toggle: 01 05 00 00 55 00 F2 9A
Relay 1 toggle: 01 05 00 01 55 00 A3 5A
Relay 2 toggle: 01 05 00 02 55 00 53 5A
Relay 3 toggle: 01 05 00 03 55 00 02 9A
```

Control All Relays

Send code: 01 05 00 FF FF 00 BC 0A

Field	Description	Note
01	Device Address	Fixed 0x01
05	05 Command	Relay control
00 FF	Address	Fixed 0x00FF

FF 00	Command	0xFF00: relay on;
		0x0000: relay off;
		0x5500: relay toggle

BC 0A	CRC16	The CRC16 checksum of the first 6 bytes of data

Return code: 01 05 00 FF FF 00 BC 0A

Field	Description	Note
01	Device Address	Fixed 0x01
05	05 Command	Relay control
00 FF	Address	Fixed 0x00FF

FF 00	Command	0xFF00: relay on;
		0x0000: relay off;
		0x5500: relay toggle

BC 0A	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]:

```
All relays on: 01 05 00 FF FF 00 BC 0A
All relays off: 01 05 00 FF 00 00 FD FA
All relays toggle: 01 05 00 FF 55 00 C2 AA
```

Read Relay Status

Send code: 01 01 00 00 00 1E BC 02

Field	Description	Note
01	Device Address	Fixed 0x01
01	01 Command	Query relay status
00 00	Relay Start Address	Relay start address, 0x0000-0x001D
00 1E	The Number of Relays	Read relay quantity
BC 02	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 01 01 04 00 00 00 00 FB D1

Field	Description	Note
01	Device Address	Fixed 0x01
01	01 Command	Query relay status
04	Byte Number	The number of all bytes of the returned status information
00 00 00 00	Query status	Received relay status The relay state is in small-endian format, with low bytes first and high bytes last Bit0: the first relay status; Bit1: the second relay status; And so on, with the idle high bit being zero
FB D1	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

```
Send: 01 01 00 00 00 1E BC 02          //Query all relays
Receive: 01 01 04 00 00 00 00 FB D1    //all relays off
Send: 01 01 00 01 00 03 2D CB          //Query relays 1, 2, 3 status
Receive: 01 01 01 05 91 8B             //Relays 1 and 3 are on, relay 2 is off
Send: 01 01 00 04 00 0C 7D CE          //Query relays 4-15
Receive: 01 01 02 0F 00 BC 0C          //Relays 4-7 are on, relays 8-15 are off
```

Write Relay Status

Send code: 01 0F 00 00 00 1E 04 FF FF FF 3F C1 92

Field	Description	Note
01	Device Address	Fixed 0x01
0F	0F Command	Write relay status
00 00	Relay Start Address	Relay start address, 0x0000-0x001D
00 1E	The Number of Relays	The number of relays to be written
04	Byte Number	The byte number of the status
FF FF FF 3F	Relay status	The relay state is in small-endian format, with low bytes first and high bytes last Bit0: the first relay status; Bit1: the second relay status; And so on, with the idle high bit being zero
C1 92	CRC16	The CRC16 checksum of the first 6 bytes of data

Return code: 01 0F 00 00 00 1E D5 C3

Field	Description	Note

01	Device Address	Fixed 0x01
0F	0F Command	Control all registers
00 00	Address	Relay start address
00 1E	The Number of Relays	The number of relays to be written
D5 C3	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

```
All relays on: 01 0F 00 00 00 1E 04 FF FF FF 3F C1 92
0-1 on; 3-29 off: 01 0F 00 00 00 1E 04 03 00 00 00 C0 12
0-1 on; 3-7 off: 01 0F 00 00 00 08 01 03 BE 94
Relays 1, 2, and 3 on: 01 0F 00 01 00 03 01 07 F3 55
```

Relay Flash ON/OFF Command

Send code: 01 05 02 00 00 07 8D B0

Field	Description	Note
01	Device Address	Fixed 0x01
05	05 Command	Single control command
02	Command	02: flash on, 04: flash off
00	Relay Address	The address of the relay to be controlled, 0x00~0x1D
00 07	Interval Time	The interval time: data*100ms Value: 0x0007, Interval time: 7*100MS = 700MS The maximum setting for the flash-on flash-off time is 0x7FFF
8D B0	CRC16	The CRC16 checksum of the first 6 bytes of data

Receive code: 01 05 02 00 00 07 8D B0

Field	Description	Note
01	Device Address	Fixed 0x01
05	05 Command	Single control command
02	Command	02: flash on, 04: flash off
00	Relay Address	The address of the relay to be controlled, 0x00~0x1D
00 07	Interval Time	The interval time: data*100ms Value: 0x0007, Interval time: 7*100MS = 700MS
8D B0	CRC16	The CRC16 checksum of the first 6 bytes of data

For example: [Address 1 device]

```
Relay 0 flash on: 01 05 02 00 00 07 8D B0  //700MS
Relay 1 flash on: 01 05 02 01 00 08 9C 74  //800MS
Relay 0 flash off: 01 05 04 00 00 05 0C F9  //500MS
Relay 1 flash off: 01 05 04 01 00 06 1D 38  //600MS
```

Read Software Version Command

Send code: 01 03 80 00 00 01 AD CA

Field	Description	Note
01	Device Address	Fixed 0x01
03	03 Command	Read holding register
80 00	Command register	0x8000: read software version

00 01	Byte Number	Fixed 0x0001
AD CA	CRC16	The CRC16 checksum of the first 6 bytes of data

Return code: 01 03 02 00 64 B9 AF

Field	Description	Note
01	Device Address	Fixed 0x01
03	03 Command	Read holding register
02	Byte Number	The number of bytes returned
00 64	Software Version	Converting to decimal and then shifting the decimal point two places to the left will represent the software version 0x0064 = 100 = V1.00
F0 B8	CRC16	The CRC16 checksum of the first 6 bytes of data

For example:

```
Send: 01 03 80 00 00 01 AD CA
Receive: 01 03 02 00 C8 B9 D2       //0x00C8 = 200 =V2.00
```

Exception Function Code

When the received command is incorrect or the device is abnormal, an exception response will be returned in the following format:

Return code: 01 85 03 02 91

Field	Description	Note
01	Device Address	0x00 indicates the broadcast address, 0x01-0xFF indicates the device address

85	Exception Function Code	Exception function code = Request function code + 0x80
03	Byte Number	Exception Code
02 91	CRC16	The CRC16 checksum of the first 6 bytes of data

An exception code is a single-byte value that indicates the type of error. Several commonly used exception codes defined by the Modbus protocol:

Exception Code	Name	Description
0x01	Illegal Function	The requested function code is not supported
0x02	Illegal Data Address	The requested data address is incorrect
0x03	Illegal Data Value	The requested data value or operation cannot be executed
0x04	Server Failure	Server equipment failure
0x05	Response	The request has been received and is being processed
0x06	Device Busy	The device is currently busy and cannot perform the requested operation

Modbus TCP Development Protocol

Here is a brief introduction to Modbus TCP and Modbus RTU protocol conversion using the above commands to open the first relay as an example.

- Modbus RTU command: 01 05 00 00 FF 00 8C 3A

Field	Description	Note

01	Device Address	Fixed 0x01
05	05 Command	Relay control
00 00	Address	The register address of the relay to be controlled, 0x00, that is, the first relay
FF 00	Command	0xFF00: Relay on
8C 3A	CRC16	The CRC16 checksum of the first 6 bytes of data

- Modbus TCP command: 00 00 00 00 00 06 01 05 00 00 FF 00

Field	Description	Note
00 00	Message Label	Both be 0x00
00 00	modbus Label	Must both be 0, which means this is Modbus communication
00 06	Byte Length	Indicates the number of all bytes that follow, followed by 6 bytes
01	Device Address	Fixed 0x01
05	05 Command	Relay control
00 00	Address	The register address of the relay to be controlled, 0x00, that is, the first relay
FF 00	Command	0xFF00: Relay on

By comparing the commands above, we can observe that to convert a Modbus RTU command to Modbus TCP protocol, the CRC check is removed, and the command is prefixed with five 0x00 bytes followed by a byte representing the length.

Advanced Applications

- Relay control through Alibaba Cloud MQTT

- Relay control through Waveshare Cloud
- Relay control through HTTP GET/POST

Resources

Demo

- Demo

Software

- Configuration software VirCom_en
- Virtual serial port driver
- Sscom software
- Modbus Poll software
- SecureCRT software

Related Resources

- Modbus Protocol Specification
- Modbus Series BootLoader Description