2 CH-RS485-ETH (V)

4 CH-RS485-ETH (V)/(VI)

8 CH-RS485-ETH (V)/(VI)

User Manual

Figure 1 2/4/8 CH-RS485-ETH (V)/(VI)

1. Overview

2CH-RS485-ETH (V), 4CH-RS485-ETH (V)/(VI) and 8CH-RS485-ETH (V)/(VI) is an RS485 IoT gateway specifically designed for industrial environments. It integrates multiple functions, including a serial server, Modbus gateway, MQTT gateway, and RS485 to JSON converter.

2CH-RS485-ETH (V) has 2 RS485 interfaces and an Ethernet interface, with optocoupler isolation, isolation voltage 1500V.

4CH-RS485-ETH (V) has 4 RS485 interfaces and 2 Ethernet interface.

4CH-RS485-ETH (VI) has 4 RS485 interfaces and 2 Ethernet interface, with optocoupler isolation, isolation voltage 3000V.

8CH-RS485-ETH (V) has 8 RS485 interfaces and 2 Ethernet interface.

8CH-RS485-ETH (VI) has 8 RS485 interfaces and 2 Ethernet interface , with optocoupler isolation, isolation voltage 3000V.

They Comes with mounting ears and rail buckles, supporting 3 installation methods. Small size and easy to install. The devices can use the same IP address, and different serial ports are distinguished by ports; each serial port can also use different IP addresses. This design improves flexibility and manageability. It can not only save IP resources through shared addresses, but also enable independent network communication through different IP address.

The devices supports transparent transmission and Modbus RTU to Modbus TCP function, has storage Modbus gateway characteristics, and also supports MQTT protocol. Supports MODBUS RTU, 645 protocol conversion to JSON and scheduled upload function through MQTT/HTTP POST.

The devices can be applied to：

- Power electronics, smart meters and energy consumption monitoring;
- As an IoT gateway and a communication bridge between devices and the cloud;
- Remote monitoring and program downloading of various automated PLCs;
- Various configuration software and device communication interfaces;
- Equipment networking in the field of access control and security;
- Network information collection of medical equipment;

Typical application connections are shown in Figure 2. The original serial device RS485 is connected to the RS485 port of 2/4/8 CH-RS485-ETH (V)/(VI), and 2/4/8 CH-RS485-ETH (V)/(VI) is connected to the computer through a network cable. The software on the computer establishes a connection with the 2/4/8 CH-RS485-ETH (V)/(VI) through TCP/IP, virtual serial port, or cloud server. After that, any data sent by the serial port device will be transparently transmitted to the computer software, and the data sent by the software to 2/4/8 CH-RS485-ETH (V)/(VI) through the network will also be

transparently transmitted to the serial port device.

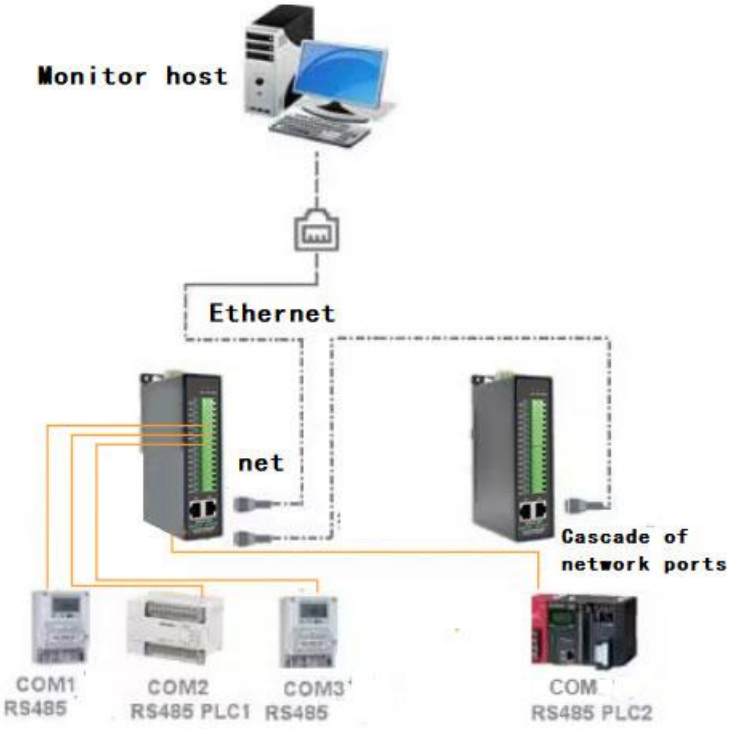


Figure 2 Connection Diagram

2. Features

2.1 Hardware Features

- Three options are available: rail mount installation, backplane mounting ear installation, and wall mounting ear fixed installation. It is especially suitable for industrial guide rail installation. The narrow width can save installation space and make installation and disassembly more convenient.
- Industrial grade power supply method: terminal block wiring method, convenient for use in industrial situations. Can be powered by a wide voltage of 9~24V and 18~36V. With shell ground wire, it can effectively guide interference and surge.
- RS485 serial port optocoupler isolation. It is suitable for strong interference environments where ordinary serial servers cannot work properly. Effectively isolate the ground circuit between the serial port device and the serial port server.
- Abundant panel indicator light for debugging: in connection, not only indicates whether the network cable is connected to the 100M_LINK light, but also indicates the LINK light TCP connection established; Data indicator light has "serial port to network port", "network port to serial port" independent indicator light.

- 4 CH-RS485-ETH (V)/(VI) have a dual network port design, supporting the cascading of serial port servers, and can be expanded to 8 serial ports, 16 serial ports, 24 serial ports, etc.
- 8 CH-RS485-ETH (V)/(VI) have a dual network port design, supporting the cascading of serial port servers, and can be expanded to 16 serial ports, 24 serial ports, 32 serial ports, etc.

2.2 Software Features

- 2/4/8 channels can be configured with different baud rates, operate independently, and do not interfere with each other.
- Supports IP "clone" technology: 2/4/8 serial ports can be distinguished by port or IP. Port differentiation: 2/4/8 IPs can be merged into one IP with different ports, suitable for networks lacking IP; IP differentiation: one IP can be transformed into 2/2~4/2~8 IPs with the same port, suitable for applications with fixed port numbers (for example, Modbus TCP port 502 needs to be fixed)
- Supports TCP server, TCP client, UDP mode, UDP multicast. When acting as a TCP client, it also supports TCP server functions. As a TCP server, it supports up to 30 TCP connections, and as a TCP client, it supports 7 destination IPs.
- Supports virtual serial port, equipped with Windows virtual serial port & device management tool Vircom.
- Supports the function of sending MAC address on device connection to facilitate cloud management of devices.
- Provides a secondary development package DLL development library for searching and configuring equipment on the computer side.
- Supports Web browser configuration, supports DHCP to dynamically obtain IP, and DNS protocol to connect domain name server address.
- Supports remote search of devices, configuration of device parameters, and device program upgrades in the cloud.

2.3 Advanced software features

- Support Modbus gateway function, support Modbus RTU to Modbus TCP. The storage Modbus can be supported, and the equipment data can be collected automatically and stored. Modbus gateway for non-storage mode is also supported.
- Supports multiple hosts: In a question-and-answer query mode, multiple computers can simultaneously access the same serial port device via the network port.
- Support JSON to Modbus RTU and 645 meter protocol, support HTTP POST, HTTP GET format upload data.

- Support for custom heartbeat package and registration package features: easy to communicate with the cloud and device identification.
- Supports Modbus RTU/DLT645/DLT698 to JSON conversion, and the 2/4/8 serial ports support different acquisition modes.
- supports MQTT connection, various public cloud data transmission and other features, and simultaneously supports different MQTT connections with 2/4/8 serial ports.

3. Technical Parameters

Table 1 Technical Parameters

Appearance			
Interface:	RS485: terminal		
Power Supply:	Terminal		
Installation mode:	35mm guide rail installation		
Size:	2CH-RS485-ETH (V)	L x W x H =8.8cm×6.2cm×3.3cm	
	4CH-RS485-ETH (V)(VI) 8CH-RS485-ETH (V)(VI)	L x W x H =15cm×10.5cm×4.1m	
Communicate Interface			
Ethernet:	10M/100M, 2KV surge protection		
Serial	RS485×2/4/8: 485A, 485B		
Serial Port Parameters			
Baud rate:	300~921600bps	Parity:	None, odd, even, mark, space
Data bits:	5～9	Flow control:	XON/XOFF, NONE
Software			
protocol:	ETHERNET、IP、TCP、UDP、HTTP、ARP、ICMP、DHCP、DNS		
configuration method :	Vircom, WEB browser, device management library, Serial port AT command configuration		
Communication method:	TCP/IP direct communicate, Virtual serial		
Operating Model			
TCP server, TCP client, UDP, UDP multicast			

Power Requirement		
Input voltage:	2CH-RS485-ETH (V) 4CH-RS485-ETH (V) 8CH-RS485-ETH (V)	9～24V DC
	4CH-RS485-ETH (VI) 8CH-RS485-ETH (VI)	18～36V DC
Input current:	2CH-RS485-ETH (V) 4CH-RS485-ETH (V) 8CH-RS485-ETH (V)	30mA@12V DC
	4CH-RS485-ETH (VI) 8CH-RS485-ETH (VI)	120mA@24V DC

Environment	
Operating temperature:	-40~85℃
Storage temp:	-45~165℃
Humidity:	5~95%RH

CONTENT

1. Overview 2

2. Features 4
- 2.1 Hardware Features 4
- 2.2 Software Features 4
- 2.3 Advanced software features 4

3. Technical Parameters 5

4. Instructions *9*
- 4.1 Hardware Description 9
- 4.2 Hardware Connection 14
- 4.3 Software Installation 14
- 4.4 Parameter Configuration 15
- 4.5 TCP communication Test 21
- 4.6 Virtul serial port test 22
- 4.7 Modbus TCP test 25
- 4.8 Web configuration 27

5. Work Mode and Transfer Protocol *28*
- 5.1 Virtual Serial Mode 30
- 5.2 Direct TCP/IP Communication Mode 300
  - 5.2.1 TCP Client Mode 30
  - 5.2.2 The Client connects to Multiple Servers 31
  - 5.2.3 TCP server Mode 32
  - 5.2.4 Acting as both client and server 33
  - 5.2.5 UDP Mode 35
- 5.3 Device Connecting Method 35

6. Device Debugging *36*
- 6.1 Network Physical Connection 36
- 6.2 Network TCP Connection 36
- 6.3 Data Sending and Receiving 37
- 6.4 Vircom Remotely Monitor Data 37

7. Modbus Advanced Features *39*
- 7.1 Enable Modbus Gateway 39
- 7.2 Storage Modbus Gateway 40
- 7.3 Disable Storage Functionality 42
- 7.4 Multi-host Function 42
- 7.5 Multi-host Parameter 44
- 7.6 Modbus Under Multi-destination IP 45

8. Registration Pack and Heartbeat Pack 46
- 8.1 Registration Packet 46
  - 8.1.1 Send MAC Address On Connection 46
  - 8.1.2 Realcom Agreement 46
  - 8.1.3 Custom registration package 46

8.1.4 Configuration File 46
8.2 Heartbeat Packets 49
8.2.1 Implied Heartbeat 50
8.2.2 REALCOM Protocol 50
8.2.3 Customize Heartbeat Packets 50
9. Remote Device Management *50*
10. MQTT and JSON 52
11. Firmware upgrade method 52
12. After-sales service and Technical Support 56

4. Instructions

4.1 Hardware Description

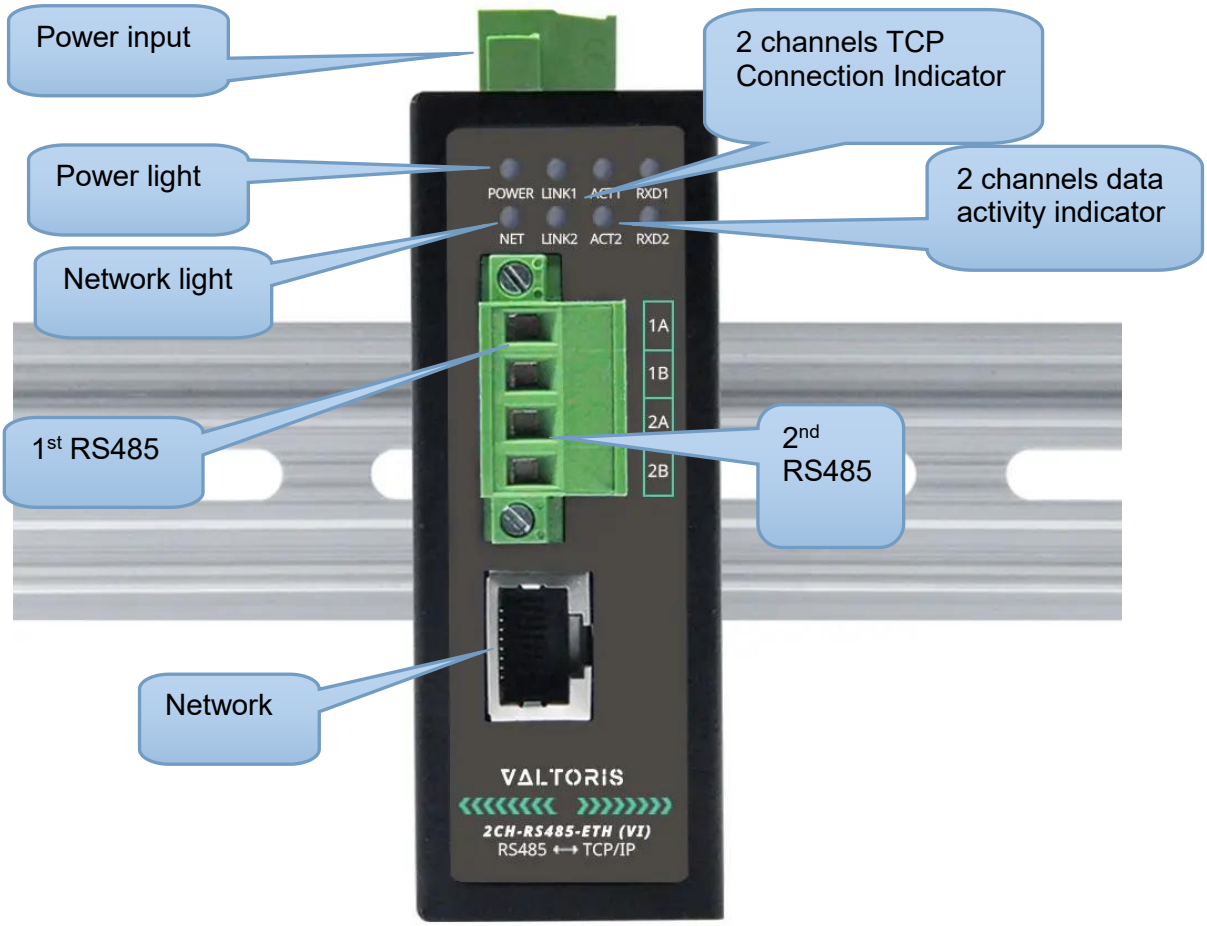
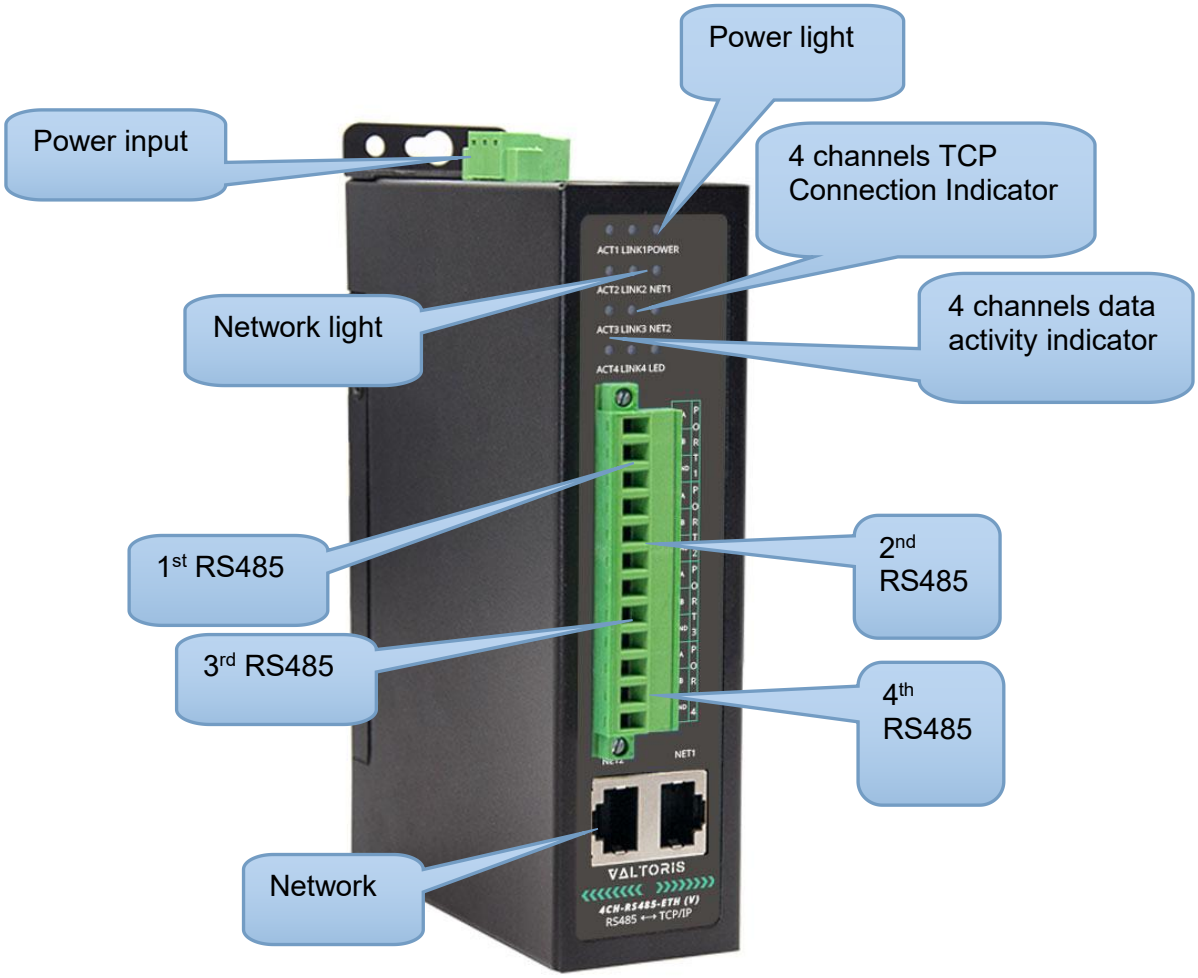
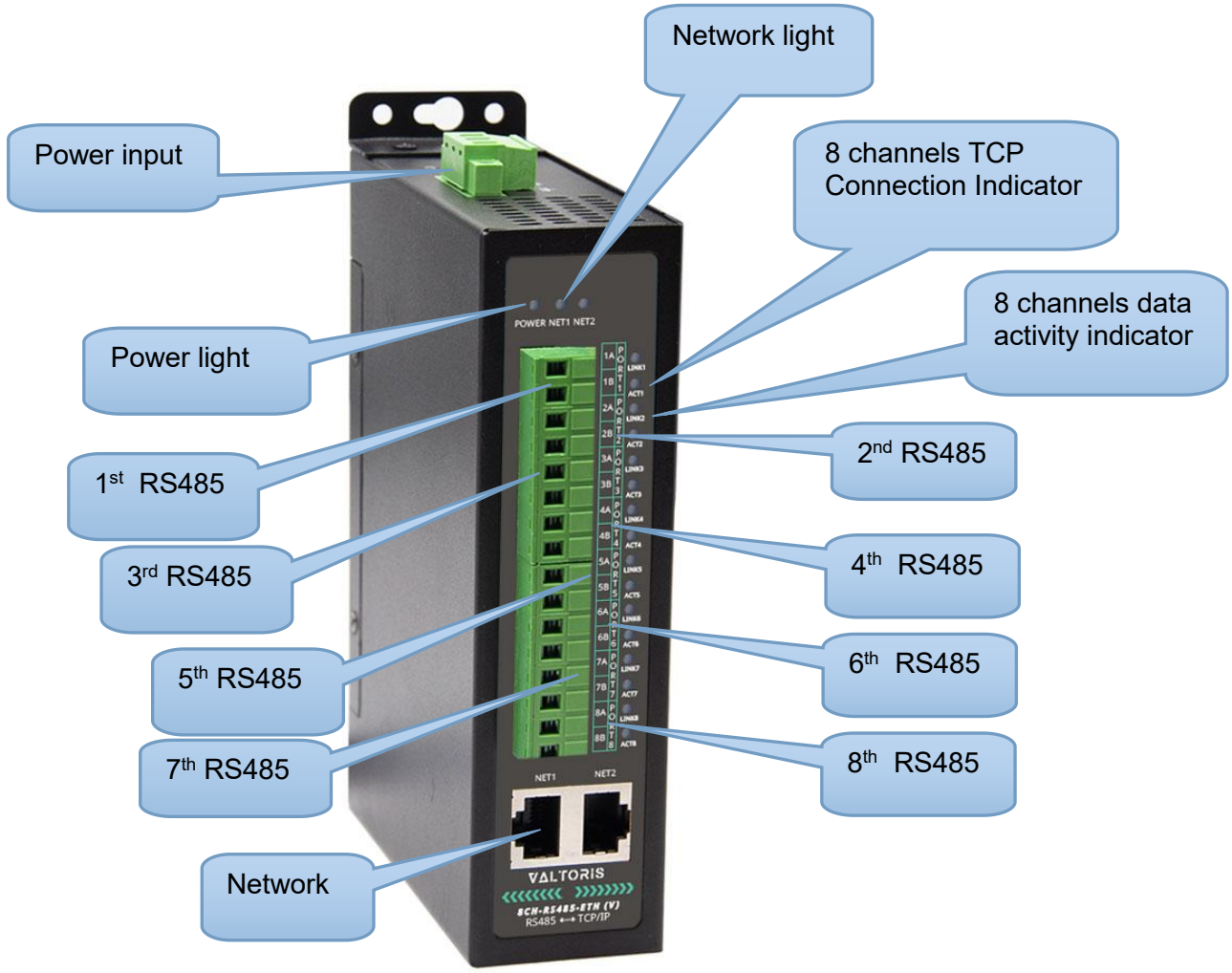


Figure 3 Front View of 2 CH-RS485-ETH (V)



Front View of 4 CH-RS485-ETH (V)(VI)



Front View of 8 CH-RS485-ETH (V)(VI)

The front view of the 2/4/8 CH-RS485-ETH(V)(VI) serial port server is shown in the figure above. The outer shell is made of black radiation-resistant SECC metal shell.

1. Power input: terminal: 5.08mm terminal, V+ is connected to 9~24V for 2CH-RS485-ETH (V) , 4CH-RS485-ETH (V) and 8CH-RS485-ETH (V); V+ is connected to 18~36V for 4CH-RS485-ETH (VI) and 8CH-RS485-ETH (VI). V- is connected to GND, and there is also the shell ground.

2. RS485: use RS485 to connect 1B and 1A, where 1B represents the negative line of 485 and 1A represents the positive line of 485; 485 with load of 32. The longest communication distance is 1200 meters. Generally RS485 line more than 300 meters when it is necessary to use the terminal resistance, 485 terminal resistance is 120 ohms.

3. Network interface: Connect the network cable; automatic crossover is supported.

The indicator lights of 4CH-RS485-ETH (V)(VI) and 8CH-RS485-ETH (V)(VI) corresponding to the main network port and the level network port are NET1 and NET2 respectively. There is no difference between the two network ports in other aspects.

4. **Indicator lights:**

- **2CH-RS485-ETH (V)** divided into Power, NET, Link, ACT, and RXD lights, which respectively represent power supply, network cable connection light, connection indication, data indication from the network port to the serial port, and data indication from the serial port to the network port.

Table 2 meaning of indicator light

Indicator Light	
POWER	Power indicator light. Red indicates that it has been powered on.
NET	The network cable is connected and the indicator light shows orange.
Link1/2	When the TCP connection is established (or in UDP mode), the Link is green. It can be used to determine whether the serial port server has established a communication link with the host computer software.
ACT1/2	The indicator light is green when the network port sends data to the serial port and the serial port sends data to the network port.
RXD1/2	When the serial port sends data to the network port, the indicator light is blue.

Debugging communication method using indicator lights:

1) If the NET light is not orange, the network cable is not connected properly. Please check the network cable.

2) If the Link light is not green (only TCP working mode is considered), the host computer software has not established a connection with the serial port server. Please consider whether the IP address is configured in the same network segment.

3) If the ACT light is not green, there is no data communication.

4) If the RXD light is not blue, the serial port data is not returned.

- **4CH-RS485-ETH (V)(VI) and 8CH-RS485-ETH (V)(VI)** divided into POWER, NET, LINK and ACT lights, representing power supply, network port indicator light, connection indicator and data indicator respectively.

Indicator Light	
POWER	Power indicator light. Red indicates that it has been powered on.
NET1/2	1) **Not lit:** there is no network cable connection between network Port 1 and Network Port 2. 2) **Green**: Solid Green: Normal network access

LINK	1) **Not lit:** TCP has not been established. 2) **Green**: When a TCP connection is established (or in UDP mode), the Link is green. It can be used to determine whether the serial server has established a communication link with the upper computer software.
ACT	1) **Green:** When the network port sends data to the serial port, the indicator light is green. The flashing time is delayed by one second compared to the actual communication time, making it more convenient to detect short data communications. 2) **Blue**: When data is sent from the serial port to the network port, both the blue and green indicator lights are on simultaneously. Since blue is relatively bright, if you see blue, it indicates that there is data returning from the serial port to the network port. This can determine whether the device responds to the commands of the upper computer. If there is no response, it indicates that the serial port baud rate is incorrect or the serial port is not properly connected.

Debugging communication method using indicator lights:

1) If the NET light is not green, it indicates that the network cable is not properly connected. Please check the network cable.

2) If the Link light is not green (only considering the TCP working mode), the upper computer software has not established a connection with the serial server. Please consider whether the IP addresses are configured in the same network segment.

3) If the Active light is green, it indicates that data is sent from the network port but no serial port device is returning data. Please check whether the baud rate is configured properly and whether the positive and negative terminals of RS485 are connected in reverse.

The installation of mounting brackets on the back panel of the device and the fixed installation of wall mounting brackets

Figure 4 Guide rail buckle

4.2 Hardware Connection

Generally speaking, the serial port server only needs to connect the power supply, serial port, and network cable.

The power supply can be an on-site 2-wire power supply, which can be directly connected to the positive and negative terminals of the power supply.

The serial port needs to be connected according to the user's serial port device. If you need to connect the first 485 port, just connect the positive terminal of 485 to 1A and the negative terminal of 485 to 1B.

The network port is connected to an ordinary network cable, which can be directly connected to the computer or connected to the network through a switch.

4.3 Software Installation

Vircom can be used to configure the device IP and other parameters, also can create virtual serial port. If no need the virtual serial port function, you can download the free-installation version.

Table 3 Software Version

Software	Description
VirCom_en	English version configuration software
Virtual-serial-port	Virtual serial port control software

4.4 Parameter Configuration

After installing Vircom, the hardware also connecting, run Vircom software and click "Device Manage" as figure 6. Use Vircom can search and configure the device parameter in different segment, which is very convenient as long as the device and computer of running Vircom are under the same switch.

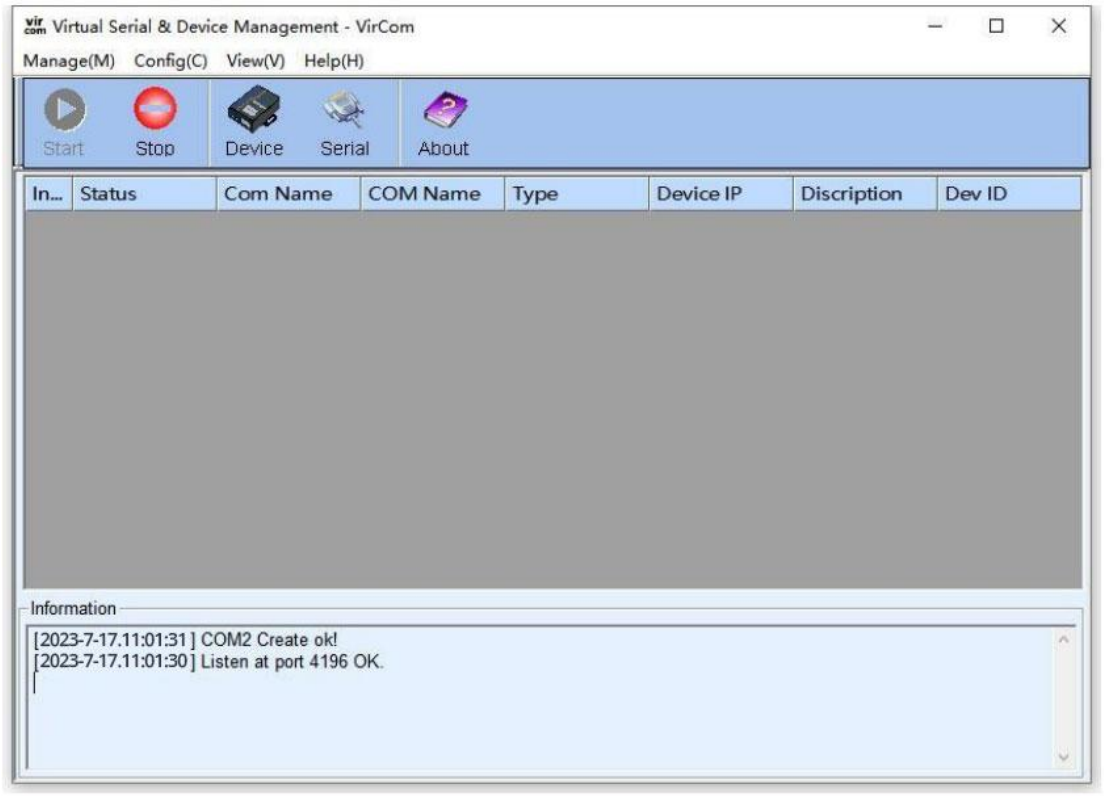


Figure 6 Vircom Main Interface

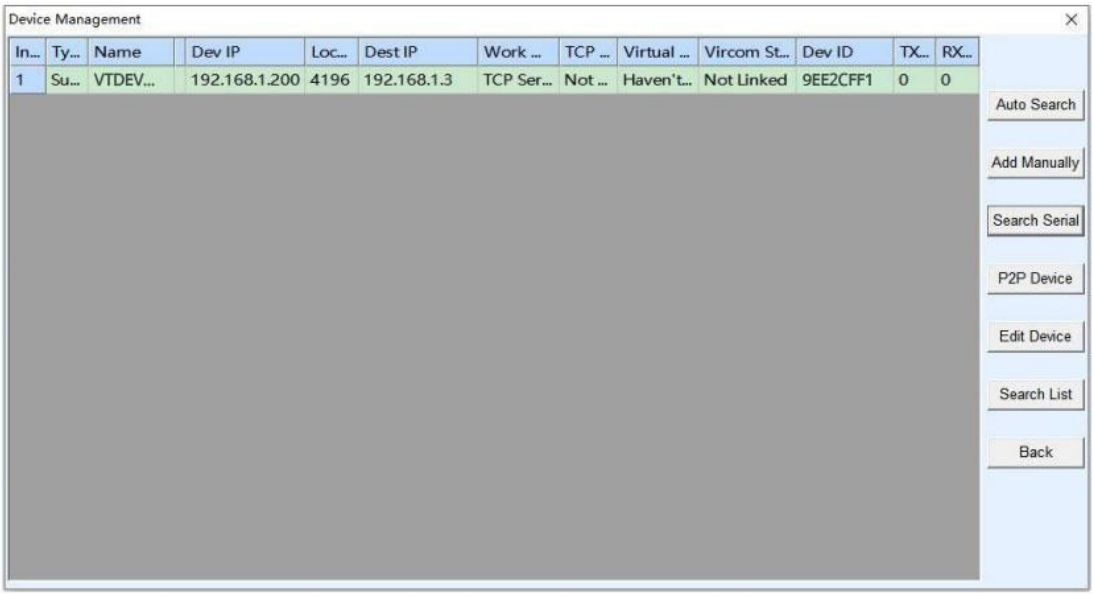


Figure 7 Device List

You can see all the currently online devices from the device list. Click "Edit Device" to configure the parameters.

4CH-RS485-ETH(V)(VI) has 4 lines devices list from 192.168.1.200 ~ 192.168.1.203

8CH-RS485-ETH(V)(VI) has 8 lines devices list from 192.168.1.200 ~ 192.168.1.207

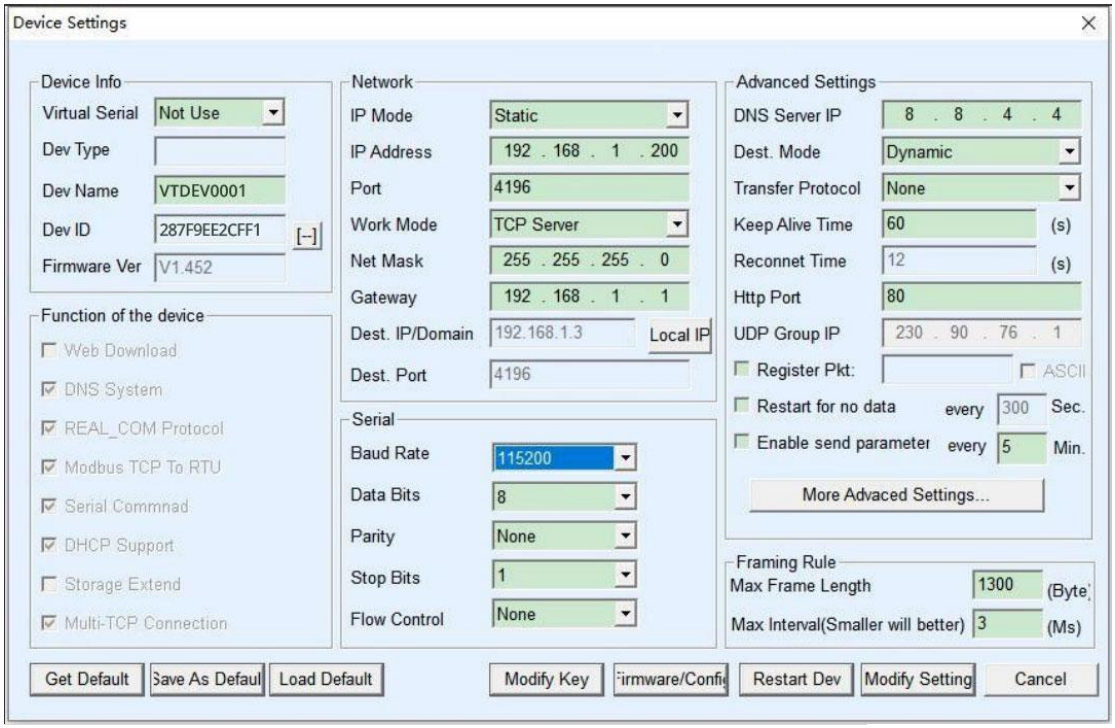


Figure 8 Device Parameters

In this interface, the user can set the parameters of the device, and then click "Modify Settings", the parameters will be set to the device's flash and will not be lost when the power is off. At the same time, the device will automatically restart.

The main parameters configured here are: baud rate, data bit, and check bit in the serial port settings; IP address, subnet mask, and gateway in the network settings; sometimes, according to the computer software, it is also necessary to configure the working mode of the serial port server.

The detailed meanings of other parameters are as follows:

Table 4 Parameter Meaning

Parameter	Value Range	Meaning
Virtual Serial	Port Unused, created virtual serial port	You can bind the current device to a created virtual serial port. Please add a COM port in "Serial Port Management" on the main interface first.
Device model		Only display the core module model

Device name	Any	You can give the device an easy-to-read name with a maximum length of 9 bytes.
Device ID		Factory unique ID, cannot be modified.
Firmware Version		Core module firmware version
Supporting Function		Refer to Table 6 for the functions supported by the device.
IP Mode	Static, DHCP	Users can choose static or DHCP (dynamic IP)
IP Address		IP address of the serial device server
Port	0～65535	The listening port of the serial port server when it is in TCP Server or UDP mode. When acting as a client, it is best to specify port 0, which is beneficial to improve the connection speed. When using port 0, the system will randomly assign a local port. The difference between this and non-zero ports is: (1) When the local port is 0, a new TCP connection is established with the PC when the module restarts. The old TCP connection may not be closed, and there may be multiple false connections with the device. Generally, the host computer hopes to close the old connection when the module restarts; specifying a non-zero port will close the old connection. (2) When the local port is 0, the TCP re-connection time is faster. When the serial port server is in TCP client mode, it also acts as a TCP server to listen for connections on the port. At the same time, the local port number used by the TCP client to connect to the server is "port + 1".
Working Mode	TCP server mode, TCP client mode, UDP mode, UDP multicast	When set as a TCP server, the serial device server waits for the computer to connect; when set as a TCP client, the serial device server actively initiates a connection to the network server specified by the destination IP.
Sub-net Mask	Eg: 255.255.255.0	Must be the same as the subnet mask of the local area network.
Gateway	Eg: 192.168.1.1	Must be the same as the local LAN gateway.
Destination IP or domain name		In TCP client or UDP mode, data will be sent to the computer indicated by the destination IP or domain name.

Destination Port		In TCP client or UDP mode, data will be sent to the destination port of the destination IP.
Baud Rate	1200、2400、4800、7200、 9600、14400、19200、 28800、38400、57600、 76800、115200、230400、460800、921600	Serial port baud rate
Data Bits	5、 6、7、8、 9	
Check digit	None, Even, Odd, Mark, Space	
Stop Bits	1、2	
Flow Control	No flow control, hard flow control CTS/RTS, hard flow control DTR/DCR, soft flow control XON/XOFF	Only valid for RS232 serial port
DNS Server		When the destination IP is described by domain name, you need to fill in the DNS server IP. When the IP mode is DHCP, you do not need to specify the DNS server, it will be automatically obtained from the DHCP server.
Destination mode	Static, Dynamic	In TCP client mode: When using static destination mode, the device will automatically restart after 5 consecutive failures in connecting to the server.
Conversion agreement	NONE、Modbus TCP、Real_COM	NONE means that data forwarding from the serial port to the network is transparent; Modbus TCP will directly convert the Modbus TCP protocol into the RTU (ASCII) protocol to facilitate coordination with the Modbus TCP protocol; RealCOM is designed to be compatible with the old version of the REAL_COM protocol and is a virtual serial port protocol. However, when using a virtual serial port, it is not necessary to select the RealCom protocol.

Keep-alive time	0～255	Heartbeat interval. (1) When set to 1-255, if the device is in TCP client mode, it will automatically send TCP heartbeats every "keep alive time". This can ensure the TCP validity of the link. When set to 0, there will be no TCP heartbeat. (2) When set to 0-254, when the conversion protocol is selected as REAL_COM protocol, every keep alive time, the device will send a data with a length of 1 and content of 0 to implement the heartbeat mechanism in the Realcom protocol. When set to 255, there will be no realcom heartbeat. (3) When set to 0-254, if the device is working in TCP client mode, the device will send device parameters to the destination computer every keep alive time. When set to 255, there will be no parameter sending function, which can realize remote device management.
Disconnection re-connection time	0～255	In TCP client mode, when the connection is not successful, each "disconnection re-connection time" will re-initiate a TCP connection to the computer. It can be 0 to 254 seconds. If it is set to 255, it means that it will never reconnect. Note that the first TCP connection (such as hardware power-on, device restart through Vircom software, no data light) will generally be made immediately. Only after the first connection fails will it wait for the "disconnection re-connection time" and try again, so the "disconnection re-connection time" will not affect the connection establishment time under normal circumstances between the network and the server.
Web access port	1～65535	The default is 80
Multicast address		Used for UDP multicast
Enable registration package		When the TCP connection is established, the registration packet is sent to the computer. After enabling the registration packet, the realcom protocol must be selected. Supports TCP server and TCP client modes.

Data packet length	1～1400	One of the serial port framing rules. After receiving data of this length, the serial port server sends the received data as a frame to the network.
Data packet interval	0～255	Serial port framing rule 2: When the data received by the serial port of the serial server pauses and the pause time is greater than this time, the received data will be sent to the network as a frame.

The functions supported by the device are explained as follows:

Table 5 Device Supported Function

Name	Description
Domain name system	The destination IP can be a domain name (such as a server address starting with www).
REAL_COM protocol	A non-transparent serial port server protocol, suitable for multiple serial port servers to bind virtual serial ports through the Internet. Because the protocol contains the device MAC address, it helps the host computer to identify the device. It can be not used in general.
Modbus TCP to RTU	It can realize Modbus TCP to RTU conversion and also supports multi-host function.
Serial port modification parameters	Supports serial port AT commands to configure and read device parameters.
Obtain IP automatically	Support for DHCP client protocol
Storage extension EX function	Subsequent expansion
Multiple TCP connections	Supports more than 1 TCP connection when acting as a TCP server.
IO port control	Supports any custom instructions to control 8 IO outputs.
UDP multicast	UDP multicast
Multi-destination IP	As a TCP client supports simultaneous connection of 7 destination IP.

4.5 TCP communication test

After configuring the device parameters, you can use serial port tools and TCP debugging tools to perform TCP connection communication tests.

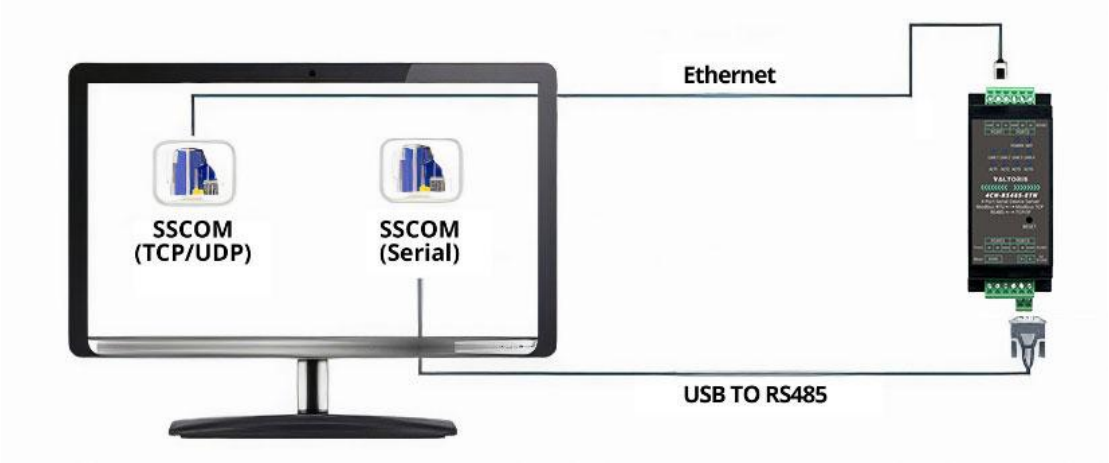


Figure 9 TCP communication diagram

The serial port server's serial port to network port, network port to serial port data transparent forwarding function.

Assuming that the COM port (USB TO RS485) of the PC is connected to the serial port of the serial server, then open the serial debugging assistant window and open the corresponding COM port, as shown in the figure below:

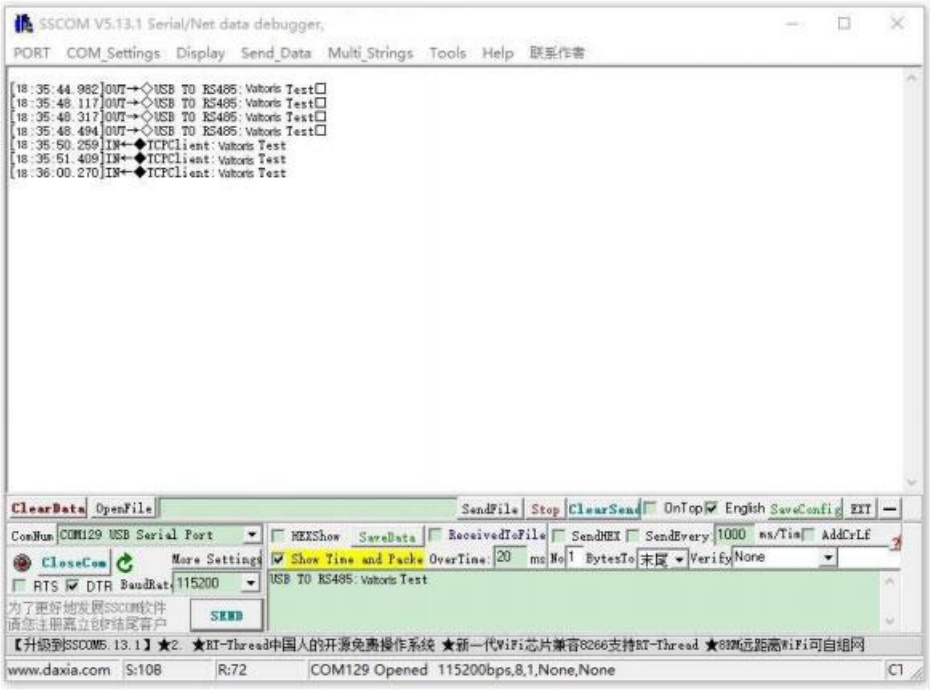


Figure 10 SSCOM1: USB to RS485 transceiver interface

In addition, open one more serial port debugging assistant window and use it as a TCP client mode. Fill in the destination IP as the IP of the serial server (currently

192.168.1.200) and the destination port as 4196, and then click the "Open" button, as shown in the figure below:

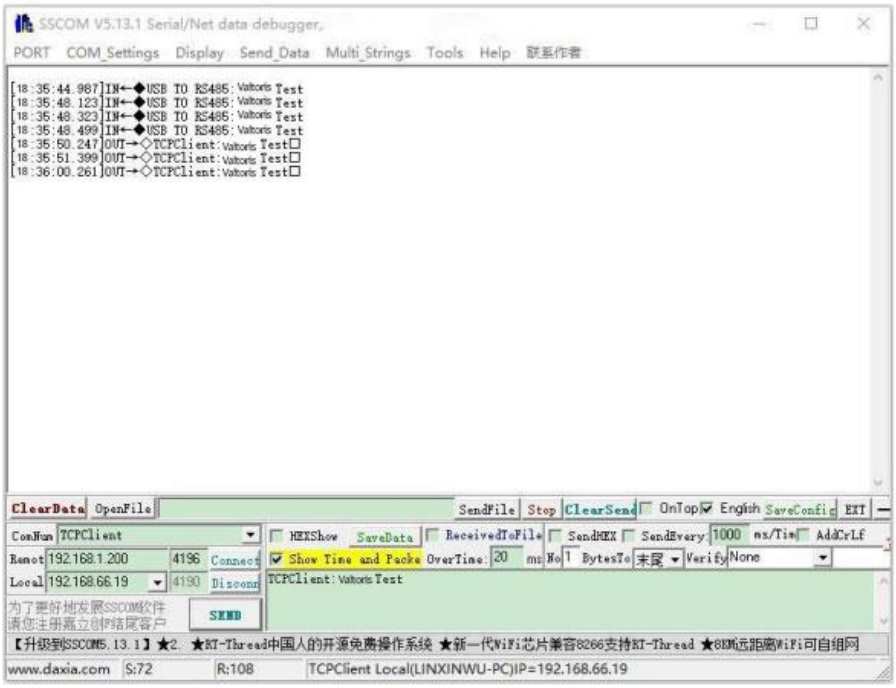


Figure 11 SSCOM2: TCP Client transceiver interface

Enter "TCP Client: “Valtoris Test" in the serial debugging assistant SSCOM2 set as TCPClient and click to send, the data will be transferred to the RS485 interface through the serial server network port, next it will be sent to the USB TO RS485, and finally be displayed in the serial debugging assistant SSCOM. In turn, input "USB TO RS485: Valtoris Test" in SSCOM1, and click to send to SSCOM2 and display it.

4.6 Virtual serial port test

The SSCOM2 in the figure communicates directly with the serial server via TCP. In order to allow the serial software that has been developed by the user to communicate with the serial server, a virtual serial port needs to be added between the user’s program and the serial server. As shown in the figure, Vircom and the user’s program run on the same computer, and Vircom virtualizes a COM port, so that this COM port corresponds to the serial server. When the user program opens the COM communication, it can be sent to the user serial device through the Vircom serial server. The following demonstrates this operation step:

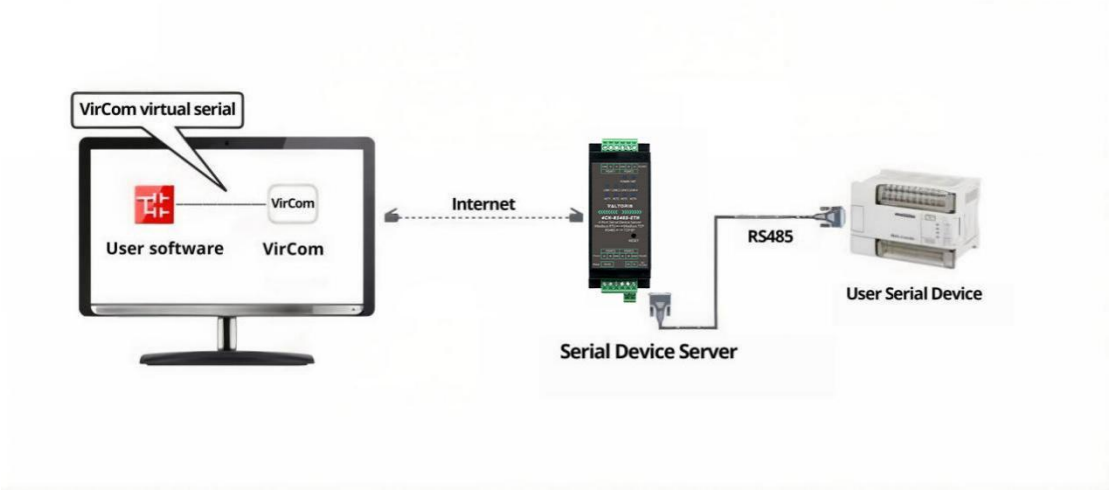


Figure 12 Function of virtual serial port

Click "Serial Port Management" on the Vircom main interface, then click "Add", and select to add COM2, where COM2 is the COM port that does not originally exist on the computer.

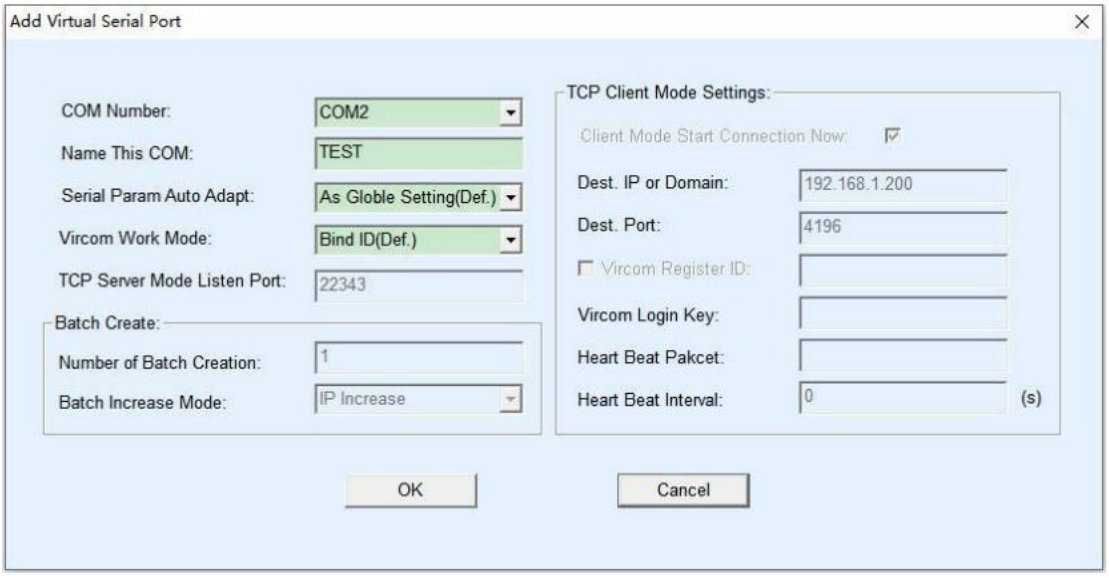


Figure 13 Add Virtual Serial Port

Then enter the “device manage”, and double-click the device that you need to bind to the COM2. As shown in Figure 13, select COM2 from the "virtual serial port" list in the upper left corner. Then click "modify Settings". And return to the main interface of Vircom. You can see that the COM2 has been connected to a device with IP 192.168.1.200. You can use COM2 instead of SSCOM2 to communicate.

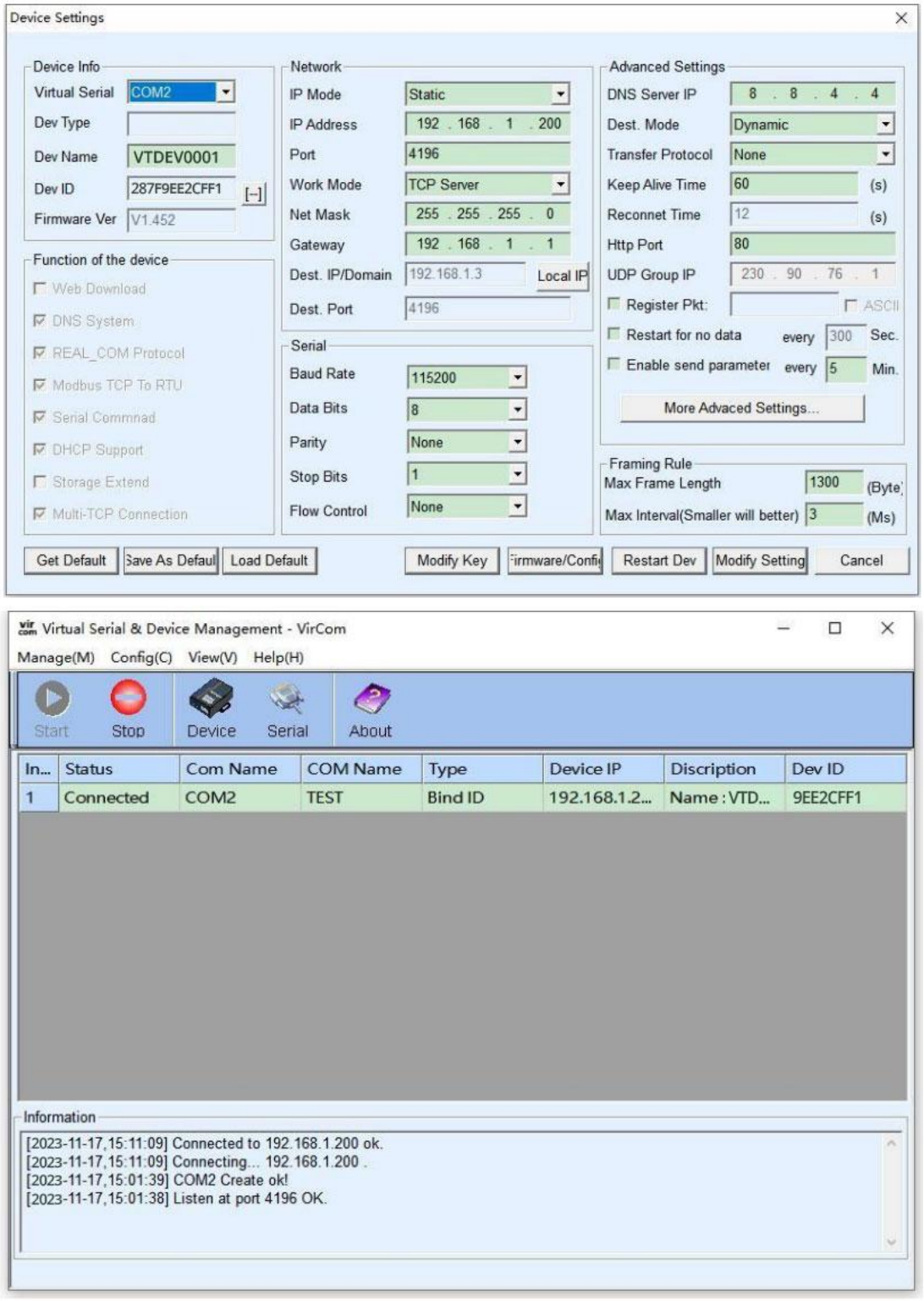


Figure 14 Virtual Serial Port has been connected

Open the SSCOM simulated user serial port program, open COM2 (the virtual serial port in the figure above), then open an SSCOM simulated serial port device, and open COM3 (hardware serial port). At this time, the COM2 data transmission link is as follows: COM2 - Vircom - serial port server network port - serial port server serial port - COM3.

Conversely, data can also be transmitted from COM3 to COM2: COM3 - serial port server serial port - serial port server network port - Vircom - COM2. The figure below shows the data transmission and reception of both parties.

If COM3 is replaced with a user serial port device, COM2 can communicate with the user device.

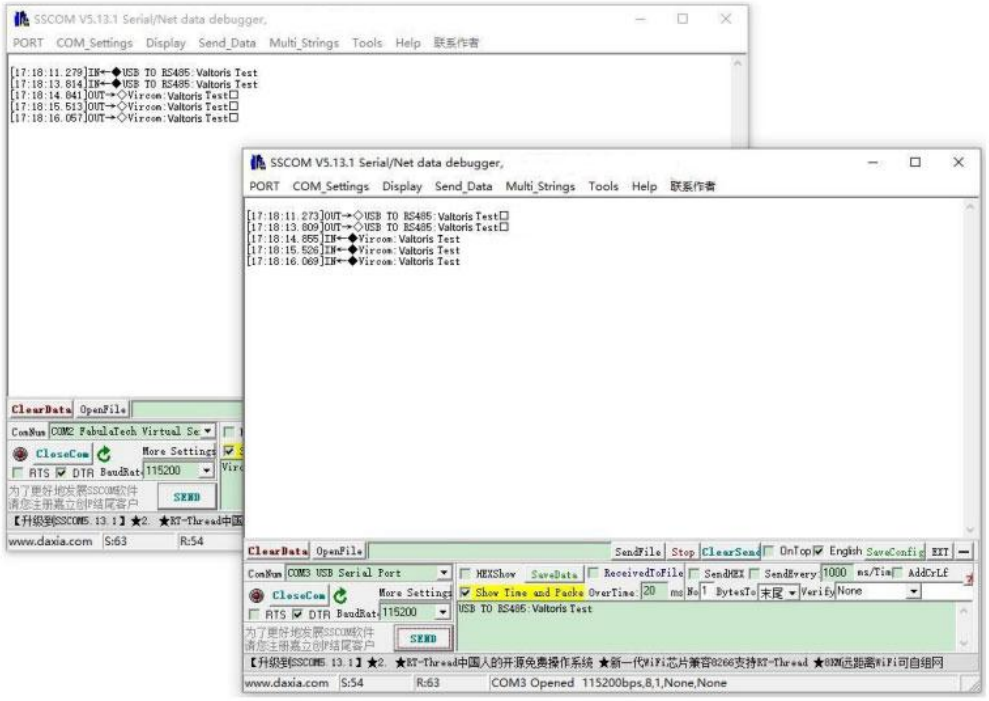


Figure 15 Communication via Virtual Serial Port

4.7 Modbus TCP test

By default, the data of the serial and network port is transparently transmitted. If you need to convert Modbus TCP to RTU, you need to select "Modbus_TCP Protocol" as the conversion protocol in the device setting, as shown in the figure 16 below.

At this time, the device port is automatically changed to 502, the user's Modbus TCP tool is connected to the 502 port of the serial server's IP, the Modbus TCP commands that have been sent will be converted into RTU commands and output via the serial port.

For example, if the serial port server receives the Modbus TCP command of 00 00 00 00 00 06 01 03 00 00 0a, the serial port outputs the command of 01 03 00 00 00 0a c5 cd.

Note: The serial port may send multiple commands of 01 03 00 00 00 0a c5 cd. This is because the default Modbus adopts the storage mode and will automatically train and query commands in turn. How to switch to non-storage mode will be explained later.

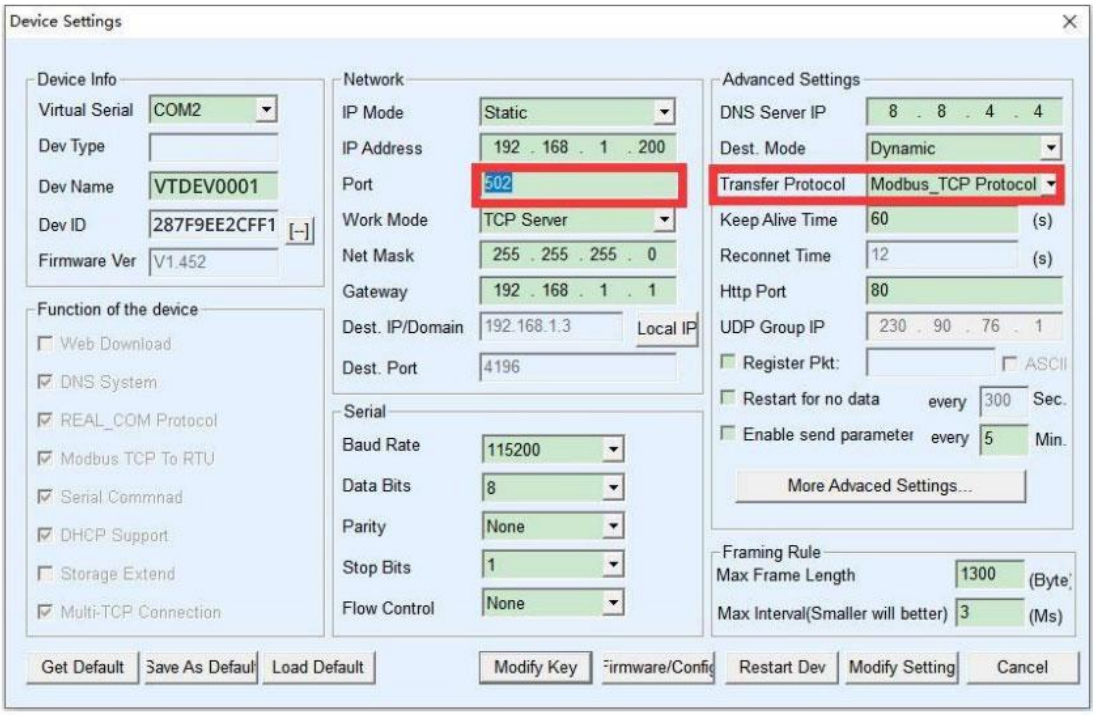


Figure 16 Enable Modbus TCP

If the user's Modbus TCP software is used as a slave, you need to change the working mode to client based on the conversion protocol, and change the destination IP to the IP of the computer where the Modbus TCP software is located, and the destination port is 502, as shown in the figure 17 blow.

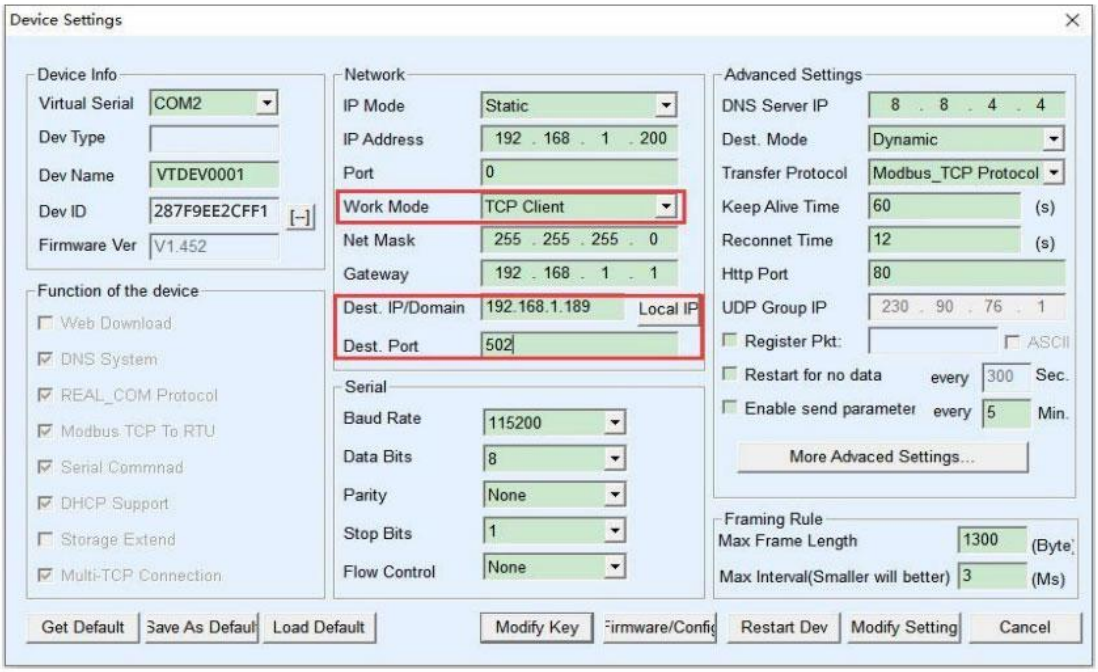


Figure 17 Modbus TCP as client

4.8 Web Configuration

Using Vircom, you can search and configure device parameters in different network segments. Web configuration requires that the computer and the serial server are in the same IP segment first, and the IP address of the serial server must be known in advance. However, Web configuration can be performed on any computer without Vircom.

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

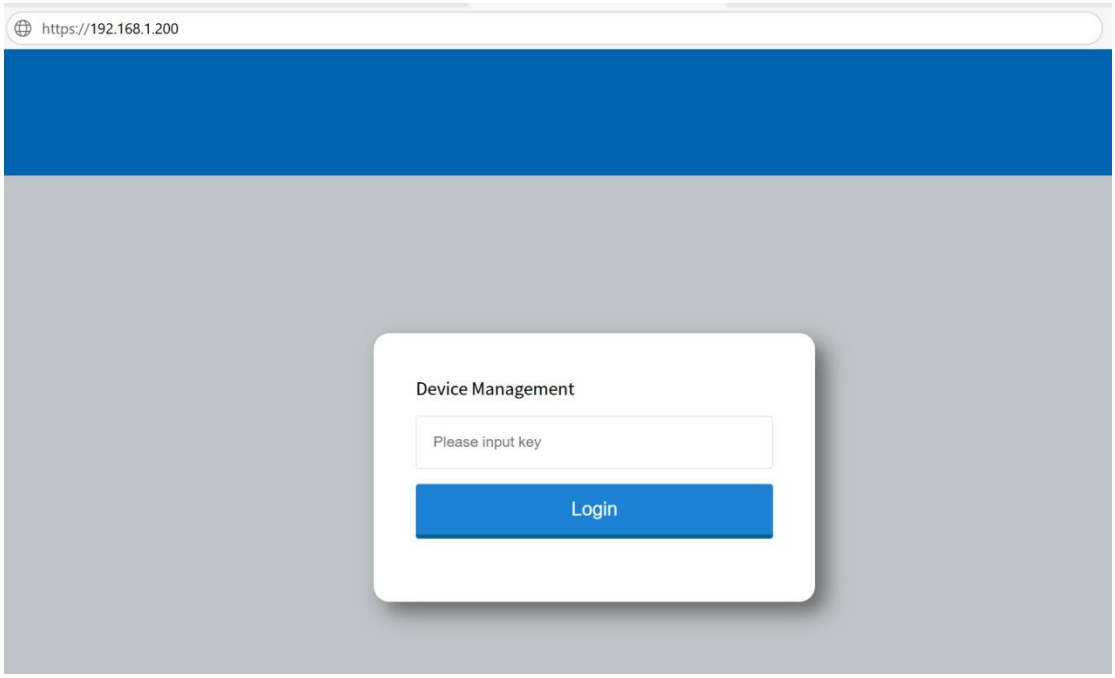


Figure 18 Web login interface

2. Enter Password in Password: default is 123456. Click the login button to login.

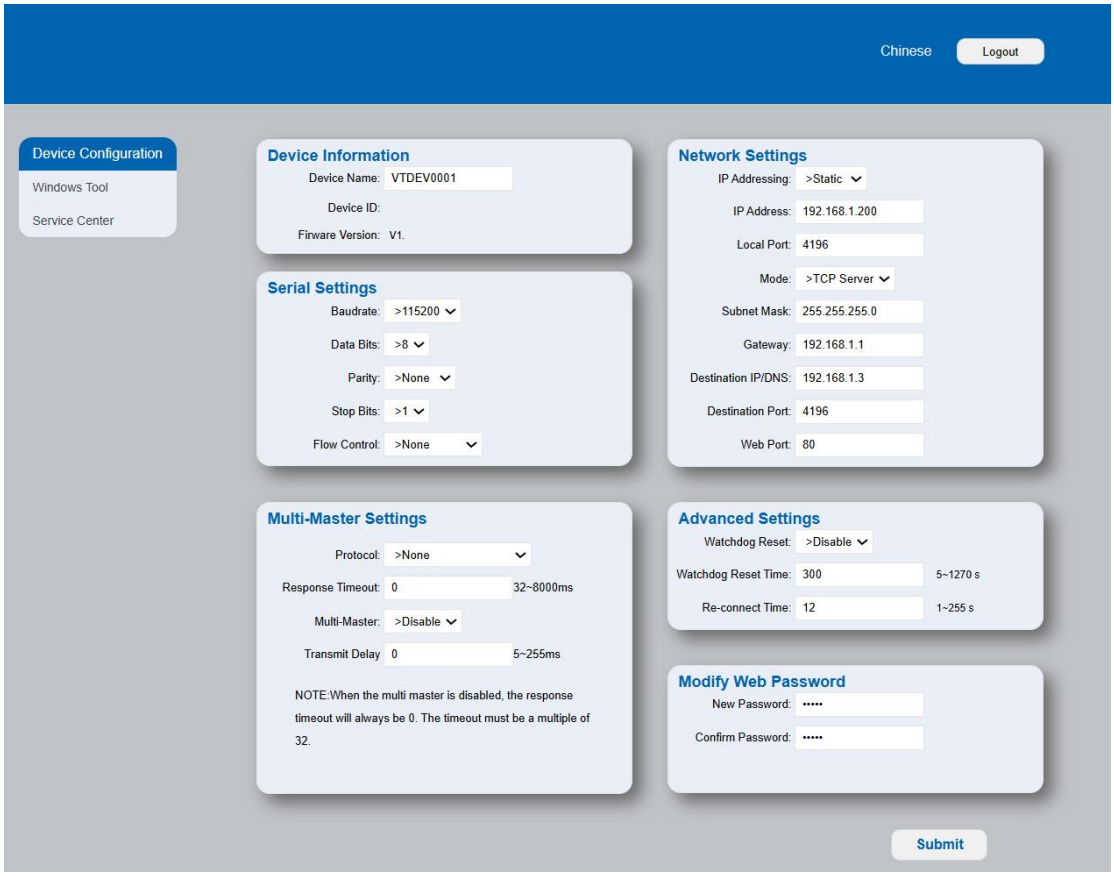


Figure 19 Web configuration interface

3. You can modify the serial port server parameters on the web page that appears. For related parameters, refer to Table 5 Parameter Meaning.

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

5. Work Mode and Transfer Protocol

Different serial server working modes and conversion protocols can be selected in different application occasions, so that it can be used more stably and reliably, which will be described in detail below.

The use of the serial port server is basically divided into two types: with virtual serial port and non-virtual serial port, as shown in Figure 9 TCP communication diagram and Figure 12 the function of virtual serial port. The virtual serial mode is for the serial interface (COM) on the user software, that is, the user software and devices are serial ports. The user software with the non-virtual serial port mode directly adopt TCP/IP communication, but the user devices still use the serial port.

In non-virtual serial port mode, the "conversion protocol part" is divided into three modes: transparent transmission, Modbus TCP to RTU and Realcom protocol. If the user software is a fixed protocol Modbus TCP protocol and the lower computer is Modbus RTU/ASCII, you need to select the Modbus TCP mode; Realcom protocol is currently

only used when the multi-serial port server is used as a TCP client to connect to a server and the server uses a virtual serial port.

Usage is summarized as follows:

Table 6 network configuration modes

No.	Virtual Serial Port	Device Working Mode	Conversion protocol	Description
1	Use	TCP Server	None	Suitable for situations where user software opens the COM port to actively collect data.
2	Use	TCP Client	None	Suitable for the occasion where the device actively sends data. If you choose TCP server, the device may not be able to reconnect after disconnection.
3	Do not use	TCP Server	Modbus TCP to RTU	Applicable to the case where the user software is Modbus TCP, the user device is Modbus RTU and Modbus TCP is the master station.
4	Do not use	TCP Client	Modbus TCP to RTU	Applicable to the user software is Modbus TCP, the user device is Modbus RTU, and the Modbus RTU is the master station.
5	Use	TCP Client	Realcom Protocol	When the multi-serial port server acts as a TCP client and uses a virtual serial port, it is best to use the Realcom protocol.
6	Do not use	TCP Client	None	Suitable for a large number of devices connected to a cloud. In general, the cloud is a server with a public IP on the Internet.
7	Do not use	TCP Server	None	Applicable when the device and computer are in the same local network, monitoring is done locally without the need to communicate across the Internet.

5.1 Virtual Serial Mode

If the user software uses the COM port for communication, it must use the virtual serial port mode including some PLC software, configuration software, instrument software, etc.

Check if the monitoring computer and the device are both in the local network:

a) If the computer is a server with a public IP leased on the Internet, then the device must use the TCP client method to connect the device to the server. At this time, you can choose ② and ⑤ in Table 5. If it is a multi-serial port server, you must choose ⑤.

b) If they are all in the local network (can ping each other), see whether the host computer actively queries or the device actively sends data. If the device actively sends, you must use the device as a TCP client method ②, otherwise you can choose method ①.

5.2 Direct TCP/IP Communication Mode

If neither Modbus TCP protocol conversion nor the virtual serial port are required, the user software may directly communicate with the network port of the serial server by TCP/IP communication, and the serial server converts TCP/IP data into serial data and sends it to the serial device.

Generally, the above operation is adopted by the users that develop the upper computer to communicate by themselves, which integrates the analysis of the device's serial communication protocol. This method is more flexible and efficient than the virtual serial port. Correspond to ⑥ and ⑦ in Table 6.

The section "4.5 TCP Communication Test" briefly describes how to communicate when the serial port server is used as a TCP server. Here we will describe how TCP client, UDP mode, and multiple TCP connections to communicate with computer software. Among them, the computer software takes SSCOM (serial port debugging, software that imitates user TCP/IP communication) as an example.

The serial server complies with the standard TCP/IP protocol, so any network terminals that complies with this protocol can communicate with the serial server. For any two network terminals (here, the network debugging tool and the serial server) to be able to communicate, their parameter configuration must be paired.

5.2.1 TCP Client Mode

There are two working modes in TCP mode: TCP server and TCP client. Regardless of which mode is adopted, one must be the server and the other must be the client. After

that, only the client can access the server. If they all are the servers or the clients, the communication is invalid.

When the serial server is used as a client, there must be three corresponding relationships, as shown in Figure blow.

(1) Working mode correspondence: The working mode of the serial port server is the server mode of the client corresponding to the network tool.

(2) IP address correspondence: The destination IP of the serial port server must be the IP address of the computer where the network tool is located.

(3) Port correspondence: The destination port of the serial server must be the local port of the network tool. After this setting, the serial server can automatically connect to the network tool, and data can be sent and received after the connection is established.

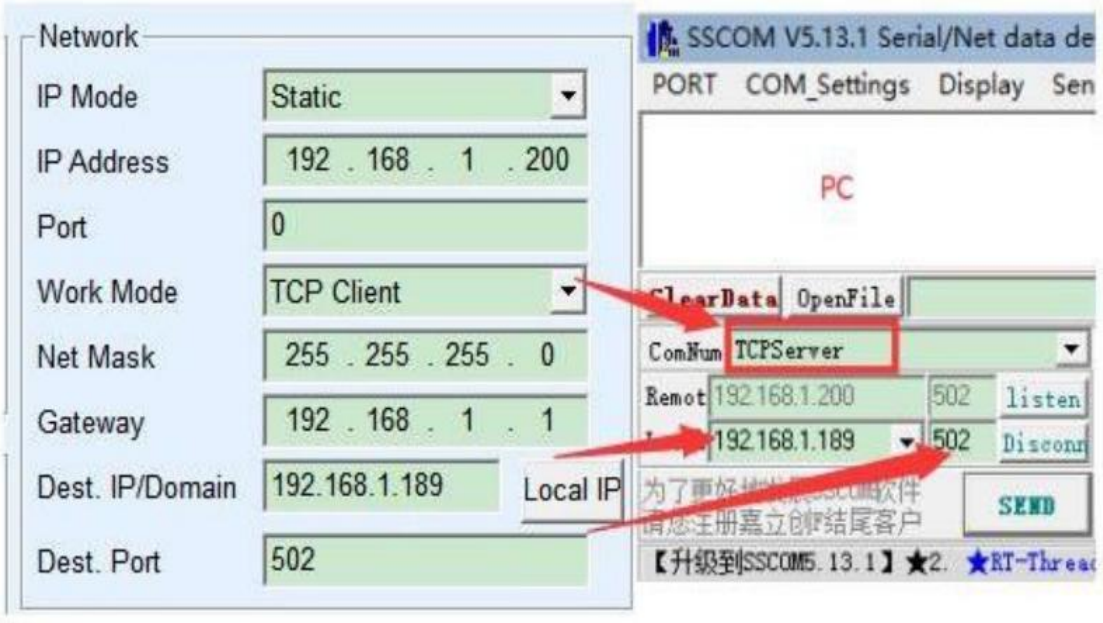


Figure 20 Serial server as a client

5.2.2 The Client connects to Multiple Servers

When the serial server is used as a TCP client, it can connect to 7 destination IP addresses at the same time, and the data sent by the serial port will be sent to 7 destination IPs too. If there are not so many servers, leave the rest of the destination IP vacant. The method is as follow：

Figure 21 The first destination IP and port

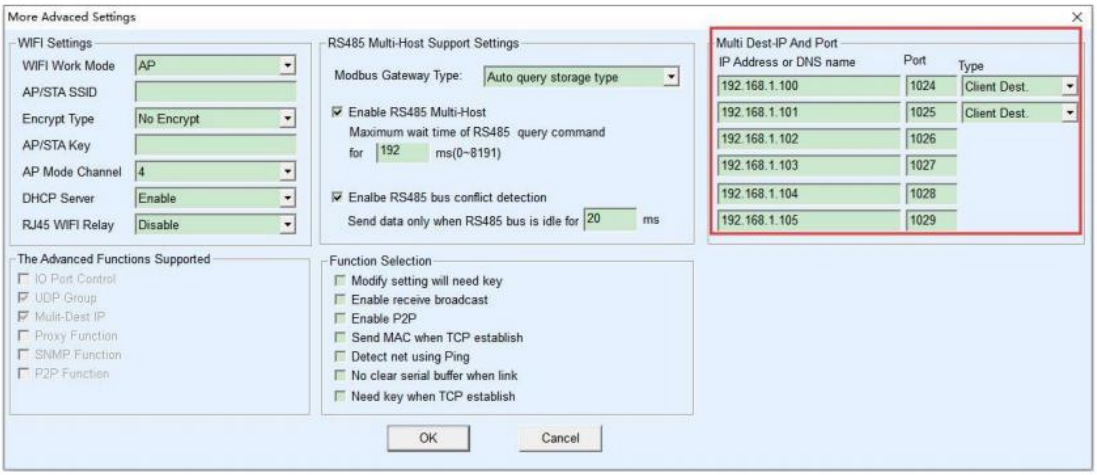


Figure 22 The remaining 2 to 7 IPs and ports

The first IP is set in the device setting interface as shown in Figure 21. The first IP can be a domain name. For the remaining 2~7 destination IPs, click the "More Advanced Setting..." button in the device setting interface to open More Advanced Settings for setting.

After all 7 destination IPs are set up, they can be connected automatically. If they are not connected, they will wait for the "Reconnect Time" time and then reconnect repeatedly.

5.2.3 TCP Server Mode

When the serial port server is used as the server, there are also three corresponding relationships, as shown in Figure 23, which will not be explained here. After setting, click the open button of the network tool to establish a TCP connection with the serial server, and then you can send and receive data after the connection is established.

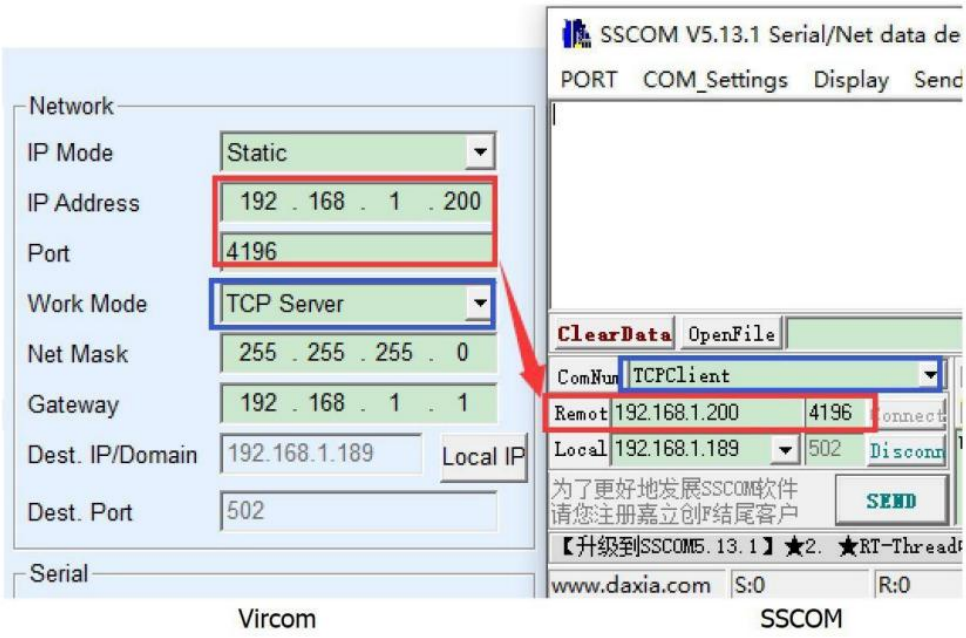


Figure 23 The serial port server as TCP Server

When the serial port server is used as a server, it can accept 30 TCP connections at the same time. The data received by the serial port will be forwarded to all established TCP connections. If you need to realize that data is only sent to the TCP that has recently received network packets, you need to enable the multi-host function, please refer to 7.4 Multi-host function.

5.2.4 Acting as both client and server

The serial server can accept TCP connections even when the device is in the TCP client mode, that is, it also has the TCP server function.

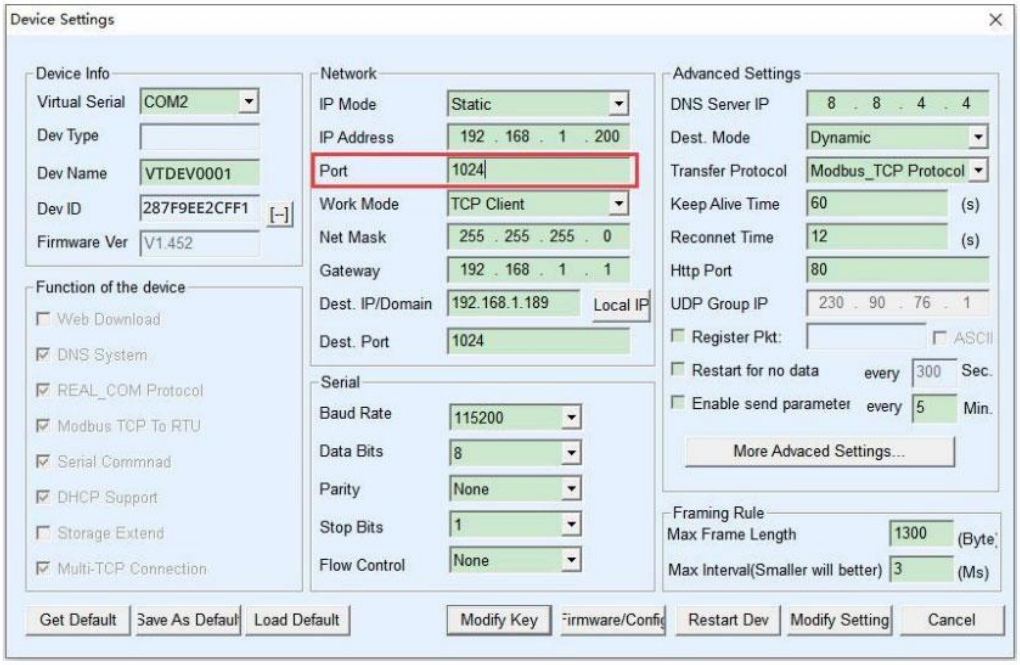


Figure 24 Acting as both Client and Server

By default, when using Vircom for configuration, if you change the working mode to "TCP client", the port (that is, the local port) will automatically become 0 (0 means a free port is randomly selected). In order to support the TCP server mode, the computer software must get the local port of the device, so a value needs to be specified here.

As shown in Figure 25, the computer software can now connect to the 1024 port of 192.168.1.200 for communication, and the device will also act as a client to connect the 1024 port of 192.168.1.189. It should be noted that since the local port 1024 is occupied by the server, the local port when used as a client is "port+1", that is, the software on 192.168.1.189 sees that the incoming port of the device is 1024+1 =1025

5.2.5 UDP Mode

In UDP mode, the parameter configuration is shown in Figure 25. The left side is the configuration for the serial server in Vircom, and the right side is the setting for the serial debugging tool SSCOM. First of all, both sides must be in UDP working modes. In addition, indicated by the red arrow, the destination IP and the destination port of the network tool must point to the local IP and local port of the serial server.

As indicated by the blue arrow, the destination IP of the serial server must be the computer IP where the network tool is located, and the destination port of the serial server must be the local port of the network debugging tool. Only after these network parameters are configured can two-way UDP data communication be ensured.

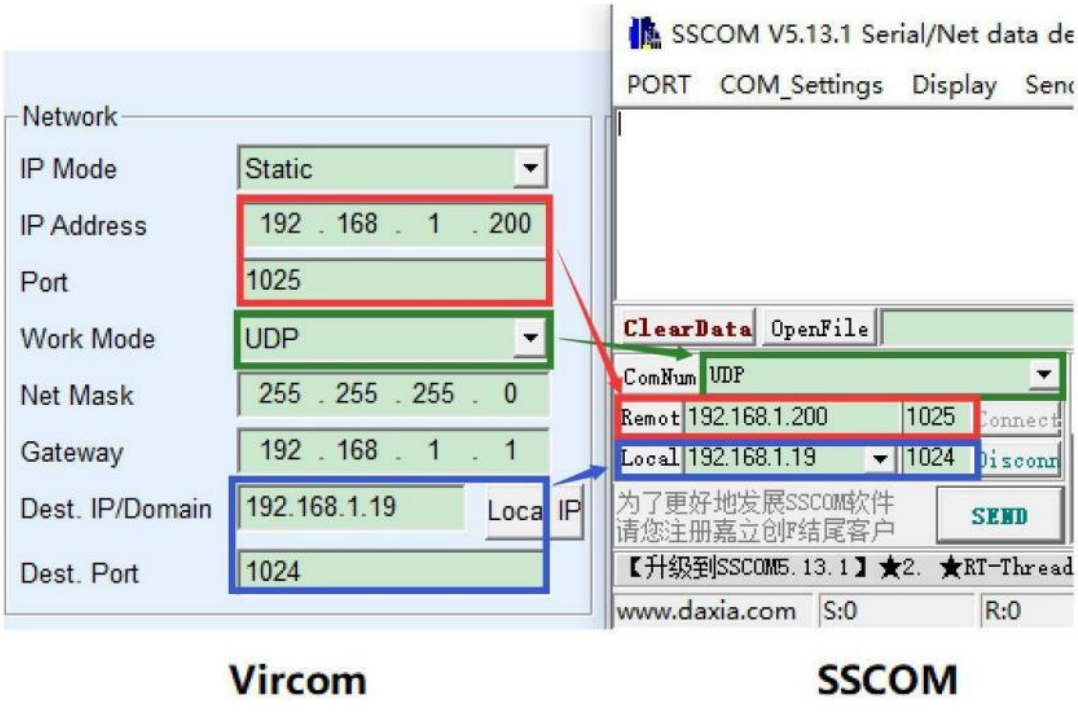


Figure 25 UDP mode parameter configuration

5.3 Device Connecting Method

If the host computer does not adopt the SSCOM program or the virtual serial port, but two devices are connected through a network port, the configuration method is similar. First, the user needs to connect two devices and one computer to the same local area network. The computer runs Vircom, and the purpose of connecting to the computer is only for configuration. After the configuration is completed, the computer does not need to be connected.

Click on Vircom's device management to find these two devices, as shown in Figure 28. Then click "Edit Device" to configure the device. Device connections can be divided into TCP and UDP. If it is TCP connection, the parameters of the two devices are shown in Figure 26. The parameters shown by the arrows must correspond like the way of connecting with a PC. After the TCP connection is successful, you can view the connection status by returning to the "Device Management" dialog box, as shown in Figure 27. If the statuses of the two devices are both "Connected", it means that the TCP link between the two devices has been established.

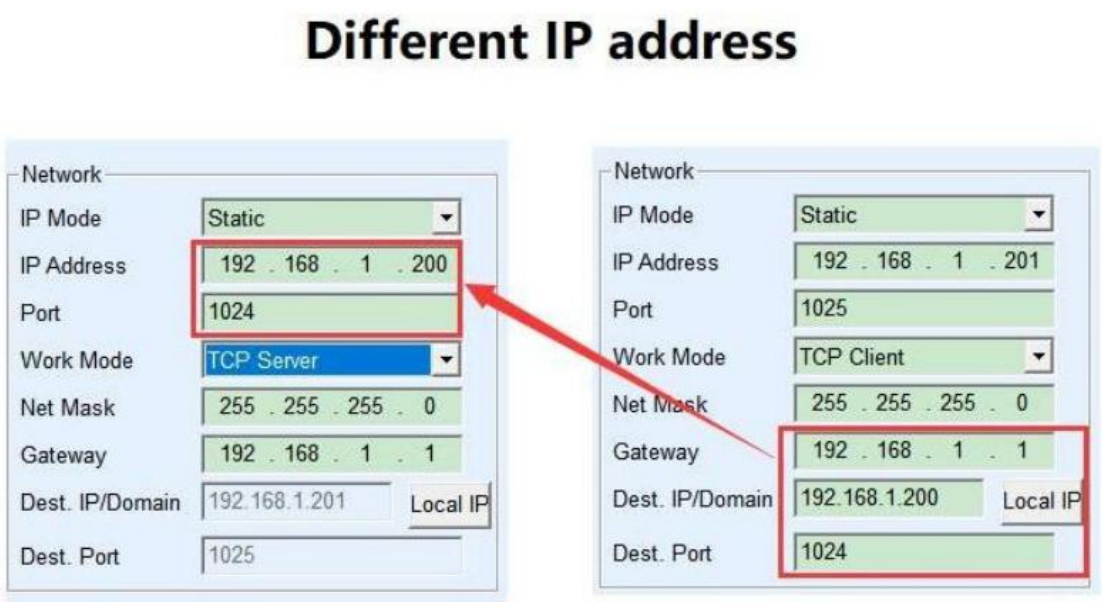


Figure 26 TCP device pairing parameter configuration

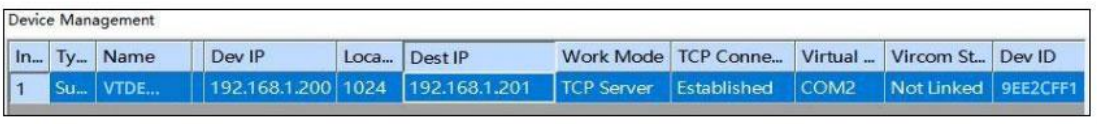


Figure 27 successful TCP device pairing check

If it is in UDP mode, the configuration parameters are shown in Figure 28, and the parameters corresponding to the arrows must be one-to-one. The UDP connection does not need to check the connection as long as the parameter configuration is correct, the sent data will be automatically sent to the designated device.

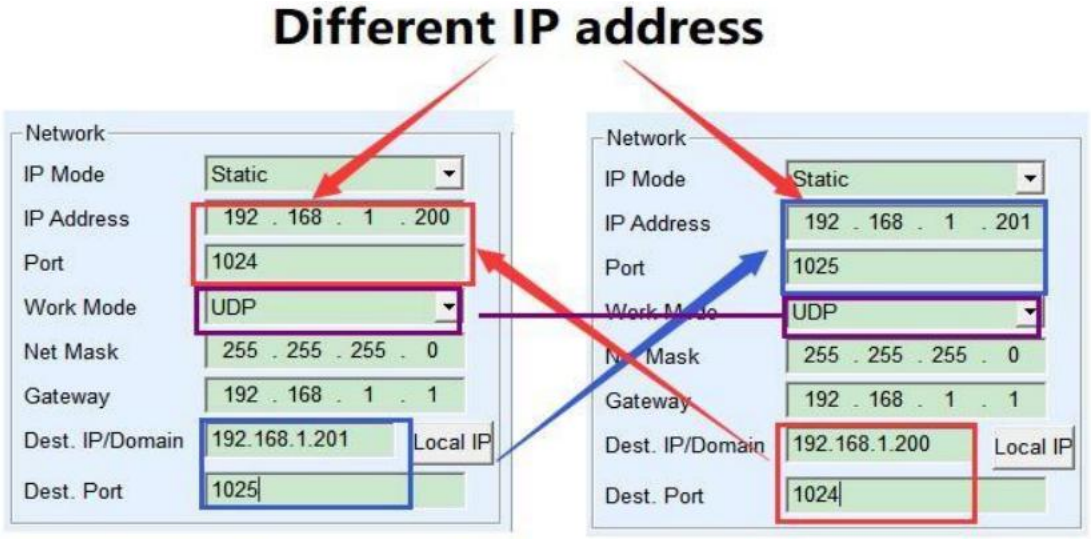


Figure 28 configuration of UDP device pairing parameters

Finally, we need to remind that if the device is connected, in addition to the network port parameters set according to the above, the correct serial port parameters must also be set. Mainly, the baud rate of the serial server needs to be consistent with the user device's. After setting, the user equipment can send data to each other through the serial ports of the two serial servers.

6. Device Debugging

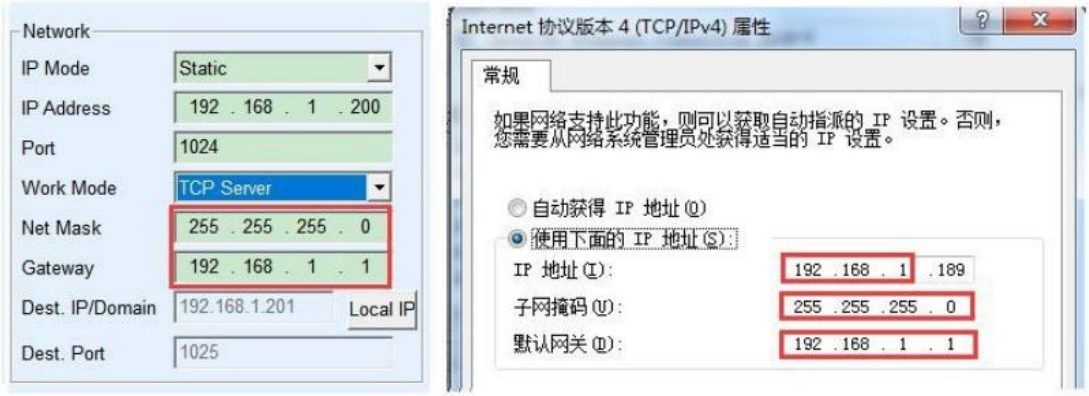
6.1 Network Physical Connection

The serial server can use a crossover cable or a network cable to connect to the switch or the computer network port directly.

After the connection is established, the first step is to check whether the Link light is green, otherwise, please check whether the network cable is connected.

6.2 Network TCP Connection

When the device is used as a dynamic IP, it cannot be directly connected to the computer network port. Because there is no DHCP server available (usually the DHCP server is the router in the LAN). Therefore, please specify the IP when connecting directly. At the same time, the computer also needs to specify a fixed IP.



Device IP in Vircom **IP configuration on computer**

Figure 29 configured in the same network segment

Whether it is directly connected or through a switch, when configured as a static IP, the device and the computer need to be in the same network segment (unless it is cross-gateway communication), as shown in Figure 29 above.

Since Vircom supports searching and configuration across network segments, the IP address that can be searched but cannot be communicated is generally not configured correctly. In this case, you can use Vircom to configure the device in the same network segment.

After configuration, please use 4.5 TCP communication test or 4.6 virtual serial port test, you can see that the Link light turns blue when the TCP connection is established. The blue link light can also be seen through Vircom. For example, in the device management list, if the TCP connection column is "established", it means the link light is blue, as shown in Figure 23. This can facilitate remote diagnosis.

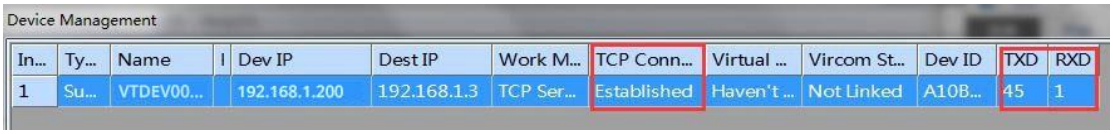


Figure 30 Connection status and data send/receive status

6.3 Data Sending and Receiving

When the Link light turns blue, data can be sent and received between the software and the serial server. By now, if the software sends a data, the Active light will turn green, and the duration will generally be at least 1 second. The data will also be output from the serial port of the serial server, but whether the output data is correct depends on whether the serial port parameters (including the baud rate, data bit, stop bit and parity bit) are configured.

For the serial device that has send the correct demand. It will respond. Once there is a reply (the serial port sends data to the network port), the Active indicator will turn blue.

Otherwise, please check the serial port parameters or whether the serial cable is connected.

In order to facilitate remote debugging, Vircom also supports remote viewing of data sent and received. As shown in Figure 30, TXD is the amount of data sent by the serial port server serial port. When the device list is refreshed, if this value changes, it indicates that there is data sent. The Active light will also be green; if you see the value of RXD changing, it indicates that the serial device has returned data, and Active is blue.

6.4 Vircom Remotely Monitor Data

When using the virtual serial port, Vircom supports real-time capture of the data of the virtual serial port. It is convenient for users to debug the system, and the method is as follows:

It is assumed that the communication of the virtual serial port has been established according to the method of 4.6 virtual serial port test. Now you need to monitor the data through the virtual serial port. Open Vircom's menu/Config/software setting/open the Vircom configuration dialog box.

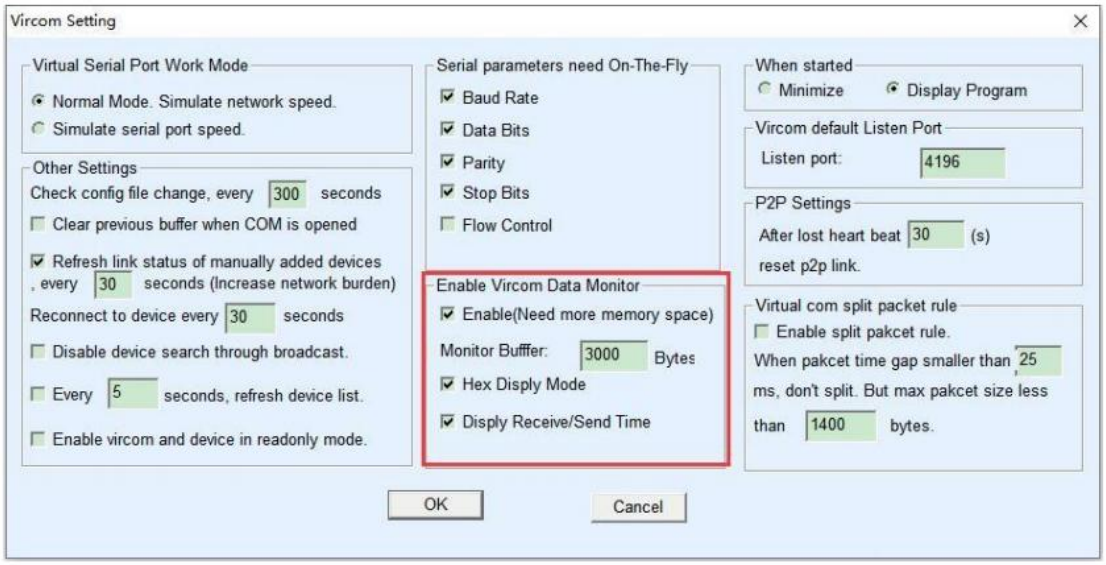


Figure 31 enabling Vircom monitoring

Check in front of the three options: open monitoring, hexadecimal monitoring mode, and display data sending and receiving time, as shown in Figure 30. Then click OK. Assuming that data has been sent and received before, now select a virtual serial port to be monitored in the main interface, and then select Menu/View/Monitor, as shown in Figure 31.

Figure 32 Open Vircom monitoring

From the opened dialog box, you can see the instructions sent by the host computer and the instructions returned by the device, as shown in Figure 33. This function can facilitate on-site communication debugging.

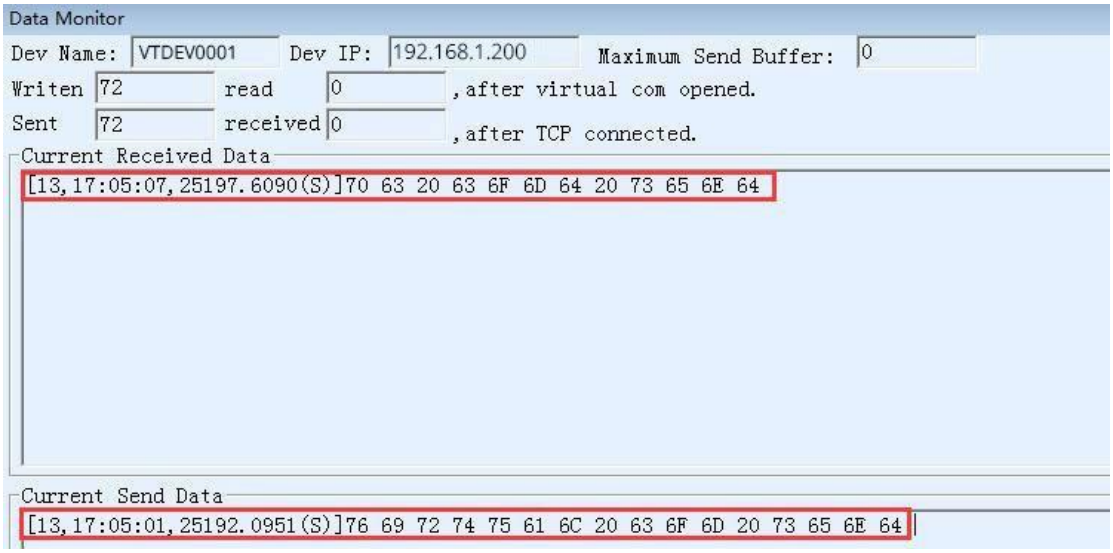


Figure 33 monitor sending & receiving data

7.Modbus Advanced Features

The serial port server with Modbus gateway function does not have station address and register itself. It is a communication bridge. It will generate Modbus RTU command according to the Salve ID, function code, register number and register quantity in the Modbus TCP command sent by the user software to the Modbus gateway, and output it from the serial port. It can be regarded as a protocol "translator".

7.1 Enable Modbus Gateway

First of all, the serial port server should support Modbus gateway, that is, the "Modbus TCP to RTU" function in Table 6 Device Supported Functions in the Device Settings dialog box should be checked.

By default, the serial port server is in normal transparent transmission mode. If you

need to convert it to Modbus gateway mode, please select the "Modbus TCP protocol" option in "Conversion protocol". After that, the device automatically changes the "Port" parameter to 502 (the port of the Modbus server). In this way, the Modbus gateway is enabled.

If the serial port RTU device is used as a slave, the host computer Modbus TCP software connects to the Modbus gateway's port 502. At this time, the Modbus gateway needs to work in TCP server mode; if the serial port RTU is used as a master, the Modbus gateway works in TCP client mode, and the destination IP is filled with the IP of the computer where the Modbus TCP software is located, and the destination port is generally 502.

7.2 Storage Modbus Gateway

The contents of the read registers can be saved inside the gateway, so that the speed of Modbus TCP query can be greatly improved, and the performance is better when supporting multi-host access.

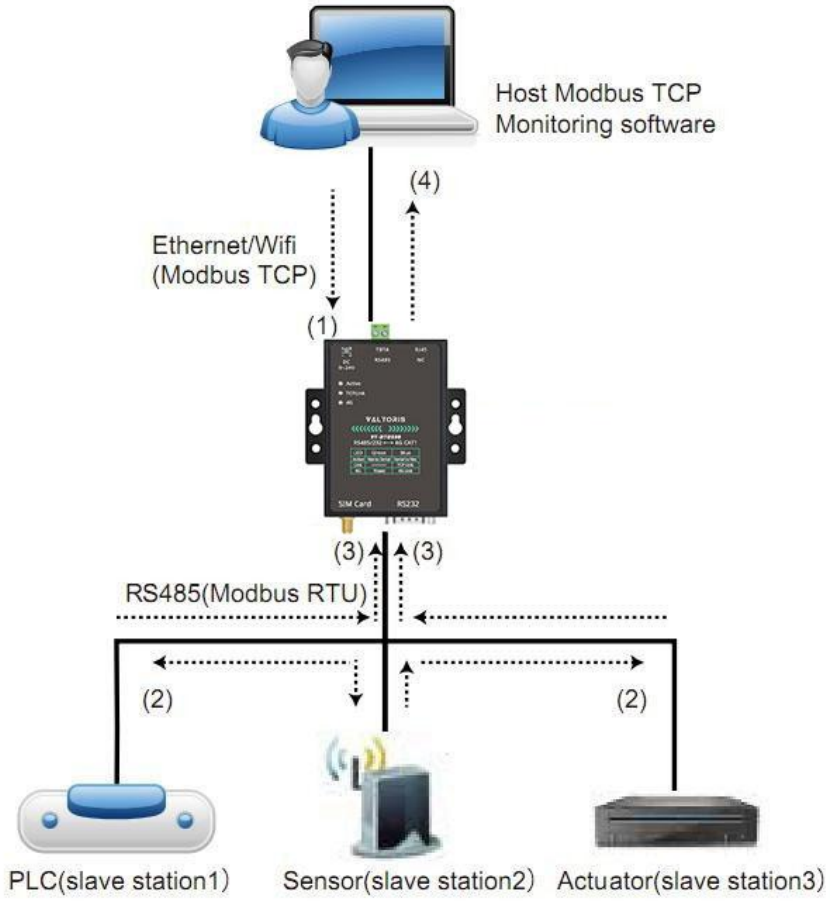


Figure 34 Storage Modbus Gateway

As shown in Figure 34: The direction of ordinary Modbus TCP data flow is （1）→（2）→（3）→（ 4 ）. That is, the Modbus TCP command is first converted into the corresponding command of Modbus RTU, and then the device responds to the Modbus RTU command to the Modbus gateway, and then the Modbus gateway converts it into Modbus TCP again and sends it to the monitoring host computer.

Modbus TCP is a network communication with a very fast transmission speed, and it can generally respond within 3ms, while Modbus RTU is RS485, which generally has a speed of only 9600bps, and it generally takes at least 30ms to send and return a command. In this way, the query response time of ordinary non-storage Modbus gateways is relatively long. In addition, if there are many host computers querying data at the same time, the serial port will be congested. If the network is compared to a highway and the serial port is compared to a single-plank bridge, then the original method is to pass the traffic of the highway on the single-plank bridge.

The register-saving Modbus gateway solves the above problems. It can temporarily save the register data obtained in the Modbus gateway, so that when the Modbus TCP query, the Modbus gateway can immediately return the command, which demonstrates Modbus TCP can be operated quickly. On the other hand, the serial port server can take the initiative to send instructions from the serial port to automatically update the contents of the currently saved register data, and save a copy of the latest register value.

In addition, this serial server is a fully automatic configuration-free Modbus gateway, users do not need to configure the required register address, function code, slave address, etc. The serial server will automatically recognize and dynamically add these registers according to the Modbus TCP commands sent from the network port.

2/4/8 CH-RS485-ETH can response quickly when monitored by multiple computers. No matter what baud rate of the serial port is, it can generally response the data to the host within 3ms, which showed a good updating speed of serial port data.

The register-saving Modbus gateway is a real Modbus TCP to Modbus RTU. It really takes advantage of the fast speed of Modbus TCP and the ability to query by multiple hosts at the same time.

Note that when the serial port server is used as a TCP client, it does not have the storage function and will automatically switch to the non-storage type.

The characteristics of storage Modbus are listed below:

1) The first Modbus TCP query command is non-storage type. Because you must wait for the RTU device to return data slowly before returning the register content to the network port.

2) If a specific command is no longer inquired by the host computer on the network side

within 5 seconds, this command will be automatically deleted and no longer sent to the RTU device from the serial port.

3) At present, it can store 10K Modbus cache. For ordinary single-register query, about 500 instructions can be stored at the same time.

4) When more commands are being queried at the same time, they are sent in order. The first command sends → the first command response → waits for 485 anti-collision time (refer to the multi-host section) → The second command sends... Return to the first command after the last command has been answered.

7.3 Disable Storage Functionality

Although the storage type Modbus has a faster response speed, some users do not want the serial port device to receive a large number of query commands, which affects the internal processing speed of the instrument. In this case, the storage type function can be turned off.

To disable the storage type, click the "More Advanced Options" button in the "Parameter Configuration" dialog box and select the non-storage type Modbus gateway. Then return to the device settings and click Modify Settings.

Note that when using the Web method to configure the conversion protocol, the default is a non-storage Modbus gateway.

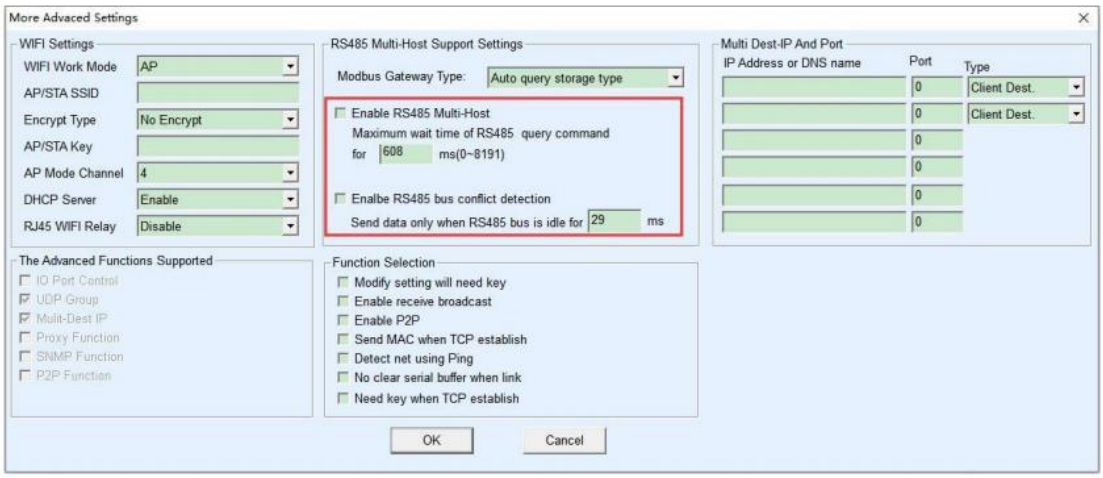


Figure 35 disable the storage function

7.4 Multi-host Function

As shown in Figure 35, "RS485 multi-host support" and "RS485 bus conflict detection function" are the multi-host functions. They are generally enabled and disabled

at the same time. After enabling, the device with the conversion protocol of Modbus TCP has the function of storage-type Modbus gateway, otherwise it is a non-storage-type Modbus gateway; if the conversion protocol is None, the user-defined RS485 protocol can generally also have the function of serial port devices accessed by multiple hosts at the same time, which is impossible in a pure RS485 network, because multiple masters sending at the same time will cause conflicts on the RS485 bus. The multi-host of this serial port server can "coordinate" the RS485 bus to achieve the purpose of multi-host access.

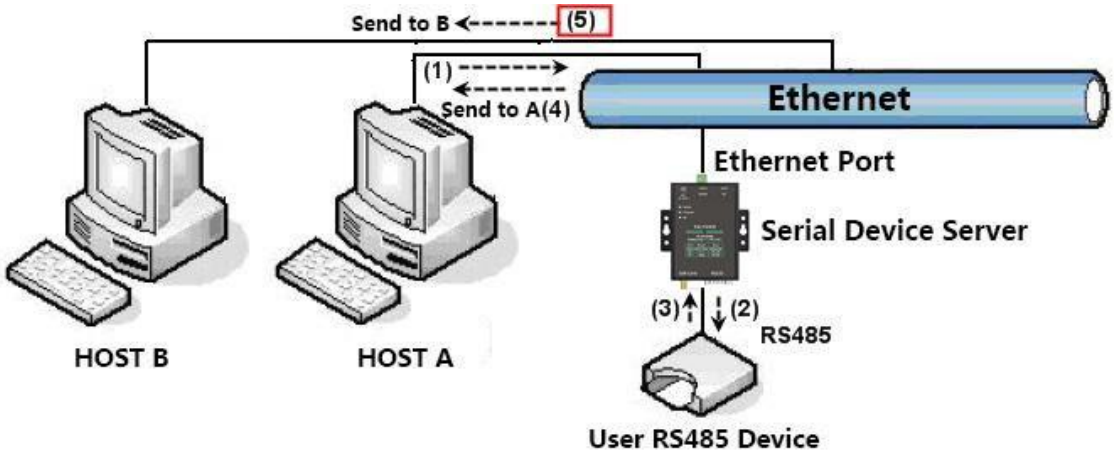


Figure 36 Multi-host function demonstration

As shown in Figure 36, in the normal mode, when two hosts: host A and host B are connected to the serial server at the same time, host A sends (1) command, RS485 device receives (2) command, RS485 device returns (3)) command, but the serial port server will send (4) to host A and (5) to host B at the same time. Since the host B did not send the query, but it also received the response command (5), the host B may generate a communication abnormal error. In the multi-host mode, there will only be command (4) but not command (5), because the serial server will automatically remember the host that needs to be returned, and only return the instructions to the most recent communication host, and the query of host A will only reply to A, host B queries and replies to host B.

Another function is that in the normal mode, when the host A and the host B send data at the same time, the combination of instructions will be generated on the RS485 bus, so that it cannot be recognized normally; in the multi-host mode, the serial server can schedule A and B to use the bus. Prioritize the conflict problem of simultaneous access by multiple machines in an effective way.

When the conversion protocol is "None", the multi-host function is not enabled by default.When you need to enable multiple hosts, please click "More Advanced Settings" in the device configuration dialog box, and then check "RS485 multi-host support".

7.5 Multi-host Parameter

The meanings of "RS458 multi-host support" and "RS485 bus collision detection function" are described as follows.

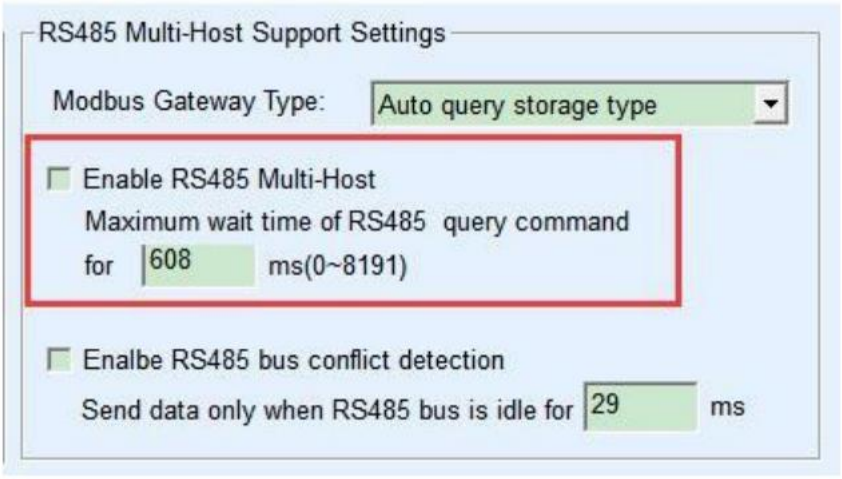


Figure 37 RS485 multi-host support

The RS485 command response timeout is the maximum time interval from when the serial server starts sending this command to when it receives a response. The time filled in should be greater than the actual maximum time interval because if it is judged to be a timeout, the next command will be sent.

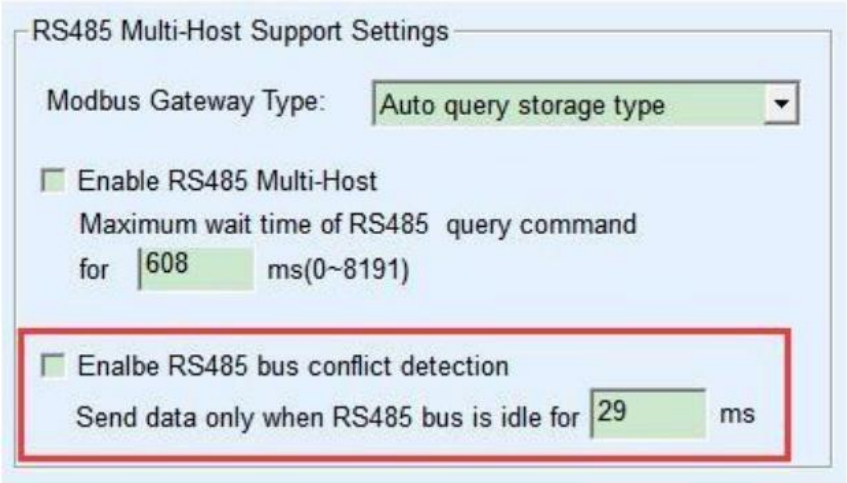


Figure 38 RS485 anti-collision idle time

RS485 bus conflict time: indicates how many milliseconds the serial server waits after receiving the reply of the first command before sending the second command. This parameter actually defines the speed of the command rotation. The value is recommended to be more than 20ms. The parameter "maximum waiting time 3 seconds" generally does not need to be modified.

When the user uses Vircom to select the conversion protocol as "Modbus_TCP Protocol", Vricom will automatically check the above two enabling boxes (unless the user manually enters the advanced option to remove it), and the above two times will also be automatically configured according to the baud rate . However, if the user's Modbus command is relatively long or the conversion protocol is "None", these two parameters need to be configured manually.

The following describes the recommended settings of the above parameters:

1. Figure 38 shows the "RS485 bus anti-collision time". Generally, it can be set to twice the "Packet Interval" in the lower right corner of the parameter configuration interface, but the minimum should not be less than 20.
2. Figure 37 shows the "RS485 command response timeout", which is generally determined according to the length of the back and forth response command. If the sending command is N bytes and the response is M bytes, the recommended value is: "Packet interval" ×(N+M+5)+100.

7.6 Modbus Under Multi-destination IP

As shown in Figure 36, if the serial device (RTU device) is used as the master station and the network port device (Modbus TCP device) is used as the slave station, and there are multiple network port slave devices at the same time. At this time, you can use the method described in 5.2.2 Client Connecting Multiple Servers to let the serial port server as a client connect to these multiple network port devices at the same time.

The function that needs to be realized at this time is: when the serial port RTU sends a command, it can be sent to multiple network port devices, the network port device recognizes whether to send it to itself through the Slave ID field, and only the network port device corresponding to the Slave ID responds. After the network port response is sent to the serial server, it is converted into an RTU command and output from the serial port to the RTU device.

At this time, it should be noted that it is necessary to remove the two check marks shown in Figure 38 as "RS485 bus anti-collision time" and Figure 37 as "RS485 command response timeout time". Otherwise, the above forwarding function cannot be realized. Another application method is: Although the serial server is connected to multiple network port devices as a Client, the RTU device is not the master station, and the network port device still sends it first, and the RTU device responds (as a slave station). At this time, the "RS485 bus anti-collision time" and "RS485 command response timeout time" still need to be checked, so that multiple hosts can access one RTU device at the same time.

8. Registration Pack and Heartbeat Pack

Registration packets and heartbeat packets are a function suitable for communication between devices and cloud software.

8.1 Registration Packet

The definition of a registration package is that when the computer software and the serial port server module (hereinafter referred to as the module) establish a TCP connection, the module will first send a string of codes to the software so that the software can know which module is communicating with itself. This string of codes is the registration package.

The registration package is very suitable for the monitoring of the Internet of Things, because the cloud software generally runs on the public network server of the Internet, and the modules are scattered in various collection and monitoring points. How to make the cloud software recognize the module is very important and is necessary to realize the communication of the Internet of Things.

Our serial port server provides the following multiple registration report methods.

8.1.1.Send MAC Address On Connection

Its method is to send its mac address to the cloud when the module is connected to the cloud. Since the MAC address is unique, it can identify the corresponding device. This method is simple and does not require writing a registration package for each device, so it is simple and effective. How to use it: Click "More Advanced Settings" in the device settings dialog box, find "Send MAC address when TCP is establish" in the upper middle, tick the front, and then return to the setting interface and click "Modify Settings".

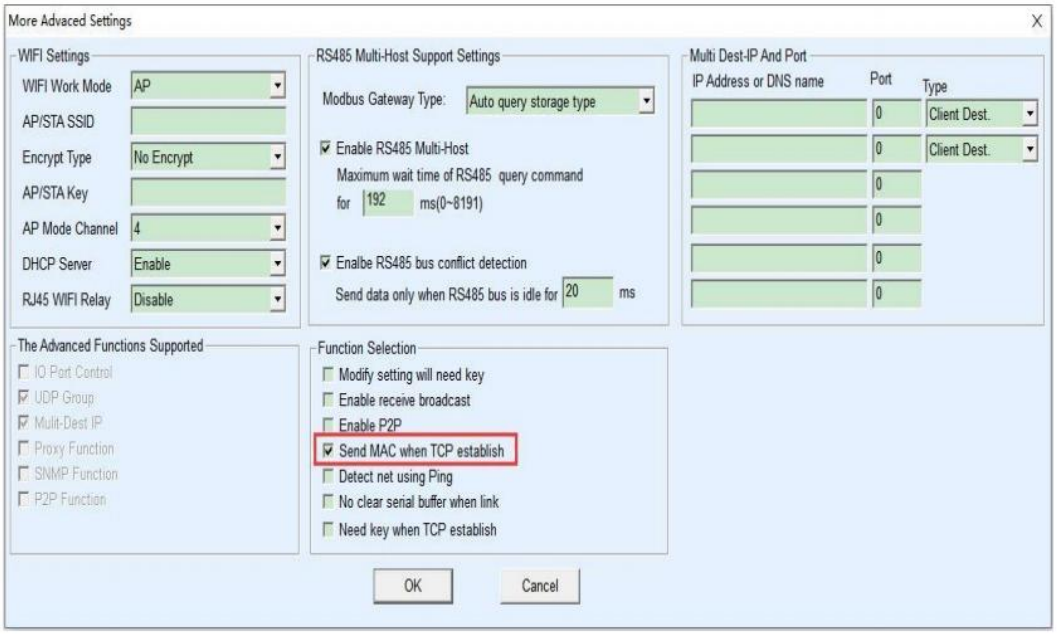


Figure 39 Sending MAC address on connection

8.1.2 Realcom Agreement

The Realcom protocol is a protocol containing a registration package and a heartbeat package. Users can use this protocol to realize the functions of the registration package and heartbeat package. The method to enable the Realcom protocol is: in the "Device Settings" dialog box, select "Conversion Protocol" to "REAL_COM Protocol", and note that the part of enabling the registration package needs to be blank and unchecked.

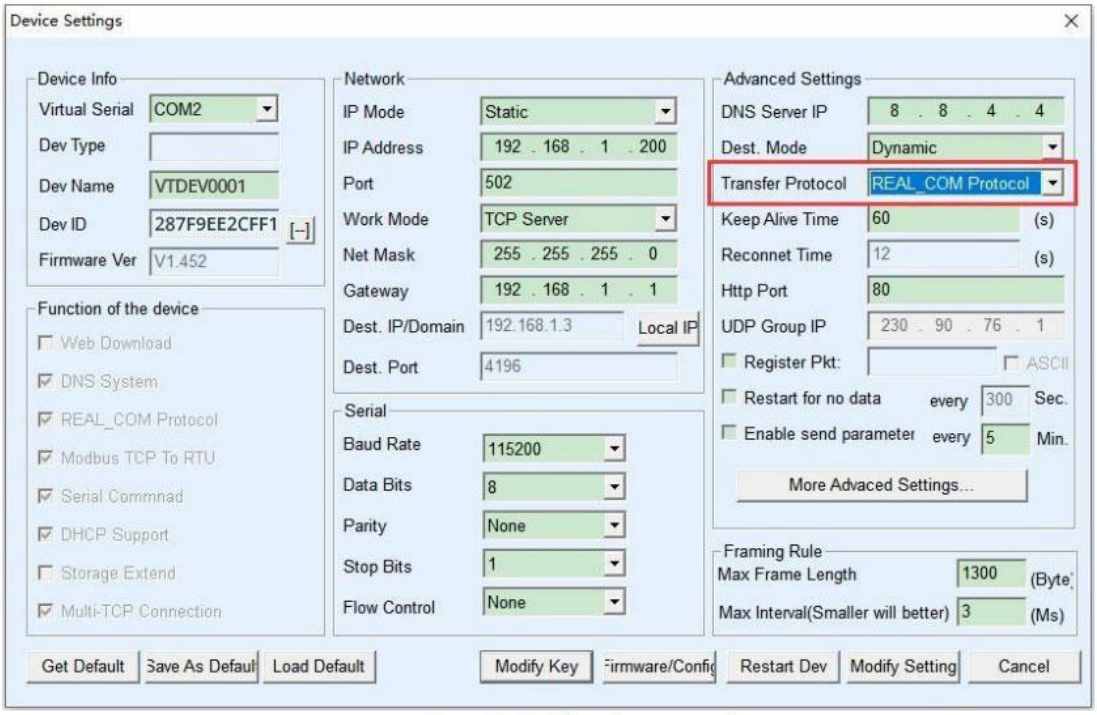


Figure 40 Enable the realcom protocol

After the Realcom protocol is enabled, it will not be transparent transmission communication. It has the following characteristics:

1. After the device establishes a TCP connection with the cloud, the device automatically sends a hexadecimal registration packet FA 07 13 02 FA 02 MAC[5] MAC[4] MAC[3] MAC[2] MAC[1] MAC[0] FA FF. Among them, MAC[5]~MAC[0] is the MAC address of the device.

2. When the device sends data to the network, it will automatically add the 3-byte header prefix of FA 01 01.

3. Every keep-alive timing interval, the device sends a 00 1-byte heartbeat packet to the software. The REAL_COM protocol can be used as a device registration package because it contains the MAC address in the registration package. However, due to its fixed format, only the cloud software can design the REALCOM protocol to be compatible with this method.

8.1.3 Custom registration package

The custom registration package method allows users to fill in any registration package format. The method is: in the device settings interface, configure as follows:

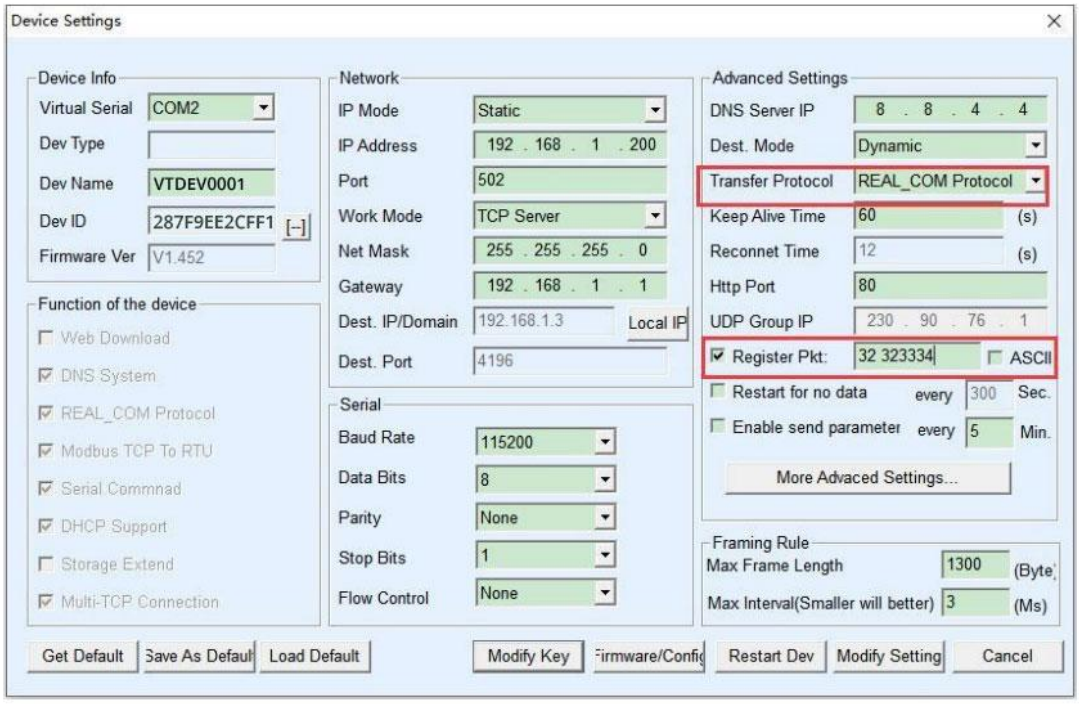


Figure 41 Set registration package

The difference from the REAL_COM protocol is that the registration package is enabled here, and the registration package information such as 31 32 33 34 is filled in. Note that this is hexadecimal, which means that the data actually sent is a string of characters 1234. If you need to display the string, click the "ASCII" option next to it.

When the device is connected to the cloud software, it can automatically send a hexadecimal registration package of 31 32 33 34. This registration package method is more flexible and allows the device to adapt to the existing cloud registration package format; however, there is no wildcard such as MAC in the registration package, and different registration packages need to be configured for each device separately, which is more cumbersome. The above two methods of sending MAC address and REALCOM have the same configuration for each device, but the registration package is naturally different due to different MAC.

The longest registration package length is 33 bytes. This method supports registration packages and heartbeat packets in UDP mode.

8.1.4 Configuration File

The serial port server is supported to write a configuration file, so as to realize a completely customized registration package by the user, and the MAC address wildcard

can be used, which can solve the trouble of writing a custom registration package for each device, and there is no limit on the length of the registration package.

8.2 Heartbeat Packets

Heartbeat packets are mainly used to detect whether the communication link is disconnected. The implementation method is that the device sends a heartbeat packet data to the server software at regular intervals. This data will be discarded after being received by the server and will not be regarded as valid communication data.

The heartbeat packet has two main functions: first, it can let the host computer software know that the device is active; second, if the device fails to send a heartbeat, the device on the TCP client will automatically re-establish the TCP connection, so it is a means to restore network communication.

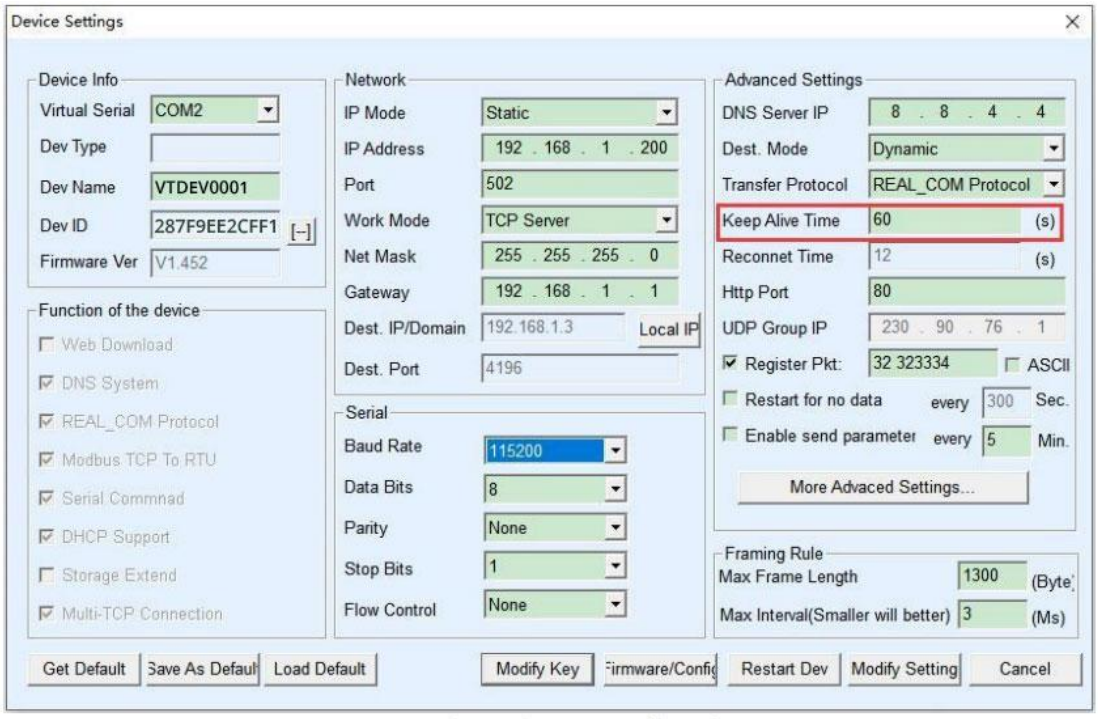


Figure 42 Keep Alive time

As shown in Figure 42, the sending time of the heartbeat packet is set by the "Keep Alive Time".

8.2.1 Implied Heartbeat

Even if you don't set any heartbeat packets, the implicit heartbeat function will be enabled when the Valtoris device is on the TCP client.

Therefore, the implicit heartbeat function means that the device sends data, but the server cannot actually receive the heartbeat data. Therefore, it cannot play the first function of the heartbeat packet, that is, the server detects whether the device is active or

not; but because the device actually sends data, it can play the second function of the heartbeat packet, that is, the device detects the TCP connection is normal, once the disconnection is detected, the TCP connection can still be automatically re-established.

8.2.2 REALCOM Protocol

As described in 8.1.2 Realcom protocol, REALCOM protocol can send a 1-byte data of 00 every keep-alive time. This data is the heartbeat packet of realcom protocol.

8.2.3 Customize Heartbeat Packets

First, fill in the registration package according to the method of 8.1.3 customizing the registration package. Then add the heartbeat packet as follows: Click the "More Advanced Settings" button in the device settings, write the hexadecimal heartbeat packet in the second line of the multi-destination IP and port, and change the option on the right to "Param. Dest".

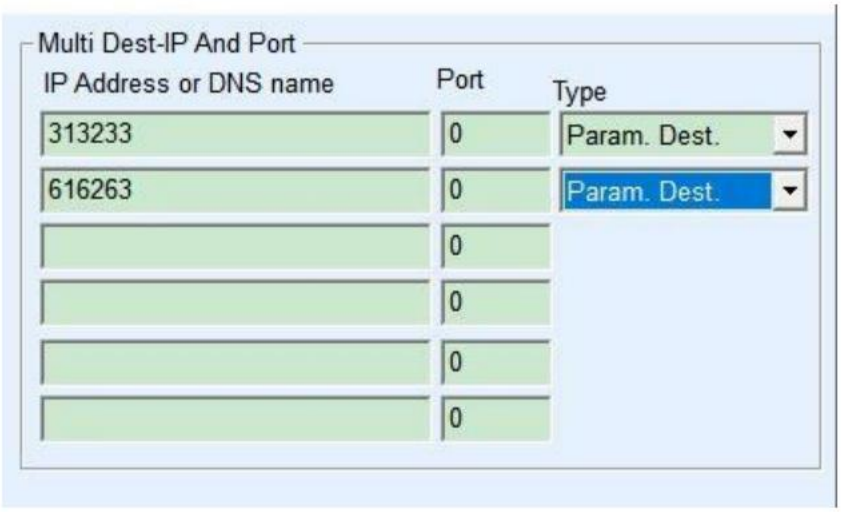


Figure 43 Customized registration package

Note that the sum of the registration packet and the heartbeat packet should be less than 33 bytes. The first line is actually the registration package.

9. Remote Device Management

The so-called remote device management refers to the ability to maintain and manage the device through Vircom software, including restarting the device, modifying parameters, and upgrading firmware. This function is suitable for users who use Vircom for device management.

For Vircom software, as long as the device can be found in the device list, it can be managed remotely. The remote management of equipment is divided into the following situations:

1. Automatic search: The device and the computer are on the same switch. At this time, no matter whether they are in the same network segment or not, the Vircom search method on the computer is: Vircom sends a broadcast query. After receiving the query, all devices will reply with their own parameters. Vircom tools. This method searches all devices at once.

2. Manually add: There are two situations:

a) Large-scale routers divide the network: In some large-scale networks, broadcast packets are divided by routers, so that the broadcast packets cannot reach the device end, but the ping device IP is all connected. At this time, it generally needs to be added manually to solve the problem. The method of adding manually is to click "Add manually" in the "Device Management" dialog box to add the first and last IP to query the devices one by one.

b) The public network server queries the internal network equipment: the serial port server is in the internal network and acts as a TCP server mode, and Vircom is on the public network IP server. At this time, you need to do a 1092 UDP port mapping on the router of the network where the device is located, and map it to the IP where the device is located, and then Vircom manually adds the device. The IP is the public network IP of the device.

3. TCP client: When the device acts as a TCP client, it will initiate a TCP connection to port 4196 of the destination IP (116.15.2.3). After the connection is established, it will automatically send its own parameter system to the UDP port (note that it is not the TCP port) of the destination port (here 4196) at each keep alive timing, so that Vircom can be able to do so on this computer (116.15.2.3) The device is found. If the destination port is not 4196, you need to modify the Vircom's default parameter receiving port. The method is to modify the menu/configuration/software configuration/default listening port, and then start Vircom. If a TCP port conflict pops up, just ignore it and continue execution.

4. Sending parameters at regular intervals: Even in a serial server in TCP server mode, you can check the "Send parameters at regular intervals" function, and the parameters will be sent to the destination port of the destination IP (here 116.15.2.3) every 5 minutes. Vircom that receives parameters on this port of this server can manage these devices.

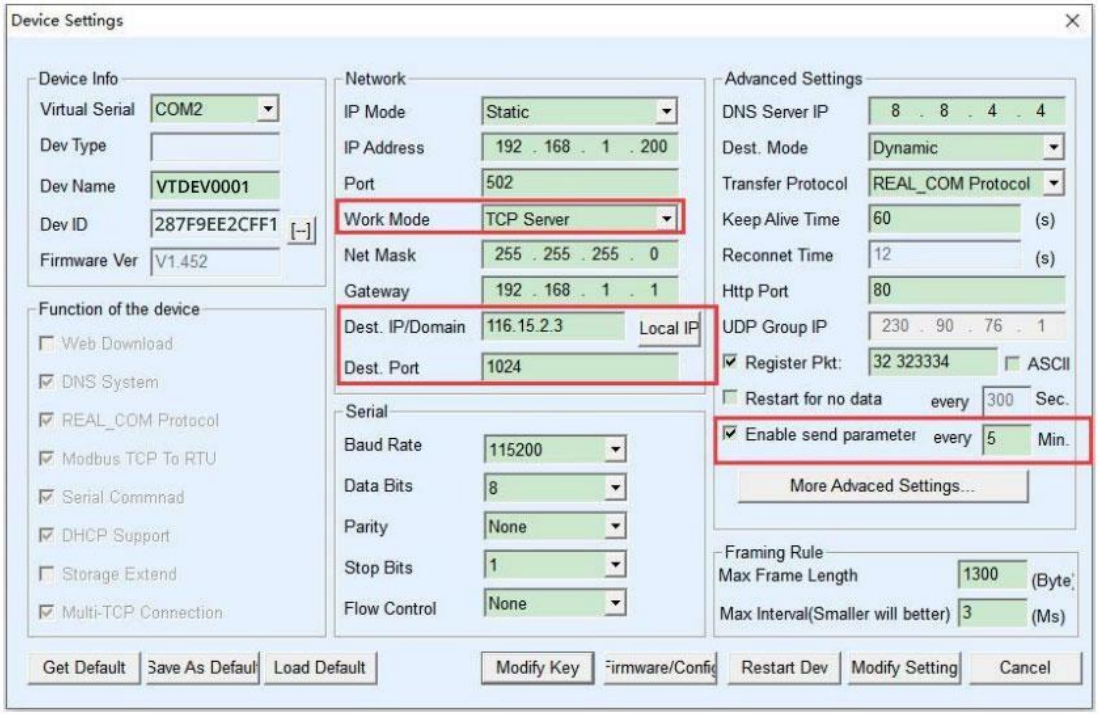


Figure 44 Timing sending parameters

In order to facilitate the identification of the device, if you need to perform remote management, please give the device a name that is easy to remember.

10. MQTT and JSON

For related usage, please refer to：《 MQTT and JSON User Manual 》

11. Firmware upgrade method

2/4/8 CH-RS485-ETH can upgrade their own programs, but not mutually upgrade programs. Whether it is by automatic search, manual addition, can use this method to upgrade the firmware.

1. Obtain the firmware file of 2/4/8 CH-RS485-ETH(V)(VI) from Valtoris.

2. Search for the device that needs to be upgraded in Vircom tool at first, and then enter the device parameter editing dialog box. Click on "Restart Dev".

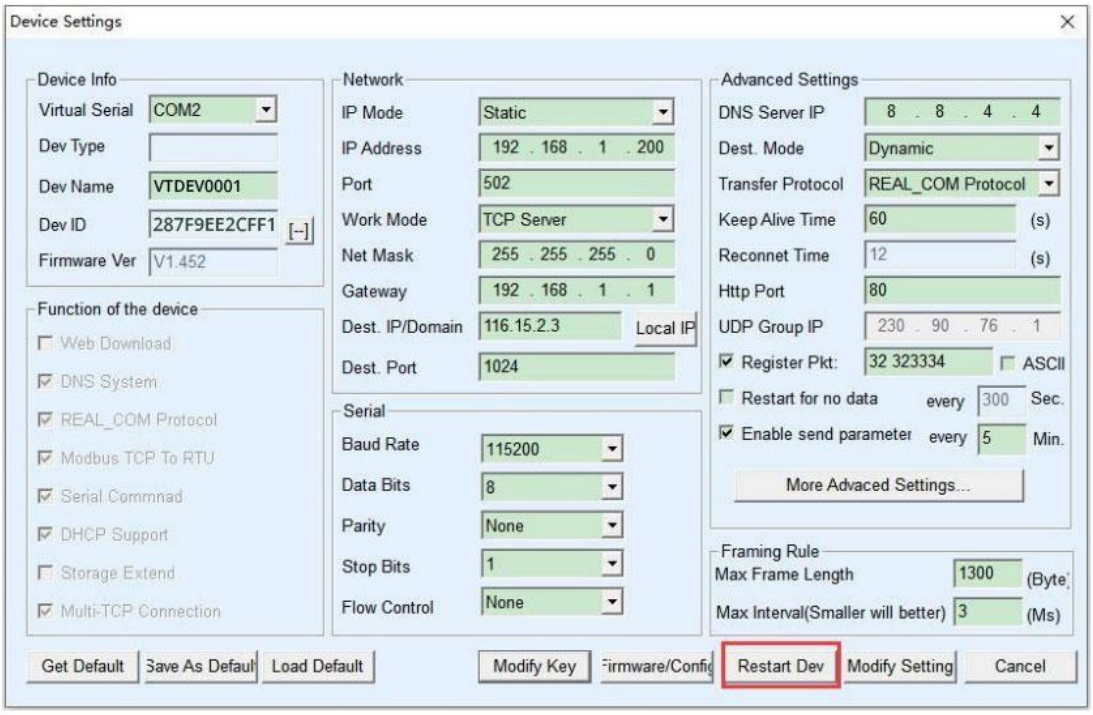


Figure 45 Restart the device

After the device restarts, use the same method to search for the device again, and enter this dialog box again. Click the "Firmware/Config" button in the lower right corner of the dialog box.

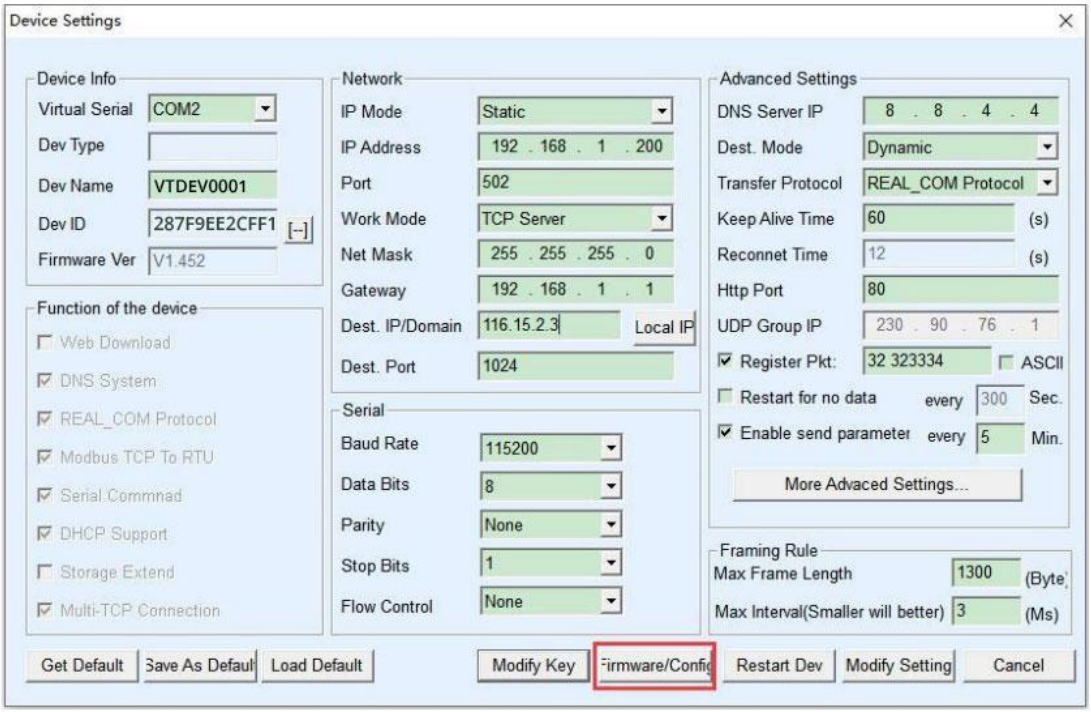


Figure 46 Upgrade firmware

3. As shown in Figure 47, select the "Program File Download" option. In the program file, select the firmware file. The IP address part of the serial server has been filled in automatically, and there is no need to write it again. The module type/model has been automatically selected. Then click download.

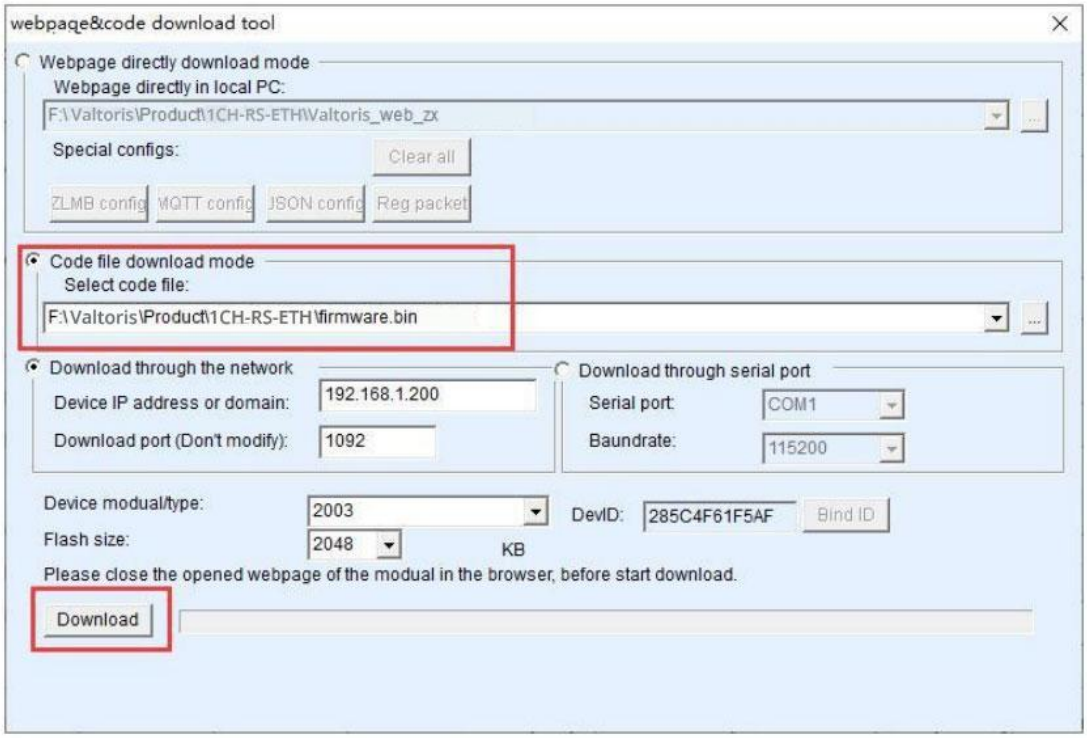


Figure 47 2/4/8 CH-RS485-ETH firmware upgrade method

4. By now, the download progress bar starts to move, and the download time is about 30 seconds. During the download, you will see the ACT light of the device flashing, and at the end of the download, you will see the LINK light flashing a few times. Then the program pops up a prompt box "Do not power off the device when the LINK light flashes after the transmission is complete". Note: This is only the completion of the transfer. The flash writing process takes about 3 seconds. At this time, the LINK light will flash. Please do not cut off the power during this period.

5. After the download is complete, the general program will restart automatically, and there is generally no need to cut off the power. If you see the running indicator flashing, if it does not restart automatically, please stop flashing the LINK indicator for more than 30 seconds, and then power on again.

6.Web configuration interface update: After firmware upgrade, the internal configuration web page of the module also needs to be updated, otherwise it will no longer be able to configure via Web, but it will not affect communication. It is not necessary to download the web page if there is no need for web configuration. The way to download the Web is: as shown in Figure 48, change the "program file" download

mode to "web directory download". And select the root directory where the local web page is located as the directory where the web page file to be downloaded is located (the directory from Valtoris), click download, and download all the files in the local web page directory to the internal file system of the device.

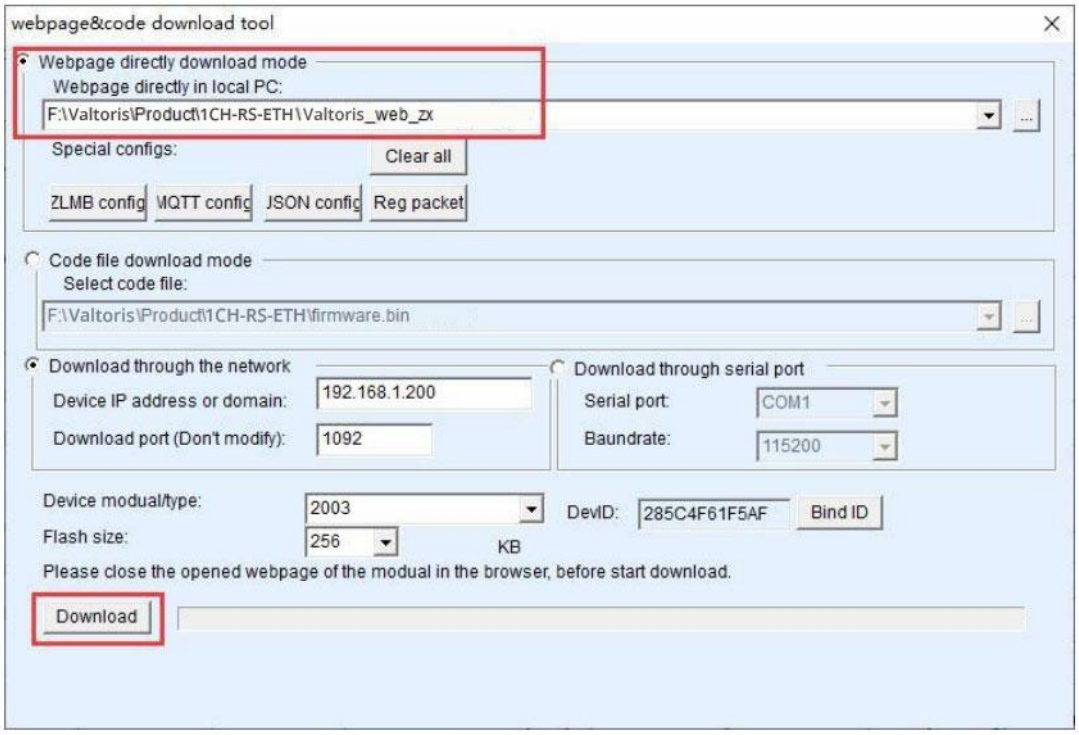


Figure 48 2/4/8 CH-RS485-ETH web upgrade method

7. Notice:

7.1 If it prompts that the download failed and the device will not be damaged, please restart the download. In addition, at the end of the download, when the LINK light is flashing, do not power off, otherwise the device will be damaged.

7.2 Check the firmware version number by Vircom, and you can know whether the new firmware has been downloaded successfully.

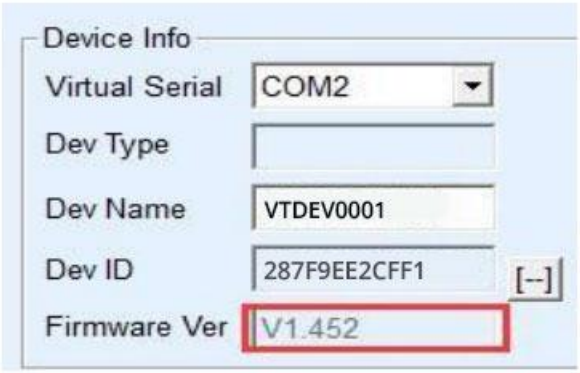


Figure 49 Check the firmware version after the level is completed

12. After-sales service and Technical Support

Phone/WhatsApp: +86 18321985506

Web: www.valtoris.com

Email: support@valtoris.com