RS485 TO ETH (B)
User Manual

Figure 1 RS485 TO ETH (B)

1. OVERVIEW

This is a RS485 device data collector/IoT gateway specially designed for industrial environments. It combines multiple functions such as serial server, Modbus gateway, MQTT gateway, RS485 to JSON and so on. It has an RS485 interface and an Ethernet interface. It adopts a guide rail type installation method, compact and easy to install. Adopt terminal type power connection, 9~24V wide voltage input. Because there is only RS485 interface, the RS232 interface that is not commonly used on industrial instruments is eliminated, which saves cost and volume, and has a higher cost performance compared with ordinary products.

When used as a common serial port server, connect the RS485 TO ETH (B) network port to the Ethernet, and the host computer software can use TCP/UDP transparent transmission or virtual serial port mode to send data to RS485 TO ETH (B) On the RS485 interface; the data received on the RS485 interface will also be transmitted to the software or virtual serial port on the TCP side. In the virtual serial port mode, the serial port software connected to the virtual serial port can be used without any need.

When used as a Modbus gateway, RS485 TO ETH (B) supports Modbus TCP to Modbus RTU, so that the host computer uses Modbus TCP protocol to collect data from RS485 Modbus RTU devices. On the contrary, if the RS485 terminal is used as the master station, it can also be used. The supported multi-master function allows multiple computer masters to access an RS485 slave device at the same time.

When used as an MQTT gateway, the device can transparently upload serial port data to the MQTT server using the MQTT protocol. Supported servers include Baidu Cloud MQTT, Alibaba Cloud MQTT, China Mobile OneNet platform, etc. Support to parse the collected Modbus RTU or non-standard serial port data into JSON format, encapsulate it in MQTT data package and upload it. Support JSON format to upload collected data, automatic data collection, collected data support Modbus RTU, 645 meter 97 version, 645 meter 07 version, various non-standard RS485 protocols. Users can configure the uploaded data format and JSON keywords with Vircom. When uploading, it can support MQTT protocol, HTTP POST protocol, HTTP GET protocol, transparent transmission protocol, and various non-standard network protocols. RS485 TO ETH (B) has various functions of a powerful IoT gateway, which is very suitable for collecting various RS485 instruments and sensors in the industrial field, including local network collection or uploading to the cloud server.

RS485 TO ETH (B) can be applied to :

- As a gateway of the Internet of Things as a communication bridge between the device and the cloud;

- Electricity, smart meters and energy consumption monitoring;
- Remote monitoring and program downloading of various automated PLCs;
- Various configuration software and equipment communication interfaces;
- Equipment networking in the field of access control and security;

A typical application connection is shown in Figure 2. The original serial device RS485 is connected to the RS485 port of RS485 TO ETH (B), and RS485 TO ETH (B) is connected to the computer through a network cable. The software on the computer establishes a connection with RS485 TO ETH (B) through TCP/IP or virtual serial port. After that, any data sent by the serial device will be transparently transmitted to the computer software, and the data sent by the software via the network RS485 TO ETH (B) will also be transparently transmitted to the serial device

Figure 2 Connection diagram

2. FEATURES

2.1.HARDWARE FEATURES

- Guide rail design: suitable for installation inside cabinets in the industrial field
- Small size: Compared with ordinary serial server, the width is smaller and does not take up space
- Terminal type power supply, 9~24V wide voltage input, with reverse power protection
  Terminal RS485 interface, support 32 slave devices, baud rate support 300~115200bps
- Abundant panel indicators are convenient for debugging: In terms of connection, there are not only LINK lights that indicate whether the network cable is connected, but also LINK lights that indicate the establishment of a TCP connection; the data indicator includes "serial port to network port" and "network port to Separate indicator light for serial port

2.2.SOFTWARE FEATURES

- Support TCP server, TCP client, UDP mode, UDP multicast. As a TCP client, it also supports TCP server functions. As a TCP server, it supports 30 TCP connections, and as a TCP client, it supports 7 destination IPs.
- The baud rate supports 1200¯¯115200bps, the data bit supports 5-9 digits, and the parity bit can be five modes: no parity, odd parity, even parity, mark, and space.
- It supports the function of sending MAC address on the device connection, which is convenient for cloud management of the device.
- Provide the secondary development kit DLL development library for searching and configuring equipment on the computer side.
- Support Web browser configuration, DHCP dynamic acquisition of IP, DNS protocol to connect to the domain name server address.
- Support cloud remote search for devices, configuration of device parameters, and device program upgrades.
- Support remote viewing of the device's TCP connection status, serial port data sending and receiving status through software. Virtual serial port supports data monitoring function.

2.3.ADVANCED SOFTWARE FEATURES

- Support Modbus gateway function, support Modbus RTU to Modbus TCP. It can support storage Modbus, which can automatically collect device data and store it; it also supports non-storage mode Modbus gateways
- Support multi-host function: In the query mode of one question and one answer, the support network port allows multiple computers to access the same serial device at the same time.
- Support MQTT gateway function.
- Support JSON to Modbus RTU and 645 instrument protocol, support HTTP POST, HTTP GET format to upload data.
- Support NTP protocol to obtain network time, which is used for serial port output and the latter is used for protocol content uploading.
- Support custom heartbeat package and registration package function: it can facilitate communication and device identification with the cloud.
- Support the function of password authentication required for TCP connection establishment to ensure connection security.

- Supports the http method of data submission and distribution. The cloud can directly use the http GET command to interact with the device's serial port data.

3. TECHNICAL PARAMETER

Table 1 Technical parameters

Appearance	
Communication interface	RS485, wiring terminal mode
Power supply	terminal mode
Dimensions	L x W x H =8.7cm×3.6cm×5.9cm
Communication interface	
Ethernet	RJ45 interface, 2 KV surge protection
Serial port	RS485×1： RXD， TXD， GND
Serial port parameters	
Baud rate	300~115200bps
Check Digit	None, Odd, Even, Mark, Space
Data bits	5~9 bits
Flow control	No flow control
software	
Protocol	ETHERNET、IP、TCP、UDP、HTTP、ARP、ICMP、DHCP、DNS
Configuration method	VirCOM tool, WEB browser, device management function library
Communication mode	TCP/IP direct communication, virtual serial port mode
Operating mode	
TCP server, TCP client (TCP server coexists at the same time), UDP, UDP multicast	
Power requirements	
Power	9~24V DC
Environmental requirements	
Operating temperature	-40~85℃
Storage temperature	-45~165℃
Humidity range	5~95% relative humidity

d

CONTENT

1. Overview....2
2. Features....3
2.1.Hardware features....3
2.2.Software features....3
2.3.Advanced software features....4
3. technical parameter....5
4. Instructions for use....8
4.1. Hardware description....8
4.2. Hardware connection....11
4.3. Software Installation....11
4.4. Parameter configuration....11
4.5. TCP communication test....17
4.6. Virtual serial port test....19
4.7. Modbus TCP test....21
4.8. Web configuration....23
5. Working mode and conversion agreement....24
5.1. Virtual serial port mode....25
5.2. Direct TCP/IP communication mode....26
5.2.1. TCP client mode....26
5.2.2. Client connects to multiple servers....27
5.2.3. TCP server mode....29
5.2.4. Acting as both a client and a server....29
5.2.5. UDP mode....30
5.3. Device couplet way....31
6. Equipment debugging....33
6.1. Network physical connection....33
6.2. Network TCP connection....33
6.3. Data sending and receiving....34
6.4. Vircom remote monitoring data....34
7. Modbus advanced functions....36
7.1. Enable Modbus gateway....36
7.2. Storage Modbus Gateway....37
7.3. Disable storage functions....39

7.4. Multi-host function....39
7.5. Multi-host parameters....40
7.6. Modbus under multi-destination IP....42
8. Registration package and heartbeat package....42
8.1. Registration package....43
8.1.1. Send MAC address on connection....43
8.1.2. Realcom Agreement....43
8.1.3. Custom registration package....44
8.1.4. Configuration file....45
8.2. Heartbeat package....46
8.2.1. Implied heartbeat....46
8.2.2. REALCOM protocol....47
8.2.3. Custom heartbeat package....47
9. Remote device management....47
10. MQTT and JSON....49
11. firmware upgrade method....49

4. INSTRUCTIONS FOR USE

The RS485 TO ETH (B) module is used to realize the two-way transparent transmission of data from the serial port to the Ethernet port. The user does not need to care about the specific details, and the module completes the protocol conversion inside. The serial port side is the serial port level data, and the Ethernet port side is the network data packet. You can specify the work details through simple settings. The parameters can be set through the built-in webpage or through the setting software, and the settings can be saved forever.

This chapter is the user manual for the product RS485 TO ETH (B). It is recommended that users read this chapter systematically and follow the instructions to get a systemic understanding of this product. Users can also choose the chapters they are interested in to read according to their needs. For specific details and instructions, please refer to the subsequent chapters.

In addition to this manual, we also provide configuration software and other materials based on this product for user design reference download link:
http://www.waveshare.com/wiki/RS485_TO_ETH_(B)_

4.1. HARDWARE DESCRIPTION

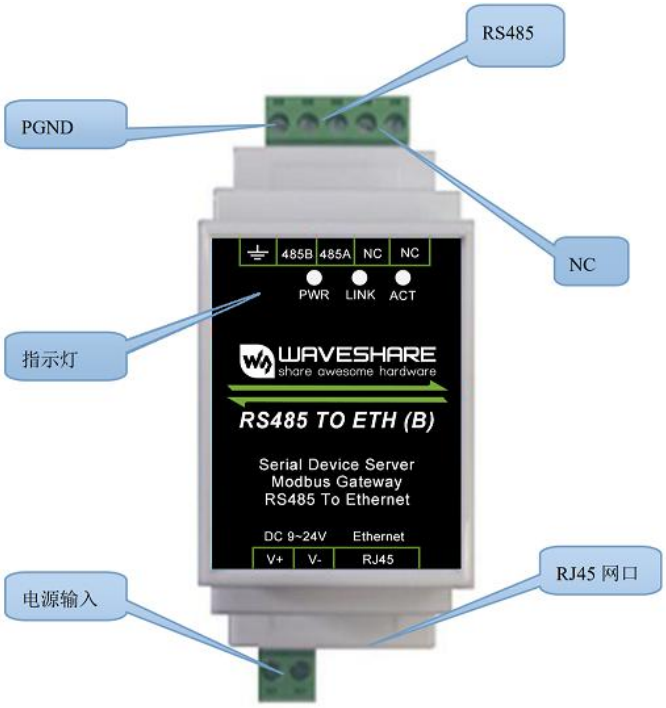


Figure 3 Front view

The front view of the RS485 TO ETH (B) serial server is shown in the figure above, using a 35mm standard rail enclosure.

1 Power input: Wiring terminal: 5.08mm terminal. The input voltage is 9~24V.

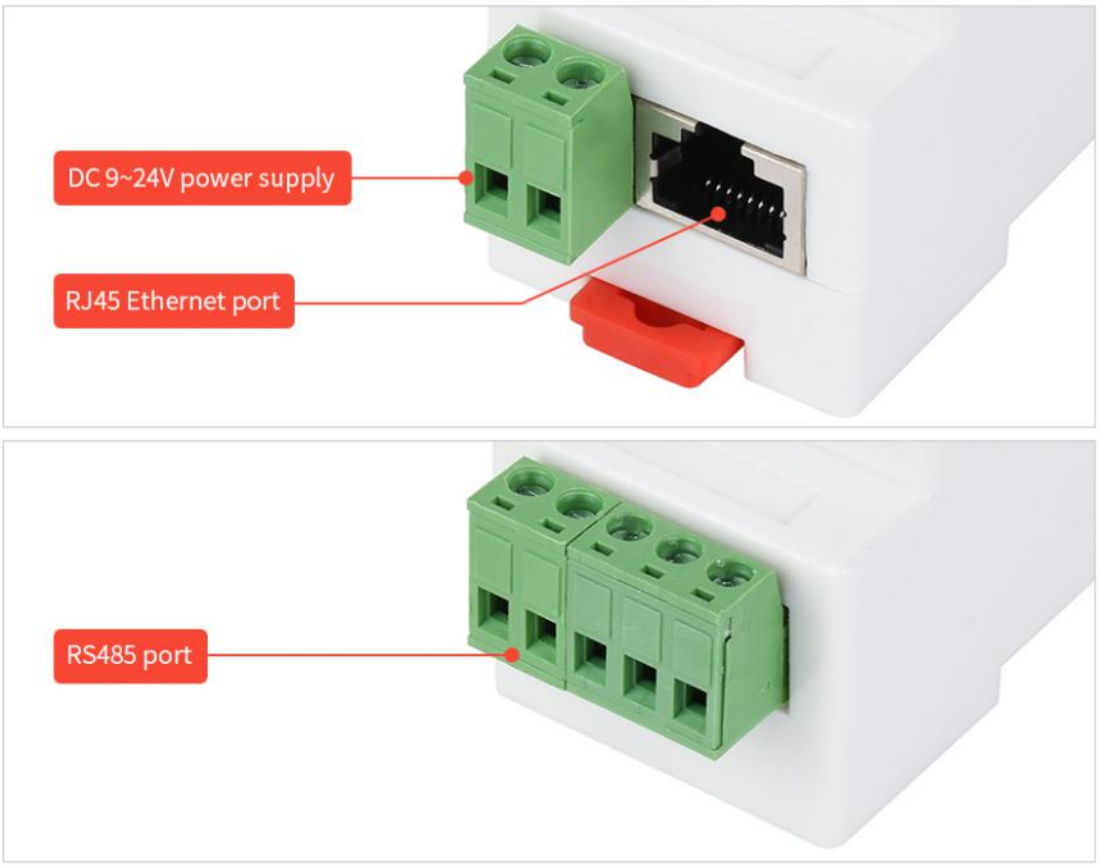


Figure 4 Power supply, network port

2 RS485: Use RS485 to connect to 485B and 485A. Among them, 485B means 485 negative line, 485A means 485 positive line; 485 can carry 32 sets of load. The longest communication distance is 1200 meters. Generally, the terminal resistance is only necessary when the RS485 line exceeds 300 meters. The 485 terminal resistance is 120 ohms.

3 Network port: Connect the network cable and support automatic crossover.

4 Indicator lights: Divided into Power (PWR), Link (Link), Active (ACT) lights. Respectively indicate power supply, connection, and data indication.

5 Reset: Short-circuit the two NC pins in the above picture for more than 5 seconds, the module will be reset to the IP of 192.168.1.254, and various downloaded configuration files will not be loaded.

Table 2 Indicator light meaning

Indicator light	
PWR	Power Indicator
LINK	(1) Link is green when the network cable is connected. (2) When the TCP connection is established (or in UDP mode), Link is blue (actually with a faint green light). It can be used to judge whether the serial port server has established a communication link with the host computer software.
ACT	(1) When the network port sends data to the serial port, the indicator light is green. The flashing time is 1 second longer than the actual communication time, which makes it easier to find short data communication. (2) When the serial port sends data to the network port, the indicator light is blue and green at the same time. Since blue is brighter, if you see blue, it indicates that there is a serial port returning data to the network port. This can determine whether the device responds to commands from the host computer. If there is no response, it indicates that the serial port baud rate is incorrect or the serial port is not connected properly.

Use indicator light to debug communication method:

1) If the Link light is not green, the network cable is not connected properly, please check the network cable.

2) If the Link light is not blue (only consider the TCP working mode), the host computer software does not have a serial port The server establishes a connection, please consider whether the IP address is configured in the same network segment.

3) If the Active light is green, it means that the network port is sending data, but there is no serial device returning data. Please check whether the baud rate is configured properly and whether the RS485 positive and negative are connected reversely.

6 Installation method: The equipment shell is equipped with a 35mm standard guide rail. If there is a guide rail, the equipment can be installed directly. To the guide rail.

Figure 5 The back of the device

4.2. HARDWARE CONNECTION

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

The power supply can be a 2-wire power supply on site, 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 device. Connect 485 positive to 485A and 485 negative to 485B. The network port is connected to a common network cable, which can be directly connected to a computer or connected to the network through a switch.

4.3. SOFTWARE INSTALLATION

Vircom can be used to configure parameters such as device IP and create virtual serial ports. If you do not need the virtual serial port function, you can only download the installation-free version of the configuration software.

Table 3 Software description

name of software	illustrate
VirCom	Chinese version configuration software
VirCom_en	English version configuration software
Virtual-serial-port	Virtual serial port software

4.4. PARAMETER CONFIGURATION

After Vircom is installed and the device hardware is connected, run the software as shown in the figure, and then click "Device Management" as shown in the figure. Using Vircom, you can

search and configure device parameters in different network segments, which is very convenient, as long as the device and the computer running Vircom are under the same switch.

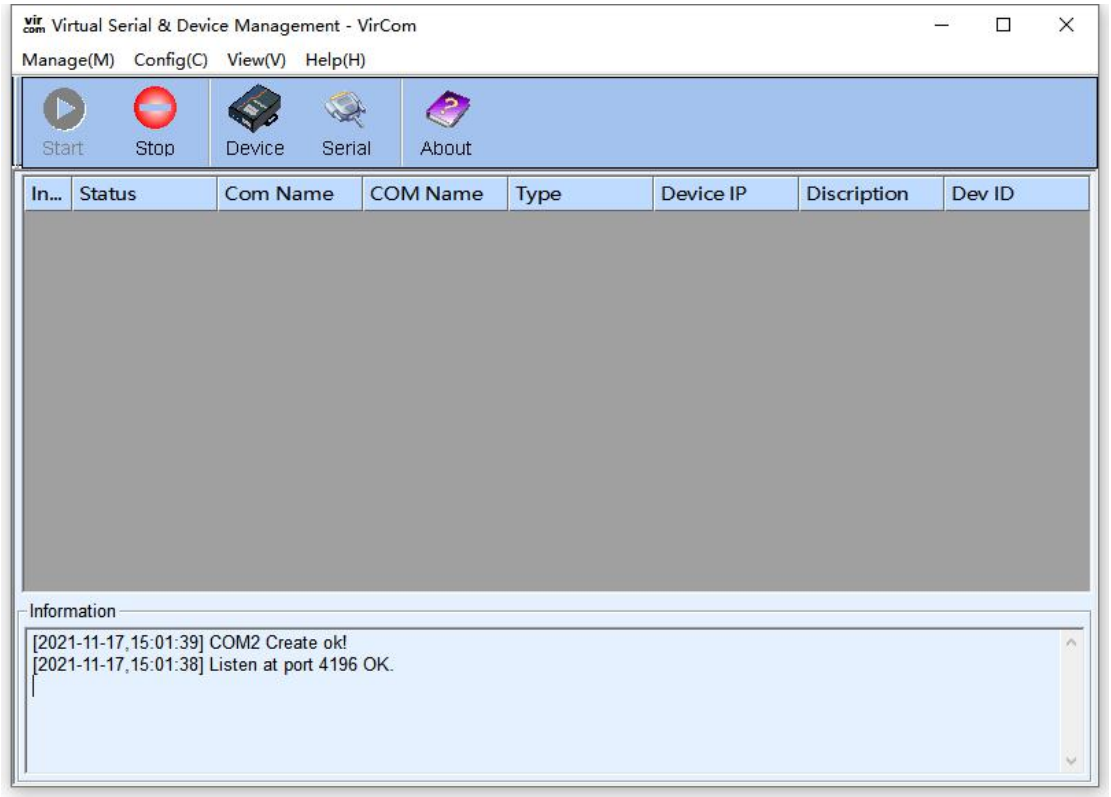


Figure 6 Vircom main interface

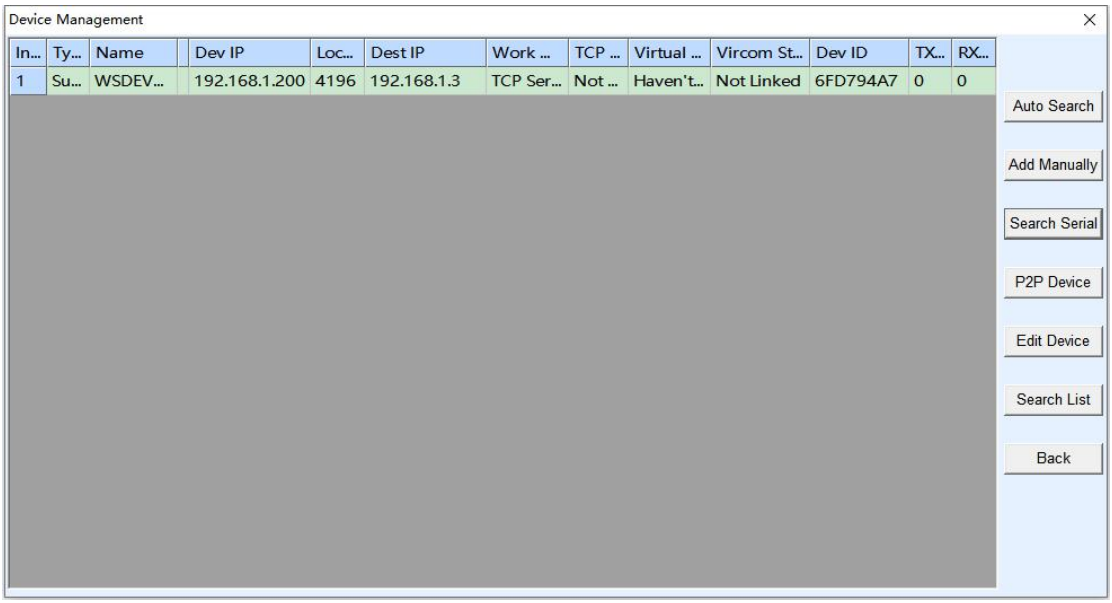


Figure 7 Device list

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

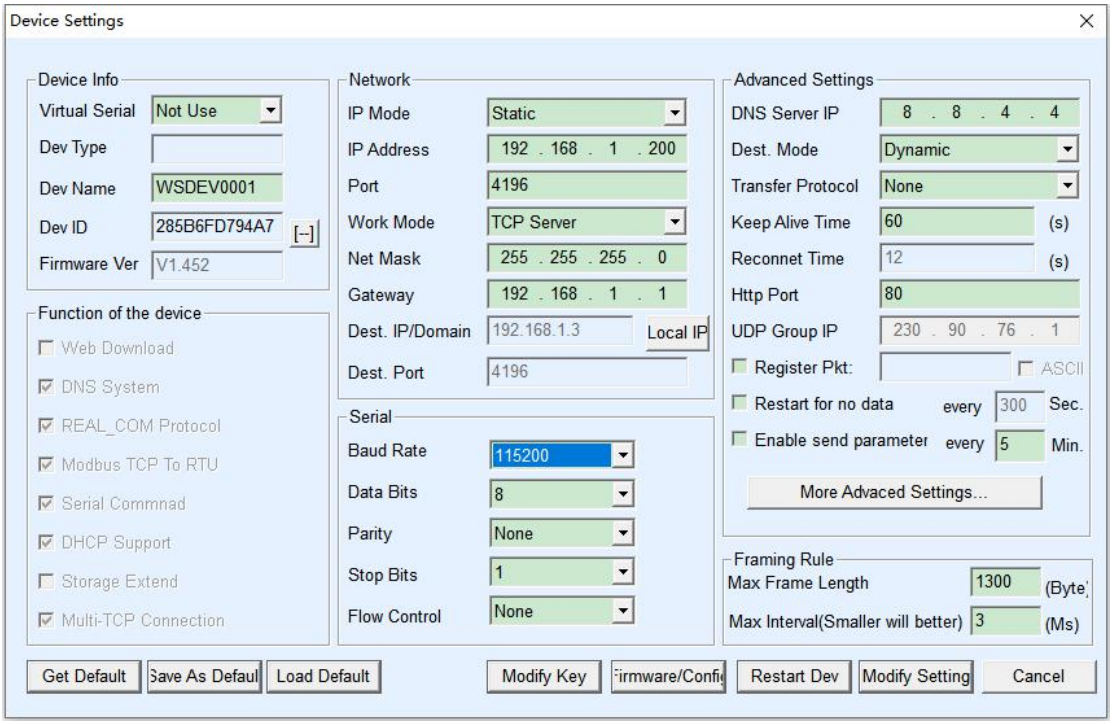


Figure 8 Device parameters

In this interface, the user can set the parameters of the device, and then click "Modify Setting", the parameters will be set to the flash of the device, and they will not be lost after power failure. At the same time, the device will automatically restart.

The main configuration parameters here are: baud rate, data bit, parity bit in serial port settings; IP address, subnet mask, gateway in network settings; sometimes according to computer software, you also need to configure the work of the serial server model.

The detailed meanings of other parameters are as follows:

Table 4 Parameter meaning

parameter name	Ranges	meaning
Virtual serial	port Unused, created virtual serial port	You can bind the current device with a created virtual serial port. Please add a COM port in the "Serial Port Management" on the main interface first.
Device model		Only display the model of the core module
Device name	Any	You can give the device an easy-to-read name, up to 9 characters long
Device ID		Factory unique ID, cannot be modified.
Firmware version		The firmware version of the core module
Supported		Refer to Table 5 Supported functions of the

functions		device
IP mode	static, DHCP	users can choose static or DHCP (dynamically obtain IP)
IP address		The IP address of the serial server
Port	0~65535	The listening port when the serial port server is in TCP Server or UDP mode. As a client, it is best to specify the port as port 0, which helps to increase the connection speed. When port 0 is used, the system will randomly allocate a local port. The difference between this and non-zero port is: (1) When the local port is 0, a new TCP connection will be established with the PC when the module restarts, the old TCP connection may not be closed, and the device may have multiple fakes. connect. 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 time for TCP to re-establish the connection is faster. When the serial port server is in TCP client mode, it also acts as a TCP server to monitor 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 to TCP server, the serial port server waits for the computer to connect; when set to TCP client, the serial port server actively initiates a connection to the network server specified by the destination IP.
Subnet mask	For example: 255.255.255.0	must be the same as the subnet mask of the local LAN.
Gateway	For example: 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	300、600、1200、2400、4800、7200、9600、14400、19200、28800、38400、57600、76800、115200、230400、460800	Serial port baud rate

Data bits	5、6、7、8、9	
Check Digit	None, Even, Odd, Mark, Space	
Stop bit	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 a domain name, this DNS server needs to be filled in. Server IP. When the IP mode is DHCP, there is no need to specify DNS Server, it will be automatically obtained from the DHCP server.
Destination mode	In static and dynamic	TCP client mode: After using the static destination mode, the device will automatically restart the device after 5 consecutive failures to connect to the server.
Conversion agreement	NONE 、 Modbus TCP<->RTU、 Real_COM	NONE means that the data forwarding from the serial port to the network is transparent; Modbus TCP<->RTU will directly convert Modbus TCP protocol to RTU protocol, which is convenient to cooperate with Modbus TCP protocol; RealCOM is designed to be compatible with the old version of REAL_COM protocol. It is a protocol of virtual serial port mode, but it uses virtual serial port. At this time, it is not necessary to choose the RealCom protocol.
Keep-alive timing time	0~255	Heartbeat interval. (1) When the selection is 1~255, if the device is in the TCP client working 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 it is set to 0~254, when the conversion protocol is selected as REAL_COM, the device will send a data with a length of 1 and a content of 0 every keep-alive timing time, which realizes the heartbeat mechanism in the Realcom protocol. When set to 255, there will be no realcom heartbeat. (3) When it is set to 0~254, if the device works as a TCP client, the device will send device parameters to the destination

		computer every keep-alive time. When it is set to 255, there will be no parameter sending function, which can realize remote device management.
Disconnection reconnection time	0~255	0~255 When in TCP client mode, when the connection is not successful, each "disconnection reconnection time" will re-initiate TCP connection to the computer. It can be 0~254 seconds. If you set 255, it means that you will never reconnect. Note that the first TCP connection (such as hardware power-on, device restart via Vircom software, no data light) will generally be carried out immediately, and only after the first connection fails will it wait for the "disconnected reconnection time" and try again, so "Disconnected reconnection time" will not affect the connection establishment time of the network and the server under normal conditions.
Web access port	1~65535	The default is 80
Adress of multicast		Used in UDP multicast
Enabling the registration package		When the TCP connection is established, the registration package is sent to the computer. After enabling the registration package, the realcom protocol must be selected. Support TCP server and TCP client mode.
Data packet length	1~1400	1~1400 One of the serial framing rules. After the serial port server serial port receives the length of data, it sends the received data to the network as a frame.
Data packet interval	0~255	

The functions supported by the device are explained as follows:

Table 5 Functions supported by the device

Name	Description
Domain Name System	The destination IP can be a domain name (such as the www server address at the beginning).
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 MAC address of the device is contained in the protocol, it is helpful for the host computer to identify the device. Under normal

	circumstances, it can not be used.
Modbus TCP to RTU	can be converted from Modbus TCP to RTU. It also supports multi-host function.
Serial port modification parameters	Support serial port AT commands to configure and read device parameters.
Obtain IP automatically	support DHCP client protocol
Storage expansion EX function	Subsequent expansion
Multiple TCP connections	When acting as a TCP server, it supports more than 1 TCP connection.
IO port control	Supports any custom commands to control 8 IO outputs.
UDP multicast	UDP multicast
Multi-destination IP	As a TCP client, it supports simultaneous connection to 7 destination IPs.

4.5. TCP COMMUNICATION TEST

After configuring the device parameters, you can use the serial port tool and TCP debugging tool to test the TCP connection communication.

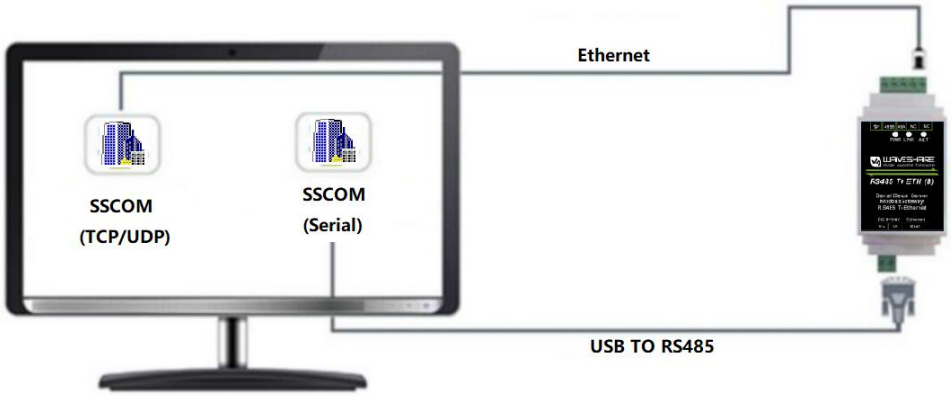


Figure 9 Schematic diagram of TCP communication

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:

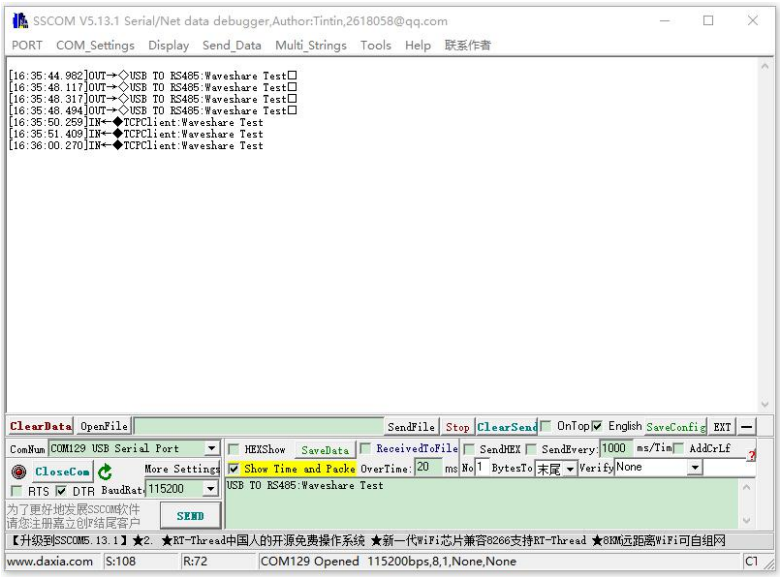


Figure 10 SSCOM1: USB TO RS485 transceiver interface

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:

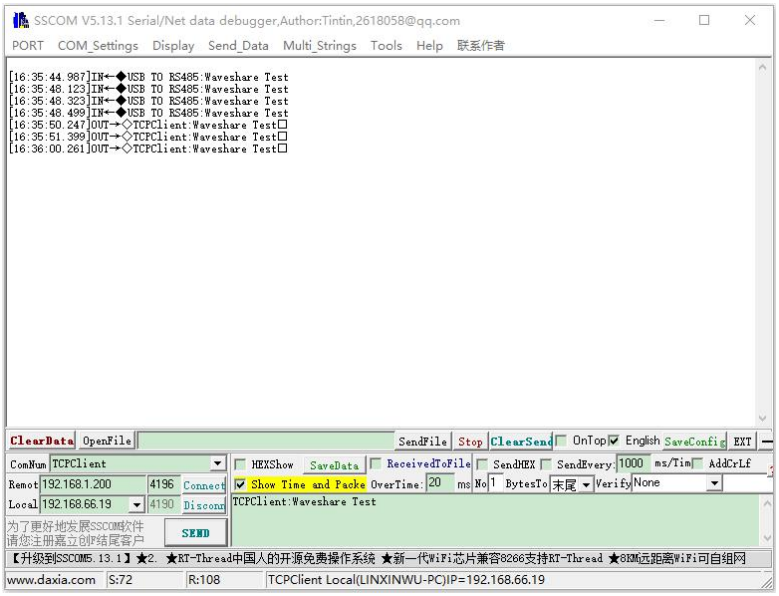


Figure 11 SSCOM2: TCP Client transceiver interface

Enter "TCPClient: Waveshare Test" in the serial debugging assistant SSCOM2 set as TCPClient and click send, then the data will be transferred to the RS485 interface through the serial server's network port, and then sent to the USB TO RS485, and then displayed in the serial debugging assistant SSCOM1 Come out; conversely, input "USB TO RS485: Waveshare Test" in SSCOM1, and click Send to send to SSCOM2 and display it.

4.6. VIRTUAL SERIAL PORT TEST

The SSCOM2 in the figure communicates directly with the serial server through TCP. In order to allow the user to communicate with the serial server as well as the serial software that has been developed by the user, a virtual serial port needs to be added between the user program and the serial server. As shown in the figure, Vircom and the user 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:

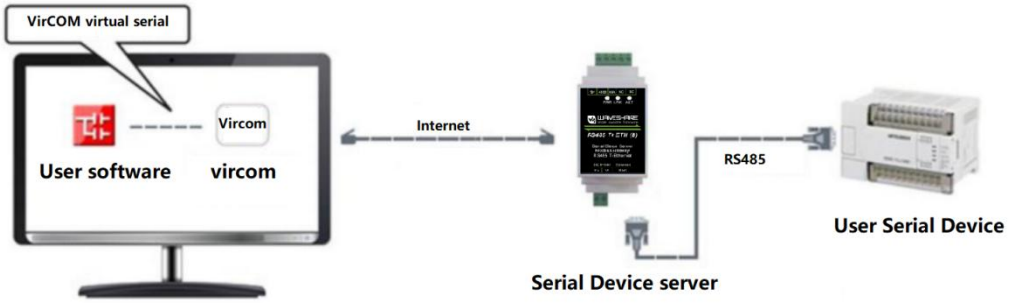
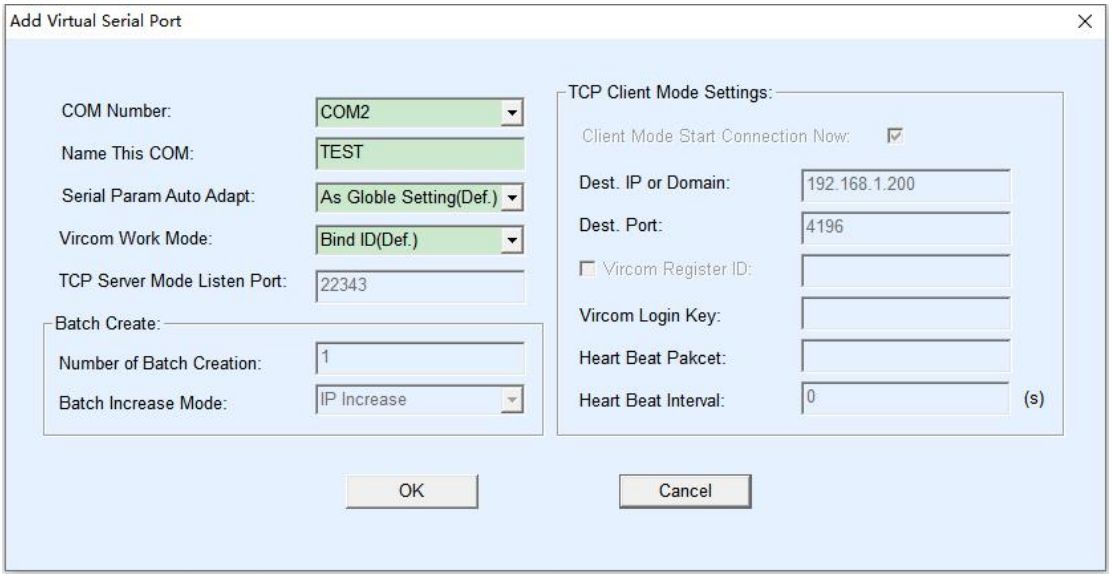


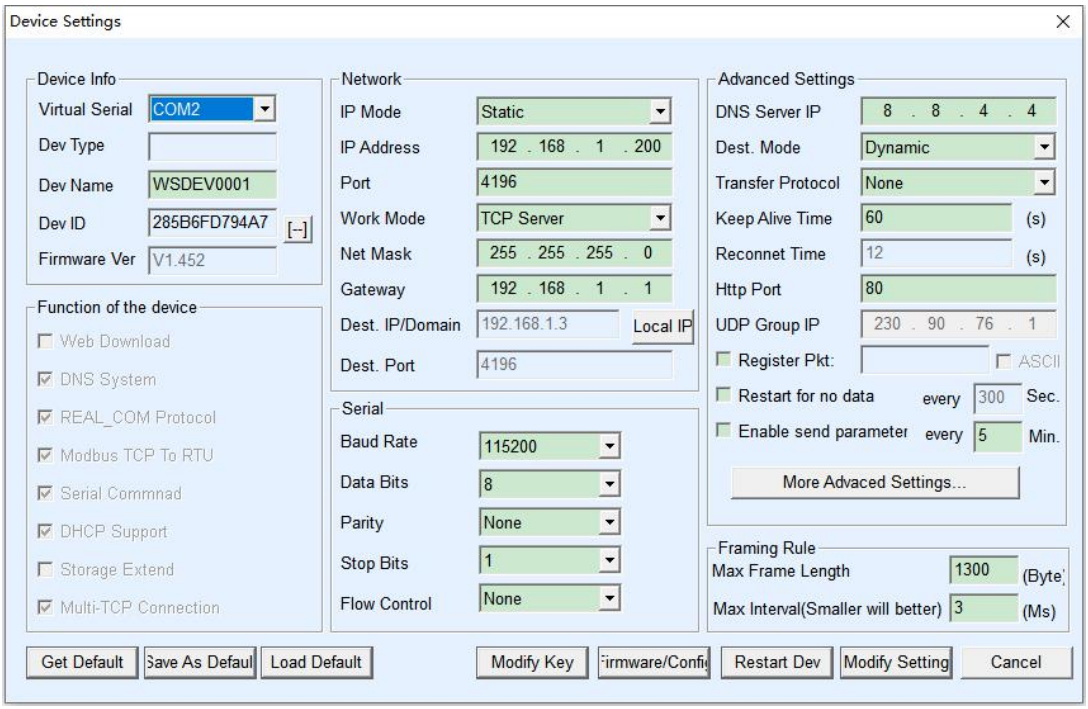
Figure 12 Application example of virtual serial port

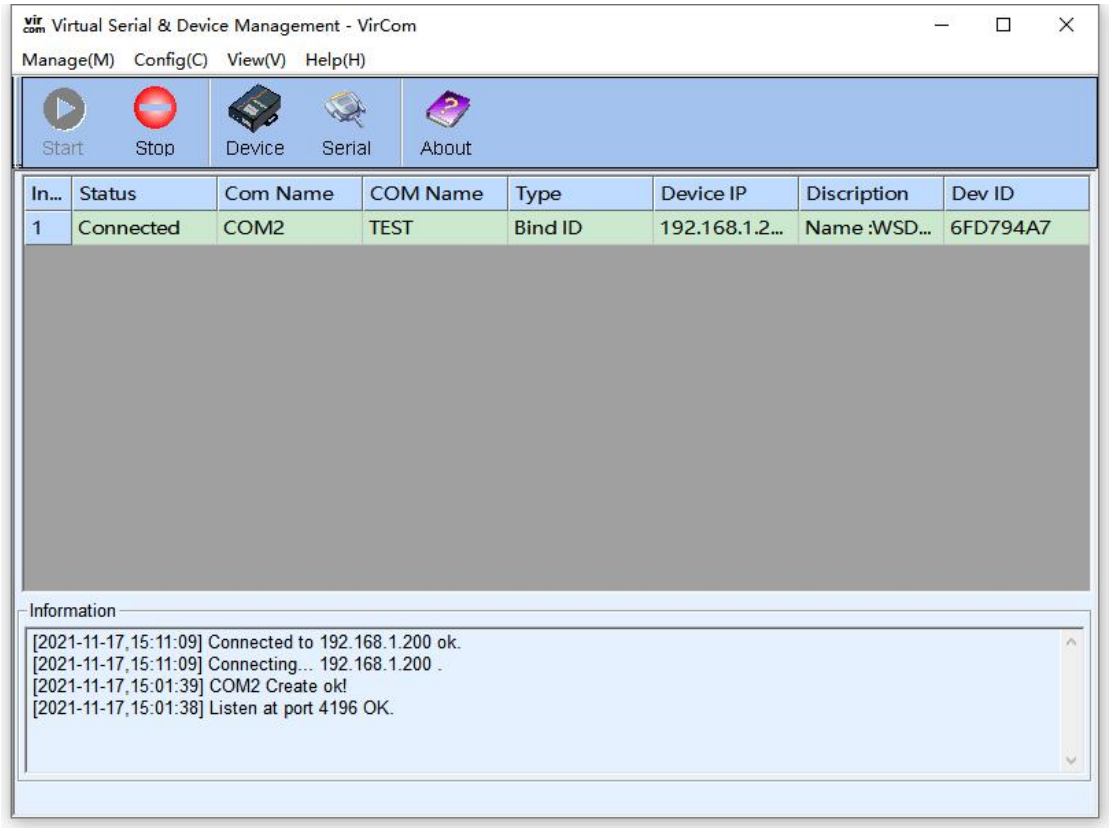
Click "Serial" on the main interface of Vircom, then click "Add", and choose to add COM2, where COM5 is a COM port that does not exist in the computer.



Add virtual serial port

Then enter the device management, and double-click the device that needs to be bound to COM2. As shown in the figure, select COM2 in the "Virtual Serial Port" list in the upper left corner. Then click "Modify Setting", and then click "Restart Dev". And return to the main interface of Vircom. You can see that COM2 has been connected to the device with IP 192.168.1.200. At this time, COM2 can be used instead of SSCOM2 for communication.



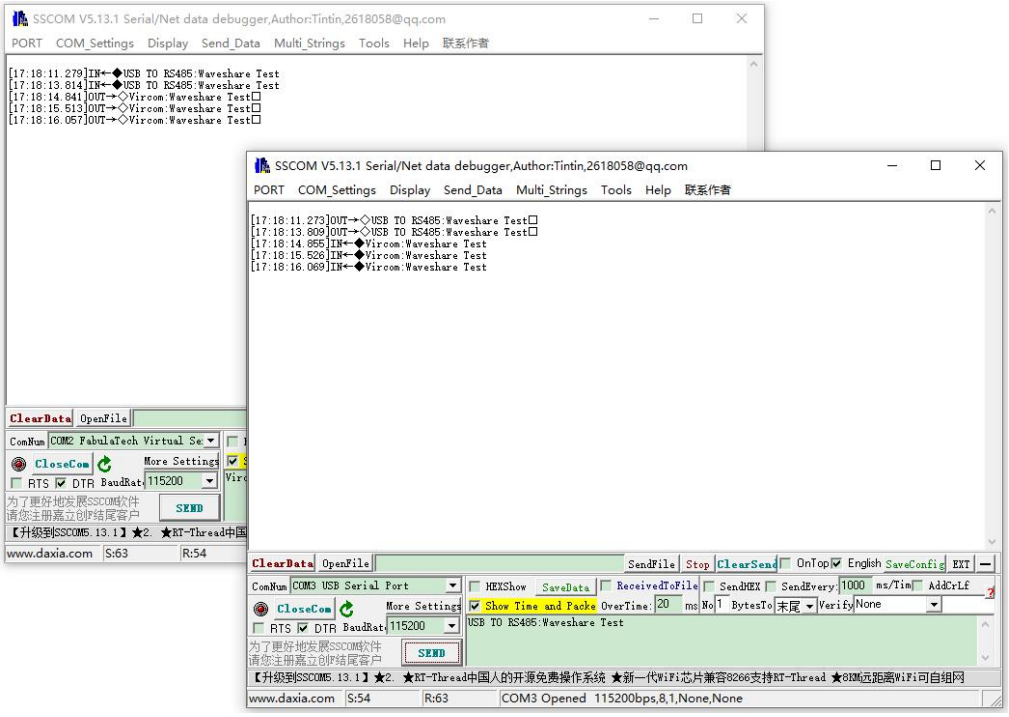


The virtual serial port has been connected

Open SSCOM to simulate the user's serial port program, open COM2 (the virtual serial port above), open another SSCOM to simulate a serial device, and open COM3 (hardware serial port). At this time, the COM2 sending data link is as follows: COM2 —"Vircom —"Serial server network

port —"Serial server serial port —"COM3. Conversely, COM3 to COM2 can also transmit data: COM3 —"Serial server serial port —"Serial server network port —"Vircom —" COM2. Figure shows the situation of both parties sending and receiving data.

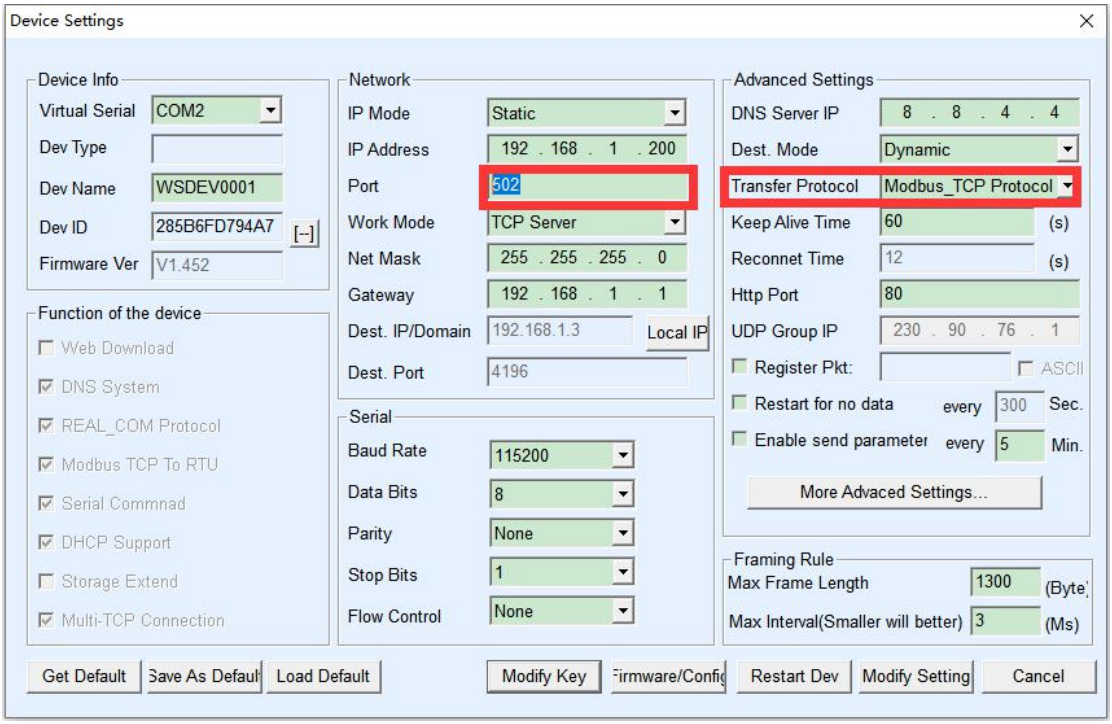
If COM4 is replaced by a user serial port device, COM5 can communicate with the user device.



Communication via virtual serial port

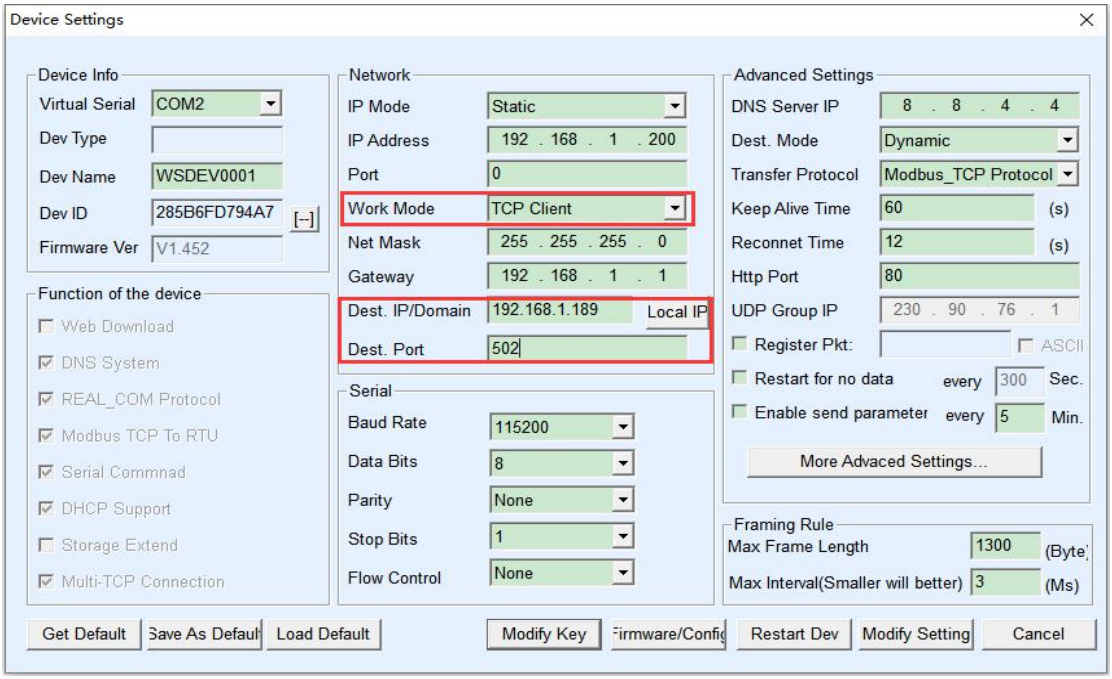
4.7. MODBUS TCP TEST

By default, serial and network port data 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 dialog box, as shown in the figure below. At this time, the device port is automatically changed to 502. At this time, the user's Modbus TCP tool is connected to the 502 port of the serial server's IP, and the sent Modbus TCP commands will be converted into RTU commands and output from 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 01 03 00 00 00 0a c5 cd commands. 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.



Enable Modbus TCP function

If the user's Modbus TCP software is used as a slave (Slave), you need to change the working mode to client based on the selection of 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 blow.

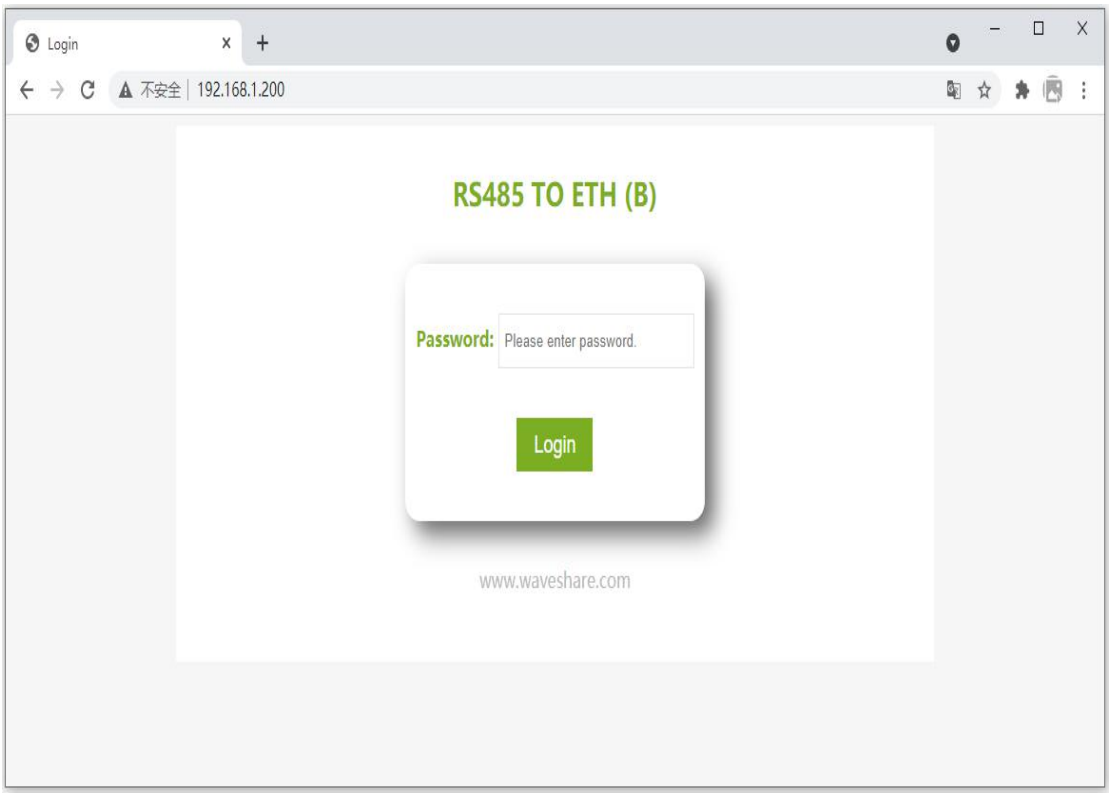


Modbus TCP as client

4.8. WEB CONFIGURATION

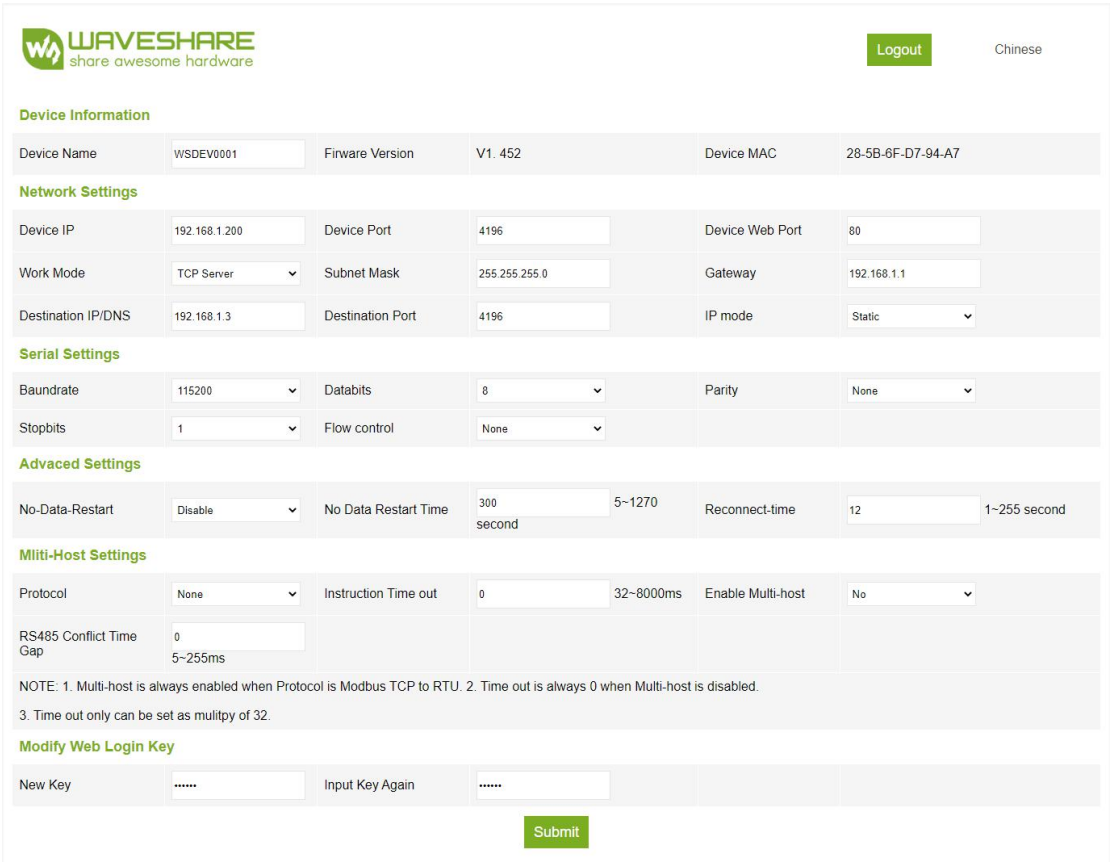
Using Vircom, you can search and configure device parameters in different network segments. Web configuration requires that the computer and serial server are in the same IP segment, and the IP address of the serial server must be known in advance. But the Web configuration can be done on any computer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

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



Web login interface

2. Enter the password in Password: there is no password set by default in the factory state, and the password needs to be set through Modify Web Login Key to take effect.



Web configuration interface

3. In the web page that appears, you can modify the serial server parameters. For related parameters, please refer to Table 4 for the meaning of the parameters.

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

5. WORKING MODE AND CONVERSION AGREEMENT

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 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 user software with virtual serial port that needs to be docked is a serial interface (COM port), that is, the user software and user equipment are serial ports; the user software of the non-virtual serial port mode communicates directly with TCP/IP, but the user equipment is still a serial port.

In the 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, you

need to select Modbus TCP to RTU mode; Realcom protocol is currently only used in a multi-serial server as a TCP client to connect to a server, and the virtual serial port is used on the server. Time to use.

The usage is summarized as follows:

Table 6 Network configuration mode

NO.	Virtual serial port	Device working mode	Conversion protocol	Description
1	Use	TCP server	None	Suitable for occasions where user software opens the COM port to actively collect data.
2	Use	TCP client	None	Suitable for occasions where the device actively sends data. If you choose a TCP server, the problem that the device cannot reconnect after disconnection may occur.
3	Do not use	TCP server	Modbus TCP to RTU	Applicable to the user software is Modbus TCP, and 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, and the user device is Modbus RTU. And Modbus RTU is the master station.
5	Use	TCP client	Realcom protocol	Multi-serial server as TCP client, and use virtual serial port, it is best to use Realcom protocol.
6	Do not use	TCP client	None	Applicable to a large number of devices connected to a cloud. And in general, the cloud is a public IP server on the Internet.
7	Do not use	TCP server	None	Suitable for devices and computers in the same local network, monitoring locally, without cross-Internet communication.

5.1. VIRTUAL SERIAL PORT 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.

Look at whether the monitoring computer and equipment are in the local network:

a) If the computer is a public IP server leased on the Internet, 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 6, if it is a multi-serial server, you must choose ⑤.

b) If they are all on the local network (can ping each other), it depends on whether the host computer actively inquires or the device actively sends data. If the device actively uploads, the ② method of using the device as a TCP client must be used, otherwise the ① method can be selected.

5.2. DIRECT TCP/IP COMMUNICATION MODE

If Modbus TCP protocol conversion and virtual serial port are not needed, the user software may directly communicate with the serial port server's network port for TCP/IP communication, and the serial server converts TCP/IP data into serial data and sends it to the serial device.

Generally, users of this kind of usage develop their own upper computer network communication software, 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" mainly briefly describes how to communicate when the serial port server is used as a TCP server. Here will describe how TCP client, UDP mode, and multiple TCP connections communicate with computer software. Among them, the computer software takes SSCOM (serial port debugging, software that imitates user TCP/IP communication) as an example.

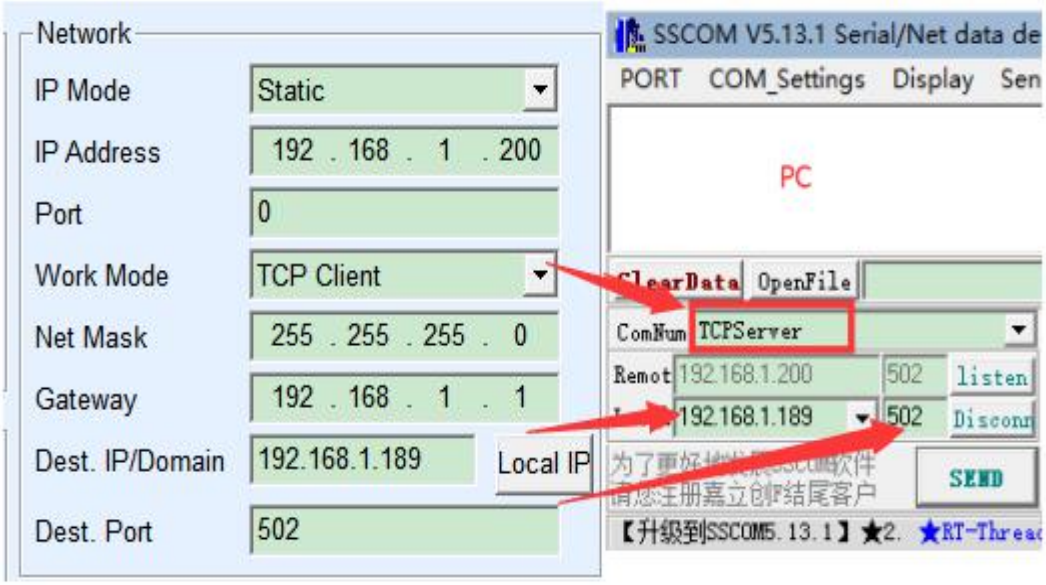
The serial server complies with the standard TCP/IP protocol, so any network terminal that complies with this protocol can communicate with the serial server. For two network terminals (here, the network debugging tool and the serial server) to be able to communicate, their parameter configuration must 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, the client can access the server, both of which are either client or service. Communication cannot be achieved at the end.

When the serial port 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.



Serial server as a client

5.2.2. 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 at the same time. If there are not so many servers, leave the rest of the destination IP vacant. The method of use is as follows:

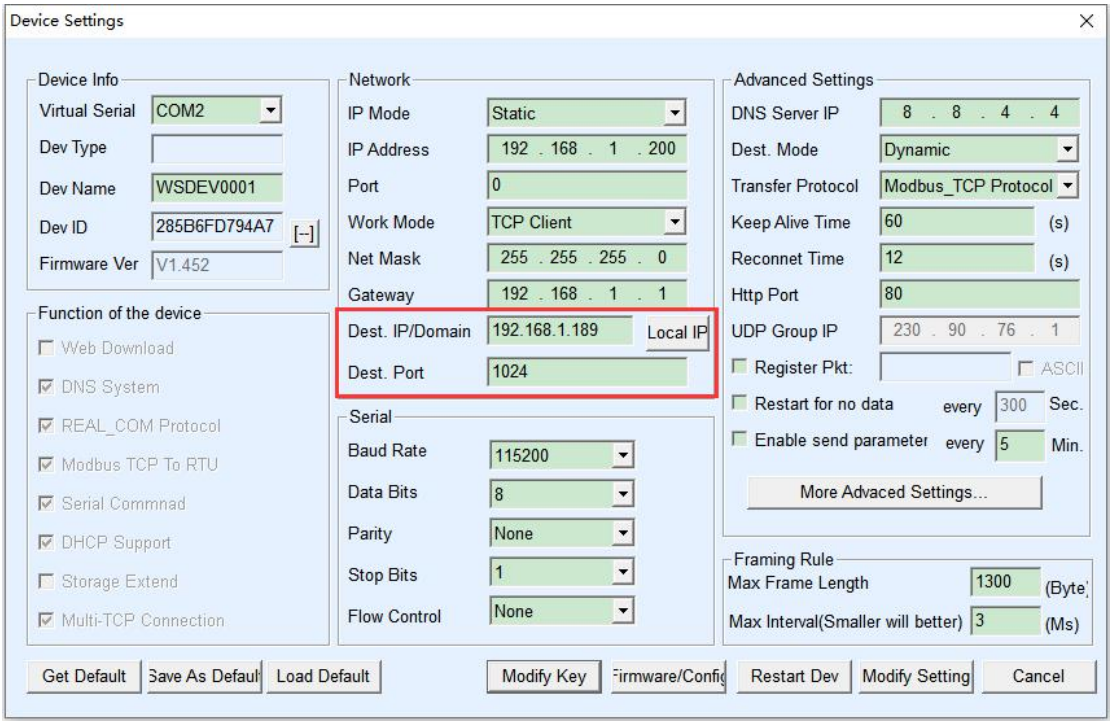


Figure 14 The first destination IP and port

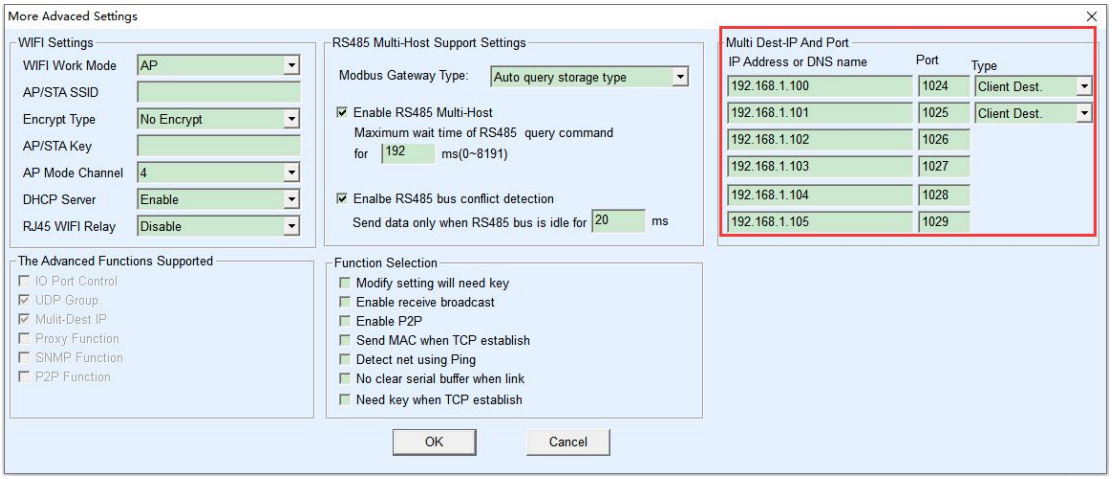


Figure 15 The remaining 2~7 IPs and ports

The first IP is set in the device setting interface as shown in Figure 14. 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 16, 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.

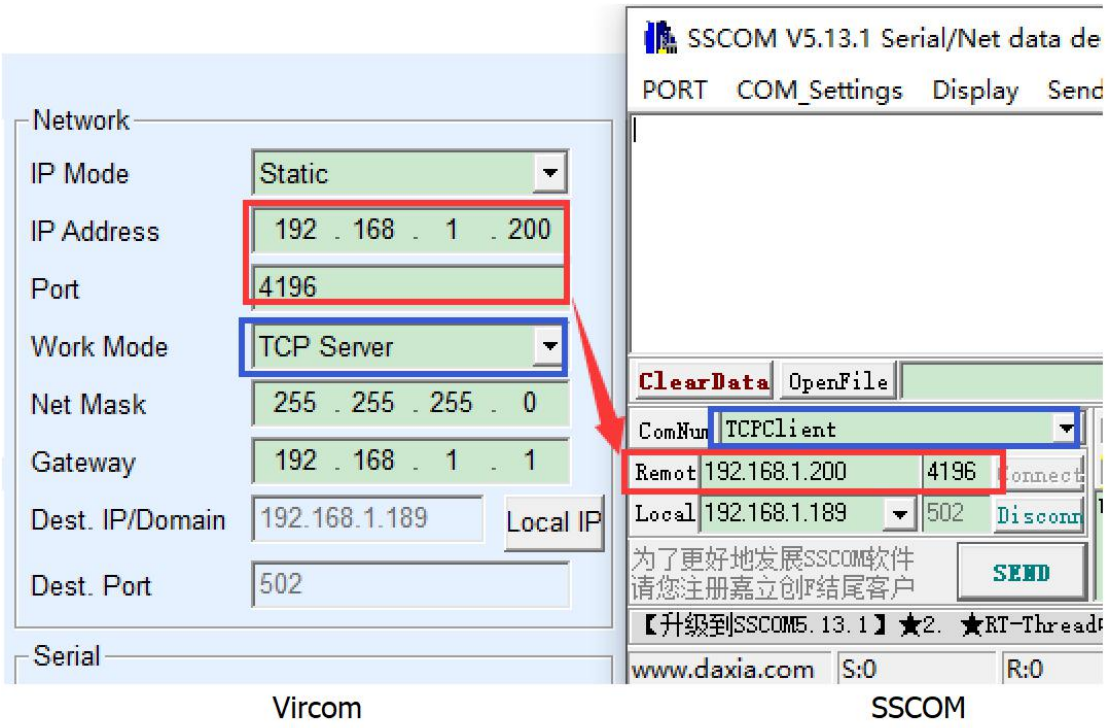


Figure 16 Serial port server as TCP server

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 A CLIENT AND A SERVER

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

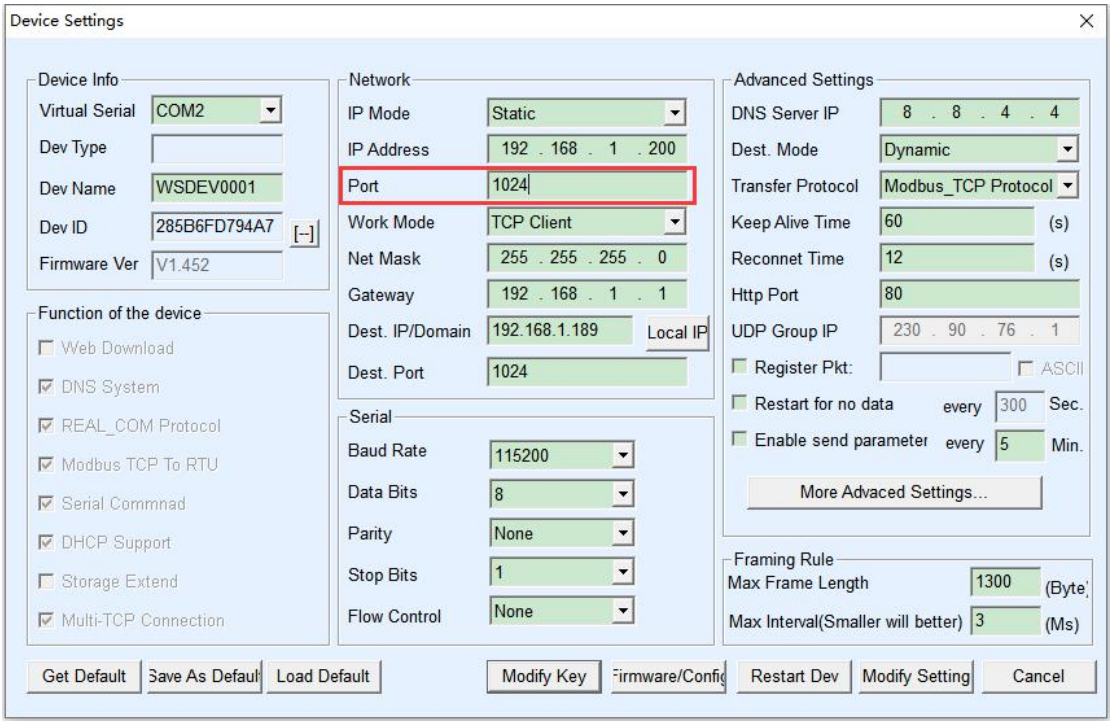


Figure 17 Acting as both a client and a server

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 be able to support the TCP server mode, the computer software must know the local port of the device, so a value needs to be specified here. As shown in Figure 17, 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 Connect to 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 18. The left side is the configuration of the serial server in Vircom, and the right side is the setting of the serial debugging tool SSCOM. First of all, both must be UDP working modes. In addition, indicated by the red arrow, the destination IP and 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 IP address of the computer 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.

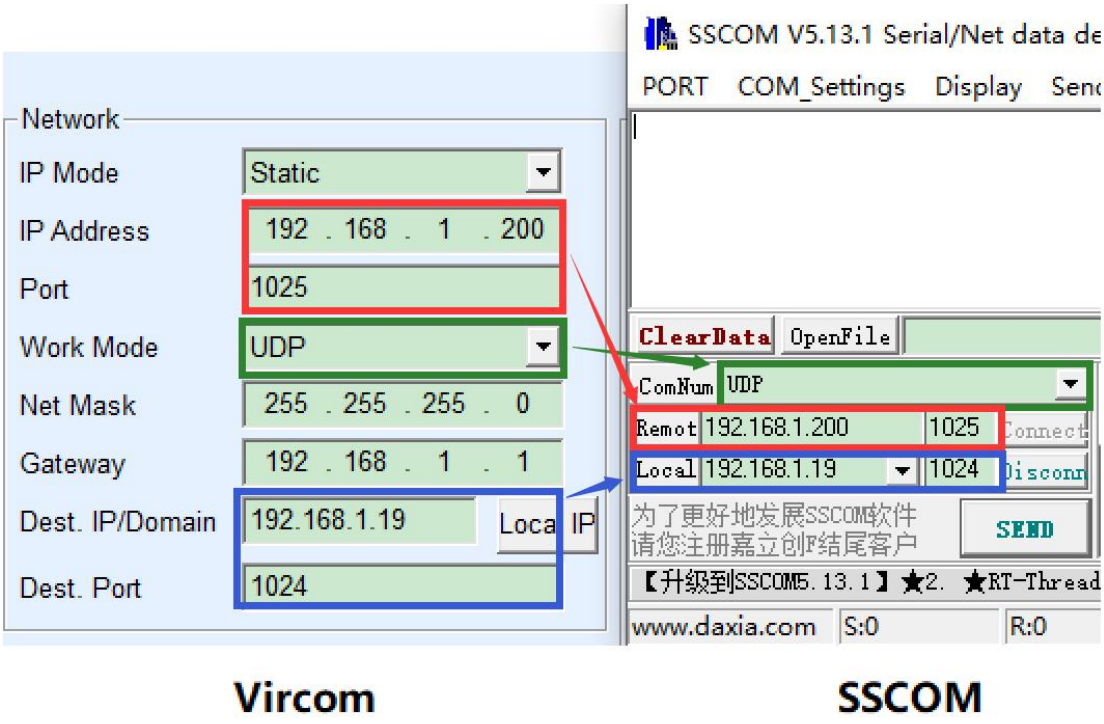


Figure 18 UDP mode parameter configuration

5.3. DEVICE COUPLET WAY

If the host computer is not an SSCOM program or a 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 computers to the same local area network. Vircom is running on this computer. 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 2 devices, as shown in Figure 20. Then click "Edit Device" to configure the device. Device couplets can be divided into TCP couplets and UDP couplets. If it is TCP couplet, the parameters of the two devices are shown in Figure 19. The parameters shown by the arrows must correspond, just like the corresponding 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 20. If the status of the two devices is both "Connected", it means that the TCP link between the two devices has been

established.

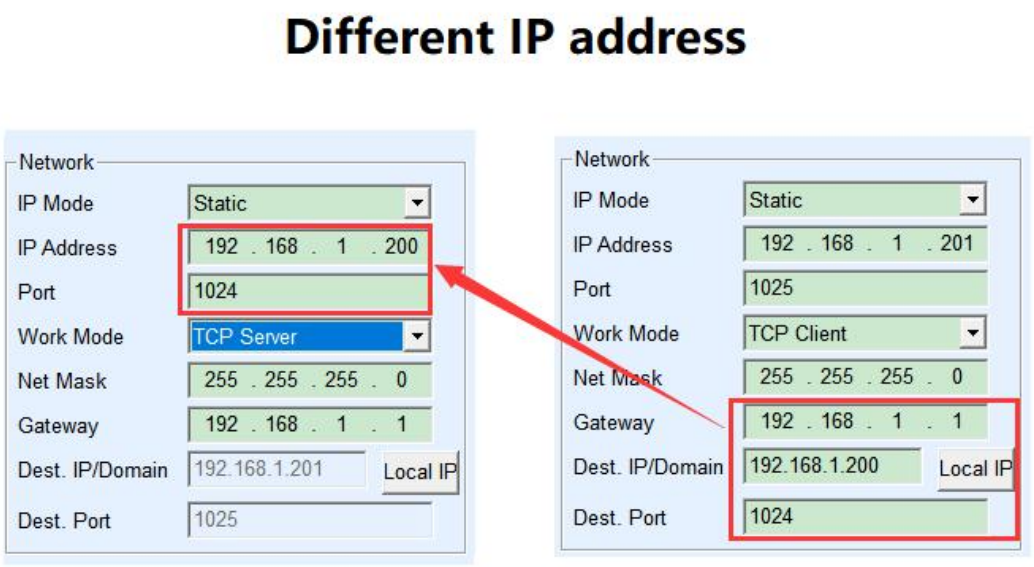


Figure 19 TCP device pairing parameter configuration

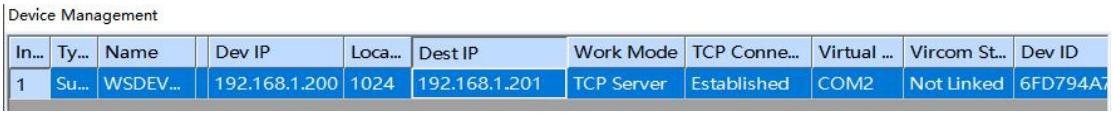


Figure 20 TCP device pairing parameter configuration

If it is couplets in UDP mode, the configuration parameters are shown in Figure 21, and the parameters corresponding to the arrows must be one-to-one correspondence. UDP couplet does not need to check the connection status as long as the parameter configuration is correct, the sent data will be automatically sent to the designated device.

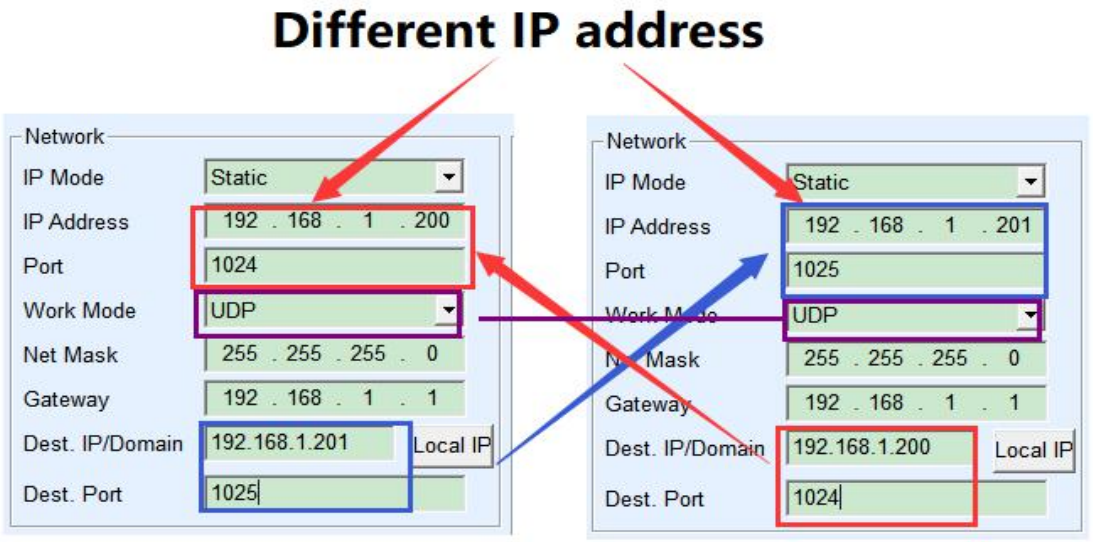


Figure 21 UDP device pairing parameter configuration

Finally, I 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 baud rate of the user's device. After this setting, the user equipment can send data to each other through the serial ports of the two serial port servers.

6. EQUIPMENT DEBUGGING

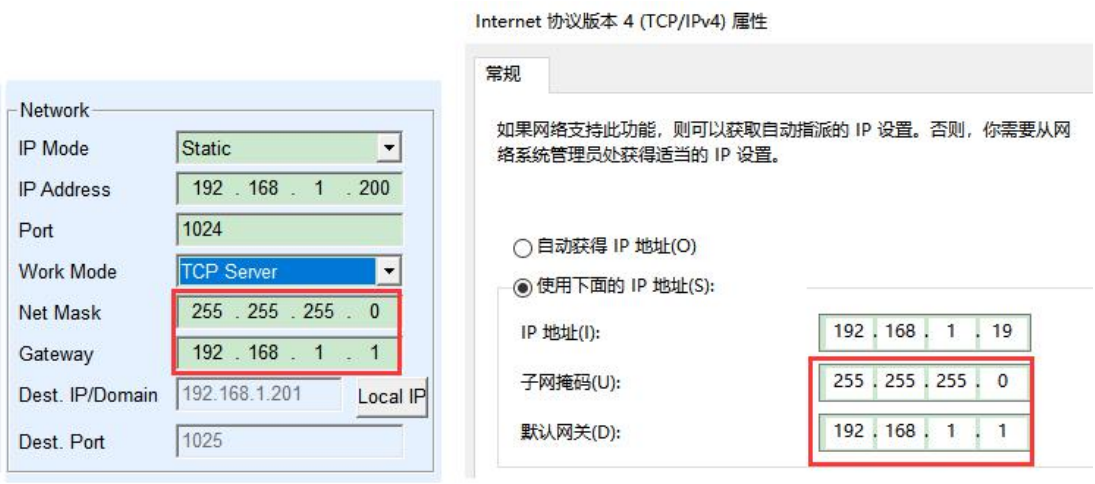
6.1. NETWORK PHYSICAL CONNECTION

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

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 way to dynamically obtain IP, it cannot be directly connected to the computer network port. Because there is no DHCP server available (generally a DHCP server is a router in the LAN). So please specify the IP when connecting directly. At the same time, the computer also needs to specify a fixed IP.



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 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 properly. In this case, you can use Vircom to configure the device in the same network segment.

After configuration, use the steps of 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 22. This can facilitate remote diagnosis.

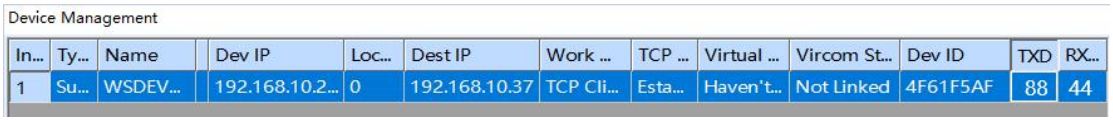


Figure 22 Connection status and data receiving and sending status

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. At this time, 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 correct serial port parameters (baud rate, data bit, stop bit, parity bit) are configured. The serial device generally responds to the correct command issued. Once there is a reply (the serial port sends data to the network port), the Active 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 22, 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 REMOTE MONITORING DATA

In the case of using a virtual serial port, Vircom supports real-time capture of the data sent and received by the virtual serial port. It is convenient for users to debug the system, and the method of use 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 need to monitor the data through the virtual serial port. Open Vircom's menu/Config/software setting/open the Vircom configuration dialog box.

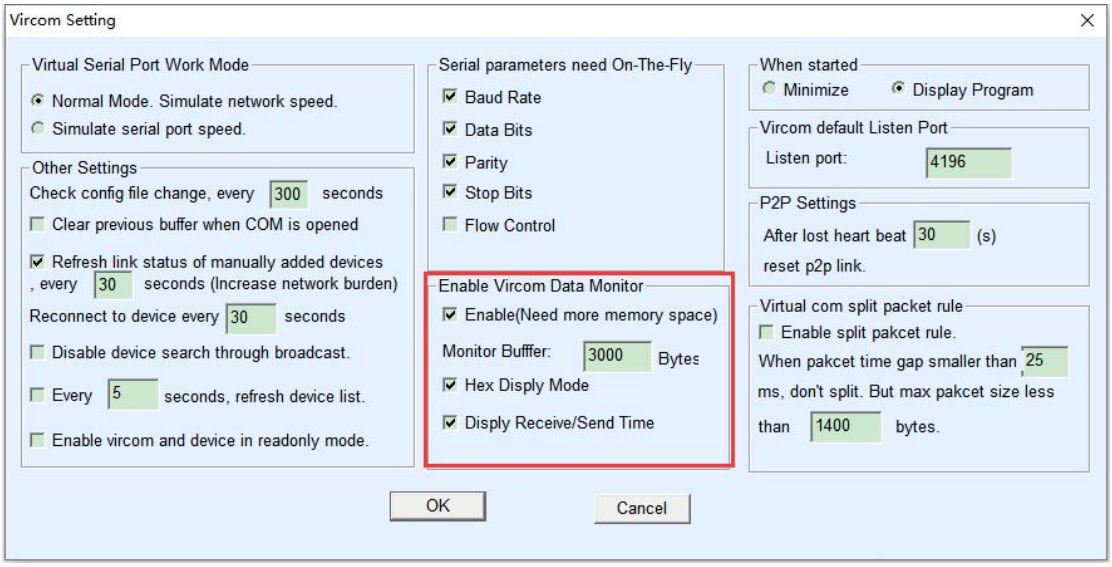


Figure 31 Enable Virocm monitoring

Tick in front of the three options: open monitoring, hexadecimal monitoring mode, and display data sending and receiving time, as shown in Figure 23. 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 24.

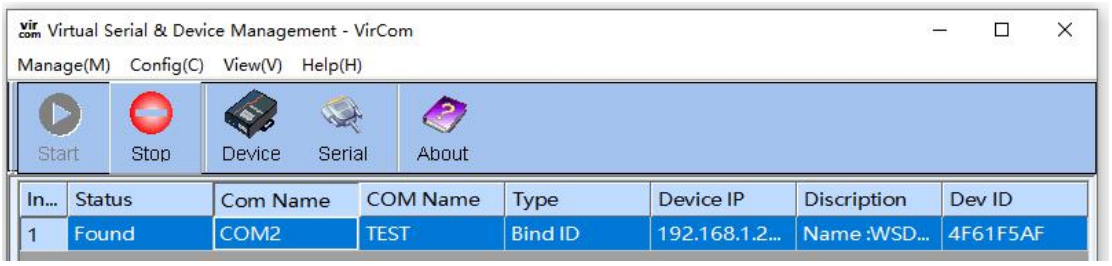


Figure 32 Open Virocm monitoring

From the opened dialog box, you can see the instructions issued by the host computer and the instructions returned by the device, as shown in Figure 25. This function can facilitate on-site communication debugging.

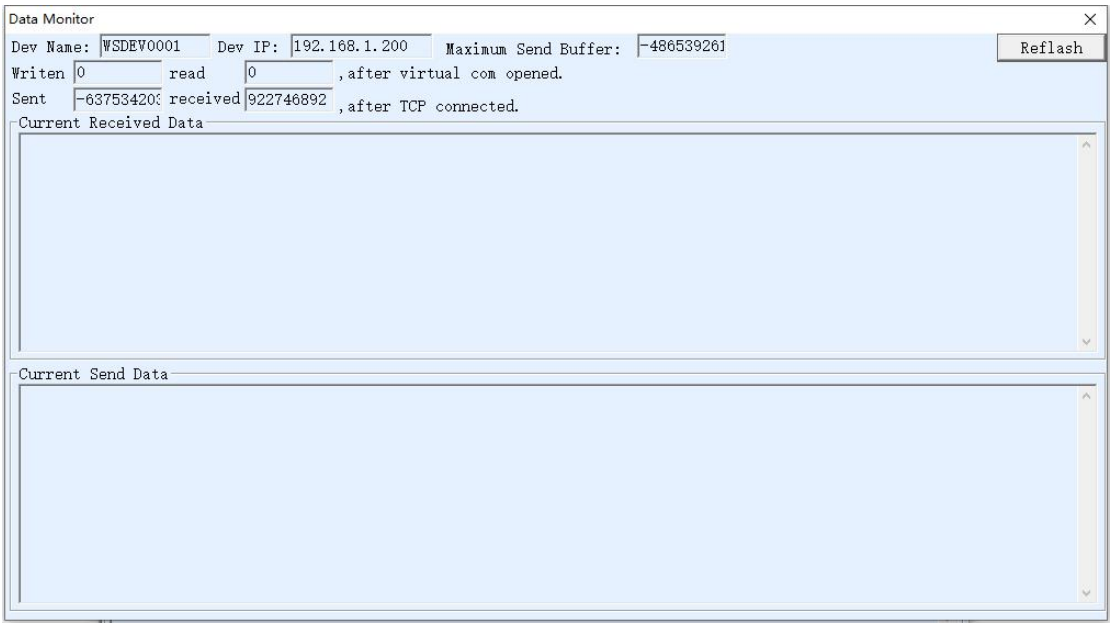


Figure 33 Monitor sending and receiving data

7. MODBUS ADVANCED FUNCTIONS

The serial port server with Modbus gateway function does not have station address and registers. It is a communication bridge. It will generate Modbus according to the Salve ID, function code, register number and number of registers in the Modbus TCP command sent by the user software to the Modbus gateway. RTU designates and outputs from the serial port. Think of it 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 5 Device Supported Functions in the Device Settings dialog box should be ticked.

By default, the serial server is in normal transparent transmission mode. If you need to switch to Modbus gateway mode, please select the option "Modbus TCP Protocol" in "Conversion Protocol". After that, the device automatically changes the "port" parameter to 502 (the port of the Modbus server). Then the Modbus gateway is enabled.

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

7.2. STORAGE MODBUS GATEWAY

This product can save the content of the read register inside the gateway, so that the speed of Modbus TCP query can be greatly improved, and the performance is even better when it supports multi-host access.

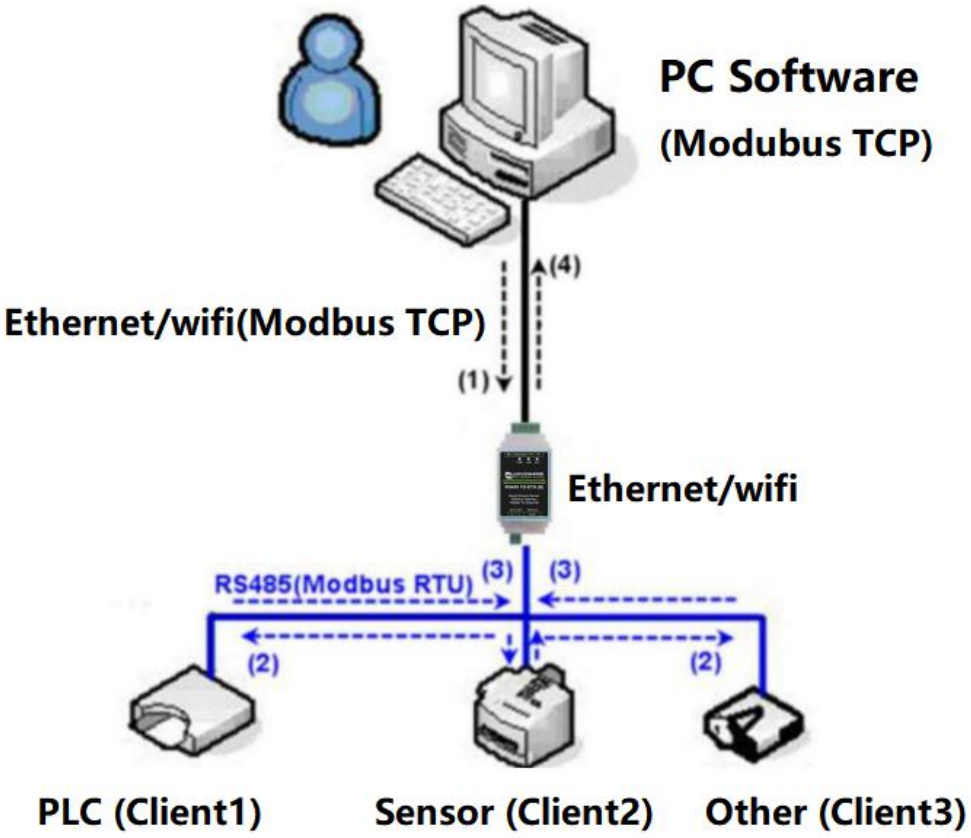


Figure 26 Working Mode of Storage Modbus Gateway

As shown in Figure 26: The general Modbus TCP data flow direction 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 is converted to Modbus TCP again and sent to the monitoring host computer.

We know that Modbus TCP is network communication, and the transmission speed is very fast. Generally, it can respond within 3ms, while Modbus RTU is RS485, which generally has a speed of 9600bps. It usually takes at least 30ms to send and return a command. The query response time of such ordinary non-storage Modbus gateways is relatively long. In addition, if there are a lot of host computers at the same time to query data at the same time, then the serial port will be congested. If the network is like an expressway, and the serial port is like a

single-plank bridge, then the original method is to pass the traffic flow of the expressway on the single-plank bridge.

The register-saving Modbus gateway solves the above problems. It can temporarily save the register data obtained by the query in the Modbus gateway, so that when the Modbus TCP query comes, the Modbus gateway can immediately return the command, which truly brings out the fast characteristics of Modbus TCP. 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.

RS485 TO ETH (B) can show good response speed when monitoring by multiple computers. No matter what baud rate of the serial port is, it can generally reply data to the host within 3ms. And showed a good real-time update 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 type 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 multiple commands are being queried at the same time, they are sent in order. The first command sends the first command response and 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 FUNCTIONS

Although the storage Modbus has a faster response speed, some users do not want the RTU device to receive a large number of query commands, which affects the internal processing speed of the instrument. The storage function can be turned off at this time.

The method to disable the storage type is to click the "More Advanced Settings" button in the "Parameter Configuration" dialog box, remove one support and one enable as shown in Figure 27 and click OK. Then go back 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.

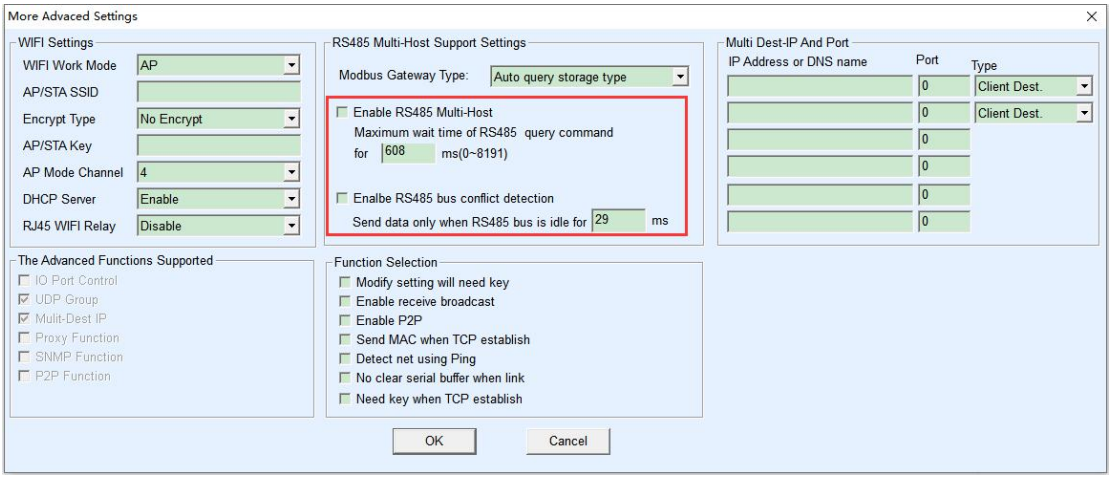


Figure 27 Disable storage function

7.4. MULTI-HOST FUNCTION

As shown in Figure 27, "Enable RS485 Multi-Host" and "Enable RS485 bus conflict detection" are multi-host functions. They are generally enabled and disabled at the same time. After enabling, let the device whose conversion protocol is Modbus TCP has the storage Modbus gateway function, otherwise it is a non-storage Modbus gateway; if the conversion protocol is None, generally the user-defined RS485 protocol can also have serial devices accessed by multiple hosts at the same time This function cannot be achieved in a pure RS485 network, because multiple master stations sending at the same time will cause conflicts on the RS485 bus. The multiple hosts of this serial server can "coordinate" the RS485 bus to achieve the purpose of multi-host access.

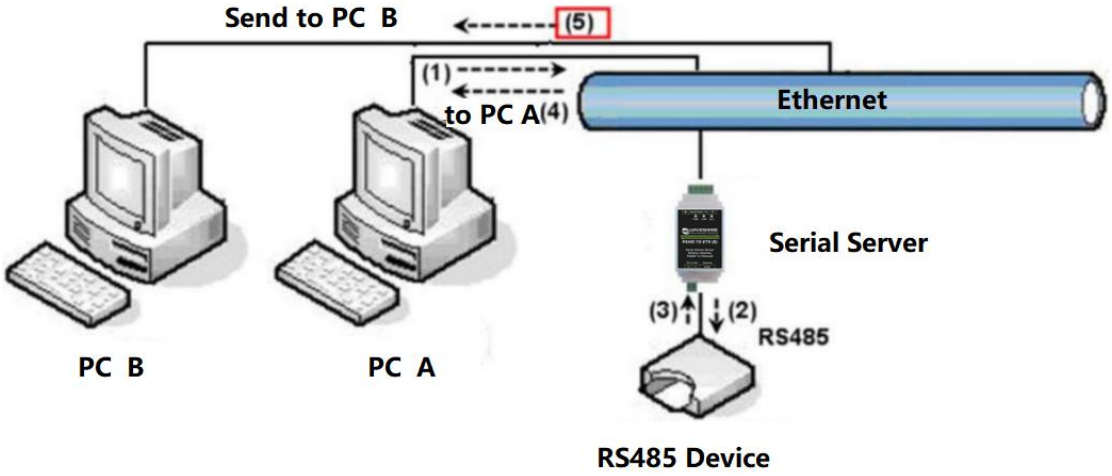


Figure 28 Multi-host function demonstration

As shown in Figure 28, 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 instructions (4) but not instructions (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 PARAMETERS

The meanings of "Enable RS485 Multi-Host" and "Enable RS485 bus conflict detection" are introduced as follows.

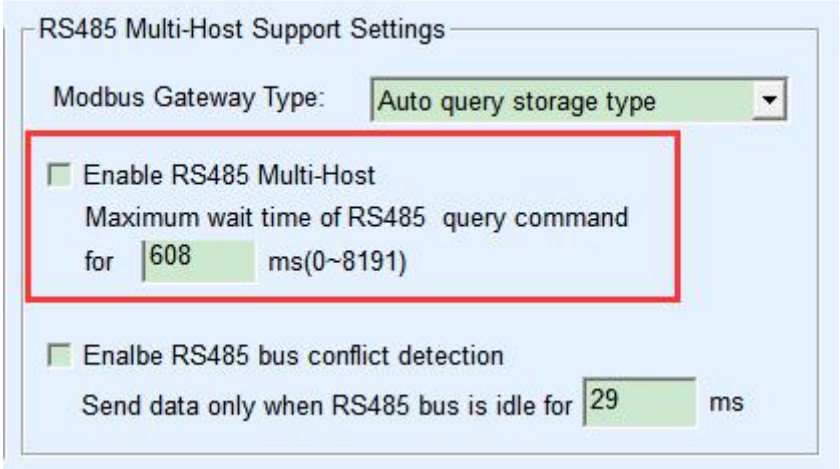


Figure 37 RS485 multi-host support

The RS485 command response timeout time is: the maximum time interval from the beginning of sending this command to receiving the response from the serial server serial port. The time to fill in should be greater than the actual maximum time interval. Because once it is determined to be timed out, the next command will be sent.

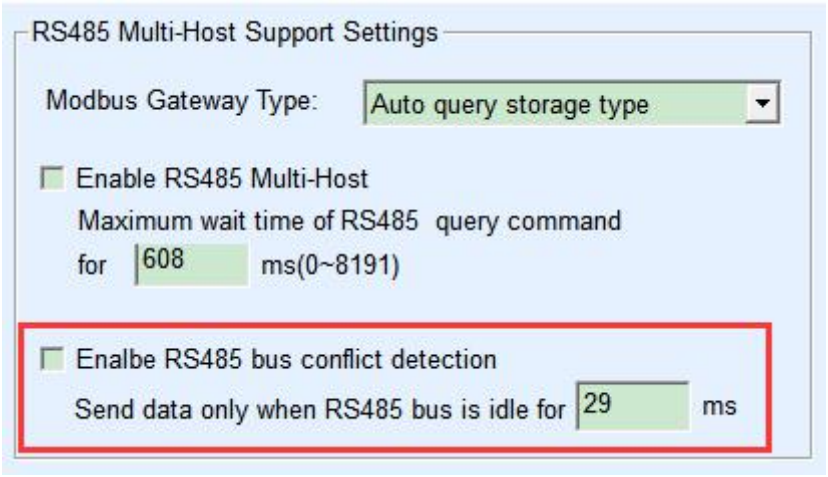


Figure 30 RS485 anti-collision idle time

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 Modus 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 30 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 29 shows the "RS485 command response timeout time", 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 28, if the serial port 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 checkmarks shown in Figure 30 as "RS485 bus anti-collision time" and Figure 29 as "RS485 command response timeout time". Otherwise, the above forwarding function cannot be realized. Another application method is: Although the serial port 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 PACKAGE AND HEARTBEAT PACKAGE

The registration package and the heartbeat package are a function suitable for communication between the device and the cloud software.

8.1. REGISTRATION PACKAGE

The definition of the registration package is that when the computer software and the serial 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 at various collection and monitoring points. How to make the cloud software recognize the module is very important, and it is necessary to realize the communication of the Internet of Things.

The serial server provides the following multiple registration package methods.

8.1.1. SEND MAC ADDRESS ON CONNECTION

Send the MAC address on the connection: This method is not only supported for 4 models (such as RS485 TO ETH (B)), but also for ordinary models. 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 uniquely identify the 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".

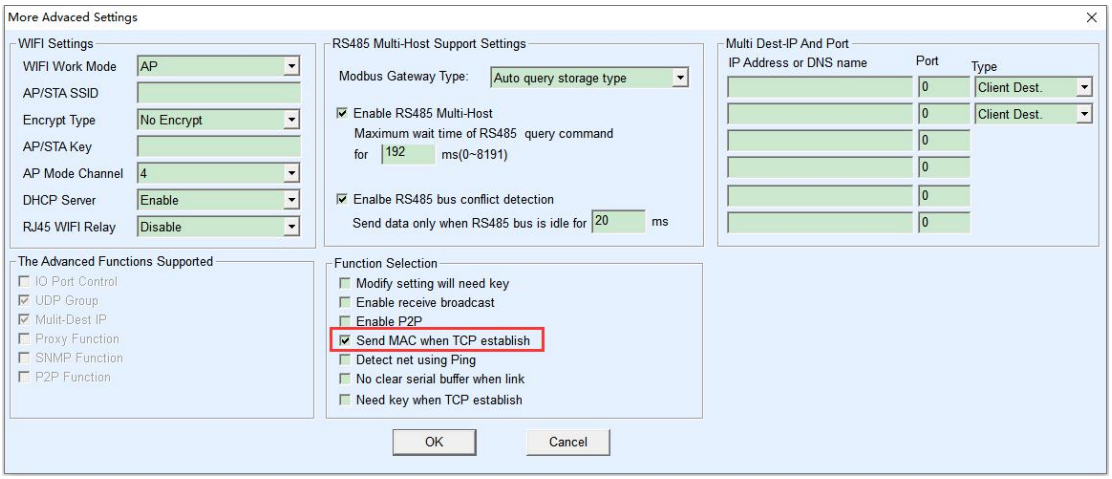


Figure 31 Sending MAC address on connection

8.1.2. REALCOM AGREEMENT

The Realcom protocol is a mature 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.

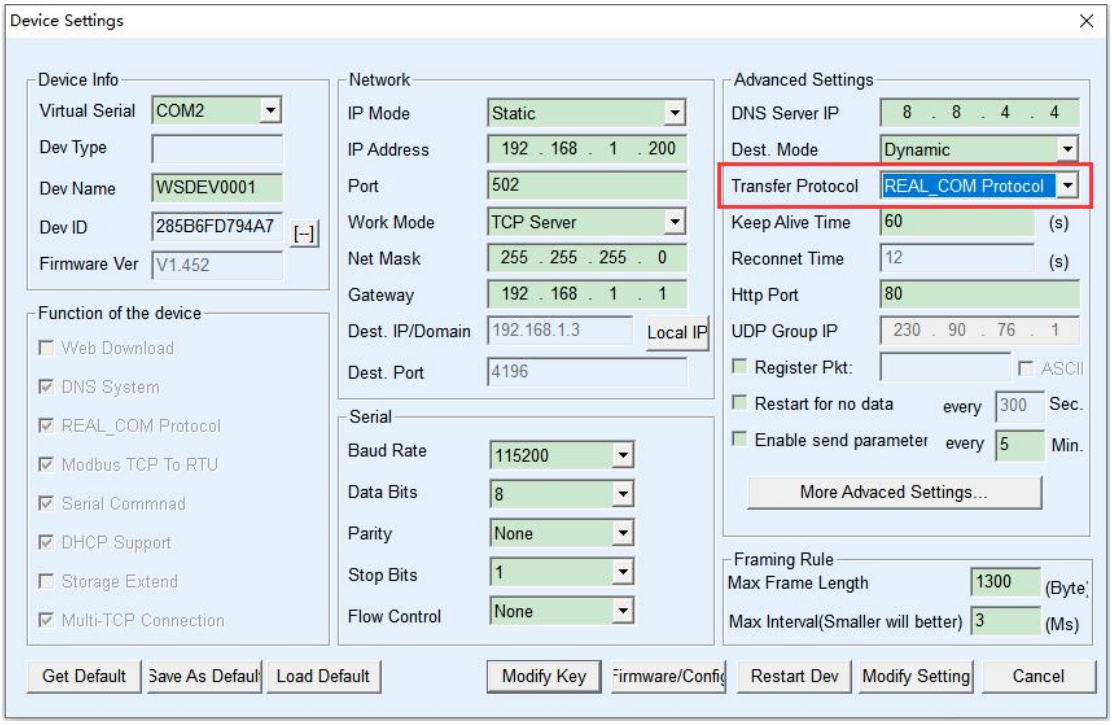


Figure 32 Enable realcom protocol

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 time, 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 is that the user can fill in an arbitrary registration package format. The method is: in the device setting interface, the configuration is as follows:

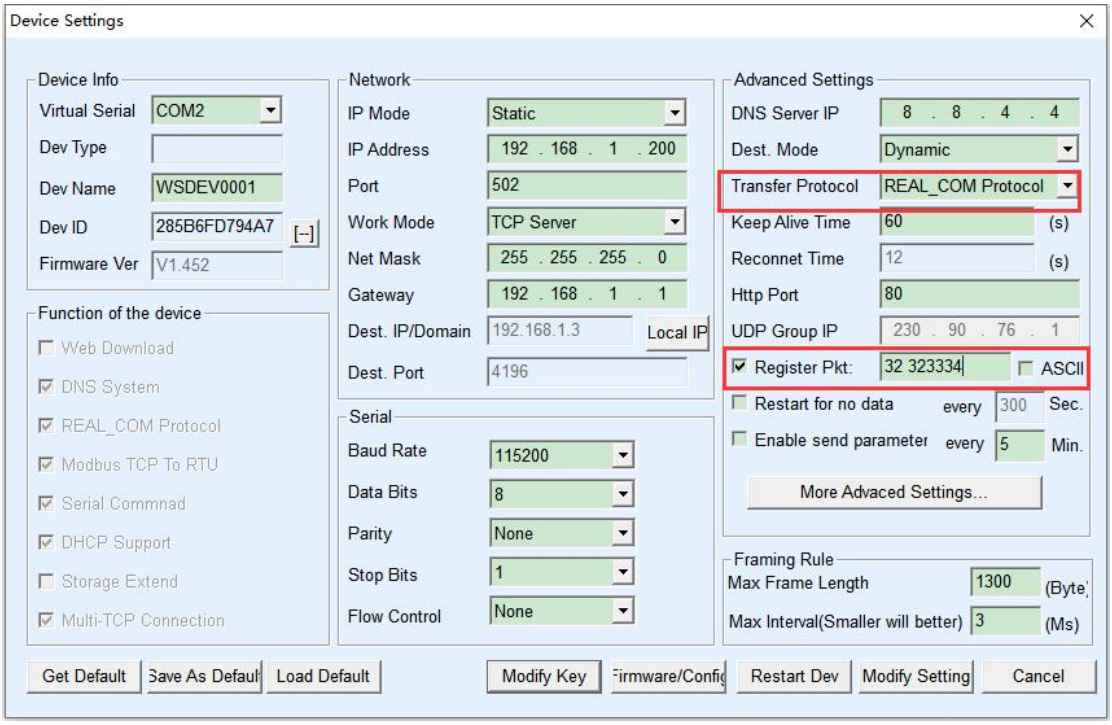


Figure 33 Set registration package

The difference with 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, that is, the actual data sent is the string 1234. If you need to display a string, click the "ASCII" option next to it.

When the device is connected to the cloud software, it can automatically send the hexadecimal registration package of 31 32 33 34. This registration package method is more flexible, allowing 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 separately for each device, 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 MACs.

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

8.1.4. CONFIGURATION FILE

Support serial server to write a configuration file, so as to realize user-defined registration package, and can use MAC address wildcard, which can solve the trouble of writing custom registration package for each device, and there is no limit to the length of registration package.

8.2. HEARTBEAT PACKAGE

The heartbeat packet is mainly used to detect whether the communication link is disconnected. The method of implementation 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; secondly, if the device fails to send the heartbeat, the device at the TCP client will automatically re-establish the TCP connection, so it is a recovery network communication A means.

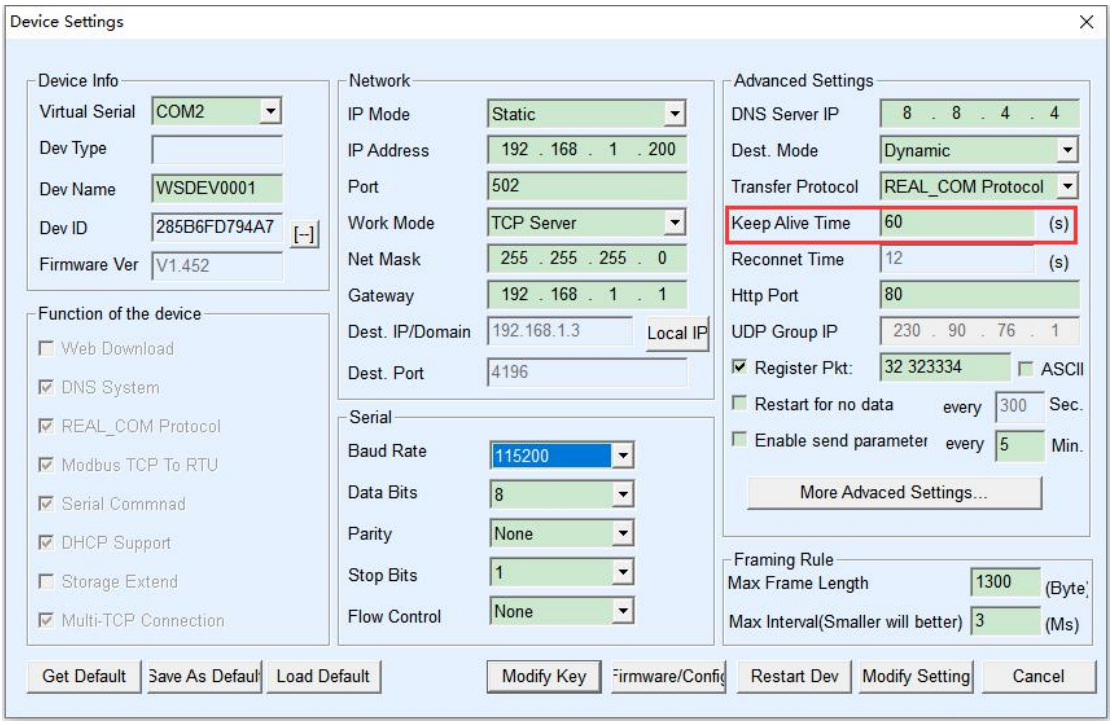


Figure 34 Keep Alive Time

As shown in Figure 34, 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, when the Micro Snow device is on the TCP client side, the implicit heartbeat function will be enabled. 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

Whether the function 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 timing time. This data is the heartbeat packet of realcom protocol.

8.2.3. CUSTOM HEARTBEAT PACKAGE

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".

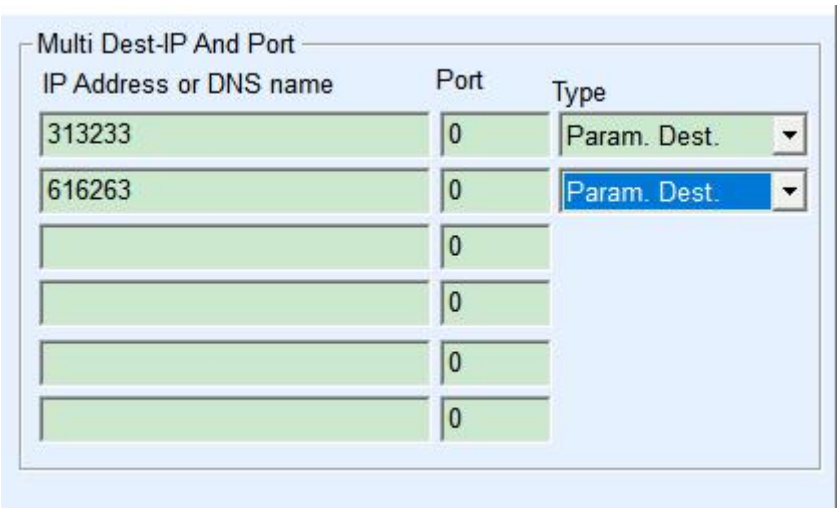


Figure 35 Customized registration package

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 keepalive 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.

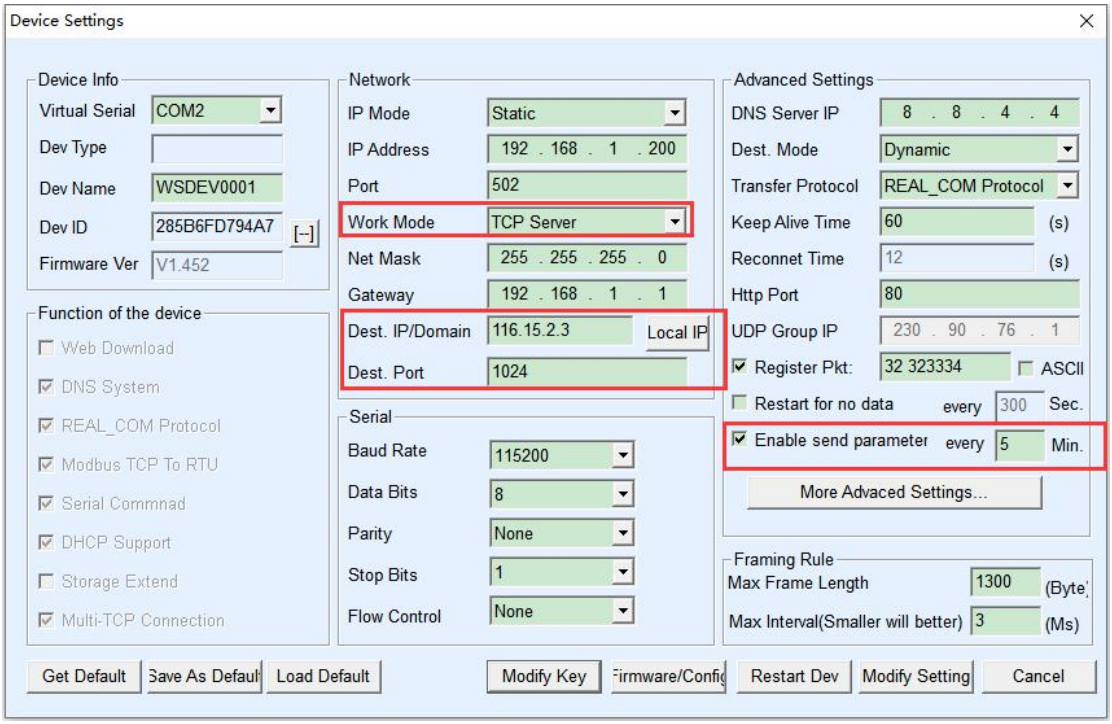


Figure 39 Timing sending parameters

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 Manual

11. FIRMWARE UPGRADE METHOD

RS485 TO ETH (B) can upgrade their respective programs, but not mutually upgrade programs. Whether it is automatic search, manual addition or P2P search and other methods found in the device list, you can use this method to upgrade the firmware.

1 Obtain the firmware file of RS485 TO ETH (B) from Waveshare.

2 In the Vircom tool, first search for the device that needs to be upgraded, and then enter the device parameter editing dialog box. First click once on "Restart Dev".

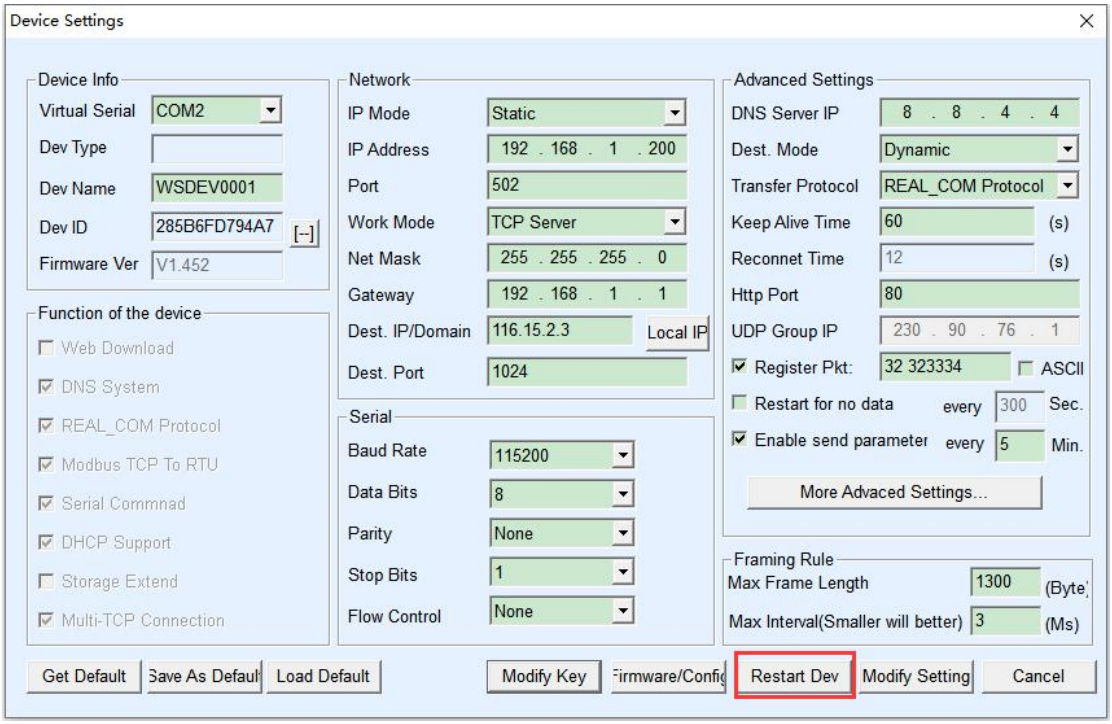


Figure 40 Restart the device

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.

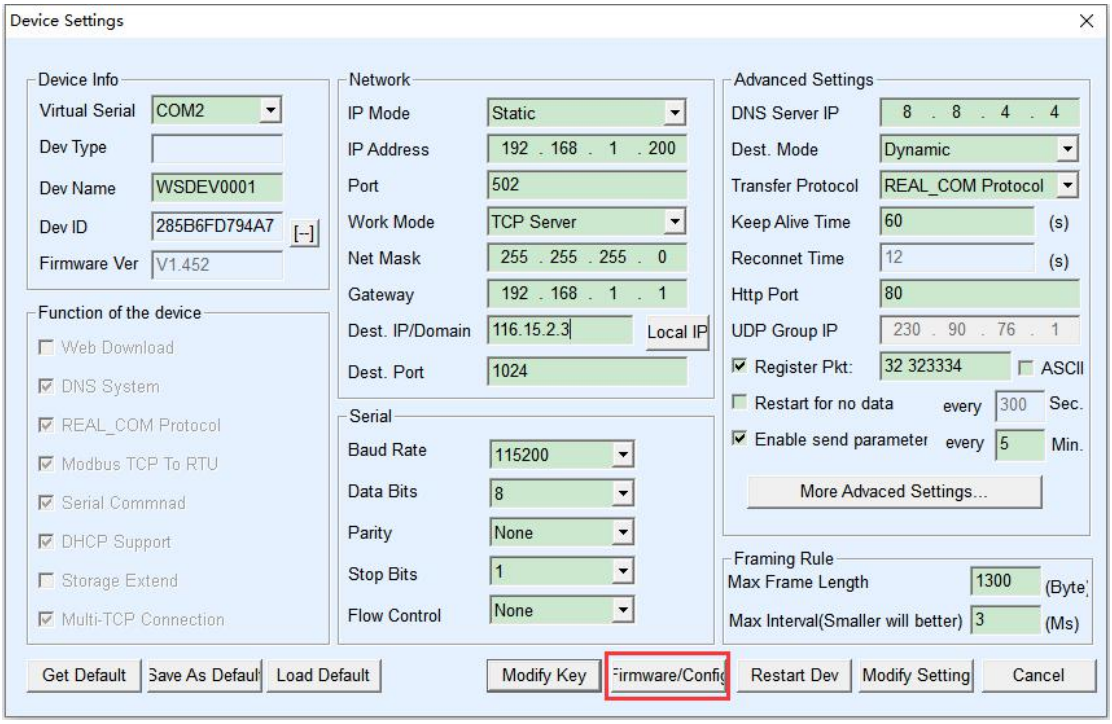


Figure 41 Upgrade firmware

3 As shown in Figure 42, 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.

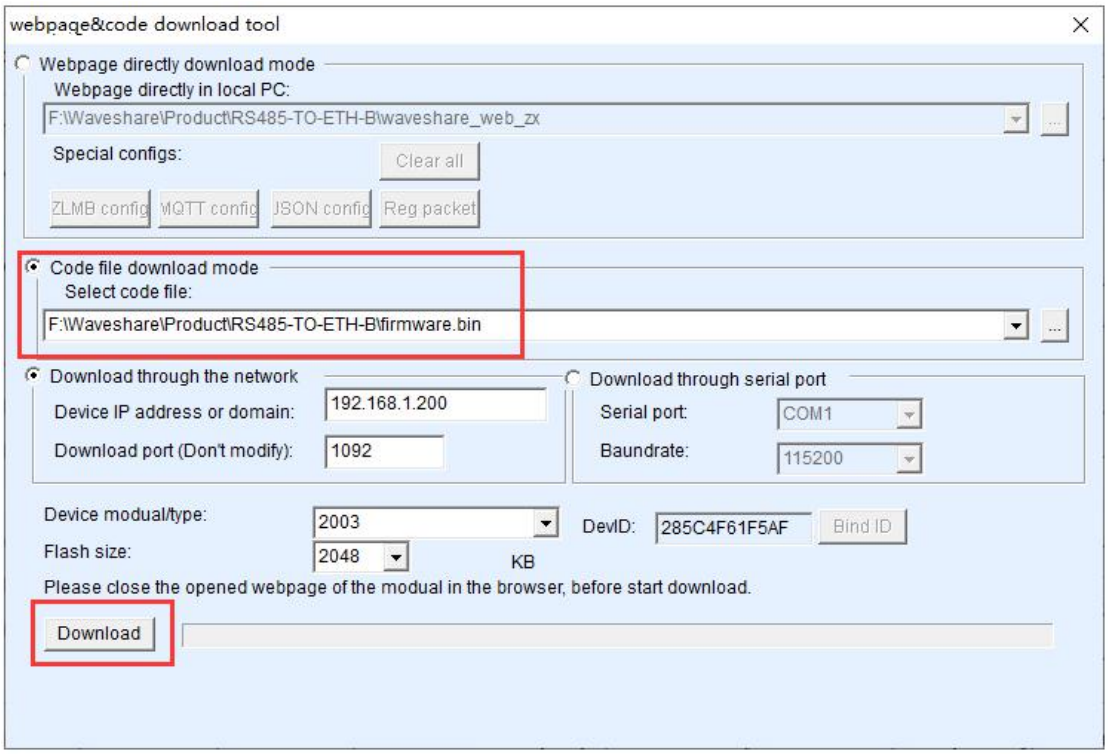


Figure 42 RS485 TO ETH (B) firmware upgrade method

4 At this time, 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 webpage 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 43, change the "program file" download mode to "web directory download". And select the root directory where the local webpage is located as the directory where the webpage file to be downloaded is located (the directory can be obtained from Weixue), click download, and download all the files in the local webpage directory to the internal file system of the device.

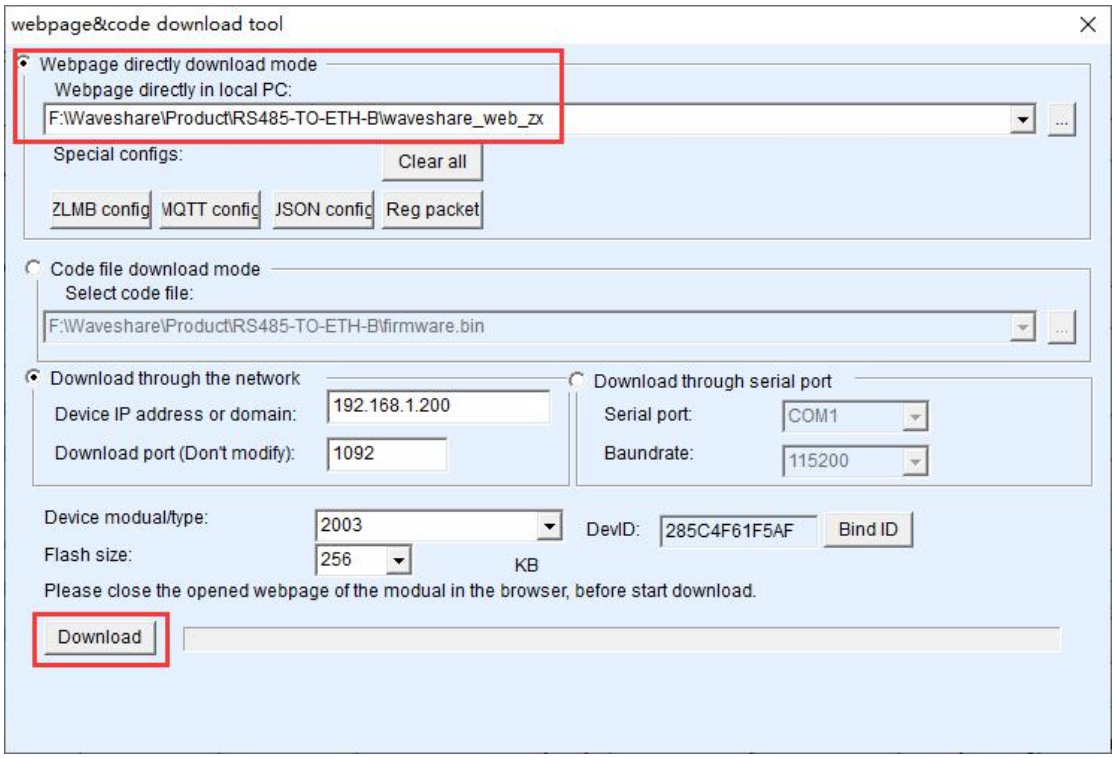


Figure 43 RS485 TO ETH (B) web upgrade method

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 through Vircom, you can know whether the new firmware has been downloaded successfully.

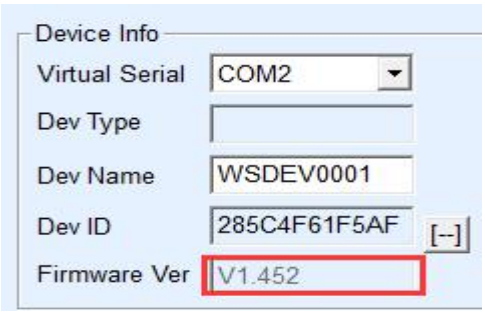


Figure 50 Check the firmware version after the level is completed