RS485_TO_ETH_Rail

Designed for Raspberry Pi Development Platform

Product Manual

Rev 1.0

2024/5/11

Seengreat Technology Co.,Ltd

Table of Contents

I、Description

1.1 Product Overview

RS485_TO_ETH_RAIL is an RS485 device data collector/Internet of Things (IoT) gateway specifically designed for industrial environments, which integrates multiple functions such as serial server, Modbus gateway, MQTT gateway, and RS485 to JSON conversion. It features one RS485 interface and one Ethernet interface. It adopts a rail-type mounting method, is compact in size, and easy to install. It uses a terminal block type power connection with a wide voltage input of 9-24V. Since it only has an RS485 interface and has eliminated the RS232 interface, which is less commonly used on industrial instruments, it saves on cost and volume, offering a higher cost-performance ratio compared to ordinary products.

Figure 1 RS485_TO_ETH

When used as a regular serial server, connect the Ethernet port of RS485_TO_ETH to the Ethernet network. The upper computer software can use TCP/UDP transparent transmission mode or virtual serial port mode to send data to

the RS485 interface of RS485_TO_ETH; data received on the RS485 interface will also be transmitted to the software on the TCP side or the virtual serial port. In virtual serial port mode, the serial port software connected to the virtual serial port can be used without any configuration.

When used as a Modbus gateway, RS485_TO_ETH supports the conversion from Modbus TCP to Modbus RTU, enabling the upper computer to collect data from Modbus RTU devices on RS485 using the Modbus TCP protocol. Conversely, if the RS485 end acts as a master station, it can also be used.

RS485_TO_ETH supports more advanced Modbus gateway functions, including configurable Modbus gateway ZLMB, storage-type Modbus gateway, etc., fully meeting the various configurations and usage methods of Modbus gateways. Its multi-host function allows multiple computer master stations to access a single RS485 slave device simultaneously.

When used as an MQTT gateway, the device can upload serial port data to the MQTT server in a Transparent transmission manner using the MQTT protocol, supporting servers such as Baidu Cloud MQTT, Alibaba Cloud MQTT, China Mobile OneNet platform, etc. It supports parsing collected Modbus RTU or non-standard serial port data into JSON format encapsulated in MQTT data packets for uploading.

It supports uploading collected data in JSON format, with automatic data collection. The collected data supports Modbus RTU, 645 instrument version 97, 645 instrument version 07, and various non-standard RS485 protocols. Users can independently configure the uploaded data format and JSON keywords using Vircom. When uploading, it supports MQTT protocol, HTTP POST protocol, HTTP GET protocol, transparent transmission protocol, and various non-standard network protocols.

The RS485_TO_ETHN model supports P2P networking, eliminating the need for server relay, and connects the device through an ID.

RS485_TO_ETH_RAIL has a powerful range of IoT gateway functions and is

highly suitable for collecting various RS485 instruments and sensors in the industrial field, including local network collection or autonomous collection and uploading to cloud servers.

RS485_TO_ETH_RAIL can be applied to:

1. Acting as an IoT gateway for communication between devices and the cloud.
2. Power, intelligent instrument and energy consumption monitoring.
3. Remote monitoring and program downloading for various automation PLC.
4. Communication interfaces for various configuration software and devices.
5. Networking of access control and security equipment.

The typical application connection is shown in Figure 2. The existing serial port device RS485 is connected to the RS485 port of RS485_TO_ETH_RAIL, and RS485_TO_ETH is connected to the computer via an Ethernet cable. The software on the computer establishes a connection with RS485_TO_ETH using either TCP/IP or a virtual serial port method. Thereafter, any data sent by the serial port device will be transparently transmitted to the computer's software, and data sent by the software over the network to RS485_TO_ETH_RAIL will also be transparently transmitted to the serial port device.

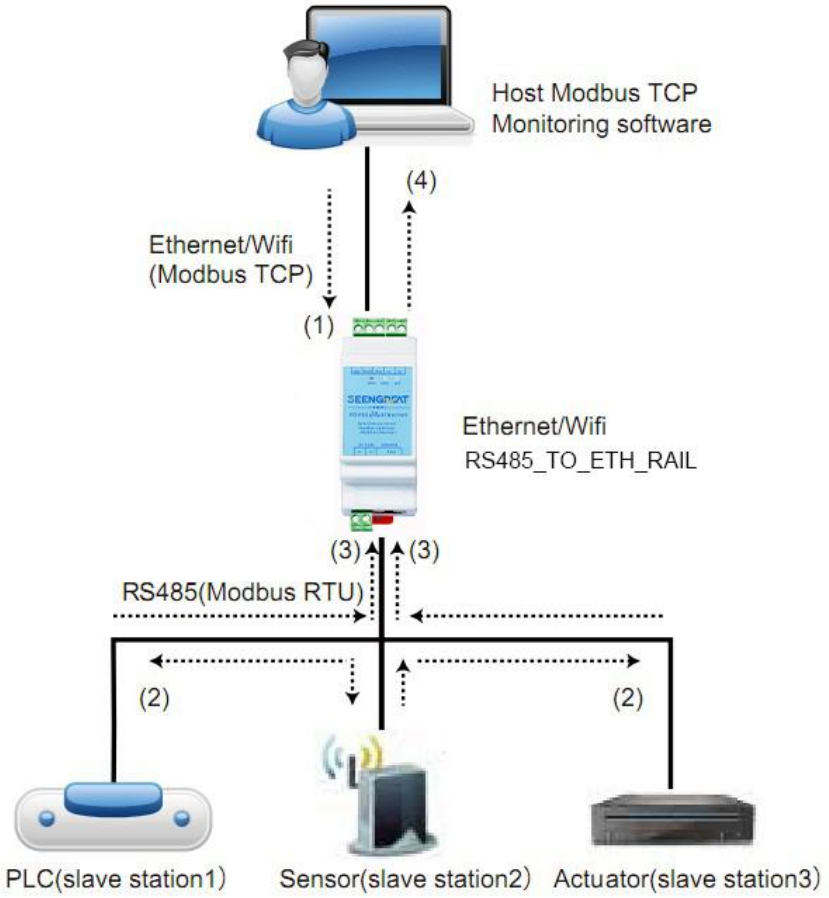

Figure 2 Connection Diagram Illustration

1.2 Product Features

1.2.1 Hardware Features

RS485_TO_ETH_RAIL has the following features:

1. Rail-type design: Suitable for installation inside industrial cabinets.

2. Compact size: Compared to ordinary serial port servers, it has a smaller width and takes up less space.

3. Terminal-type power supply, wide voltage input of 9~24V, with reverse connection and reverse power protection.

4. Terminal-type RS485 interface, supports up to 32 slave devices, with baud rates ranging from 300 to 115200bps.

5. Rich panel indicator lights for easy debugging: In terms of connectivity, there are indicator lights for indicating whether the network cable is connected (LINK), as well as indicator lights for indicating TCP connection establishment (LINK); in terms of data transmission, there are independent indicator lights for "serial port to network port" and "network port to serial port".

1.2.2 Software Features

1. Supports TCP server, TCP client, UDP mode, and UDP multicast. When functioning as a TCP client, it also supports TCP server functionality. As a TCP server, it supports up to 30 TCP connections, and as a TCP client, it supports connections to 7 destination IP.

2. Baud rate supports 1200 to 115200 bps, data bits range from 5 to 9 bits, and supports five types of parity: None, Odd, Even, Mark, Space. CTS/RTS hardware flow control and XON/XOFF soft flow control.

3. Supports sending MAC addresses upon device connection to facilitate cloud-based device management.

4. Provides a secondary development package DLL development library for searching and configuring devices on the computer.

5. Supports configuration via a web browser, dynamic IP acquisition through DHCP, and connection to domain name servers via the DNS protocol.

6. Supports cloud-based remote device search, configuration of device parameters, and device firmware upgrades.

7. Supports remote viewing of device TCP connection status, serial data sending and receiving status via software. The virtual serial port supports data monitoring functionality.

1.2.3 Advanced Software Features

1. Supports Modbus gateway functionality, including Modbus RTU to Modbus TCP conversion. It can handle both storing and non-storing types of Modbus, automatically collecting and storing device data when required. It also supports ZLMB configurable table Modbus gateway functionality.

2. Supports multi-host functionality: Under a query-response mode, it allows multiple computers to simultaneously access a single serial port device via the network port.

3. Supports MQTT gateway functionality.

4. Supports JSON to Modbus RTU and 645 instrument protocols, and enables data upload via HTTP POST and HTTP GET formats.

5. Supports obtaining network time via the NTP protocol, for use in serial port output or protocol content transmission.

6. Supports custom heartbeat and registration packet functionality, facilitating communication with the cloud and device identification.

7. Supports password authentication for establishing TCP connections, ensuring connection security.

8. Support "transcoding" function, can achieve specific device protocol translation work, so that different devices docking unified software platform.

9. Supports HTTP-based data submission and issuance functionality, allowing the cloud to interact directly with the device's serial port data using HTTP GET commands.

1.3 Product Parameters

Table 1 Technical Specifications

Physical Characteristics			
Communication Mechanical Interface	Terminal block, Ethernet port		
Power Mechanical Interface	Terminal block		
Maximum Forming Size	L × W × H = 10.6 × 3.6 × 5.9 (unit: cm)		
Communication Interface			
Ethernet	RJ45 interface, 2KV surge protection		
Serial Port	RS485 × 1: RXD, TXD, GND		
Serial Port Parameters			
Baud Rate	300~115200bps	Parity	None, Odd/Even Parity, Mark, Space
Data Bits	5~9 bits	Flow Control	No flow control
Software			
Protocols	ETHERNET、IP、TCP、UDP、HTTP、ARP、ICMP、DHCP、DNS		
Configuration Methods	VirCOM tool, web browser, device management function library		
Communication Modes	TCP/IP direct communication, virtual serial port mode		
Operating Modes			
TCP server, TCP client (TCP server and client coexist simultaneously), UDP, UDP			

multicast	
Power Requirements	
Power Voltage	9~24V DC
Environmental Requirements	
Operating Temperature	-40~85℃
Storage Temperature	-45~165℃
Temperature Range	5~95%

II、Usage

2.1 Hardware Interface

2.1.1 Hardware Interface Configuration

Figure 3: Front View

The front view of the RS485_TO_ETH_RAIL serial server is as shown in the figure above, with a 35mm standard rail enclosure.

1. Power input: Terminal block: 5.08mm terminal. Input voltage 9~24V.
2. RS485: Use RS485 to connect to 485B and 485A. 485B represents the negative line of 485, and 485A represents the positive line of 485. 485 can support up to 32 loads. The maximum communication distance is 1200 meters. It is generally necessary to use terminal resistors for RS485 lines longer than 300 meters. The terminal resistance of 485 is 120 ohms.
3. Ethernet port: Connect with a network cable, supports auto crossover.

4. Indicators: Divided into Power (PWR), Link (LNK), and Active (ACT) lights, representing power, connection status, and data transmission respectively.
5. Reset: Short-circuit the two NC pins shown in the figure above for more than 5 seconds, and the module will reset to the IP address 192.168.1.254, and various downloaded configuration files will not be loaded.

Table 2: Meaning of Indicators

Power Light	Indicates power status.
Link light	（1）When the network cable is properly connected, the Link light is green. （2）After TCP connection is established (or in UDP mode), the Link light is blue (with a faint green light). This can be used to determine whether the serial server has established a communication link with the host software.
Active light	（1） When data is sent from the Ethernet port to the serial port, the indicator light is green. The shining time is 1 second longer than the actual communication time, making it easier to detect short data communications. （2）When data is sent from the serial port to the Ethernet port, the indicator light is both blue and green. Since blue is brighter, seeing blue indicates that data is being returned from the serial port to the Ethernet port. This can determine whether the device has responded to commands from the host software. If there is no response, it indicates that either the serial port baud rate is incorrect or the serial port is not properly connected.

Troubleshooting Communication Using Indicator Lights:

1. If the Link light is not green, the network cable is not properly connected. Please

check the network cable.

2. If the Link light is not blue (only considering TCP operation mode), the host software has not established a connection with the serial server. Consider whether the IP addresses are configured in the same network segment.
3. If the Active light is green, it indicates that data is being sent from the Ethernet port, but there is no response from the serial port device. Please check if the baud rate is properly configured and if the RS485 polarity is reversed.
4. Installation Method: The device enclosure has a 35mm standard mounting rail. In situations where there is a mounting rail, the device can be directly mounted on the rail.

Figure 4 Device Rear View

2.1.2 Hardware Interface Connection

Generally, for a serial port server, it only needs to be connected to power, serial port, and Ethernet cable. The power supply can use a two-wire power supply available on-site, and 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. Connect 485 positive to 485A and 485 negative to 485B. The Ethernet port is connected with a regular Ethernet cable, which can be directly connected to a computer or connected to the network via a switch.

2.2 Software Interface

2.2.1 Software Installation

Vircom can be used to configure the device IP and other parameters, also can create virtual serial port.

If you don't need virtual serial port function, just need device management and configuration, please run vicom.exe directly.

If you need the virtual serial port function, instal "virtual-serial-port-control3.5 .msi", install it following the default prompts, and then run vircom.exe.

2.2.2 Parameter Configuration

After installing Vircom and completing the hardware connection of the device, run the Vircom software as shown in the figure, and then click on "Device" as shown in the figure. With 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 on the same switch, it can be used.

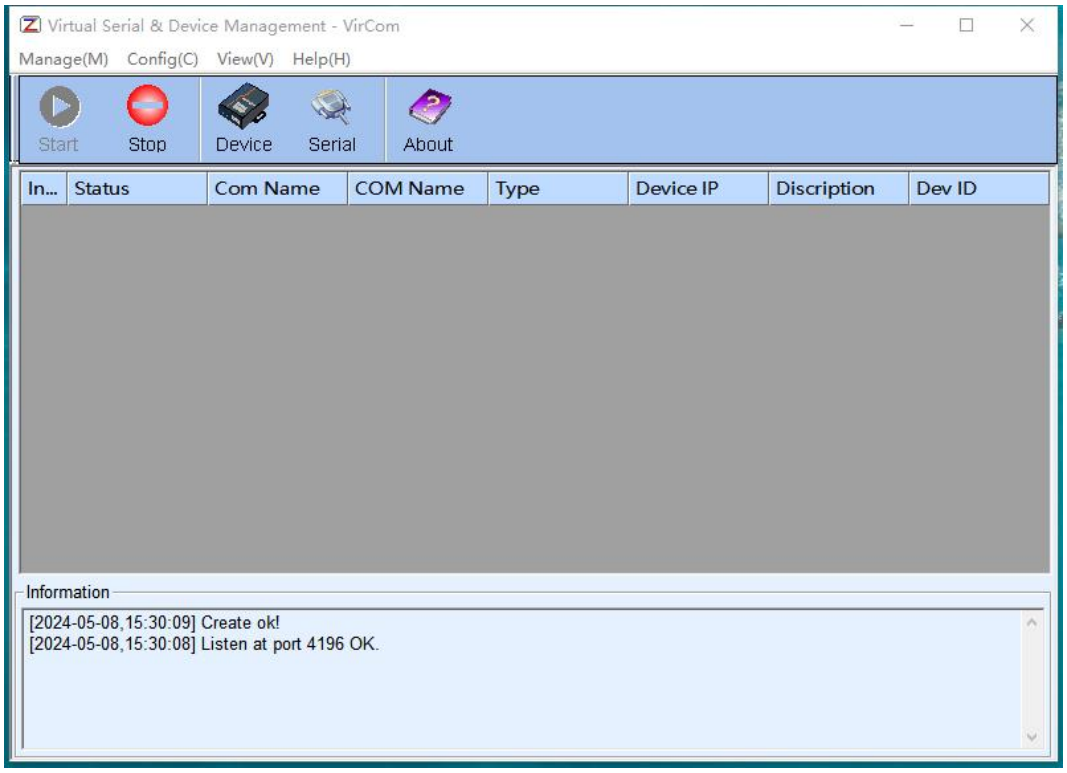

Figure 5: Vircom Main Interface.

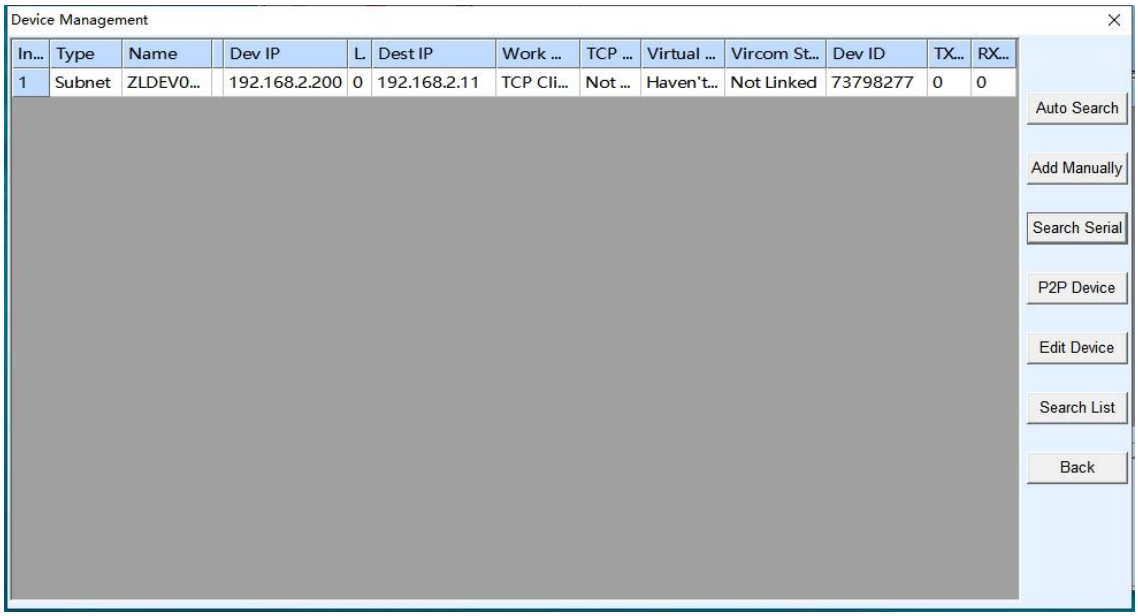

Figure 6: Device List.

From the device list, you can see all currently online devices.Double-click the device to be configured, or click the device to be configured and then click "Edit Device" to configure parameters.

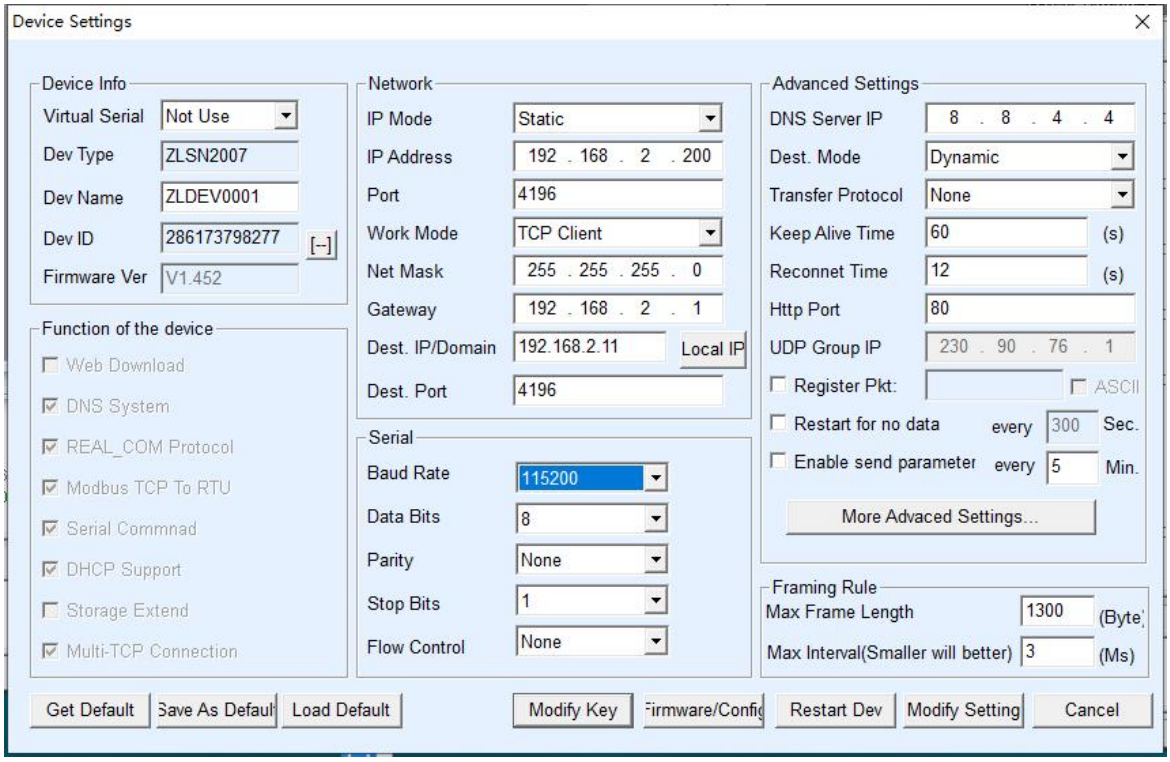

Figure 7: Device Parameters

In this interface, users can set the parameters of the device. Clicking 'Modify Settings' will save the parameters to the device's flash memory, ensuring they are not lost during power loss. The device will also automatically restart.

The main parameters to configure here include: baud rate, data bits, and parity in the serial port settings; IP address, subnet mask, and gateway in the network settings. Depending on the computer software, sometimes it's necessary to configure the operating mode of the serial port server.

The detailed meanings of other parameters are as follows:

Table 4: Parameter Meanings

Parameter Name	Value Range	Meaning
Virtual Serial Port	Not Used, Created Virtual Serial Ports	Can bind the current device to a previously created virtual serial port. Please add COM port in "Serial Port Management" on the main interface first.

Device Model		Displays only the model of the core module.
Device Name	Any	Can give the device a readable name, up to 9 bytes long, supports Chinese characters.
Device ID		Factory unique ID, cannot be modified.
Firmware Version		Firmware version of the core module.
Functions Supported by the Device		Refer to Table 5 for the functions supported by the device.
IP Mode	Static, DHCP	Users can choose static or DHCP (dynamic IP acquisition).
IP Address		IP address of the serial port server.
Port	0~65535	When the serial port server is in TCP Server or UDP mode, it listens on the specified port. When used as a client, it's preferable to specify the port as 0 because it enhances connection speed. In this case, the system will allocate a random local port. The difference between using port 0 and a non-zero port is as follows: (1) When the local port is set to 0, upon module or PC restart, a new TCP connection is established. However, the old TCP connection may remain open, potentially leading to multiple false connections with the device. Generally,

		the upper computer (PC) prefers to close the old connections upon module restart. Specifying a non-zero port will close the old connections. (2)When the local port is set to 0, TCP reconnects faster. When the serial port server operates as a TCP client, it simultaneously acts as a TCP server listening for connections on a port. Additionally, 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 Mode, the serial port server waits for computer connections; when set to TCP Client Mode, the serial port server actively initiates connections to the network servers specified by the destination IP.
Subnet Mask	For example: 255.255.255.0	It 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. If it is not crossing outer network (such as the cable connecting computer), it is best to set the gateway as the IP address of the connected computer.
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	Serial port baud rate.
Data Bits	5、6、7、8、9	
Parity	None, Even, Odd, Mark, Space	
Stop Bits	1、2	
Flow Control	No Flow Control, Hardware Flow Control (CTS/RTS), Hardware Flow Control (DTR/DCR), Software Flow Control (XON/XOFF)	Only valid for RS232 serial ports
DNS Server		When the destination IP is described by a domain name, this DNS server IP needs to be filled in. When the IP mode is DHCP, you do not need to specify the DNS server, as it will be automatically obtained from the DHCP server.

Destination Mode	Static, Dynamic	UDP working mode: if the destination computer is described by a domain name, it's best to choose the static mode; If there are multiple computers in the LAN communicating with networking products through UDP, it is best to choose dynamic mode. TCP server mode: this parameter must be dynamic. In TCP client mode: (1)when IP mode is static, the device will automatically restart if it fails to connect to the server for 5 consecutive times. (2)when IP mode is dynamic, the destination IP is reconnected after the device is restarted, so that the correct IP address can be obtained again. Otherwise, it will do direct connection without automatically restarting the device.
Protocol Conversion	NONE、Modbus TCP<->RTU、Real_COM	NONE: Indicates that the data forwarding from the serial port to the network is transparent. Modbus TCP<->RTU: Converts Modbus TCP protocol directly to RTU protocol, facilitating compatibility with Modbus TCP protocol. Real_COM: Designed to be compatible with older versions of the REAL_COM

		protocol. It is a protocol of virtual serial port mode, but selecting RealCom protocol is not necessarily required when using virtual serial ports.
Keep-Alive Timer	0～255	Heartbeat interval. (1)When set to 1~255, if the device is in TCP client mode, it will automatically send a TCP heartbeat every "Keep-Alive Timer" interval. This ensures the TCP link's effectiveness. When set to 0, there will be no TCP heartbeat. (2)When set to 0~254 and the protocol conversion is set to REAL_COM protocol, the device will send a data packet of length 1 containing 0 every "Keep-Alive Timer" interval, implementing 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 works in TCP client mode, the device will send device parameters to the destination computer every "Keep-Alive Timer" interval. When set to 255, there will be no parameter sending function, enabling remote device management.
Reconnect Time After	0～255	When in TCP client mode, if the connection is unsuccessful, the device will

Disconnection.		attempt to reconnect to the computer every "Reconnect Time After Disconnection" interval. This interval can be set from 0 to 254 seconds. If set to 255, it means the device will never attempt to reconnect. It's important to note that the first TCP connection, such as when the hardware is powered on, when restarting the device via Vircom software, or when there is no data light, typically occurs immediately. The device will only wait for the "Reconnect Time After Disconnection" interval to retry after the first connection attempt fails. Therefore, the "Reconnect Time After Disconnection" setting will not affect the normal connection establishment time of the network and the server.
Web Access Port	1～65535	Default is 80
Multicast Group Address		Used in UDP multicast
Enable Registration Packet		When a TCP connection is established, send this registration packet to the computer. After enabling the registration packet, the realcom protocol must be selected. Supports TCP server and TCP client modes.
Packet Length	1～1400	Serial Port Framing Rules 1:After the

		serial port server's serial port receives data of this length, the received data is sent as one frame onto the network.
Packet Interval	0～255	Serial Port Framing Rule 2: When there is a pause in the data received by the serial port server's serial port, and the duration of the pause exceeds this specified time, the received data is sent as one frame onto the network.

Explanation of Supported Device Features:

Table 5: Supported Device Features

Name	Description
Domain Name System	Destination IP addresses can be specified using domain names (e.g., addresses starting with "www").
REAL_COM Protocol	The REAL_COM Protocol is a non-transparent protocol for serial port servers, suitable for binding virtual serial ports of multiple serial port servers via the Internet. It includes the device MAC address internally, aiding device recognition by upper-level equipment. In general, it can be omitted unless specifically required.
Modbus TCP to RTU	It enables the conversion from Modbus TCP to RTU and also supports multi-master functionality.
Serial Port Parameter Modification	This feature supports configuring and reading device parameters using serial port AT commands.
Automatic IP Acquisition	Supports DHCP client protocol

Storage Extension (EX) Functionality	Extended later
Multiple TCP Connections	Supports more than one TCP connection when acting as a TCP server.
IO Port Control	Supports arbitrary custom instructions to control 8 IO outputs.
UDP Multicast	UDP Multicast
Multiple Destination IP	When operating as a TCP client, it supports simultaneous connections to 7 destination IPs.
Proxy Server	Supports proxy server functionality (requires specific model).

2.3 Communication Testing

2.3.1 TCP Communication Testing

After configuring the device parameters, you can use serial port tools or TCP debugging tools for TCP connection communication testing.

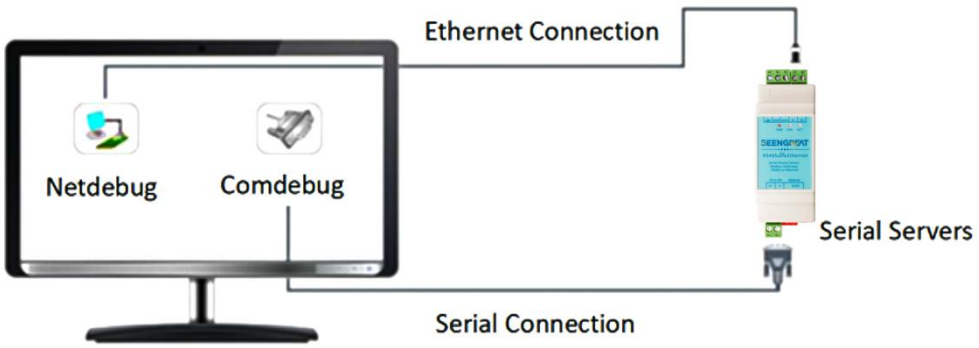

图 8 TCP 通讯示意图

As shown in Figure 6, connect the PC to the RS485_TO_ETH_Rail via an Ethernet cable, and then connect the serial server to the computer by using a USB-to-RS485 cable.

Disable the firewall and antivirus software on the computer. Set the computer's IP

address to static, ensuring it is in the same network segment as the serial port server and does not conflict with the server's IP address. Currently, the server's IP address is 192.168.2.200 (reference the parameter configuration content in section 2.2.2 for device IP settings). You can set the computer's IP address to 192.168.2.88, as shown in Figure 9.

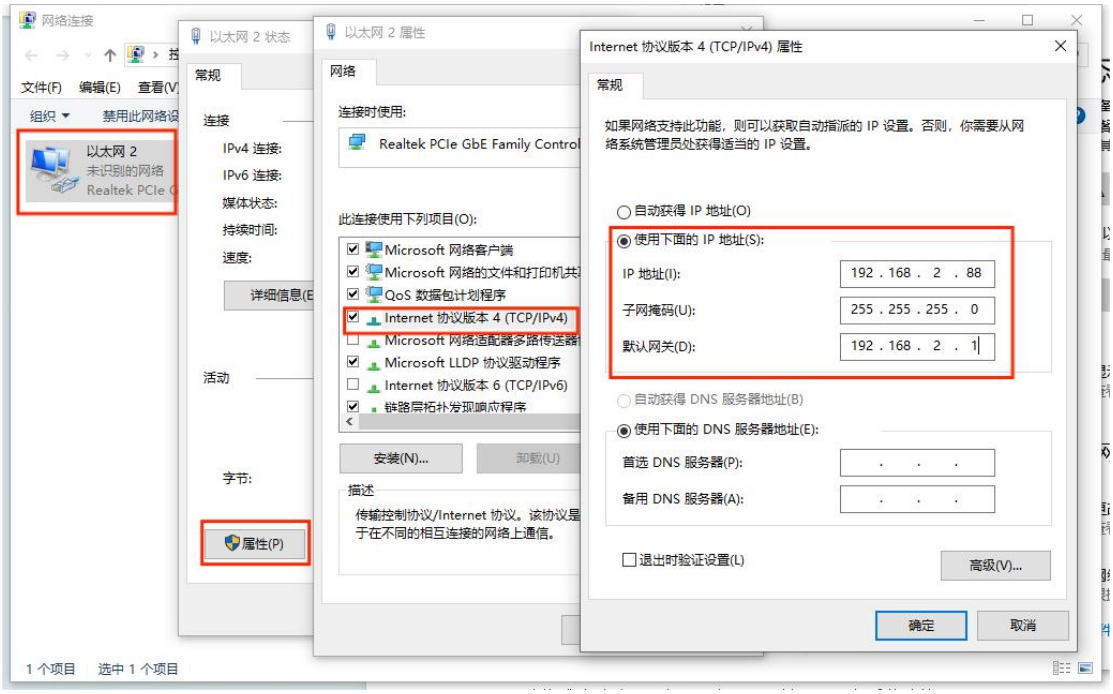

Figure 9: Schematic Diagram of Modifying Computer IP Address

After completing the setup, you can use the Comdebug serial debugging assistant software and the Netdebug network debugging assistant software to perform TCP communication testing on this product.

Configure the Netdebug software, which will be used for TCP debugging, as a TCP client. Enter the IP address and port number of the serial port server (previously configured using the Vircom software) for connection. Also, use the serial port software to open the corresponding serial port.

Once the connection is successful, both sides will be able to achieve TCP to 485 bidirectional transparent transmission functionality.

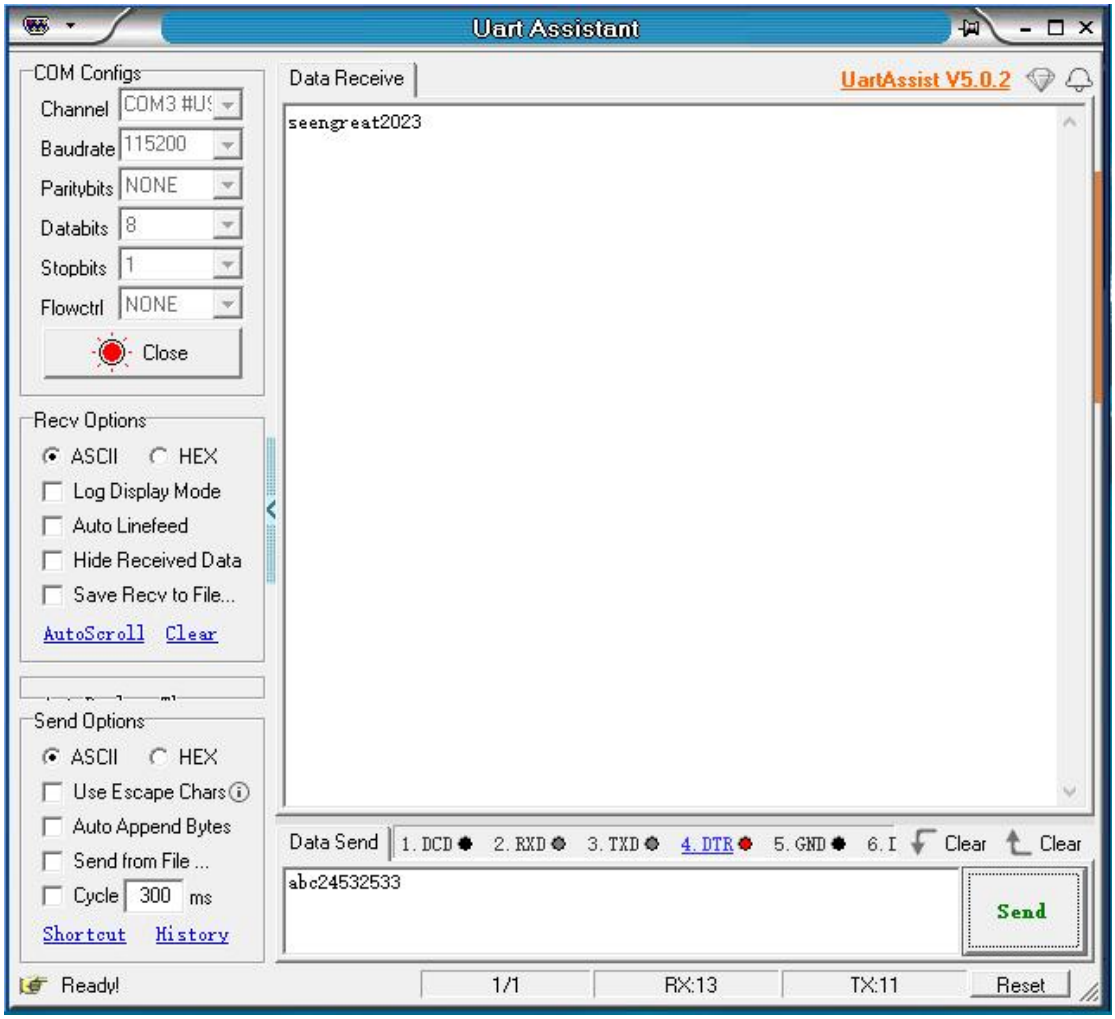

Figure 10: Comdebug Serial Debugging Interface

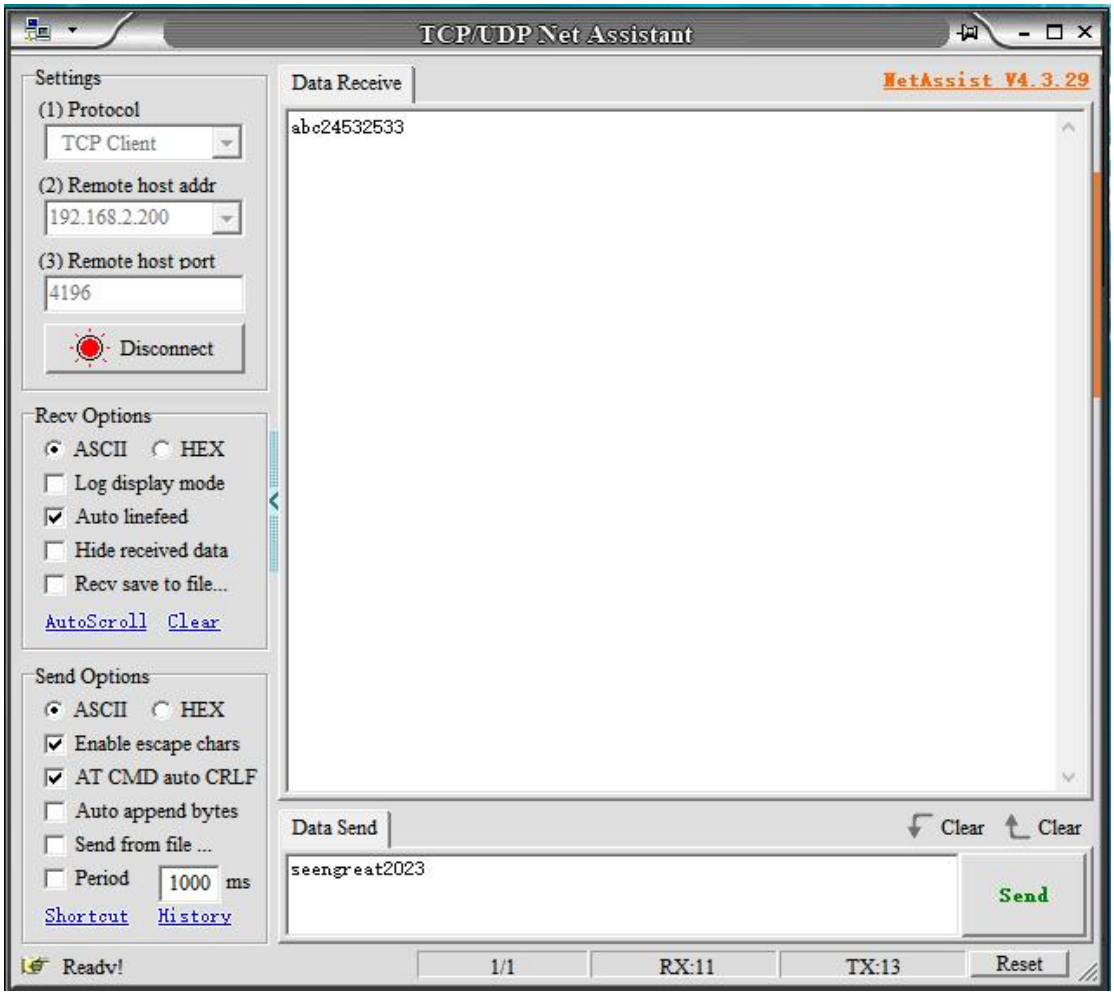

Figure 11: Netdebug TCP Debugging Interface

2.3.2 Virtual Serial Port Testing

If your host computer software only supports serial communication, and you want to communicate with the lower-level device through this product, you can use Vircom to create a virtual serial port. The communication principle and structure are shown in Figure 12.

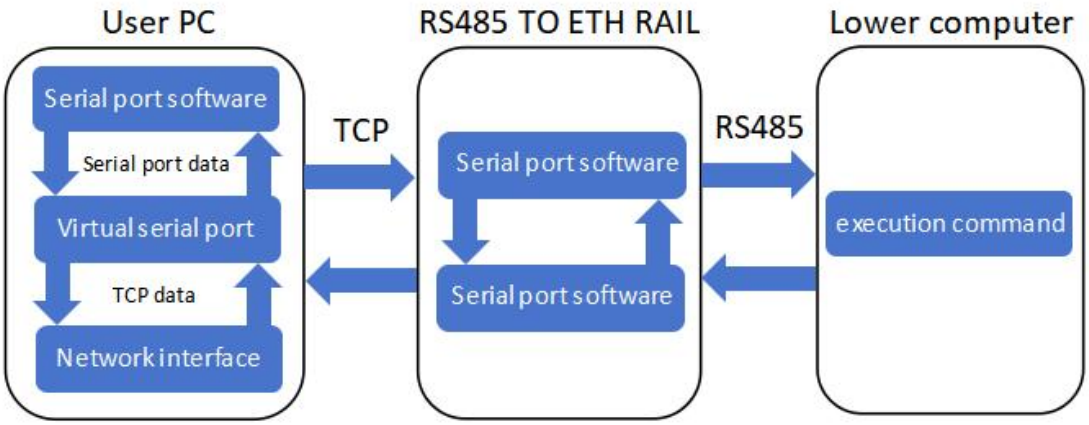

Figure 12: Communication Principle

Click on "Serial" in Vircom, then select "Add". From the COM Number dropdown menu, choose "COM2" (COM2 being a serial port number that doesn't already exist on the computer). Finally, click "OK", as shown in Figure 13 below.

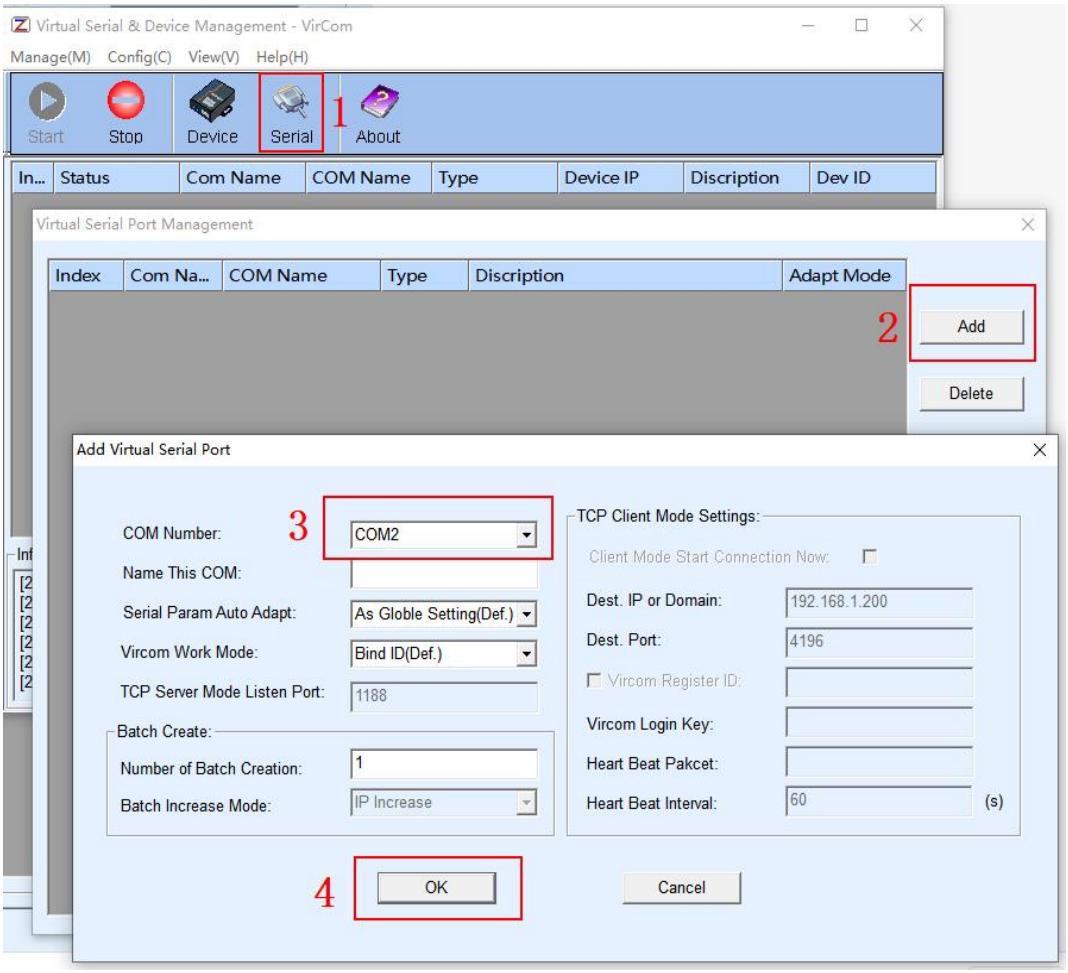

Figure 13

Return to the main interface of Vircom, then click on "Device". Double-click on the device listed (such as "2" in Figure 14). In the "Virtual Serial" section at the top left, select COM2. Next, click on "Modify Setting" in the bottom right corner. You'll see a screen like the one shown in Figure 15, indicating that COM2 is now connected to a device with the IP address 192.168.2.200.

If the connection is not established, please check if the IP address of the serial port server and the computer are in the same network segment. If not, modify the serial port server to be in the same network segment following the parameter settings mentioned earlier in section 2.2.2.

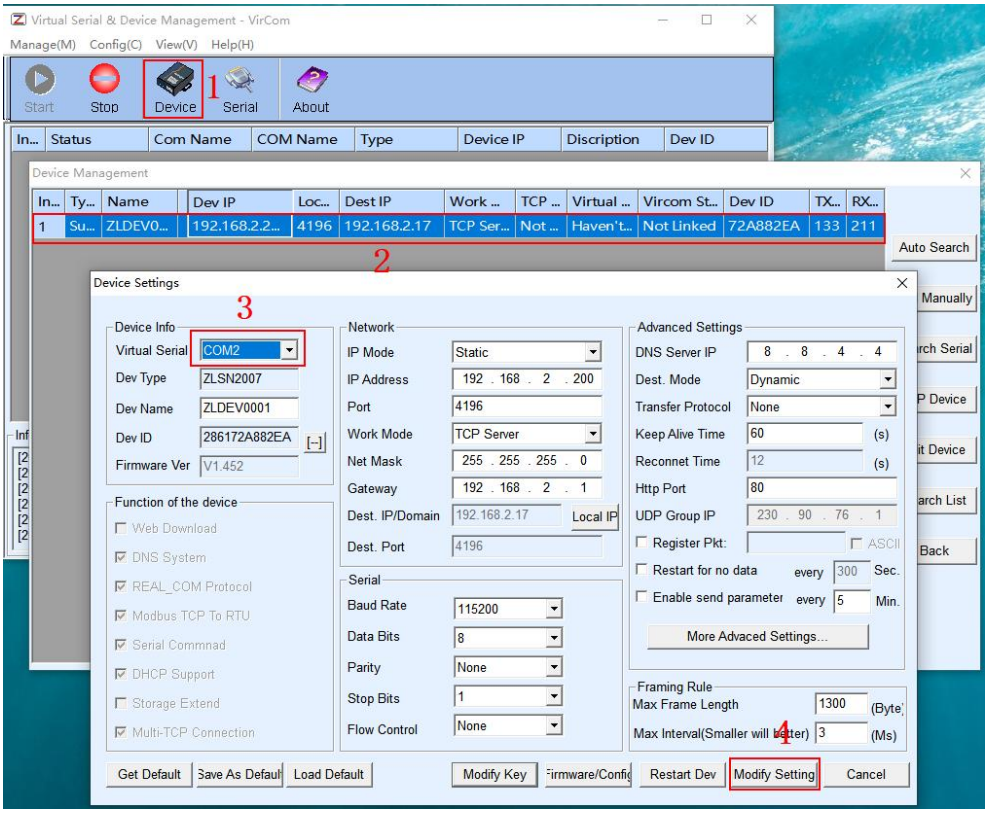

Figure 14

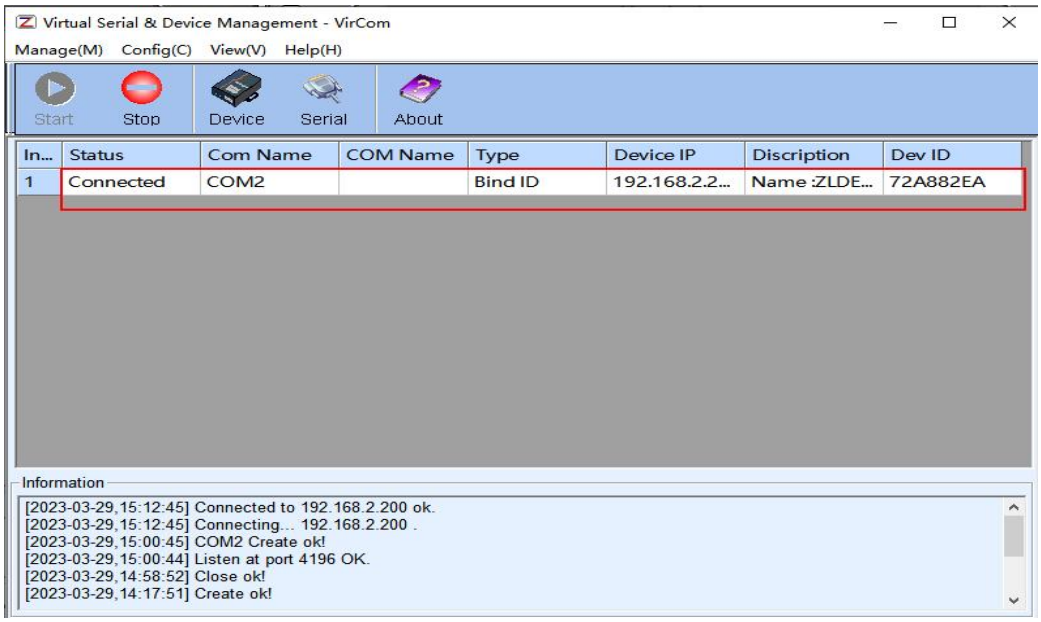

Figure 15

Enable two serial port debugging assistant software Comdebug at the same time. In one Comdebug instance, select COM3 (choose according to the actual situation; this port is connected to the RS485 end). In the other Comdebug instance, select COM2 configured according to the steps above. Then, sequentially open the serial ports to conduct communication testing, as shown in Figure 16.

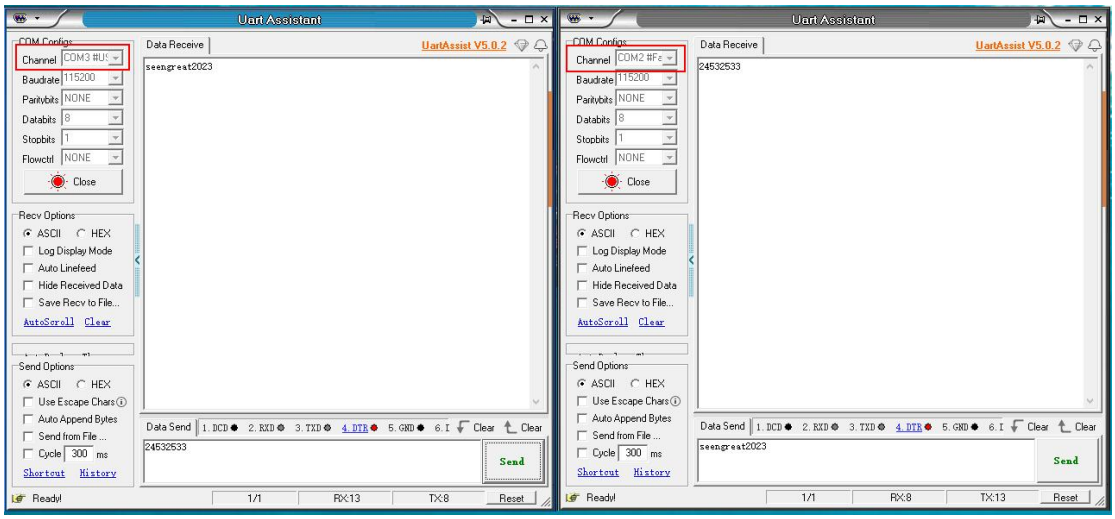

Figure 16

2.3.3 ModbusTCP Test

By default, serial and network data are transparently transmitted. If you need to implement Modbus TCP to RTU conversion, you need to select "Modbus_TCP Protocol" in the device settings dialog box, as shown in Figure 17. At this time, the device port automatically becomes 502. When the user's Modbus TCP tool connects to port 502 of the serial port server's IP, the Modbus TCP command sent will be converted into an RTU command and output from the serial port. For example, if the serial port server's network port receives a Modbus TCP command of 00 00 00 00 00 06 01 03 00 00 0a, then the serial port will output the command 01 03 00 00 00 0a c5 cd. Note: The serial port may send multiple 01 03 00 00 00 0a c5 cd commands because the default Modbus uses a storage method and will automatically poll query commands.It'll show you how to switch to non-storage.

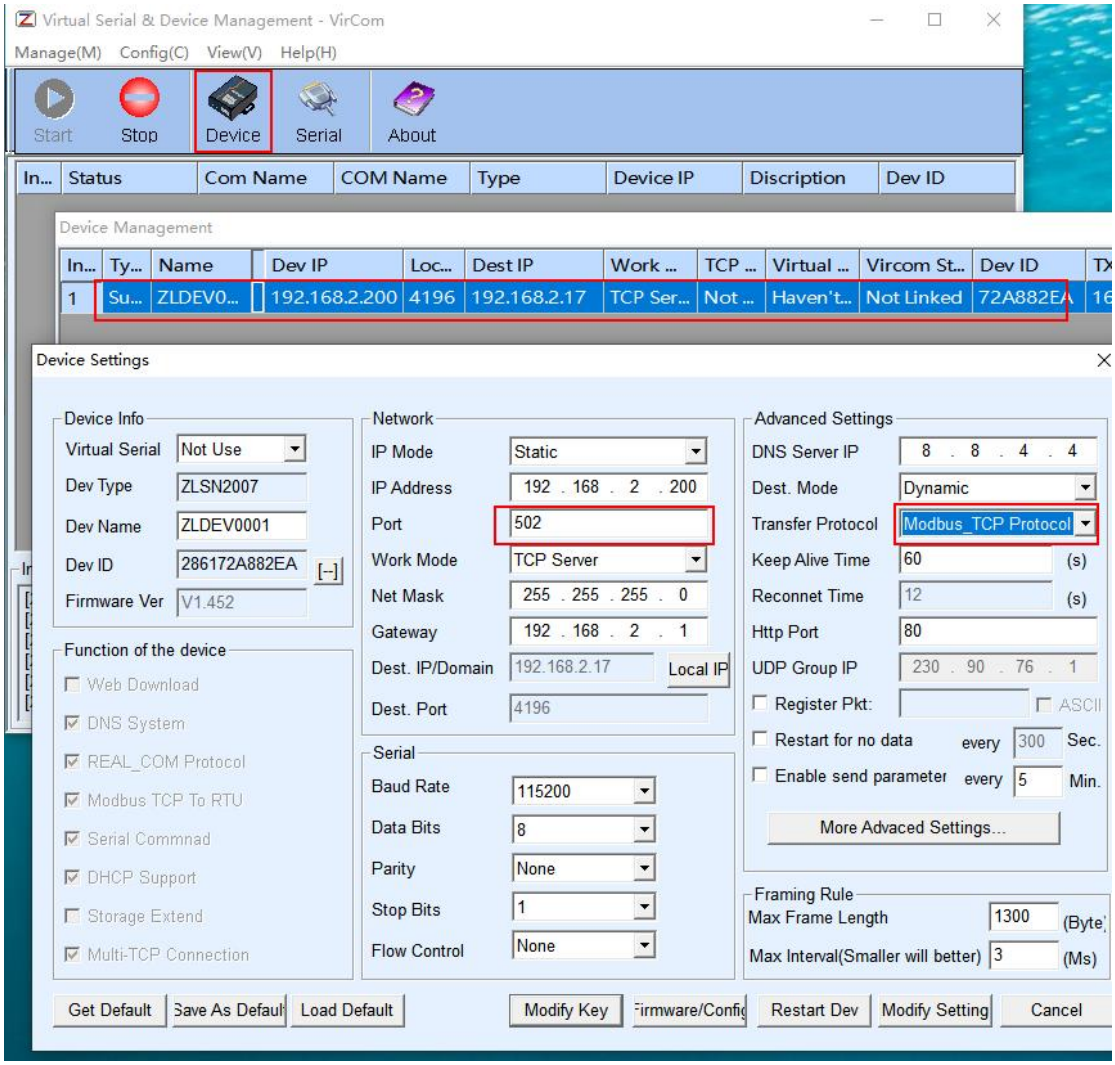

Figure 17 Enable Modbus TCP Functionality

If the user's Modbus TCP software is acting as a slave, then in addition to selecting the conversion protocol, the operating mode should be changed to client. The destination IP should be set to the IP address of the computer where the Modbus TCP software is located, and the destination port should be set to 502, as shown in Figure 18.

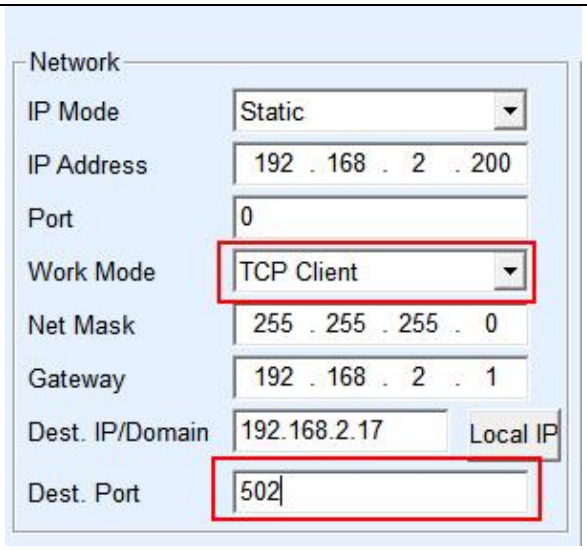

Figure 18 Modbus TCP Operating as Client

2.3.4 Web-based Configuration

Using Vircom, you can search for and configure device parameters within different network segments. Web-based configuration requires that the computer and the serial port server are in the same IP range, and you need to know the IP address of the serial port server beforehand. However, web configuration can be performed on any computer without Vircom installed.

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

Figure 19

2. Enter the password in the "Password" field: The default is 123456. Click the "login" button to log in.

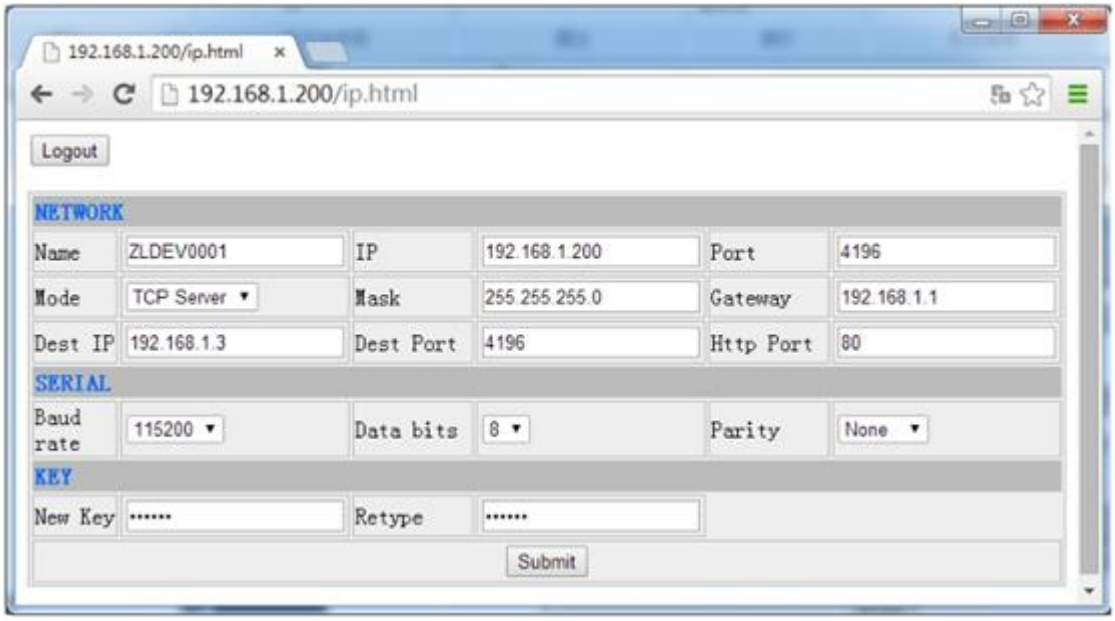

Figure 20 Web Configuration Interface

3. In the webpage that appears, you can modify the parameters of the serial port server. For the meanings of the parameters, please refer to Table 4.

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

2.4 Working Mode and Conversion Protocol

Different applications may require different working modes and conversion protocols for the serial port server, aiming for more stable and reliable usage. The following provides detailed explanations:

There are two basic ways to use a serial port server: with virtual serial ports and without. This is illustrated respectively in Figure 9 TCP Communication Diagram and Figure 12 The Role of Virtual Serial Ports. Using virtual serial ports requires the user software to interface with COM ports, meaning both the user software and devices are serial. Without virtual serial ports, the user software communicates directly via TCP/IP, but the user devices are still serial.

In the case of non-virtual serial ports, the "Conversion Protocol" section offers three options: transparent transmission, Modbus TCP to RTU conversion, and

Realcom protocol. If the user software employs a fixed protocol such as Modbus TCP while the lower-level device uses Modbus RTU, then Modbus TCP to RTU conversion is necessary. The Realcom protocol is currently only applicable when multiple serial port servers act as TCP clients connecting to a server, and the server employs virtual serial ports.

Summary of Usage:

Number	Use of Virtual Serial Port	Device Working Mode	Conversion Protocol	Description
1	Used	TCP Server	None	Suitable for occasions where user software opens COM port to actively collect data.
2	Used	TCP Client	None	Suitable for occasions where devices actively send data. If TCP server is chosen, there might be issues with reconnection after device disconnection.
3	Not used	TCP Server	Modbus TCP to RTU	Suitable for cases where user software uses Modbus TCP and user devices use Modbus RTU, with Modbus TCP acting as main station.
4	Not used	TCP Client	Modbus TCP to RTU	Suitable for cases where user software uses Modbus TCP and user devices use Modbus RTU, with Modbus RTU acting as main station.
5	Used	TCP Client	Realcom Protocol	When multiple serial port servers act as TCP clients and virtual serial ports are used, Realcom protocol is preferred.
6	Not used	TCP Client	None	Suitable for scenarios with numerous devices connecting to a cloud, typically

				an Internet-based server with a public IP.
7	Not used	TCP Server	None	Suitable for scenarios where devices and computers are on the same local network, monitoring locally without the need for Internet communication.

2.4.1 Virtual Serial Port Mode

If the user software communicates via COM ports, then virtual serial port mode is necessary. This includes some PLC software, configuration software, instrument software, etc.

Next, consider whether the monitoring computer and devices are both on the local network:

1. If the computer is a publicly accessible server leased on the Internet, then the devices must use TCP client mode to connect to the server. In this case, options ② and ⑤ from Table 6 can be selected. If multiple serial port servers are used, option ⑤ must be chosen.

2. If both are on the local network (able to ping each other), then consider whether it's the host computer initiating queries or the devices actively sending data. If the devices are actively sending data, they must use TCP client mode, option ②. Otherwise, option ① can be selected.

2.4.2 Direct TCP/IP Communication Mode

If there's no need for Modbus TCP protocol conversion and virtual serial ports, the user software might directly communicate with the Ethernet port of the serial port server via TCP/IP. The serial port server then converts TCP/IP data into serial data and sends it to the serial devices.

In general, users who employ this method develop their own upper computer

network communication software, integrating the parsing of the device's serial communication protocol. This approach is more flexible and efficient compared to virtual serial ports. Corresponding options from Table 6 are ⑥ and ⑦.

The section "4.5 TCP Communication Testing" briefly discusses communication when the serial port server acts as a TCP server. Here, we'll explain how TCP clients, UDP mode, and multiple TCP connections communicate with computer software. SocketTest (a software emulating user TCP/IP communication) is used as an example for computer software.

Seengreat serial port servers adhere to standard TCP/IP protocols, enabling communication with any network terminal that follows these protocols. Seengreat Technology provides communication debugging tools (Seengreat program) to simulate network terminals and communicate with serial port servers.

For two network terminals (in this case, a network debugging tool and a serial port server) to communicate, their parameter configurations must be paired.

2.4.3 TCP client mode.

There are two operating modes under TCP mode: TCP server and TCP client. Regardless of which mode is adopted, one side must be a server and the other side must be a client for communication to take place. If both sides are either clients or servers, communication cannot be established.

When the serial port server acts as a client, there must be three corresponding relationships as shown in Figure 21:

1. Correspondence of working modes: The working mode of the serial port server corresponds to the server mode of the network tool.

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

3. Correspondence of ports: The destination port of the serial port server must be the local port of the network tool.

After these settings, the serial port server can automatically connect to the network tool, and once the connection is established, data can be sent and received.

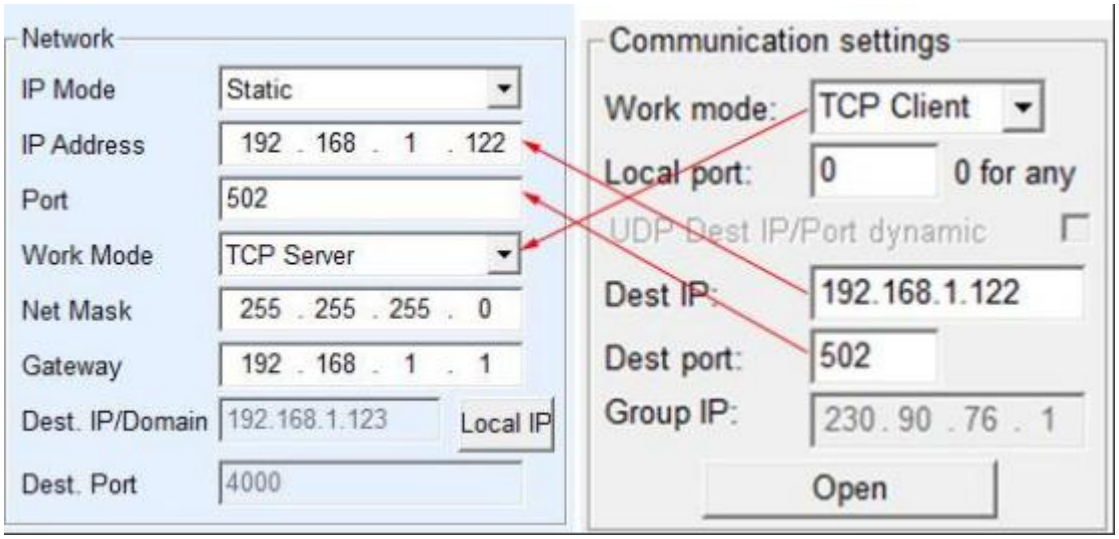

Figure 21 Serial Port Server as Client

When the Seengreat serial port server acts as a TCP client, it can simultaneously connect to 7 destination IP addresses, and the data sent from the serial port will be transmitted to all 7 destination IP simultaneously. If there are not enough servers to fill all the slots, the remaining destination IP will be left vacant.

The usage method is as follows:

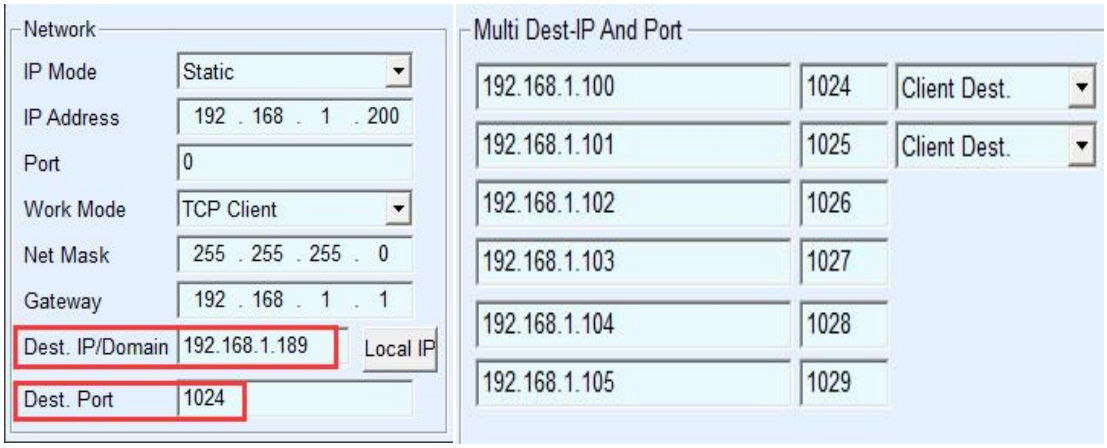

Figure 22 7 IP and ports

The first IP is configured in the device settings interface as shown in Figure 22, where the first IP can be a domain name. The remaining 2 to 7 destination IP can be set by clicking the "More Advanced Options" button in the device settings interface to

open additional advanced options for configuration.

Once all 7 destination IP are set, the device will automatically initiate connections. If a connection cannot be established, it will repeatedly attempt to reconnect after a set "reconnection" interval.

2.4.4 TCP server mode

When the serial port server acts as a server, there are also three corresponding relationships, as shown in Figure 23. Here, I won't go into detail about each one. After configuring in this manner, clicking the "Open" button on the network tool will establish a TCP connection with the serial port server. Once the connection is established, data can be sent and received.

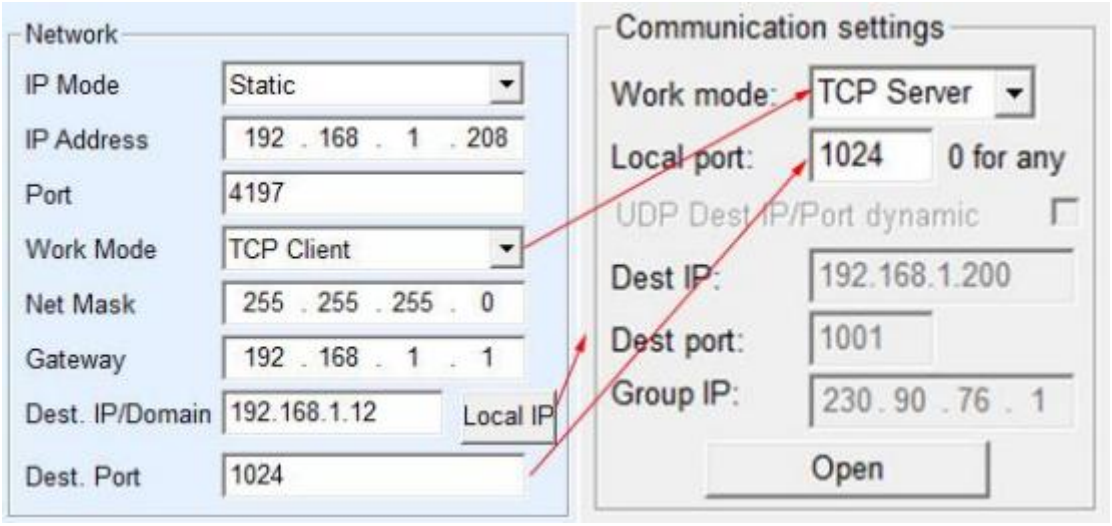

Figure 23 Serial Port Server as Server

When the serial port server acts as a server, it can accept up to 30 TCP connections simultaneously. Data received from the serial port will be forwarded to all established TCP connections. If you need to implement data sending only to the TCP that most recently received a network data packet, you need to enable the multi-host feature. Please refer to the multi-host feature for more information.

2.4.5 Acts as both a client and a server.

The Seengreat Serial Port Server supports accepting TCP connections even when the device is operating as a TCP client, meaning it also functions as a TCP server.

By default, when configuring with Vircom, if the operating mode is changed to "TCP client" mode, the port (i.e., the local port) automatically becomes 0 (0 represents selecting a random available port). To support TCP server mode, the computer software must know the local port of the device, so a specific value needs to be specified. As shown in Figure 24, the computer software can now communicate with the device using port 1024 on 192.168.1.200, while the device also acts as a client connecting to port 1024 on 192.168.1.189.

It's important to note that since the local port 1024 is occupied by the server, when operating as a client, the local port used will be "port + 1," which means the software on 192.168.1.189 will see the incoming port from the device as 1024+1=1025.

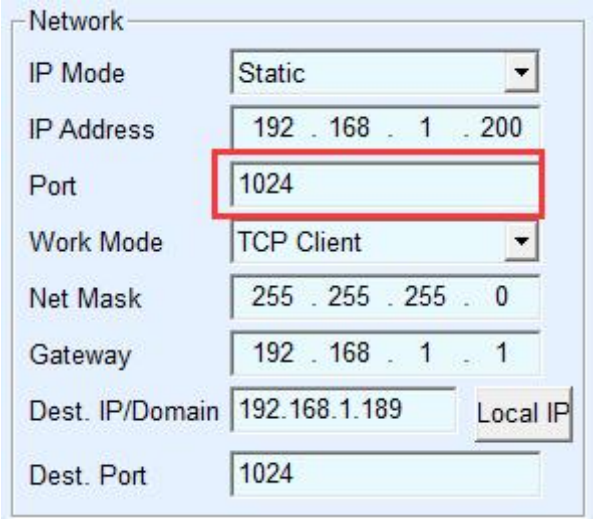

Figure 24: Both acting as a client and a server.

2.4.6 UDP Mode

In UDP mode, the parameter configuration is as shown in Figure 25. On the left side is the configuration of the serial port server in Vircom, and on the right side is the

settings of the SocketDlgTest network debugging tool. Firstly, both must be in UDP working mode. Additionally, as indicated by the red arrows, the destination IP and destination port of the network tool must point to the local IP and local port of the serial port server. As indicated by the blue arrows, the destination IP of the serial port server must be the IP address of the computer where the network tool is located, and the destination port of the serial port server must be the local port of the network debugging tool. Only when these network parameters are configured correctly can bidirectional UDP data communication be ensured.

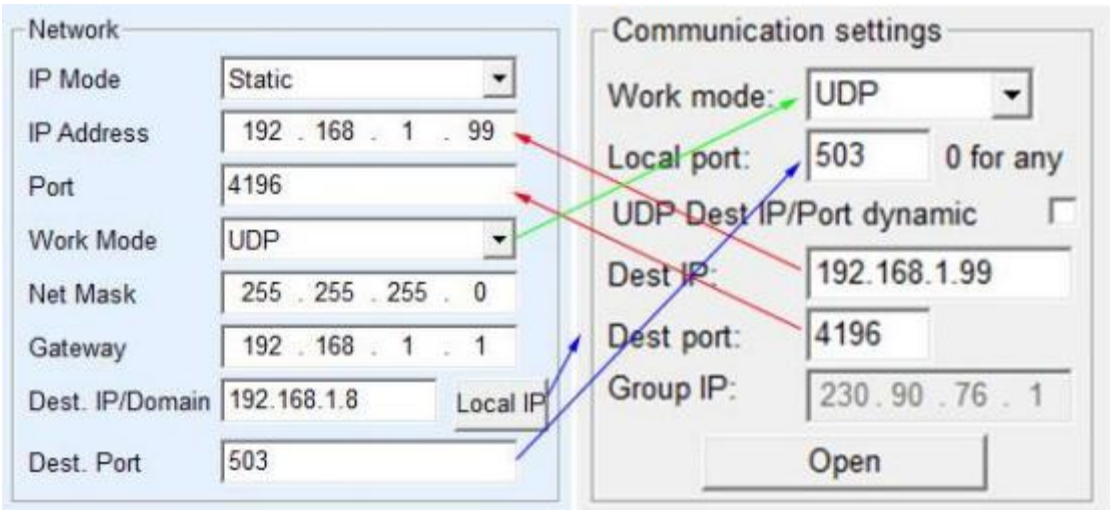

Figure 25: Configuration of Parameters in UDP Mode

2.5 Device Interconnection Method

If the host computer is not a Socket program (SocketDlgTest) or a virtual serial port, but rather two devices connected via Ethernet, the configuration method is similar. Firstly, the user needs to connect the two devices and the computer to the same local area network. The purpose of connecting the computer is only for configuration, and it can be disconnected after configuration is completed.

Open Vircom on this computer, go to device management, find the two devices, and then click "Edit Device" to configure the devices. Device interconnection can be divided into TCP interconnection and UDP interconnection. If it is TCP

interconnection, the parameters of the two devices are as shown in Figure 26. The parameters indicated by arrows must correspond to each other, similar to the corresponding method when connecting to a PC. After the TCP connection is successful, you can check the connection status by returning to the "Device" dialog. If the status of both devices is "Connected," it means that the TCP link between the two devices has been established, as shown in figure 27.

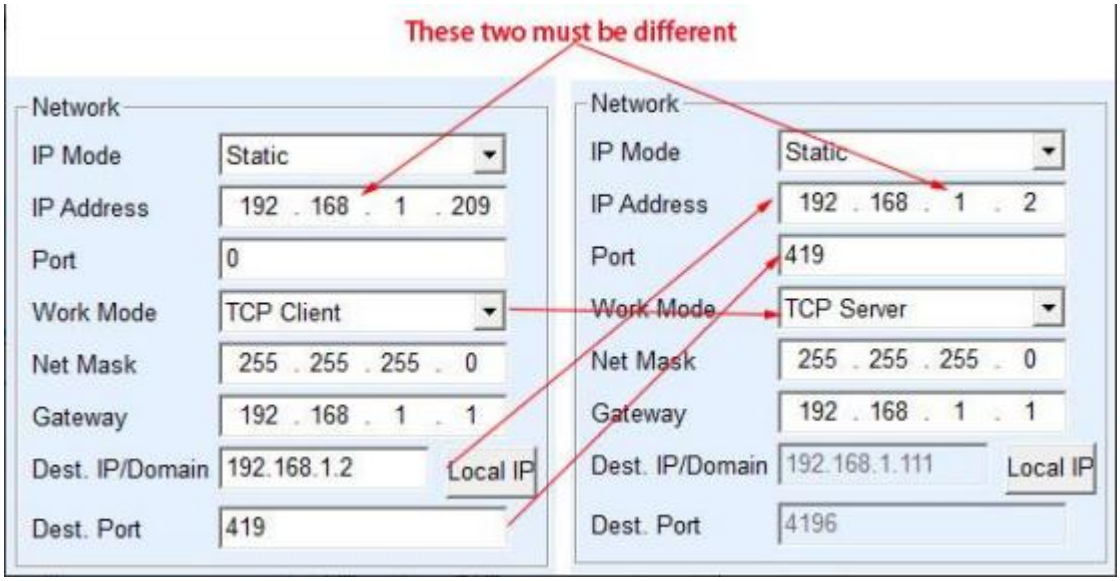

Figure 26: Configuration of Parameters for TCP Device Interconnection

In...	Ty...	Name	I	Dev IP	Dest IP	Work M...	TCP ...	Virtual ...	Vircom St...	Dev ID	TXD	RXD	
1	Su...	LYH		192.168.1.2	192.168.1.3	TCP Ser...	Estab...	Haven't ...	Not Linked	B8AC6D4F	0	0	
2	Su...	LYH		192.168.1.209	192.168.1.2	TCP Clie...	Estab...	Haven't ...	Not Linked	B7F74C2A	0	0	Auto Search

Figure 27: Successful TCP device pairing check

If it is UDP interconnection, the configuration parameters are as shown in Figure 28, and the parameters corresponding to the arrows must correspond one-to-one. With UDP interconnection, as long as the parameters are configured correctly, there is no need to check the connection status. The data will be automatically sent to the specified device.

Finally, it's important to note that for device interconnection, in addition to setting the network parameters as described above, the serial port parameters must also be correctly configured. This mainly involves ensuring that the baud rate and other serial port settings on the serial port server match those of the user's device. With these

settings in place, user devices can send data to each other through the serial ports of the two serial port servers.

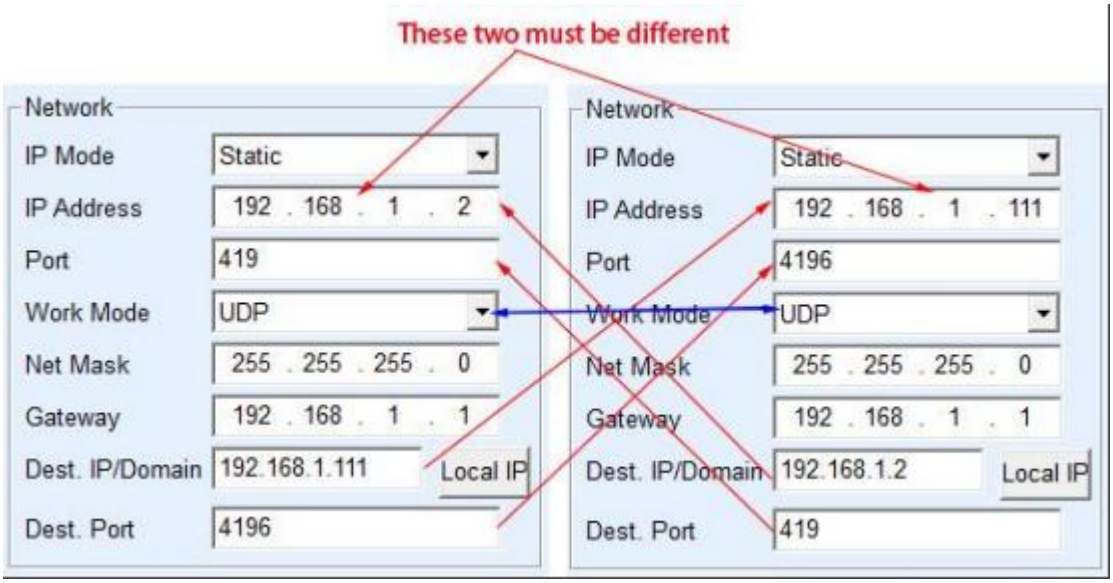

Figure 28: Configuration of Parameters for UDP Device Interconnection

2.6 Modbus Advanced Functionality

Serial port servers with Modbus gateway functionality do not have station addresses or registers themselves; they act as communication bridges. They generate Modbus RTU designation based on Modbus TCP commands sent by user software to the Modbus gateway, including Slave ID, function codes, register numbers, and register quantities, and then output them through the serial port. It can be seen as a protocol "translator."

2.6.1 Enable Modbus Gateway

Firstly, the serial port server should support Modbus gateway, which means that the "Modbus TCP to RTU" function in Table 5 of the device settings dialog should be checked. By default, the serial port server is in normal transparent mode. If you need to switch to Modbus gateway mode, please select the "Modbus_TCP Protocol" option in the "Transfer Protocol" section. Afterward, the device will automatically modify

the "port" parameter to 502 (the port of the Modbus server). This enables the Modbus gateway.

If the serial RTU device acts as a slave, the upper computer Modbus TCP software connects to port 502 of the Modbus gateway. At this time, the Modbus gateway needs to operate in TCP server mode. If the serial RTU acts as a master, the Modbus gateway operates 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.

2.6.2 Storage-type Modbus Gateway

RS485_TO_ETH is a register storage-type Modbus gateway, capable of storing the contents of the read registers internally. This significantly improves the speed of Modbus TCP queries and enhances performance, especially when multiple hosts access it simultaneously.

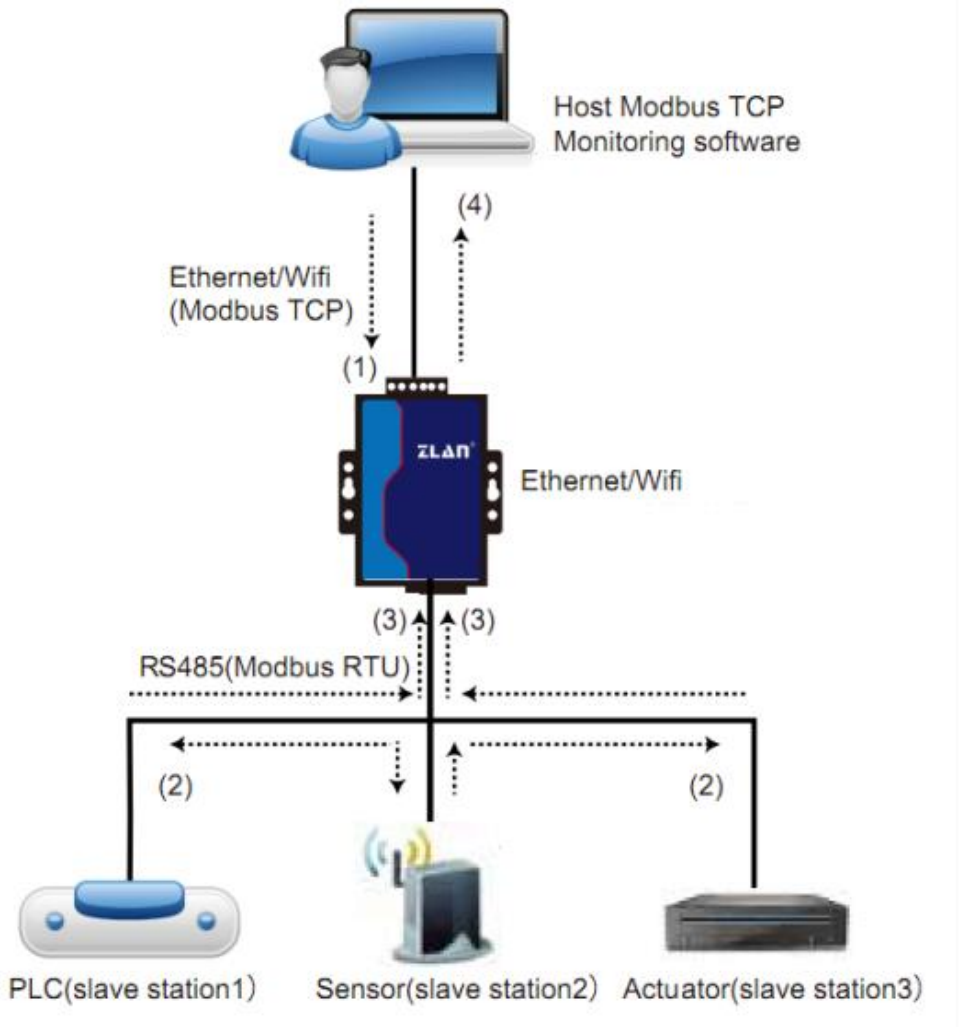

Figure 29 The operational mode of the storage-type Modbus gateway

As shown in Figure 29, the typical flow of Modbus TCP data is as follows: (1) (2) (3) (4). First, the Modbus TCP command is converted into the corresponding Modbus RTU command, then the device responds with the Modbus RTU command to the Modbus gateway, and finally, the Modbus gateway converts it back to Modbus TCP and sends it to the supervisory host.

We know that Modbus TCP, being a network communication protocol, has fast transmission speeds, typically responding within 3ms. On the other hand, Modbus RTU, which operates over RS485, typically has a speed of only 9600bps, requiring at least 30ms for sending and returning a single command. Consequently, the query response time of a typical non-storage Modbus gateway is relatively long.

Additionally, if multiple supervisory hosts simultaneously query data, the serial port may become congested. If we liken the network to a highway and the serial port to a single-lane bridge, the original method would be like directing highway traffic through the single-lane bridge.

The storage-type Modbus gateway (RS485_TO_ETH) addresses these issues by temporarily storing the queried register data internally. This allows the Modbus gateway to immediately respond to Modbus TCP queries, thereby leveraging the fast nature of Modbus TCP. Furthermore, the RS485_TO_ETH_RAIL model can actively send instructions from the serial port to automatically update the current saved register data, keeping a record of the latest register values.

Moreover, RS485_TO_ETH_RAIL is a fully automatic, configuration-free Modbus gateway. Users do not need to configure register addresses, function codes, slave addresses, etc. RS485_TO_ETH dynamically adds these registers based on the Modbus TCP commands received from the network port.

When monitoring multiple computers, RS485_TO_ETH_RAIL exhibits good response times, typically providing responses within 3ms regardless of the serial port's baud rate, and demonstrates excellent real-time updating speeds for serial port data.

The storage-type Modbus gateway is a true Modbus TCP to Modbus RTU converter, effectively leveraging the advantages of fast Modbus TCP speeds and the ability to handle queries from multiple hosts simultaneously.

It's important to note that when the serial port server acts as a TCP client, it does not have storage functionality and will automatically switch to non-storage mode.

Below are the features of the storage-type Modbus:

1. The first Modbus TCP query command is non-storage. This is because it must wait for the slow response from the RTU device before replying with the register content to the network port.

2. If a specific command is not queried by any supervisory host on the network

within 5 seconds, it is automatically deleted, and no longer forwarded from the serial port to the RTU device.

3. Currently, it can store up to 10K of Modbus cache, allowing for approximately 500 simultaneous single-register queries.

4. When multiple commands are being queried simultaneously, they are sent sequentially. The first command is sent, then the Modbus gateway waits for the RS485 anti-collision time (referencing the section on multiple hosts) before sending the response to the first command. Then, it proceeds to send the second command, and so on, until all commands have been responded to. After the last command is responded to, it returns to the first command.

2.6.3 Disabling Storage-Type Functionality

Although the storage-type Modbus provides faster response times, some users may not want their RTU devices to receive a large number of query commands, as this could impact the internal processing speed of the instruments. In such cases, the storage-type functionality can be disabled.

To disable storage-type functionality, follow these steps:

1. In the "Device Settings" dialog, click on the "More Advanced Options" button.
2. Uncheck the boxes labeled as shown in Figure 30.
3. Click "OK" to confirm the changes.
4. Return to the device settings and click on "Modify Settings".

Please note that when configuring the protocol conversion using the web interface, the default setting is for a non-storage-type Modbus gateway.

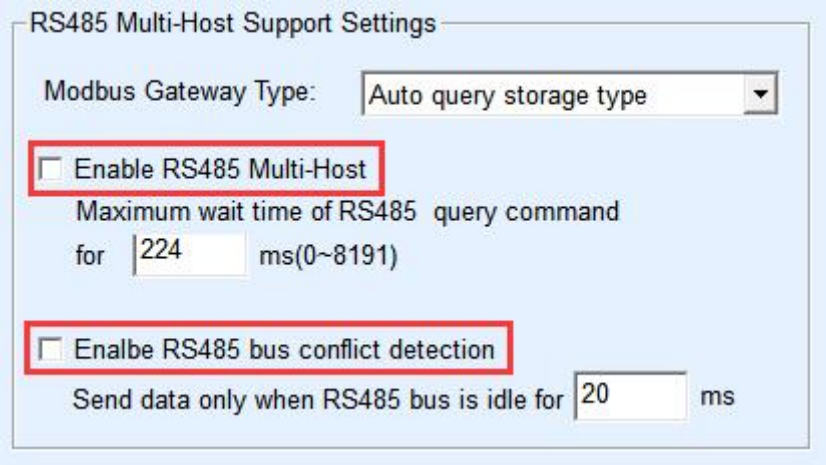

Figure 30: Disabling Storage-Type Functionality

2.7 Multi-host Function

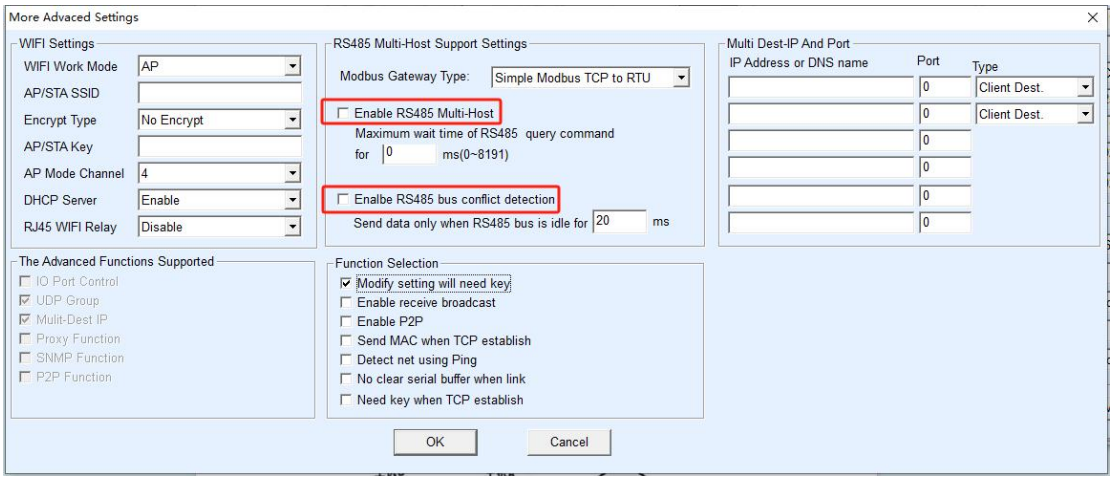

Figure 31

"RS485 Multi-Host" and "RS485 Bus Conflict Detection" shown in Figure 31 represent Seengreat's multi-host functionality. They are generally enabled or disabled simultaneously. Enabling these features allows devices with Modbus TCP protocol conversion to have storage-type Modbus gateway functionality. Otherwise, they function as non-storage-type Modbus gateways. If the protocol conversion is set to none, it typically allows user-defined RS485 protocols to support multiple hosts accessing serial devices simultaneously. This is not achievable in a pure RS485 network because simultaneous transmissions from multiple masters would result in

conflicts on the RS485 bus. Seengreat's multi-host feature for serial port servers can "coordinate" the RS485 bus to achieve multi-host access.

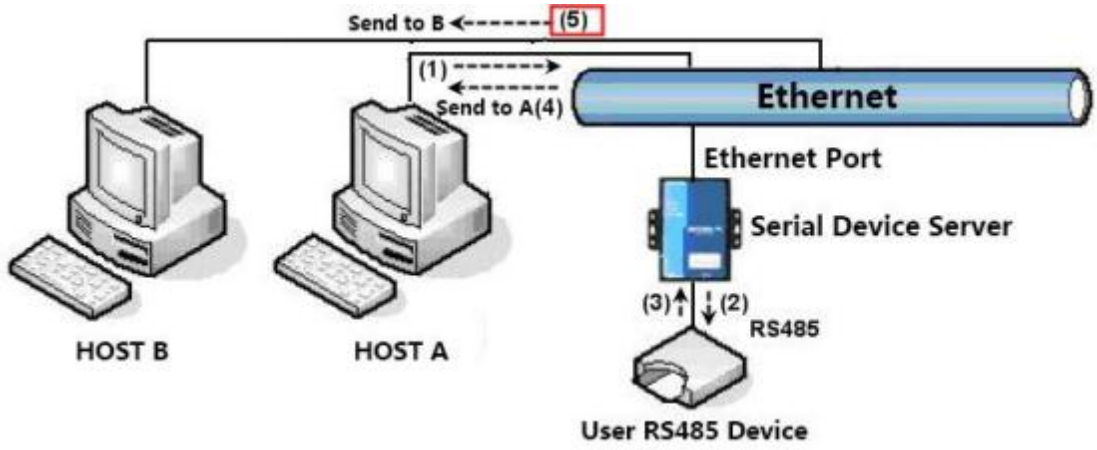

Figure 32: Multi-Host Functionality Demonstration

As shown in figure 36, in normal mode, when two hosts, host A and host B, are connected to the serial port server at the same time, host A sends (1) instruction, RS485 device receives (2) instruction, and RS485 device returns (3) instruction, but the serial port server sends (4) to host A and (5) to host B at the same time. Since host B did not send a query, but it also received a reply instruction (5), host B may generate a communication exception error.

In the multi-host mode, only instructions (4) will be given and no instructions (5) will be given, because the serial port server will automatically remember the host that needs to be returned and only return the instructions to the nearest communication host. The query of host A will only reply to A, and the query of host B will reply to host B.

Another benefit is in preventing data collision on the RS485 bus. In standard mode, simultaneous transmissions from Host A and Host B would result in merged commands on the RS485 bus, making it difficult to recognize individual commands. However, in multi-host mode, the serial port server can manage the sequence of commands from Hosts A and B on the bus, effectively resolving conflicts arising from multiple hosts accessing the bus simultaneously.

If the Transfer Protocol is "none",multi-host functionality is not enabled by default. To activate multi-host functionality, navigate to the device configuration dialog, click on "More Advanced Options," and then check "RS485 Multi-Host Support."

2.7.1 Multiple Host Parameters

The meanings of "RS458 Multi-Host Support" and "RS485 Bus Conflict Detection Function" are as follows.

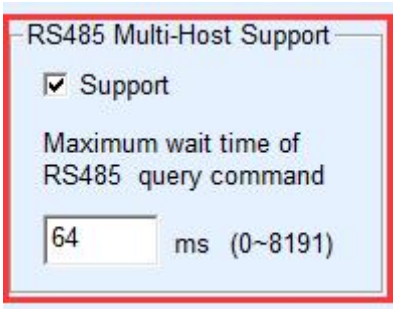

Figure 33: RS485 Multi-Host Support

The RS485 command response timeout refers to the maximum time interval from when the serial port server's serial port begins sending this command to when it receives a response. The specified time should be greater than the actual maximum time interval. This is because if it is determined to be a timeout, the next command will be sent.

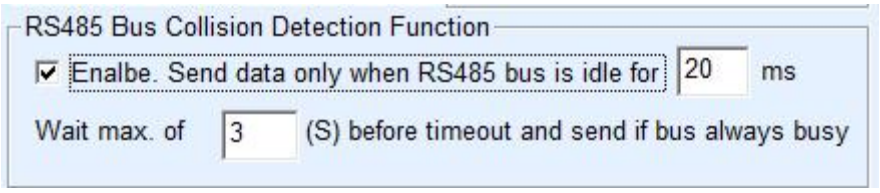

Figure 34: RS485 Anti-Collision Idle Time

RS485 bus collision time: Indicates the number of milliseconds the serial port server waits after receiving a reply to the first command before sending the second command. This parameter effectively defines the speed of instruction rotation. This value is recommended above 20ms.The parameter "maximum wait time 3 seconds"

generally does not need to be modified.

When users select the Transfer Protocol as "Modbus_TCP Protocol" using Vircom, Vircom will automatically check the above two enable boxes (unless the user manually enters the advanced options to remove them), 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 if the Transfer Protocol is set to "none", then these two parameters need to be manually configured.

Below are the recommended settings for the parameters mentioned above:

1. Figure 34 shows the "RS485 bus anti-collision time": Generally, it can be set to twice the value of the "Packet Interval" in the parameter configuration interface, but it should not be less than 20ms.

2. Figure 33 shows "RS485 instruction reply timeout time": Generally, it is determined based on the length of the command and its response. If the sent command is N bytes and the response is M bytes, then the recommended value to set is: "Packet Interval" × (N + M + 5) + 100 ms.

2.7.2 Modbus under Multiple Destination IP

If the serial port device (RTU device) acts as the master station, and the network port device (Modbus TCP device) acts as the slave station, with multiple network port slave devices present at the same time, then the method introduced in section 2.4.3 for a client to connect to multiple servers can be used to allow the serial server to act as a client and connect to these multiple network port devices simultaneously.

The functionality that needs to be implemented at this time is: after the serial RTU sends a command, it should be able to send it to multiple network port devices. The network port devices identify whether the command is for them through the Slave ID field, and only the network port device with the corresponding Slave ID responds. After the network port responds and sends it to the serial server, it is converted into an RTU command and output from the serial port to the RTU device.

It should be noted that at this time, the two checkboxes shown in Figure 34 as "RS485 bus anti-conflict time" and Figure 33 as "RS485 command response timeout time" need to be unchecked. Otherwise, the above forwarding function cannot be realized.

Another application method is: although the serial server connects as a Client to multiple network port devices, the RTU device does not act as the master station and still responds to the network port device's transmission (acting as a slave station). In this case, the two checkboxes for "RS485 bus anti-conflict time" and "RS485 command response timeout time" still need to be checked. This way, the function of multiple hosts accessing a single RTU device simultaneously can be realized.

2.8 Registration Packets and Heartbeat Packets

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

2.8.1 Registration Packets

The definition of a registration packet is that when the computer software establishes a TCP connection with the serial server module (hereinafter referred to as the module), the module will first send a string of codes to the software. This allows the software to know which module is communicating with it. This string of codes is the registration packet.

Registration packets are very suitable for IoT monitoring because cloud software generally runs on public Internet servers, while the modules are distributed at various collection and monitoring points. How to let the cloud software identify the module is very important and is essential for realizing IoT communication.

The serial server provides the following multiple ways of registration packets:

1. Sending MAC Address upon Connection

Sending MAC Address: The approach is that when the module connects to the cloud, it sends its MAC address 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 packet for each device, making it a simple and effective approach. To use this method: In the device settings dialog box, click on "More Advanced Options," find "Send MAC Address upon TCP Connection Establishment" in the upper middle part, check the box next to it, then return to the settings interface and click "Modify Settings."

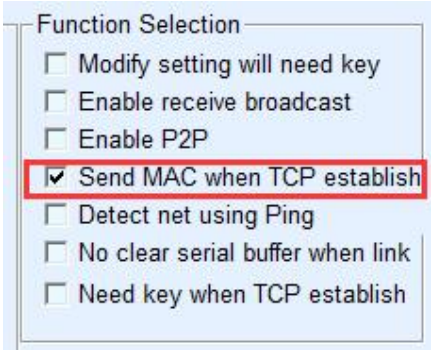

Figure 35 Sending MAC Address upon Connection

2. Realcom Protocol

The Realcom protocol is a mature protocol that includes registration packets and heartbeat packets. Users can employ this protocol to implement the functionalities of registration packets and heartbeat packets. The method to enable the Realcom protocol is: in the "Device Settings" dialog box, select "Conversion Protocol" as "REAL_COM Protocol". Note that for enabling the registration packet part, leave it unchecked (blank).

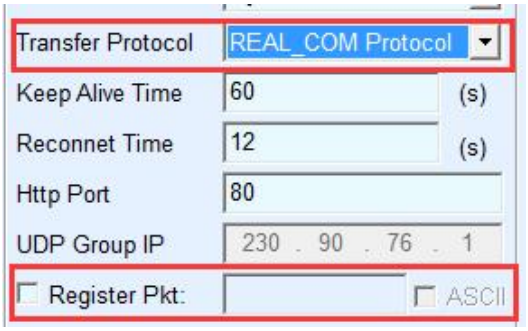

Figure 36 Enabling Realcom Protocol

It is not a transparent transport after the Realcom protocol enabled. It has the following features:

(1)When the device establishes a TCP connection with the cloud, it automatically sends a hexadecimal registration packet of FA 07 13 02 FA 02 MAC[5] MAC[4] MAC[3] MAC[2] MAC[1] MAC[0] FA FF. The MAC[5] to MAC[0] represents the device's MAC address.

(2)When the device sends data to the network, it automatically adds a 3-byte header prefix of FA 01 01.

(3)Every time the keep-alive timer interval elapses, the device sends a 1-byte heartbeat packet of 00 to the software.

The REAL_COM protocol, which contains the MAC address within its registration packet, can be used as a registration packet for the device. However, due to its fixed format, only the cloud software can design the REALCOM protocol to be compatible with this method.

3. Custom Registration Packet

The custom registration packet method allows users to specify an arbitrary registration packet format. The approach is as follows: in the device settings interface, configure as follows:

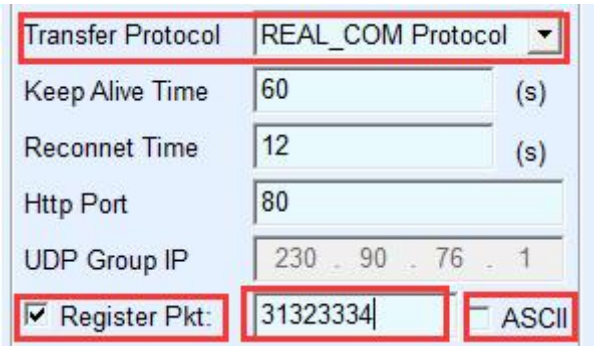

Figure 37 Setting Registration Packet

The difference from the REAL_COM protocol is that here, the registration packet is enabled, and registration packet information such as 31 32 33 34 is filled in. Note that this is in hexadecimal, which means the actual data sent is the string "1234". If

you need to display it as a string, click on the "ASCII" option next to it.

When the device connects to the cloud software, it can automatically send a hexadecimal registration packet of 31 32 33 34. This method of registration packet is more flexible and allows the device to adapt to the existing cloud registration packet format. However, the registration packet does not contain a wildcard such as MAC, so a different registration packet needs to be configured for each device, which can be more cumbersome.

The above methods of sending MAC addresses and REALCOM have the same configuration for each device, but due to different MAC address, the registration packets are naturally different.

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

4. Configuration File

The RS485 TO ETH RAIL from Seengreat supports the serial server writing a configuration file, which allows users to fully customize their own registration packets and also use MAC address wildcards. This can solve the trouble of writing custom registration packets for each device and there is no limit to the length of the registration packet.

2.8.2 Heartbeat Packets

Heartbeat packets are primarily used to detect whether the communication link has been broken. The implementation method is that the device sends a heartbeat packet data to the server software at regular intervals. Once the server receives this data, it will be discarded and not treated as valid communication data.

Heartbeat packets serve two main purposes: firstly, they allow the upper computer software to know that the device is active; secondly, if the device fails to send the heartbeat, the device acting as a TCP client will automatically re-establish the TCP connection. Therefore, it is a means of restoring network communication.

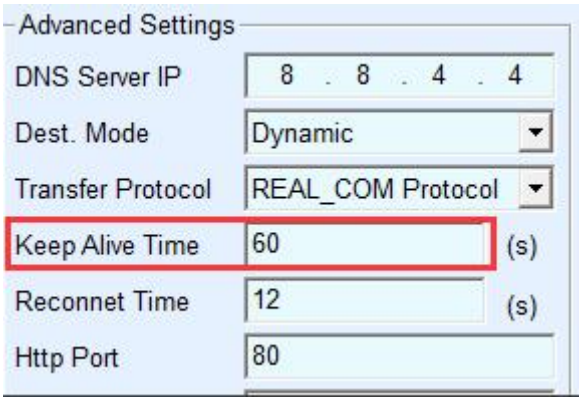

Figure 38 Keep-Alive Timing Interval

As shown in Figure 38, the sending interval of the heartbeat packet is set by the "Keep-Alive Timing Interval."

1、Implicit Heartbeat

Even if no heartbeat packets are set, the implicit heartbeat function is enabled when the Seengreat device is in TCP client mode. Therefore, the implicit heartbeat function refers to the device sending data, but the server actually does not receive the heartbeat data. As a result, it cannot serve the first function of the heartbeat packet, which is for the server to detect whether the device is active or not. However, since the device is actually sending data, it can fulfill the second function of the heartbeat packet, which is to detect whether the TCP connection is normal. If a disconnection is detected, the device can still automatically re-establish the TCP connection.

2、REALCOM Protocol

As described in the aforementioned Realcom protocol, the REALCOM protocol can send a 1-byte data of 00 at every keep-alive timing interval. This data is the heartbeat packet of the Realcom protocol.

3、Custom Heartbeat Packet

Firstly, fill in the registration packet according to the method described above for customizing the registration packet. Then, to add a heartbeat packet, follow these steps: Click the "More Advanced Options" button in the device settings. In the second line of the multiple destination IP and ports, write the hexadecimal heartbeat packet

and change the option on the right to "Param. Dest.".

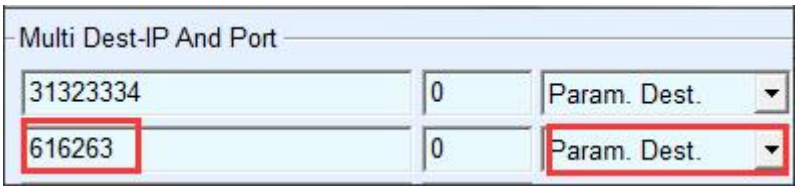

Figure 39 Custom Registration Packet

Note that the total combined length of the registration packet and the heartbeat packet must be less than 33 bytes. The first line is essentially the registration packet.

2.9 HTTP Client Communication Function

This feature is used to send the data from the serial server directly to a web-based server program, which can simplify the development workload of cloud-side software.

When the IoT collection terminal interacts with the web server (httpd program), if the data can be submitted to the web server in a format that complies with the HTTP GET and POST instructions, then the web server can use existing PHP/ASP languages to process and store the data. This saves the user the effort of developing a web application interface.

To support this feature, a configuration file named httpd.txt needs to be downloaded within the Seengreat serial server. The download can be achieved using the firmware upgrade function of Vircom.

The characteristics of the HTTP client communication function include:

1. Device Upload: Supports direct conversion of serial port data into HTTP format via GET/POST methods, which can be directly recognized by the server.

2. Web Server Download: The web server can also send the required data to the serial server via GET/POST instructions. The effective data content can be output from the serial port of the serial server. When the serial server receives the data, it can also provide a specific response to the web server to indicate that the data has been received.

3. Supports arbitrary conversion between input and output data in hexadecimal and string formats, facilitating the web server to send data in character format, while the serial port outputs hexadecimal data to control serial devices.

2.10 Firmware Upgrade Method

RS485_TO_ETH_RAIL can upgrade its own programs, but it is not possible to upgrade programs across different devices. Regardless of whether devices are found in the device list through automatic search, manual addition, or P2P search, this method can be used to upgrade the firmware.

1. Obtain the firmware file from Seengreat.

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

Figure 40 Restart Button

After the device restarts, search for the device again using the same method, and enter this dialog box once more. Click the "Upgrade Firmware" button located at the bottom right corner of the dialog box.

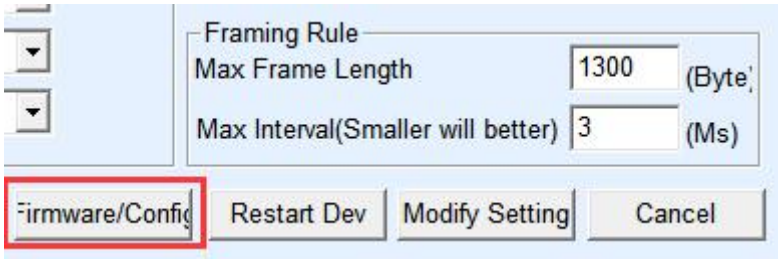

Figure 41 Firmware Upgrade Button

3. As shown in Figure 42, select the "Code File Download" radio button. In the

program file section, select the firmware file. The IP address for the serial server has already been automatically filled in and does not need to be re-entered. The module type/model has also been automatically selected. Then click on "Download."

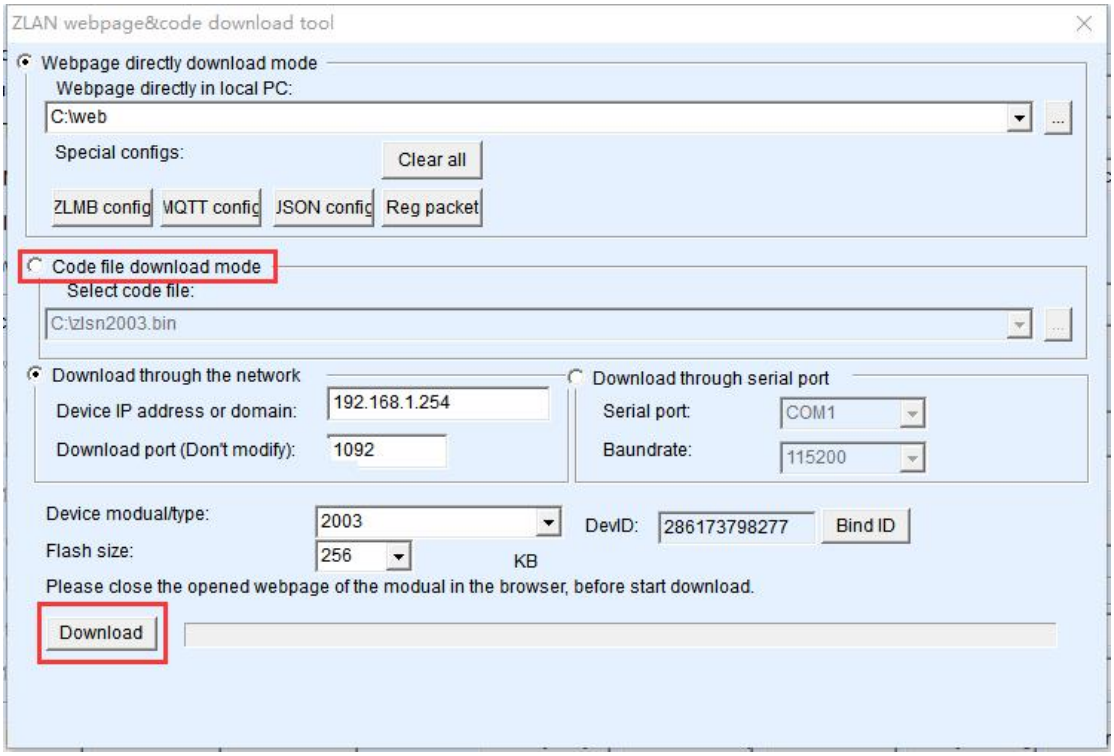

Figure 42 Firmware Upgrade Method

4. At this point, the download progress bar will start moving, and the download time is approximately around 30 seconds. During the download process, you will see the device's ACT LED flashing. At the end of the download, you will see the LINK LED flash a few times. Then a prompt box will pop up saying "Transmission complete, do not power off when the LINK LED flashes." Note: This only indicates that the transmission is complete. The writing process to the flash memory takes about 3 seconds, during which the LINK LED will flash. Please do not power off the device during this time.

5. After the download is finished, the program will usually restart automatically, and there is generally no need to power off. You will see the operating indicator LED flashing. If it does not restart automatically, wait for more than 30 seconds after the LINK LED stops flashing, and then power on again.

6. Web Configuration Interface Update: After the firmware upgrade, the internal configuration web page of the module also needs to be updated, otherwise, it will not be possible to configure via the web interface again, but it will not affect communication. If there is no need for web configuration, there is no need to download the web page. The method to download the web page is as follows: As shown in Figure 43, change the "Code File download mode" to "Webpage directly Download." Select the local web page's root directory as the directory where the web page files to be downloaded are located (this directory can be obtained from Seengreat), click "Download," and download all the files in the local web directory to the device's internal file system.

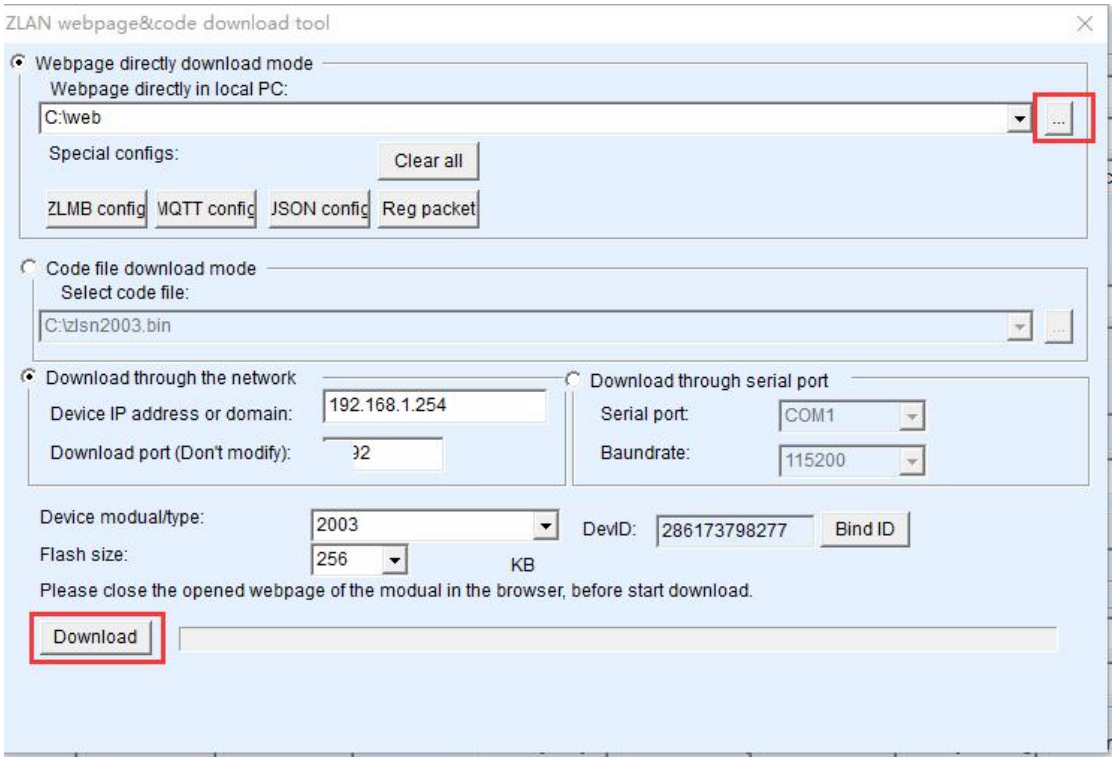

Figure 43 ZLSN2003 Web Upgrade Method

7.Note:

(1)If a failure message is prompted during the download, the device will not be damaged. Simply restart the download process. Additionally, do not power off the device when the LINK LED is flashing at the end of the download, as this could

damage the device.

(2)You can check the firmware version number through Vircom to determine if the new firmware has been successfully downloaded.

Figure 44 Check Firmware Version After Upgrade Completion

III、Data

3.1 MQTT Gateway and JSON Usage

Refer to the attached Seengreat MQTT and JSON User Manual.

IV、Related links

V、FAQ

VI、Appendix

6.1 Product Notes and Maintenance

6.1.1 Notes

- Do not plug or unplug modules while they are powered on.
- Follow all warnings and guidelines provided on the product.
- Keep the product dry. In case of accidental splashing or immersion in any liquid, immediately disconnect the power and thoroughly dry the product.
- Ensure proper ventilation and heat dissipation in the operating environment to avoid damage to components due to high temperatures.
- Do not use or store the product in dusty or dirty environments.
- Avoid using the product in environments with frequent temperature changes to prevent condensation damage to the components.
- Handle the product gently, as dropping, hitting, or severe shaking may damage the circuits and components.
- Do not clean the product with organic solvents or corrosive liquids.
- Do not attempt to repair or dismantle the product by yourself. In case of any malfunction, please contact our company for repairs. Unauthorized repairs may damage the product, and any resulting damage will not be covered under warranty.

6.1.2 Maintenance

- Address: Room 9#-211, Building 9, Minhua Technology Information Incubation Base, No.D-10-2, Chaoyang Road, Qixing District, Guilin, Guangxi, China.
- Contact Person: After-Sales Maintenance Department
- Phone：0773-2608835

6.2 Technical Support

Technical Support Hours (Working Days):

Hong Kong Time: 9:30 AM - 11:30 AM, 1:30 PM - 5:30 PM

Phone: 0773-7799838

Fax: gxshengui@163.com