# VT-DTU500

# User Manual


Figure 1 VT-DTU500

# CONTENT

# 1. Overview

VT-DTU500 series is a compact and powerful Internet of Things gateway. In terms of hardware, it integrates a variety of interfaces: 4G, WIFI, network port and 232/485 interface. In terms of software, it has multiple practical functions such as data acquisition, Modbus gateway, MQTT gateway, RS485 to JSON conversion and routing. In order to meet the needs of different countries for network frequency bands, the VT-DTU500 series also offers multiple seed models to choose from.

VT-DTU500 sub-models as followings：

Table 1 VT-DTU500 sub-models

| Item No. | Function | 4G suitable zones |
|---|---|---|
| VT-DTU500E | WiFi/Eth/CAT1 4G Single serial port server /includes router function | Asia, Europe, Middle East, Africa |
| VT-DTU500SA | WiFi/Eth/CAT1 4G Single serial port server /includes router function | South America |
| VT-DTU500G | WiFi/Eth/CAT1 4G Single serial port server /includes router function | Global |

VT-DTU500 series support 9 ~ 24V wide voltage, round hole power access (can change the terminal access).

The basic application of VT-DTU500 can realize the conversion of 1 TCP/IP protocol to the serial port, and can realize the data collection of the serial port device through the network. The network side can use the virtual serial port to connect the serial port software or the direct TCP/IP communication software.

VT-DTU500 also supports the Modbus RTU to Modbus TCP function, and supports the storage Modbus gateway feature. It can also be used as a non-storage Modbus gateway.

**VT-DTU500 series applications：**

- Power electronics, smart meters and energy consumption monitoring;
- As a gateway to the Internet of Things as a communication bridge between devices and the cloud;
- Various configuration software and device communication interfaces;
- Access control security field equipment networking；

A typical application connection is shown in Figure 2. The original serial device is connected to the serial port of VT-DTU500, and the VT-DTU500 is connected to the computer through network cable /WiFi or to the cloud server through 4G. The software on the computer connects to the VT-DTU500 through TCP/IP, virtual serial port, or cloud server. Thereafter, any data sent by the serial device is transparently transmitted to the computer's software, and the data sent by the software to the VT-DTU500 over the network is transparently transmitted to the serial device.

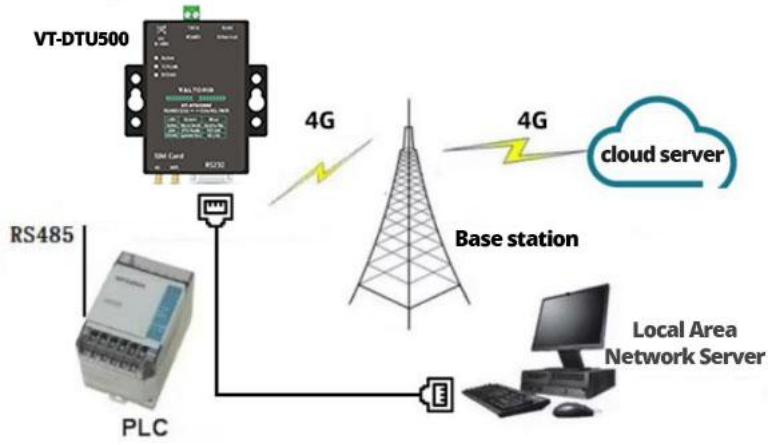


Figure 2 Connection map

# 2. Features

## 2.1 Hardware

VT-DTU500 series character as followings：

1. Various network connection methods: Support network ports, WIFI, 4G multiple ways to connect to the network.

2. Micro design: alloy metal shell anti-interference, small size, easy to install.

3. Abundant panel indicators are convenient for debugging: SYS/4G lights indicate the working status of the equipment, while independent TCP Link lights and Active lights indicate the TCPlink status and data communication status respectively.

## 2.2 Software

1. Support TCP server, TCP client, UDP mode, UDP multicast. The TCP client

also supports the TCP server function. The TCP server supports 30 TCP connections and the TCP client supports 7 destination IP addresses.

2. The baud rate ranges from 300 to 921600bps, and the data bit ranges from 7 to 8 bits. The parity bit can be none, odd, even, mark, or space.

3. Supports the MAC address sending function when the device is connected, facilitating device management by the cloud.

4. Provide computer side search, configure device secondary development kit DLL development library.

5. Supports Web browser configuration, dynamic IP address acquisition by DHCP, and DNS server addresses.

6. Support cloud remote device search, device parameter configuration, device program upgrade.

7. Remotely check the TCP connection status and serial port data sending and receiving status through the software. The virtual serial port supports data monitoring.

## 2.3 Advanced software features

Advanced software features supported by the VT-DTU500 series as followings：

1. Support Modbus gateway function, support Modbus RTU to Modbus TCP; Support ZLMB configurable table Modbus gateway function.

2. Multi-host function: In the question-and-answer query mode, multiple computers can access the same serial port at the same time. It can also realize multi-host application from one serial port to multiple serial ports.

3. Support MQTT gateway function.

4. Support Modbus RTU, Modbus TCP and 645 instrument-to-JSON protocols, support HTTP POST, HTTP GET format upload data.

5. The NTP protocol is used to obtain network time for sending protocol content.

6. Support custom heartbeat packet and registration packet functions: can facilitate communication and device identification with the cloud.

# 3. Technical parameter

Table 2 VT-DTU500 Technical parameter

| | | | |
|---|---|---|---|
| **Outline** | | | |
| Interface： | LAN:RJ45、485：connection terminal 、232：DB9 | | |
| Power supply： | Round hole mode (Can be customized terminal power supply) | | |
| Size： | length×width×height=9.4cm×6.5cm×2.5cm | | |
| Antenna： | WIFI*1、4G*1 Interface:50Ω/SMA male connector | | |
| SIM card | Power supply：3V，1.8V；size：big card（Card can purchase card sleeves to use） | | |
| **Communication interface** | | | |
| Frequency Band | VT-DTU500E | LTE Cat-1 | LTE-FDD: B1, B3, B5, B7, B8, B20 |
| | | 2G | GSM/GPRS/EDGE: 900/1800 MHz |
| | VT-DTU500SA | LTE Cat-1 | LTE-FDD: B1, B2, B3, B4, B5, B7, B8, B28,B66 |
| | | 2G | GSM/GPRS/EDGE: 850/900/1800/1900 MHz |
| | VT-DTU500G | LTE Cat-1 | LTE-FDD: B1, B2, B3, B4, B5, B7, B8, B20,B28,B66,B71<br>LTE-TDD: B34,B38,B39,B40,B41 |
| | | 2G | GSM/GPRS/EDGE: 850/900/1800/1900 MHz |
| **Serial port parameter** | | | |
| Baud rate： | 300~921600bps | | |
| Digit bits： | 7～8 bits | | |
| Parity bits： | None, odd, even, mark, space | | |
| **Software** | | | |
| Protocol： | ETHERNET、IP、TCP、UDP、HTTP、ARP、ICMP、DHCP、DNS | | |
| Configuration： | VirCOM tool、WEB | | |
| Transfer protocol： | Modbus TCP、MQTT、JSON、HTTP | | |
| Modbus gateway： | Support multi-host mode, storage mode, Pre-configuration table (ZLMB) mode | | |

| | |
|---|---|
| Communication mode： | TCP/IP Direct communication, virtual serial port mode |
| **Working mode** | |
| TCP server, TCP client (while TCP server also exists), UDP, UDP multicast | |
| **Power supply requests** | |
| Power supply： | 9～24V DC |
| **Secondary development** | |
| Development platform | Linux environment; Developed based on openwrt open source router system |
| Development language | C language |
| Compiler： | mipsel-openwrt-linux-gcc |
| **Environmental requirement** | |
| Operating temperature： | -40~85℃ |
| Storage temperature： | -45~165℃ |
| Humidity range： | 5~95% relative humidity |

# 4. Instruction

## 4.1 Hardware specifications

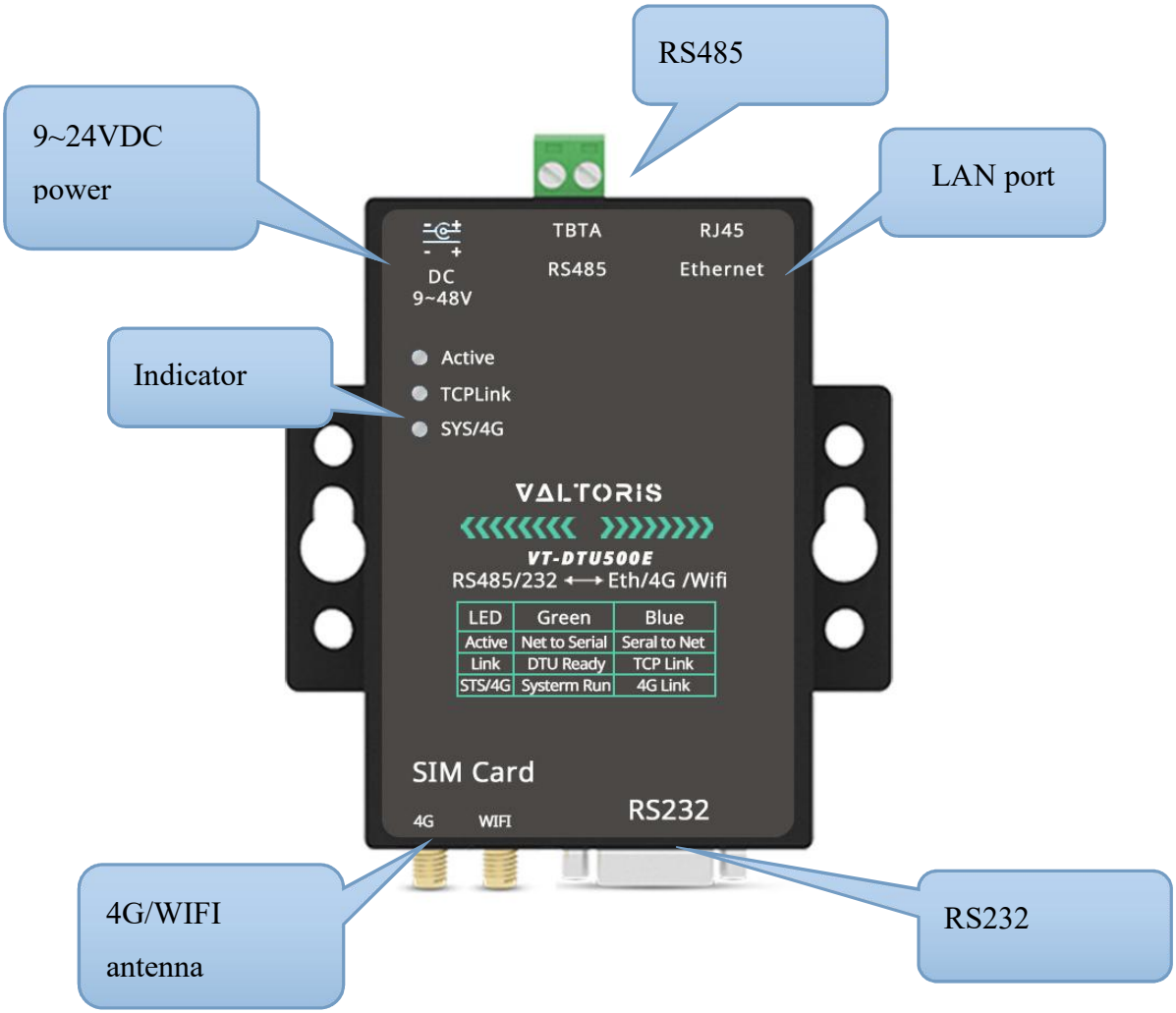


Figure 3 VT-DTU500 front view

The main view of the VT-DTU500 is shown above, with a black radiation-resistant SECC metal housing.

1. one Power supply: 9~24VDC round hole input, can change the terminal.
2. RS232: DB9 male input.
3. RS485:TA (+),TB (-) terminal input.
4. RJ45 port: The network port is a LAN port.
5. Antenna: The left is the 4G antenna, the antenna interface adopts SMA (male head), and the external antenna must use an antenna suitable for 4G working band; On the right is the WiFi (2.4G) antenna, the antenna interface adopts 50 Ω /SMA (male head), and the external antenna must use an antenna suitable for the 2.4G WIFI working band.

6. Indicator: Divided into Active light, TCPLink, SYS/4G light.

Table 3 Indicator contents

| | |
|---|---|
| Active | (1) When the network port sends data to the serial port, the indicator is green. The flashing time is one second longer than the actual communication time, making it easier to detect short data communications.<br>(2) When the serial port sends data to the network port, the indicator is on blue and green at the same time. If the blue color is displayed, the serial port returns data to the network port. This can determine whether the device has a response to the command of the upper computer, if there is no corresponding indicates that the serial port baud rate is not correct or the serial port is not connected. |
| TCP Link | (1) Steady green indicates that the internal module is working properly.<br>(2) If it is steady blue, the TCP connection is established (or in UDP mode). It can be used to determine whether the gateway establishes a communication link with the host software. |
| SYS/4G | (1) Blinking green indicates that the gateway WIFI function works normally.<br>(2) If it is steady blue, the gateway is dialing up 4G.<br>(3) If the indicator blinks blue, the 4G dial-up of the gateway is successful. |

Use the indicator to debug communication methods:

1) If the 4G light is on, it indicates that the 4G dial has not succeeded, please check the SIM card.

2) If the Link indicator is not blue (only considering the TCP working mode), the host software is not connected to the serial port server. Please consider whether the IP address is configured in the same network segment.

3) If the ACT indicator does not blink, it indicates that there is no data communication. Check the parameter Settings and serial port connection.

Figure 4 interface picture

Figure 4 shows the upper end interface:

1. one Power input: Q2.1 socket, DC+9V~ +24VDC. The default adapter is 12V. Can be customized as power terminal type input.
2. Input RS485 signals. Be careful not to connect the power supply.
3. RJ45 port, Ethernet access port, LAN port.

Figure 5 shows the lower ports：

Figure 5 lower ports picture

4. Antenna: The antenna interface uses 50Ω/SMA, and the external antenna must use an antenna of the appropriate working band.

5. SIM Card installation: When installing the SIM card, ensure that the device is not powered on. Use a pen tip or screwdriver to push the SIM card out of the slot and push the SIM card face down into the slot.

6. DB9: RS232 signal input.

## 4.2 Hardware connection

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

The serial port must be connected according to the user serial port device. To connect port 485, connect the positive part of port 485 to 485A and the negative part of port 485 to 485B.

At the same time, the network port must be connected to the LAN port of the VT-DTU500. It can be directly connected to the computer or connected to the network through the switch.

## 4.3 Software installation

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

Table 4 Software Version

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

## 4.4 Parameter configuration

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

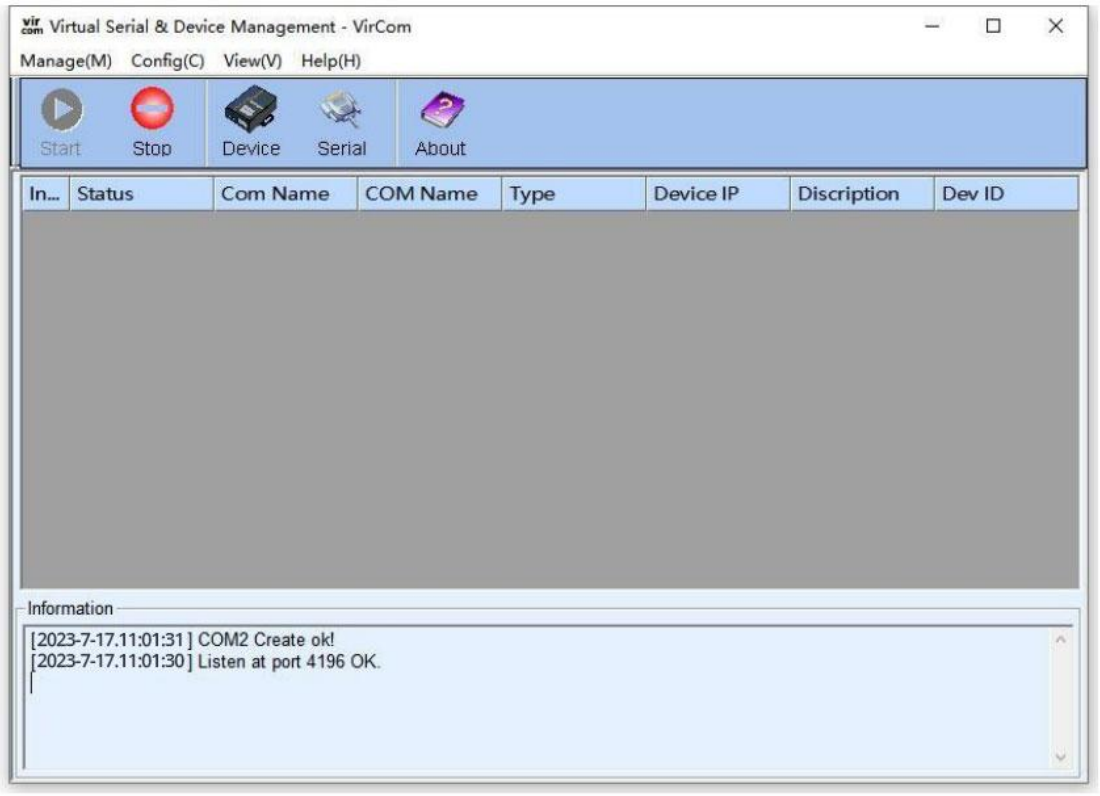


Figure 6 Vircom Main Interface

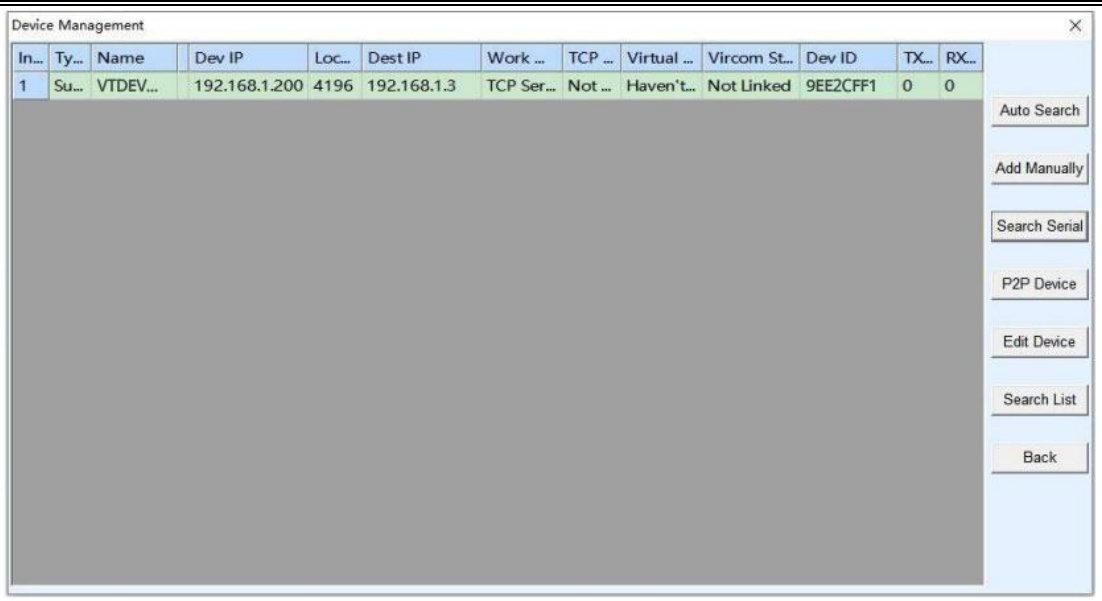


Figure 7 Device List

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

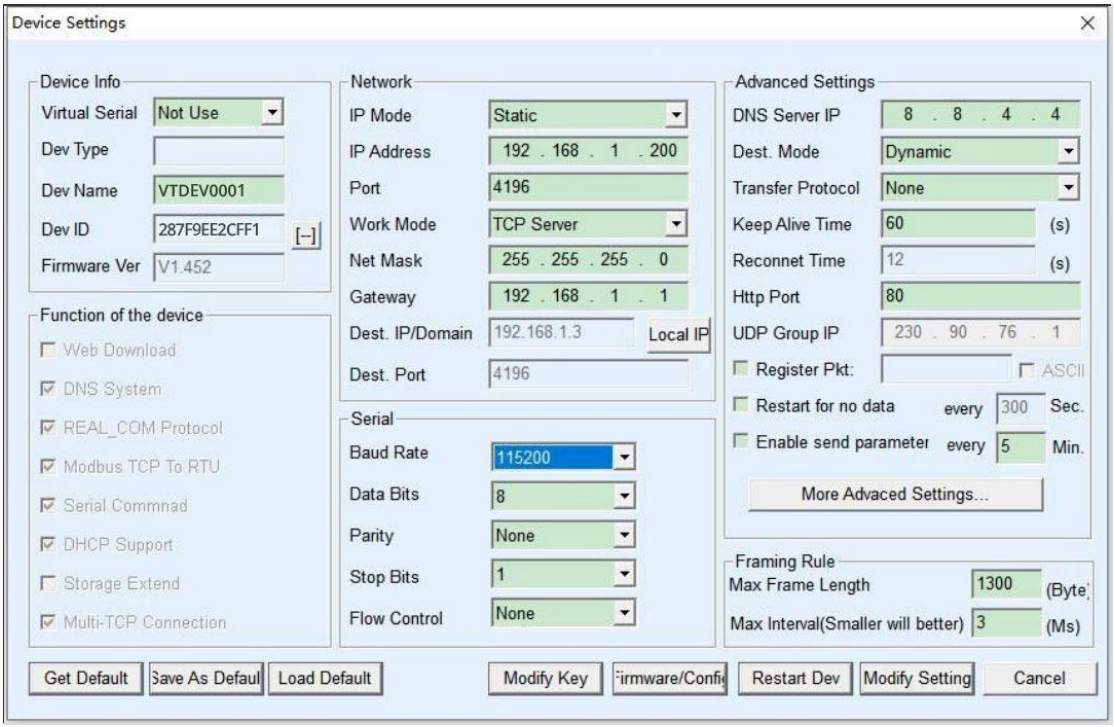


Figure 8 Device Parameters

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

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

The detailed meanings of other parameters are as follows:

Table 5 Parameter Meaning

| Parameter | Value Range | Meaning |
|---|---|---|
| Virtual Serial | Port Unused, created virtual serial port | You can bind the current device to a created virtual serial port. Please add a COM port in "Serial Port Management" on the main interface first. |
| Device model | | Only display the core module model |
| Device name | Any | You can give the device an easy-to-read name with a maximum length of 9 bytes. |
| Device ID | | Factory unique ID, cannot be modified. |
| Firmware Version | | Core module firmware version |
| Supporting Function | | Refer to Table 6 for the functions supported by the device. |
| IP Mode | Static, DHCP | Users can choose static or DHCP (dynamic IP) |
| IP Address | | IP address of the serial device server |

| | | |
|---|---|---|
| Port | 0～65535 | The listening port of the serial port server when it is in TCP Server or UDP mode. When acting as a client, it is best to specify port 0, which is beneficial to improve the connection speed. When using port 0, the system will randomly assign a local port. The difference between this and non-zero ports is: (1) When the local port is 0, a new TCP connection is established with the PC when the module restarts. The old TCP connection may not be closed, and there may be multiple false connections with the device. Generally, the host computer hopes to close the old connection when the module restarts; specifying a non-zero port will close the old connection. (2) When the local port is 0, the TCP re-connection time is faster. When the serial port server is in TCP client mode, it also acts as a TCP server to listen for connections on the port. At the same time, the local port number used by the TCP client to connect to the server is "port + 1". |
| Working Mode | TCP server mode, TCP client mode, UDP mode, UDP multicast | When set as a TCP server, the serial device server waits for the computer to connect; when set as a TCP client, the serial device server actively initiates a connection to the network server specified by the destination IP. |
| Sub-net Mask | Eg: 255.255.255.0 | Must be the same as the subnet mask of the local area network. |
| Gateway | Eg: 192.168.1.1 | Must be the same as the local LAN gateway. |
| Destination IP or domain name | | In TCP client or UDP mode, data will be sent to the computer indicated by the destination IP or domain name. |
| Destination Port | | In TCP client or UDP mode, data will be sent to the destination port of the destination IP. |
| Baud Rate | 300~9121600 | Serial port baud rate |
| Data Bits | 7、8 | |
| Check digit | None, Even, Odd, Mark, Space | |
| Stop Bits | 1、2 | |

| | | |
|---|---|---|
| Flow Control | No flow control, hard flow control CTS/RTS, hard flow control DTR/DCR, soft flow control XON/XOFF | Only valid for RS232 serial port |
| DNS Server | | When the destination IP is described by domain name, you need to fill in the DNS server IP. When the IP mode is DHCP, you do not need to specify the DNS server, it will be automatically obtained from the DHCP server. |
| Destination mode | Static, Dynamic | In TCP client mode: When using static destination mode, the device will automatically restart after 5 consecutive failures in connecting to the server. |
| Conversion agreement | NONE、Modbus TCP、Real_COM | NONE means that data forwarding from the serial port to the network is transparent; Modbus TCP will directly convert the Modbus TCP protocol into the RTU (ASCII) protocol to facilitate coordination with the Modbus TCP protocol; RealCOM is designed to be compatible with the old version of the REAL_COM protocol and is a virtual serial port protocol. However, when using a virtual serial port, it is not necessary to select the RealCom protocol. |
| Keep-alive time | 0～255 | Heartbeat interval. (1) When set to 1-255, if the device is in TCP client mode, it will automatically send TCP heartbeats every "keep alive time". This can ensure the TCP validity of the link. When set to 0, there will be no TCP heartbeat. (2) When set to 0-254, when the conversion protocol is selected as REAL_COM protocol, every keep alive time, the device will send a data with a length of 1 and content of 0 to implement the heartbeat mechanism in the Realcom protocol. When set to 255, there will be no realcom heartbeat. (3) When set to 0-254, if the device is working in TCP client mode, the device will send device parameters to the destination computer every keep alive time. When set to 255, there will be no parameter sending function, which can realize remote device management. |

| | | |
|---|---|---|
| Disconnection re-connection time | 0～255 | In TCP client mode, when the connection is not successful, each "disconnection re-connection time" will re-initiate a TCP connection to the computer. It can be 0 to 254 seconds. If it is set to 255, it means that it will never reconnect. Note that the first TCP connection (such as hardware power-on, device restart through Vircom software, no data light) will generally be made immediately. Only after the first connection fails will it wait for the "disconnection re-connection time" and try again, so the "disconnection re-connection time" will not affect the connection establishment time under normal circumstances between the network and the server. |
| Web access port | 1～65535 | The default is 80 |
| Multicast address | | Used for UDP multicast |
| Enable registration package | | When the TCP connection is established, the registration packet is sent to the computer. After enabling the registration packet, the realcom protocol must be selected. Supports TCP server and TCP client modes. |
| Data packet length | 1～1400 | One of the serial port framing rules. After receiving data of this length, the serial port server sends the received data as a frame to the network. |
| Data packet interval | 0～255 | Serial port framing rule 2: When the data received by the serial port of the serial server pauses and the pause time is greater than this time, the received data will be sent to the network as a frame. |

**The functions supported by the device are described as follows：**

Table 6 Device Supported Function

| Name | Description |
|---|---|
| Domain name system | The destination IP can be a domain name (such as a server address starting with www). |

| | |
|---|---|
| REAL_COM protocol | A non-transparent serial port server protocol, suitable for multiple serial port servers to bind virtual serial ports through the Internet. Because the protocol contains the device MAC address, it helps the host computer to identify the device. It can be not used in general. |
| Modbus TCP to RTU | It can realize Modbus TCP to RTU conversion and also supports multi-host function. |
| Serial port modification parameters | Supports serial port AT commands to configure and read device parameters. |
| Obtain IP automatically | Support for DHCP client protocol |
| Multiple TCP connections | Supports more than 1 TCP connection when acting as a TCP server. |
| UDP multicast | UDP multicast |
| Multi-destination IP | As a TCP client supports simultaneous connection of 7 destination IP. |

## 4.5 TCP communication test

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

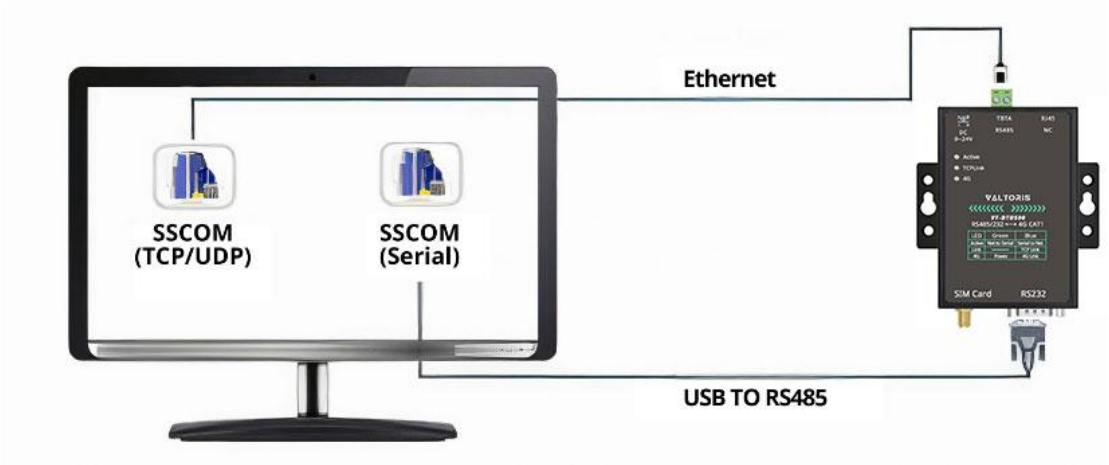


Figure 9 TCP communication diagram

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

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

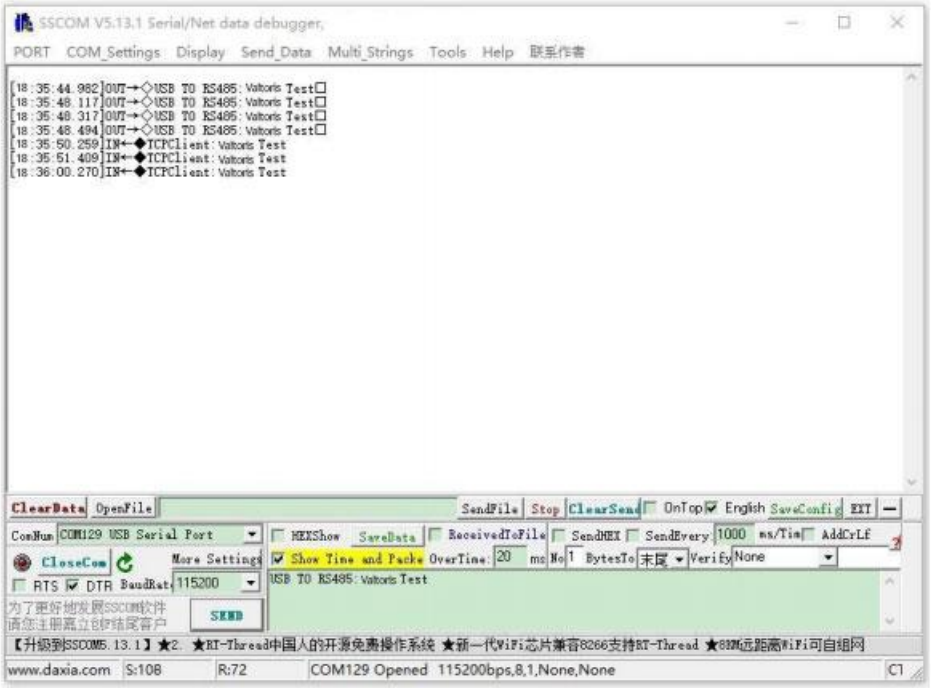


Figure 10 SSCOM1: USB to RS485 transceiver interface

In addition, open one more serial port debugging assistant window and use it as a TCP client mode. Fill in the destination IP as the IP of the serial server (currently 192.168.1.200) and the destination port as 4196, and then click the "Open" button, as shown in the figure below:

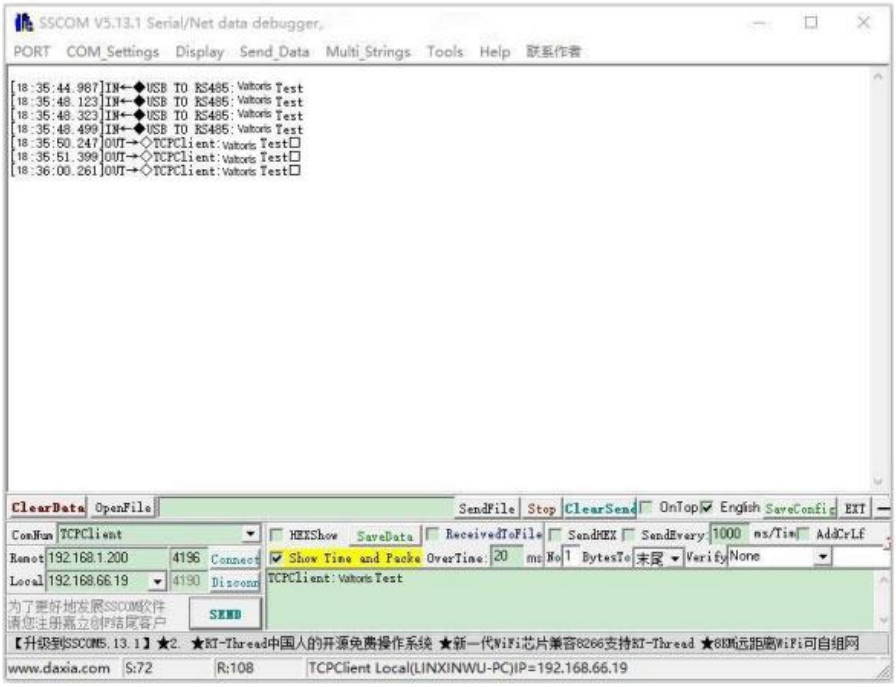


Figure 11 SSCOM2: TCP Client transceiver interface

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

## 4.6 Virtual serial port test

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

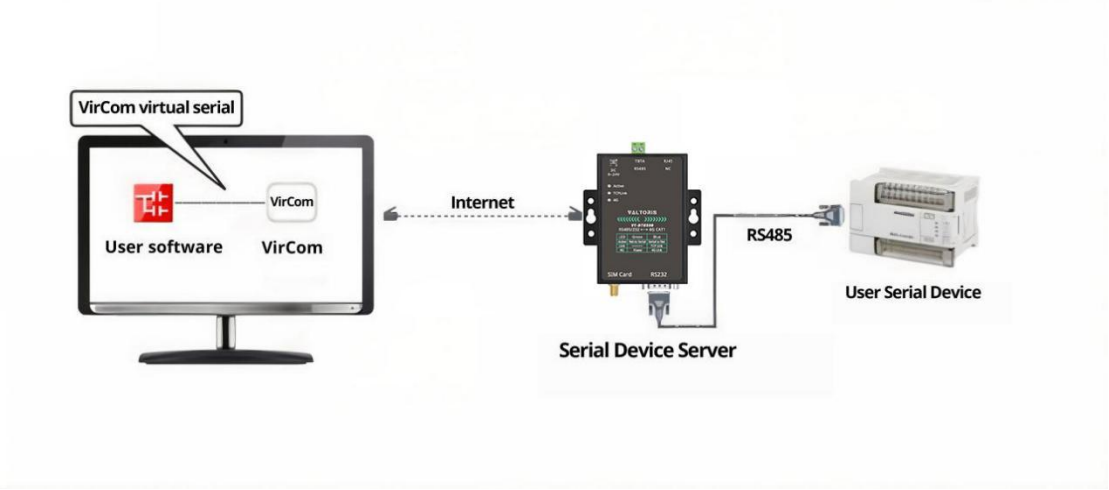


Figure 12 Function of virtual serial port

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

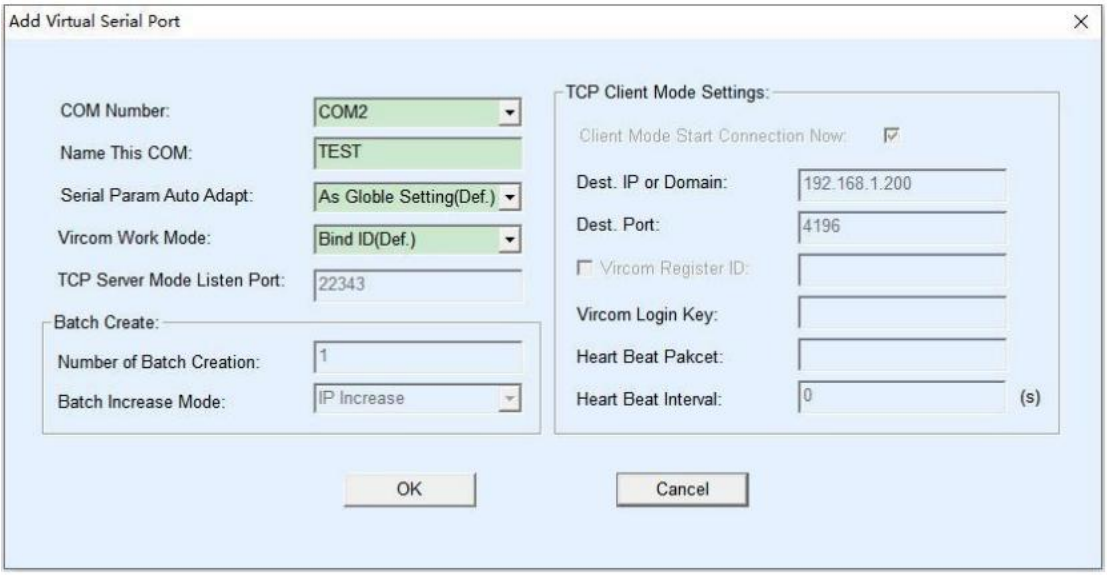


Figure 13 Add Virtual Serial Port

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

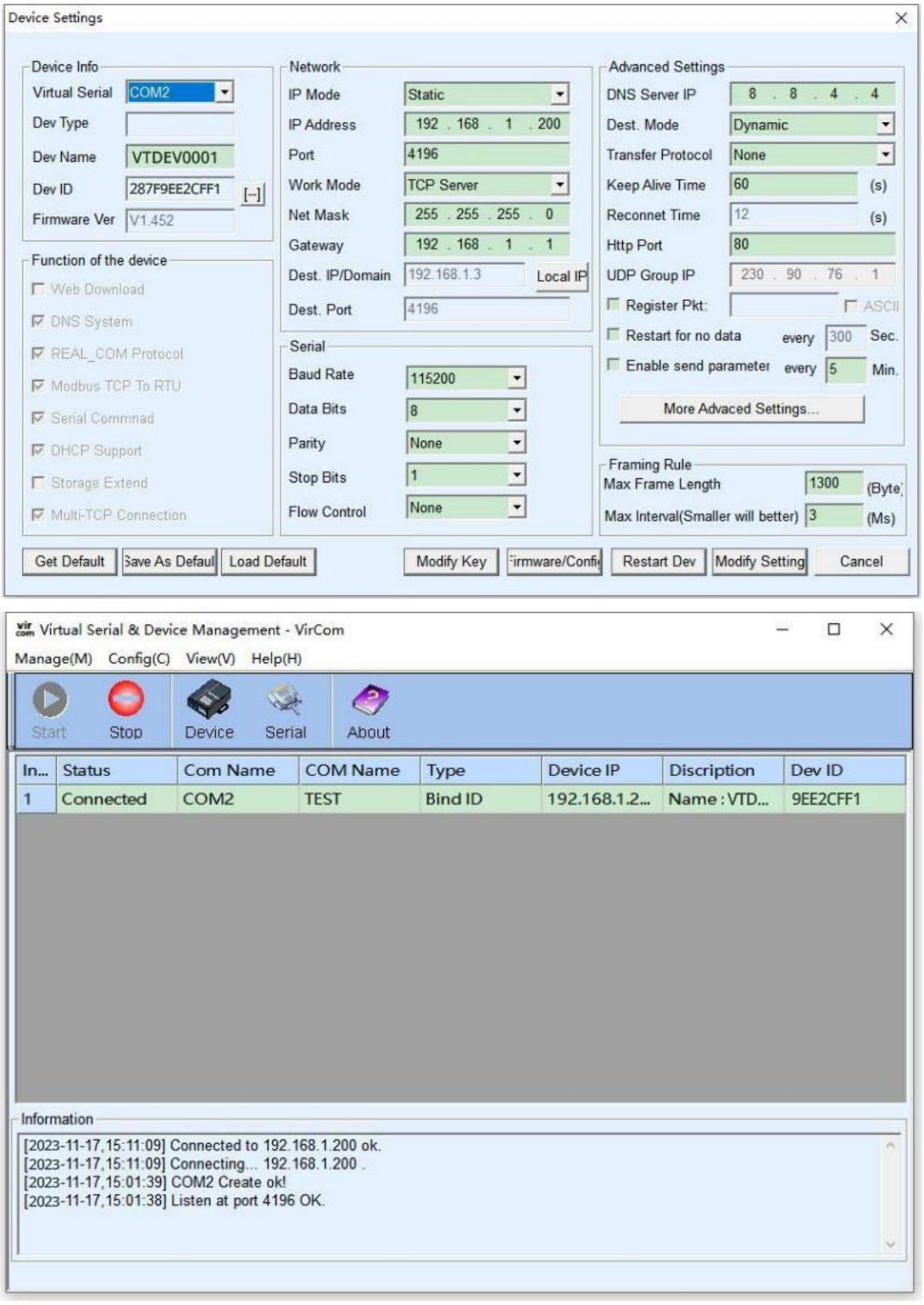


Figure 14 Virtual Serial Port has been connected

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

COM2: COM3 - serial port server serial port - serial port server network port - Vircom - COM2. The figure below shows the data transmission and reception of both parties.

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

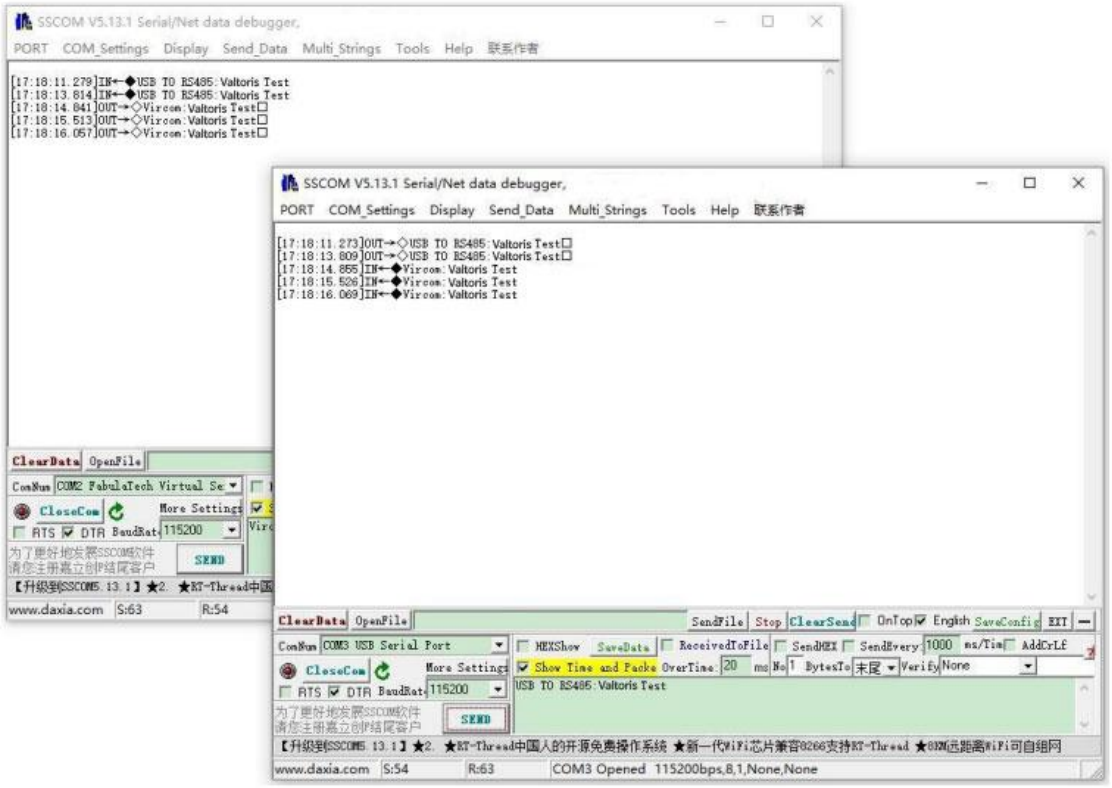


Figure 15 Communication via Virtual Serial Port

## 4.7 Modbus TCP test

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

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

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

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

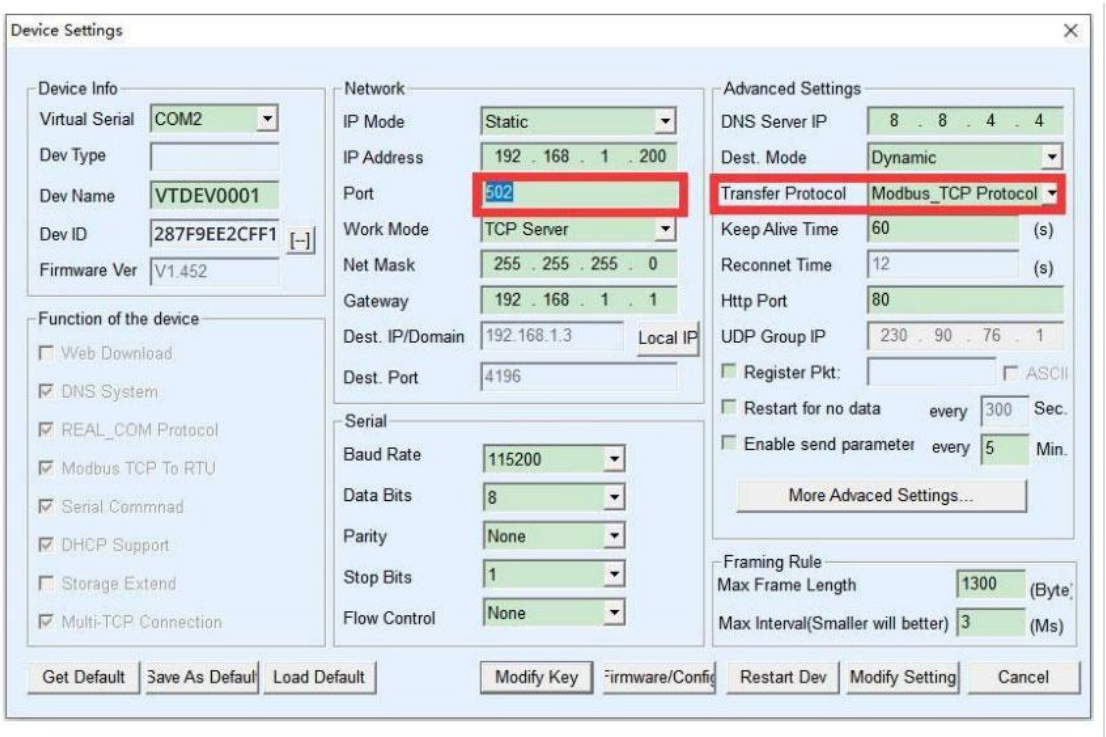


Figure 16 Enable Modbus TCP

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

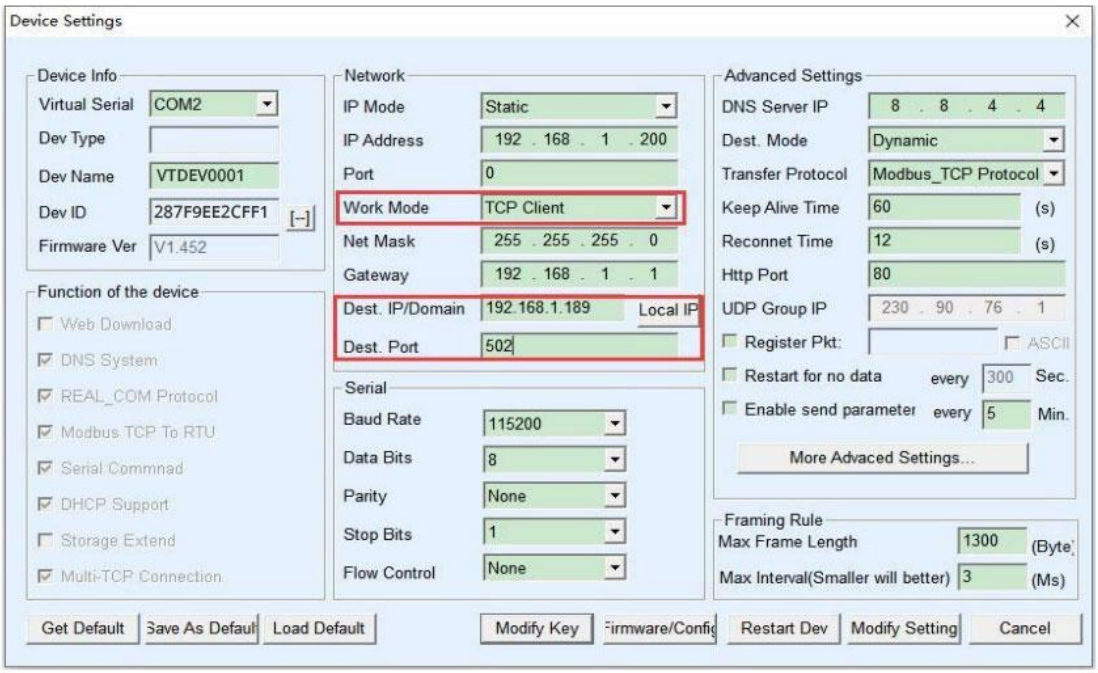


Figure 17 Modbus TCP as client

## 4.8 Web Configuration

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

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

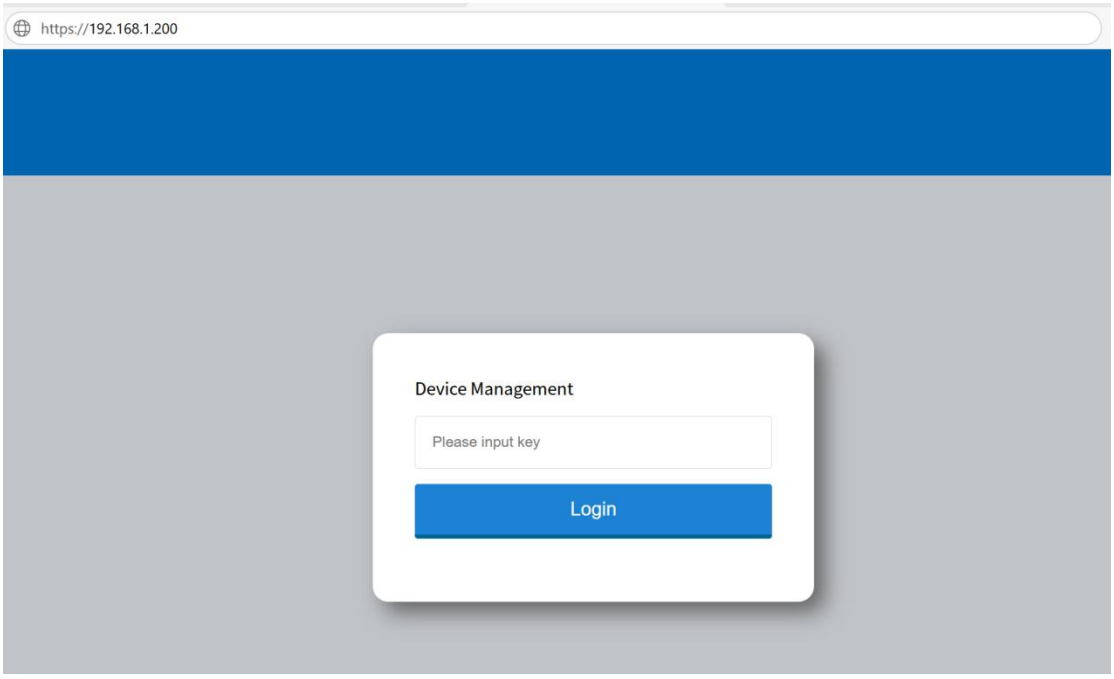


Figure 18 Web login interface

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

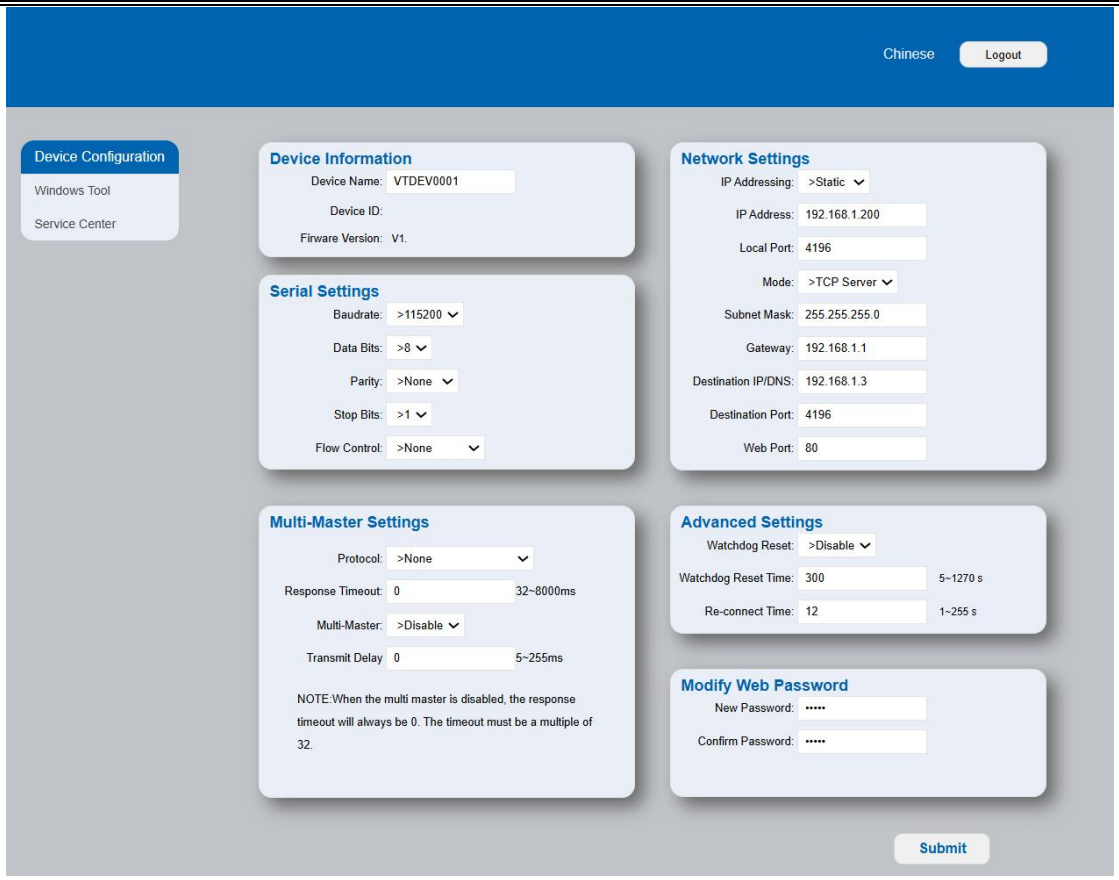


Figure 19 Web configuration interface

3. You can modify the serial port server parameters on the web page that appears. For related parameters, refer to Table 4 Parameter Meaning.

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

# 5. Work Mode and Transfer Protocol

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

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

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

Modbus TCP mode; Realcom protocol is currently only used when the multi-serial port server is used as a TCP client to connect to a server and the server uses a virtual serial port.

Usage is summarized as follows:

Table 7 network configuration modes

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

## 5.1 Virtual Serial Mode

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

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

a) If the computer is a server with a public IP leased on the Internet, then the device must use the TCP client method to connect the device to the server. At this time, you can choose ② and ⑤ in Table 6. If it is a multi-serial port server, you must choose ⑤.

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

## 5.2 Direct TCP/IP Communication Mode

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

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

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

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

### 5.2.1 TCP Client Mode

There are two working modes in TCP mode: TCP server and TCP client. Regardless of which mode is adopted, one must be the server and the other must be the client. After that, only the client can access the server. If they all are the servers or the clients, the communication is invalid.

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

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

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

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

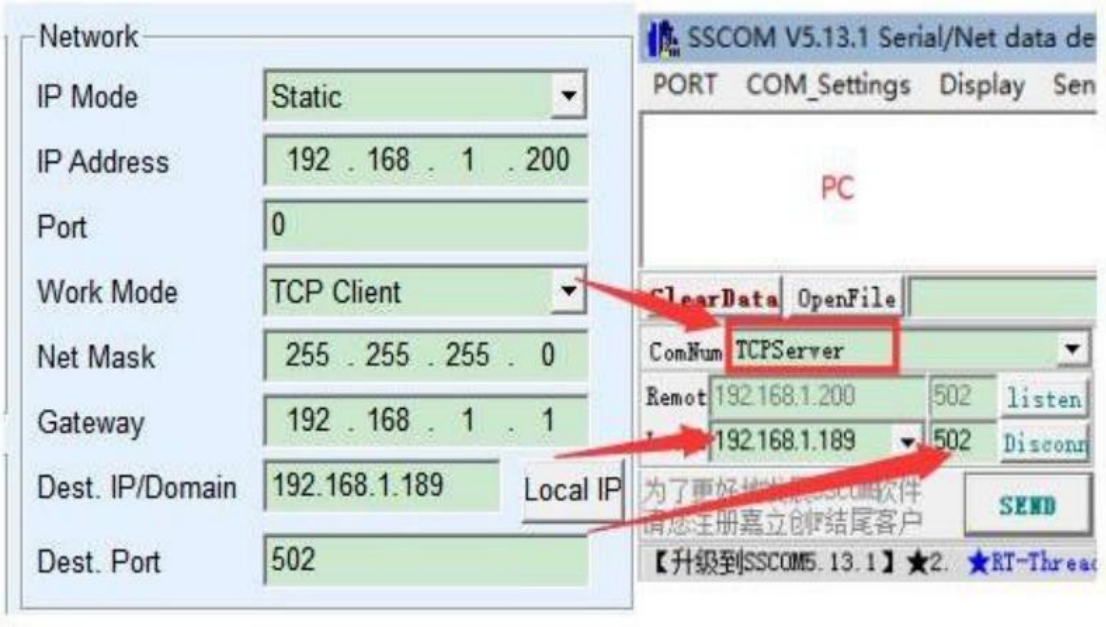


Figure 20 Serial server as a client

### 5.2.2 The Client connects to Multiple Servers

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

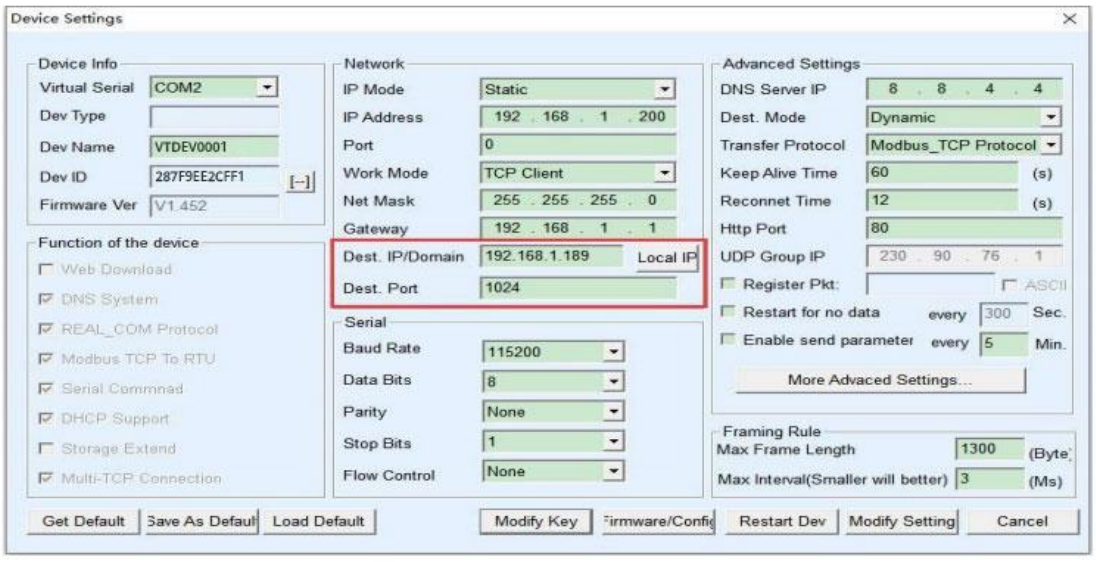


Figure 21 The first destination IP and port

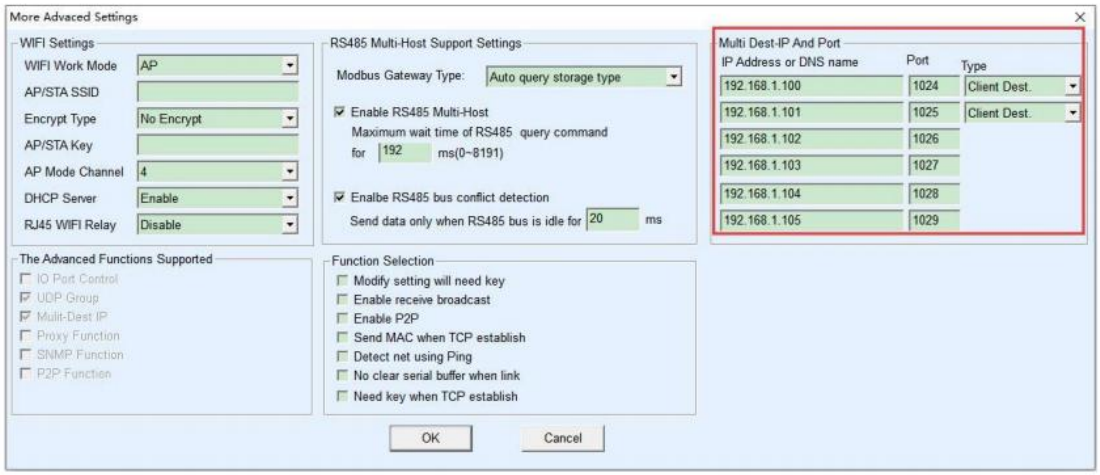


Figure 22 The remaining 2 to 7 IPs and ports

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

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

### 5.2.3 TCP Server Mode

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

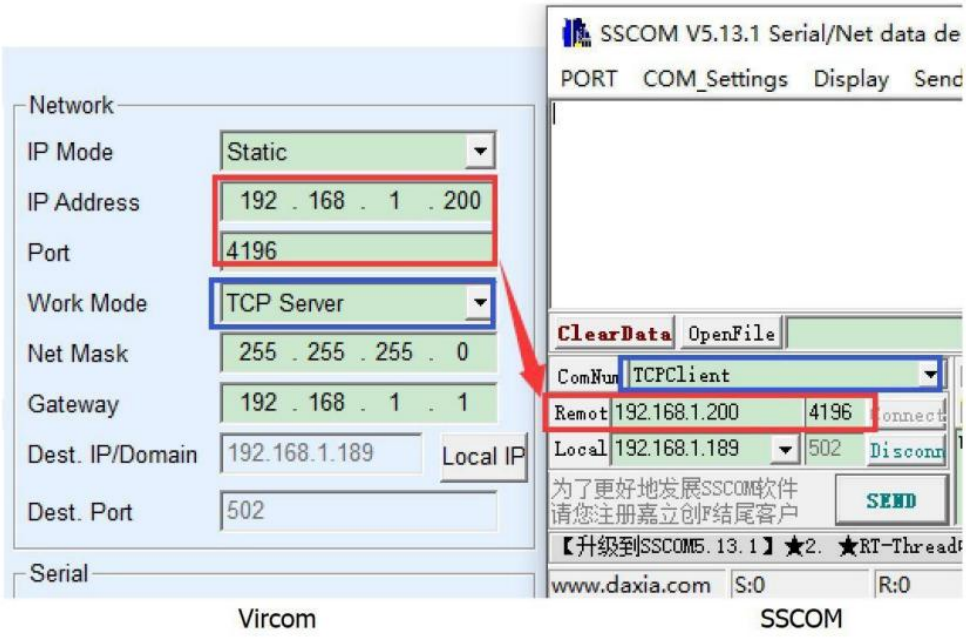


Figure 23 The serial port server as TCP Server

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

### 5.2.4 Acting as both client and server

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

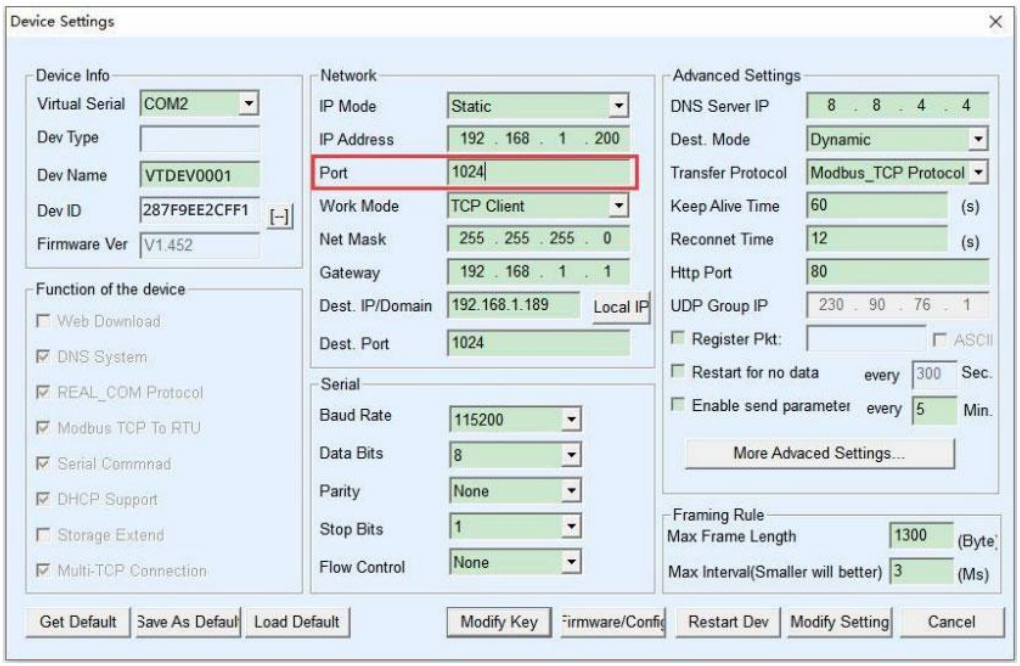


Figure 24 Acting as both Client and Server

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

As shown in Figure 24, the computer software can now connect to the 1024 port of 192.168.1.200 for communication, and the device will also act as a client to connect the 1024 port of 192.168.1.189. It should be noted that since the local port 1024 is occupied by the server, the local port when used as a client is "port+1", that is, the software on 192.168.1.189 sees that the incoming port of the device is 1024+1 =1025

### 5.2.5 UDP Mode

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

serial server.

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

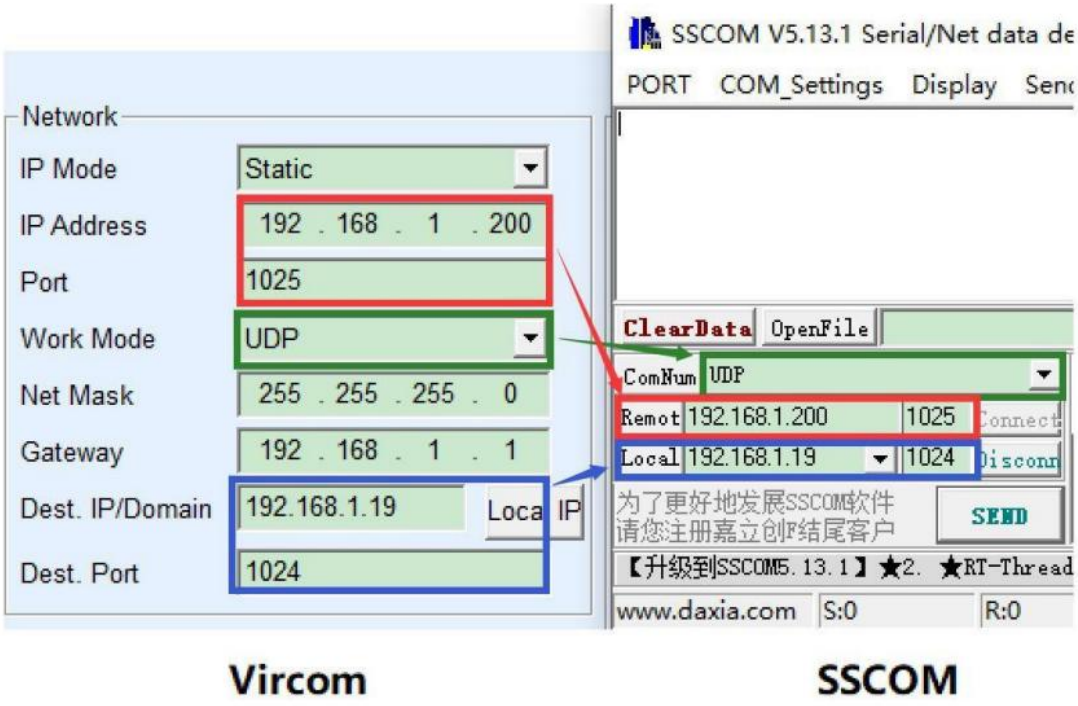


Figure 25 UDP mode parameter configuration

## 5.3 Device Connecting Method

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

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

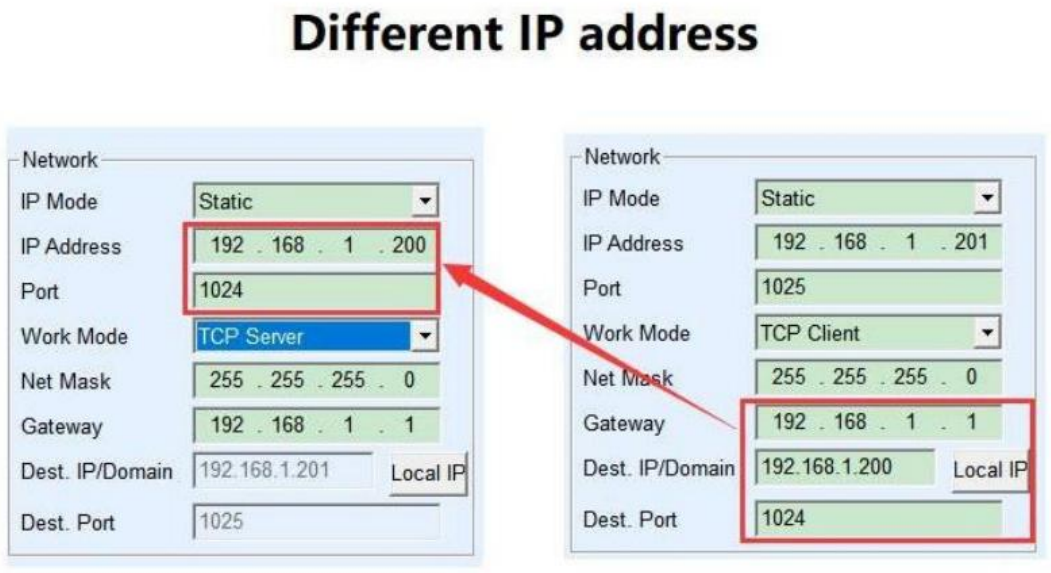


Figure 26 TCP device pairing parameter configuration

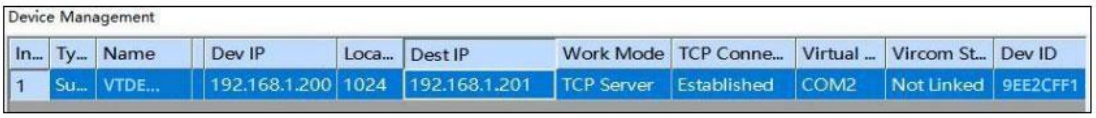


Figure 27 successful TCP device pairing check

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

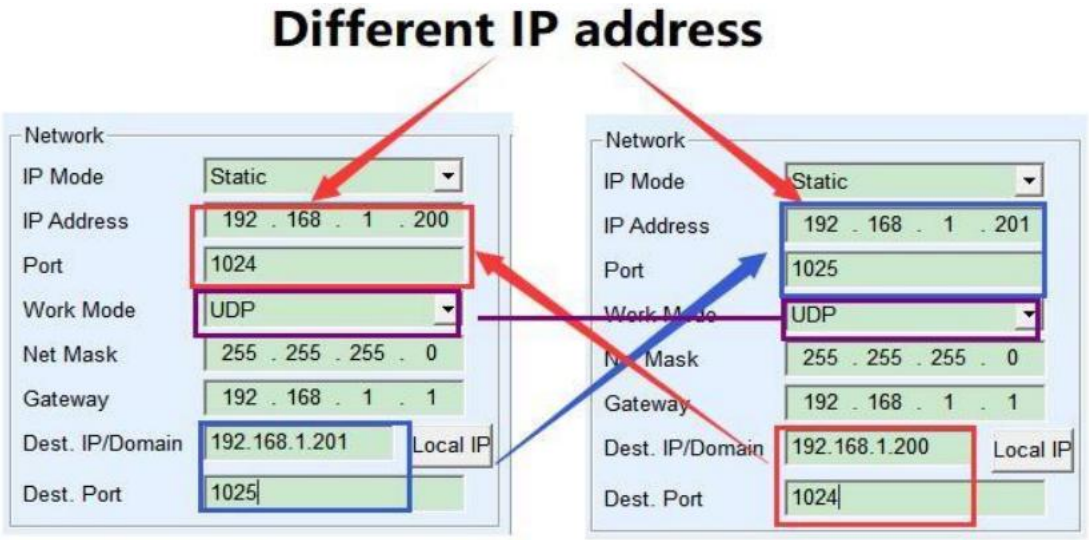


Figure 28 configuration of UDP device pairing parameters

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

# 6. Configure the communication mode

If you don't have an Internet cable and want to connect your device via WiFi, you need to do the following:

Turn on WLAN in the lower right corner of your computer：

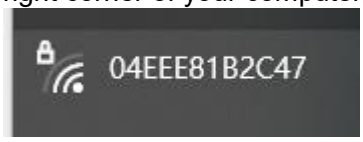


Figure 29 VT-DTU500 Hotspot name

Connect 04EEXXXXXXXX WiFi. The default password is 66666666.

If you have a network cable and want to connect the device through the cable, directly connect the network cable to the LAN port, open your browser, enter 192.168.8.1 in the address bar, press enter to confirm, you can open the route configuration page of VT-DTU500.

(Using a wired connection requires the same network segment as the computer's Ethernet and VT-DTU500)

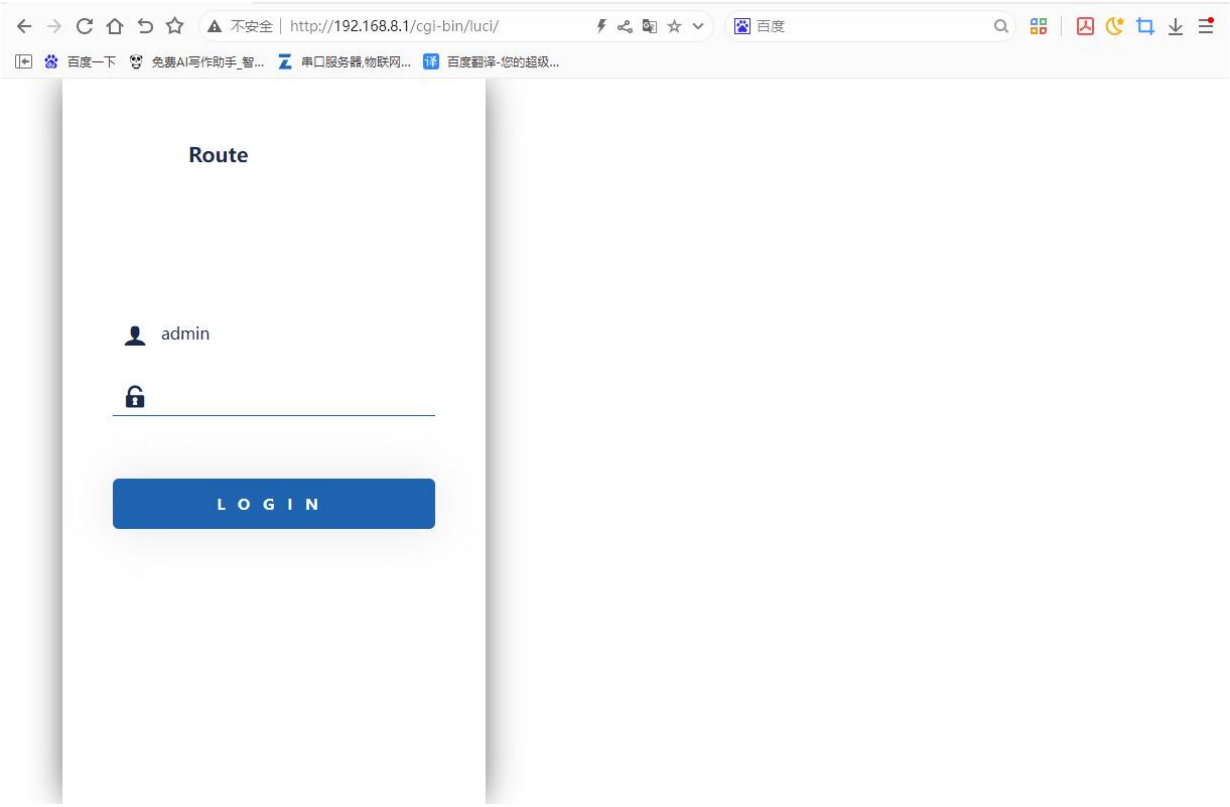


Figure 30 Communication method login page

The initial password is not required. You can directly click to log in. After login, the configuration page will be displayed：

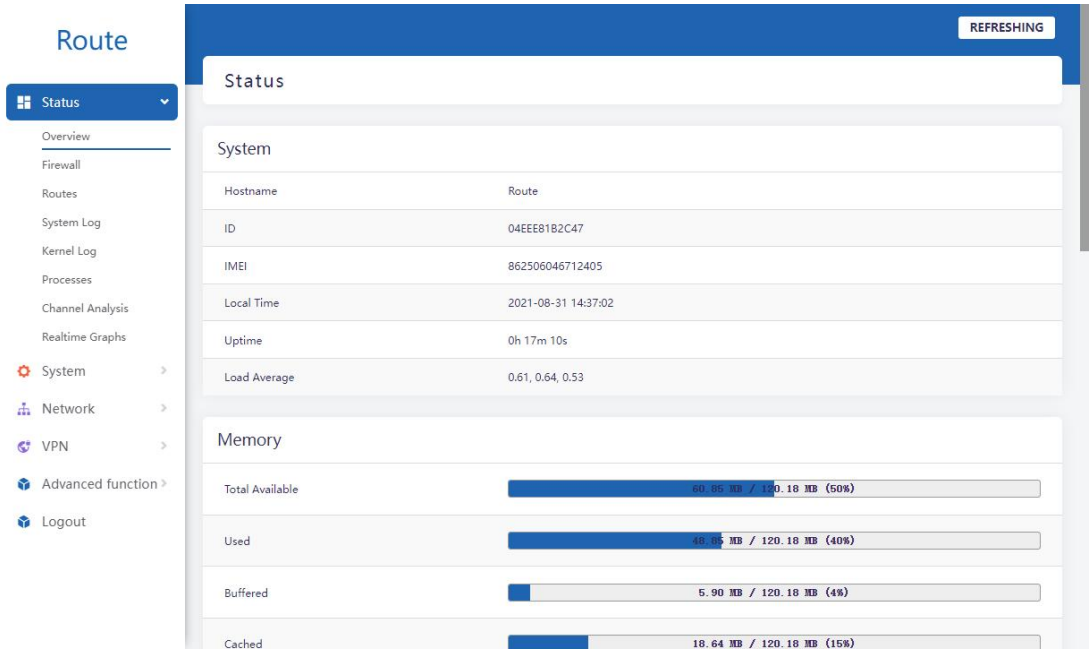


Figure 31 Communication mode configuration page

Click Network > Interface, you can see the existing interfaces of the device, mainly LAN port, WAN port, and WAN_4G port.

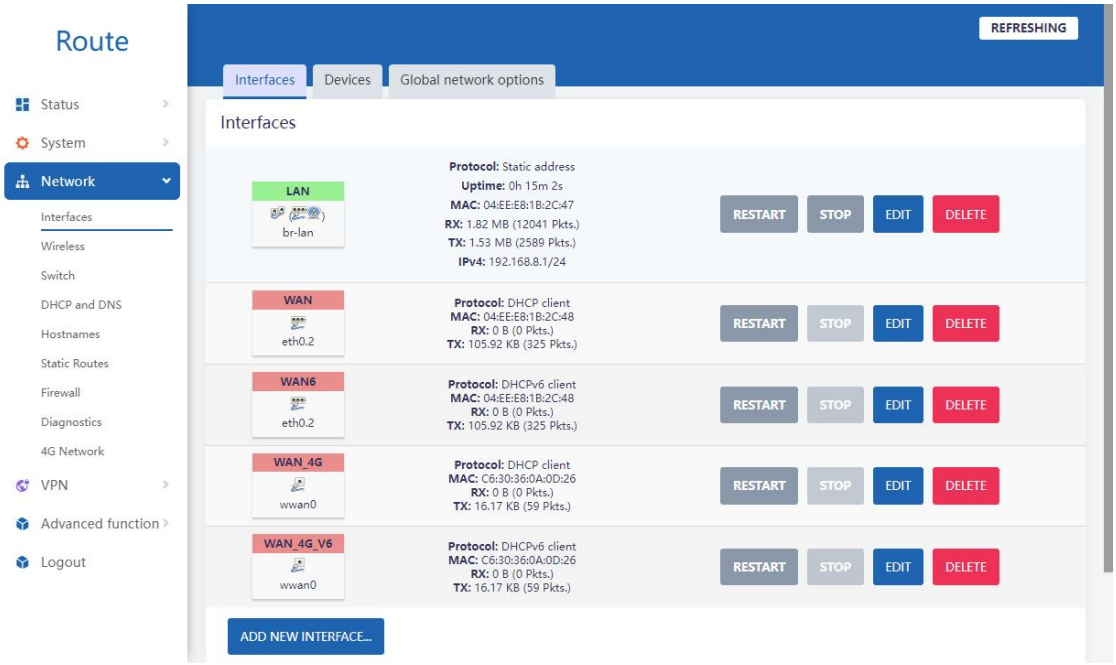


Figure 32 Interface page

Initial configuration: Click the menu bar on the left side of the web page: Network -- > Interface, you can see the interface page as shown above. WAN Port mode The default mode is Wired priority mode, that is, WAN ports access the Internet through network cables. Related parameters can be viewed by clicking Network -- >4G network.

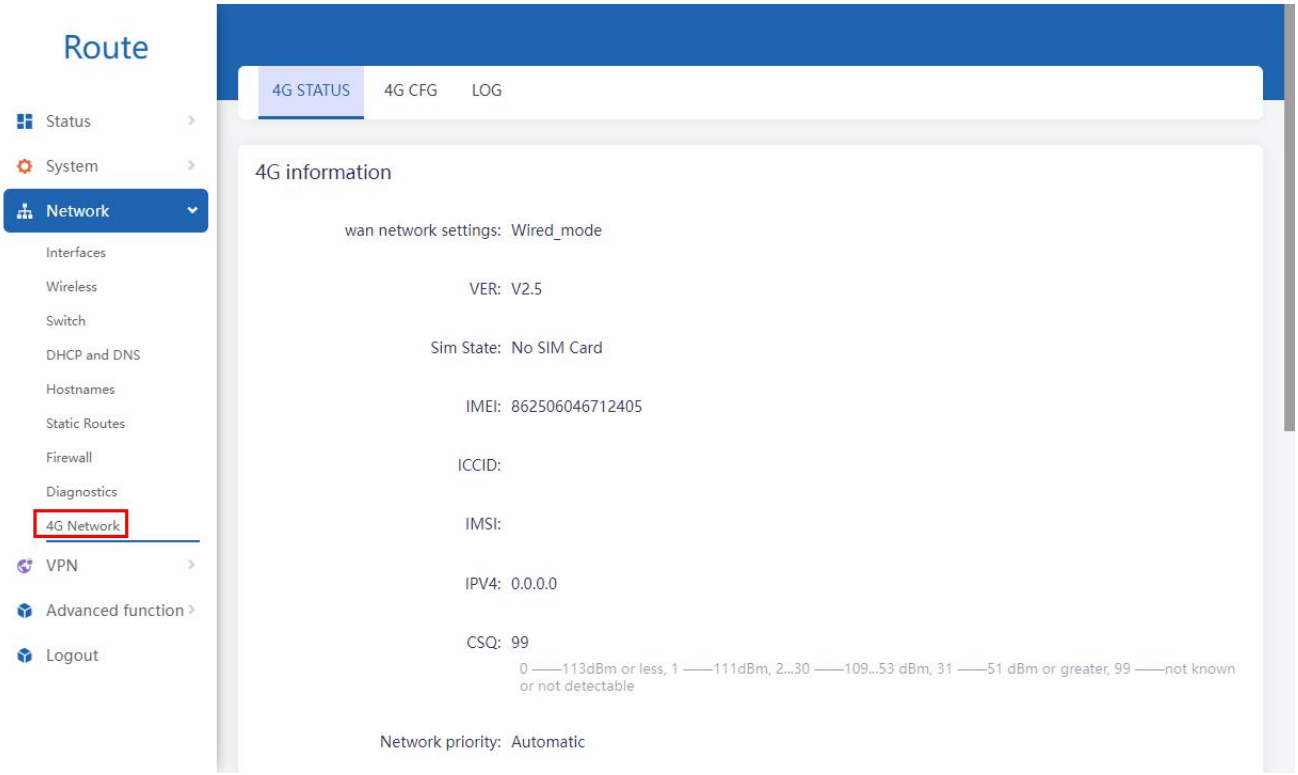


Figure 33 4G LAN status

## 6.1 Network cables connection

### 6.1.1. LAN port mode

If VT-DTU500 is used as a network port serial port server, you do not need to enter the route configuration page. Connect the LAN port of the VT-DTU500 to the company's local area network (LAN) using a crossover cable or a direct connection cable to communicate with other network devices in the LAN.

## 6.2 WiFi Connection

WiFi communication has two forms, the first WiFi relay: In the relay mode, the serial IP and LAN port of the VT-DTU500 device is in a sub-net under the local area network.

The second kind of WiFi bridge: the serial port IP of the VT-DTU500 will be in the same network segment as other devices and the parent network of the LAN port.

### 6.2.1 WiFi Relay mode

Click the menu bar on the left side of the web page: Network -- > Wireless, you can see the

wireless overview, VT-DTU500 device is equipped with a wireless network card in the 2.4G band.

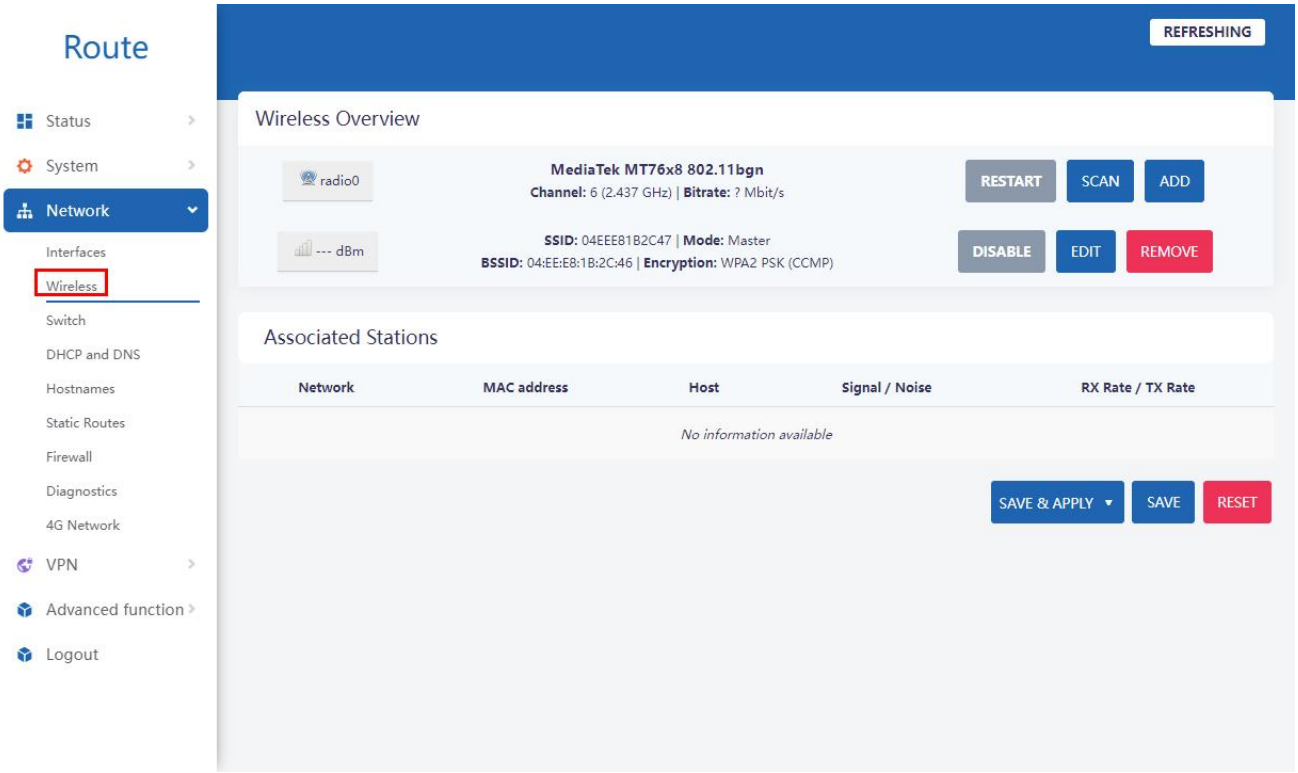


Figure 34 Wireless overview

WiFi relay mode, that is, the VT-DTU500 accesses the upper-level network through the upper-level WiFi. Enter the communication mode configuration page, click the left menu bar: Network -- >WiFi, and click the scan button on the right of the network card：

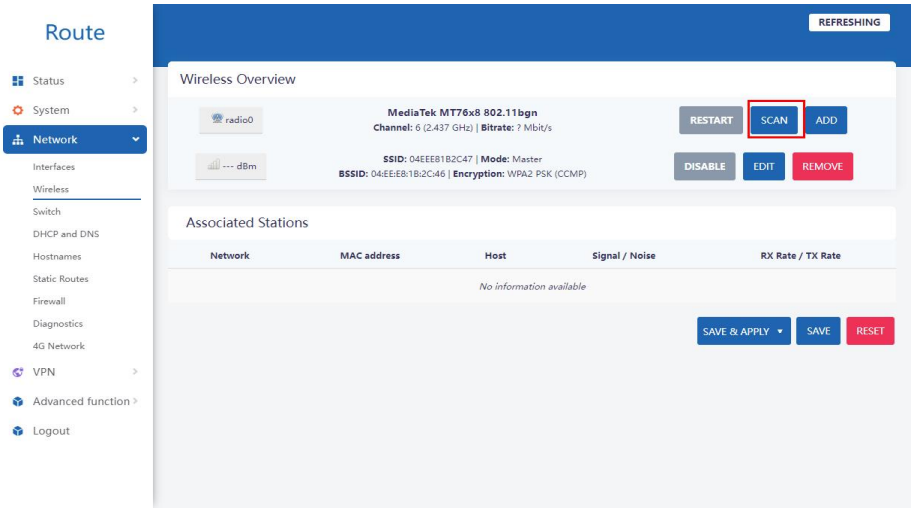


Figure 35 scan button

Step 2: Select the upper-level network you want to access on the open page and click Join Network.

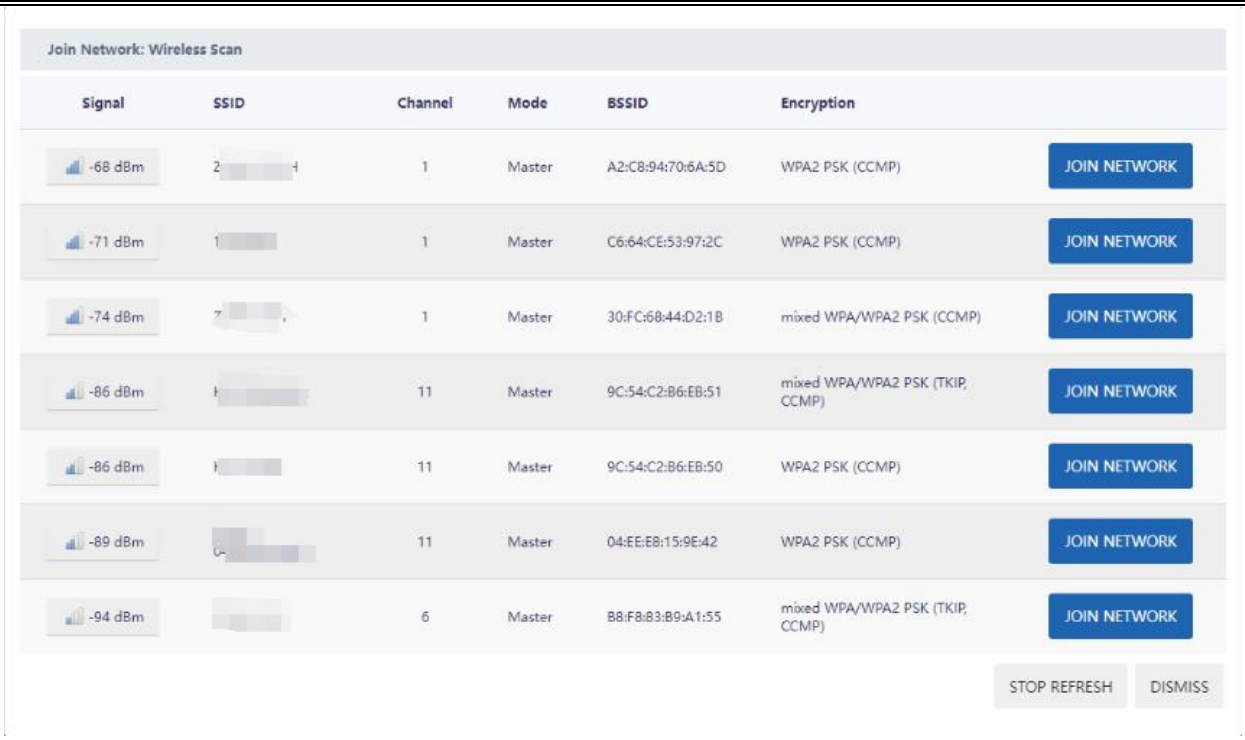


Figure 36 Joining the network

Step 3: Enter your upper-level network password on the open page, the default name of the new network interface is wwan, you can modify it yourself, and then click the Submit button in the lower right corner, the second interface will pop up.

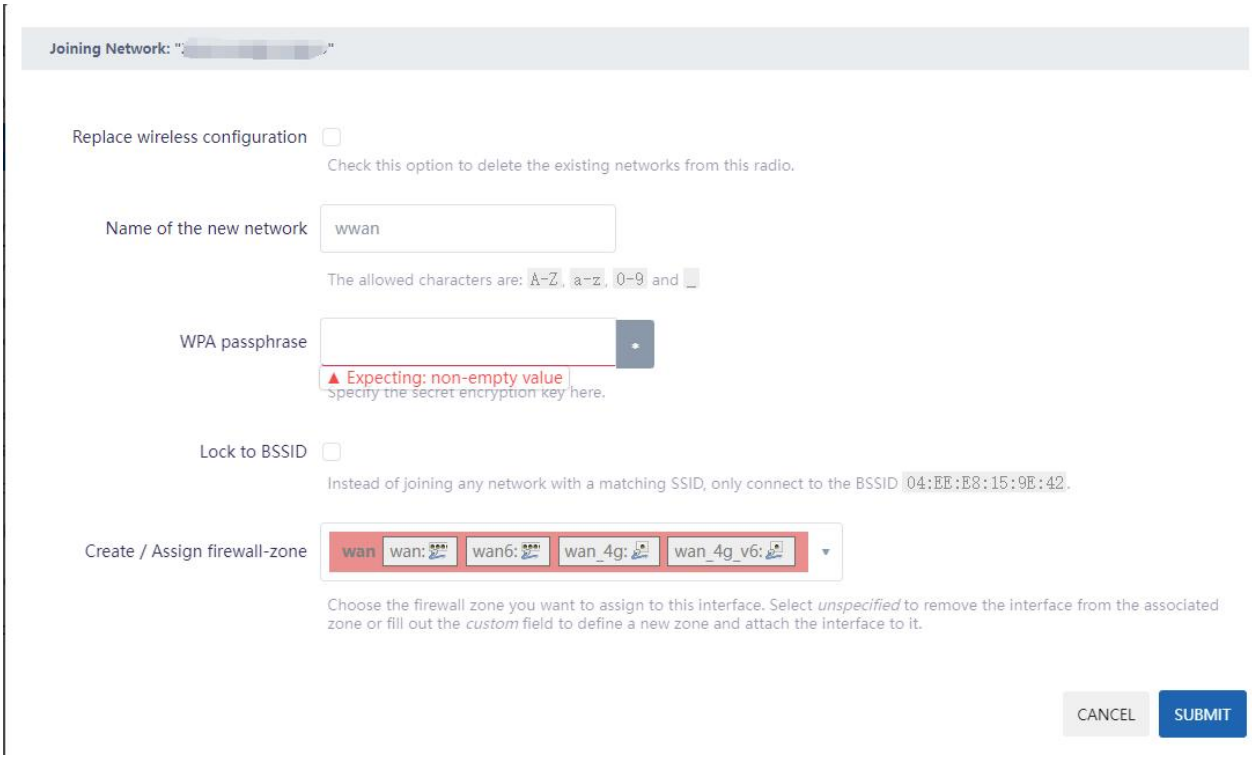


Figure 37 Example Modify the name of a network interface

The second page has operating frequency, transmission power and other options.If the WiFi version of the device to be connected is older and does not support 802.11/N, you can change the operating frequency to Legacy. Under normal circumstances, there is no need to set any parameters, just click Save.

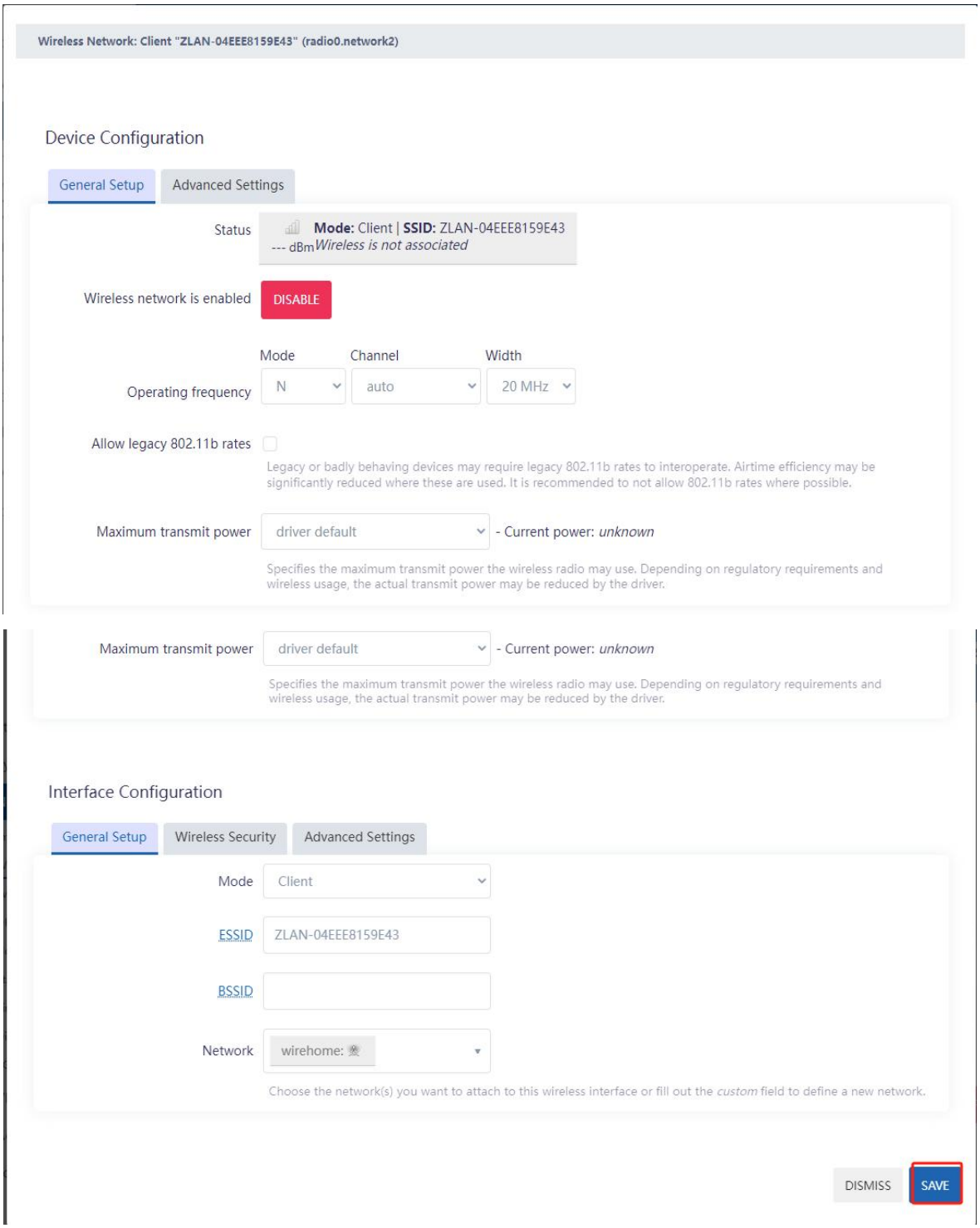


Figure 38 Interface configuration

Click Save and enter the page as shown in the following figure. You can see an additional mode in the wireless overview: Client wireless. The web page indicates that the interface has multiple unapplied changes. Click Save and apply them to take effect.

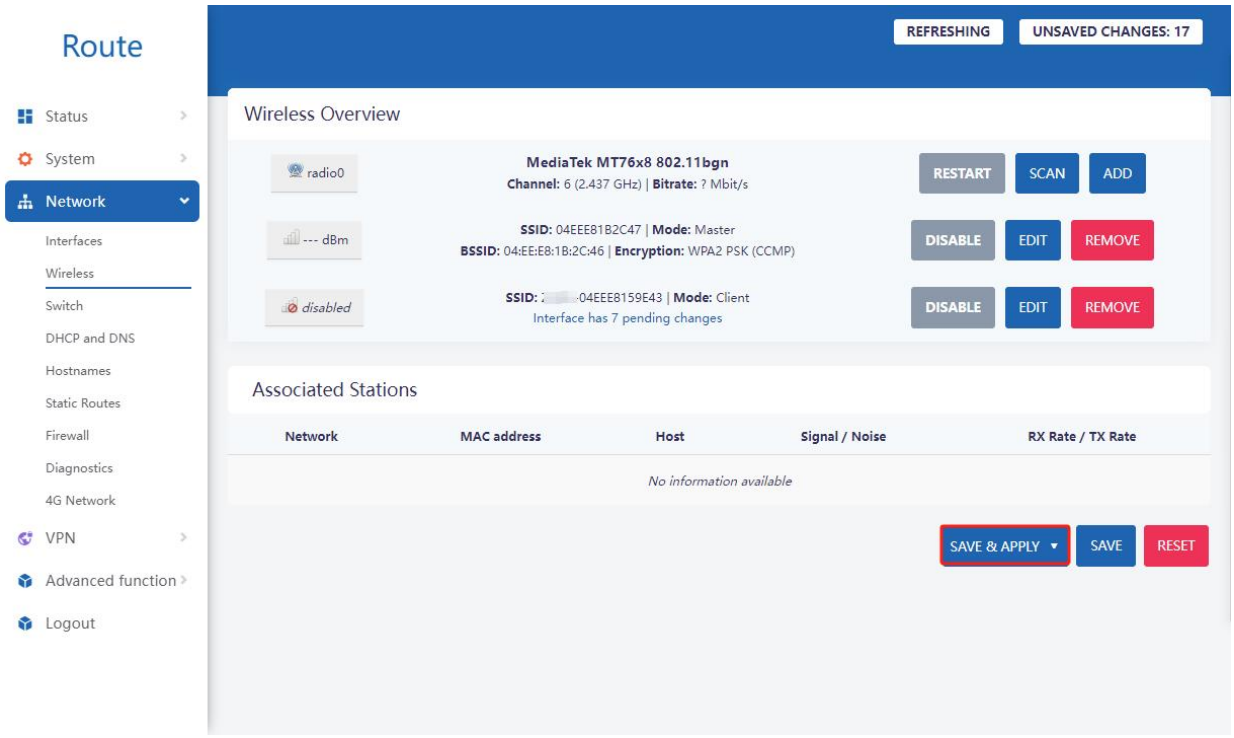


Figure 39 save button

Step 4: Click on the left menu bar: Network -- > Interface, then we can see the newly added interface.

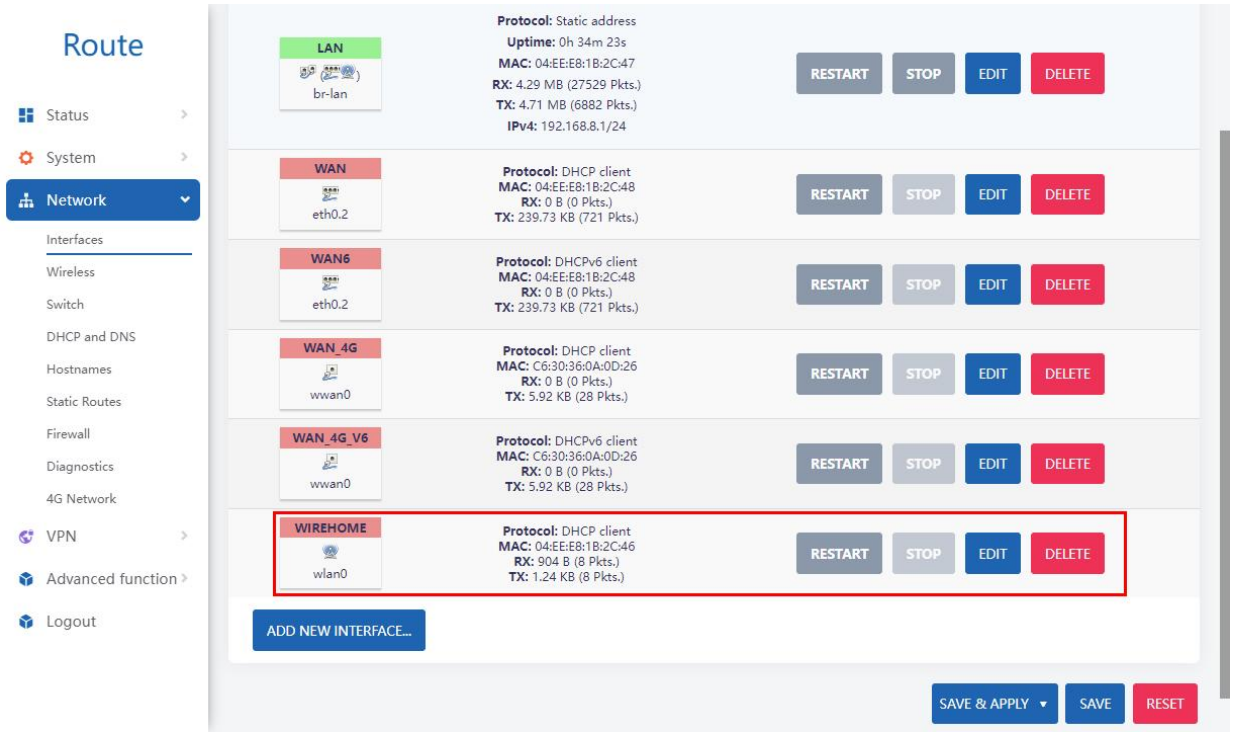


Figure 40 Interface page

Step 5: Click on the left menu bar: Network -- >4G Network -- >4G Settings: Set the WAN port mode to wired_mode: (if it is already there is no need to perform this step)

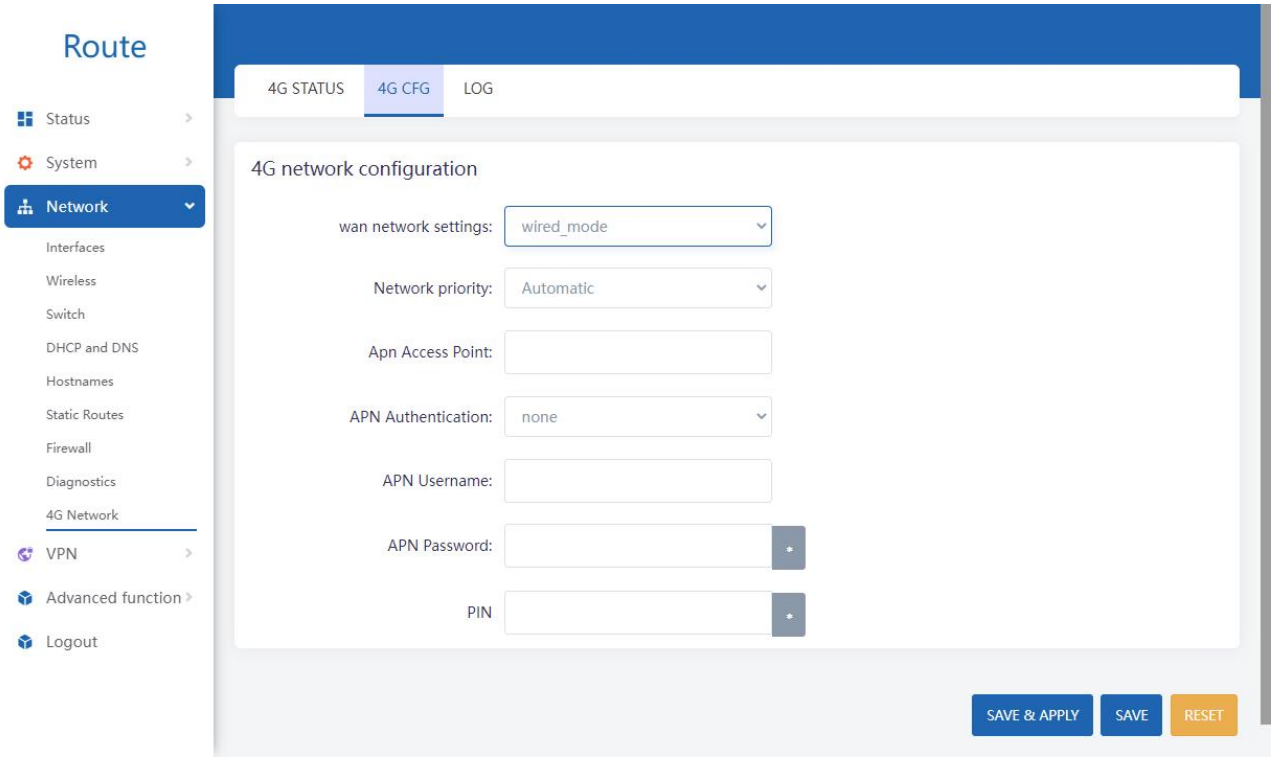


Figure 41 Wired mode

It is worth noting that the serial port IP of the VT-DTU500 is not in the same network segment as the WiFi upper-layer network.

### 6.2.2 WiFi bridge connection mode

If the serial port IP address of the VT-DTU500 and other LAN ports are in the same network segment as the upper-layer network, set the WiFi mode to bridge mode.

Click Advanced Function -> Trunk, select trunk bridging mode, select the name of the AP to be brided, enter the password of the AP for the upper-level WiFi, and select the corresponding encryption mode. The IP address of the device should be set to an IP address in a different network segment from the upper-level network port：

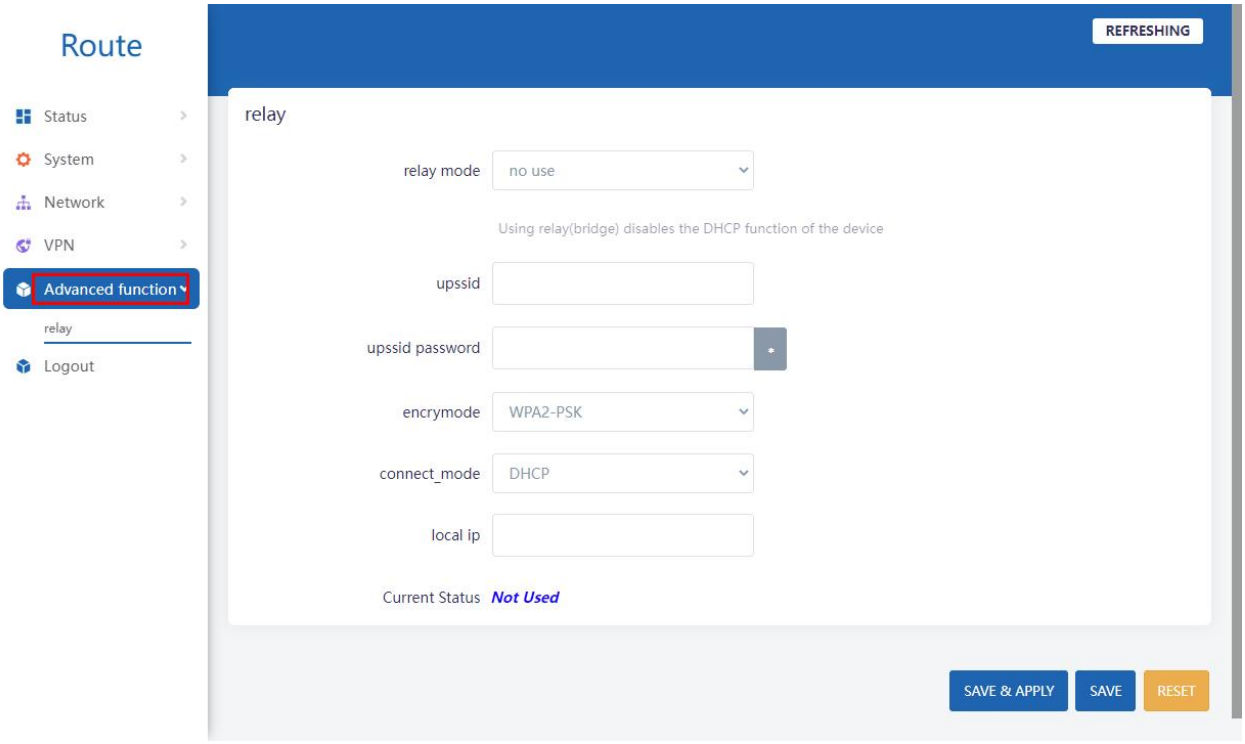


Figure 42 Advanced function

Step 2: Fill it out as required, click "Save and apply" in the lower right corner, and wait for the application to complete. After the relay is successful, menu bar: Network -> Wireless:

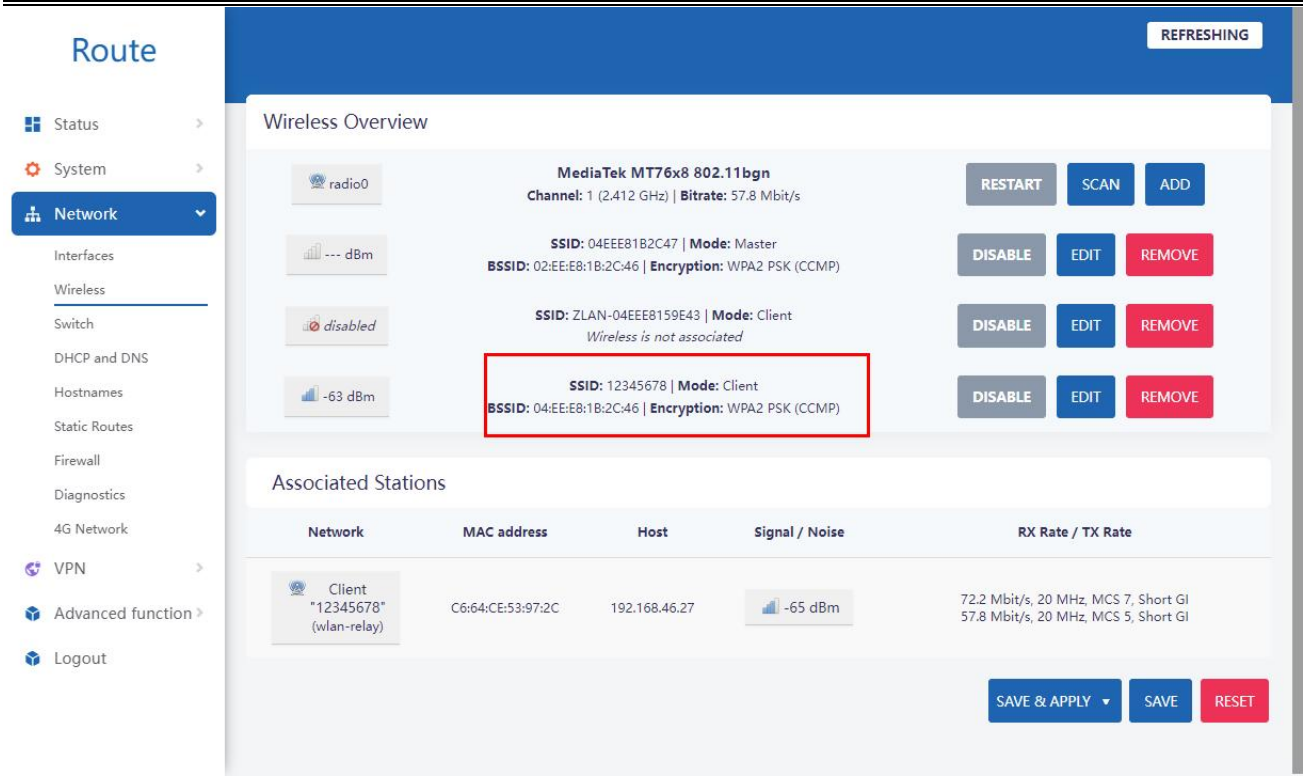


Figure 43 Relay wireless Overview

The upper-level WiFi is displayed here. If the encryption mode is also displayed, the connection to the upper-level WiFi is successful.

Then click on the menu bar: Network -> Interface. Check whether the relay interface has an IP address. If an IP address exists, the relay is successful.

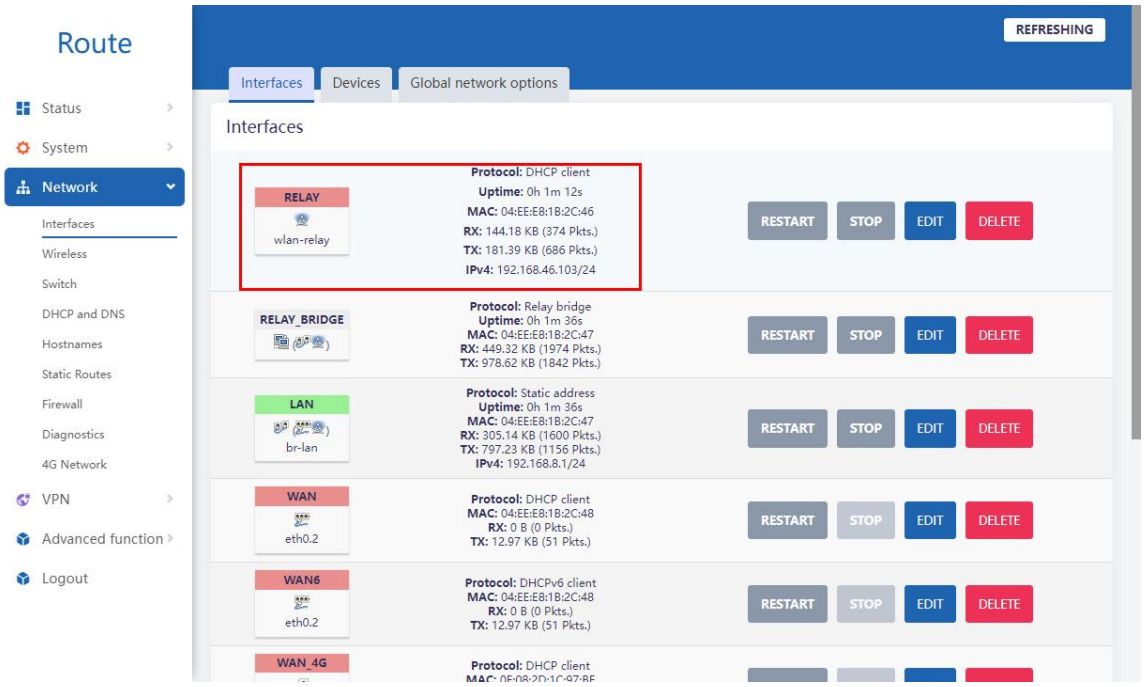


Figure 44 relay interface

After the configuration is complete, wait for 10 seconds. In this case, the VT-DTU500 can bridge to the upper-level AP over WiFi, and the serial port IP address of the VT-DTU500 can obtain the network segment assigned by the upper-level AP. It can communicate directly with the upper-layer network.

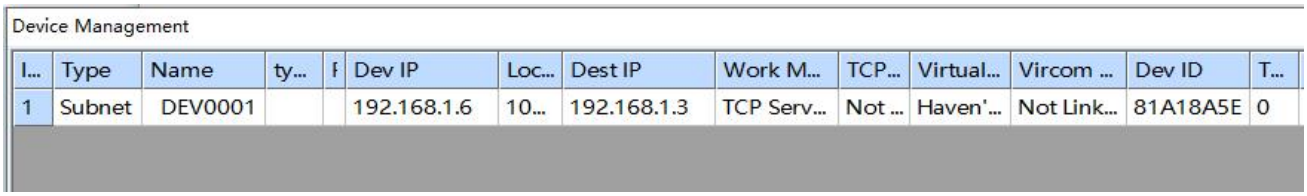


Figure 45 VT-DTU500 serial port IP

## 6.3 4G mode

In 4G mode, the VT-DTU500 accesses the external network by inserting a 4G SIM card.

4G mode supports APN (Access point name). Enter the APN name, authentication mode, user name, and password as required by the APN provider.

Step 1: Insert the SIM card and connect the 4G antenna.

Step 2: Click the left menu bar: Network -- >4G Network -- >4G Settings: Set the WAN port mode to 4G priority mode (if it is already, there is no need to perform this step) and click Save and set in the lower right corner.

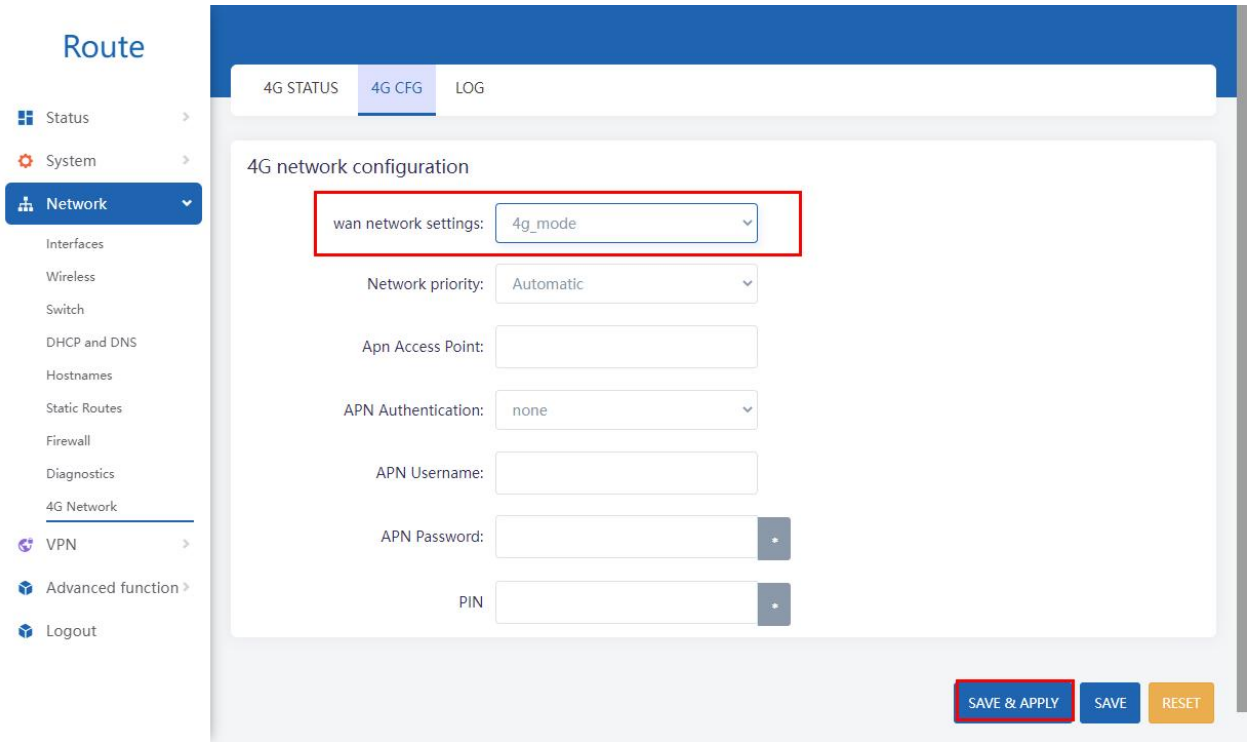


Figure 46 4G Priority mode

After completion, VT-DTU500 can access the external network via 4G, and the serial port IP of VT-DTU500 and other devices on the LAN port can access the Internet via 4G.

# 7. Modbus Advanced Features

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

## 7.1 Enable Modbus Gateway

First of all, the serial port server should support Modbus gateway, that is, the "Modbus TCP to RTU" function in Table 5 Device Supported Functions in the Device Settings dialog box should be checked.

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

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

## 7.2 Storage Modbus Gateway

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

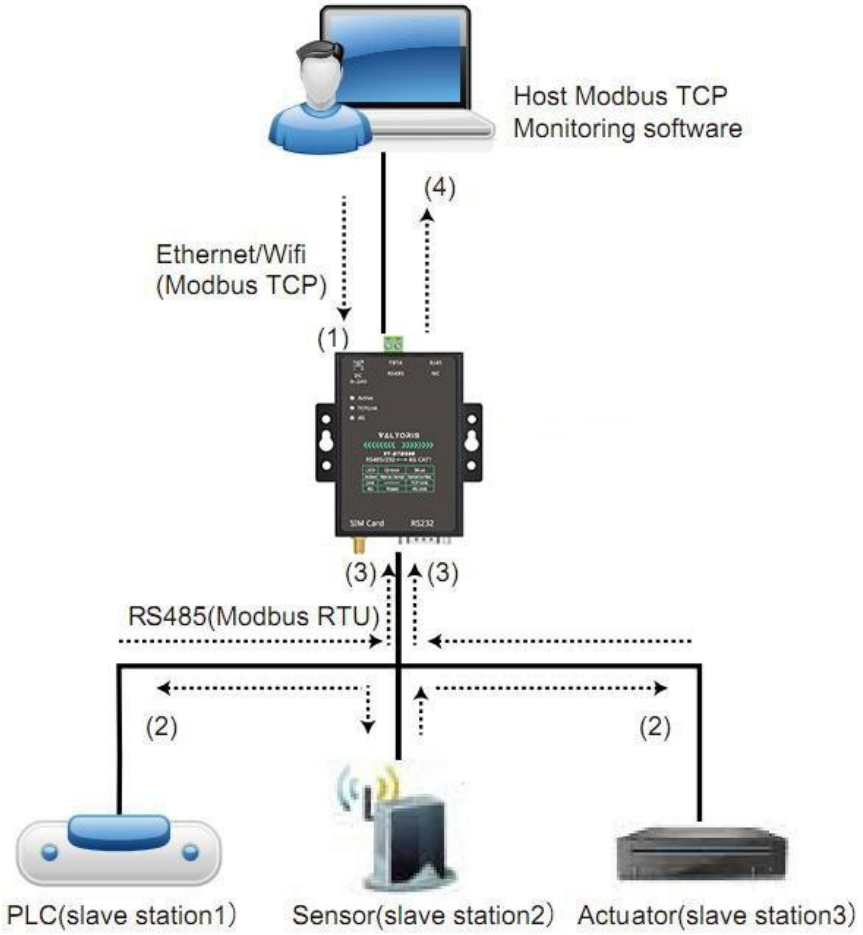


Figure 47 Storage Modbus Gateway

As shown in Figure 47: The direction of ordinary Modbus TCP data flow is （1）→ （2）→（3）→（ 4 ）. That is, the Modbus TCP command is first converted into the corresponding command of Modbus RTU, and then the device responds to the Modbus RTU command to the Modbus gateway, and then the Modbus gateway converts it into Modbus TCP again and sends it to the monitoring host computer.

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

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

serial port to automatically update the contents of the currently saved register data, and save a copy of the latest register value.

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

VT-DTU500 can response quickly when monitored by multiple computers. No matter what baud rate of the serial port is, it can generally response the data to the host within 3ms, which showed a good updating speed of serial port data.

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

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

The characteristics of storage Modbus are listed below:

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

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.

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

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

## 7.3 Disable Storage Functionality

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

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

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

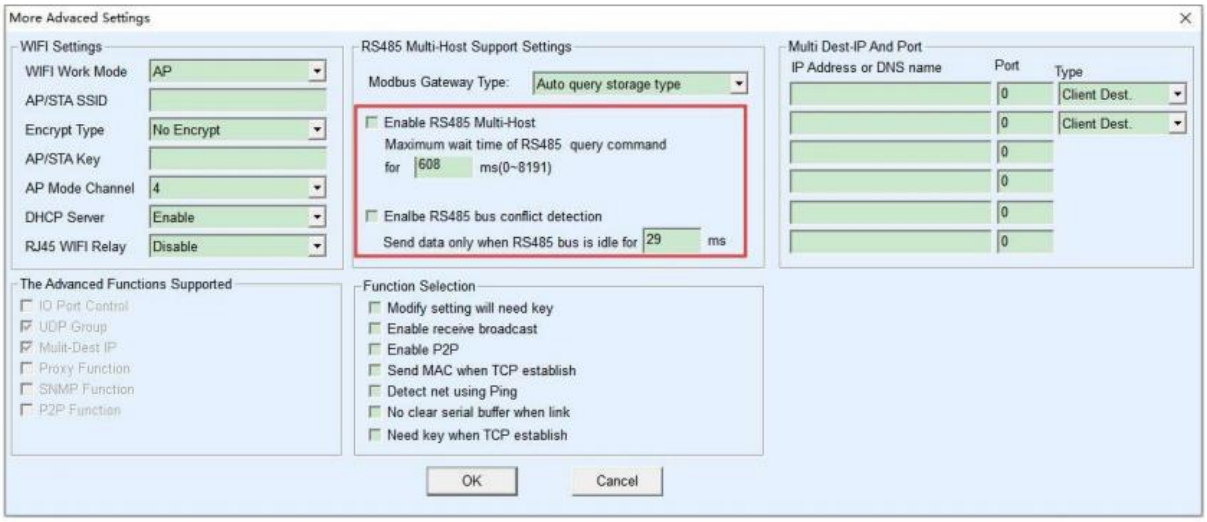


Figure 48 disable the storage function

## 7.4 Multi-host Function

As shown in Figure 48, "RS458 multi-host support" and "RS485 bus conflict detection function" are the multi-host functions. They are generally enabled and disabled at the same time. After enabling, the device with the conversion protocol of Modbus TCP has the function of storage-type Modbus gateway, otherwise it is a non-storage-type Modbus gateway; if the conversion protocol is None, the user-defined RS485 protocol can generally also have the function of serial port devices accessed by multiple hosts at the same time, which is impossible in a pure RS485 network, because multiple masters sending at the same time will cause conflicts on the RS485 bus. The multi-host of this serial port server can "coordinate" the RS485 bus to achieve the purpose of multi-host access.

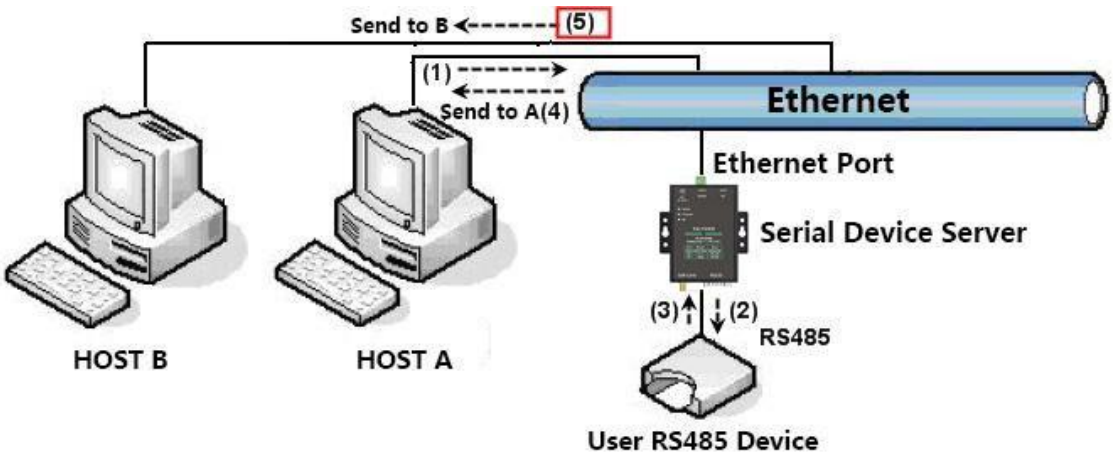


Figure 49 Multi-host function demonstration

As shown in Figure 49, in the normal mode, when two hosts: host A and host B are connected to the serial server at the same time, host A sends (1) command, RS485 device receives (2) command, RS485 device returns (3) ) command, but the serial port server will send (4) to host A and (5) to host B at the same time. Since the host B did not send the query, but it also received the response command (5), the host B may generate a communication abnormal error. In the multi-host mode, there will only be command (4) but not command (5), because the serial server will automatically remember the host that needs to be returned, and only return the instructions to the most recent communication host, and the query of host A will only reply to A, host B queries and replies to host B.

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

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

## 7.5 Multi-host Parameter

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

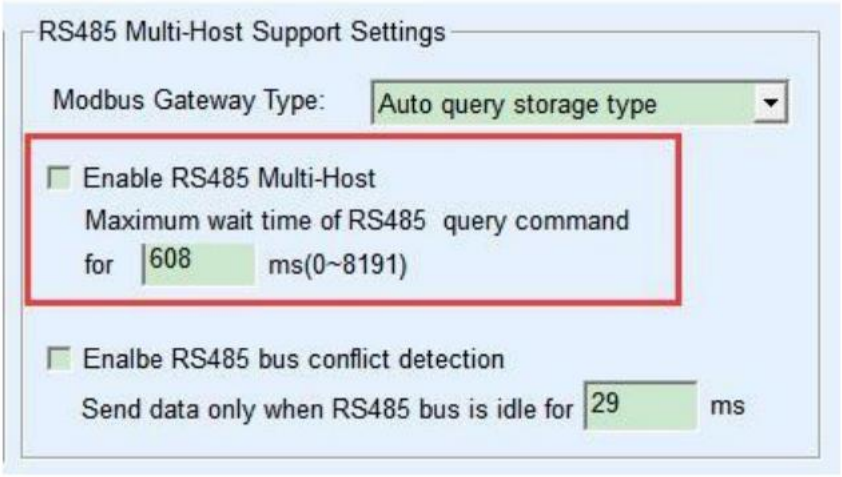


Figure 50 RS485 multi-host support

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

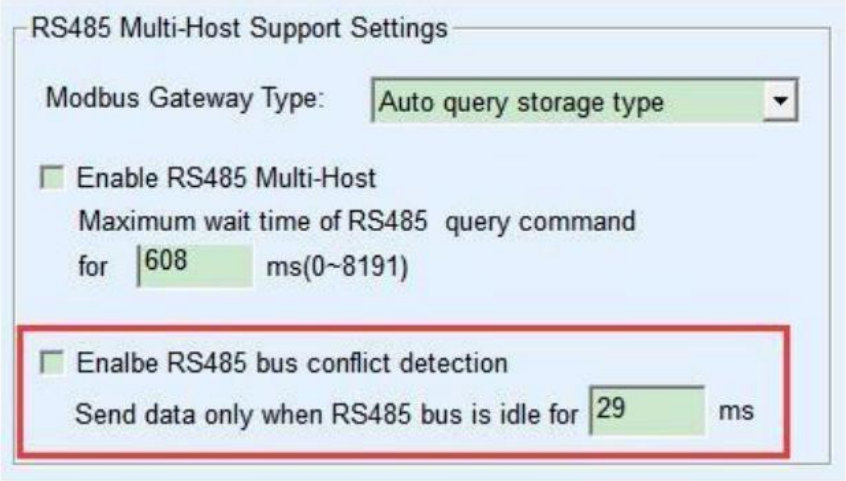


Figure 51 RS485 anti-collision idle time

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 51 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 50 shows the "RS485 command response timeout", which is generally determined according to the length of the back and forth response command. If the sending command is N bytes and the response is M bytes, the recommended value is: "Packet interval" ×(N+M+5)+100.

## 7.6 Modbus Under Multi-destination IP

As shown in Figure 49, if the serial device (RTU device) is used as the master station and the network port device (Modbus TCP device) is used as the slave station, and there are multiple network port slave devices at the same time. At this time, you can use the method described in 5.2.2 Client Connecting Multiple Servers to let the serial port server as a client connect to these multiple network port devices at the same time.

The function that needs to be realized at this time is: when the serial port RTU sends a command, it can be sent to multiple network port devices, the network port device recognizes whether to send it to

itself through the Slave ID field, and only the network port device corresponding to the Slave ID responds. After the network port response is sent to the serial server, it is converted into an RTU command and output from the serial port to the RTU device.

At this time, it should be noted that it is necessary to remove the two checkmarks shown in Figure 38 as "RS485 bus anti-collision time" and Figure 50 as "RS485 command response timeout time". Otherwise, the above forwarding function cannot be realized. Another application method is: Although the serial server is connected to multiple network port devices as a Client, the RTU device is not the master station, and the network port device still sends it first, and the RTU device responds (as a slave station). At this time, the "RS485 bus anti-collision time" and "RS485 command response timeout time" still need to be checked, so that multiple hosts can access one RTU device at the same time.

# 8. Router function

For routing functions of the VT-DTU500, please refer to the

**《VT-LTE400 User Manual-EN》**

# 9. After-sales service and Technical Support

**Phone/WhatsApp:** +86 18321985506

**Web:** www.valtoris.com

**Email:** support@valtoris.com