ZLAN5107I Optocoupler isolation type Serial port server /Modbus gateway

RS232/485/422 RTU TO TCP/MQTT/JSON/ModbusTCP

ZL DUI 20231123.1.0

The following modifications have been made to the document:

Change log

Date	Version number	Document number	Modify content.
2023-11-23	Rev.1	ZLDUI 20231123.1.0	Release version

CATALOGUE

1. Overview

ZLAN5107I is a specially designed RS232/485/422 device data collector/iot gateway for industrial environments. It integrates multiple functions such as a serial server, Modbus gateway, MQTT gateway, and RS232/485/422 to JSON conversion. It has three interfaces: RS232/485/422 (choose any one to use) and one Ethernet interface. It supports desktop screw hole installation and is easy to install. It adopts terminal block power supply /5.5MM round hole adapter power supply, with a wide voltage input of 9 to 24V. It has a higher cost performance compared with ordinary serial port servers.

Figure 1 ZLAN5107I

When used as a regular serial port server, connect the network port of the 5107I to the Ethernet. The upper computer software can send data to the serial port of the 5107I by using TCP/UDP transparent transmission or virtual serial port. The data

received on the serial port will also be transmitted to the software on the TCP end or the virtual serial port. In the virtual serial port mode, the serial port software for connecting the virtual serial port can be used without the need.

When used as a Modbus gateway, 5107I supports Modbus TCP to Modbus RTU conversion, thereby enabling the upper computer to collect data from RS232/485/422 Modbus RTU devices using the Modbus TCP protocol. On the contrary, if the serial port end serves as the master station, it can also be used. The 5107I supports relatively advanced Modbus gateway functions, including configurable Modus gateway ZLMB, storage-type Modbus gateway, etc., fully meeting the various configurations and usage requirements of Modbus gateways. Its supported multi-host function enables multiple computer masters to simultaneously access a slave device of an RS232/485/422 serial port.

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. Supported servers include Baidu Cloud MQTT, Alibaba Cloud MQTT, China Mobile OneNet platform, etc. Support parsing the collected Modbus RTU or non-standard serial port data into JSON format and encapsulating it in MQTT data packets for upload.

Supports uploading collected data in JSON format. Data is automatically collected. The collected data supports Modbus RTU, 645 instrument 97 version, 645 instrument 07 version, and various non-standard serial port protocols can be customized. Users can independently configure the uploaded data format and JSON keywords using ZLVircom. When uploading, it can support MQTT protocol, HTTP POST protocol, HTTP GET protocol, transparent transport protocol, and various non-standard network protocols can be customized.

The ZLAN5107I is equipped with various powerful functions of an iot gateway and is highly suitable for the industrial field to collect data from various serial port meters and sensors, including local network collection or autonomous collection and upload to cloud servers.

ZLAN5107I can be applied to:

- As an iot gateway, it serves as a communication bridge between devices and

the cloud.

- Electricity, intelligent meters and energy consumption monitoring;
- Remote monitoring and program download for various RS232/485/422 interface automated PLCS;
- Communication interfaces for various configuration software and devices;
- Networking of equipment in the field of access control and security;

A typical application connection is shown in Figure 2. The original serial port device (choose one of the three serial ports to use) is connected to the serial port of ZLAN5107I, and then connected to the computer via network cable. The software on the computer establishes a connection with 5107I via TCP/IP or virtual serial port. From then on, any data sent by the serial port device will be transparently transmitted to the computer's software, and the data sent by the software to the ZLAN5107I via the network will also be transparently transmitted to the serial port device.

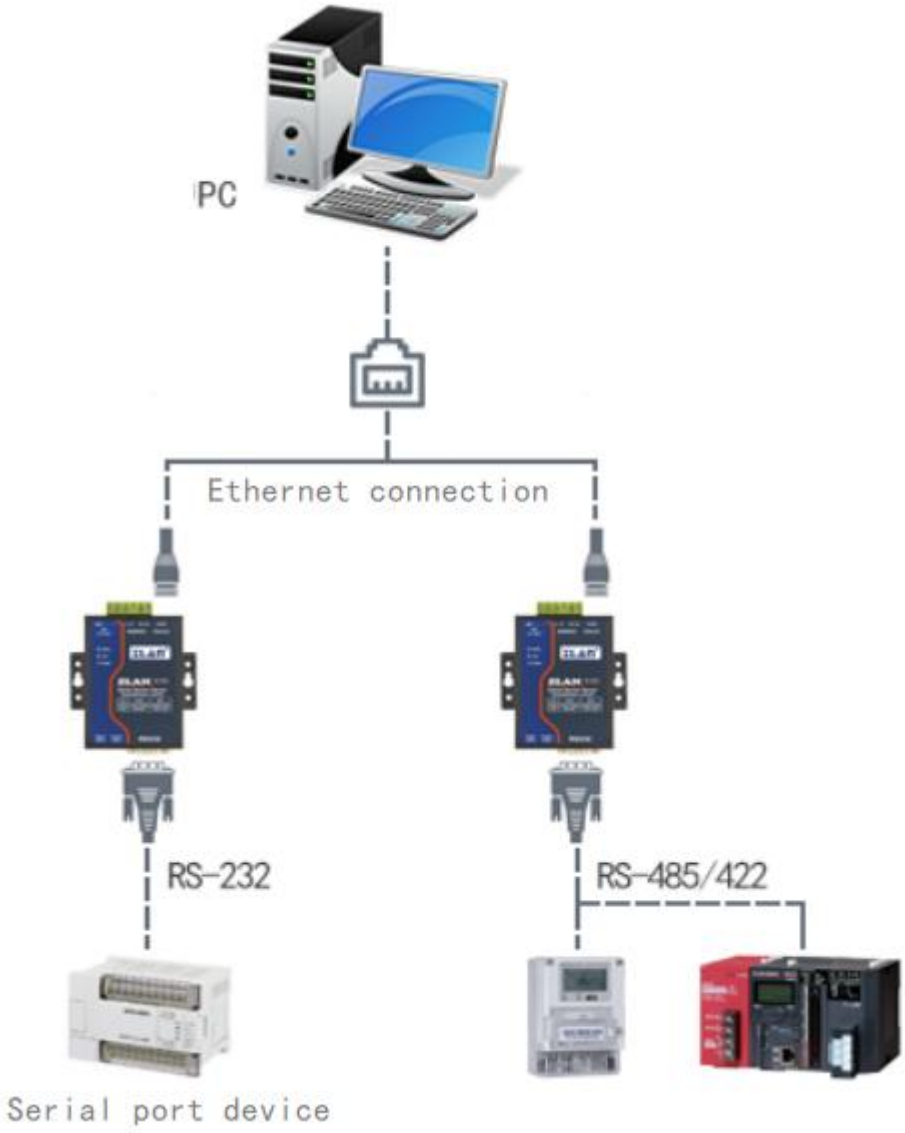

Figure 2 Connection diagram

2. Functional Features

2.1 Hardware Features

ZLAN5107I has the following features:

1. RS485/422 serial port optocoupler isolation, isolation voltage 1500V. It is suitable for strong interference environments where ordinary serial port servers cannot operate normally. Effectively isolate the ground circuit between serial port

devices and serial port servers.

2. Built-in power protection: Built-in varistors can effectively protect the internal circuits of the serial server in the event of power surges.
3. Industrial-grade power supply method: It features two power connection methods, namely terminal blocks and power plugs, making it convenient for use in industrial Settings. It can be powered by a wide voltage range of 9 to 24V.
4. It has three types of serial port interfaces: RS232/485/422. One can be chosen for use.
5. 2KV-level surge protection at the network port.
6. Rich panel indicator lights facilitate debugging: In terms of connection, there are not only LINK lights indicating whether the network cable is properly connected, but also LINK lights indicating the establishment of TCP connections. There are independent indicator lights for "serial port to network port" and "network port to serial port" in terms of data indicator lights.

2.2 Software Functions

1 Supports TCP server, TCP client, UDP mode, and UDP multicast. When serving as a TCP client, it also supports TCP server-side functions. As a TCP server, it supports 30 TCP connections, and as a TCP client, it supports 7 destination ips.

2 The baud rate supports 300 to 921600bps, the data bits support 7 to 8 bits, and the check bits can be in five ways: no check, odd check, even check, tag, and space.

3 Support the function of sending MAC addresses when devices are connected, facilitating cloud management of devices.

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

5 Supports Web browser configuration, supports dynamic acquisition of IP addresses through DHCP, and supports connection to domain name server addresses via DNS protocol.

6 Support remote cloud search for devices, configuration of device parameters, and device program upgrades.

7 Support remote viewing of the TCP connection status, serial port data sending and receiving status of the device through software. The virtual serial port supports data monitoring function.

2.3 Advanced Software Functions

5107I supports advanced software features, including:

1 Supports Modbus gateway function and Modbus RTU to Modbus TCP conversion. It can support storage-type Modbus, automatically collect device data and store it. It also supports Modbus gateways in non-storage mode; Support the Modbus gateway function of the ZLMB configurable table.

2 Support for multi-host function: Under the question-and-answer query mode, the network port end allows multiple computers to access the same serial port device simultaneously.

3 Support MQTT gateway function.

4 Supports JSON to Modbus RTU and 645 instrument protocol, and supports uploading data in HTTP POST and HTTP GET formats.

5 Supports custom heartbeat packets and registration packets: Facilitates communication with the cloud and device identification.

6 Support the function that requires password verification for establishing TCP connections to ensure connection security.

7 Supports http data submission and distribution functions. The cloud can directly interact with the serial port data of the device using http GET instructions.

3. Technical parameters

Table 1 Technical Parameters

Outline			
Interface：	485/422: Terminal blocks; 232:DB9		
Power supply：	5.5mm, positive inside and negative outside, standard power socket; Terminal block method		
Size：	L x W x H =9.4cm×5.5cm×2.5cm（±1mm）		
Communication			
Ethernet：	RJ45		
Serial port：	RS232/485/422x1:RXD，TXD，GND，CTS，RTS		
Serial port parameter			
Baud rate：	300~921600bps	Check bit：	None, Odd , even Mark, Space
Digit bits：	7～8bits	Flow control：	RTS/CTS，DTR/DCR, NONE
Software			
Protocol：	ETHERNET、IP、TCP、UDP、HTTP、ARP、ICMP、DHCP、DNS		
Configuration：	ZLVirCOM tool, WEB browser (can customize the web page), device management function library, serial port AT command configuration, device cloud management (ZLan server)		
communication method：	Socket、virtual serial port、Device management library		
Working mode			
TCP server, TCP client, UDP, Real Com Driver, Modbus TCP, Telnet			
Power supply：			
Power：	9～24V DC;		
Environment requests			
Operation temperature：	-40~85℃		
Storage temperature：	-45~120℃		

Humidity range：	5~95% humidity accordingly

4. Instructions for Use

3 Hardware Description

Figure 3 Front view

The front view of the ZLAN5107I serial port server is shown in the above picture. The shell is made of black radiation-resistant SECC metal shell.

1 **Power input**： You can choose either a DC socket or a terminal block input. Socket interface Outer diameter 5.5mm, inner diameter 2.1mm; Terminal block:

5.08mm terminal. The input voltage is 9 to 24V. 30mA@12V.

Figure 4: Network port, RS485/422, power supply

2 **RS485/422：** If only RS485 is used, connect TB and TA. Here, TB represents the 485 B line (negative line), and TA represents the 485 A line (positive line). If it is RS422, RA and RB also need to be connected, and the positive and negative wires of 422 should be connected. The 485 can support a load of 32 units. The maximum communication distance is 1,200 meters. Generally, it is only necessary to use a terminal resistor when the RS485 line exceeds 300 meters. The terminal resistor of 485 is 120 ohms.

3 **NET：** Connect the network cable and support automatic crossover.

4 **RS232：** The DB9 male head is adopted, and the pin sequence of RS232 is shown in Figure 5.

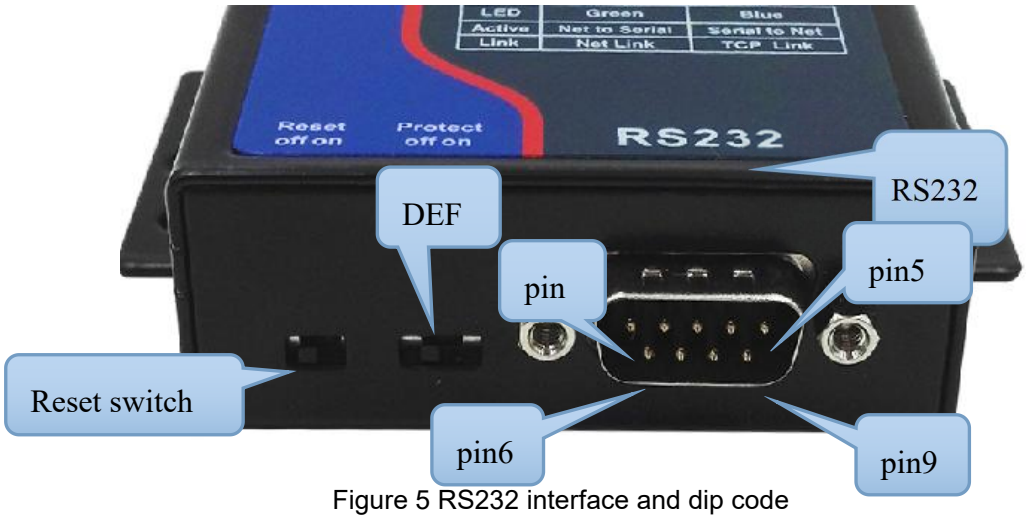

Figure 5 RS232 interface and dip code

The definition of the corresponding needle is shown in the following table:

Table 2 RS232 Line Sequence

Serial	Name	Input/Out	Function

number		put	
2	RXD	Input	The receiving pin of the serial server
3	TXD	Output	The serial port server sends pins
5	GND	/	Ground wire
4	DTR	Output	The same as the RTS function
6	DCR	Input	The same as CTS function
7	RTS	Output	After the hardware flow control is enabled, when the logic of this pin is 0 (greater than 3V), it indicates that the reception is ready. Otherwise, it indicates that more data cannot be received.
8	CTS	Input	After the hardware flow control is enabled, the serial server will detect this pin before sending data. If it is logical 0 (greater than 3V), data will be sent. Otherwise, stop sending data immediately.

5 **Indicator light**：It is divided into Power (PWR), Link(LNK), and Active(ACT) lights, representing power supply, connection indication, and data indication respectively.

6 **Reset**：Press and hold the reset button in the above picture for more than 5 seconds. The module will be reset to the IP address 192.168.1.254, and all downloaded configuration files will not be loaded.

Table 2 Meanings of Indicator Lights

Power	Power indicator light
Link	（1） Link is green when the network cable is properly connected. （2） When a TCP connection is established (or in UDP mode), the Link is blue (actually with a faint green light). It can be used to determine whether the serial port server establishes a communication link with the host software.
Active	（1） (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.
	（2）	When the serial port is sending data to the network port, the indicators are on both blue and green. 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.

The method of debugging communication using indicator lights

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

8 If the Link light is not blue (only considering TCP working mode), it indicates that the upper computer software has not established a connection with the serial port server. Please consider whether the IP address is configured in the same network segment.

9 If the Active light is green, it indicates that data is sent from the network port but no data is returned from the serial port device. Please check whether the baud rate is properly configured, whether the RS232 has poor contact, and whether the positive and negative terminals of the RS485 are reversed.

10 Installation Method: The serial port server can be installed on a flat surface or hung on a screw by using the "installation holes" shown in Figure 4. For applications with guide rails, it is recommended to purchase additional guide rail accessories to facilitate installation on the guide rails, as shown in Figure 6.

Figure 6: Back of the device

4.2 Hardware Connection

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

Among them, the power supply can adopt the configured 12V power adapter or the on-site 2-wire power supply, which can be directly connected to the positive and negative terminals of the power supply.

Among them, the serial port needs to be connected according to the user's serial port device. If the user has an RS232 device, it is necessary to consider whether it is a DB9 male or female connector. If it is a male connector, an RS232 male-to-female crossover cable can be used to connect the serial server and the device. Otherwise, a male-to-female RS232 straight-through cable should be used. If the user is an RS485 device, simply connect the positive 485 to TA and the negative 485 to TB.

The network port is connected to a regular network cable. It can be directly connected to a computer or connected to the network through a switch.

4.3 Software Installation

ZLVircom can be used to configure parameters such as device IP and create virtual serial ports. If the virtual serial port function is not required, the installation-free version can be downloaded. Download link:http://www.zlmcu.com/download.htm

Table 3 Versions of ZLVircom

Software name	Specifications
ZLVircom Device Management Tool (non-installed version)	The non-installed version does not include the virtual serial port function.
ZLVircom-Device Management Tool (Installation version)	The installed version contains ZLVircom_x64.msi and ZLVircom_x86.msi. 64-bit operating systems install x64, 16-bit operating systems install x86 versions.

Just follow the default prompts when installing. After the installation is completed, zlvircom will be launched each time the computer starts up, which is used to create a virtual serial port at startup.

4.4 Parameter Configuration

After ZLVircom is installed and the hardware connection of the device is completed, run the ZLvircom software as shown in the figure, and then click "Device Management" as shown in the figure. Using ZLVircom, you can search for and configure device parameters within different network segments, which is very convenient. As long as the device and the computer running ZLVircom are under the same switch, it can be done.

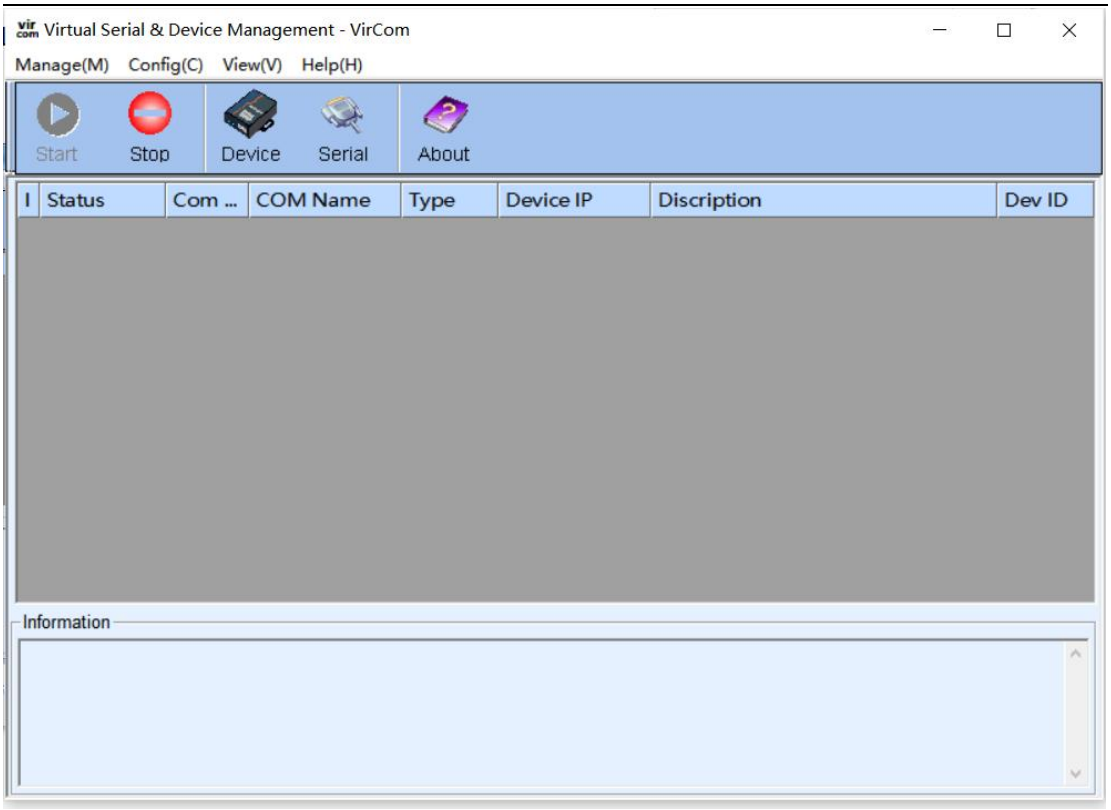

Figure 7 Main Interface of ZLVircom

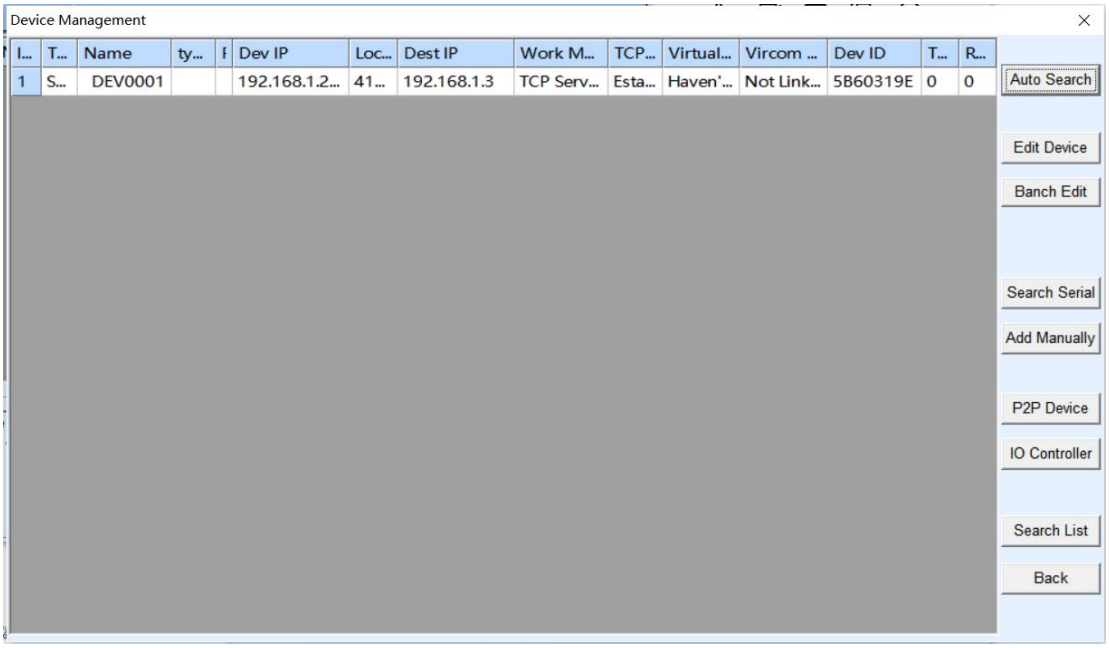

Figure 8 List of devices

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

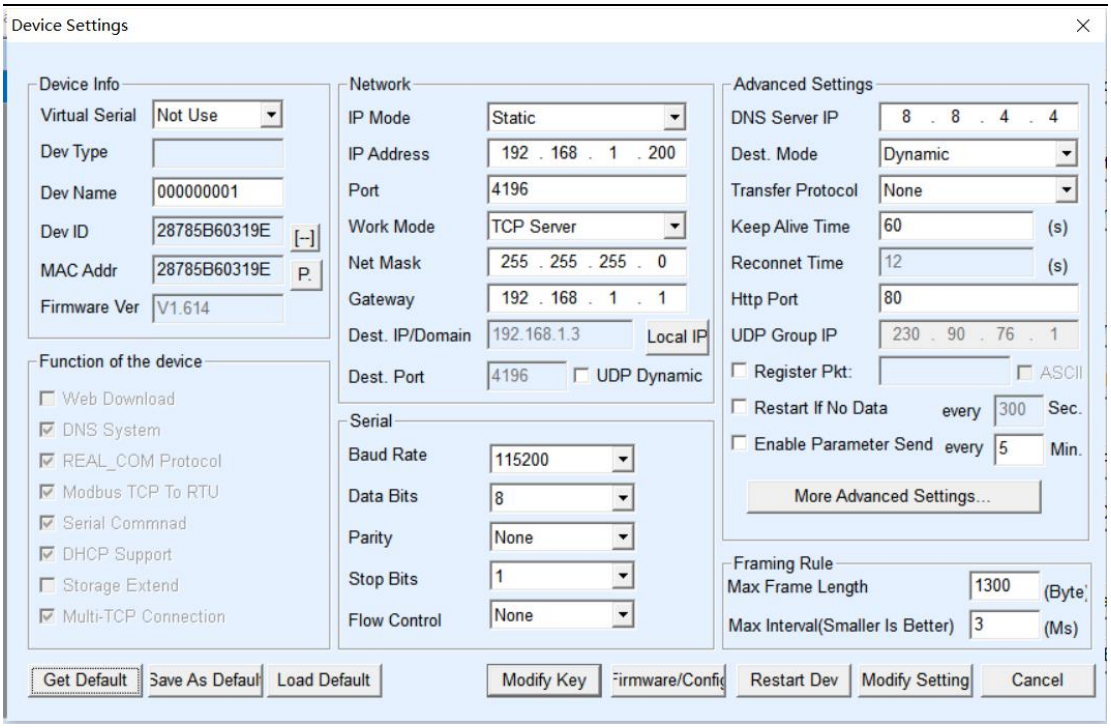

Figure 9 Equipment parameters

In this interface, users 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 in case of power failure. Meanwhile, the device will restart automatically.

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

The detailed meanings of other parameters are as follows:

Table 4 Parameter Meanings

Parameter name	value range	Contents
virtual serial port	none, created virtual serial port	You can bind the current device to an existing virtual serial port. Add a COM port in Serial Port Management on the home screen.
Device model		Only the model of the core module is displayed
Device name	random	You can give the device an easy-to-read name, up

		to 9 bytes, support Chinese names.
Device ID		factory unique ID, cannot be modified.
Firmware version		Firmware version of the core module
Functions supported by the device		See Table 3 for features supported by the device
IP mode	static、DHCP	Users can choose between static or DHCP (dynamic IP acquisition)
IP address		IP address of the serial port server
Interface	0～65535	Listening port of the serial port Server in TCP Server or UDP mode. If you use port 0 as the client, you are advised to set port 0 to improve the connection speed. If port 0 is used, the system randomly assigns a local port. The difference between this and non-zero port is: (1) When the local port is 0, a new TCP connection is established with the PC when the module restarts, and the old TCP connection may not be closed, and the device may have multiple fake connections. Generally, the host computer wants to close the old connection when the module restarts; Specifying a non-zero port closes the old connection. (2) If the local port is 0, the TCP connection takes a shorter time to re-establish. When the serial port server is in TCP client mode, it also acts as the TCP server to listen for incoming connections on the port. In this case, the local port number used by the TCP client to connect to the server is Port +1000.
Working mode	TCP server mode, TCP	When set to TCP server, the serial server waits for

	client mode, UDP mode, UDP multicast mode	the computer to connect. If TCP client is configured, the serial port server initiates a connection to the network server specified by the destination IP address.
Subnet mask	For eg.：255.255.255.0	The subnet mask must be the same as that of the local LAN.
Gateway	For eg.：192.168.1.1	It must be the same as the local LAN gateway
Destination IP address or domain name		In TCP client or UDP mode, data is sent to the computer indicated by the destination IP or domain name.
Destination port		In TCP client or UDP mode, data is sent to the destination port of the destination IP address.
Baud rate	300、600、1200、2400、4800 、 7200 、 9600 、14400、19200、28800、38400、57600、76800、115200 、 230400 、460800、921.6K	Serial port baud rate
Digit bits	5、6、7、8、9	
Check bits	None, Even, Odd, tag, 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 available for RS232 serial port
DNS server		If the destination IP address is described by a domain name, enter the IP address of the DNS server. If the IP address mode is DHCP, you do not

		need to specify the DNS server. The DNS server automatically obtains the IP address from the DHCP server.
Destination mode	Static , dynamic	TCP client mode: In static destination mode, the device automatically restarts after five consecutive failed attempts to connect to the server.
Transfer protocol	NONE 、 Modbus TCP<->RTU 、 Real_COM、TELNET	NONE indicates that data is transmitted transparently from the serial port to the network. Modbus TCP<->RTU will convert Modbus TCP protocol directly into RTU protocol, which is convenient to cooperate with Modbus TCP protocol; RealCOM is designed to be compatible with the older version of the REAL_COM protocol. It is a virtual serial port protocol. However, it is not necessary to select the RealCom protocol when using the virtual serial port. The TELNET protocol allows the network to log in to our device through TELNET to communicate with the serial port
Keepalive timing time	0～255	Heartbeat interval. (1) If the value ranges from 1 to 255 and the device is in TCP client working mode, the device automatically sends TCP heartbeat packets at Keepalive intervals. This ensures the TCP validity of the link. If the value is set to 0, there is no TCP heartbeat. (2) If the value is set to 0 to 254, and the conversion protocol is REAL_COM, the device will send data with length 1 and content 0 at keepalive intervals to implement the heartbeat mechanism in the Realcom protocol. If the value is set to 255, there is no realcom heartbeat. (3) When the value is set to 0 to 254, if

		the device works on the TCP client, the device will send device parameters to the destination computer at keepalive intervals. If the value is set to 255, no parameter is sent, enabling remote device management.
Disconnected reconnection time	0～255	In TCP client mode, when the connection fails, the TCP connection is re-initiated to the computer at disconnection Reconnection time intervals. The value ranges from 0 to 254 seconds. If the value is set to 255, the reconnection is never performed. Note that the first TCP connection (such as hardware power-on, device restart through zlvircom software, and no data light) is generally carried out immediately, and only after the first connection fails will it wait for the "disconnection reconnection time" to try again, so the "disconnection reconnection time" will not affect the normal connection establishment time between the network and the server.
Web access port	1～65535	Default is 80
Multicast address		Under UDP multicast
Enable registration package		When a TCP connection is established, the registration packet is sent to the computer. The realcom protocol must be selected after the registration package is enabled. TCP server and TCP client modes are supported。
Digit packet length	1~1400	One of the serial port framing rules. Serial port server After receiving data of this length, the serial port sends the received data to the network as one

		frame.
Packet interval	0～255	Serial frame rule 2. When the data received by the serial port server stops for a period longer than the specified period, the received data is sent to the network as a frame。

The functions supported by the device are described as followings：

Figure 5 Device supported functions

Name	Specifications
Domain name system	The destination IP address can be a domain name (such as the first www server address).
REAL_COM protocol	A non-transparent serial port server protocol, suitable for multiple serial port servers to bind virtual serial ports over the Internet. Because the protocol contains the MAC address of the device, it helps the host computer to identify the device. Generally, it can not be used.
Modbus TCP to RTU	Can realize Modbus TCP to RTU. It also supports the multi-host function.
Serial port modification parameter	Support serial port class AT instruction to configure and read device parameters.
Automatic acquisition IP	Support DHCP client terminal protocol
Multi TCP connection	The TCP server supports more than one TCP connection.
UDP broadcast	UDP broadcast
Multi purpose IP	As a TCP client, seven destination IP addresses can be connected at the same time.
P2P function	Supports the function of accessing devices in any network through P2P traversing technology. Models ending in N support this function.

4.5 TCP Communication Test

After configuring the device parameters, you can use the serial port tool or TCP debugging tool to conduct TCP connection communication tests.

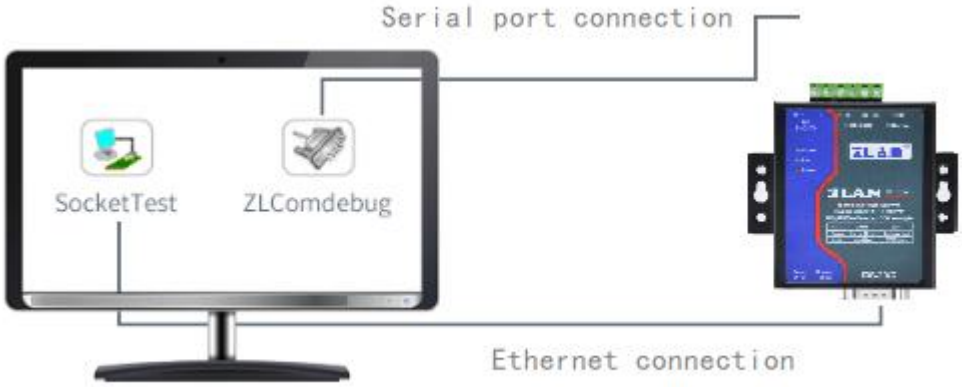

Figure 10 Schematic diagram of TCP communication

Assuming the PC COM port (USB turn RS232/485 lines), and serial port serial port server connection, then open the ZLComDebug (http://www.zlmcu.com/download/Comdebug.rar) serial debugging assistant, and open the corresponding COM figure 11; Open TCP&UDP debugging assistant SocketTest (http://www.zlmcu.com/download/SocketTest.rar), and as a way of TCP client, fill in the destination IP for a serial port server IP (currently 192.168.1.200), The destination port is 4196, and then click the "Open" button. Enter "socket send" in SocketTest and click send. Then the data will be transferred from the network port of the serial server to the serial port, and then sent to ZLComDebug, and then displayed in ZLComDebug. Conversely, enter "Comdebug send" in ZLComDebug, click Send, and it can also be sent to socket test and displayed.

The serial port to network port and network port to serial port transparent data forwarding function of the serial port server.

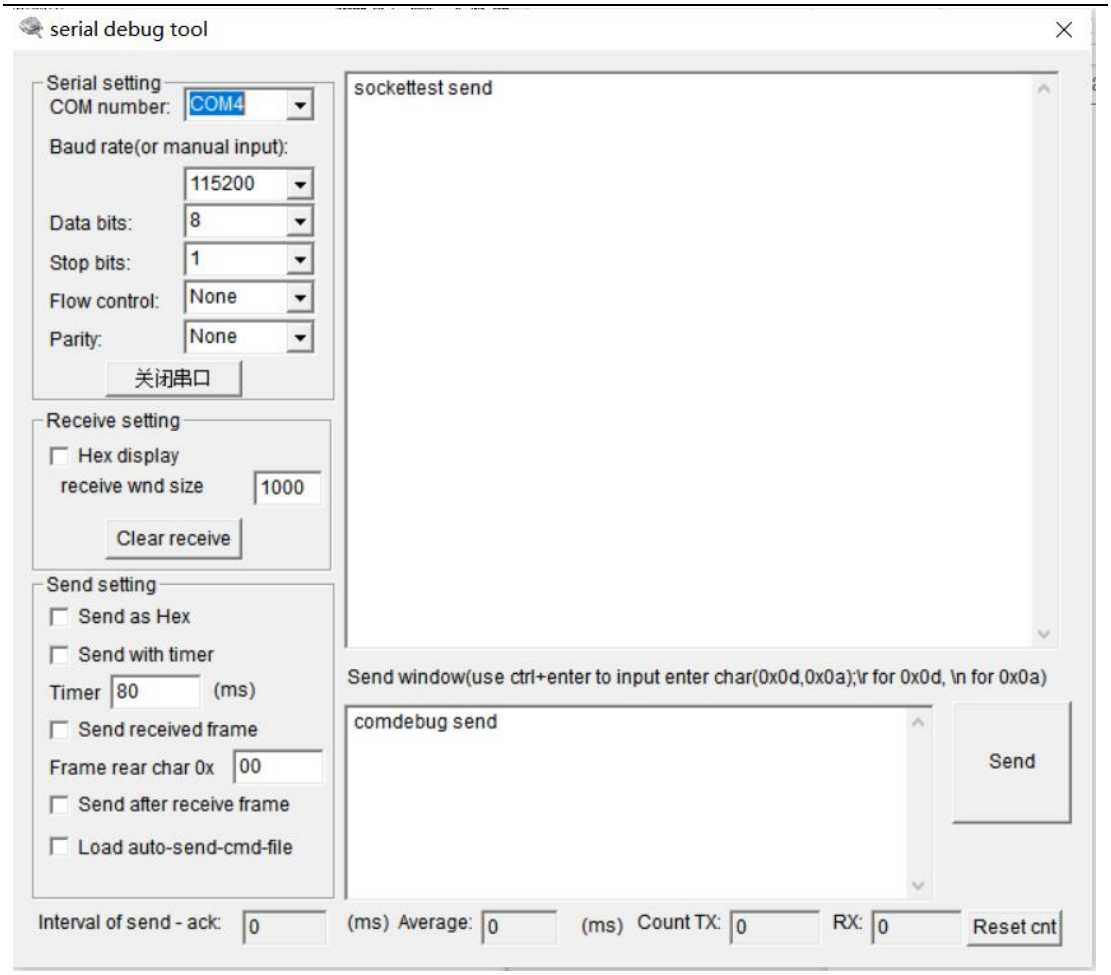

Figure 11 comdebug transceiver interface

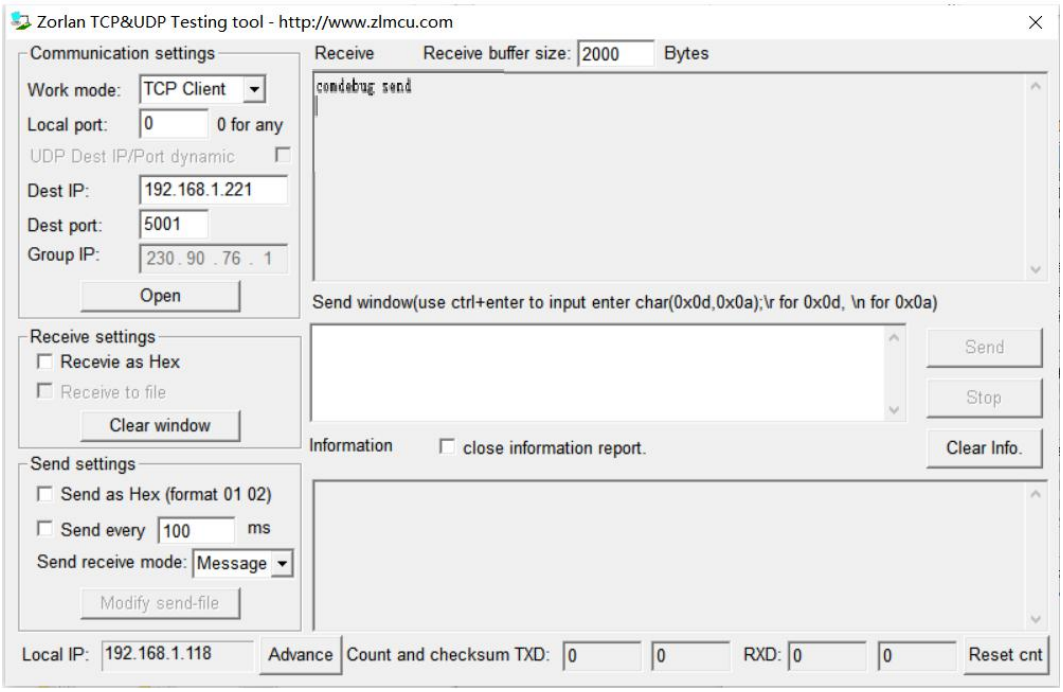

Figure 12 sockettest transceiver interface

4.6 Virtual Serial Port Testing

In Figure 10, SocketTest communicates directly with the serial port server via TCP. To enable the user's already developed serial port software to also communicate with the serial port server, a virtual serial port needs to be added between the user program and the serial port server. As shown in the figure, ZLVircom and the user program are running on the same computer. ZLVircom virtualizes a COM port and makes this COM port correspond to this serial server. When the user program enables COM communication, it can be sent to the user's serial port device through the ZLVircom serial port server. The following demonstrates the operation steps:

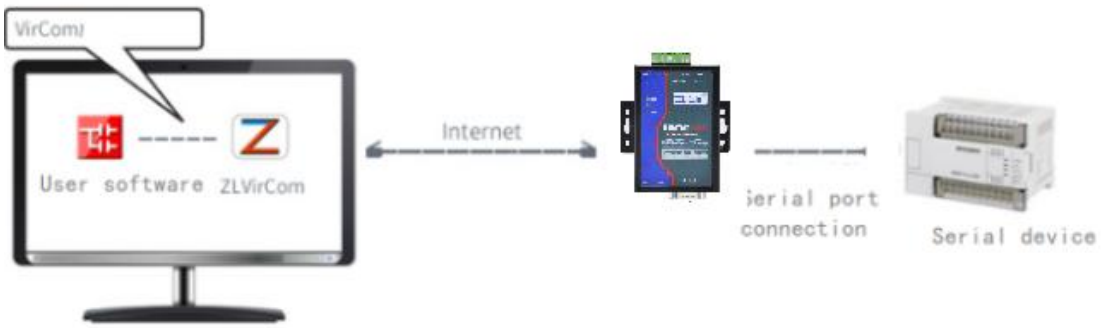

Figure 13 The Function of the virtual serial port

Click on "Serial Port Management" on the main interface of ZLVircom, then click "Add", and select "Add COM5", where COM5 is the COM port that did not originally exist on the computer.

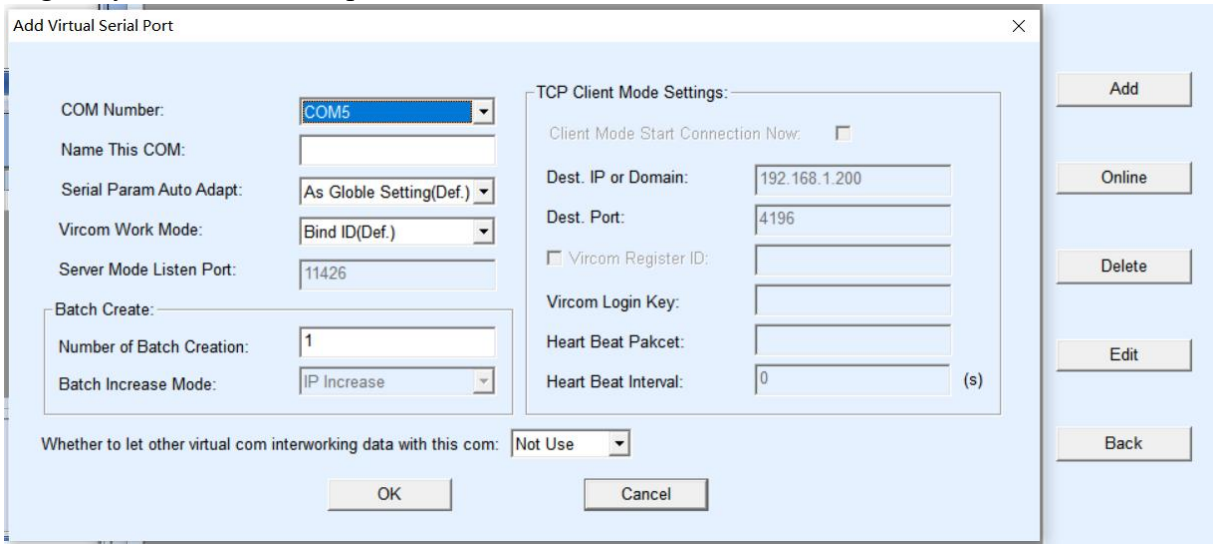

Figure 14 Adds a virtual serial port

Then enter Device Management and double-click the device that needs to be bound to COM5. As shown in the figure, select COM5 from the "Virtual Serial Port" list in the upper left corner. Then click "Modify Settings". And return to the main interface of ZLVircom. It can be seen that COM5 has been connected to the device with IP address 192.168.1.200. At this point, COM5 can be used instead of SocketTest for communication.

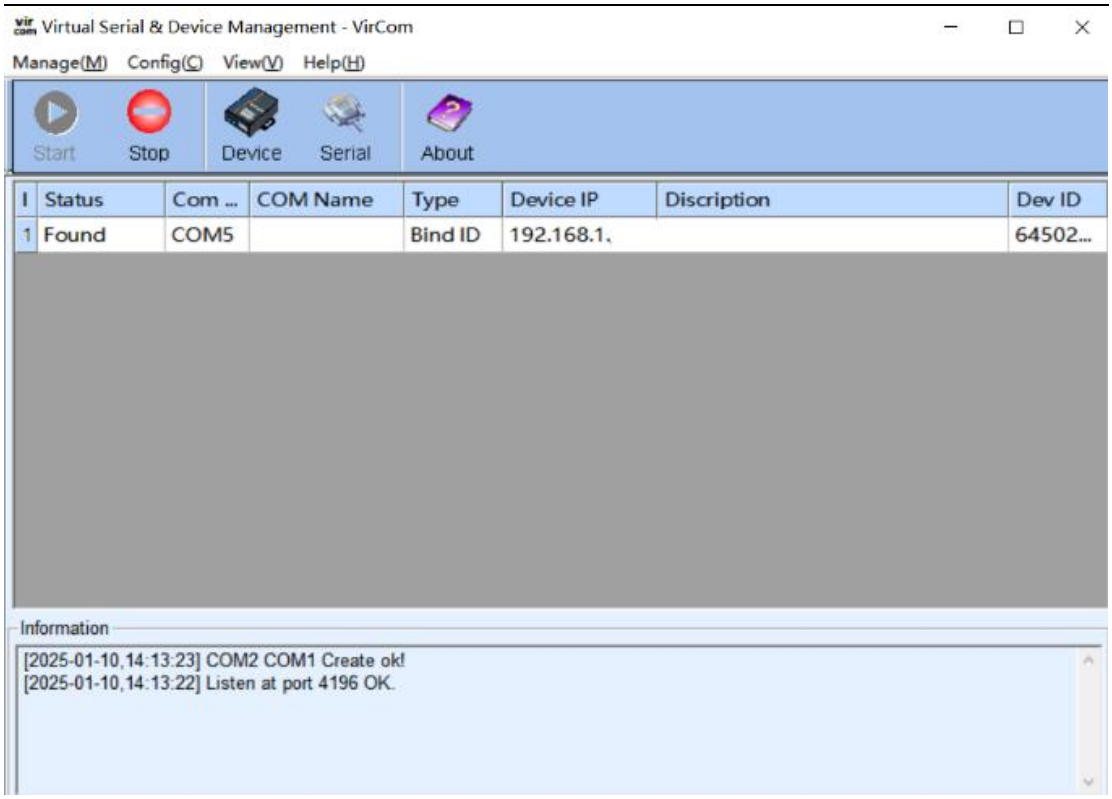

Figure 15 shows that the virtual serial port has been connected

Open ZLComdebug to simulate the user's serial port program, open COM5(the virtual serial port mentioned above), and then open another ZLComdebug to simulate a serial port device, open COM4(hardware serial port). At this point, the data transmission link of COM5 is as follows: COM5ZLVircom serial server network port Serial server Serial port COM4. Conversely, COM4 to COM5 can also transfer data: COM4 serial port server Serial port server network port ZLVircomCOM5. As shown in Figure 15, the data sending and receiving situations of both parties.

If COM4 is replaced with the user's serial port device, COM5 can achieve communication with the user's device.

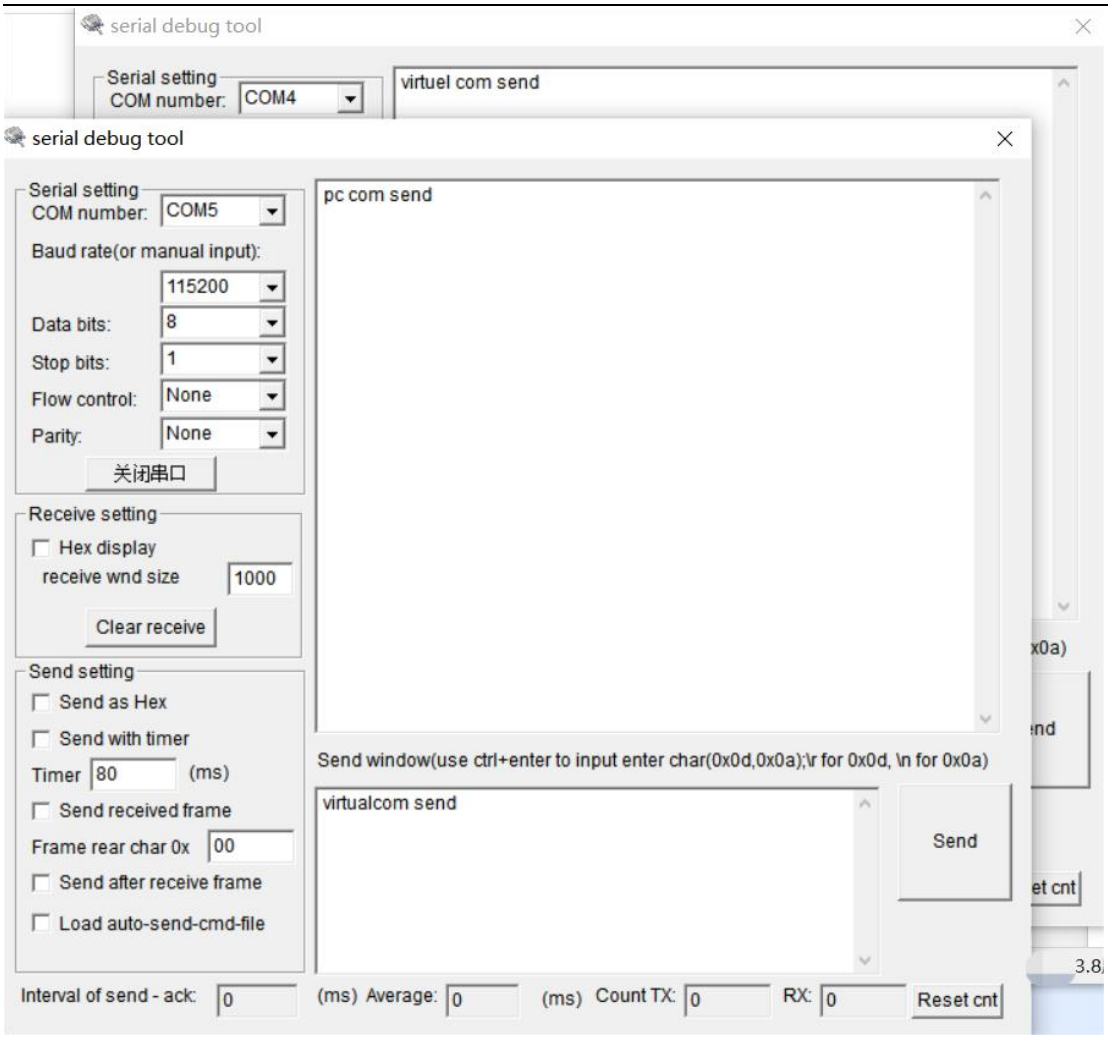

Figure 16: Communication via virtual serial port

4.7 Modbus TCP Testing

By default, serial port and network port data are transmitted transparently. If you need to convert Modbus TCP to RTU, you need to select "Modbus TCPRTU" as the conversion protocol in the device setup dialog box, as shown in Figure 23. In this case, the device port automatically changes to 502, and the Modbus TCP tool connects to port 502 of the serial port server IP address. The Modbus TCP command is converted to RTU command and output from the serial port. For example, if the port of the serial port server receives the Modbus TCP command 00 00 00 00 00 06 01 03 00 00 00 00 00 0a, the serial port outputs the command 01 03 00 00 00 00 00 a c5 cd. Note: The serial port may send more than one 01 03 00 00 00 00 0a c5 cd

command, this is because the default Modbus is stored mode, will automatically rotate the query command. I'll explain how to switch to non-storage mode later.

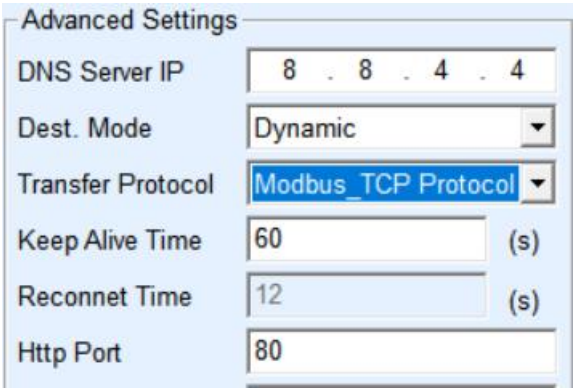

Figure 17 Enables the Modbus TCP function

If the user's Modbus TCP software is used as a Slave, then on the basis of choosing the conversion protocol, the working mode needs to be changed to client, 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 Figure 18.

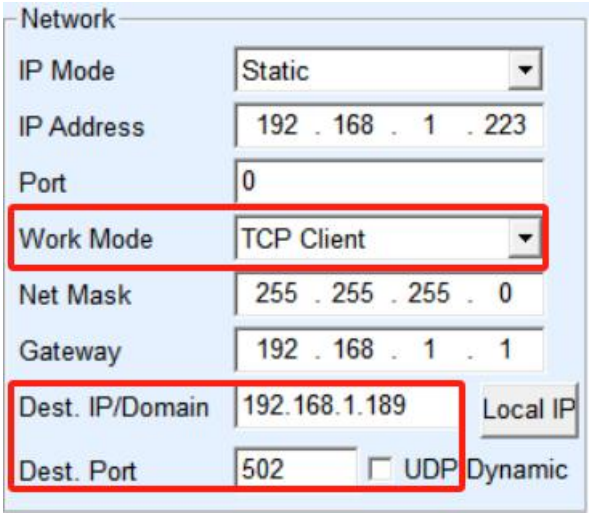

Figure 18: Modbus TCP as the client

4.8 Web-based configuration

ZLVircom can search and configure device parameters in different network segments. Web configuration requires first to ensure that the computer and the serial server are in the same IP segment, and the IP address of the serial server needs to be

known in advance. But Web configuration can be done on any computer without ZLVircom.

1. In the address box of your browser, enter the IP address of PORT1 of the serial port server, for example, http://192.168.1.200.

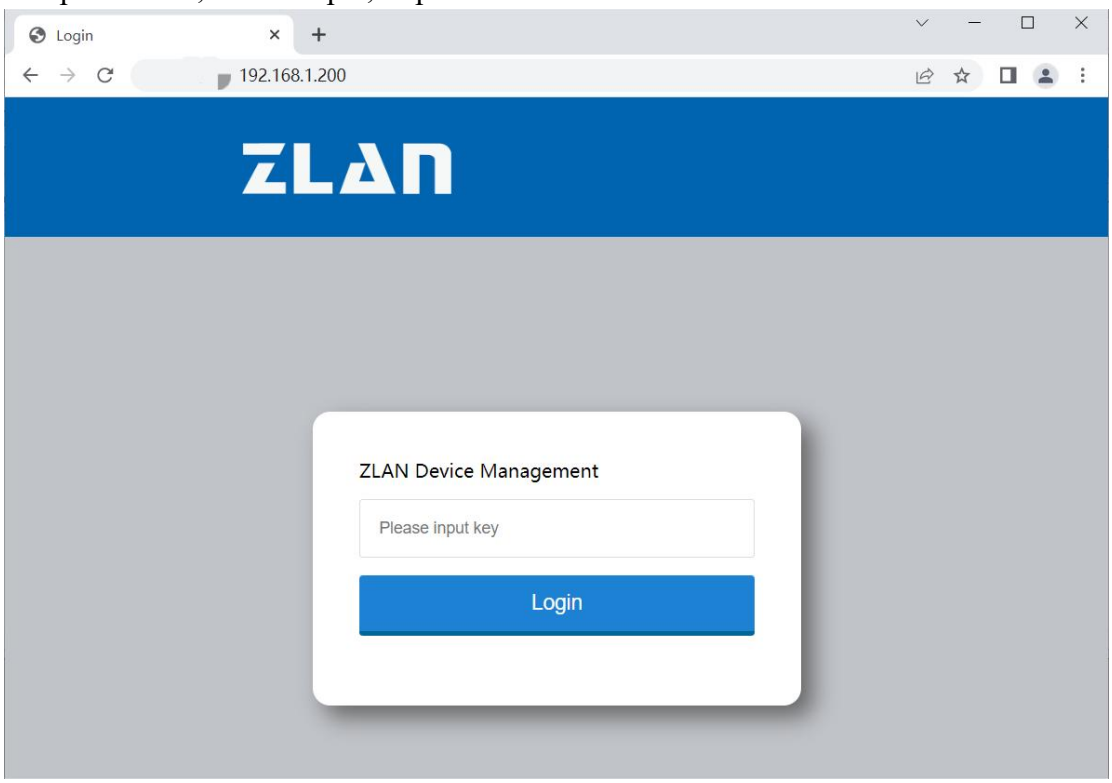

Figure 19 Login interface

2. Enter a Password in Password: there is no password by default. Click the login button to sign in.

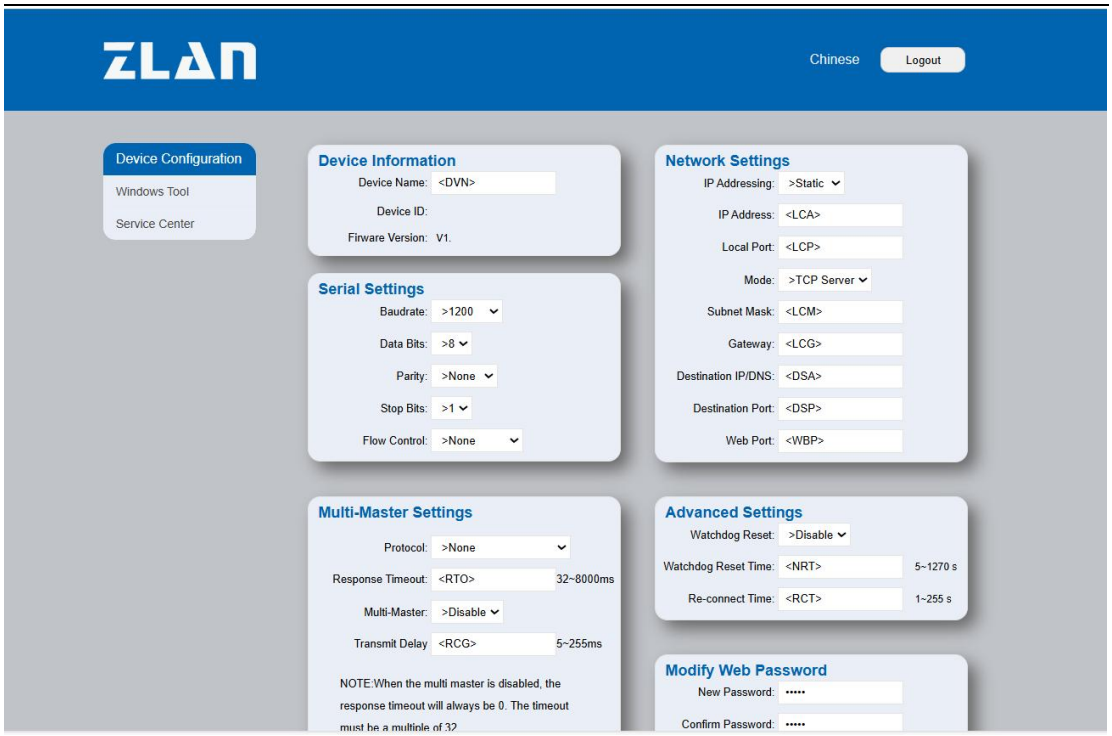

Figure 20 Web Configuration Interface

3. You can modify serial port server parameters in the displayed web page. For details about related parameters, see Table 4.
4. Click the "Submit Modification" button after modifying the parameters.

5. Working mode and conversion agreement

Different serial port server working modes and conversion protocols can be selected in different application scenarios, thus enabling more stable and reliable use. The following is a detailed introduction.

The use of serial port servers is basically divided into two types: with virtual serial ports and without virtual serial ports, as shown in the TCP communication schematic diagram in Figure 10 and the function of virtual serial ports in Figure 13 respectively. The user software that needs to be connected with the virtual serial port mode is the serial port interface (COM port), that is, both the user software and the user device are serial ports. In the non-virtual serial port mode, the user software communicates directly via TCP/IP, but the user device remains a serial port.

In the non-virtual serial port mode, the "conversion protocol section" is further

divided into three methods: transparent transmission, Modbus TCP to RTU, and Realcom protocol. If the user software is of the fixed Modbus TCP protocol and the lower-level machine is Modbus RTU, the Modbus TCP to RTU conversion mode needs to be selected. The Realcom protocol is currently only used when a multi-serial server serves as a TCP client to connect to a certain server and virtual serial ports are used on the server.

Usage is summarized as follows:

Table 6 Network configuration modes

No.	Virtual serial port usage	device working mode	Transfer protocol	Specifications
1	Use	TCP server	none	It is suitable for the occasion when the user software opens COM port to collect data actively.
2	Use	TCP client	none	If the TCP server is selected, the device may fail to reconnect after disconnection.
3	None	TCP server	Modbus TCP to RTU	The user software is Modbus TCP and the user device is Modbus RTU. And the Modbus TCP master station.
4	None	TCP client side	Modbus TCP to RTU	The user software is Modbus TCP and the user device is Modbus RTU. And Modbus RTU is the master station.
5	Use	TCP client side	Realcom protocol	When the multi-serial port server serves as the TCP client and the virtual serial port is used, the Realcom protocol is recommended.

6	None	TCP server	Telnet protocol	This mode applies to monitoring device serial ports when connecting to the Zlan serial port server over Telnet.
7	None	TCP client side	None	It is suitable for connecting a cloud with a large number of devices. In general, the cloud is a server with a public IP address on the Internet.

5.1. Virtual Serial Port Mode

If the user software uses the COM port to communicate, it must use the virtual serial port mode. Including some PLC software, configuration software, instrument software and so on.

Then see if the monitoring computers and devices are on the local network：

a) If the computer is a server with a public IP address leased on the Internet, the device must use TCP client mode to connect the device to the server. In this case, you can select 2 and 5 in Table 6. If multiple serial servers are used, you must select 5.

b) All in the local network (can ping each other), it depends on whether the host computer actively queries or the device actively sends data. If the device acts as the TCP client to send packets, mode 2 must be used. Otherwise, mode 1 can be used.

5.2. Direct TCP/IP communication mode

If Modbus TCP protocol conversion is not required and virtual serial ports are not needed, at this time, the user software may directly communicate with the network port of the serial port server via TCP/IP. The serial port server then converts the TCP/IP data into serial port data and sends it to the serial port device.

Generally, users of such usages develop their own upper computer network communication software, which integrates the parsing of the serial communication protocol of the device. This method is more flexible and efficient than virtual serial ports. Corresponding to ⑥ and ⑦ in Table 6.

In the section "4.6TCP Communication Test", it mainly briefly describes how a serial port server communicates when it serves as a TCP server. Here, we will discuss how TCP clients, UDP mode, and multiple TCP connections communicate with computer software. Among them, for computer software, SocketTest (software that mimics users' TCP/IP communication) is taken as an example.

The Zhuolan serial port server adheres to the standard TCP/IP protocol. Therefore, any network terminal that complies with this protocol can communicate with the serial port server. Zhuolan Technology provides a network debugging tool (SocketDlgTest program) to simulate network terminals to communicate with the serial port server.

For two network terminals (here the network debugging tool and the serial port server) to communicate, their parameter configurations must be paired.

5.2.1. TCP Client Mode

There are two working modes in TCP mode: TCP server and TCP client. No matter which mode is adopted, one party must be the server and the other must be the client. Only then can the client access the server. If both are the client or the server cannot achieve communication.

When the serial port server serves as the client, there must be three corresponding relationships, as shown in Figure 21. (1) Working mode correspondence: The working mode of the serial port server is the server mode corresponding to the network tool on the client side. (2)IP address correspondence: The destination IP of the serial port server must be the IP address of the computer where the network tool is located. (3) Port correspondence: The destination port of the serial port server must be the local port of the network tool. After setting it up in this way, the serial port server can automatically connect to the network tool. Once the connection is established, data can be sent and received.

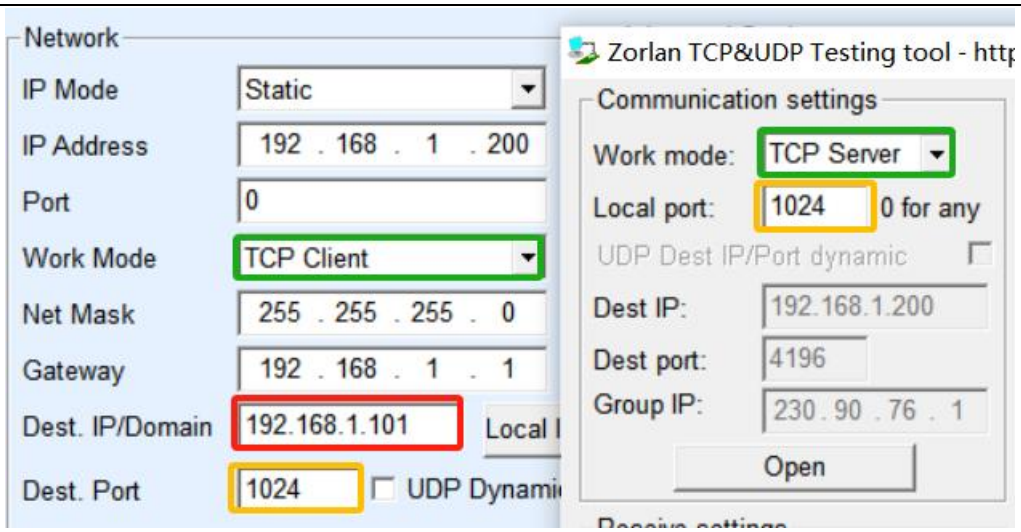

Figure 21 shows the serial port server as the client

5.2.2. The client connects to multiple servers

When the Zhuolan serial server serves as a TCP client, it can simultaneously connect to 7 destination IP addresses, and the data sent through the serial port will be sent to all 7 destination IP addresses at the same time. If there are not so many servers, the remaining destination ips will be left vacant. Its usage method is as follows:

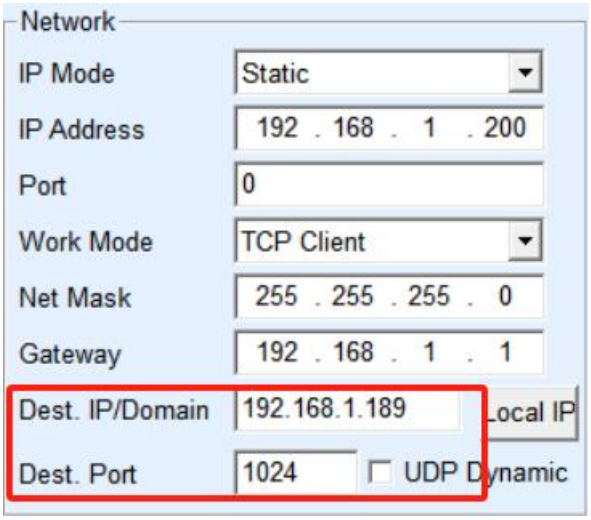

Figure 22 The first destination IP and port

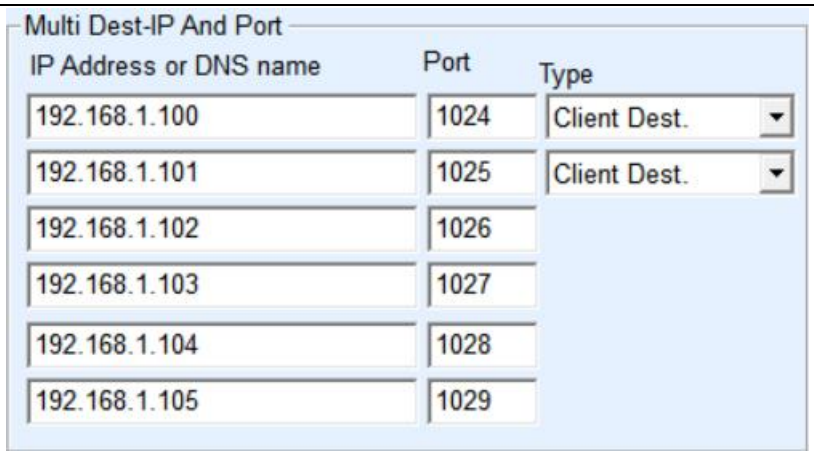

Figure 23 shows 2 to 7 remaining ips and ports

The first IP is set in the device Settings interface as shown in Figure 23, where the first IP can be a domain name. For the remaining 2 to 7 destination ips, click the "More Advanced Options" button in the device Settings interface to open More Advanced Options for Settings.

After all 7 destination ips are set up, the connection can be automatically established. If the connection fails, it will wait for the "disconnection and reconnection" time and then repeatedly reconnect.

5.2.3. TCP Server Mode

When the serial port server serves as the server, there are also three corresponding relationships, as shown in Figure 24. We will not explain them one by one here. After setting it up in this way, click the Open button of the network tool to establish a TCP connection with the serial port server. Once the connection is established, you can send and receive data.

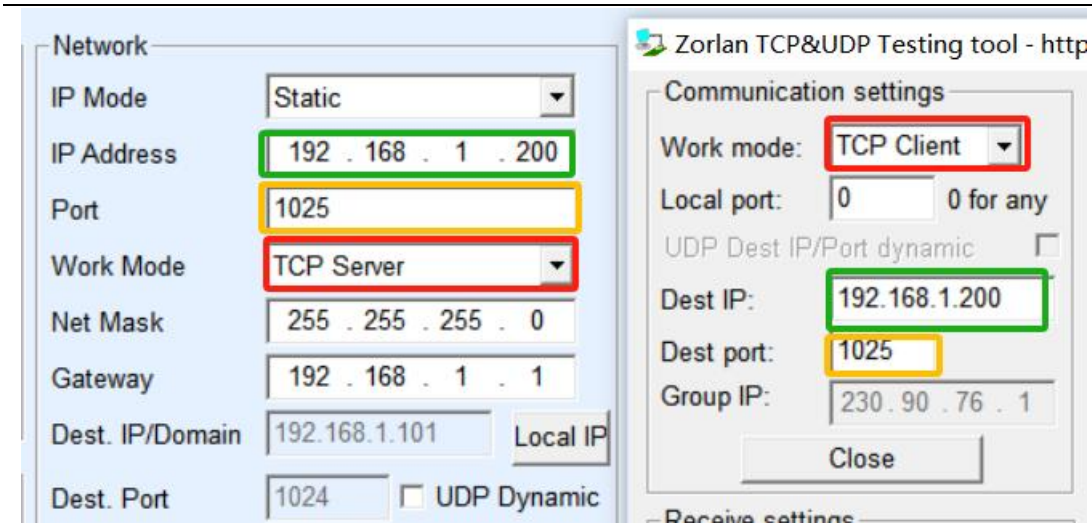

Figure 24: The serial port server serves as the server

When the serial port server serves as the server, it can simultaneously accept 30 TCP connections. The data received through the serial port will be forwarded to all established TCP connections. If you need to send data only to TCP that has recently received network packets, you need to enable the multi-host function. Please refer to 7.4 for multi-host function.

5.2.4. Act as both the client and the server

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

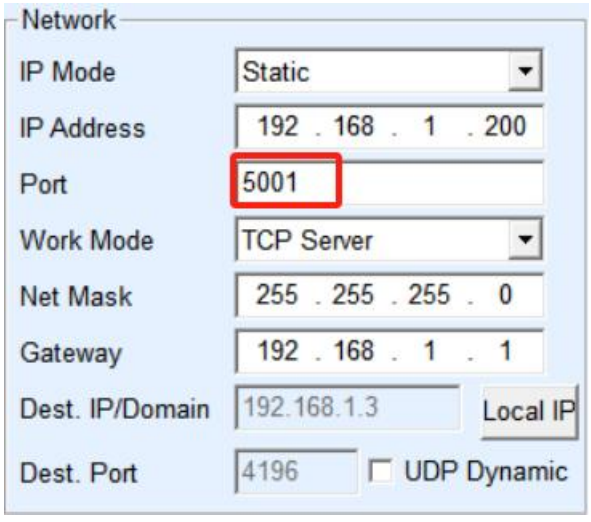

Figure 25: Acting as both a client and a server

By default, when configuring with ZLVircom, if the working mode is changed to

"TCP client" mode, the port (that is, the local port) will automatically become 0 (0 indicates randomly selecting an idle port). In order to support the TCP server mode, the computer software must know the local port of the device. Therefore, a value needs to be specified here. As shown in Figure 25, the computer software can now connect to port 5001 of 192.168.1.200 for communication. At the same time, the device will also connect to port 1024 of 192.168.1.189 as a client. It should be noted that since the local port 1024 is occupied by the server, when serving as the client, the local port used 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 26. The left side is the configuration of the serial server in ZLVircom, and the right side is the setting of the network debugging tool SocketDlgTest. First of all, both must be in UDP working mode. In addition, as indicated by the red arrow, the destination IP and destination port of the network tool must point to the local IP and local port of the serial server. As indicated by the blue arrow, the destination IP of the serial server must be the IP address of the computer where the network tool is located, and the destination port of the serial server must be the local port of the network debugging tool. Only after these network parameters are configured can two-way UDP data communication be guaranteed.

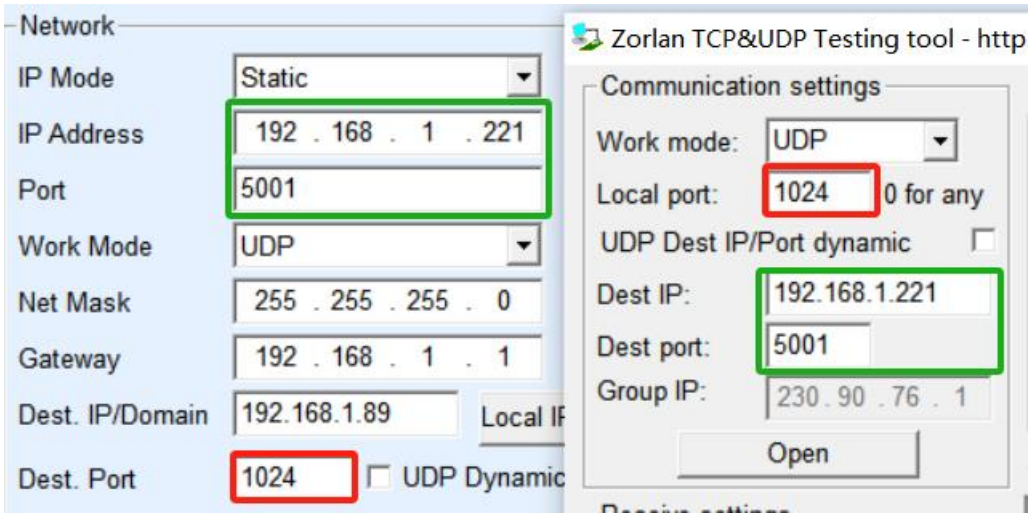

Figure 26 Configuration of UDP mode parameters

5.3. Equipment couplet methods

If the host computer is neither a Socket program (SocketDlgTest) nor a virtual serial port, but two devices are connected through the network port, the configuration method is similar. First, the user needs to connect two devices and a computer to the same local area network. ZLVircom is running on this computer. The purpose of connecting to the computer is only for configuration. After the configuration is completed, the computer does not need to be connected.

Click on the device management of ZLVircom and find these two devices, as shown in Figure 27. Then click "Device Editing" to configure the device. Device couplets can be divided into TCP couplets and UDP couplets. If it is in TCP coupling mode, the parameters of the two devices are shown in Figure 20. The parameters indicated by the arrows must correspond, just as they do when connected to a PC. After a successful TCP connection, you can check the connection status by returning to the "Device Management" dialog box, as shown in Figure 27. If the status of both devices is "Connected", it indicates that the TCP link between the two devices has been established.

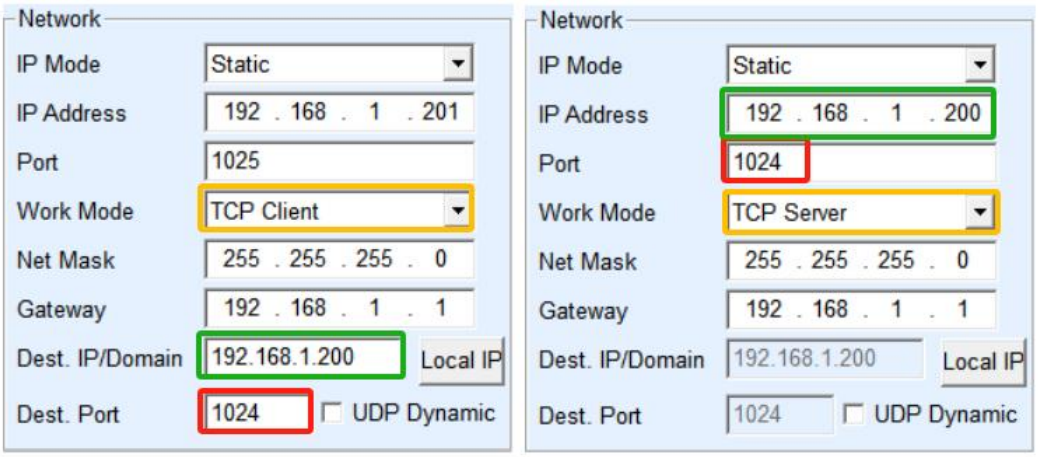

Figure 27 Configuration of TCP device couplet parameters

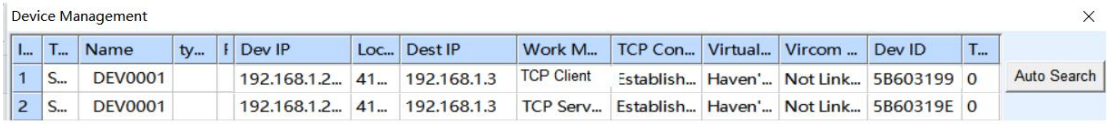

Figure 28: Successful check of TCP device couplets

If the connection is made in UDP mode, the configuration parameters are shown

in Figure 29. The parameters corresponding to the arrows must be one-to-one corresponding. As long as the parameters of UDP couplets are configured correctly, there is no need to check the connection status. The sent data will be automatically sent to the designated device.

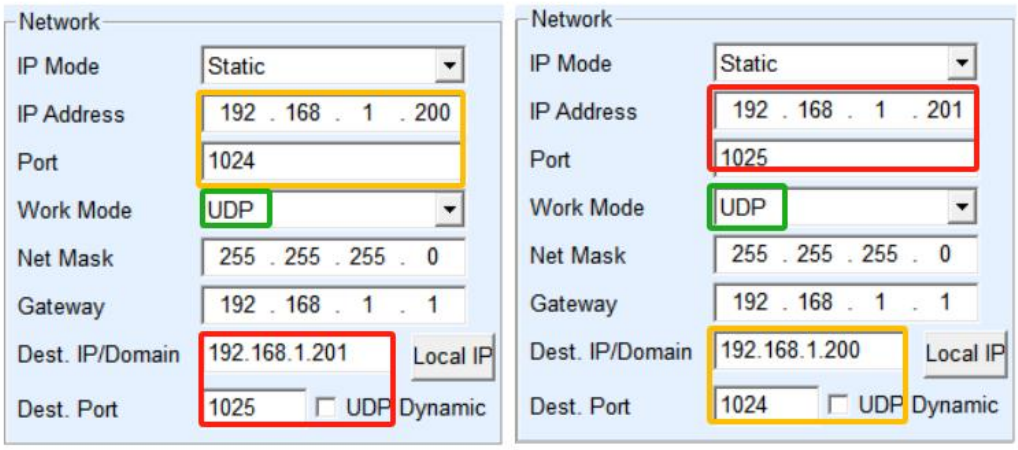

Figure 29 Configuration of UDP device couplet parameters

Finally, it should be reminded that if it is a device connection, in addition to setting the network port parameters as above, the correct serial port parameters must also be set. The main point is that the baud rate and other aspects of the serial port server need to be consistent with those of the user's equipment. After such Settings, user devices can send data to each other through the serial ports of the two serial port servers.

6. Equipment commissioning

6.1. Physical network connection

Serial port servers can be connected to switches or directly to the network ports of computers using crossover network cables or direct network cables.

After the connection is established, the first step is to check if the Link light is green. If not, please check if the network cable is properly connected.

6.2. Network TCP connection

When the device is used as a dynamic IP acquisition method, it cannot be directly

connected to the computer's network port. Because there is no DHCP server available (generally, a DHCP server is a router in a local area network). So when connecting directly, please specify the IP address. At the same time, the computer also needs to specify a fixed IP address.

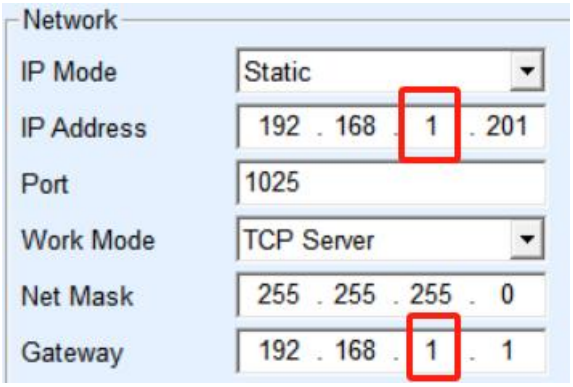

Figure 30 is configured in the same network segment

Whether directly connected or through a switch, when configured as a static IP, the device and the computer need to be in the same network segment (unless it is cross-gateway communication), as shown in Figure 30.

Since ZLVircom supports cross-segment search and configuration, devices that can be searched but cannot communicate usually have IP addresses that have not been properly configured. In such cases, ZLVircom can be used to configure the device within the same network segment.

After configuration, when using the steps of 4.6 TCP communication test or 4.7 Virtual serial port test, you can see that the Link light turns blue when establishing a TCP connection. The blue Link light can also be seen through ZLVircom. For instance, in the device management list, if the TCP connection column is "established", it indicates that the Link light is blue, as shown in Figure 30. This facilitates remote diagnosis.

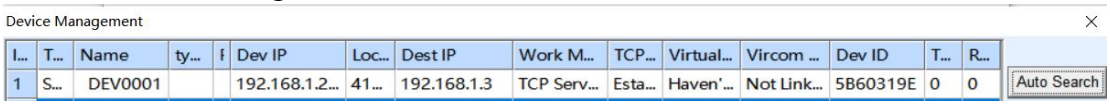

Figure 31 Connection status and data transmission and reception status

6.3. Data Transmission and Reception

When the Link light turns blue, data transmission and reception can be carried out between the software and the serial port server. At this point, if the software sends a piece of data, the Active light will turn green and the duration is usually at least one second. Data will also be output from the serial port of the serial server, but whether the output data is correct still depends on whether the correct serial port parameters (baud rate, data bit, stop bit, check bit) have been configured.

For correct instructions issued, the serial port device usually responds. Once there is a response (data sent from the serial port to the network port), the "Active" will turn blue. Otherwise, please check the serial port parameters or whether there is a problem with the connection of the serial port cable.

To facilitate remote debugging, ZLVircom also supports remote viewing of the data transmission and reception status, as shown in Figure 30. Here, TXD represents the amount of data sent from the serial port of the serial server. When refreshing the device list and seeing a change in this value, it indicates that data has been sent, and the Active light will also turn green. If the value of RXD is seen changing, it indicates that the serial port device has returned data, and "Active" is blue.

6.4. ZLVircom remotely monitors data

When using a virtual serial port, ZLVircom supports real-time capture of the data sent and received by the virtual serial port. To facilitate users' system debugging, the usage method is as follows:

Suppose the communication of the virtual serial port has now been established in accordance with the method of virtual serial port testing in 4.7. Now it is necessary to monitor the data through the virtual serial port. Open the menu/Configuration/Software Configuration of ZLVircom/Open the vircom Configuration dialog box.

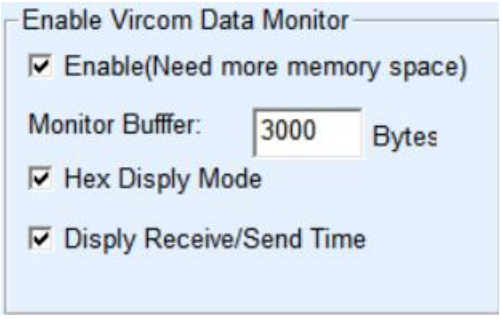

Figure 32 Enables ZLVirocm monitoring

Check the boxes in front of the three options of enabling monitoring, hexadecimal monitoring mode, and displaying data sending and receiving time, as shown in Figure 32. Then click OK. Assuming that data transmission and reception have been carried out before, now in the main interface, select a virtual serial port that needs to be monitored, and then choose Menu/View/Monitor, as shown in Figure 33.

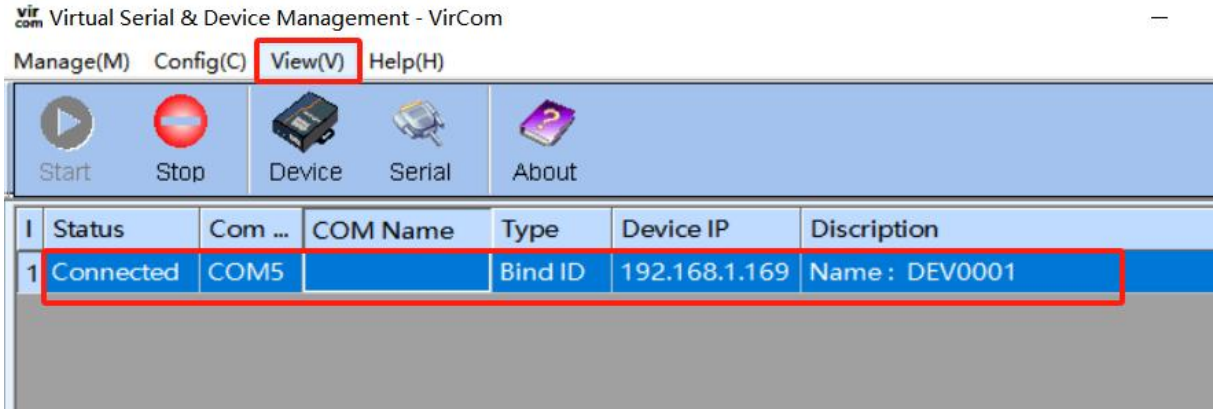

Figure 33: Enable ZLVirocm monitoring

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

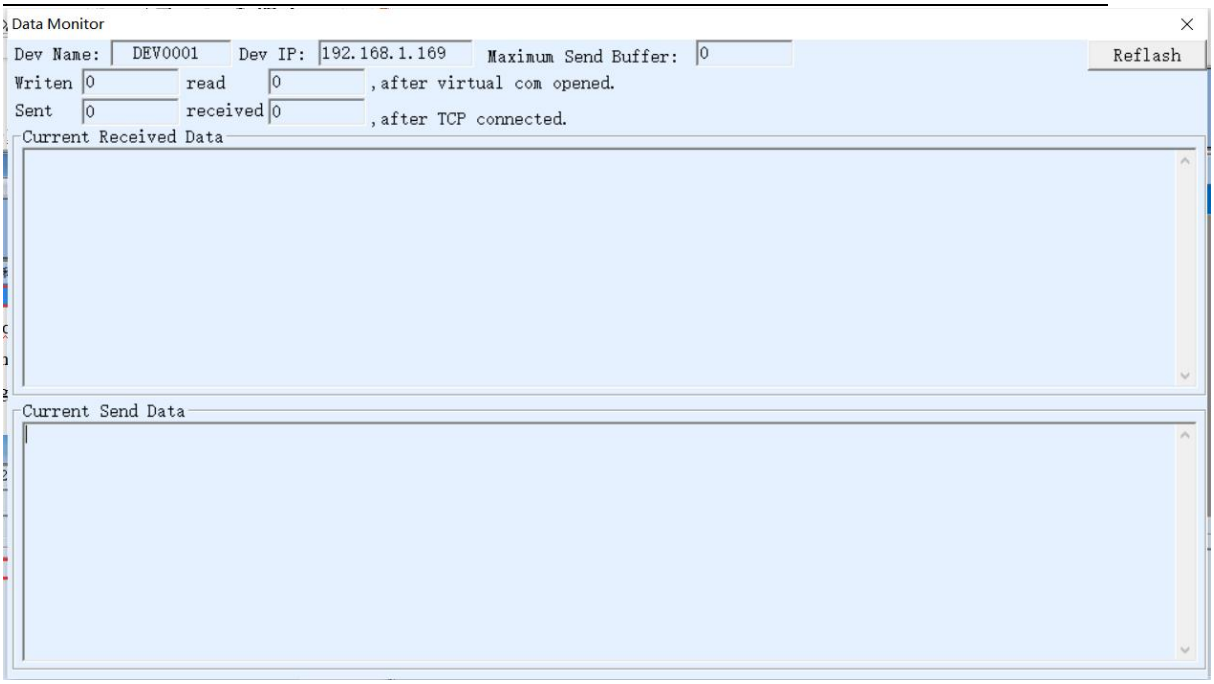

Figure 34 Monitoring the transmission and reception of data

7. Advanced functions of Modbus

The serial port server with Modbus gateway function itself does not have a station address or registers. It serves as a communication bridge. It generates a Modbus RTU specification based on the Salve ID, function code, register number, and register quantity in the Modbus TCP instructions sent by the user software to the Modbus gateway, and outputs it from the serial port. It can be regarded as a protocol "translator".

7.1. Enable the Modbus gateway

Firstly, the serial port server should support the Modbus gateway. That is to say, the "Modbus TCP to RTU" function in Table 5 of the device Settings dialog box should be checked.

By default, the serial port server is in the normal transparent transmission mode. If you need to convert to the Modbus gateway mode, please select the "Modbus TCPRTU" option in the "Convert Protocol". After that, the device automatically

modifies 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 serves as the slave station, the Modbus TCP software of the upper computer connects to port 502 of the Modbus gateway. At this time, the Modbus gateway needs to operate in TCP server mode. If the serial port RTU serves as the master station, the Modbus gateway operates on the TCP client side, and the destination IP is filled with the IP of the computer where the Modbus TCP software is located. The destination port is generally 502.

7.2. Storage Modbus gateway

ZLAN5107I is a register-saving Modbus gateway. ZLAN5107I can store the contents of the read registers inside the gateway, which can significantly increase the speed of Modbus TCP queries and offer superior performance when supporting multi-host access.

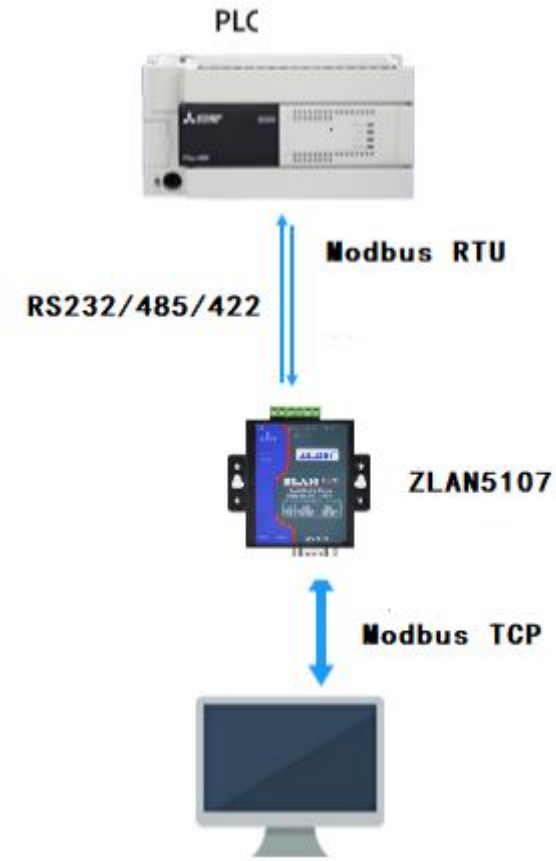

Figure 35 Working mode of the storage-type Modbus gateway

As shown in Figure 35: First, the Modbus TCP instructions are converted into the corresponding instructions of the Modbus RTU. Then, the device responds to the Modbus RTU instructions and sends them to the Modbus gateway. Subsequently, the Modbus gateway converts them back into Modbus TCP and sends them to the monitoring upper computer.

We know that Modbus TCP is a network communication with a very fast transmission speed. Generally, it can respond within 3ms, but the speed is usually only 9600bps. Generally, it takes at least 30ms to send and return an instruction. The query response time of such a common non-storage Modbus gateway is relatively long. In addition, if many upper computers query data simultaneously, the serial port will become congested. If the network is compared to a highway and the serial port to a single-log bridge, then the original way was to pass the traffic volume of the highway on the single-log bridge.

The register-saving Modbus gateway (ZLAN5107I) has solved the above problems. It can temporarily store the register data obtained from the query inside the Modbus gateway. Thus, when the Modbus TCP query comes in, the Modbus gateway can immediately return instructions, truly bringing out the fast feature of Modbus TCP. On the other hand, ZLAN5107I can actively send instructions from the serial port to automatically update the content of the currently saved register data and save the latest register value.

In addition, ZLAN5107I is a fully automatic Modbus gateway that requires no configuration. Users do not need to configure the required register addresses, function codes, slave addresses, etc. ZLAN5107I will automatically identify and dynamically add these registers based on the Modbus TCP instructions sent from the network port.

When multiple computers are monitoring, the ZLAN5107I can demonstrate an excellent response speed. Regardless of the baud rate of the serial port, it can generally provide response data to the upper level within 3ms. And it demonstrated a good real-time update speed of serial port data.

The region-saving Modbus gateway is a true conversion of Modbus TCP to Modbus RTU. It truly exploits the advantages of Modbus TCP, such as high speed and the ability for multiple hosts to query simultaneously.

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

The following lists the characteristics of the stored Modbus:

1. The first Modbus TCP query instruction is non-stored. Because it is necessary to wait for the RTU device to return data slowly before the content of the register can be restored to the network port.
2. If a specific instruction is no longer queried by the upper computer on the network end within 5 seconds, this instruction will be automatically deleted and will no longer be sent from the serial port to the RTU device.
3. Currently, it can store 10K of Modbus cache. For ordinary single-register queries, approximately 500 instructions can be stored simultaneously.
4. When multiple instructions are being queried simultaneously, they are sent in sequence. The first instruction is sent. The first instruction responds and waits for

the 485 anti-collision time (refer to the multi-host section). The second instruction is sent... . After the last instruction is responded to, return to the first instruction.

7.3. Disable the storage function

Although the storage-type Modbus has a relatively fast response speed, some users do not want the RTU device to receive a large number of query instructions, which may affect the internal processing speed of the instrument. At this point, the storage function can be turned off.

To disable the storage type, click the "More Advanced Options" button in the "Parameter Configuration" dialog box, remove one support and one enable as shown in Figure 35, and then click OK. Then go back to the device Settings and click "Modify Settings".

When configuring the conversion protocol via the Web, the default is a non-storage Modbus gateway.

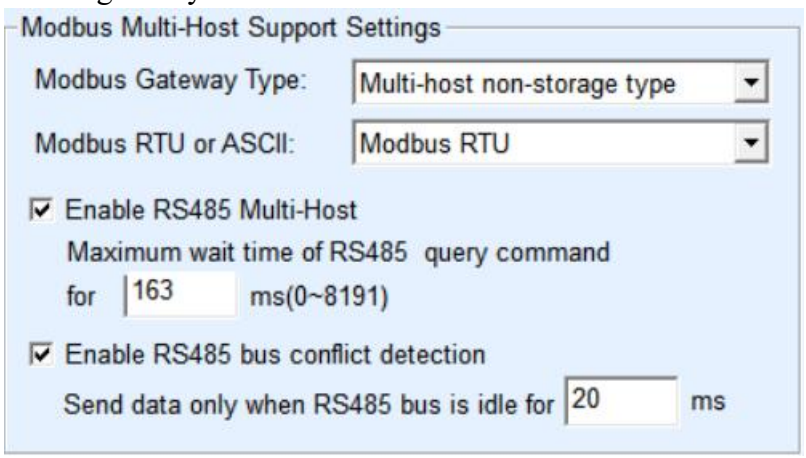

Figure 36 Disables the storage function

7.4. Multi-host function

As shown in Figure 36, the "RS458 multi-host support" and "RS485 bus conflict detection function" are the multi-host functions of Zhuolan. They are generally enabled and disabled simultaneously. After enabling, devices that convert the protocol to Modbus TCP should have the function of a stored Modbus gateway; otherwise, they should be non-stored Modbus gateways. If the protocol is converted to nothing,

generally, the user-defined RS485 protocol can also have the function of multiple hosts simultaneously accessing serial port devices. This is impossible to achieve in a pure RS485 network because multiple master stations sending simultaneously would cause conflicts on the RS485 bus. The multiple hosts of the Zhuolan serial port server can "coordinate" the RS485 bus to achieve the purpose of multi-host access.

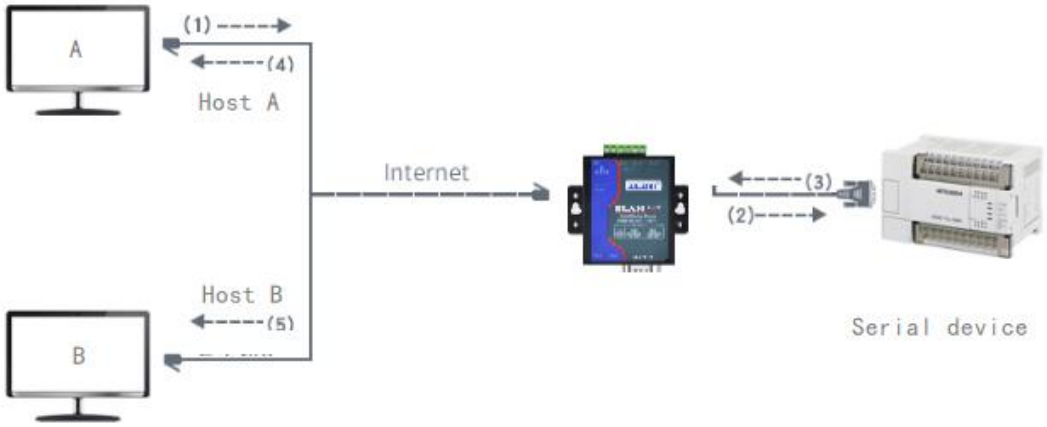

Figure 37 Demonstration of multi-host functions

As shown in Figure 37, in the normal mode, when two hosts, Host A and Host B, are simultaneously connected to the serial server, at this time, Host A sends instruction (1), the RS485 device receives instruction (2), and the RS485 device returns instruction (3). However, at the network port end of the serial server, it will simultaneously send instruction (4) to host A and instruction (5) to host B. Since Host B did not send the query but also received the response instruction (5), Host B may generate a communication exception error. In multi-host mode, there will only be instruction (4) and no instruction (5), because the serial server will automatically remember the host to be returned and only return the instruction to the nearest communicating host. Host A's query will only reply to host A, and Host B's query will reply to host B.

Another function is that in the normal mode, if both Host A and Host B send data simultaneously, instructions will be merged on the RS485 bus, making it impossible to recognize them normally. In A multi-host mode, the serial port server can schedule the sequence of A and B on the bus usage, thereby effectively resolving the conflict issue of multiple machines accessing simultaneously.

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

7.5. Multi-host parameters

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

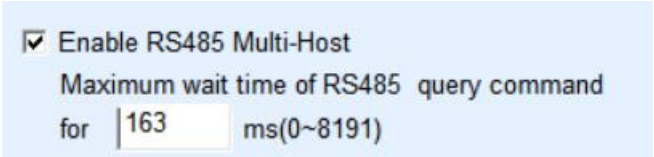

Figure 38 Multi-host support for RS485

Among them, the RS485 command response timeout time is: the maximum time interval from the start of sending this command to the receipt of the response by the serial port server's serial port. The time filled in should be greater than the actual maximum time interval. Because if a timeout is determined, the next instruction will be sent.

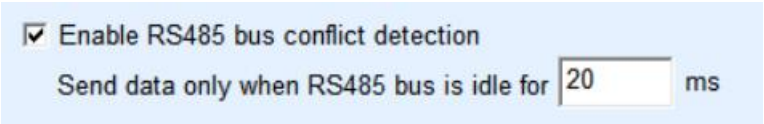

Figure 39: RS485 anti-collision Idle Time

RS485 bus collision time: It indicates how many milliseconds the serial port server waits after receiving the response to the first instruction before sending the second instruction. This parameter actually defines the speed of instruction rotation training. It is recommended that this value be above 20ms. The parameter "maximum waiting time of 3 seconds" generally does not need to be modified.

When the user selects the conversion protocol as "After Modbus TCP to RTU" using ZLVircom, ZLVricom 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 Modus instruction is relatively long or the conversion protocol is set to "none", these two parameters need to be manually configured.

The following introduces the recommended values for setting the above parameters:

1. Figure 39 shows the "RS485 bus collision prevention time", which can generally be set to twice the "packet interval" at the lower right corner of the parameter configuration interface, but the minimum cannot be less than 20.
2. Figure 38 shows the "RS485 instruction response timeout time", which is generally determined by the length of the instructions that respond back and forth. If the sending instruction is N bytes and the response is M bytes, it is recommended to set the value as: "Data packet interval" × (N+M+5) +100.

7.6. Modbus in Multi-Destination IP

As shown in Figure 37, if the serial port device (RTU device) serves as the master station and the network port device (Modbus TCP device) acts as the slave station, and there are multiple network port slave stations simultaneously. At this point, the serial port server can be used as the client to simultaneously connect to these multiple network port devices, following the method introduced in 5.2.2 for client connection to multiple servers.

The function to be achieved at this point is: when the serial port RTU sends instructions, it can be sent to multiple network port devices. The network port devices identify whether to send them 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 port server, it is converted into an RTU instruction and output from the serial port to the RTU device.

7.7. Configurable Modbus gateway

For the configurable Modbus gateway ZLMB, please refer to "Four Types of Modbus Gateways"http://www.zlmcu.com/document/Modbus_Gateway.html。

8. Registration package and heartbeat package

The registration package and heartbeat package are features suitable for communication between devices and cloud software.

8.1. Registration package

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

The registration package is highly suitable for the monitoring of the Internet of Things, as cloud software typically runs on public network servers on the Internet, while modules are scattered across various collection and monitoring points. How to make the cloud software recognition module very important is essential for achieving Internet of Things communication.

The serial port server of Shanghai Zhuolan offers the following multiple registration methods.

8.1.1. Connect to send the MAC address

Connect and send the MAC address: This method is not only supported for model 4 (such as 5143), but also for ordinary models. Its method is to send its mac address to the cloud when the module is connected to it. Because the MAC address is unique, the device can be uniquely identified. This method is simple and does not require the writing of registration packages for each device, so it is simple and effective. The usage method is as follows: In the device Settings dialog box, click "More Advanced Options", in the middle and upper part, find "Send MAC address when TCP is established", check the box in front, then return to the Settings interface and click "Modify Settings".

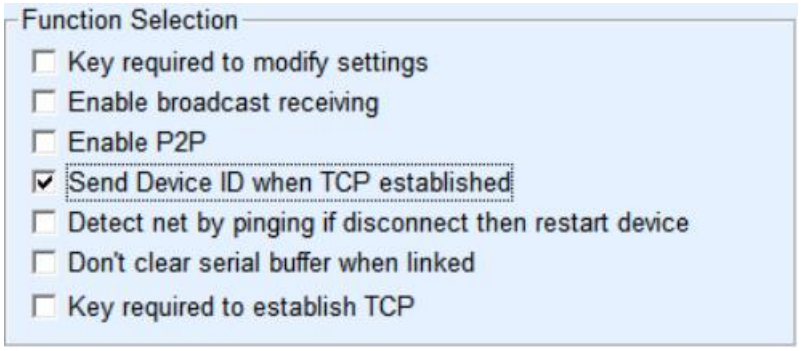

Figure 40 connects to send the MAC address

8.1.2. Realcom Protocol

The Realcom protocol is a mature protocol that contains registration packets and heartbeat packets. Users can use this protocol to implement the functions of registration packets and heartbeat packets. The method to enable the Realcom protocol is: In the "Device Settings" dialog box, select "Convert Protocol" as "REAL_COM Protocol". Note that the "Enable Registration Package" section needs to be left blank and not checked.

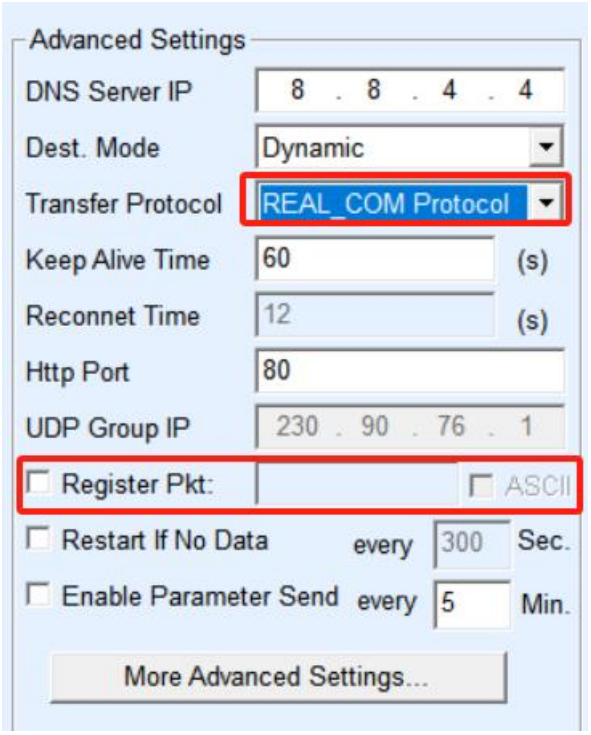

Figure 41 Enables the realcom protocol

After a TCP connection is established between the device and the cloud, the device automatically sends a hexadecimal registration packet FA 07 13 02 FA 02 MAC[5] MAC[4] MAC[3] MAC[2] MAC[1] MAC[0] FA FF. MAC[5] to MAC[0] indicates the MAC address of the device.

1. When the device sends data to the network, it automatically adds the three-byte prefix FA 01 01.

2. At a keepalive interval, the device sends a 1-byte heartbeat packet of 00 to the software.

The REAL_COM protocol can be used as the registration package of the device

because the registration package contains the MAC address. However, due to its fixed format, only cloud software can design the REALCOM protocol to be compatible with this approach.

8.1.3. Custom Registration Package

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

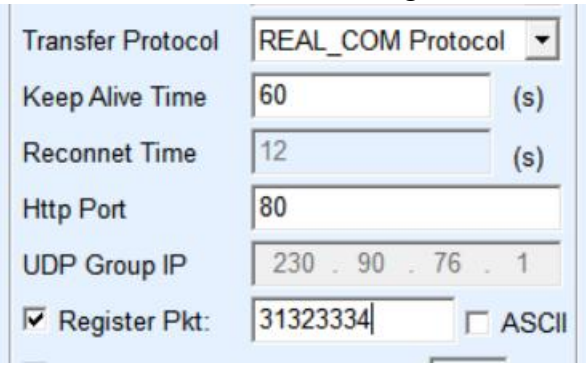

Figure 42 Sets up the registration package

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

After the device is connected to the cloud software, it can automatically send hexadecimal registration packets of 31, 32, 33, 34. This registration package method is relatively flexible and enables the device to adapt to the existing cloud registration package format. However, there is no wildcard like MAC in the registration package. Different registration packages need to be configured separately for each device, which is rather cumbersome. The two methods of sending MAC addresses and REALCOM above have the same configuration for each device, but due to different Macs, the registration packages are naturally different.

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

8.2. Heartbeat Pack

The heartbeat packet is mainly used to detect whether the communication link is disconnected. The implementation method is that the device sends a heartbeat packet data to the server software at regular intervals. Once this data is received by the server, it will be discarded and not regarded as valid communication data.

The heartbeat packet mainly has two functions: Firstly, it enables the upper computer software to know that the device is in an active state; Secondly, if the device fails to send a heartbeat, the device on the TCP client will automatically re-establish a TCP connection. Therefore, it is a means to restore network communication.

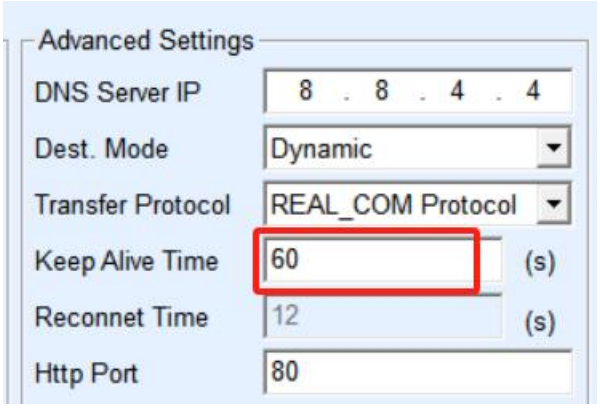

Figure 43 Survival timing

As shown in Figure 42, the sending time of the heartbeat packet is set through the "survival timing".

8.2.1. Implied Heartbeat

Even if no heartbeat packets are set, the hidden heartbeat function will be enabled when the Zhuolan device is in the TCP client. So the implicit heartbeat function refers to the situation where the device sends data, but the server actually does not receive the heartbeat data. So it cannot perform the first function of the heartbeat packet, that is, the function of the server detecting whether the device is active or not. However, since the device actually sends data, it can serve the second function of a heartbeat packet, that is, to detect whether the TCP connection is normal. Once a disconnection is detected, it can still automatically re-establish the TCP connection.

8.2.2. REALCOM Protocol

As described in 8.1.2 of the Realcom protocol, the REALCOM protocol can send a 1-byte data of 00 at regular retention intervals. This data is the heartbeat packet of the realcom protocol.

8.2.3. Custom Heartbeat packs

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

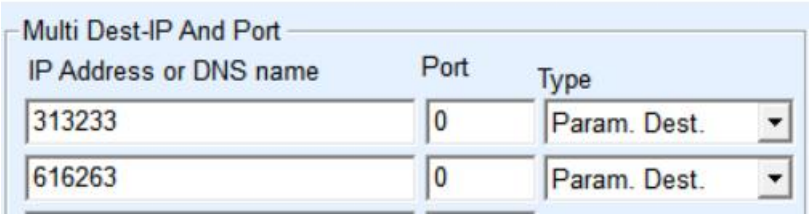

Figure 44 Custom Registration Package

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

9. httpd client communication function

This feature is suitable for users who need to integrate device management functions into their own software.

The "UDP Management Port Protocol" has been integrated into the device management function library ZLDevManage. This is a DLL development library for the Windows platform, which can be called by various development tools such as VC, VB, Delphi, etc.

Detailed API interface documentation and VC calling Demo cases are provided. It can achieve device search, parameter modification, P2P function calls, etc.

The development library can be obtained from the ZLan official website: find "Device Management Function Library" on the http://zlmcu.com/download.htm page. For details, please refer to "ZLan WinP2p and Device Management Development

Library".

10. MQTT Gateway

For the usage of the MQTT gateway function, please refer to "The Usage Method of MQTT Gateway"http://www.zlmcu.com/document/Usage_of_MQTT_Gateway.html。"Usage of Zhuolan MQTT and JSON to Modbus Gateway"http://www.zlmcu.com/document/MQTT&JSON_to_Modbus.html；

11. JSON TO Modbus RTU

For information on converting JSON to Modbus RTU and related JSON usage, please refer to "JSON Data Acquisition Gateway"http://www.zlmcu.com/document/jsondata.html ； "Method for Uploading 645 Instrument in JSON Format" 》http://www.zlmcu.com/document/645_Instrument_JSON.html ； "Usage of Zhuolan MQTT and JSON to Modbus Gateway"http://www.zlmcu.com/document/MQTT&JSON_to_Modbus.html；

12. NTP time function

For details about how to obtain and use NTP time, please refer to "Zlan NTP Time Module Usage"http://www.zlmcu.com/document/zlan_NTP.html。

13. Modify the parameters of the network port

Network port parameter modification is to realize the function of searching equipment and modifying equipment parameters like zlvircom software, that is, to manage equipment and modify parameters through the network port of serial server. Suitable for users who integrate search and configuration capabilities into their software.

Network port modification parameters are implemented through the UDP Management Port Protocol, for example:

1. The computer software sends UDP broadcast packets whose destination port is 1092 on the network. When the device receives the packet, it will return its information to the computer software to search for the device.
2. The computer software sends the UDP parameter modification command to port 1092 of the device to modify the device parameters.

For details about network port modification parameters, see the ZLAN Network Products UDP Management Port Protocol. It can also be done directly using the device management library of the 14 Device management library.

14. Device Management Function library

This feature is intended for users who need to integrate device management functions into their own software.

The "UDP management port protocol" has been integrated in the device management function library ZLDevManage. This is a DLL windows platform development library, can be called by VC, VB, Delphi and other development tools.

Provide detailed API interface documentation and VC call Demo cases. It can realize device search, parameter modification, P2P function call and so on.

Development libraries can be obtained from the ZLan's official website: http://zlmcu.com/download.htm looking for "equipment management function library" page. For details, please refer to "Zlan WinP2p and Device Management Development Library"

15. Modify parameters through the serial port

Users can read and set parameters by sending commands to the serial port server's serial port. This is suitable for users who need to control and configure products at the chip or module level through serial port. The parameters that can be

set include: IP address, baud rate, device name, working mode, etc. After the new parameters are set, the serial port server can be restarted through serial port commands.

The serial commands of Zlan have the following features:

1. Serial port commands use a 10-byte data preamble, so there is no need to distinguish between communication data and commands by pulling down or pulling up additional configuration pins, nor is there a need to switch between command mode and communication mode, making usage more flexible and convenient.
2. The command set includes various command formats such as saving parameters, not saving parameters, and restarting the device.
3. It can implement a variety of applications, such as reading the MAC address of the serial port server, or modifying the operating mode of the serial port server. For example, when switching from TCP server to TCP client mode, it can actively connect to the server; when switching from TCP client to TCP server mode, it can disconnect from the server connection.

Please refer to "Serial Port Parameter Modification and Hardware TCP/IP Stack Protocol" for detailed operation methods on modifying serial port parameters.

16. Remote device management

Remote device management refers to the ability to maintain and manage devices through ZLVircom software, including restarting devices, modifying parameters, and updating firmware. This feature is suitable for users who manage devices through ZLVircom.

For ZLVircom software, as long as the device can be found in the device list, remote management can be performed. Remote management of devices can be divided into the following situations:

1. Automatic search: Under the same switch, regardless of whether they are on the same network segment, the way ZLVircom searches for devices on a computer is:

ZLVircom sends a broadcast query (all devices receive the query and reply with their parameters to the ZLVircom tool. This method searches for all devices at once.)

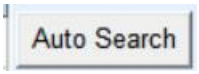

Figure 45 automatic search

2. Manually add: There are two situations:

Figure 46 manually added

   a) Large routers divide the network: In some large networks, broadcast packets are divided by routers, so broadcast packets cannot reach the device end, but pinging the device IP is always through. At this point, manual addition is generally needed to solve the problem. The method of manual addition is to click "Manual Addition" in the "Device Management" dialog box to add the first and last IP for individual queries of the device.
   b) Public network server queries internal network devices: Serial servers are in the internal network and operate in TCP server mode, while zlvircom is on a server with a public IP. At this point, a UDP port mapping of 1092 needs to be set up on the router of the network where the device is located, mapping to the IP of the device, and then zlvircom manually adds this device, with the IP being the public IP of the device end.
3. TCP Client: When the device acts as a TCP client, it will initiate a TCP connection to the destination IP (116.15.2.3) on port 4196. Once the connection is established, it will automatically send its parameter system to the destination's UDP port (not the TCP port) every keep-alive time interval, allowing zlvircom to detect the device on this computer (116.15.2.3). If the destination port is not 4196, you will need to modify zlvircom's default parameter receiving port by changing the menu/config/software configuration/default listening port. After starting zlvircom, if a TCP port conflict message appears, ignore it and continue with the execution.

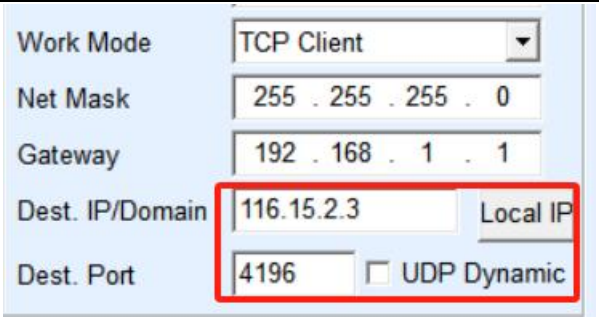

Figure 47 Client

4. Scheduled transmission parameters: Even when in TCP server mode, the serial server can select the "Scheduled Transmission Parameters" feature to send parameters every 5 minutes to the destination IP (here, 116.15.2.3) on the destination port. The zlvircom on this server's port can receive parameters and manage these devices.

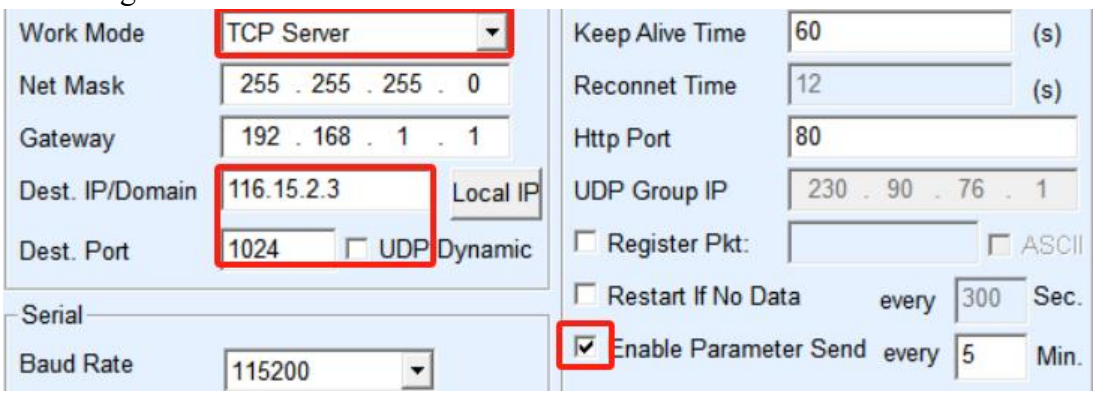

Figure 48 Timing transmission parameters

To facilitate the identification of devices, if remote management is required, please name the device something easy to remember.

17. Firmware upgrade method

ZLAN5107I can upgrade individual programs, but not each other. You can use this method to upgrade the firmware of devices found in the device list by automatic search, manual add, or P2P search.

1 Get ZLSN2007 firmware files from ZLAN, such as 1.452(2007).bin.

2 In the ZLVircom tool, first search the device that needs to be upgraded, and then

enter the device parameter editing dialog box. First click "Restart Device" once.

Figure 48 Upgrade button

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

Figure 49 Upgrade button

3 Select the "Program File Download" option, as shown in Figure 50. In the program file, select the firmware file. The IP address of the serial port server is automatically filled in, and the module type/model is automatically selected. Then click Download.

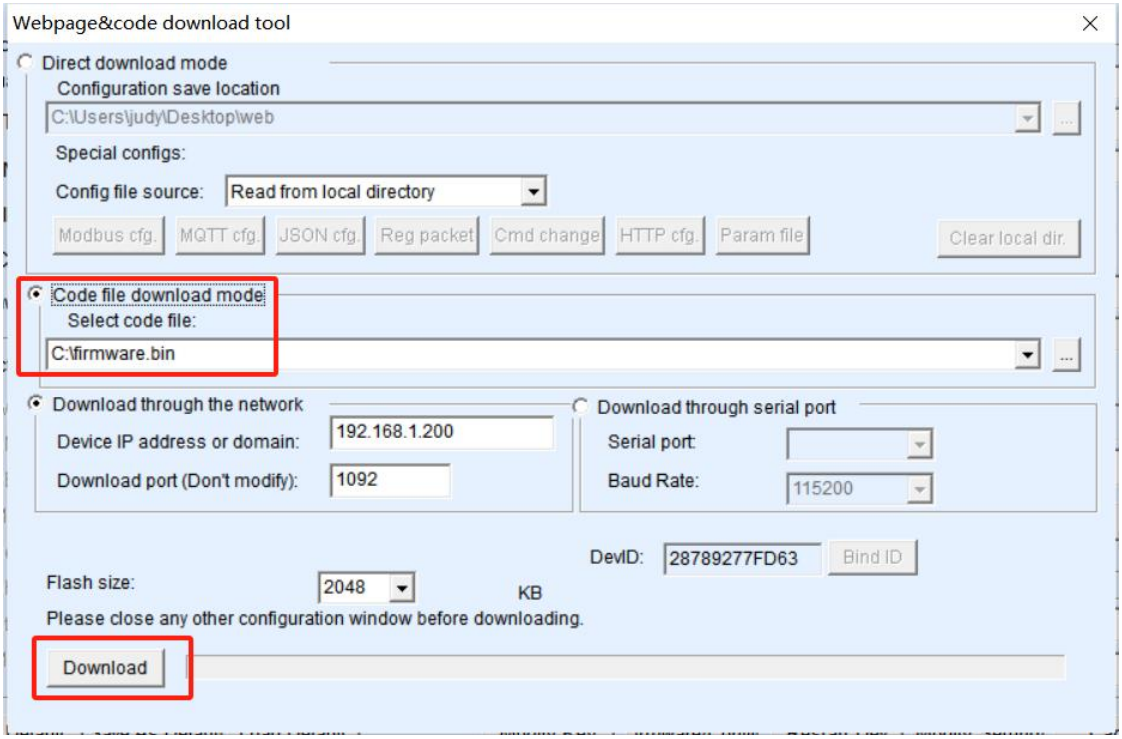

Figure 50 ZLSN2007 firmware upgrade method

4 At this time, the download progress bar starts to move, and the download time is about 30 seconds. During the download process, you will see the ACT light of the device blink, and at the end of the download, you will see the LINK light blink several times. Then the program pops up "transmission completed LINK light flashing device do not power off" prompt box. Note: This is only the completion of transmission, write to the flash process takes about 3 seconds, at this time the LINK light will blink, during this period please do not power off.

5 After downloading the general program will automatically restart, generally do not need to power off. If the running indicator is blinking, stop the LINK indicator blinking for more than 30 seconds and power it on again.

6 Web Configuration interface update: After the firmware upgrade, the configuration page inside the module also needs to be updated. Otherwise, the configuration cannot be configured through the Web, but the communication will not be affected. It is also possible not to download web pages without web configuration. The method of downloading Web is: as shown in Figure 51, change the download mode of "program file" to "Web directory download". In addition, select the root directory of the local web page as the directory of the web file to be downloaded (this directory can be obtained from ZLAN), and click Download to download all files in the local web page directory to the internal file system of the device.

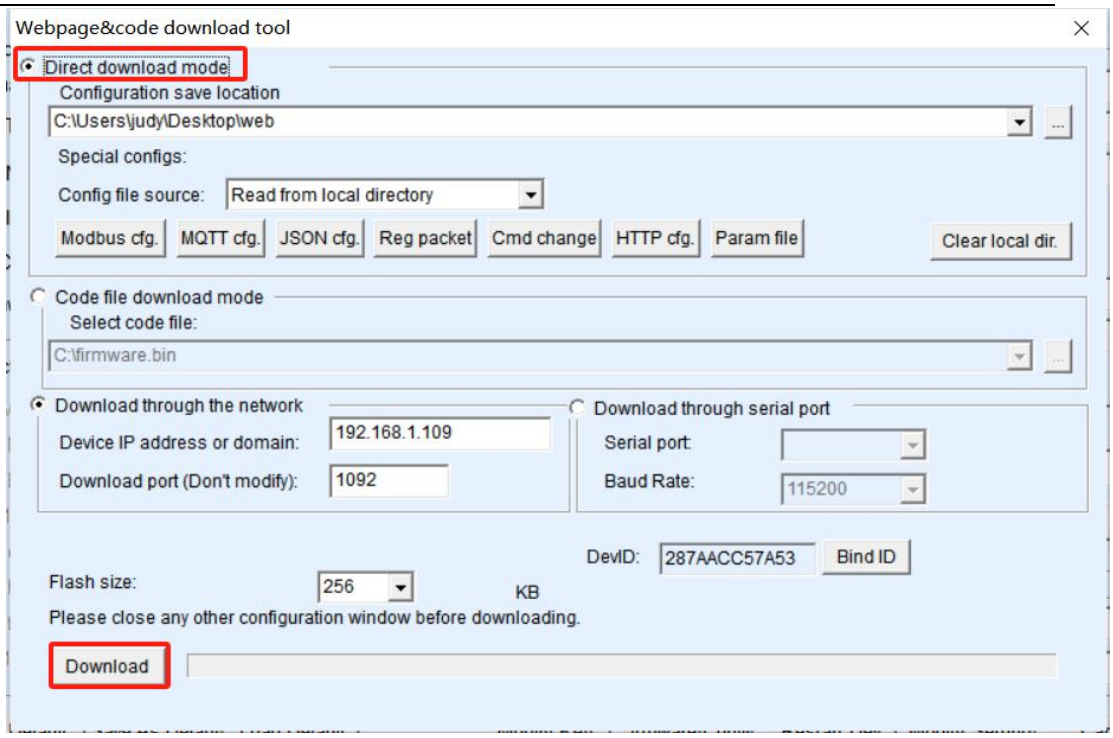

Figure 51 web upgrade method

7 Look out：

7.1 If the message "Download failure" is displayed, the device will not be damaged. You can download the device again. In addition, when the LINK light blinks at the end of the download, do not power off, otherwise the device will be damaged.

7.2 Check the firmware version through ZLVircom to know whether the new firmware has been downloaded successfully.

18. After-sales service and technical support

Shanghai Zlan Information Technology Co., Ltd.

Address: Room 2001, No. 28 Yuanwen Road, Minhang District, Shanghai

Phone: 021-64165189

Fax: 021-64165200

URL：http://www.zlmcu.com

Email：support@zlmcu.com