COME-STAR

Industrial Modbus Gateway

User Manual

- Cgate3202
- Cgate3204-4D232
- Cgate3204-4D485
- Cgate3208-8D232
- Cgate3208-8D485

Version: V1.6

Trademark

COME-STAR is the exclusive trademark of Come-Star Communication (Wuhan) Co., Ltd.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

All related trademarks mentioned in this operating manual belong to the respective manufacturers.

Copyright

Copyright © Come-Star Communication (Wuhan) Co., Ltd.

Statement

1. This user manual applies to the Cgate3202, Cgate3204, and Cgate3208 Modbus gateways, including Cgate3202, Cgate3204-4D232, Cgate3204-4D485, Cgate3208-8D232, and Cgate3208-8D485.

2. Before using this manual, please read the following disclaimer carefully. You may only use the products described in this manual if you agree to the following disclaimer.

Disclaimer

Any information provided by our company in this manual does not constitute an authorization for such information.

Our company strives to ensure that the information provided in this manual is accurate and applicable; however, we do not assume any responsibility for the use of this information, nor do we assume any joint liability. The products and user manual may contain technical or typographical errors.

Our company reserves the right to change all or part of the content of this manual without prior notice.

Safe Use Instructions

This product has good and reliable performance within its designed usage range, but it is important to avoid any damage or destruction caused by human actions.

- Please read this manual carefully and keep it for future reference.
- Do not place the device near water sources or in damp areas.
- Do not place anything on the power cable; it should be kept in a location where it cannot be touched.
- To avoid fire hazards, do not knot or wrap the cables.
- The power connectors and other device connections should be securely connected; please check them regularly.
- Please keep the device clean; if necessary, wipe it with a soft cotton cloth.
- Please do not attempt to repair the device yourself, unless explicitly instructed in the manual.

If the following situations occur, please disconnect the power immediately and contact our company.

- Water has entered the device.
- Equipment damaged
- Equipment is working abnormally or the performance displayed has completely changed
- Equipment emits odors, smoke, or noise

Note: Important information regarding the use of this device

Caution: Important precautions to observe when using this device

Contents

1 Product Introduction 5
1.1 Product overview 5
1.2 Product Features 5
1.3 Product Display 6
1.4 Specifications 7
1.5 Interface Introduction 9
1.5.1 Cgate3202 View 9
1.5.2 Cgate3204 Series View 11
1.5.3 Cgate3208 Series View 12
1.5.4 Power Interface 13
1.5.5 Ethernet Interface 13
1.5.6 RS232/485 Serial Port (Cgate3202) 13
1.5.7 RS232 Serial Port (Cgate3204-4D232) 15
1.5.8 RS232 Serial Port (Cgate3208-8D232) 15
1.5.9 RS485 Serial Port (Cgate3204-4D485) 15
1.5.10 RS485 Serial Port (Cgate3208-8D485) 16
1.5.11 RS485 Termination Resistor 16
1.5.12 Button 17
1.5.13 CONSOLE Port 17
1.5.14 Grounding Screw 17
1.5.15 Indicator Light 17
1.6 Installation Dimensions 19
2 Quick Start 21
2.1 Environmental preparation 21
2.2 Login to Web Interface 21
2.2.1 Modify IP Address 21
2.2.2 Login to Web 22
3 Network Management Function 25
3.1 Main Page 25
3.1.1 Function Menu 25
3.1.2 Help Document 27
3.2 Device Info 28
3.3 Serial Port Configuration 28
3.3.1 Modbus Gateway Configuration 28
3.3.2 Modbus Function 37
3.3.2.1 Master Mode (Modbus Master) 38

3.3.2.2 Slave mode (Modbus Slave) 41
3.3.2.3 Slave Address Mapping 45
3.3.2.4 Modbus Slave Pre-read 46
3.3.2.5 Modbus Feature Function 46
3.3.3 RealCOM Function Use 47
3.3.4 Httpd Client Functions 49
3.3.5 WebSocket Client Function 51
3.3.6 MQTT Function 52
3.3.7 JSON Function 54
3.3.7.1 JSON Up Send 54
3.3.7.2 JSON Down Send 58
3.4 Serial port information 59
3.5 Network Address 60
3.5.1 Network Address 60
3.6 Security Management 61
3.6.1 User Password 62
3.6.2 Access Control 63
3.6.3 Filter Settings 64
3.7 System Information 64
3.8 Log Management 65
3.9 Alarm Management 65
3.10 Port Management 66
3.11 System Management 67
4 Principles Overview 71
4.1 Modbus Communication Principle 71
4.2 Transmission Mode 72
4.2.1 Modbus RTU Mode 72
4.2.2 Modbus ASCII Mode 73
4.2.3 Modbus TCP Mode 74
4.3 Modbus Basic Architecture Diagram 76
4.4 Modbus Parameter Settings 78
4.4.1 Initial Delay 78
4.4.2 Modbus TCP Exception 78
4.4.3 Modbus Reponse Timeout 78
4.4.4 Interval Time 78
4.4.5 Frame Interval Delay 79
4.4.6 Slave Address Mapping 79
5 Maintenance and Service 81

5.1 Internet Service 81
5.2 Technical Support Phone Service 81
5.3 Product Repair or Replacement 81

1 Product Introduction

1.1 Product overview

The Cgate3202, Cgate3204, and Cgate3208 series are dual-port rail-mounted industrial Modbus gateways that support protocols such as serial to UDP, TCP, Modbus, HTTPD, WebSocket, MQTT, OPC UA, IEC104, and HJ212, enabling conversion between Modbus RTU/ASCII and Modbus TCP protocols. Additionally, the collection end supports DLT645, IEC101, IEC103, and IEC104 collection protocols. This series offers 2 100Mbps Ethernet ports and 2/4/8 RS232 or RS485 serial ports, adopting a standard DIN rail mounting method to meet various network site requirements.

This series of Modbus gateways specifically includes the following models:

Standard Model	100M Ethernet Port	RS232/485	RS232	RS485	Power Range
Cgate3202	2	2	/	/	Dual DC9~60V
Cgate3204-4D232	2	/	4	/	
Cgate3204-4D485	2	/	/	4	
Cgate3208-8D232	2	/	8	/	Dual DC12~48V
Cgate3208-8D485	2	/	/	8	

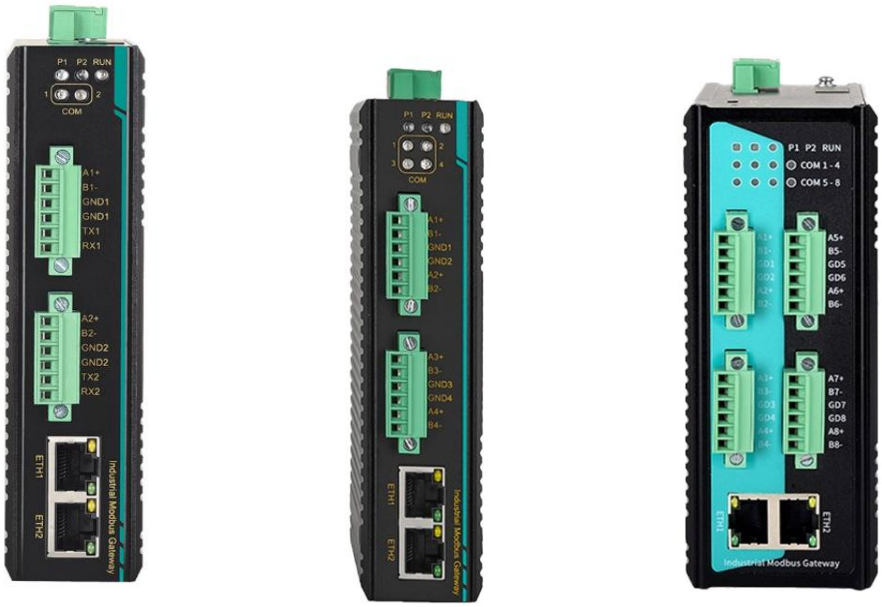
1.2 Product Features

- High-performance CPU processing capability, Cortex-A7 core, with a maximum operating frequency of 800MHz
- Dual Ethernet ports can be configured as two independent network segments or as cascading ports
- Serial port supports baud rates from 300bps to 460800bps
- Support UDP Client/Server and UDP Multicast modes, enabling point-to-point, point-to-multipoint, or multipoint-to-multipoint communication via the UDP protocol, fast and efficient
- Support TCP Client/Server mode, establishing session connections through the TCP protocol. TCP Client supports up to 16 session connections, while TCP Server support up to 32 session connections, and supports dynamic modification of serial port baud rates and other communication parameters via RFC2217 commands
- Support Pair Connection Master/Slave mode, allowing devices to be used in pairs, with simple operation
- Support Modbus RTU/ASCII Master/Slave mode, enabling conversion between Modbus TCP and Modbus RTU/ASCII protocols
- Support RealCOM_MCP/CCP/MW mode, mapping the network to local COM, for seamless connection
- Support HTTPD Client mode, allowing GET or POST operations with an HTTPD server
- Support WebSocket Client mode, enabling bidirectional communication with a WebSocket server
- Support OPC UA Server mode, achieving protocol conversion between Modbus RTU, DLT645, IEC101, IEC103, and OPC UA. Through the OPC Unified Architecture, it provides secure, reliable, and efficient data transmission, enabling interoperability

between devices with different protocols.

- Support IEC104 Server mode, enabling Modbus RTU, Modbus TCP, DLT645, IEC101, and IEC103.
- Convert to IEC104 protocol, providing higher-level remote monitoring and management capabilities.
- Support HJ212 Client mode, enabling conversion between Modbus RTU and HJ212 protocols for environmental monitoring device access.
- Support Modbus TCP mode, enabling conversion between DLT698, IEC101, IEC103, IEC104, and Modbus TCP protocols, facilitating data interaction between various power and meter devices and upper computer software that supports Modbus protocol.
- Support multiple packetization mechanisms, converting serial data to Ethernet packets based on data length or time to meet different network real-time requirements.
- Support frame header and footer mode, allowing the serial port to filter data frames based on the starting and ending bytes of the frame.
- Support registration packets and heartbeat packets for connection verification and connection status detection.
- Support Modbus virtual IDs, mapping Modbus slave device's real ID to a virtual ID for data communication, thus avoiding duplicate slave device IDs.
- Support various data encryption algorithms such as DES, 3DES, AES, RC4, and BlowFish to ensure data security.
- Support SSL (TLS1.0/1.1/1.2) connection encryption, one-way / two-way certificate verification, ensuring connection security.
- Support HTTP/HTTPS/SSH/TELNET access control, along with IP/MAC address filtering.
- Support event alarms for device reboot, login events, configuration changes, password changes, etc., with support for email and SNMP alarm methods.
- Support serial communication parameters, working modes, and statistics for sent and received frames.
- Support local log storage, network storage, and serial log output.
- Support serial/device reboot, factory reset, device upgrade, and NTP client.

1.3 Product Display



1.4 Specifications

Software Functions	
Network Protocol	IP, TCP, UDP, DNS, ARP, SNMP Trap, SSH, ICMP, HTTP, HTTPS, DHCP Client, RFC2217, NTP, SMTP, TELNET
IP Acquisition Method	Static IP/DHCP
Domain Name Resolution	Support
User Configuration	Web Configuration/CONSOLE Simple Network Parameter Configuration
Simple Transparent Transmission Mode	UDP Client / UDP Server / UDP Multicast / TCP Client / TCP Server / RealCOM / Pair Connection
Modbus	Modbus RTU /ASCII to Modbus TCP
Serial port packaging mechanism	Time and length can be set; maximum packaging length 1460 bytes
Data encryption	DES/3DES/AES/RC4/BlowFish
SSL encryption	TLS1.0/TLS1.1/TLS1.2
TCP Server connection	Single serial port supports up to 32 TCP Client connections
TCP Client Connect	Single serial port supports up to 16 connections TCP Server
Network Buffer	Send:16Kbyte; Receive:16Kbyte
Serial Port Buffer	Send:16Kbyte; Receive:16Kbyte
Heartbeat Packet	Supports TCP Keepalive mechanism, customizable heartbeat packet content
Registration Packet	Customizable registration packet content
RFC2217	Support
Serial Port Forwarding	The user can choose the data forwarding rules between the current serial port and other serial ports.
Modbus Address Mapping	This function is only effective in Modbus RTU Slave and Modbus ASCII Slave modes, and is related to Modbus Slave. The pre-reading of the machine and the Modbus slave address mapping function are mutually exclusive.
HTTPD Client	Support

WebSocket Client	Support
MQTT Client	Support
OPC UA Server	Support Modbus RTU, DLT645, IEC101, IEC103 to OPC UA protocol
IEC104 Server	Supports Modbus RTU, Modbus TCP, DLT645, IEC101, IEC103 to IEC104 protocol
HJ212 Client	Supports Modbus RTU to HJ212 protocol
Modbus TCP	Supports DLT698, IEC101, IEC103, IEC104 to Modbus TCP
JSON	Supports JSON functionality in HTTPD Client, WebSocket Client, and MQTT Client modes
RealCOM	Supports working modes of Moxa, Kanghai, etc.
Average transmission delay	<10ms
Supporting Software	Network Management Configuration Tool, Virtual Serial Port Software, MixView, MaxView

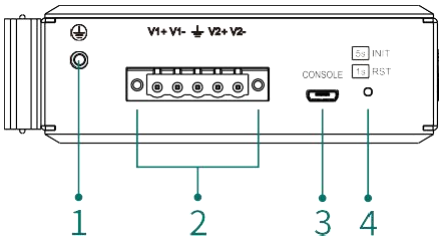
Interface Specifications	**Cgate3202**	**Cgate3204-4D232**	**Cgate3204-4D485**	**Cgate3208-8D232**	**Cgate3208-8D485**
100 M Ethernet	2 ports 10/100Base-T(X) adaptive RJ45 port, supports full/half duplex, auto MDI/MDI-X, supports 1.5kVelectromagnetic isolation protection				
Serial port	2 ports RS232/485	4 ports RS232	4 ports RS485	8 ports RS232	8 ports RS485
	Connection method:3.81mm pitch6PINterminal block Baud rate:300bps-460800bps Data bits: 7bit, 8bit Stop bits: 1bit, 2bit Parity Bit: None, Odd, Even Serial Port Isolation: 2kVAC/3kVDC Isolation Protection				
CONSOLE Port	1 Channel CONSOLE Port, compliant with USB2.0 specification, using Micro-B USB2.0 socket				
Button	Supports one-click restart and factory reset				
Indicator Light	Power indicator, running indicator, port speed and connection activity indicator, serial port indicator				
Power Parameters					
Power Input	DC9~60V, dual power redundancy			DC12~48V, dual power redundancy	
Full load power consumption (DC24)	＜1.8W	＜2W	＜1.8W	＜2.6W	＜2W
Connection method	5-position 5.08mm pitch locking terminal block				
Power protection	Non-polar				

Mechanical parameters		
Dimensions	140×35×100 (mm) (excluding DIN rail)	140×54×110 (mm) (excluding DIN rail)
Installation Method	35mm Standard DIN Rail Mounting	
Enclosure Protection	IP40	
Weight	Approximately 0.45kg	Approximately 0.6kg
Working Environment		
Operating Temperature	-40 ℃ ~ +85 ℃	
Storage Temperature	-40 ℃ ~ +85 ℃	
Relative Humidity	5%~95% (non-condensing)	
Industry Standard		
EMC	IEC 61000-4-2 (ESD): Level 4 (Contact discharge ± 8kV, Air discharge ± 15kV) IEC 61000-4-5 (Surge): Level 4 (Power supply: Common mode ± 4kV, Differential mode ± 2kV; RS485: Common mode ± 4kV, Differential mode ±2kV; Network port: Common mode ± 6kV, Differential mode ± 2kV) IEC 61000-4-4 (EFT): Level 4 (Power supply: ± 4kV; Communication port: ±2kV) *Cgate3202 Air discharge ±8kV, ESD is Level 3; other models meet the above ESD Level 4	

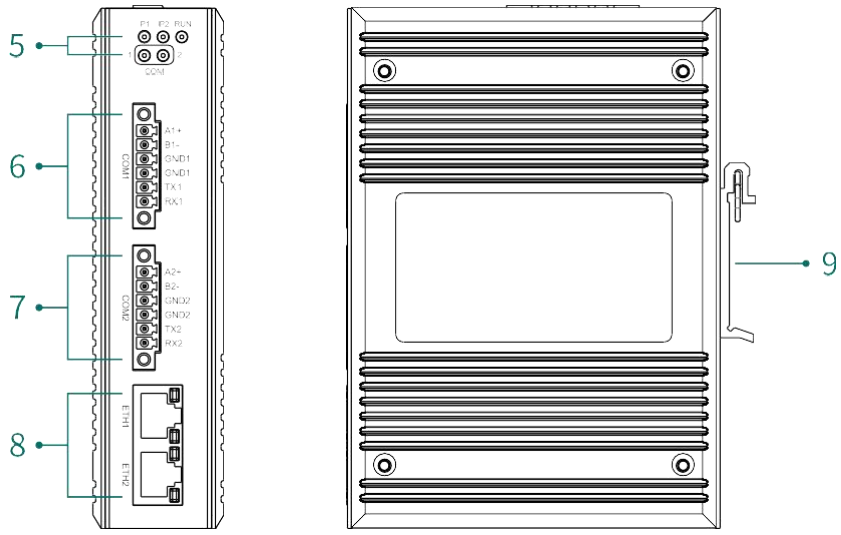
1.5 Interface Introduction

1.5.1 Cgate3202 View

Top View



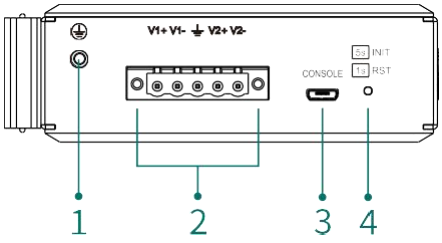
Front View and Right View



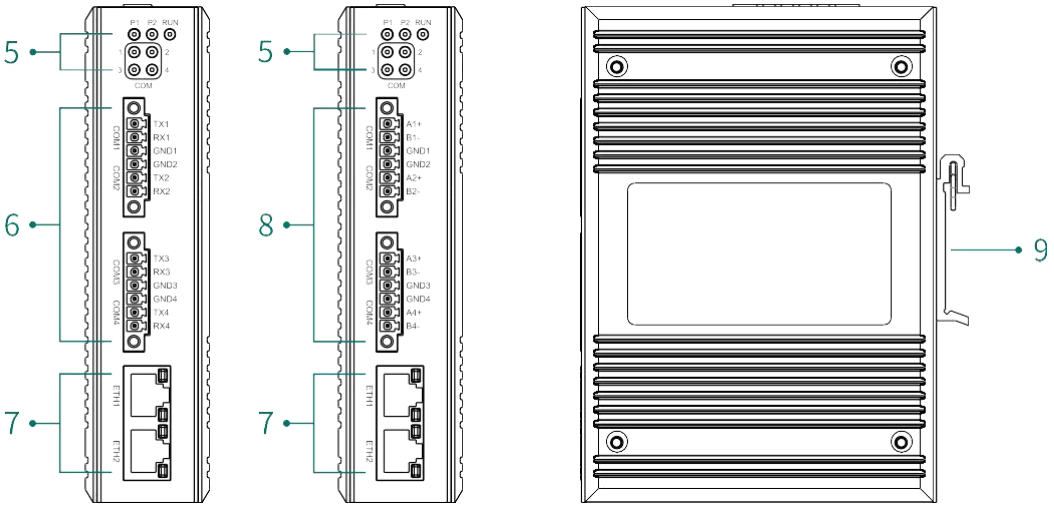
1. Grounding Screw
2. Power Input Terminal
3. CONSOLE Port
4. Restart or Factory Reset Button (RST/INIT)
5. Indicator Light
6. RS232/485 Serial Port (COM1)
7. RS232/485 Serial Port (COM2)
8. 100 Megabit Ethernet Port (ETH1-ETH2)
9. 35mm DIN-Rail Card Slot

1.5.2 Cgate3204 Series View

Top View



Front View and Right View

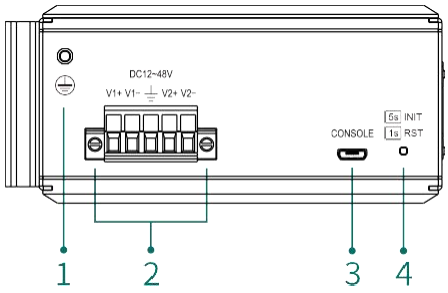


Cgate3204-4D232 Cgate3204-4D485

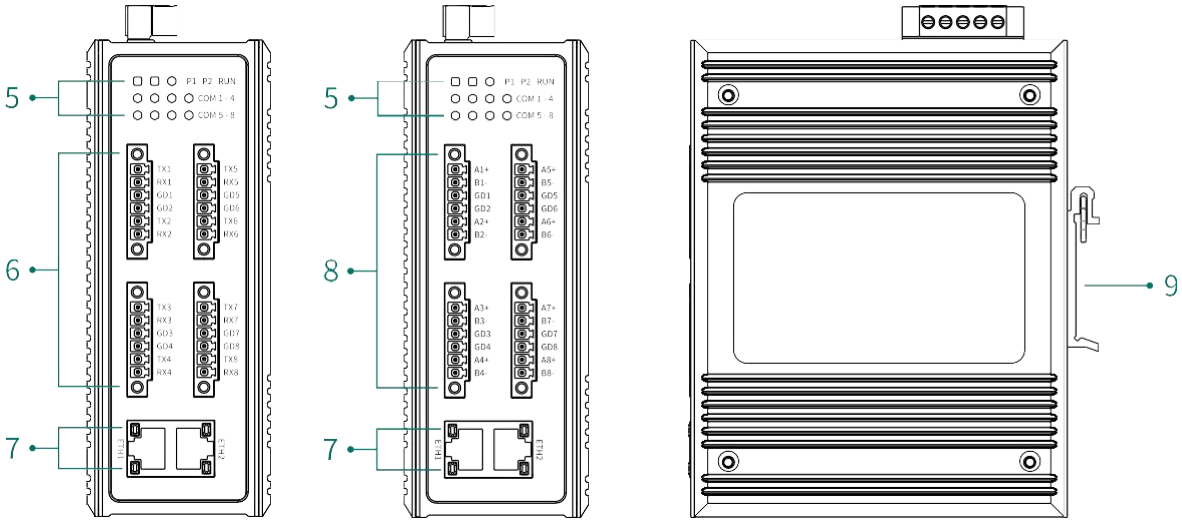
1. Grounding Screw
2. Power Input Terminal
3. CONSOLE Port
4. Restart or Factory Reset Button (RST/INIT)
5. Indicator Light
6. RS232Serial Port (COM1-COM4)
7. 100 Megabit Ethernet Port (ETH1-ETH2)
8. RS485Serial Port (COM1-COM4)
9. 35mm DIN-Rail Card Slot

1.5.3 Cgate3208 Series View

Top View



Front View and Right View



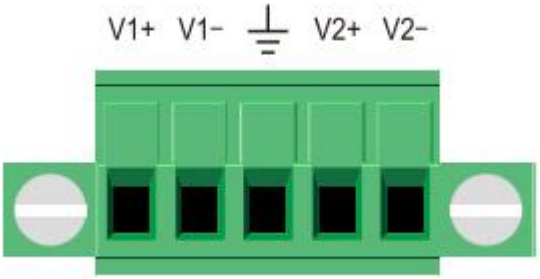
Cgate3208-8D232 Cgate3208-8D485

1. Grounding Screw
2. Power Input Terminal
3. CONSOLE Port
4. Restart or Factory Reset Button (RST/INIT)
5. Indicator Light
6. RS232Serial Port (COM1-COM8)
7. 100 Megabit Ethernet Port (ETH1-ETH2)
8. RS485Serial Port (COM1-COM8)
9. 35mm DIN-Rail Card Slot

1.5.4 Power Interface

Uses 5-position 5.08mm pitch terminal blocks, supporting dual power redundancy input:

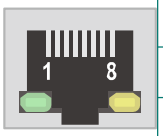
- Cgate3202/3204 voltage input range is DC9~60V.
- Cgate3208 voltage input range is DC12~48V.
-



1.5.5 Ethernet Interface

Provides 2 channels of 10/100Base-T(X) adaptive fast Ethernet ports, using RJ45 connectors, supporting full duplex/half duplex mode adaptation,

MDI/MDI-X auto-detection. RJ45 interface pin definitions are shown in the table below.

Appearance	Pin Number	Pin Definition	Description
	1	TXD+	Ethernet interface transmit data positive terminal.
	2	TXD-	Ethernet interface transmit data negative terminal.
	3	RXD+	Ethernet interface receive data positive terminal.
	6	RXD-	Ethernet interface receive data negative terminal.
	4、5、7、8	–	Empty.

1.5.6 RS232/485 Serial Port (Cgate3202)

Cgate3202 provides 2 channels of RS232/485 serial ports (COM1-COM2), each serial port supports either RS232 or RS485. It uses 2 six-position 3.81mm pitch locking terminal blocks. The pin definitions for the first terminal (COM1) are shown in the table below,

The pin definitions for the second terminal (COM2) are similar.

Appearance	Pin Number	Pin Definition	Description
	1	A1+	First serial port RS485 differential signal A+
	2	B1-	First serial port RS485 differential signal B-
	3	GND1	First serial port RS485 signal ground
	4	GND1	First serial port RS232 signal ground
	5	TX1	Serial Port 1 RS232 Transmit Signal TXD

	6	RX1	Serial Port 1 RS232 Receive Signal RXD

1.5.7 RS232 Serial Port (Cgate3204-4D232)

Cgate3204-4D232 provides 4 channels of RS232 serial ports (COM1-COM4), using 2 six-position 3.81mm pitch locking terminal blocks. Each terminal supports 2 channels of RS232 serial ports. The pin definitions for the first terminal (COM1-COM2) are shown in the table below, and the pin definitions for the second terminal (COM3-COM4) are similar.

Appearance	Pin Number	Pin Definition	Description
	1	TX1	Serial Port 1 RS232 Transmit Signal TXD
	2	RX1	Serial Port 1 RS232 Receive Signal RXD
	3	GND1	First serial port RS232 signal ground
	4	GND2	Serial Port 2 RS232 Signal Ground
	5	TX2	Serial Port 2 RS232 Transmit Signal TXD
	6	RX2	Serial Port 2 RS232 Receive Signal RXD

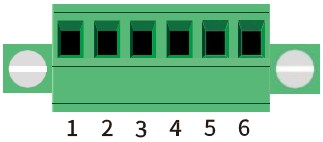
1.5.8 RS232 Serial Port (Cgate3208-8D232)

Cgate3208-8D232 provides 8 RS232 serial ports (COM1-COM8), using 4 six-position 3.81mm pitch locking terminal blocks, with each terminal supporting 2 RS232 serial ports. The pin definitions for the first terminal (COM1-COM2) are shown in the table below, and the pin definitions for the other terminals (COM3-COM8) are similar.

Appearance	Pin Number	Pin Definition	Description
	1	TX1	Serial Port 1 RS232 Transmit Signal TXD
	2	RX1	Serial Port 1 RS232 Receive Signal RXD
	3	GD1	First serial port RS232 signal ground
	4	GD2	Serial Port 2 RS232 Signal Ground
	5	TX2	Serial Port 2 RS232 Transmit Signal TXD
	6	RX2	Serial Port 2 RS232 Receive Signal RXD

1.5.9 RS485 Serial Port (Cgate3204-4D485)

Cgate3204-4D485 provides 4 RS485 serial ports (COM1-COM4), using 2 six-position 3.81mm pitch locking terminal blocks, with each terminal supporting 2 RS485 serial ports. The pin definitions for the first terminal (COM1-COM2) are shown in the table below, and the pin definitions for the second terminal (COM3-COM4) are similar.

Appearance	Pin Number	Pin Definition	Description
	1	A1+	First serial port RS485 differential signal A+
	2	B1-	First serial port RS485 differential signal B-
	3	GND1	First serial port RS485 signal ground
	4	GND2	Serial Port 2 RS485 Signal Ground
	5	A2+	Serial Port 2 RS485 Differential Signal A+
	6	B2-	Serial Port 2 RS485 Differential Signal B-

1.5.10 RS485 Serial Port (Cgate3208-8D485)

Cgate3208-8D485 provides 8 RS485 serial ports (COM1-COM8), using 4 six-position 3.81mm pitch locking terminal blocks. Each terminal supports 2 RS485 serial ports. The pin definitions for the first terminal (COM1-COM2) are shown in the table below, and the pin definitions for the other terminals (COM3-COM8) are similar.

Appearance	Pin Number	Pin Definition	Description
	1	A1+	First serial port RS485 differential signal A+
	2	B1-	First serial port RS485 differential signal B-
	3	GND1	First serial port RS485 signal ground
	4	GND2	Serial Port 2 RS485 Signal Ground
	5	A2+	Serial Port 2 RS485 Differential Signal A+
	6	B2-	Serial Port 2 RS485 Differential Signal B-

1.5.11 RS485 Termination Resistor

The RS485 interface is equipped with a 120-ohm termination resistor at the factory. To change the RS485 termination resistor, please open the casing. The connection status is shown in the table below.

Termination Resistor Connected	Termination Resistor Not Connected

The jumper cap inserted on the "ON" side indicates that the termination resistor is connected, while inserting it on the other side indicates that the termination resistor is not connected.

1.5.12 Button

Restart: Press and release the button within 1 second. The system will reset, the RUN light will go out, and the system will return to normal after startup.

Restore Factory Settings: Press and hold the button for more than 5 seconds. The RUN light will blink (once every 0.2 seconds). At this point, release the button, and the parameters will be restored to factory settings, and the system will reset.

1.5.13 CONSOLE Port

Provide 1 USB debugging port, using Micro-USB interface. Connect one end of the USB adapter cable to the computer and the other end to the device. Note that the computer needs to install the corresponding CH340 driver.

CONSOLE debugging serial port communication parameters are as follows:

- Baud rate: 115200
- Data bits: 8
- Parity: none
- Stop bits: 1
- Flow control: none

1.5.14 Grounding Screw

After crimping one end of the grounding wire to the cold-pressed terminal, secure it with the grounding screw at the grounding hole of the casing. The other end of the grounding wire should be connected to the ground, and the cross-section of the grounding wire should not be less than 2.5mm².

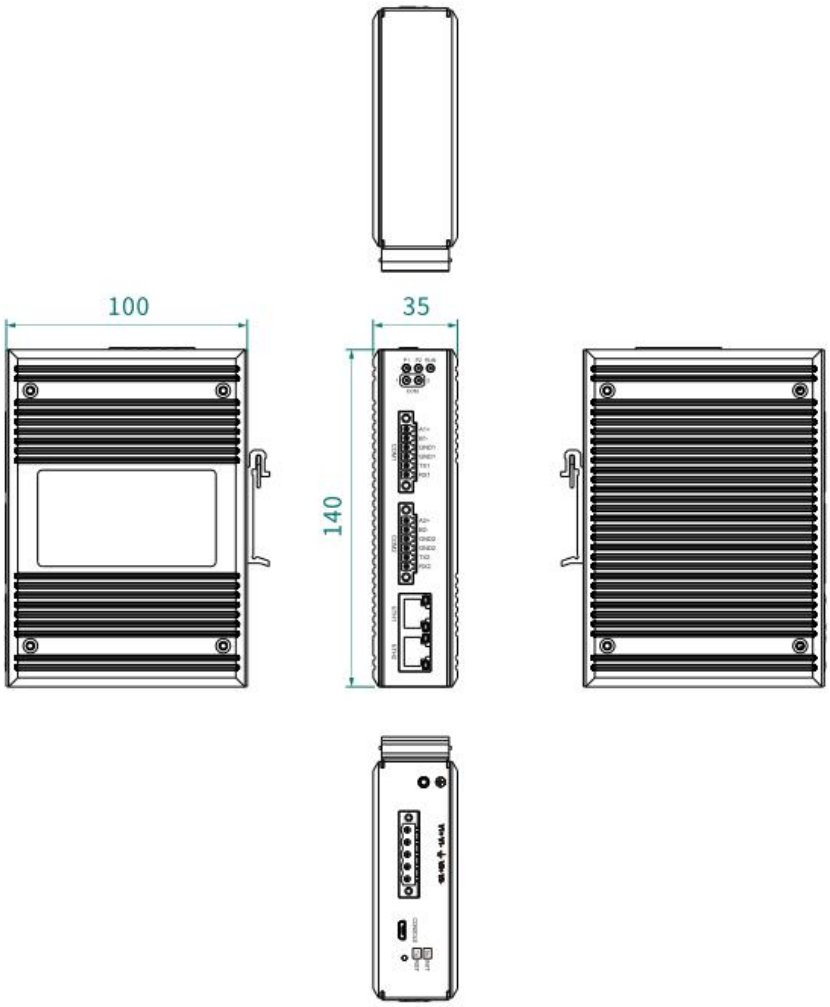
1.5.15 Indicator Light

Indicator Light	Status	Definition
P1-P2	Constant Light	Power supply is normal.
	Off	Power is not connected or there is a power supply issue.

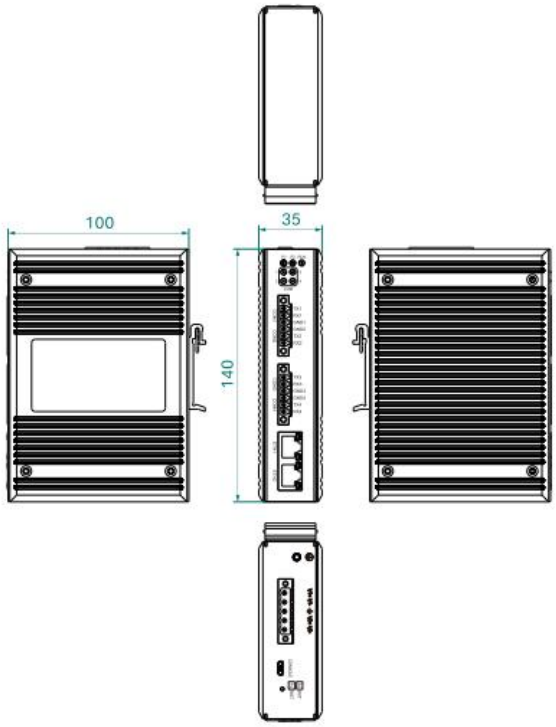
RUN	Flashing	System is running normally.
	Constant Light/Off	System is running abnormally.
COM	Flashing	The serial port is sending and receiving data.
	Off	The serial port has no data transmission.
SPD (Network port yellow light)	Constant Light	100M Working mode.
	Off	10M Working mode or port not connected.
L/A (Network port green light)	Constant Light	Ethernet port has established a valid network connection.
	Flashing	Data is being transmitted over the Ethernet port.
	Off	The Ethernet port has not established a valid network connection.

1.6 Installation Dimensions

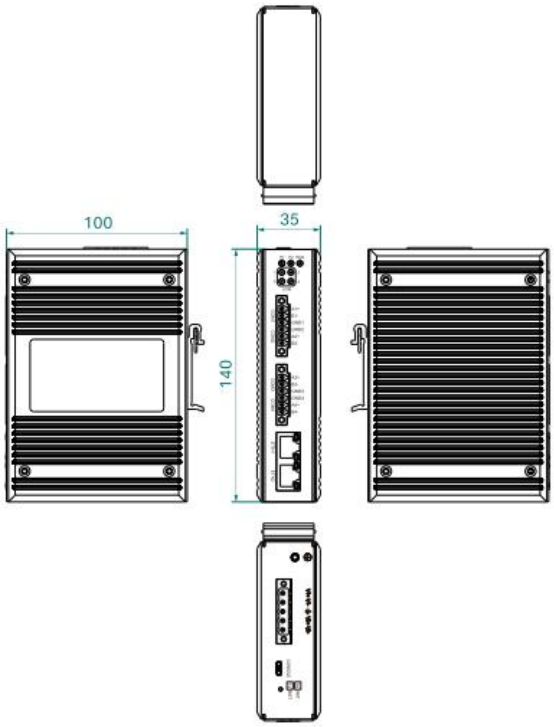
Unit: mm (First Angle Projection)



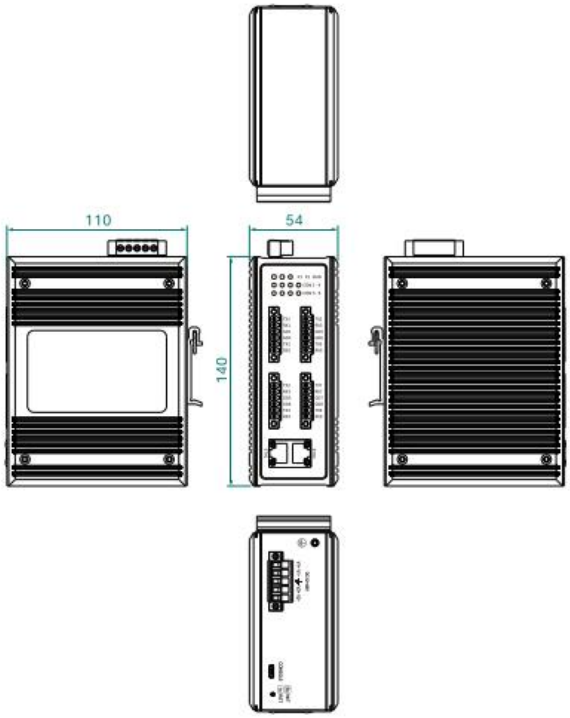
Cgate3202



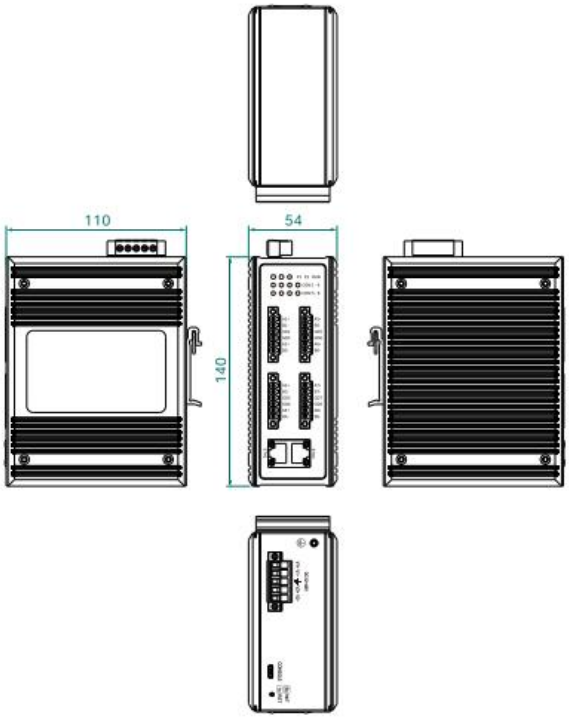
Cgate3204-4D232



Cgate3204-4D485



Cgate3208-8D232



Cgate3208-8D485

2 Quick Start

CGate3202, CGate3204, CGate3208 Modbus gateways have a built-in web server, which provides a convenient way to access and configure Modbus gateways. Users can use IE, Firefox or Google Chrome to access it.

This chapter is a quick introduction toCGate3202, CGate3204 and CGate3208 Modbus gateway products. It is recommended that users systematically read this chapter and follow the instructions to have a basic understanding of the product. For specific function details and instructions, please refer to the subsequent chapters.

If you need relevant information about the product, you can download the corresponding product manual from the official website link: http://www.Come-Star.com

2.1 Environmental preparation

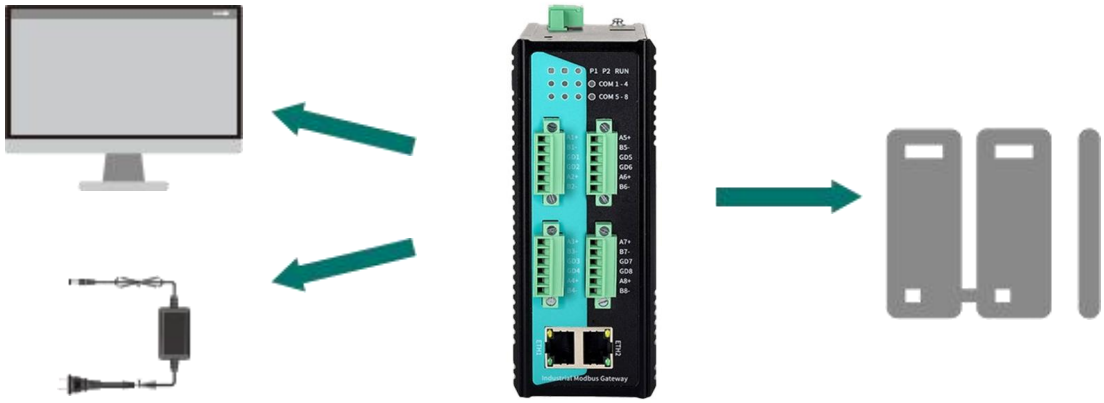


Figure 2-1 Hardware connection

2.2 Login to Web Interface

2.2.1 Modify IP address

When accessing the Modbus gateway via the web, the IP addresses of the Modbus gateway and the PC must be on the same subnet. Therefore, you must change the PC's IP address to ensure it is on the same local area network as the Modbus gateway. Windows users should follow these steps:

Start → Control Panel → Network and Internet Connections → Network Connections → Local Area Connection → Properties → Internet Protocol (TCP/IP)

The default IP address for this model of Modbus gateway is: 192.168.16.253. Set the PC's IP address to: 192.168.16.X

(where X is any valid value between 2 and 253, excluding 253). The specific Windows system configuration screen is shown in Figure 2-2.

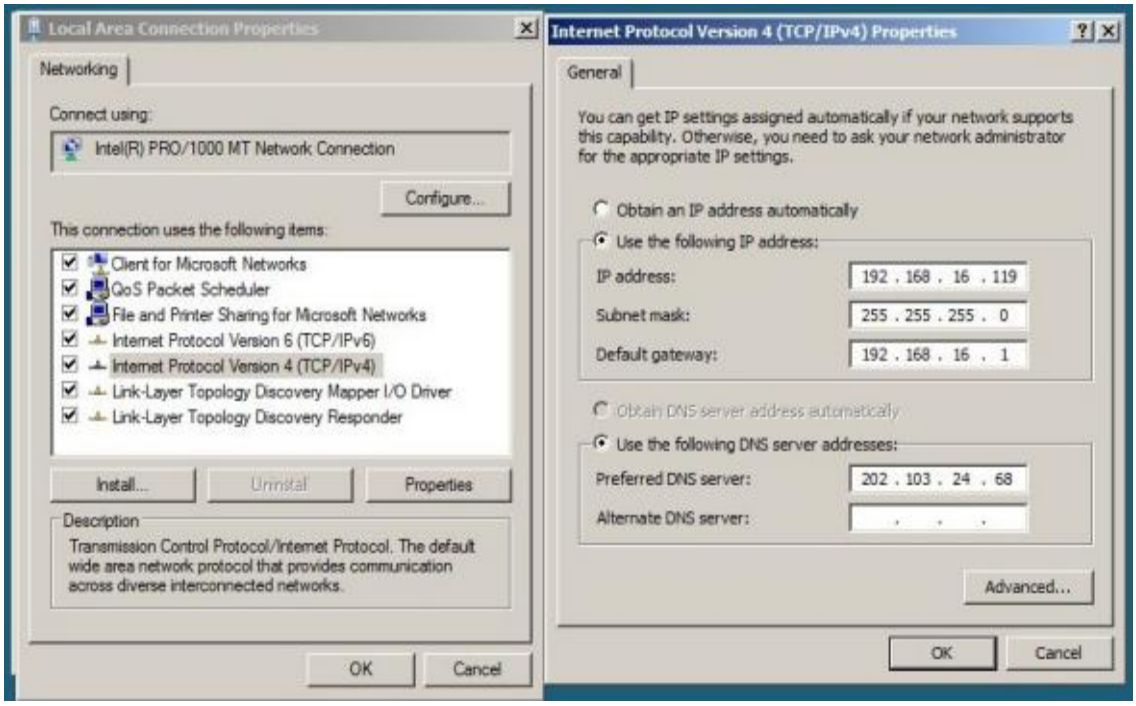


Figure 2-2 IP setting interface in Windows environment

After changing the IP address of the PC, you can access the web page of the Modbus gateway using the default IP address 192.168.16.253 and perform related configuration operations.

2.2.2 Login to Web

Open the browser and enter the default IP address of the Modbus gateway in the address bar, as shown in Figure 2-3.

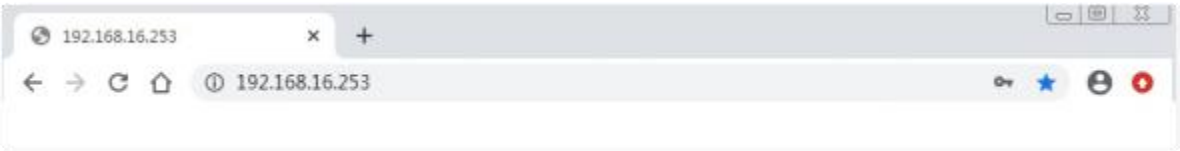


Figure 2-3 Enter the IP address interface in the address bar

After pressing the Enter key, a dialog box will pop up as shown in Figure 2-4, prompting the user to enter their username and password.

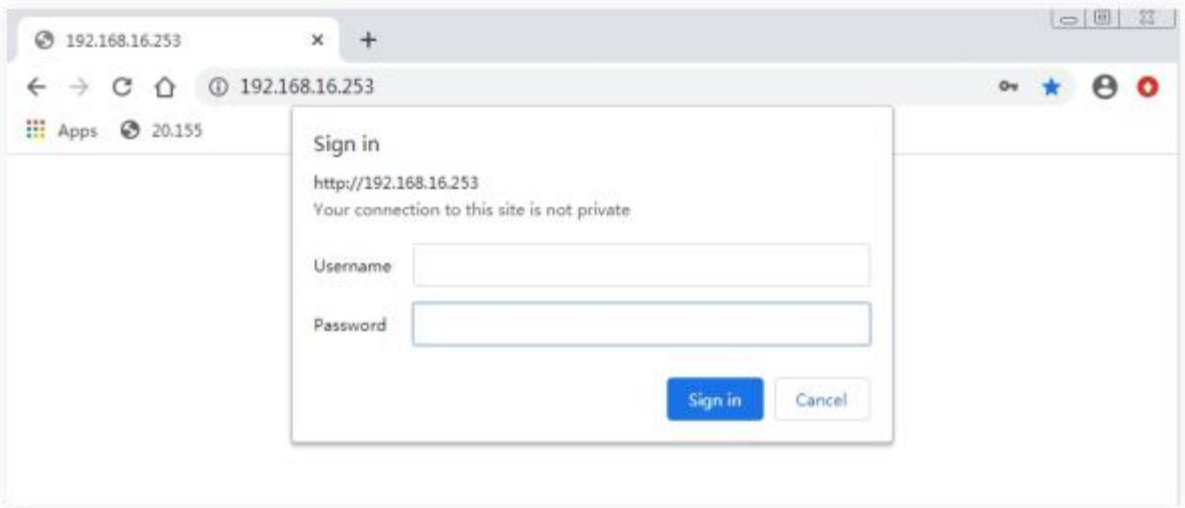


Figure 2-4 Input user name and password interface

The login users of the Modbus gateway are divided into three types:

- The first type is a regular user, with both the username and initial password as "admin", which is used for normal access to the Web.
- The second type is a guest, with both the username and password as "none". After logging in, they can only view the current configuration of the Modbus gateway and cannot make any changes.
- The third type is an administrator, with the username as "admin" and the password as the last six digits of the MAC address of the Modbus gateway's eth2 network card. When we forget the regular user's password, we can log in with the administrator account and modify the local password.

After entering the username and password, click "OK". The server will perform authentication, and upon successful authentication, it will enter the main page of the Web server, as shown in Figure 2-5.

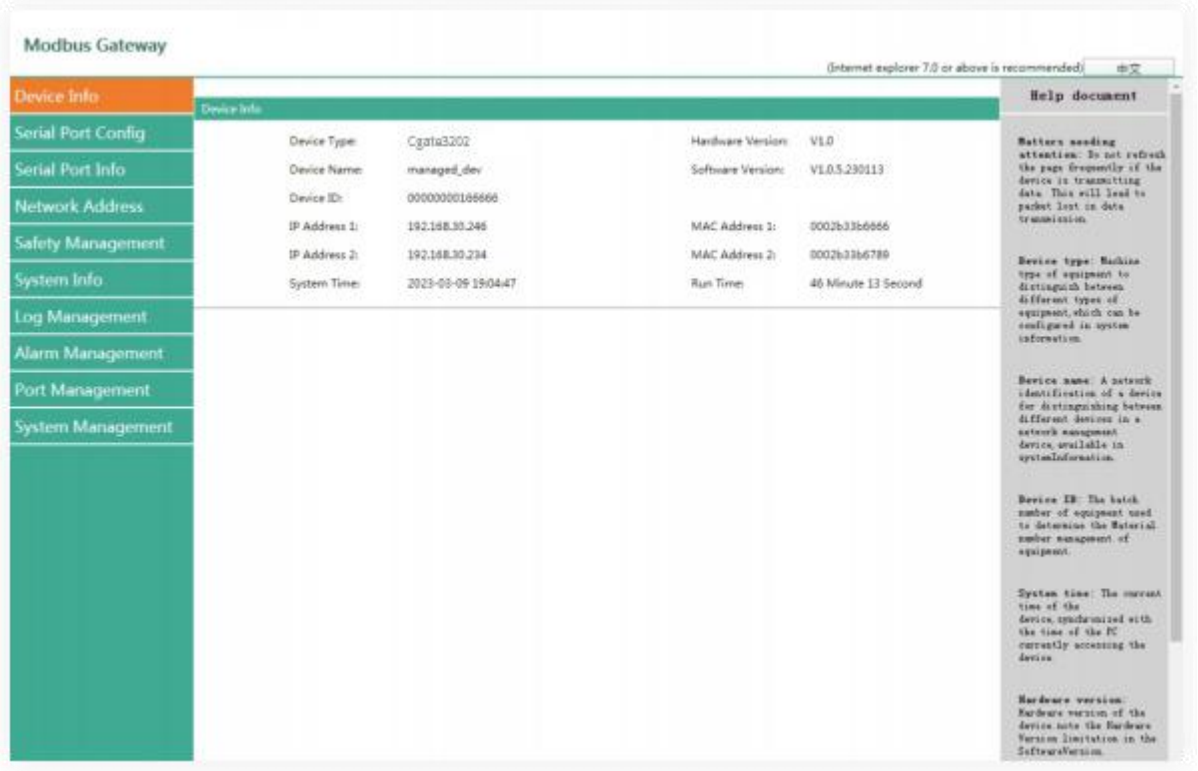


Figure 2-5 The main interface of the WEB server

The Modbus gateway has been extensively tested with IE9.0 and above, Firefox, and mainstream Google browsers, all of which work normally. However, it is recommended to use Google Chrome during device upgrades. Additionally, the Modbus gateway does not support access to the web via Safari browser.

3 Network Management Function

The interface and configuration of all functions below are based on the physical product.

3.1 Main Page

The main page can be roughly divided into three areas: the upper area displays the LOGO, the lower left area is the function menu area, the middle area is the main function display area, and the lower right area is the help document area, as shown in Figure 3-1

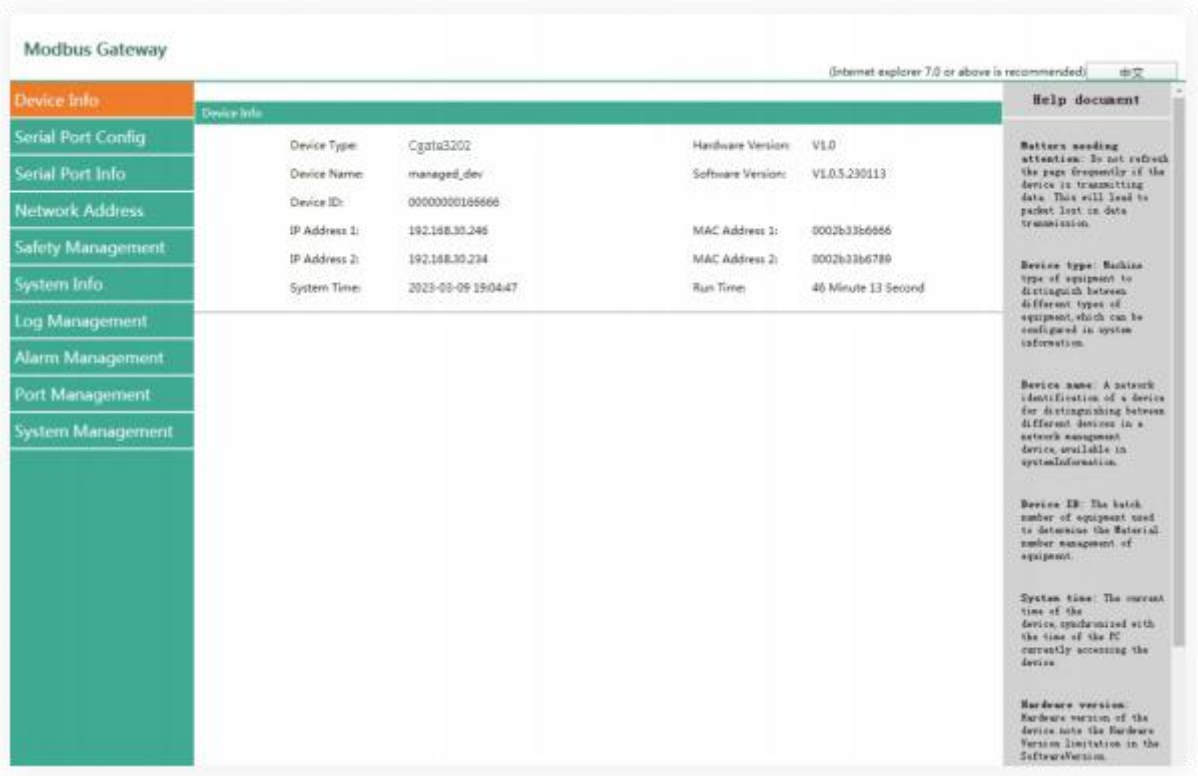


Figure 3-1 Main page

3.1.1 Function Menu

The left side of the webpage is the function menu area, displaying all configurable software functions of this Modbus gateway. The function menu includes device information, serial port configuration, serial port information, network address, security management, system information, log management, alarm management, port management, and system management, with several sub-functions under each menu. Their functions are shown in Table 3-1.

Table 3-1 Menu Function Description Table

Menu Bar	Function	Description
Device Info	Device Information	Display device information, such as: name, number,

		software version, IP address, etc.
Serial Port Confi	Modbus Gateway Configuration	Configure basic information for each port, such as: serial port number.
	Serial Port Parameters	Configure the working mode, baud rate, packet length, etc. for each serial port.
	Network Parameters	Configure the local port range and heartbeat interval for each port, etc.
	Number of network connections	Configure the destination address and remote port for each port.
	Data encryption	Configure the encryption method, padding method, etc. for data transmission.
	Serial Port Forwarding	Users can choose the data forwarding rules between the current serial port and other serial ports.
Serial Port Info	Serial Port Information	Display the serial port number as well as the total number of received and sent data.
	Network Connection Mode	Display the working mode of the serial port / local port and other information.
Network Address	Network Address	Configure the device's network address type, IP address, subnet mask, default gateway, and DNS server address.
Security Management	User Password	Configure username and password.
	Access Control	Configure the enabling and disabling of Web, SSH, and Telnet console functions.
	Filtering Settings	Configure network data filtering for IP and MAC addresses.
System Info	System Information	Configure device model, device name, etc.
Log Management	Local Storage	Log records are stored internally on the device, allowing users to export logs to a PC.
	Network Storage	Transfer logs to a remote UDP server.
	Serial Port Log Output	Output logs to a PC via the specified serial port.
Alarm Management	Sender Email Settings	Configure sender server, sender email, authorization code, and other information.
	Recipient Email Address	Configure recipient email.
	SNMP Settings	Configure SNMP related parameters.
	Alarm Event Settings	Configure whether to enable the corresponding event alarm function.
Port Management	Port Restart	Configure the specified serial port to restart.
System Management	Device Restart	Configure the device's restart function.
	Restore Factory Settings	Restore the factory settings of the device.
	Device upgrade	Configuration upgrade file for the device.
	No data device restart	Configuration device restarts if there is no data transmission over the network and serial port for a certain period. .
	NTP Synchronize time	Configure NTP server address and time zone, the device can synchronize time from the network.

		Information.

3.1.2 Help Document

The functional area in the lower right corner is the help documentation. Clicking on any of the main function pages in the lower left will display the help documentation in the corresponding functional area on the main page, as shown in Figure 3-2.

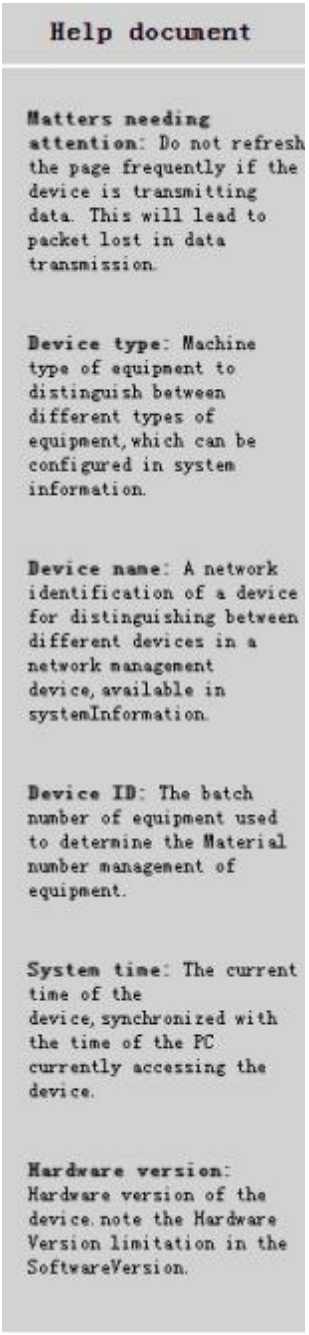


Figure 3-2 Help Documet

3.2 Device Info

The device information section displays specific information about the current device, including device model, hardware version, device name, software version, device number, IP address, MAC address, system time, and uptime. As shown in Figure 3-3.

Figure 3-3 Device Info

- Device Type: The model of the Modbus gateway, which can be customized by the user on the "System Information" page.
- Device Name: The name of the Modbus gateway, which can be customized by the user on the "System Information" page.
- Device ID: The number of the Modbus gateway.
- IP Address 2: The IP address of the Modbus gateway network card 2.
- IP Address 1: The IP address of the Modbus gateway network card 1.
- System Time: The current time of the Modbus gateway, synchronized with the time of the PC accessing the Modbus gateway.
- Hardware version: Modbus The current hardware version of the gateway.
- Software Version: Modbus Gateway's current software version.
- MAC Address 2: MAC address of Modbus Gateway Network Card 2.
- MAC Address 1: MAC address of Modbus Gateway Network Card 1.
- Run Time: The running time of the Modbus Gateway, from the moment the device is powered on to the current time.

3.3 Serial Port configuration

The serial port module includes: serial port server configuration, serial port parameter configuration, network parameter configuration, and network connection number configuration

3.3.1 Modbus Gateway Configuration

The Modbus Gateway product is an industrial-grade gateway device, specifically designed for communication between Modbus TCP devices and Modbus RTU/ASCII devices. This series of products also integrates Modbus TCP, Modbus RTU, and Modbus ASCII protocols, allowing users to easily achieve seamless integration of Modbus Ethernet devices with Modbus serial devices, even in mixed networks of multiple Masters and Slaves. This device also supports data encryption transmission, JSON format upload and download functions. Additionally, users can configure it via WEB, and its simplified design not only enables rapid application but also ensures flexibility for all practical industrial environment applications.

The configurable parameters on the Modbus Gateway configuration page are shown in Figure 3-4.

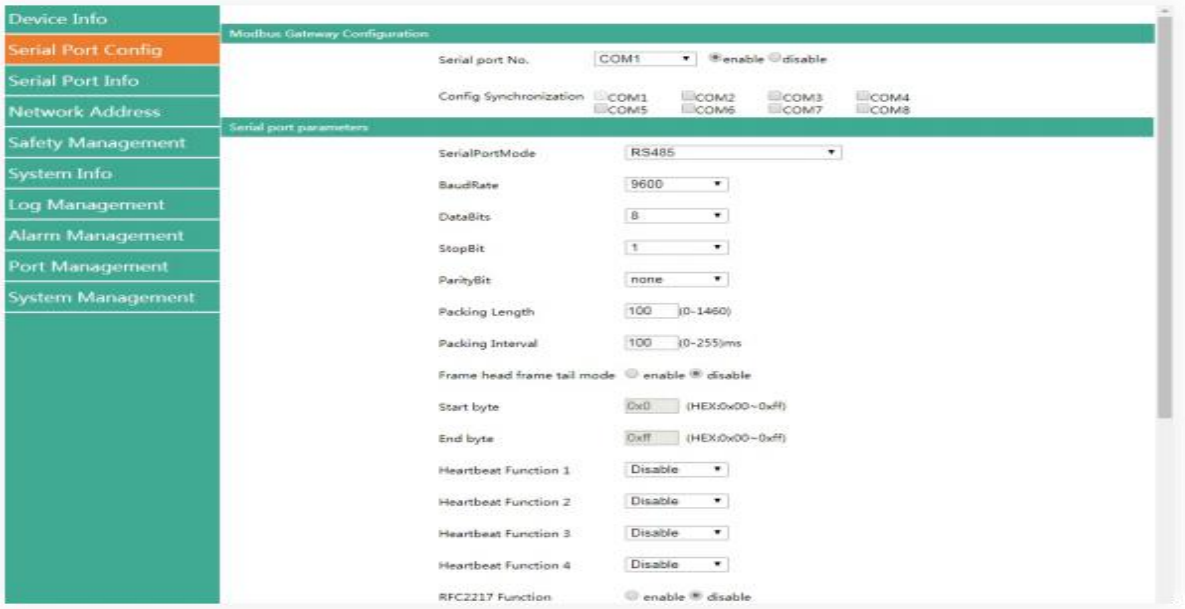


Figure 3-4 Serial Port configuration

The configuration parameters of this interface are detailed in Table 3-2

Table 3-2 Description of Modbus gateway configuration parameters

Parameters	Instructions
Modbus Gateway Configuration	
Serial Port No	Select the current serial port to configure, enable or disable it. Cgate3202 supports 2 serial ports, Cgate3204 supports 4 serial ports, and Cgate3208 supports 8 serial ports.
Configuration Synchronization	The parameters of the current serial port configuration can be synchronized to other serial ports.
Serial Port Parameters	
Serial Port Mode	Select the working mode of the current serial port. • The mode of the Cgate3202 2 serial ports is RS232/485. • The mode of the Cgate3204-4D232 4 serial ports is RS232. • The mode of Cgate3204-4D485 4-port serial port is RS485. • The mode of Cgate3208-8D232 8-port serial port is RS232. • The mode of Cgate3208-8D485 8-port serial port is RS485.
Baud rate	The baud rate for serial communication, measured in bps, can be set to one of the following values: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, and 460800. The factory default is 9600. The baud rate for each serial port is independent; they are set separately and do not affect one another.
Data Bits	Set the number of effective data bits for serial communication, which can be selected as: 7 or 8. The factory default setting is 8 bits.

Stop Bit	Set the stop bit length for serial communication, which can be selected as: 1 or 2. The factory default setting is 1.
Parity Bit	Select a verification method, which includes: None, Odd parity, and Even parity. The factory default setting is None. The verification method for each serial port is also independent of each other, set separately, and does not affect one another.
Packing Length	When the length of data continuously received by the serial port exceeds the value set for packet length, a network transmission will be triggered for this data. Forwarded to the network end. Subpackage length range 0~1460 bytes.
Packing Interval	When the intermittent time of receiving data through the serial port exceeds this value, a network transmission will be triggered regardless of how much data has been received. When transmitted, this data is forwarded to the network port. Subpackage interval range 0~65535ms.
Frame Header Frame tail Mode	After enabling this mode, the serial port will determine the data based on the frame start byte and end byte. The frame header and footer do not match. The data will be discarded.
Start Byte	The range for setting the start byte of the serial port is between the hexadecimal numbers 0x00 and 0xff.
End Byte	The range for setting the end byte of the serial port is between the hexadecimal numbers 0x00 and 0xff.
Registration Packet Function	Select the sending method for the registration packet: • Disable: Disable this function; • LINK: The registration packet is sent only once when the network connection is established; • DATA: The registration packet is filled in front of the serial data each time data is sent from the serial port to the network; this function is only allowed when the network working mode is UDP Client and TCP Client.
Registration Packet Content	The content of the registration package, Hex format is up to 128 bytes, ASCII code format is 64 bytes.
Heartbeat function	Select the method for sending heartbeat packets: • Disable: Disable this feature; • to COM: Heartbeat packet sent to the serial port; • to Network: Heartbeat packet sent to the network; This function is only available in network operating modes: UDP Client, UDP Server, TCP Server, and TCP Client. It is allowed to use.
Heartbeat packet content	The content of the heartbeat packet, Hex format is up to 128 bytes, ASCII code format is 64 bytes.
Heartbeat packet interval	The time interval for sending heartbeat packets, in seconds, ranges from 1 to 65535 seconds, with a default of 30 seconds.
RFC2217 Functionality	Enabling this feature allows users to dynamically modify parameters such as baud rate, data bits, stop bits, and parity bits of the serial port using standard RFC2217 commands over the network.

	This feature is only allowed when the network operating mode is TCP Server or TCP Client.
Serial Port Forwarding	Users can choose the data forwarding rules between the current serial port and other serial ports. • Prohibit means that the data received by the current serial port will not be forwarded to the corresponding serial port; • One-way forwarding means that the data received by the current serial port will be forwarded to the corresponding serial port; • One-way receiving means that the corresponding serial port is allowed to forward data to the current serial port; • Two-way forwarding means that the data received by the current serial port and the data received by the corresponding serial port will be mutually forwarded.
Network Parameters	
Working Mode	This setting supports the following operating modes: UDP Client, UDP Server, UDP Multicast, TCP Client, TCP Server, Modbus RTU Master, Modbus RTU Slave, Modbus ASCII Master, Modbus ASCII Slave, RealCOM_MCP, RealCOM_CCP, RealCOM_MW, Pair Connection Master, Pair Connection Slave, HTTPD Client, WebSocket Client, MQTT Client, OPC UA Server, IEC104 Server, HJ212 Client, Modbus TCP, and 21 other communication methods are available for selection. The factory default setting is UDP Client mode. • Modbus RTU Master: When this device is in the Modbus RTU Master mode, if the Modbus Over TCP function is not enabled, the remote device must operate in Modbus TCP Slave mode; otherwise, if the Modbus Over TCP function is enabled, the remote device must operate in Modbus RTU Slave mode, which supports a maximum of 16 connections. • Modbus RTU Slave: When this device is in the working mode of Modbus RTU Slave, if not enabled For the Modbus Over TCP function, the remote device must operate in Modbus TCP Master mode; otherwise, when enabling the Modbus Over TCP function, the remote device must operate in Modbus RTU Master mode, which supports a maximum of 32 connections. • Modbus ASCII Master: When this device is in the Modbus ASCII Master mode, if not To enable the Modbus Over TCP function, the remote device must operate in Modbus TCP Slave mode; otherwise, to enable the Modbus Over TCP function, the remote device must operate in Modbus ASCII Slave mode, which supports a maximum of 16 connections.

	• Modbus ASCII Slave: When this device is in the Modbus ASCII Slave mode, if not enabled With the Modbus Over TCP function, the remote device must operate in Modbus TCP Master mode; otherwise, when enabling the Modbus Over TCP function, the remote device must operate in Modbus ASCII Master mode, which supports a maximum of 32 connections. • UDP Client: When this device is in UDP Client mode, the remote device is required to operate in UDP. In Server mode. In this mode, you can establish UDP connections with 16 remote devices, and you can configure the IP addresses and port numbers of the remote devices within the page. • UDP Server: When this device is in UDP Server mode, the remote device is required to operate in UDP. In Client mode. This mode supports a maximum of 32 UDP connections, and connections will be aged based on the aging time configured on the web interface. • UDP Multicast: When this device operates in UDP Multicast mode, remote devices must operate. In UDP Multicast mode, this device can join up to 16 UDP multicast groups, which can be accessed on the web. Configure multicast IP address and port number. • TCP Client: When this device is in TCP Client mode, the remote device must operate in TCP Server mode. Both the IP address and port number must be configured, which can be done in the corresponding options for network connection. • TCP Server: When this device is in TCP Server mode, the remote device must operate in TCP Client mode. In this mode, it can accept up to 32 remote TCP Client connections. • RealCOM_MCP, RealCOM_CCP, RealCOM_MW: When the device operates in the mode of RealCOM_MCP, RealCOM_CCP, or RealCOM_MW, the PC needs to install the corresponding virtual serial port software for use. The virtual serial port software maps the device's serial port to the local serial port of the PC, enabling data communication between the original serial port software and the serial port device. The Modbus gateway's one serial port supports a maximum of 32 virtual serial port accesses. • Pair Connection Master: When this device is in Pair Connection Master mode, The remote device must operate in Pair Connection Slave mode, and both its IP address and port number must be configured. This can be done in the options corresponding to the network connection, while the local port number can be ignored and does not need to be configured. • Pair Connection Slave: When this device is in Pair Connection Slave

	mode, it requires remote The device must operate in Pair Connection Master mode. In this mode, it can accept up to 32 remote Pair Connection Master connections. • HTTPD Client: When this device is in HTTPD Client mode, the user needs to specify the remote HTTP The server's address, port, method, and other parameters. The device will submit the data received from the serial port to the HTTP server using either GET or POST. It can also pass the data sent from the HTTP server through to the serial port. • WebSocket Client: When the device is in WebSocket Client mode, the user needs to specify The main parameters of the WebSocket server include the address, port, and method. The time interval for Ping can also be set. This maintains the connection between the device and the server. The device will upload the data received from the serial port to the WebSocket server in hexadecimal format, and it can also pass through the data sent from the server to the serial port. • MQTT Client: When this device is in MQTT Client mode, the user needs to select the server platform type. The model supports optional platforms such as Alibaba Cloud, OneNet, Huawei Cloud, Tencent Cloud, Amazon Cloud, Microsoft Cloud, and other clouds. Then configure the MQTT address, port number, subscription and publishing topics, device secret key, etc. The device will send the data received from the serial port to the cloud platform. Data sent from the cloud platform can also be transmitted to the serial port. • OPC UA Server: When the device is in OPC UA Server mode, the user needs to configure the local port. Set the OPC UA response timeout based on the user-configured table information. The serial port side selects the corresponding acquisition protocol, collects the data from the lower-level device, and then forwards the communication over the network using the OPC UA protocol. • IEC104 Server: When the device is in IEC104 Server mode, the user needs to configure the port number and response Timeout, T3 timeout, public address, K value, W value, and change upload enable switch. According to the user-configured table information and collection protocol, the collected data from the lower machine will be forwarded for communication over the network using the IEC104 protocol. • HJ212 Client: When the operating mode of this device is HJ212 Client, this function allows the device to act as TCP Client, the serial port side collects data using the Modbus protocol

	and uploads the collected data to the HJ212 server using the HJ212 protocol. • Modbus TCP: When the device operates in Modbus TCP mode, this function allows the device to act as Modbus TCP slave, converts serial port protocol collected data into Modbus protocol data, and then uploads the Modbus. TCP Master Station.
Fixed port enable	In TCP Client, Modbus RTU Master, Modbus ASCII Master, Pair Connection Master, HTTPD Client, WebSocket Client, MQTT Client, HJ212 Client mode,
	You can choose a local port as the set value. When the local port is enabled, the connection initiated by this serial port uses a fixed local port. When this function is disabled, the connection initiated by this serial port uses a random local port.
Local Port	The local port on the network connection side.
SSL encryption	The encryption method and authentication method for the network connection. • One-way authentication means that this device only verifies the encrypted authentication certificate of the remote server; • Two-way authentication means that both this device and the remote server need to verify each other's certificates.
Certificate Scheme	Select the encrypted certificate scheme. The user selects the certificate scheme uploaded on the system management page for the network connection of this serial port. Each serial port can use its own certificate scheme, or multiple serial ports can use the same certificate scheme.
Certificate Type	Select the type of encryption certificate. • Allowing expired certificates indicates that this device accepts expired authentication certificates. • Refusing expired certificates means that if the certificate uploaded by the user has expired, the connection will not be successful. • The acceptance of self-signed certificates indicates that this device allows user-signed certificates, while the acceptance of commercial certificates means that this machine only accepts certificates issued by commercial organizations.
Keepalive Heartbeat Interval	When the network operating mode is in TCP mode, TCP heartbeat packets are sent at specified intervals to test the connection. If the connection exists, it will automatically disconnect if it does not exist, with a range of 1 to 6000 seconds.
Keepalive Timeout	When the network operating mode is in TCP mode, detect the current connection and the corresponding serial port idle time. When it exceeds The set value will disconnect the TCP connection.
Idle aging time	When the network operating mode is in UDP Server mode, it detects the idle time of the current connection to the network. If it exceeds the set value, it will release the IP address and port number of this connection, allowing other connections to access. The range is (0, 10-65535) s,

	where 0 means prohibited.
Multicast TTL value	The TTL value of multicast messages can limit the distance that multicast messages are forwarded, with a range of 1-255.
Modbus Over TCP	Modbus (RTU/ASCII) protocol passthrough enabled.
Modbus Response Timeout	Modbus serial port reception timeout, after the Modbus gateway forwards the request to the slave device, within this parameter time If no response is received, it is considered a timeout response.
Modbus ID Filtering	Filtering of Modbus Slave ID Range.
Modbus Initial Delay	After the device is powered on, there is a delay before starting Modbus message communication.
Modbus TCP Exception	When the RTU slave responds abnormally, the Modbus gateway sends an exception message to the TCP master.
Modbus Character Interval Delay	RTU message sending time interval (supported in both master and slave modes).
Modbus Frame Interval Delay	The time interval between the current RTU response and the next RTU request (supported only in slave mode).
Modbus Slave Address Mapping (Virtual ID, Offset, Real ID)	By enabling this function, the Modbus gateway will establish a mapping relationship between virtual addresses and real addresses based on the offset. The Modbus gateway will convert the slave addresses within the received virtual address range into real slave addresses, request data from the slaves, and then convert the real slave addresses back into virtual addresses to return to the Modbus master.
Modbus Slave Pre-read	After the pre-read function is enabled, the Modbus gateway will automatically record up to 256 RTU or 128 ASCII commands, then it will automatically execute these commands and save the results in the Modbus gateway. When the Modbus master sends a command, a quick response can be obtained.
Modbus Slave Polling Time	Polling interval for the slave device.
Pre-read command aging time	If the user enables the slave pre-read function, a recorded Modbus command that exceeds the set time without continuing The command to continue reading from the slave will be deleted from the Modbus gateway.
Modbus Address Mapping	This function is only effective in Modbus RTU Slave and Modbus ASCII Slave modes, and is related to Modbus The pre-reading of the slave and the Modbus slave address mapping function are mutually exclusive.
Data encryption	
Encryption method	Methods of Data Encryption: • Crypto_DES_ECB/ CBC/ NCBC/ PCBC/ CFB/ CFB64/ OFB/ OFB64 • Crypto_3DES_ECB/ CBC/ CFB/ CFB64 /OFB64 • Crypto_AES_ECB/ CBC/ CFB1/ CFB8/ CFB128/ OFB128 • Crypto_RC4

	• Crypto_BlowFish_ECB/ CBC/ CFB64/ OFB64 • Crypto_None
Filling Method	Encryption padding methods and keys, PKCS5 padding, PKCS7 padding, Zero padding, ANSIX923 padding Filling, ISO10126, filled, unfilled.
Number of network connections	
Destination Address	IP address of the network connection peer
Destination Port	Port number of the network connection peer
Modbus_RTU/ASCII_Master mode network connection count	
Destination Address	IP address of the network connection peer.
Destination Port	Port number of the network connection peer.
Modbus ID range	Data of the Modbus slave ID within this range will be forwarded to the corresponding destination network address.
JSON configuration	
Encoding format	Supports both GB2312 and UTF-8 encoding formats. If you need to switch encoding formats, you must first clear the previous format. Clearing the format will also remove the previous JSON configuration. Make sure to determine the encoding format before configuring JSON.
JSON Method	Selection of JSON data upload or download mode.
JSON Polling Interval	The interval time for sending JSON commands to the slave station.
JSON Timeout	If no response is received within this parameter time after the gateway forwards the request to the slave device, it is considered a timeout response.
Timeout Handling	After a timeout, fill in the corresponding value to form JSON format data. Timeout clear 00/ Timeout clear FF/ No change on timeout.
Time Prefix	Add date and time information to the reported JSON data frontend.
Collection Protocol	Two collection protocols, Modbus and DLT645, are available for selection.
JSON upload command parameters	
When the collection end is Modbus:	
Device Address	The address of the slave device to be read.
Function Code	Currently, JSON upload only supports function codes 01, 02, 03, and 04.
Register Address	The register address corresponding to the slave that needs to be read.
Number of Registers	Number of registers to be read.
Byte Order	Sort the data of a single Modbus register by byte order.
Data Conversion	When the number of registers is 2, provide 4 conversion modes: Decimal (Big Endian), Decimal (Little Endian), IEEE754 (Normal Order), IEEE754 (Reverse Order).
Data Type	Supports unsigned integers, signed integers, floating-point numbers, and boolean types. Upload JSON data according to the settings.
When the acquisition end is DLT645:	
DI1 High	When the protocol version is 1997version, the range is 9~C. When the protocol version is 2007version, the range is 0~255.

DI1 Low	When the protocol version is 1997version, the range is 0~E. When the protocol version is 2007version, the range is 0~255.
DI0 High	When the protocol version is 1997version, the range is 1~6. When the protocol version is 2007version, the range is 0~255.
DI0 Low	When the protocol version is 1997version, the range is 0~E. When the protocol version is 2007version, the range is 0~255.
DLT Data Type	Select the corresponding data format based on the function type. XXXXXX.XX XX.XXXX MMDDHHmm(97) XX.XX X.XXX XXX.X XXX.XX X XXXXX X XX.XXXXYYMMDDHHMM(07)
JSON Name	The name of the corresponding data when uploaded in JSON format.
Unit	The unit corresponding to the value read from the register, which can be set to empty, meaning no unit.
Scaling	The read value is multiplied by this scaling value for data scaling processing. For example: if the value is enlarged by 100 times, the scaling value should be filled in as 100. If the value is reduced by 100 times, the scaling value should be filled in as 0.01.
Offset	Offset the read value. (The offset value is calculated after scaling).
Key Value	For the data part, it can be configured as unquoted or quoted, depending on the actual situation.
JSON Command Parameters	
Keyword	The keyword that matches the issued command.
Device Address	The address of the device that needs to be operated on.
Function Code	The function code to be issued, currently only supports 05 and 06. If multiple registers or coil values need to be written, multiple issuing commands must be configured. Multiple issuing commands.
Register Address	The address of the register that needs to be operated on.

3.3.2 Modbus Function

Next, the master is simulated by software such as Modbus Poll, and the slave is simulated by software such as Modbus Slave.

3.3.2.1 Master Mode (Modbus Master)

Taking Modbus RTU Master as an example (Modbus ASCII Master is the same)

In RTU Master mode, the RTU master device accesses the TCP slave device through the Modbus gateway.

- The RTU master device sends a request to the Modbus gateway;
- The Modbus gateway forwards the request to the TCP slave device;
- The TCP slave device returns a response;
- The Modbus gateway returns the response.

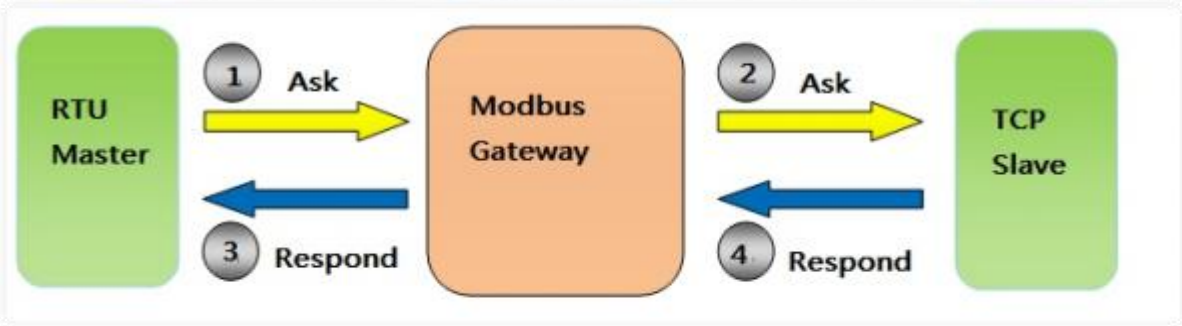


Figure 3-5 Modbus Master mode

Configure the Modbus gateway's "serial port parameters" to 9600-8-N-1, operating mode to Modbus RTU Master, and set the network address in "number of network connections" to the slave's IP and port.

RTU master configuration parameters are as follows	Modbus gateway parameter configuration
• Serial port number: COM1 • Baud rate: 9600 • Data bits: 8 • Stop bits: 1 • Parity bit: None	• Destination address: 192.168.30.100 • Destination port: 31501 • Modbus IDRange:1-1

The physical connection is described as follows:

- Serial port: Connect to the master
- Network port: Connect to the slave

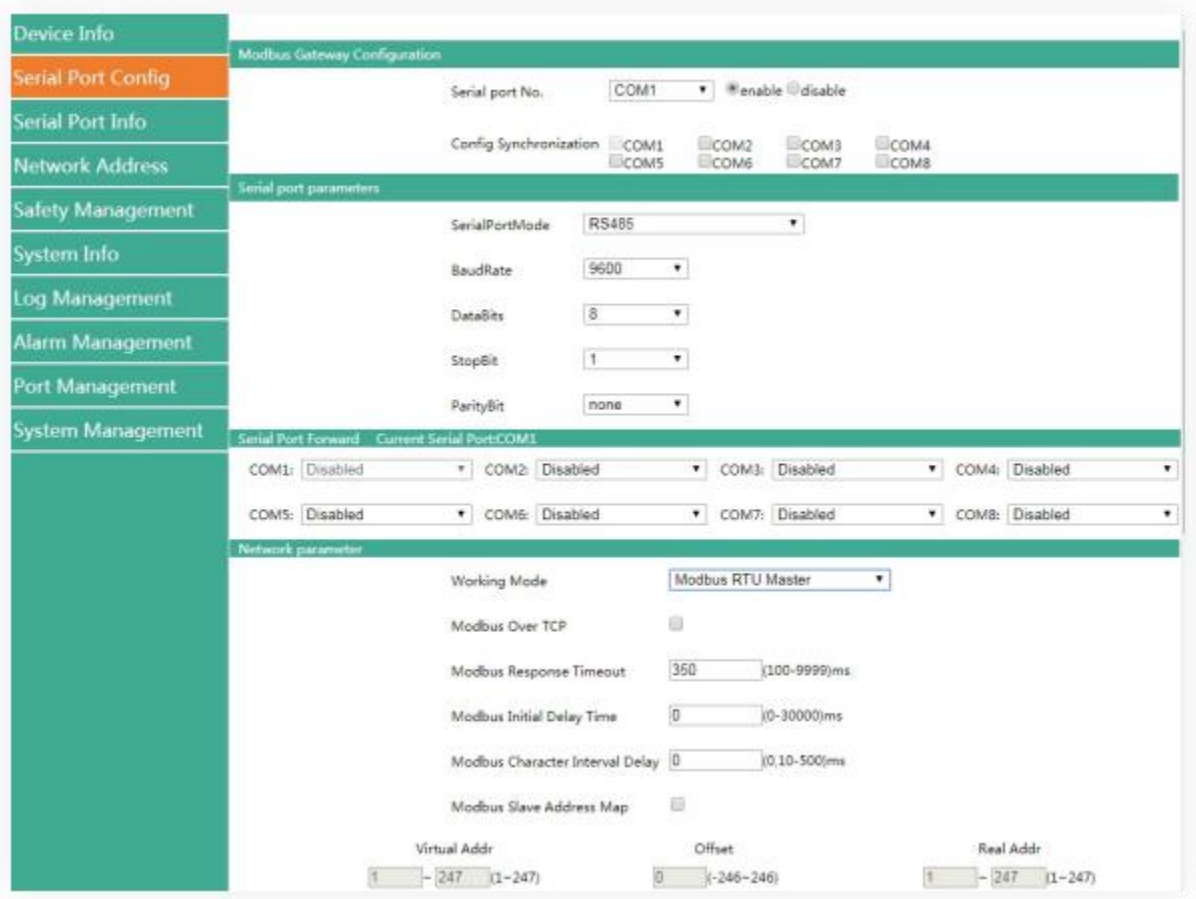


Figure 3-6 Modbus Web Parameter Configuration

Modbus Poll Software Configuration:

- Open the Modbus Poll software, go to “Connection”→“Connect”, the connection parameter configuration and reading parameter configuration are as follows:

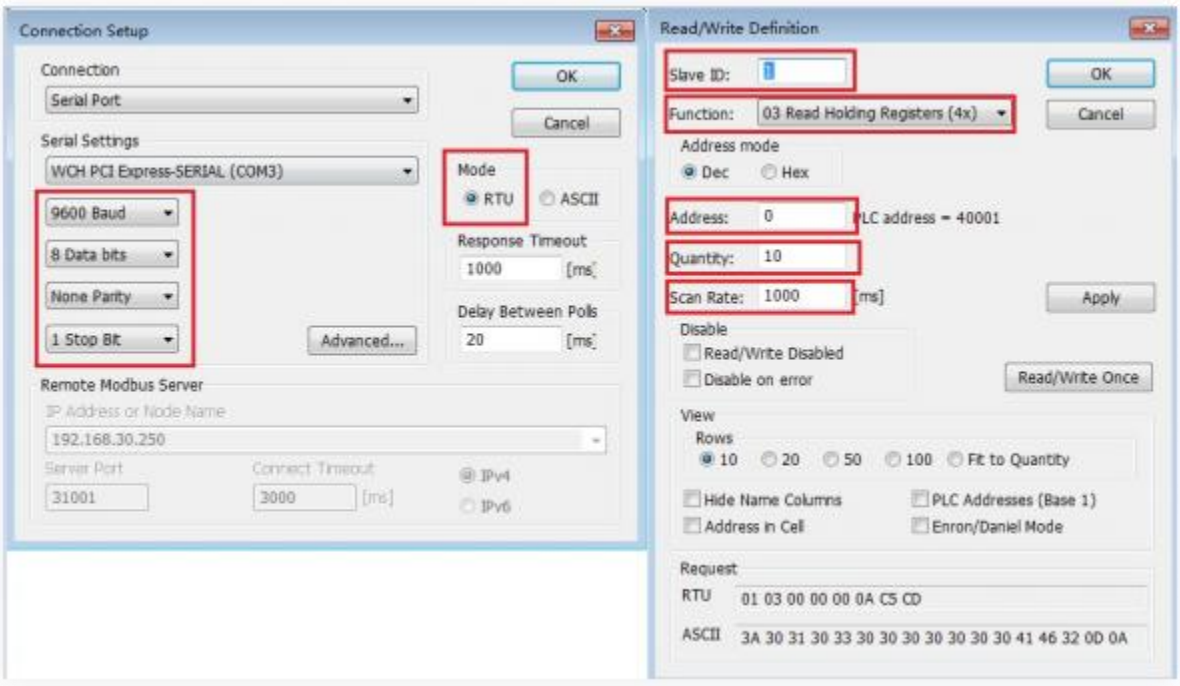


Figure 3-7 Modbus host serial port parameter configuration and read parameter configuration

- Reading parameter configuration: Slave ID is 1, function code is 03, the starting address of the registers to be read is 0, the number of registers to be read is 10, and the interval for cyclic reading is 1000ms.

Modbus Slave Software Configuration:

- Open the Modbus Slave software, go to "Connection" → "Connect", and configure the connection parameters and slave device properties as follows:

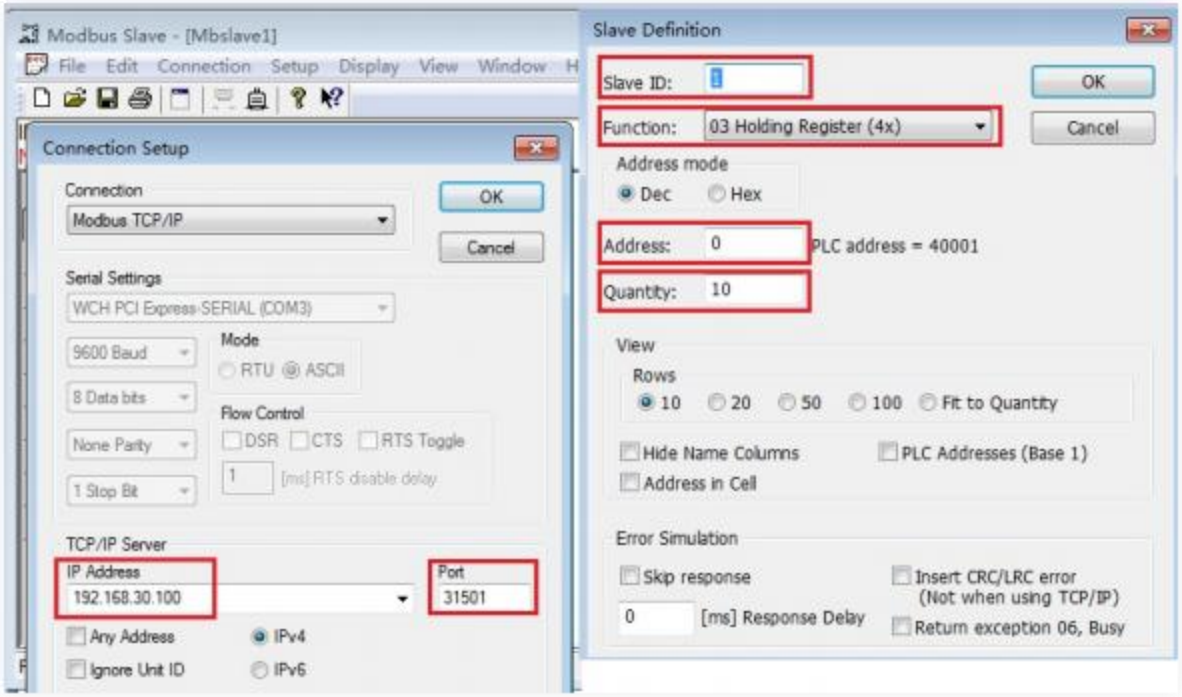


Figure 3-8 Modbus slave connection parameter configuration and slave device property configuration

- Slave Device Definition Configuration: Slave ID is 1, Function Code is 03, Register Starting Address is 0, Total Number of Registers is 10.

View test results:

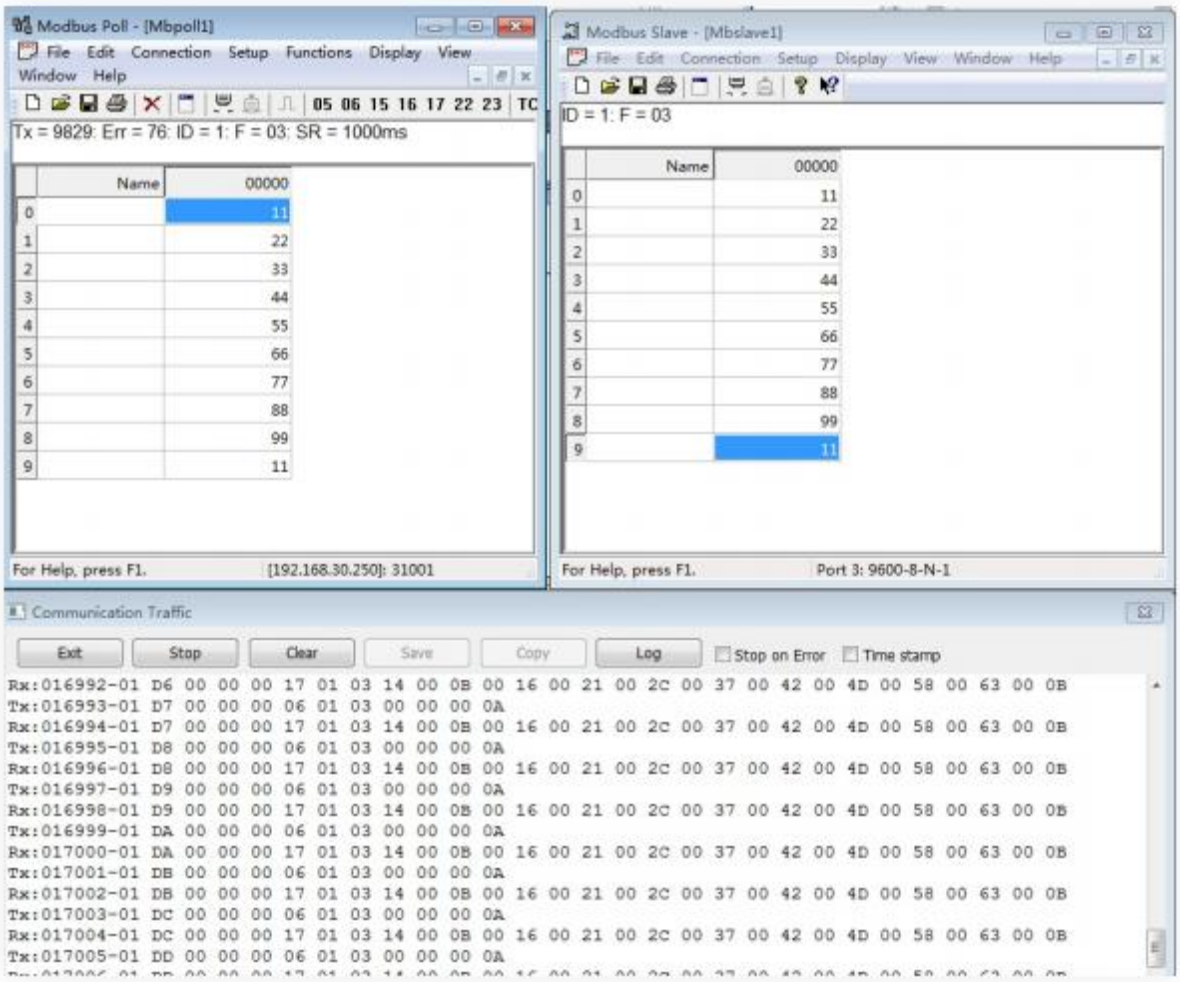


Figure 3-9 Modbus Slave Register Values Responding Normally to the Master

Communication is normal, and the master can read the register data from the slave through the Modbus gateway device.

3.3.2.2 Slave mode (Modbus Slave)

Taking Modbus_RTU_Slave as an example (Modbus ASCII Slave is the same)

In Modbus RTU Slave mode, TCP master devices access RTU slave devices through the Modbus gateway.

- TCP master device sends a request to the Modbus gateway;
- Modbus gateway forwards the request to the RTU slave device;
- RTUSlave device returns response;
- ModbusGateway sends back response.

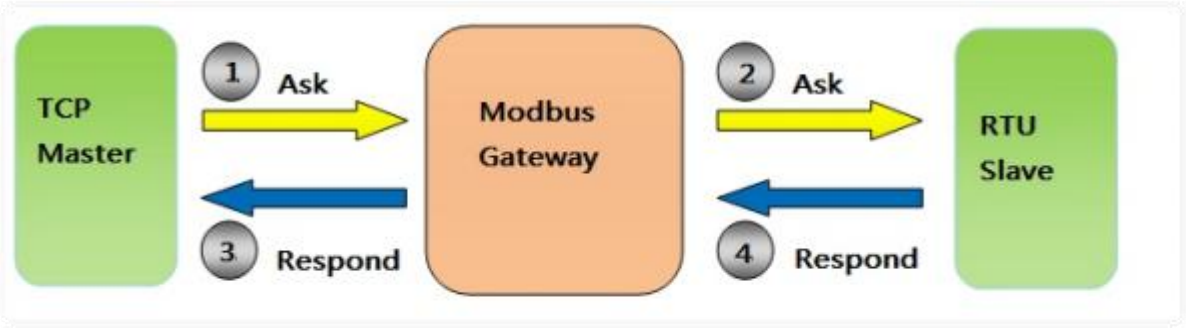


Figure 3-10 Modbus Slave mode

Configure Modbus Gateway's "serial port parameters" to 9600-8-N-1, "network parameters" working mode to Modbus RTU Slave, local port to 31001.

RTU master configuration parameters are as follows	Modbus gateway parameter configuration
• Serial port number: COM1 • Baud rate: 9600 • Data bits: 8 • Stop bits: 1 • Parity bit: None	• IPaddress:192.168.30.250 • Port number:31001

The physical connection is described as follows:

- Serial port: Connect to the master
- Network port: Connect to the slave

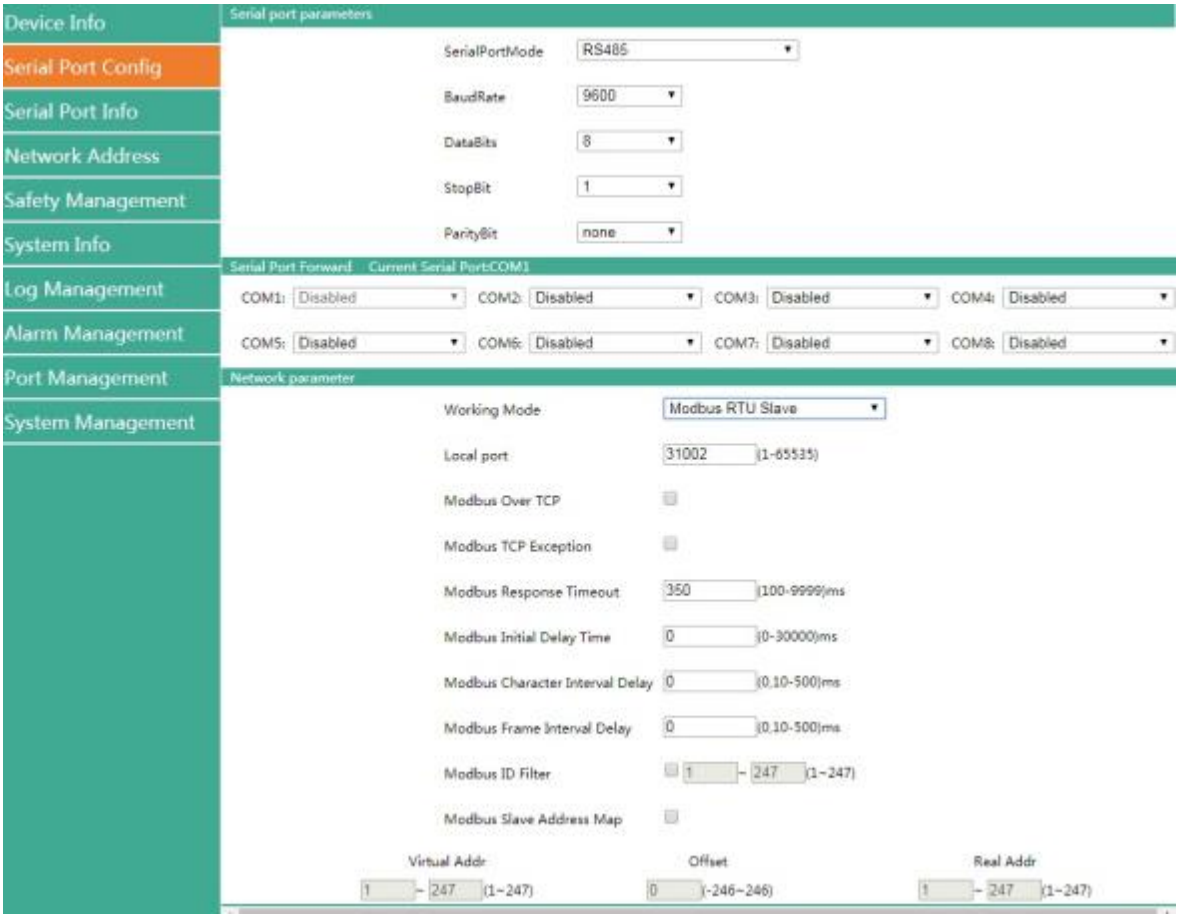


Figure 3-11 Modbus WEB parameter configuration

- Open the Modbus Poll software, go to “Connection”→“Connect”, the connection parameter configuration and reading parameter configuration are as follows:

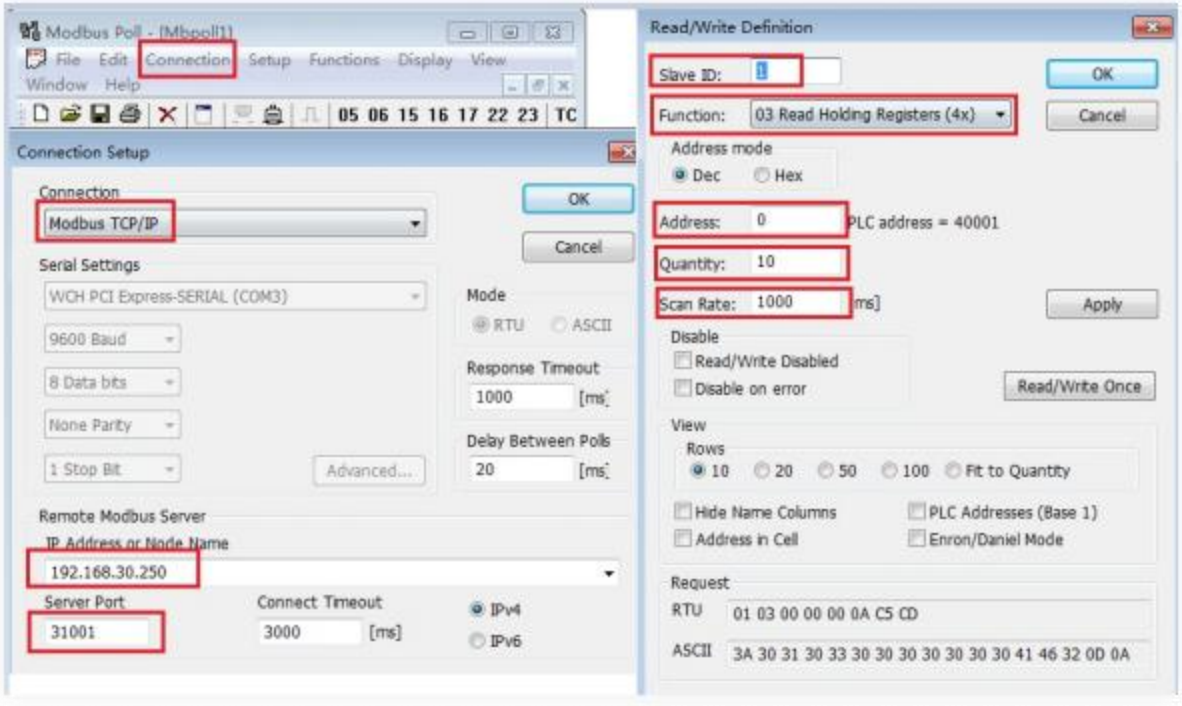
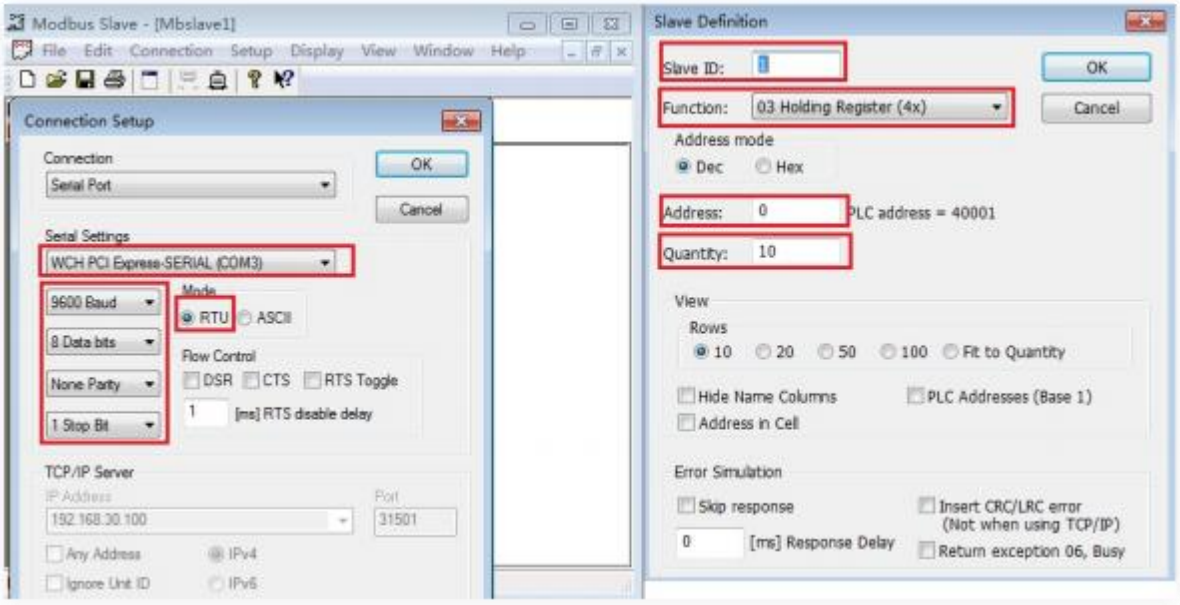


Figure 3-12 Modbus Host network connection parameter configuration and device property definition

- Reading parameter configuration: the slave ID is 1, the function code is 03, the starting address of the register to be read is 0, the number of registers to be read is 10, and the cycle reading interval is 1000ms.

Modbus Slave Software Configuration:

- Open the Modbus Slave software, go to "Connection" → "Connect...", and configure the connection parameters as follows:



3-13 Modbus slave serial port parameter configuration and slave property definition

- Slave Device Definition Configuration: Slave ID is 1, Function Code is 03, Register Starting Address is 0, Total Number of Registers is 10.

Test Results:

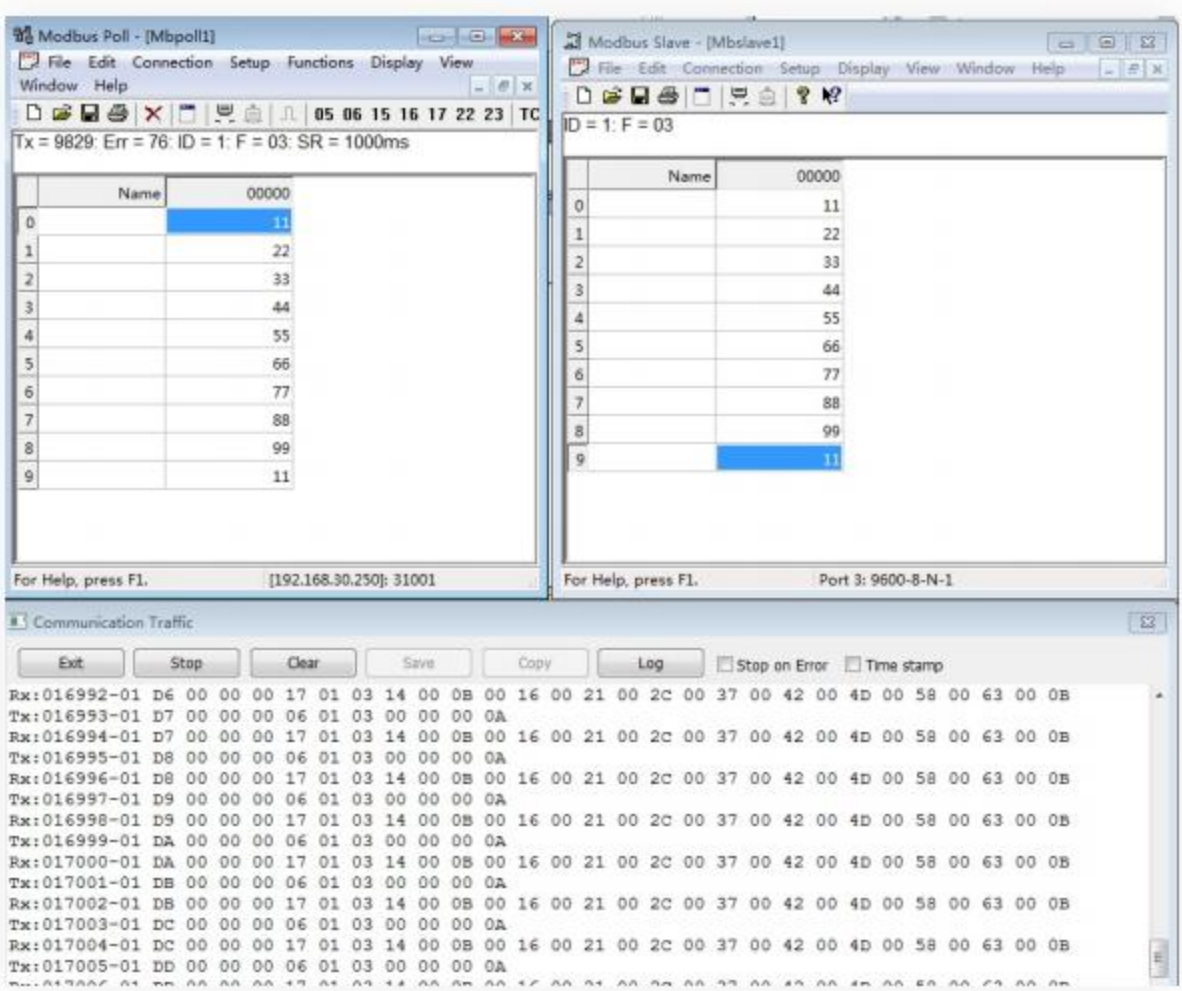


Figure 3-14 Modbus Slave Register Values Responding Normally to the Master

3.3.2.3 Slave Address Mapping

The Modbus protocol stipulates that all slave devices must have a unique ID number (1~247). This ID number is used to identify the slave address in response to requests from the master device. The Modbus device ID number is set by the manufacturer.

- **Slave ID Mapping:** Each slave device has 2 ID addresses, namely a virtual ID address and a real ID address. The real ID exists in the slave device, and other devices access the slave device directly through the real ID. The virtual ID exists in the gateway and is the unique access address for the slave device in the gateway, allowing other devices to indirectly access the slave device through the virtual ID.

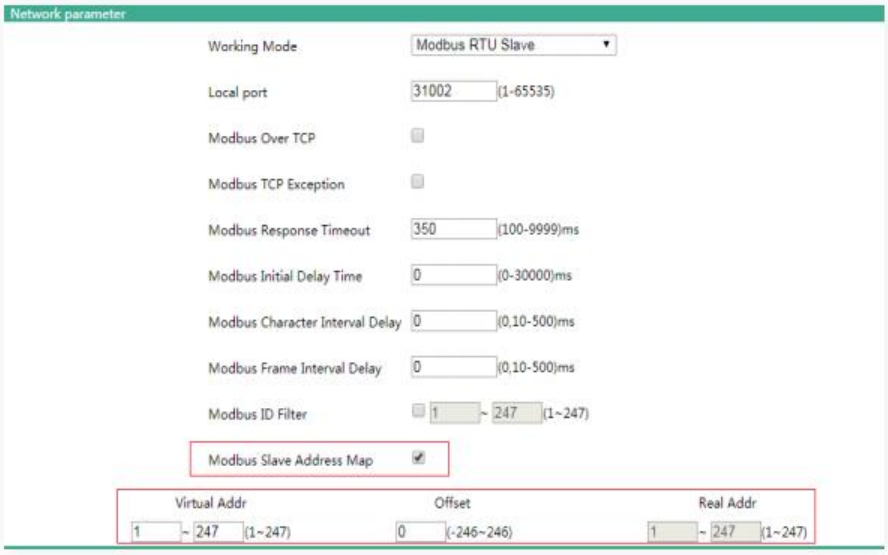


Figure 3-15 Slave Address Mapping Function Settings

3.3.2.4 Modbus Slave Prior Read

After the slave pre-read function is enabled, the gateway will automatically record up to 256 RTU or 128 ASCII commands, then automatically execute these commands and save the results in the Modbus gateway. When the Modbus master sends a command, a quick response can be obtained.

Figure 3-16 Modbus slave read-ahead function setting

- **Modbus slave polling interval:** If the user enables the slave pre-read feature, setting this polling interval allows them to control the time interval at which the Modbus gateway polls the slave.
- **Pre-read command expiration time:** If the user enables the slave pre-read feature, any recorded Modbus command that has not been read from the slave within the set time will be deleted from the Modbus gateway.

3.3.2.5 Modbus Feature Function

This device also has the following feature functions:

- **Modbus response timeout:** The timeout for receiving Modbus serial port responses. If the gateway does not receive a response within this parameter time after forwarding the request to the slave device, it is considered a timeout response.
- **Modbus initial delay time:** After powering on the device, there is a delay before starting

Modbus message communication.

- **ModbusTCP exception:** When the RTU slave responds abnormally, the Modbus gateway sends an exception message to the TCP master.
- **Modbus character interval delay:** The time interval for sending RTU messages (supported in both master and slave modes).
- **Modbus frame interval delay:** The time interval between the current RTU response and the next RTU request (supported only in slave mode).

Configuration in slave mode is as follows:

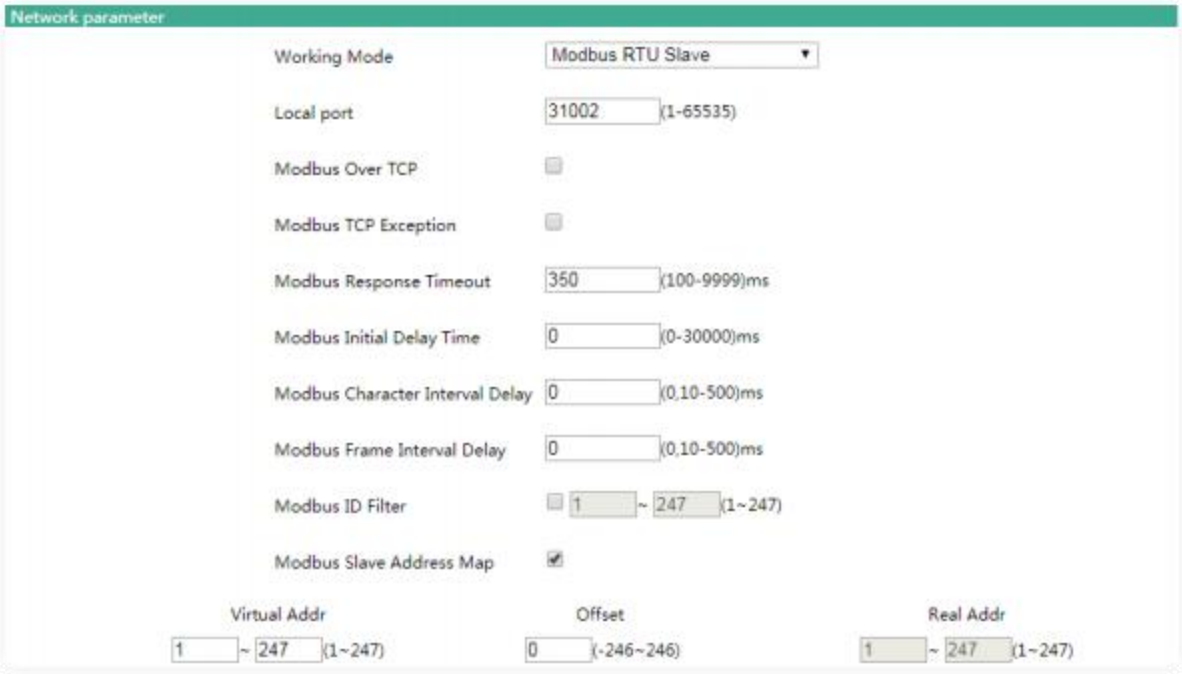
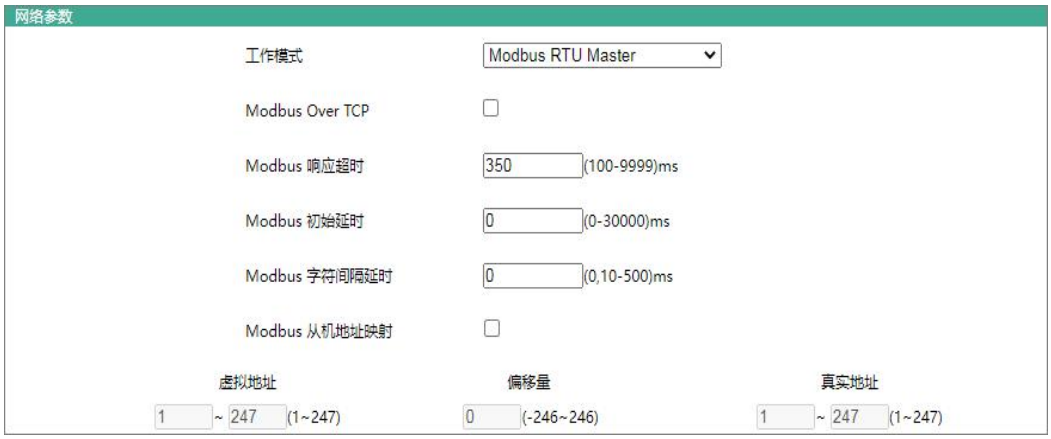


Figure 3-17 Modbus features in station mode

The configuration in master station mode is as follows:



3.3.3 RealCOM Function Use

In RealCOM mode, the Modbus gateway works in conjunction with the operating system that has the RealCOM driver software installed. The RealCOM driver software maps the serial port of the Modbus gateway to the local COM interface of the host, allowing the

existing serial device software or communication modules on the host to be used directly without modification.

The RealCOM driver software transparently transmits the data received by the virtual COM interface on the host to the serial port of the Modbus gateway in TCP/IP format. The Modbus gateway transparently transmits the data received from the serial port to the virtual COM interface of the host in TCP/IP format.

The Modbus gateway supports three types of RealCOM protocols:

- RealCOM_MCP mode is compatible with Moxa's driver management software;
- RealCOM_CCP mode is compatible with Kanghai's serial port management tools;
- RealCOM_MW mode supports Come-Star's virtual serial port tool software.

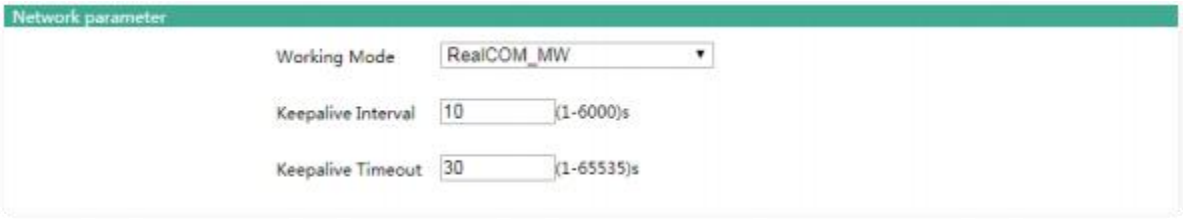


Figure 3-19 RealCOM function configuration

How to use Come-Star RealCOM:

1. Configure the Modbus gateway in WEB as RealCOM_MW mode;
2. Install and open the Come-Star virtual serial port management software;
3. Click on [Add Device] to pop up the add serial port mapping interface;
4. Click on [Scan], and the software will scan for Modbus gateway devices in the local area network;
5. Select the corresponding Modbus gateway device based on the MAC address and IP address;
6. Click on [Serial Port Mapping] and wait for the local virtual serial port to be created;

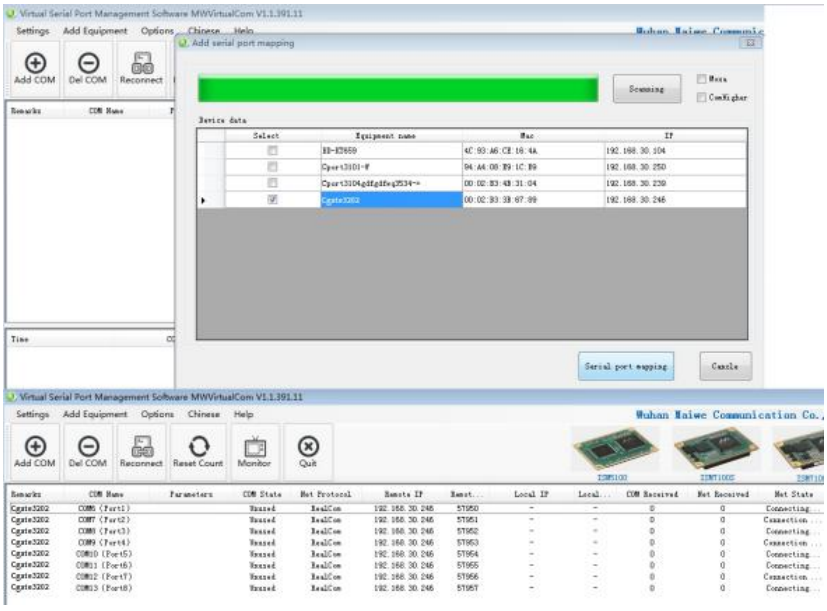


Figure 3-20 Virtual Serial Port Management Software Creating Virtual Serial Port

Connect the Modbus gateway's serial port to the real serial port on the host, and use a serial port debugging tool to open both the virtual serial port created by the management software and the real serial port on the host, allowing data to be sent back and forth for testing.

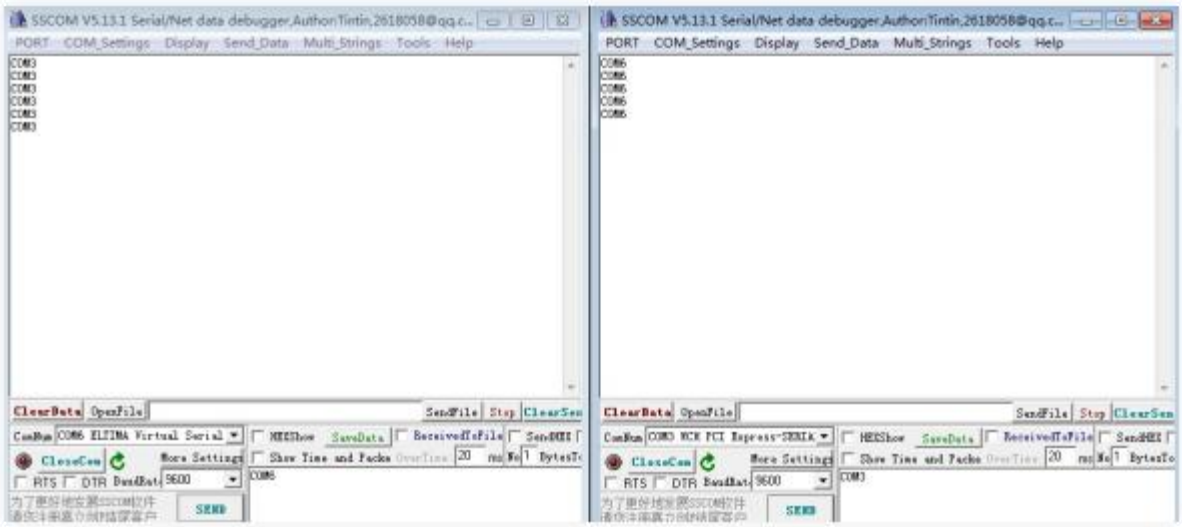


Figure 3-21 Serial Port Communication Test

3.3.4 Httpd Client Functions

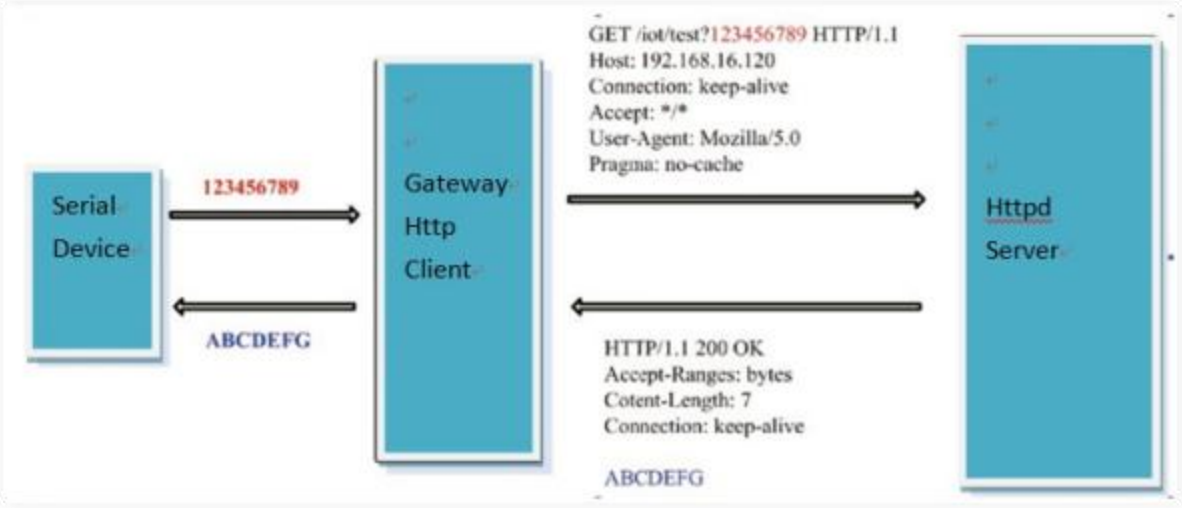


Figure 3-22 HTTPD Client Mode Introduction

This function allows the Modbus gateway to submit data received from the serial port to the HTTP server in HTTP format. If the HTTP server has data to send, the Modbus gateway will pass the HTTP body data to the serial port.

Specific usage method:

1. Select "HTTPD Client" working mode;
2. Fill in the HTTPD address, which is the HTTP server address. It can be an IP address or a domain name (must have the ability to connect to the external network);
3. Fill in the HTTPD port number;
4. The HTTPD method requires the correct URL path to be filled in, and select either the GET

or POST method as needed;

5. The user fills in the HTTPD request header as needed;
6. Finally, click the configuration button to save the parameters.

Figure 3-23 Introduction to HTTPD Client Configuration

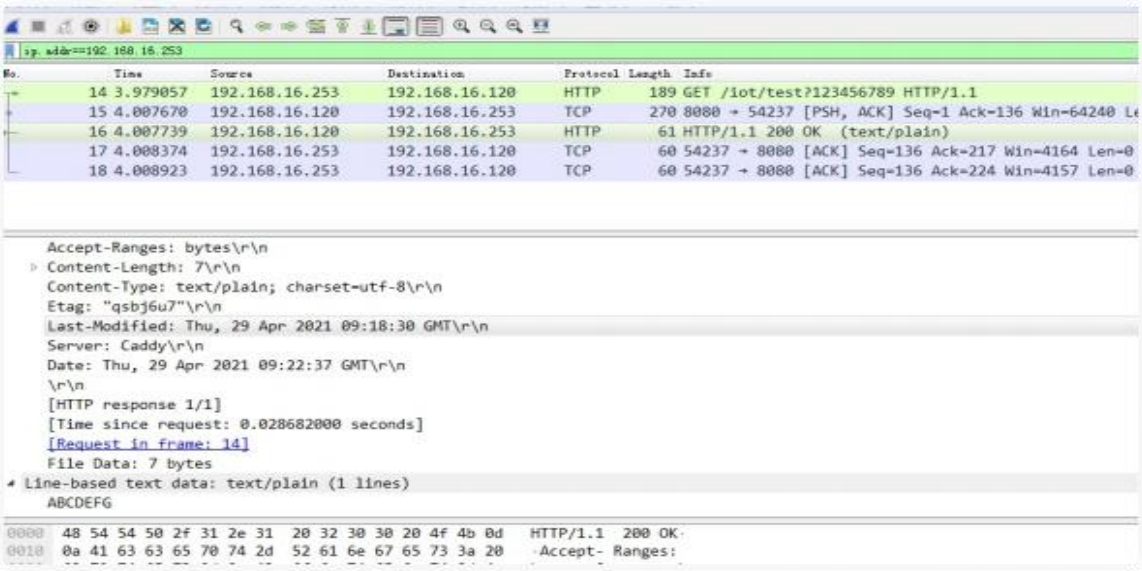


Figure 3-24 HTTPD Client Communication Wireshark Packet Capture Example

3.3.5 WebSocket Client Function

This function allows the Modbus gateway to act as a WebSocket Client, transmitting data received from the serial port in hexadecimal format to the WebSocket server, which can also send data to the serial device at any time.

Figure 3-25 Introduction to WebSocket Client Configuration

Specific usage method:

1. Select the working mode "WebSocket Client";
2. Fill in the WebSocket server address, which can be an IP address or a domain name (needs to have the ability to connect to the external network);
3. Fill in the WebSocket server port number;
4. The WebSocket method requires the correct URL path to be filled in;
5. Users can choose the WebSocket Ping interval as needed, filling in 0 indicates that the Ping function is not used;
6. Finally, click the configuration button to save the parameters.

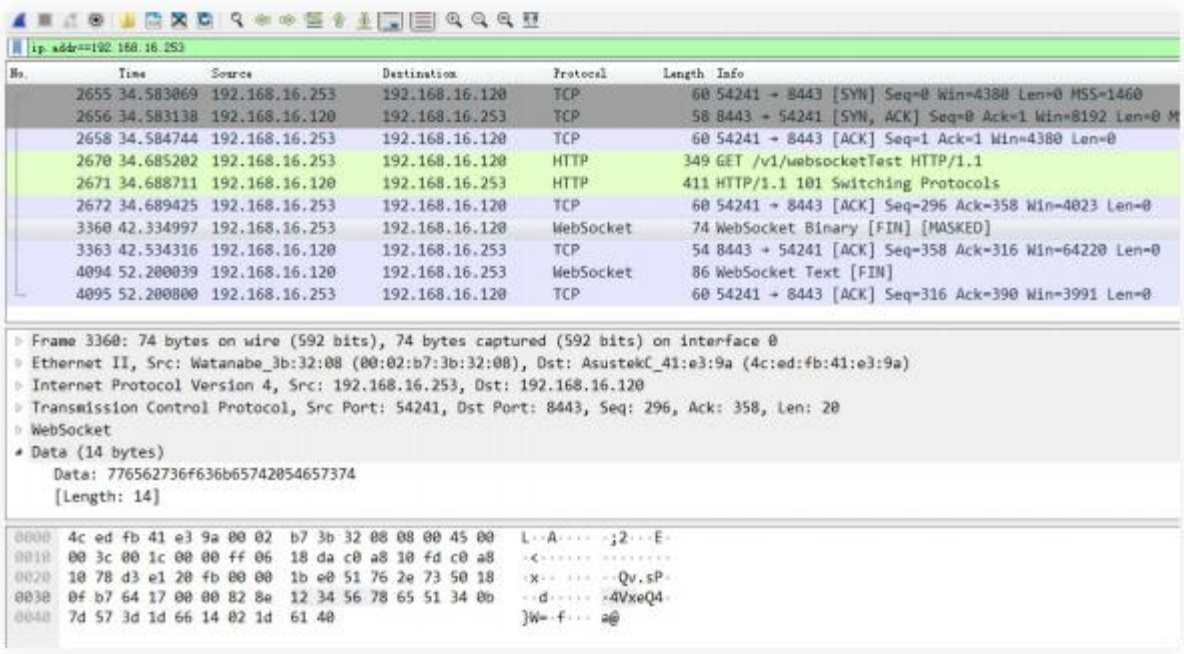


Figure 3-26 WebSocket Client communication Wireshark packet capture example

3.3.6 MQTT Function

This feature allows the Modbus gateway to act as an MQTT client, communicating with the cloud platform server, and supports connections to two independent MQTT

cloud platforms. Data received via the serial port is sent to the MQTT server, and the MQTT server can also forward messages back to the

serial port. (This requires that the device be connected to the internet.)

Let’s use Come-Star Cloud as an example for the MQTT server:

• Platform selection: Come-Star Cloud.

• MQTT address: 120.202.25.215.

• MQTT port: 1883.

Other specific settings are shown in the figure below:

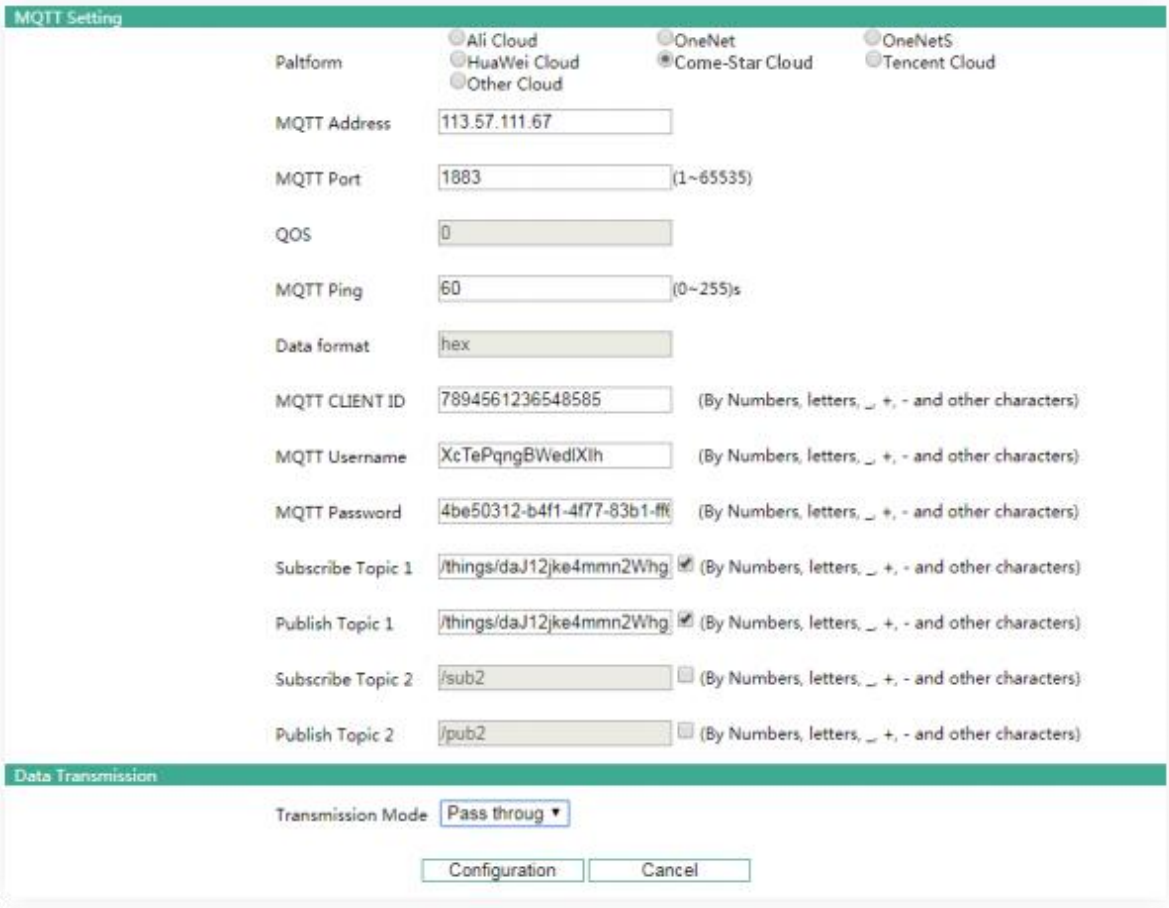


Figure 3-27 MQTT configuration introduction

Then log in to the Come-Star cloud:

- Find our gateway device in device management, click in to connect.

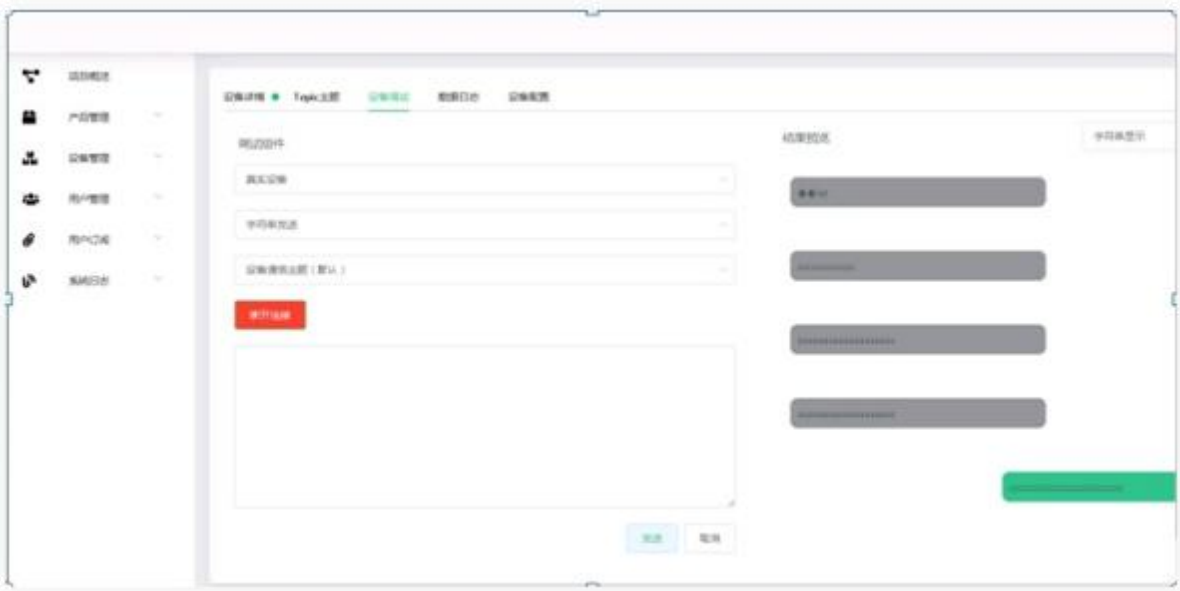


Figure 3-28 Example of sending and receiving information on the MQTT Come-Star cloud platform

- Once connected, use a serial tool to open the corresponding serial port, then send strings. Data can be normally sent and received between the Come-Star cloud server and the serial port, confirming that the MQTT function is working properly.

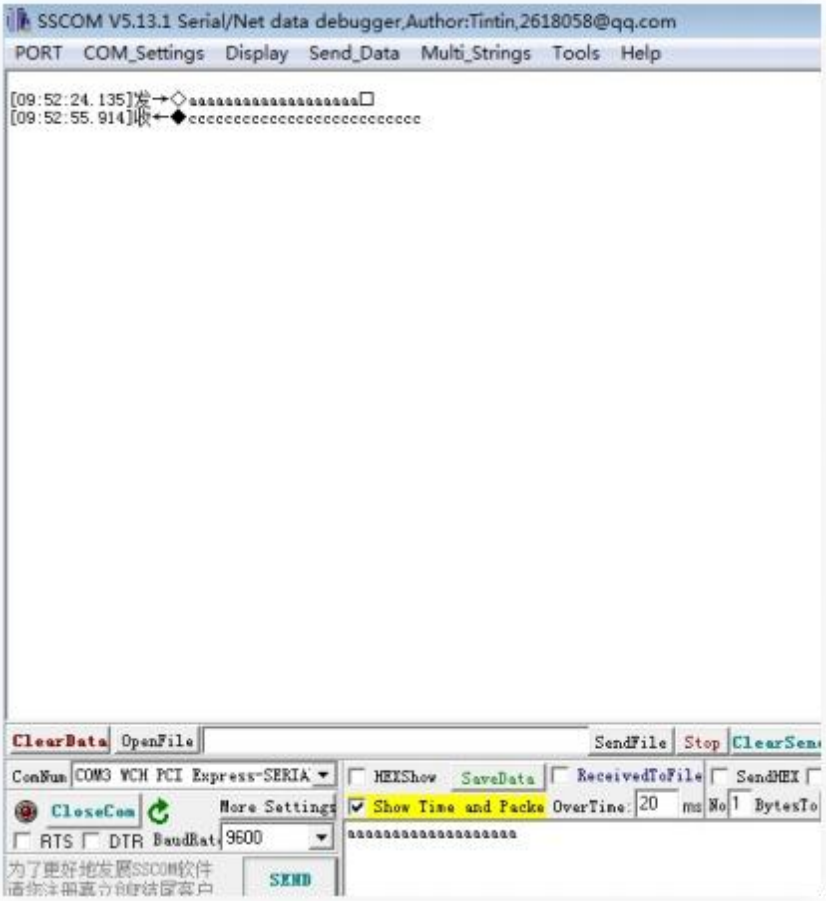


Figure3-29 MQTT Example of Message Exchange between Serial Port and Come-Star Cloud

3.3.7 JSON Function

First, the JSON function is only available in three working modes: WebSocket Client, HTTPD Client, and MQTT Client.

The JSON function is a data reporting feature under the Modbus gateway mode. The serial port can choose the Modbus or DLT645 protocol. This product acts as the master station, collecting data information from slave devices, and then forming JSON format data to be sent to the server.

3.3.7.1 JSON Up Send

Taking Modbus as the collection end and WebSocket

Client as the network end, for example: first configure

the serial port WebSocket Client parameters.

- WebSocket Address: 192.168.30.100 (i.e., the PC address of the WebSocket server).
- WebSocket Port : 8443.
- WebSocket Function: /v1.

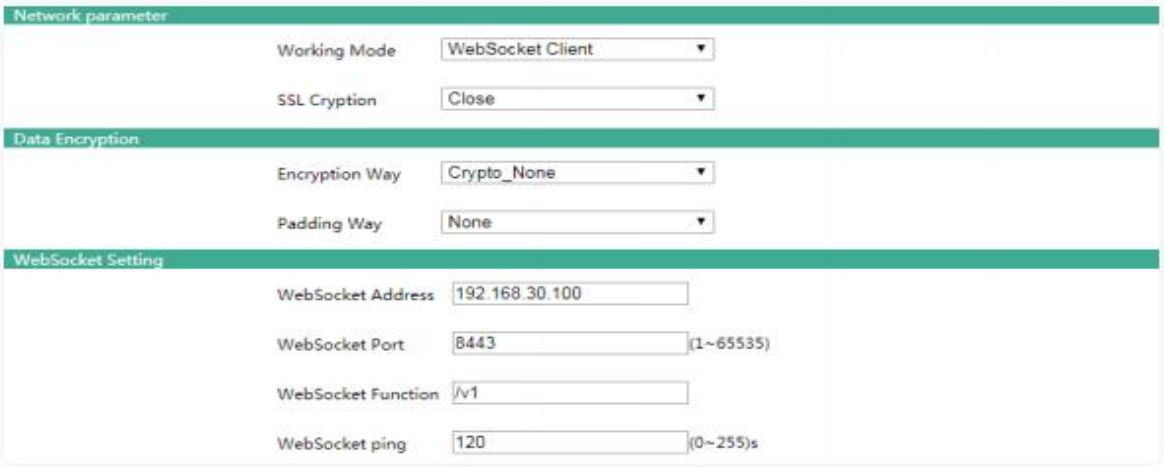


Figure 3-30 WebSocket Parameter Configuration Example

- Select Transmission Mode JSON: Then configure a JSON sending command.
- JSON Way: JSON Sending.
- JSON Poll Time: 1000ms (the polling interval time for the JSON command).
- JSON Timeout: 350ms.
- Timeout Dispose: No change on timeout.

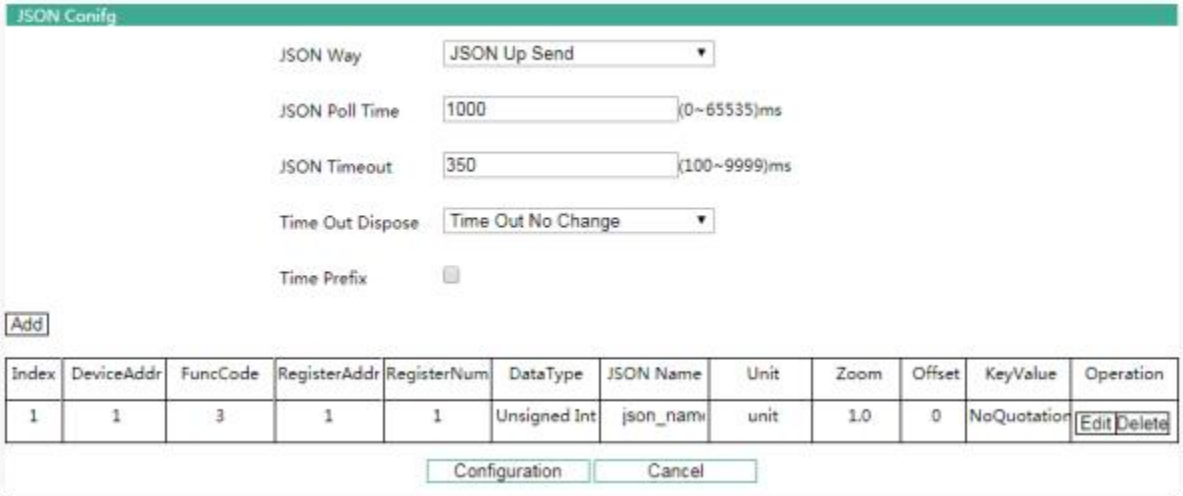


Figure 3-31 Example of JSON up send mode parameter configuration

Open the Modbus Slave software to simulate a slave device.

Verify that the serial port settings are configured correctly.

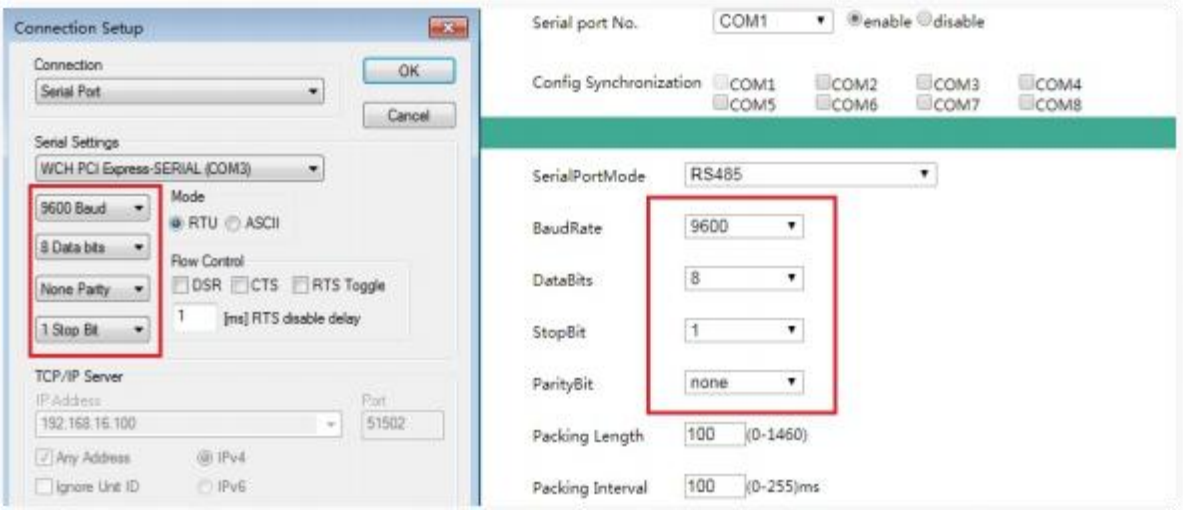


Figure 3-32 Serial Port Parameter Configuration and Modbus Slave Software Configuration Example

Open a WebSocket server; here we are using WebSocketMan.exe software. Configure the server parameters: mainly the listening IP address, set to the IP address of the Modbus gateway device. Click to start listening. At this point, a connection from the device side will be received, and communication will begin.

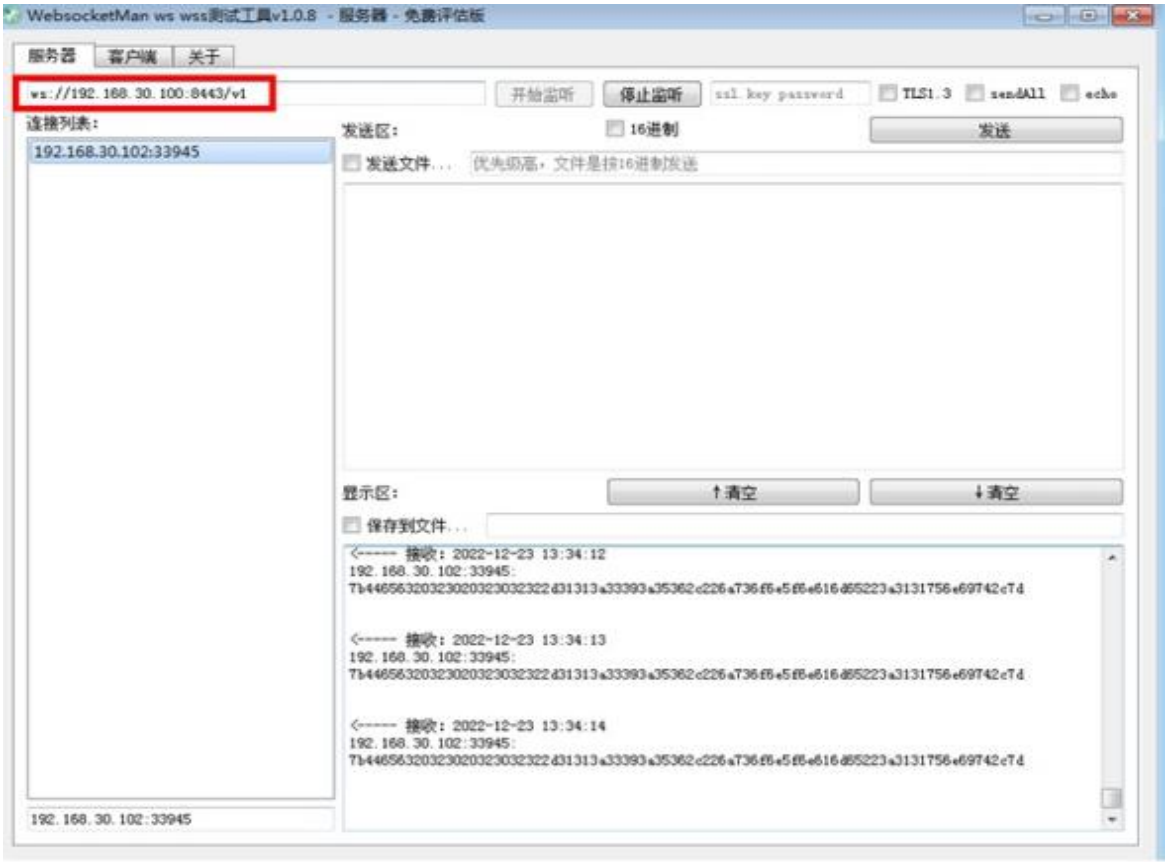


Figure 3-33 WebSocket Server Configuration Example

Since the JSON command we configured is to query the device with ID 1, check the value at address 1 in the Modbus Slave software, which is 11.

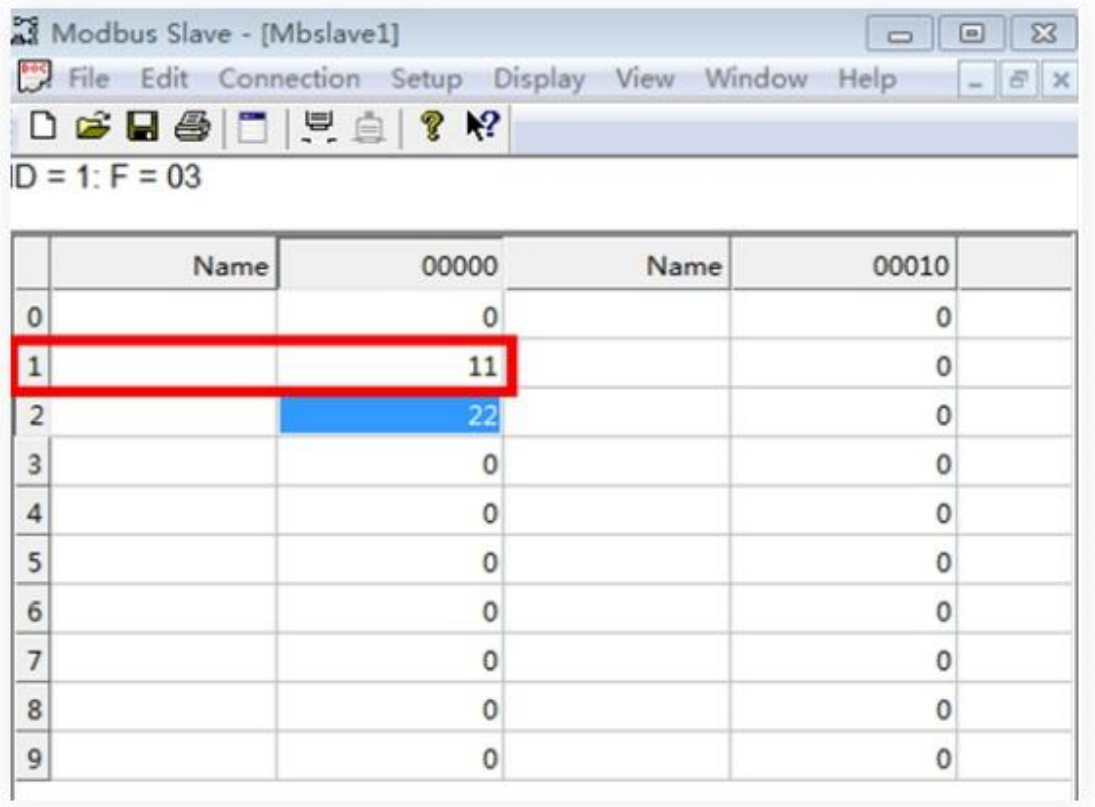


Figure 3-34 Modbus Slave Address Value Example

Then open Wireshark software to capture packets and check the data sent from the device to the WebSocket server: {Dec 20 2022-11:39:56,"json_name":11unit}, completing the upload of the value at the corresponding address of the slave device in JSON format to the server.

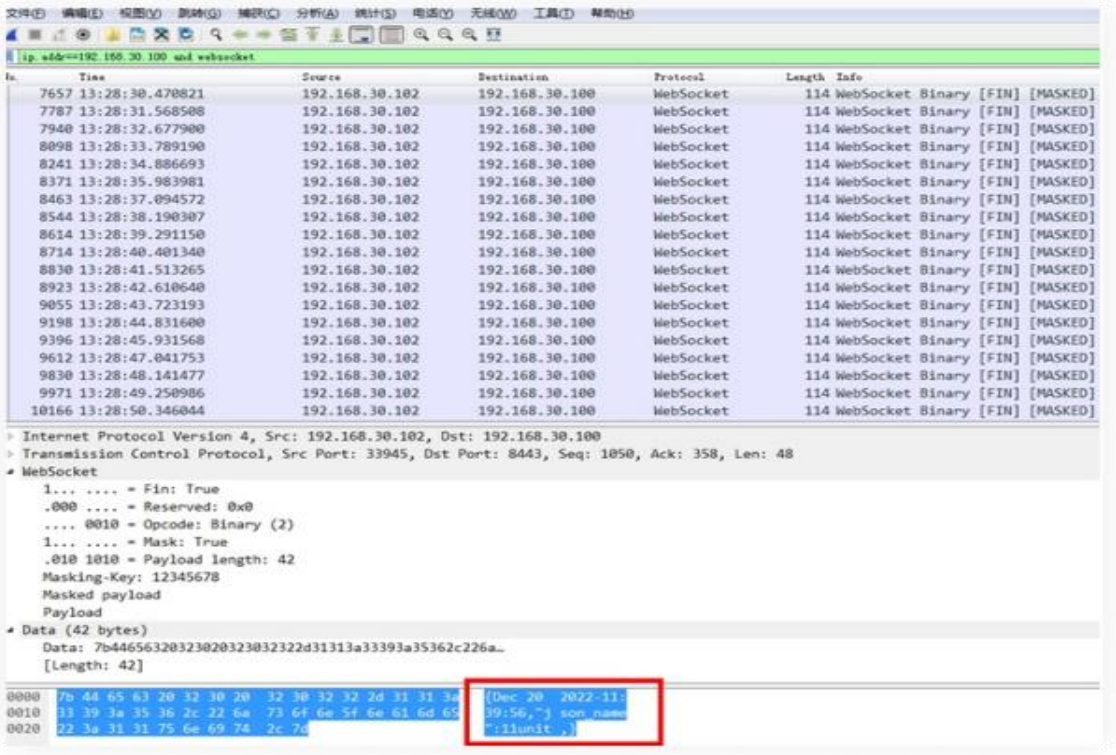


Figure 3-35 Wireshark Packet Capture Example

3.3.7.2 JSON Down Send

JSON Down Send Function:

- JSON Way: JSON Down Send
- Then configure a delivery instruction: device address is 1, function code is 6, and register address is 5.

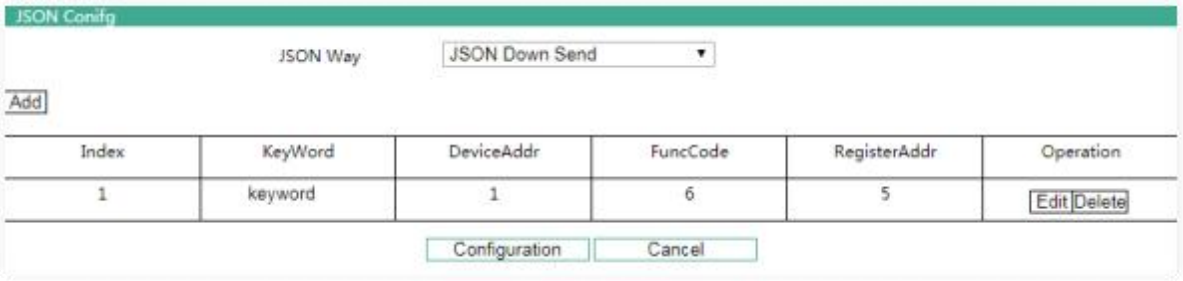


Figure 3-36 Example of JSON Down Send Parameter Configuration

- Then, on the WebSocket server side, modify the corresponding register value of the slave device by using the delivery keyword plus value method. The delivery format is keyword+colon+value+semicolon, as follows:
  keyword: value;

For example: to change the value of the register address 5 of the slave device with device address 1 to 1123, the command to be entered on the WebSocket server side should be keyword:1123;

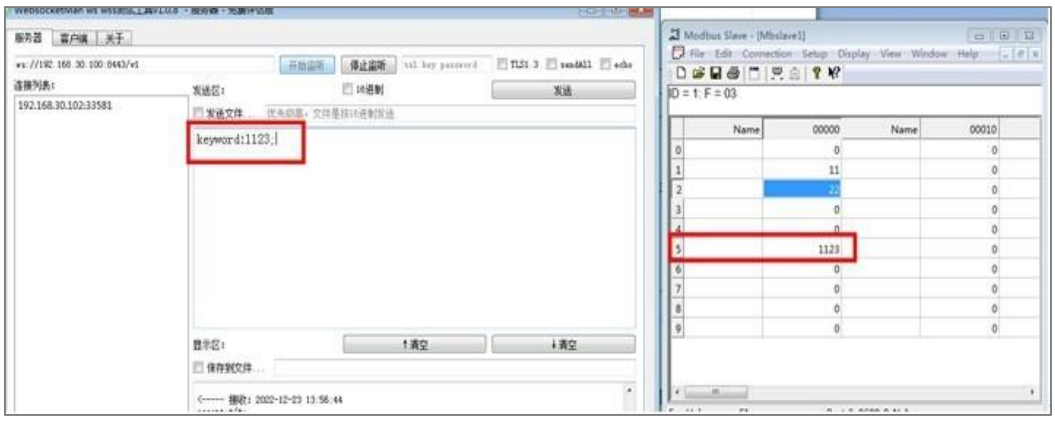


Figure 3-37 Example of JSON Down Send Parameter Function Implementation

Note:

- The port numbers 22, 80, 443, 4500, 4800, 57050, 57051, 57052, 57850, and 57851 are already in use by the system. Please do not reuse these port numbers when configuring.
- The device and the remote device must have the same baud rate, parity bit, data bits, and stop bits.
- If this device operates in UDP Client mode, the remote device must operate in UDP Server mode; if this device operates in UDP Server mode, the remote device must operate in UDP Client mode; if this device operates in TCP Client mode, the remote device must operate in TCP Server mode; if this device operates in TCP Server mode, the remote device must operate in TCP Client mode.

- If the operating mode of this device is UDP multicast mode, the same multicast address can only be used once and is not allowed to be used in different serial port configurations.
- When long frame data needs to be frequently used or when there are high requirements for data transmission, please adjust the baud rate and extend the sending interval appropriately to prevent garbled characters or packet loss caused by slow serial port speed.
- When configuring the device, the user should ensure that the external serial devices stop sending data to the Modbus gateway to avoid garbled characters.
- Due to the influence of the serial port rate, when Modbus data experiences a timeout, it is important to configure an appropriate Modbus receive timeout on the web. Additionally, the read timeout on the host computer should also be appropriately extended based on the baud rate.
- In the JSON Down Send feature, when the command code is 6, the valid input range is -32768 to 65535; values outside this range are considered invalid. When the command code is 5, it indicates writing the coil status; the sent value must be 0 or 1, and any non-zero value will be treated as 1.

3.4 Serial port information

The serial port information module includes: serial port information and network connection information.

The serial port information page is used to display the current connection information of the serial port, as shown in Figure 3-38:

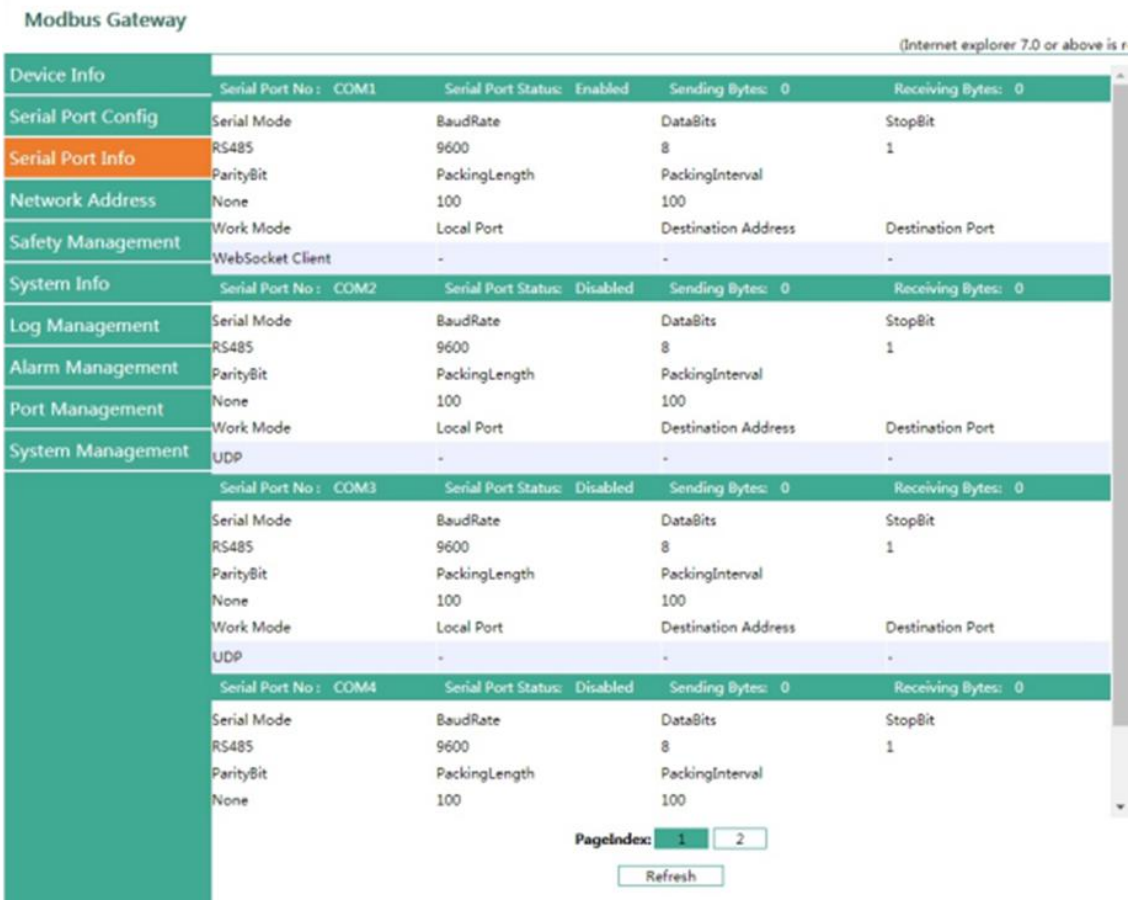


Figure 3-38 Serial port information page

3.5 Network Address

The network address module includes: network protocol, IP address, subnet mask, default gateway, and DNS address.

3.5.1 Network Address

The purpose of this function is to assign a specified IP address to the Modbus gateway. The default IP address of the Modbus gateway when it leaves the factory is 192.168.16.253.

The network address configuration page is shown in Figure 3-39.

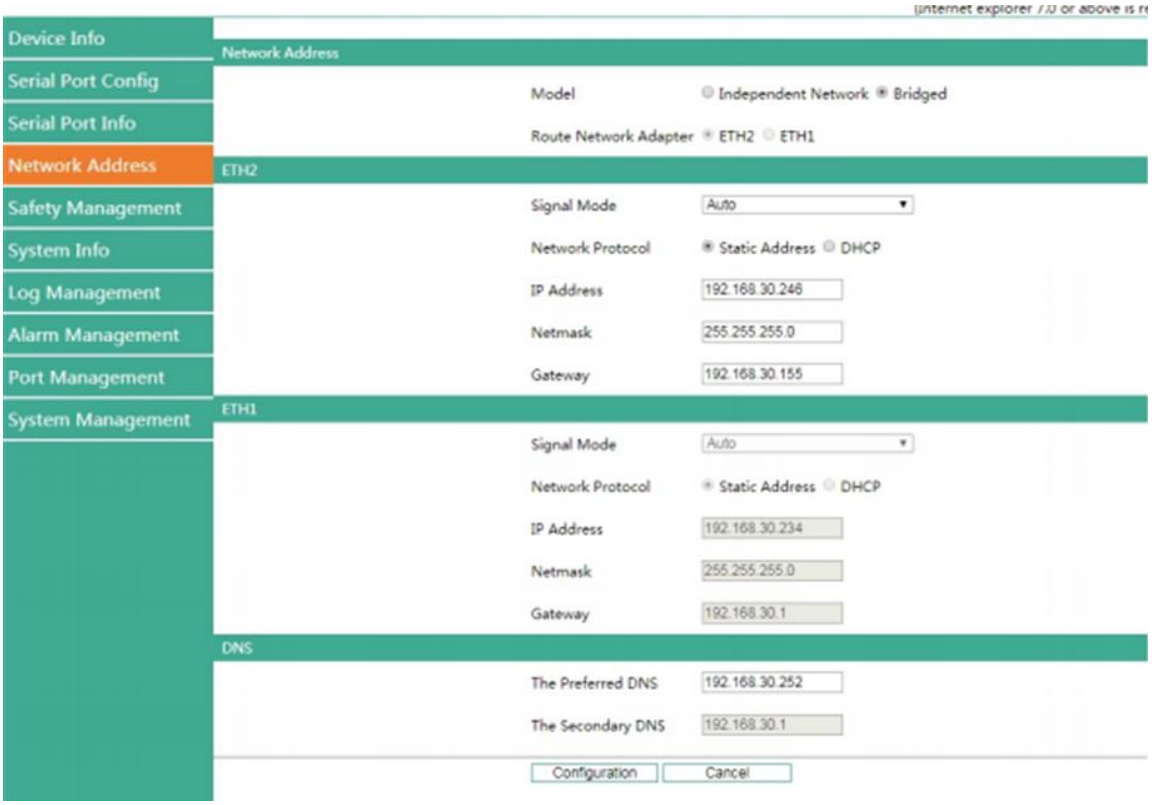


Figure 3-39 Network address interface

- **Model:** Select independent network or bridged mode.
  - Independent network: Two network cards are in mutually independent subnets;
  - Bridged mode: Two devices can be connected in series via an Ethernet cable. In this case, only one external network cable needs to be connected to the first device, and the subsequent devices can communicate (in cascade mode, the parameters configured for the ETH2 network card are used).
- **Routing network adapter:** This option is only effective in a standalone subnet, routing according to the specified network card configuration.
- **Signal mode:** This option is used to specify the network card speed.
- **Network protocol:** Choose static address or DHCP.
- **IP address:** An IP address is a 32-bit length address assigned to devices connected to the Internet. The IP address consists of two fields: the network number field (net-id) and the host number field (host-id).
- **Netmask:** The mask is a 32-bit number corresponding to an IP address, with some bits being 1 and others being 0. The mask can divide the IP address into two parts: the

subnet address and the host address. The part of the IP address corresponding to the bits that are 1 in the mask is the subnet address, while the other bits are the host address.

- **Gateway:** The default gateway in a host is usually referred to as the default route. The default route is the route chosen by the device when no other route exists for the destination address in the IP packet.
- **DNS:** The IP address of the DNS server. When a device uses a static IP address, the user needs to fill in this information. If a specific DNS server is not used, the default gateway IP address can generally be filled in.

Whenever the user modifies the address settings, they must click the [Configure] button to submit to the Modbus gateway and switch to a waiting page as shown in Figure 3-40.

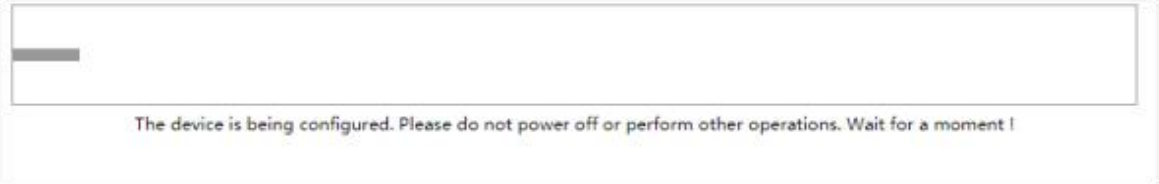


Figure 3-40 waiting interface after the user modifies the address

When the progress bar on the screen is complete, the Modbus gateway will restart the Web server, and the user will need to log in again.

Note:

- When entering the waiting page after configuring the IP, please do not turn off the power or perform other operations to avoid failure in modifying the IP address.
- The configured IP address and default gateway need to be in the same subnet.
- If the device obtains the IP address using DHCP, after the device restarts, the user needs to use the company's network management assistant to search for the device in order to accurately know the device's new IP address.

3.6 Security Management

The safety management module is mainly used to configure parameters related to device security.

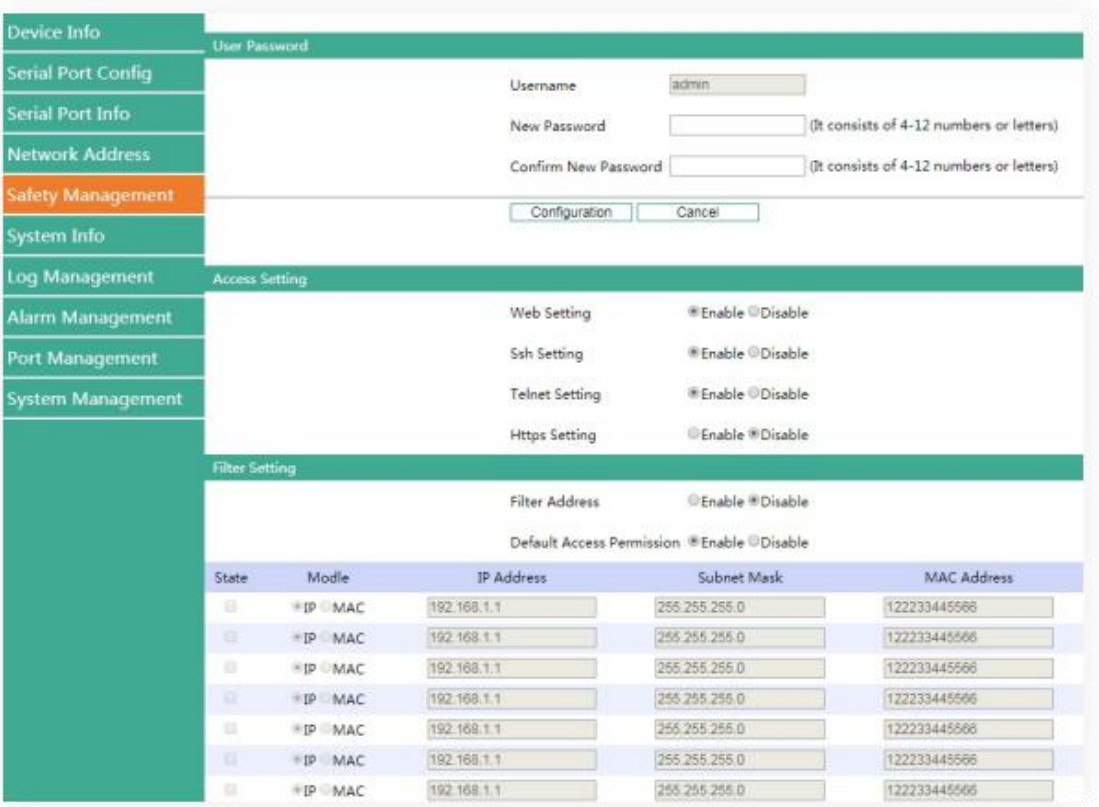


Figure 3-41 Safety Management Interface

3.6.1 User Password

The Web server of this Modbus gateway provides three different levels of user access.

- The first type is a visitor, who can only view the various current configurations of the Modbus gateway but cannot modify them. The username and password are both "none" and cannot be changed;
- The second type is a regular user, who can configure the various functional parameters of the Modbus gateway. The username is "admin," which cannot be changed, and the initial password is "admin," which can be modified on this page;
- The third type is an administrator, who has the highest privileges. If a regular user forgets their password, the administrator can log in and change the local password. The username is fixed as "admin," and the password is the last six digits of the local eth2 network card's MAC address.

  (If you do not know the local MAC address, you can first log in as a visitor to check.)

The login password must consist of valid characters, including 4-12 letters (case-sensitive), numbers, and special symbols. When changing the password, it must be entered twice to ensure both entries match.

The page is shown in Figure 3-42.

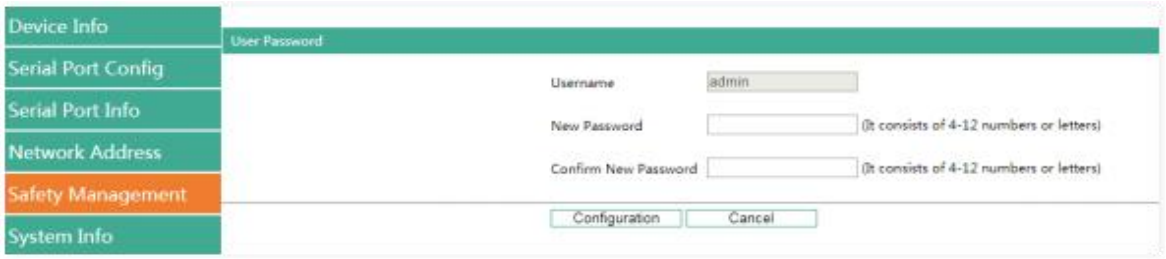


Figure 3-42 User password interface

- Username: The local username is fixed as "admin" and cannot be changed.
- New Password: Set the user password for this group, consisting of 4-12 letters (case-sensitive), numbers, and special characters.
- Confirm New Password Confirm: Re-enter the password to prevent input errors.

3.6.2 Access Control

In the access control bar, you can configure whether the web, ssh, and telnet consoles are enabled. By default, the web function is enabled, and the https option is used to configure whether the web supports the https function, and it takes effect when the web function is enabled. Serial port, ssh, telnet console login account: user login password: user@123

The serial port, SSH, and Telnet console can only modify basic network parameters, as well as the account passwords for the serial port, SSH, and Telnet console. After modifying the console parameters, the command line must execute the reboot command for the changes to take effect. The console command list page is shown in Figure 3-43.

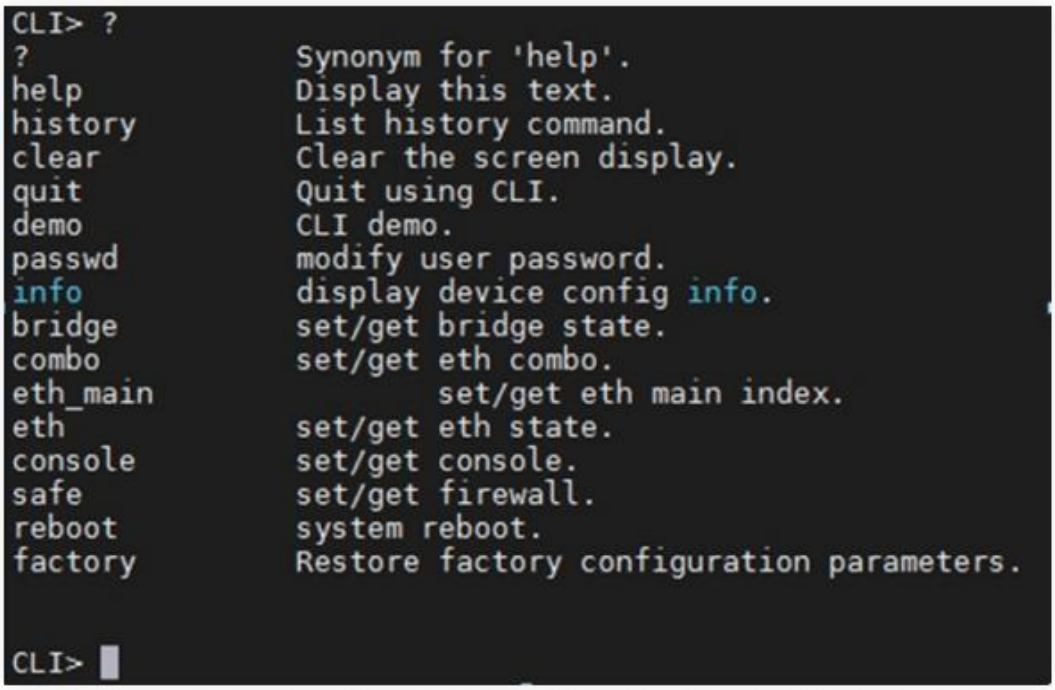


Figure 3-43 Serial Port, Telnet, SSH Console CLI Interface

3.6.3 Filter Settings

The address filtering bar is used to configure whether to implement network access permission control.

过滤设置

地址过滤 ○启用 ◉禁用

默认访问许可 ◉启用 ○禁用

状态	模式	IP 地址	子网掩码	MAC 地址(格式 XX:XX:XX:XX:XX:XX)
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66
☐	IP MAC	192.168.1.1	255.255.255.0	12:22:33:44:55:66

配置 取消

3.7 System Information

The system information page can configure the device's model, name, and number, as shown in Figure 3-44.

Figure 3-44 System information interface

- **Device Type**: Users can customize the model of the Modbus gateway.
- **Device Name:** Users can customize the name of the Modbus gateway.
- **Device Family:** Manufacturer platform name, user cannot configure.

- **Device ID:** Modbus gateway factory number, user cannot configure.

3.8 Log Management

This page manages user information for the Modbus gateway's operation logs, including local storage, network storage, and serial port log output, as shown in the figure 3-45.

Figure 3-45 Log Management

- **Local Storage:** This feature is used to save key information from the operation logs of the Modbus gateway internally on the device. When local logging is enabled, clicking the [Export] button will download the local logs in text format to the PC. If local logging is disabled, the local logs cannot be exported. Since local storage logs frequently overwrite the internal storage chip of the Modbus gateway, it is advisable not to enable the local storage function unless necessary.
- **Network Storage:** This function is used to transmit the operation logs of the Modbus gateway to a remote UDP server via UDP. Users need to specify the IP address and remote port of the remote UDP server.
- **Serial port log output:** This function is used to output the operation log of the Modbus gateway to a specified serial port. Users can select a serial port number from the drop-down menu. Once a certain serial port is selected as the log output port, other transparent transmission or Modbus functions of that port will be disabled.

3.9 Alarm Management

This page can configure mailbox alarms and SNMP alarms, as shown in page 3-46.

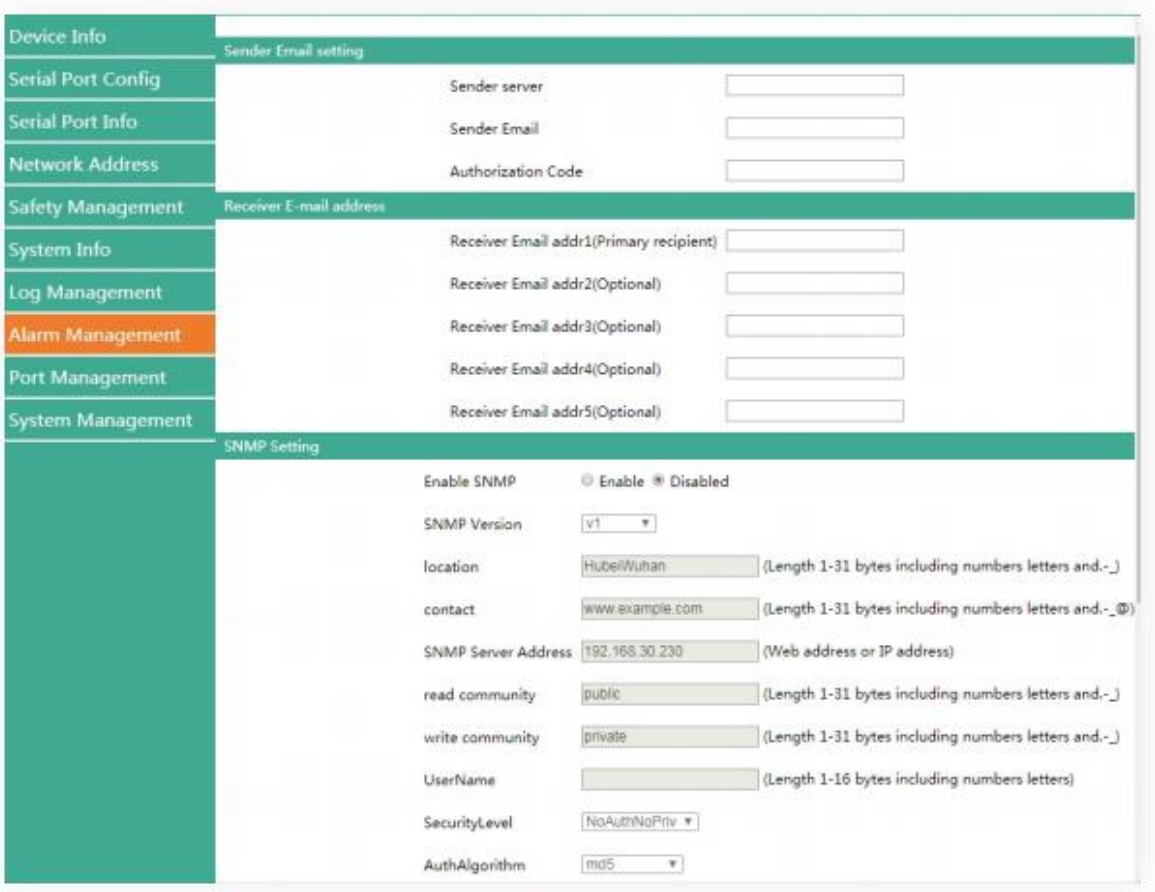


Figure 3-46 Alarm Management Interface

- **Sender Email Settings:** This section is used to configure parameters such as the sending server, sender email, authorization code, etc.
- **Receiver Email Address:** This section is for filling in the recipient email list, and alarm events will be sent to the emails in the recipient list.
- **SNMP Settings:** This section is used to configure whether to enable SNMP functionality, SNMP version, device installation address, contact information of the device maintainer, SNMP server address, read-only community name, write community name, and other parameters.
- **Alarm Event Settings:** This section is used to configure whether to enable device reboot alarms, device web login alarms, device parameter configuration change alarms, and administrator password change alarms. When the email function is filled in, notifications will be sent to the specified email; when SNMP functionality is enabled, notifications will be sent to the specified SNMP server.

3.10 Port Management

This page allows for rebooting the specified serial port of the Modbus gateway, as shown in Figure 3-47.

Figure 3-47 Port management interface

Check the corresponding serial port, click "Apply" to restart the corresponding serial port.

3.11 System Management

This page allows for some system operations on the Modbus gateway, including rebooting, restoring factory settings, and upgrading. Users are advised to use these functions with caution, as improper operation may damage the Modbus gateway.

- Device Restart: This function is used for the software restart of the Modbus gateway. The device will not function and cannot forward any data packets until the Modbus gateway has successfully restarted completely. This restart is different from a hardware reset that occurs when power is cycled; it is merely a reset of the Modbus gateway's system software, similar to a "hot reboot" in the Windows operating system. The main benefit of this function is that it provides a way to remotely restart the Modbus gateway. As long as the user can remotely access the Modbus gateway, they can restart it remotely. Click the [Start] button, and a prompt box will pop up as shown in Figure 3-49. Click [OK] to be redirected to the waiting page. Once the progress bar on the page completes, the Modbus gateway restart will be finished.

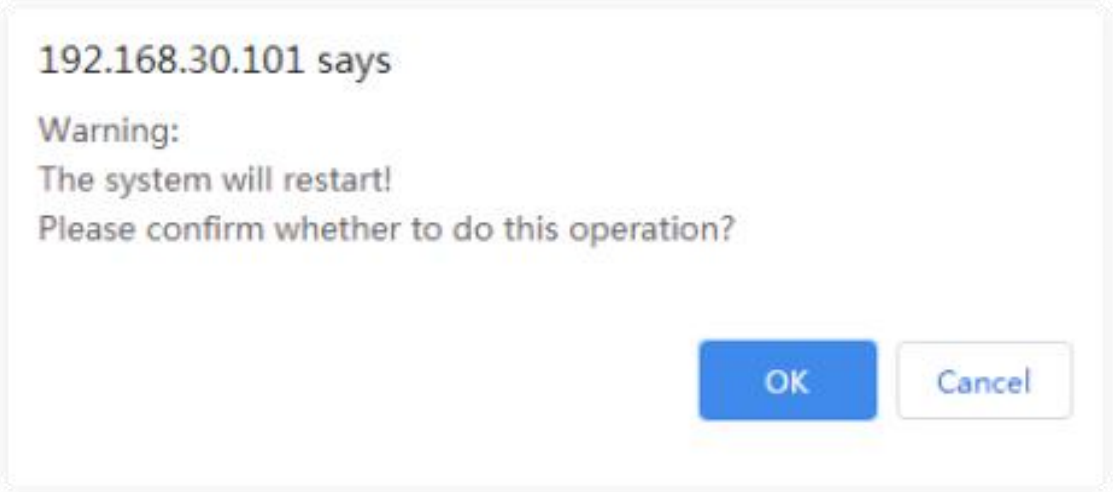


Figure 3-49 Warning message of device restart

- Factory Reset of the Device: This function is used to restore the Modbus gateway to its factory settings, while automatically rebooting the Modbus gateway. The Modbus gateway will not function and cannot forward any data packets until it successfully reboots. This function allows users to restore the factory default configuration values when incorrect parameters have been set, causing the Modbus gateway to operate abnormally. On the right side, there is an option to "Keep Current IP Address." When checked, the current IP address will be retained; if unchecked, the IP address will also be restored to the factory default address: 192.168.16.253. Click the [Start] button, and a prompt box will appear, as shown in Figure 3-50. Click [OK] to proceed to the waiting page. Once the progress bar on the page completes, the Modbus gateway will have successfully restored its factory configuration.

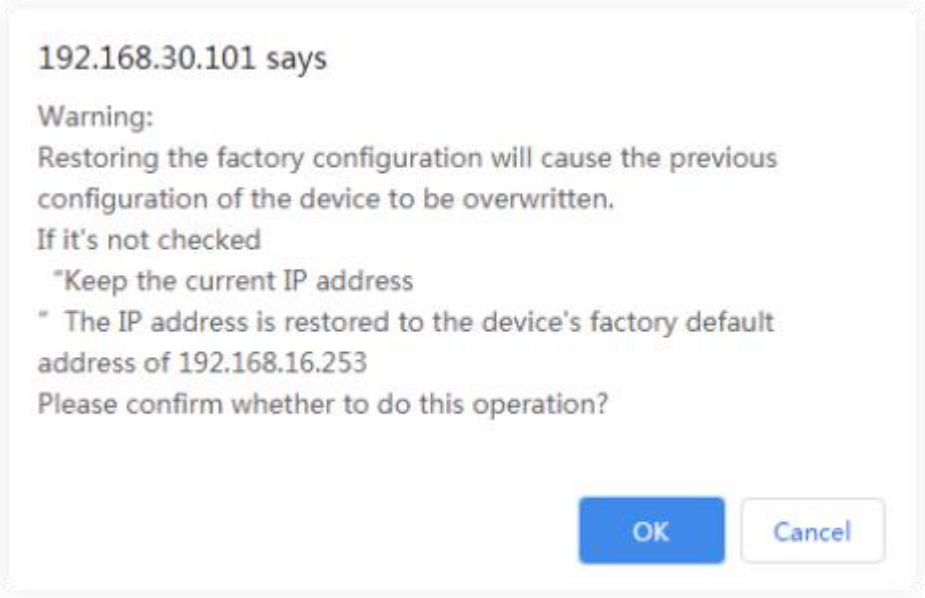


Figure 3-50 The warning message of restoring the factory configuration of the device

- **Device Upgrade:** This function is used for a system upgrade of the Modbus gateway software. Users can obtain the upgrade program for the Modbus gateway via email or the company's website. Please pay attention to the compatibility of the device model and version; using an incompatible upgrade program will result in a failed upgrade. After obtaining the upgrade program, the user should click the [Browse] button to select the upgrade program, then click the [Start] button. A prompt box will pop up, as shown in Figure 3-51. Click [OK] to proceed to the waiting page. Once the progress bar on the page completes, the Modbus gateway software upgrade will be finished.

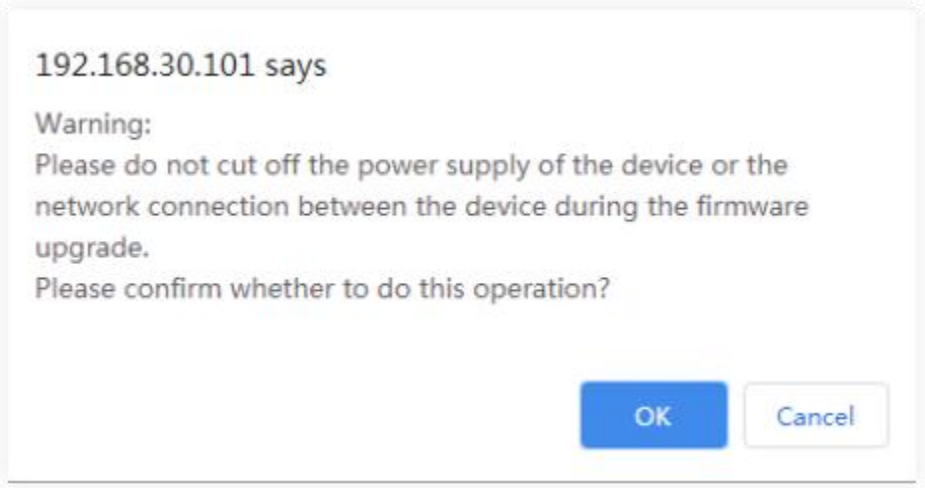


Figure 3-51 Warning message for device system upgrade

- **Certificate Status:** This function is used to upload the authentication certificate for establishing a connection with the Modbus gateway, and a total of 2 can be uploaded (Cgate3202) or 4 (Cgate3204) or 8 (Cgate3208) certificate options.
- **No Data Device Restart:** This function is used for the Modbus gateway to automatically restart when there is no data transmission on the network port or serial port for an extended period. After enabling this function, the no data time setting range is 60 to

65535 seconds. When disabled, this function will not take effect.

- **NTP Time Synchronization:** This feature allows the Modbus gateway to synchronize time information with an NTP server. Users must specify the IP address or domain name of the NTP server and select the local time zone from the drop-down menu.
- **Backup Configuration File:** This feature is used to back up the device's configuration parameters.
- **Restore Configuration File:** This feature is used to import the device's configuration parameters. The import file can be obtained from the backup configuration files. Checking the box to "Keep Current IP Address" will prevent changes to the device's network parameters; all other parameters will be updated to match those in the imported configuration file.

Note:

- Restoring factory settings will cause all configured states to revert to their original factory state. If you want to keep the IP address, please check the box on the right that says "Keep current IP address"; otherwise, the IP address will also revert to the default configuration of 192.168.16.253.
- Do not upgrade the device casually. When an upgrade is necessary, you must ensure that the upgrade file is correct; otherwise, it may damage the device's software, leading to Modbus gateway failure.
- Please do not operate the Modbus gateway during the upgrade process. Do not click on the Modbus gateway web page. If an accidental operation causes the upgrade to be interrupted, please restart the Modbus gateway and try again.
- The entire upgrade process must not be interrupted by a power outage, as this may cause permanent damage to the Modbus gateway. If a power outage occurs during the upgrade, please immediately send the product to our company for possible solutions.
- When setting the no-data device restart parameters, avoid using Google Chrome version 68, as this may cause the configuration page to become unresponsive.
- The restore configuration function will retain the original configuration's device number, device name, MAC address, and other fixed parameters of the device.

4 Principles Overview

4.1 Modbus communication principle

The Modbus protocol is a client/server application layer protocol that follows the process below during communication:

- The client sends a request to the server.
- The server analyzes and processes client requests and sends results.
- If an error occurs, the server returns an exception code.

The message data field sent by the client to the server includes additional information, which the server uses to perform operations defined by the function code. If there are no errors related to the requested Modbus function in a correctly received Modbus ADU, the response data sent to the client includes the requested data; if there are errors related to the requested Modbus function, the field includes an exception code, which the server application can use to determine the next operation.

- Processing without errors: The host issues a data request, and after the slave receives the request and successfully verifies the checksum, it indicates that processing is error-free, and the operation defined by the function code is executed.

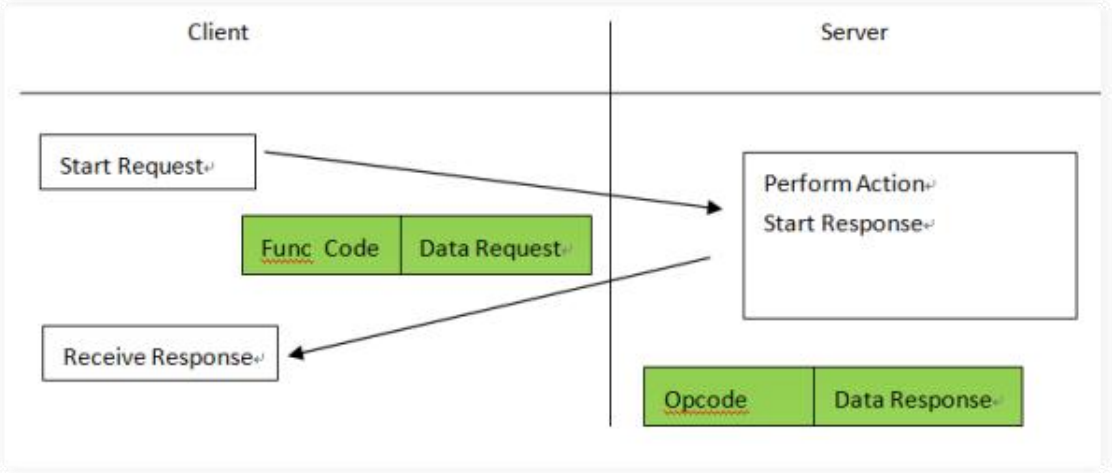


Figure 4-1 Modbus Normal Communication Diagram

- Processing during an exception response: The host sends a data request, and after the slave receives the request and fails to verify the checksum, it indicates that there was an error in transmission. It returns the error code and exception code.

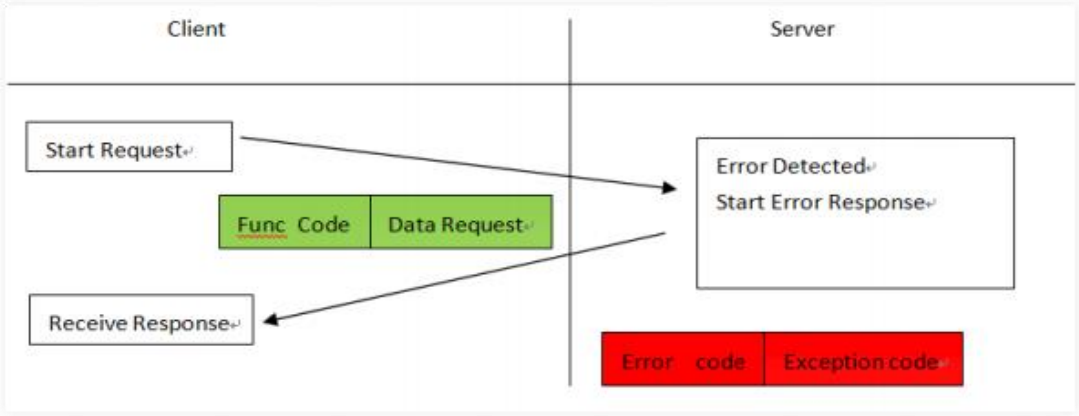


Figure 4-2 Modbus Exception Communication Diagram

4.2 Transmission Mode

4.2.1 Modbus RTU Mode

In RTU mode for Modbus serial link communication, each 8-bit byte in the message contains two 4-bit hexadecimal characters.

Main advantages: Higher data density, with a higher throughput compared to ASCII code mode at the same baud rate, allowing for the transmission of more data. The interoperability between Modbus devices requires that each device has the same mode, either RTU or ASCII code, with the default setting being RTU mode. RTU message frame:

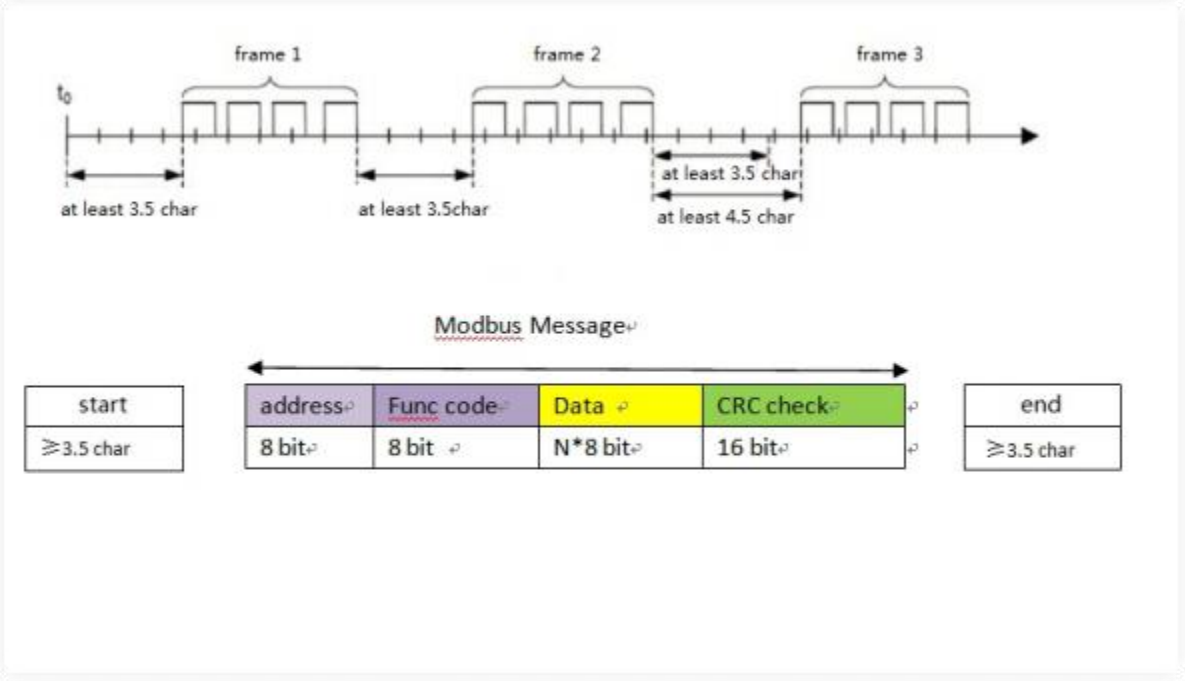


Figure 4-3 Modbus RTU Message Frame

4.2.2 Modbus ASCII Mode

When the devices on the Modbus serial link are configured to communicate using ASCII mode, each 8-bit byte in the message is represented by two ASCII

Character transmission. This mode is used when the communication link

or device cannot meet the timing management of the RTU mode.

Advantage: The time interval for character transmission can reach 1

second without errors.

Disadvantage: This mode is less efficient than RTU because one byte requires two characters.

ASCII code message frame:

Start	address	Func code	Data	LRC	End
1 char	2 char	2 char	0~2*252 char	2 char	2 char CR,LF

Figure 4-4 ASCII message frame

RTU and ASCII mode comparison:

Comparison of Modbus RTU and ASCII modes:		
Mode	RTU	ASCII
Coding System	8-bit binary	Hexadecimal
	Each ASCII character in the data contains 2 hexadecimal characters	Each ASCII character in the data contains 1 hexadecimal characters
Number of bits per byte	1 starting position	1 starting position
	8 data bits	8 data bits
	Odd/Even Parity Check 1 Bit	
	Parity Check None	
	If there is a parity check, there is 1 stop bit	
	If there is no parity check, there are 2 stop bits	
Error Checking	CRC	LRC

4.2.3 Modbus TCP Mode

The Modbus protocol defines a simple protocol data unit (PDU) that is independent of the underlying communication layer. Modbus protocol mapping can introduce some additional fields on the application data unit (ADU) for specific buses or networks.

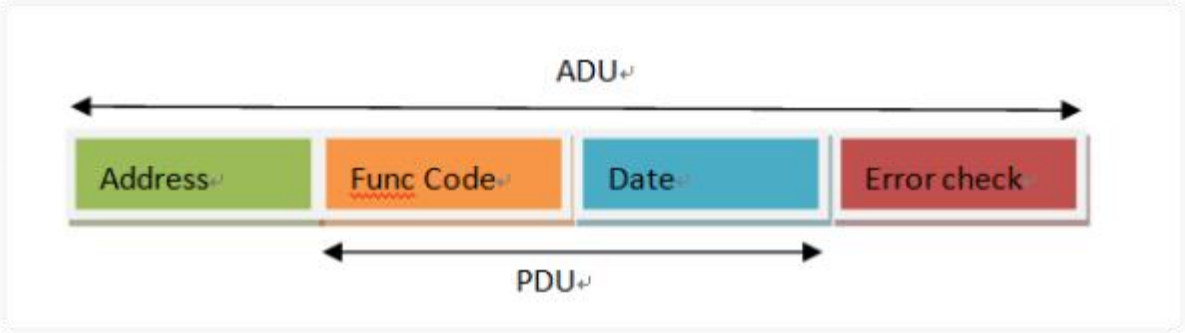


Figure 4-5 General Modbus frame

The client initiating the Modbus transaction establishes the Modbus Application Protocol Data Unit. This function code indicates to the server which operation to perform. The Modbus Application Protocol Data Unit over TCP/IP: describes the encapsulation of Modbus requests or responses occurring in the Modbus TCP/IP network.

Modbus Request/Response over TCP/IP:

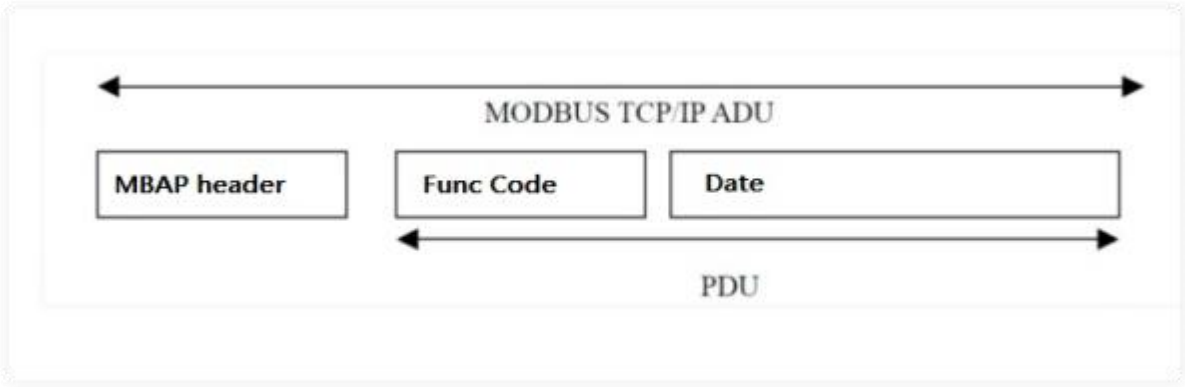


Figure 4-6 Modbus request/response over TCP/IP

A dedicated message header is used to identify the Modbus Application Protocol Data Unit over TCP/IP. This message header is referred to as the MBAP header (Modbus Application Protocol header). This header provides some differences compared to the Modbus RTU Application Protocol Data Unit used on serial links:

- The single byte unit identifier in the MBAP header replaces the Modbus slave address field typically used on Modbus serial links. This unit identifier is used for communication with devices that support multiple independent Modbus terminal units using a single IP address, such as bridges, routers, and gateways.

- All Modbus requests and responses are designed in a way that allows the receiver to verify the completion of the message. For function codes with a fixed length in the Modbus PDU, the function code alone is sufficient. For function codes that carry variable data in the request or response, the data field includes the byte count.
- When carrying Modbus over TCP, even if the message is divided into multiple packets for transmission, additional length information is carried in the MBAP header so that the receiver can identify the message boundaries. The existence of explicit and implicit length rules, along with the use of the CRC-32 error check code, will result in minimal undetected interference in the request or response messages.

MBAP Header Description

The MBAP header includes the following fields:

Domain	Degree	Description	Client	Server
Transaction Identifier	2 bytes	MODBUS request/response transaction identifier Identification code	Client startup	Server from the received request Recopy
Protocol identifier	2 bytes	0=MODBUSProtocol	Client startup	Server from the received request Recopy
Length	2 bytes	The number of bytes below	Client startup (request)	Server (Response) Startup
Unit Identifier	1 byte	Connected via serial link or other bus The identification code of the remote slave station	Client startup	Server from the received request Recopy

The message header is 7 bytes:

- Transaction Identifier: Used for pairing transactions. In the response, the Modbus server copies the transaction identifier from the request.
- Protocol Identifier: Used for multiplexing within the system. Identified by the value 0 for the Modbus protocol.
- Length: The length field is the byte count of the next field, including the unit identifier and data field.
- Unit Identifier: Used for routing within the system, this field is specifically for communication between the Modbus or Modbus+ serial link slave and the gateway over Ethernet TCP-IP networks. The Modbus client sets this field in the request, and the server must return this field with the same value in the response.

4.3 Modbus Basic Architecture Diagram

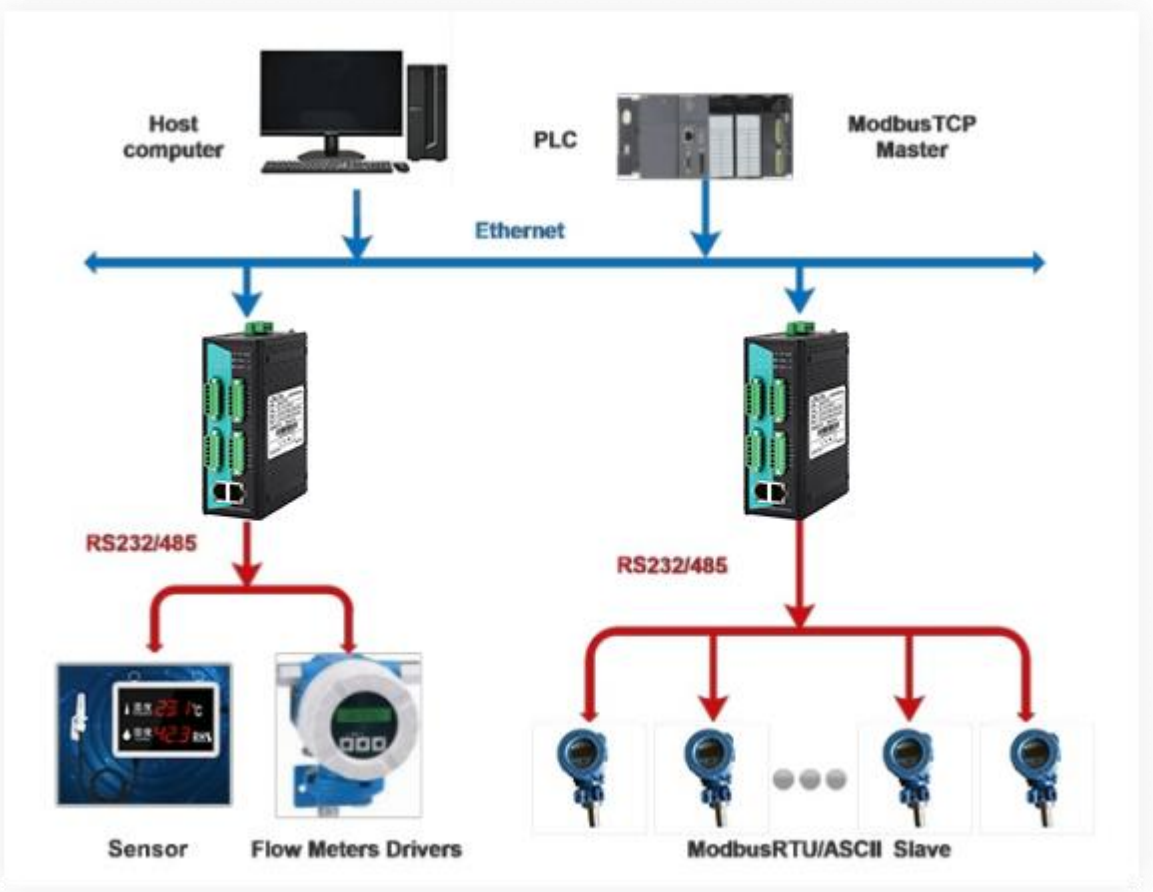


Figure 4-7 Modbus Basic Architecture Diagram Slave Mode

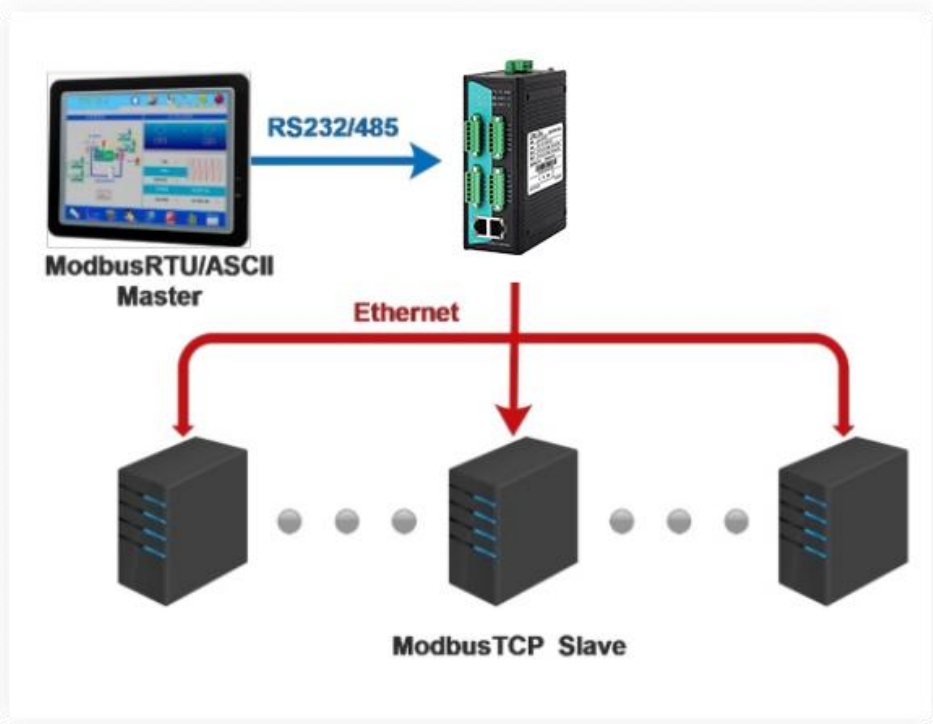


Figure 4-8 Modbus Basic Architecture Diagram Master Mode

4.4 Modbus Parameter Settings

4.4.1 Initial delay

When the Modbus gateway device is powered on, it delays for a period before starting Modbus. This function is designed to accommodate scenarios where the slave device has a long startup time. To avoid sending a request message before the slave has completed initialization, the gateway sets an initialization delay time to determine when it can send the first request frame.

4.4.2 Modbus TCP Exception

This parameter is used in Modbus monitoring software to check for any Modbus TCP communication anomalies. If the data returned by the slave is abnormal or the ID address is incorrect, the gateway will assemble a message to notify the master station of the communication anomaly.

4.4.3 Modbus Reponse Timeout

Modbus is a master-slave protocol, where the master requires a response timeout when accessing the slave. This timeout is referred to as the "Modbus receive timeout" of the gateway. This parameter exists between steps 2 and 3 in the figure below. After the gateway forwards the request to the slave device, if no response is received within this parameter time, it is considered a timeout response, and an exception response message is sent back to the master device.

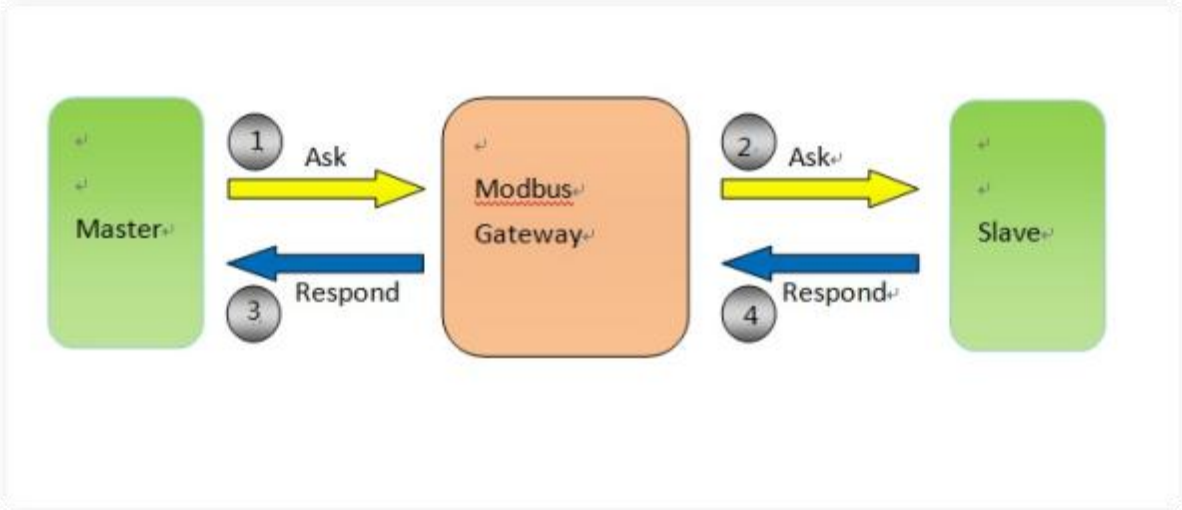


Figure 4-9 Modbus communication process

4.4.4 Interval Time

In an RTU frame, the time interval between this frame and the next frame is the character interval time. In some customer sites, the serial port does not have FIFO buffering, which does not meet the timing standards of the Modbus protocol. However, the character interval

time can be defined by the customer to accommodate older devices with poor timing requirements.

4.4.5 Frame Interval Delay

The time interval between the current RTU response and the next RTU request is set to 0ms by default. This function is designed to prevent the slave device from being unable to quickly process RTU requests, allowing for a time interval that gives the RTU slave device sufficient time to handle the request.

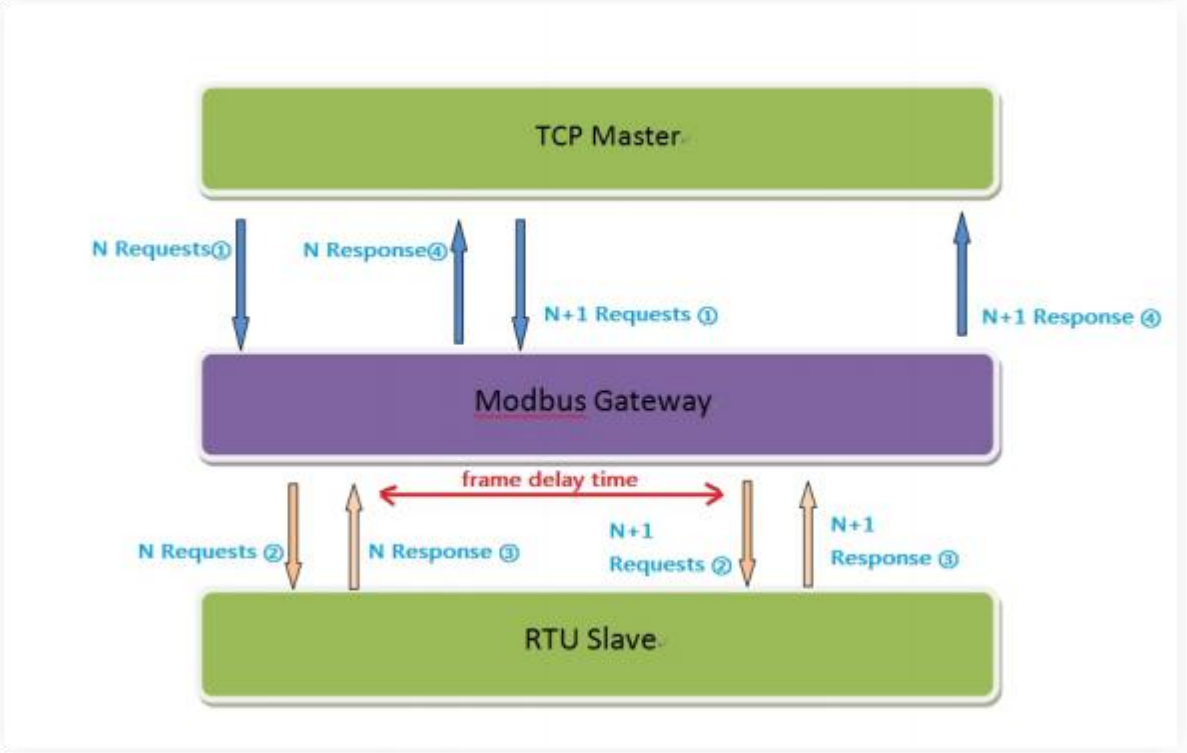


Figure 4-10 Modbus communication process

4.4.6 Slave Address Mapping

The Modbus protocol stipulates that all slave devices must have a unique ID number (1~247). This ID number is used to identify the slave address in response to requests from the master device. The Modbus device ID number is set by the manufacturer.

Slave Station ID Mapping: Each slave device has 2 ID addresses, namely the virtual ID address and the real ID address. The real ID exists in the slave device, and other devices access the slave device directly through the real ID. The virtual ID exists in the gateway and serves as the unique access address for the slave device in the gateway, allowing other devices to indirectly access the slave device through the virtual ID.

ID Message Processing Flow:

- ① The master station device uses a virtual ID to access the gateway;
- ② The gateway converts the virtual ID into a real ID to access the slave station device;

- ③ The slave station device responds using the real ID;
- ④ The gateway converts the real ID back into a virtual ID to send the response.

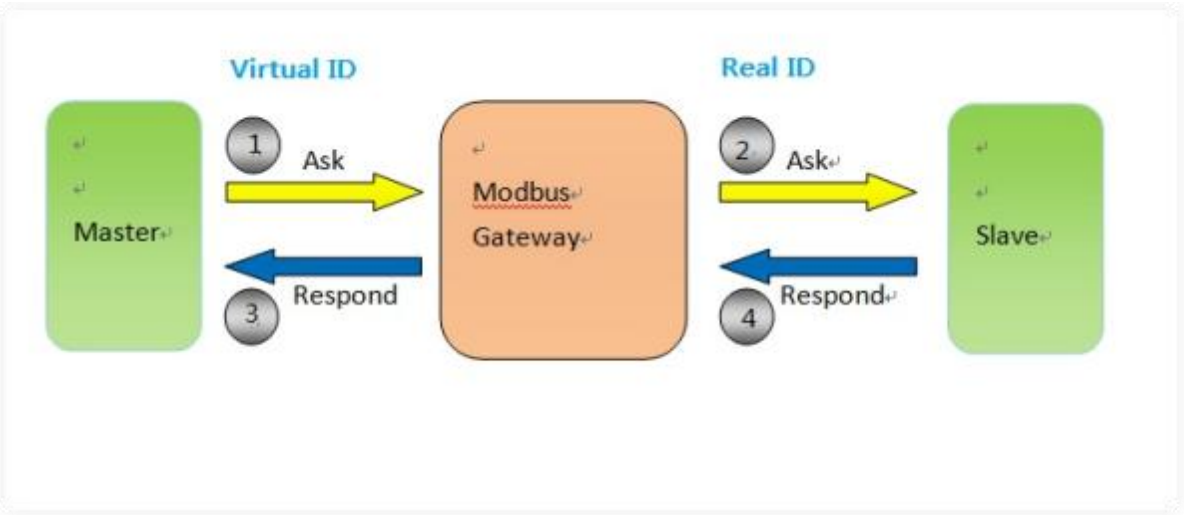


Figure 4-11 ID mapping

The gateway implements the function of ID mapping by setting the ID range and ID offset, with the ID range starting from the slave's starting ID to the slave's ending ID. Only channels with a slave mode type can establish mapping, including serial port slaves (RTU Slave or ASCII Slave) and TCP Slave. The relationship of the Slave ID mapping table is Virtual ID + ID Offset = Real ID, where the ranges of Virtual ID and Real ID are both 1~247.

The gateway implements the function of ID mapping by setting the ID range and ID offset, with the ID range starting from the slave's starting ID to the slave's ending ID. Only channels with a slave mode type can establish mapping, including serial port slaves (RTU Slave or ASCII Slave) and TCP Slave. The relationship of the Slave ID mapping table is Virtual ID + ID Offset = Real ID, where the ranges of Virtual ID and Real ID are both 1~247.

5 Maintenance and Service

From the date of product shipment, COME-STAR COMMUNICATION (WUHAN) CO., LTD. provides a five-year product warranty. According to the product specifications of COME-STAR COMMUNICATION (WUHAN) CO., LTD., during the warranty period, if the product has any faults or operational failures, COME-STAR COMMUNICATION (WUHAN) CO., LTD. will repair or replace the product free of charge for the user. However, this commitment does not cover damages caused by improper use, accidents, natural disasters, incorrect operation, or improper installation. To ensure consumers benefit from the series of products from COME -STAR COMMUNICATION (WUHAN) CO., LTD., assistance and problem resolution can be obtained through the following methods:

- Internet Service
- Technical Support Phone Service
- Product Repair or Replacement

5.1 Internet Service

You can find more useful information and usage tips on the COME-STAR COMMUNICATION (WUHAN) CO., LTD. website.

5.2 Technical Support Phone Service

Users of COME-STAR COMMUNICATION (WUHAN) CO., LTD. products can call the technical support office of COME-STAR COMMUNICATION (WUHAN) CO., LTD. where professional technical engineers will communicate with you to help resolve any product usage issues you encounter as quickly as possible.

5.3 Product Repair or Replacement

For product repairs, replacements, or returns, according to the procedures of COME-STAR COMMUNICATION (WUHAN) CO., LTD., you should first confirm with the technical personnel of COME-STAR COMMUNICATION (WUHAN) CO., LTD., and then negotiate with the sales personnel of COME-STAR COMMUNICATION (WUHAN) CO., LTD. to complete the product repair, replacement, or return.