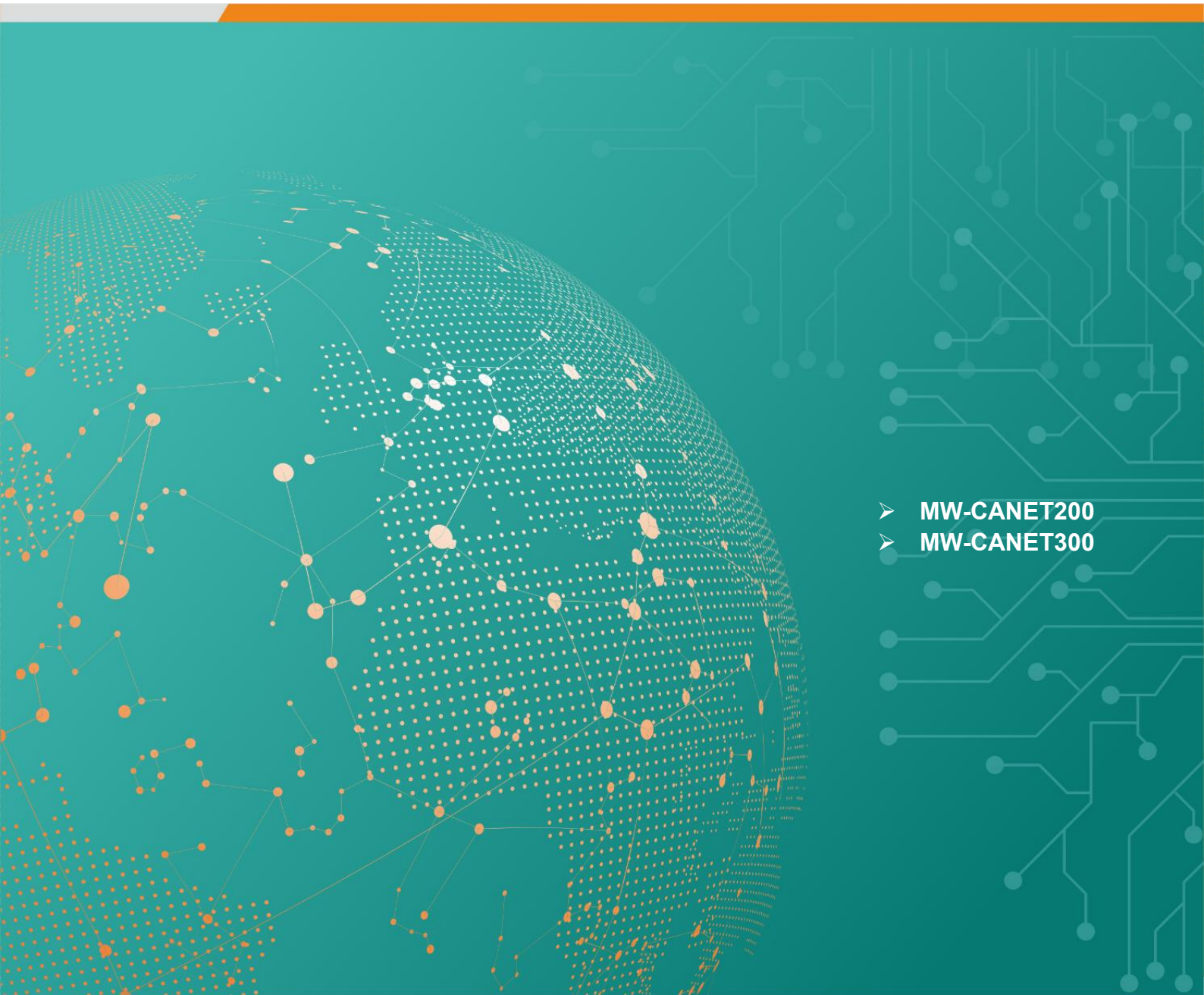
Maiwe

CAN server

Software Manual



- MW-CANET200
- MW-CANET300

Version number: V1.0

Please read this installation manual carefully before using this product

Contents

1 Basic Information

The CAN server has a built-in web server, which provides a convenient way to access and configure the CAN server. Users can use Firefox or Google browser to access it.

If you need product-related information, you can download the corresponding product manual from the official website link: http://www.maiwe.com.

1.1 Environmental Preparation

Hardware Connection

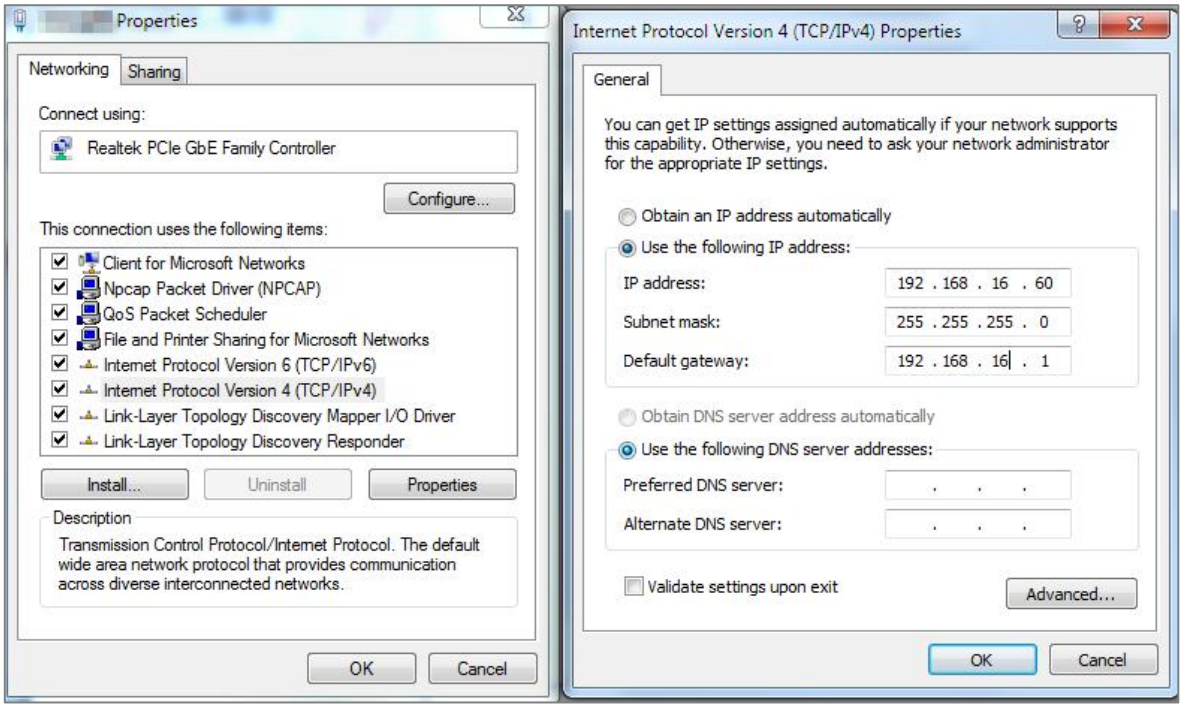
1.2 Log in Web

1.2.1 Revise IP address

When accessing the CAN server through the Web, the IP address of the serial server and the PC must be in the same network segment, so the IP address of the PC must be modified to ensure that it is in the same local area network as the IP of the serial server. For Windows users, please refer to the following operations:

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

The default IP address of this model of serial server is: 192.168.16.253. Set the PC's IP address as: 192.168.16.X (X is any valid value from 2 to 253 except 253). The specific Windows system operation page is shown in below Figure

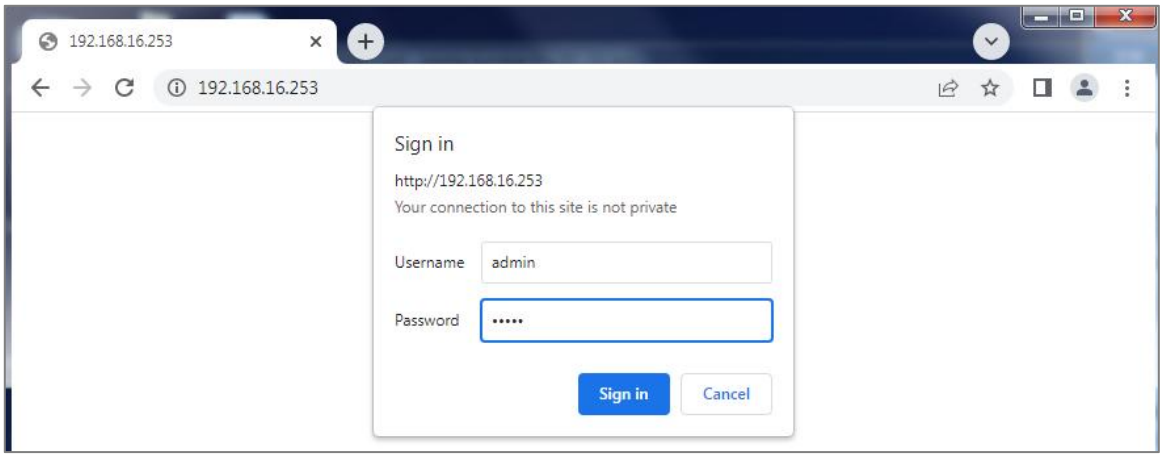


After changing the IP address of the PC, you can access the Web page of the CAN server through the default IP address 192.168.16.253, and perform related configuration operations on It.

1.2.2 Login in WEB

Open the browser and enter the default IP address of the serial server in the address bar, as shown in below:

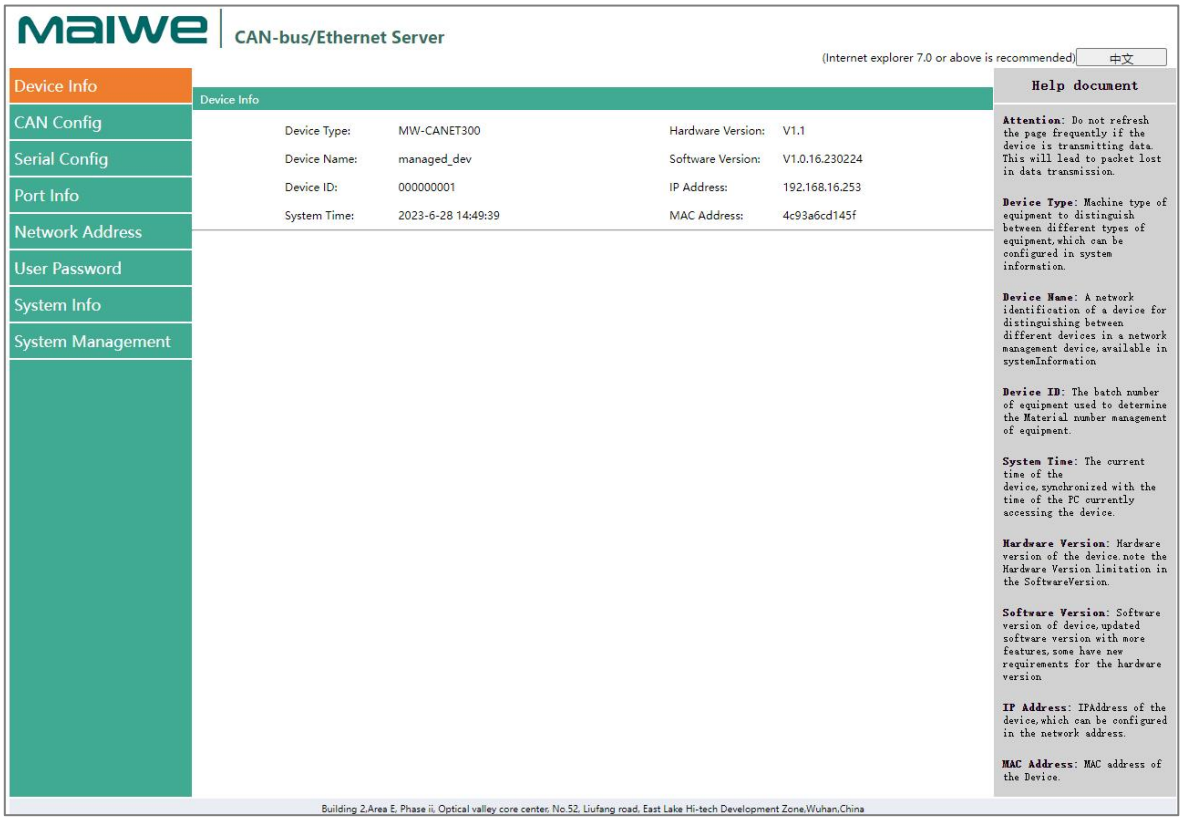
After hitting the Enter key, the window shown in Figure 5 pops up, prompting the user to enter the user name and password.



The login users of this CAN serial server are divided into three types.

The first is a normal user, the user name and initial password are both "admin", it is used when accessing the Web normally; the second is a guest, the user name and password are both "none", after logging in, only the configuration of the current serial server can be viewed, And can't be configured; the third type is the administrator, the user name is "admin", and the password is the last six digits of the serial server's MAC address. When we forget the password of an ordinary user, we can log in with the administrator account and modify this Machine

password. After entering the user name and password, click "OK" and the server will authenticate. After success, you will enter the main page of the Web server. As show in below figure:

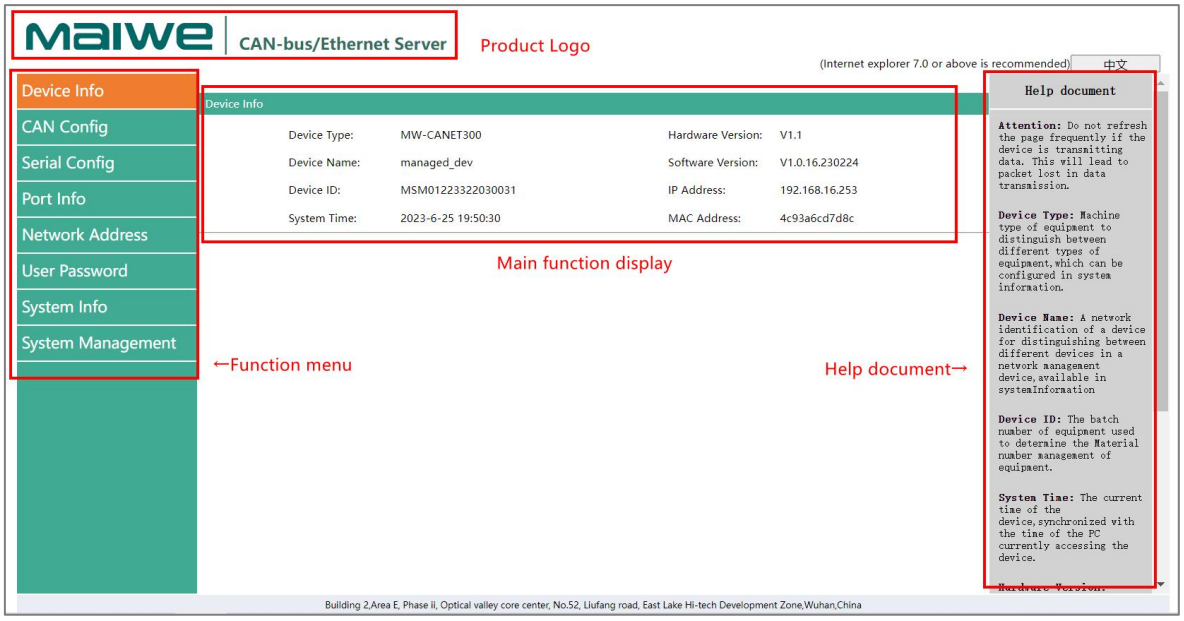


Attention:This CAN serial server has been tested extensively with Firefox, and Google's mainstream browsers, and it can be used normally, but it is recommended to use Google browser when upgrading the device.

2 Network Management Function

2.1 Main Page Introduction

After entering the correct user name and password and the authentication is successful, you will enter the main page of the Web, as shown in Figure 7. 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.



2.1.1 Function menu

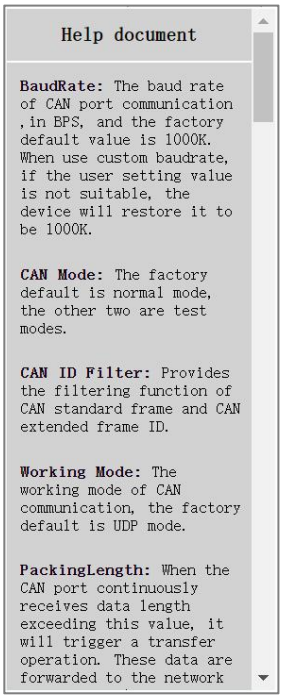
The left side of the webpage is the function menu area, which displays all the configurable software functions of CAN series serial server. The function menus are basic information, serial port, serial port information, network address, user password, system information and system management. Each function menu contains several sub-functions. Its function is shown in below:

Basic information	Device Information	Display device information, such as: name, number, software version, IP address, etc.
CAN Configuration	CAN server configuration	Configure the basic information of each port, such as: CAN port number.
	CAN basic parameters	Configure the working mode, baud rate, packet length, etc. of each port.

	Network parameters	Configure the local port range and heartbeat interval of each port.
	Number of network connections	Configure the destination address and remote port of each port.
Serial port configuration	Serial server configuration	Configure the basic information of each port, such as: serial port number.
	Serial basic parameters	Configure the working mode, baud rate, packet length, etc. of each port.
	Network parameters	Configure the local port range and heartbeat interval of each port.
	Number of network connections	Configure the destination address and remote port of each port.
Port information	Serial port information	Display the serial port number and the total number of serial ports received and sent.
	Network connection mode	Display the working mode of the serial port/local port and other information.
Website address	Website address	Configure the IP address, subnet mask and default gateway for serial communication.
User password	User password	Configure username and password.
System info	system info	Configure the device model, device name, etc. of the device.
	Device restart	Configure the restart function of the device.
System management	Factory reset of the device	Configure the device to restore factory values.
	Equipment upgrade	Configure the upgrade file of the device.
	No data device restart	The network and serial port of the configuration device have no data transmission for a certain period of time, and the device restarts.

2.1.2 Help documentation

The function area at the bottom right is the help document. Click any main function page in the lower left, and the help document will be displayed in the lower right function area corresponding to the main page, as shown in below:



2.2 Basic information

The basic information module includes: device information. The function of the device information part is to display some specific information of the current device, including device model, device name, device number, system time, hardware version, software version, IP address and MAC address. As shown in below:

- **Device Type**: The model of the CAN server, which can be customized by the user on the "System Information" page.
- **Device name**: The name of the CAN server, which can be customized by the user on the "System Information" page.
- **Device number**: CAN port server number.
- **System time:** The current time of CAN server is synchronized with the time of the PC accessing the serial server.
- **Hardware version**: the current hardware version of the CAN server.
- **Software version:** the current software version of CAN server.
- **IP address**: the IP address of CAN server.
- **MAC address:** the MAC address of CAN server.

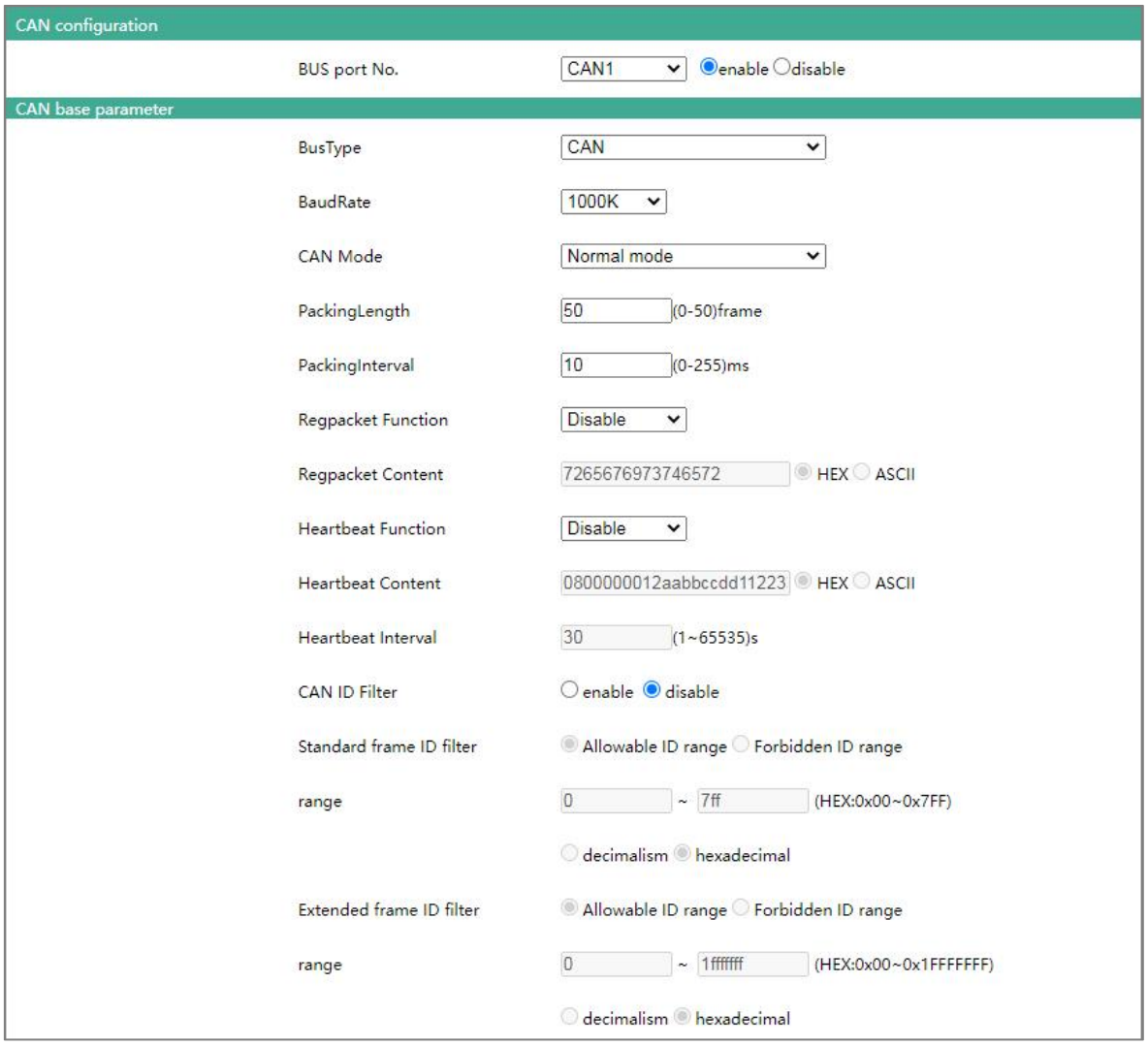
2.3 Port Configuration

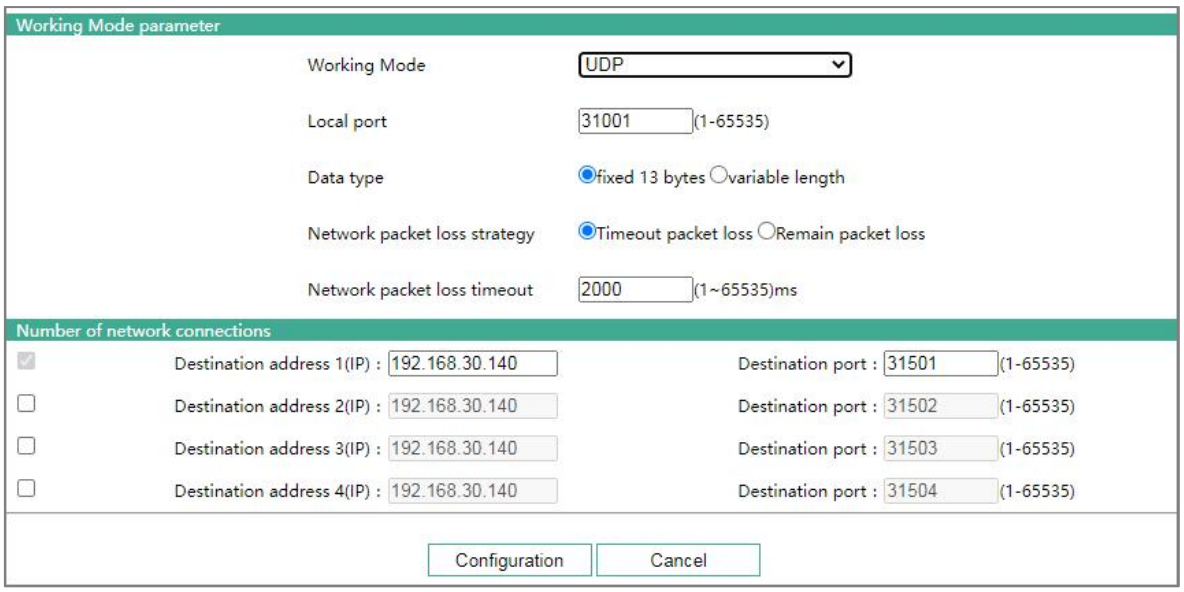
CAN configuration includes: CAN basic parameters, network parameters, and number of network connections.

Serial port configuration includes: serial port parameters, network parameters, and number of network connections.

2.3.1 CAN port configuration

The main function of CAN Ethernet server is to bidirectionally transmit the CAN bus protocol and the standard Ethernet data supporting the TCP/IP protocol to solve the networking problem of ordinary CAN bus devices on the Internet. The port configuration page can configure the parameters required by CAN bus and Ethernet related parameters is show below:





The detailed description of the configuration parameters on this interface

Item	Instruction
BUS port No.	Select the current CAN port to be configured, enable or disable it.
CAN basic parameters	
CAN Bus type	The default is CAN bus.
Baud rate	The baud rate of CAN bus communication, the unit is bps, the options are: 5K, 10K, 20K, 25K, 40K, 50K, 62.5K, 80K, 100K, 125K, 200K, 250K, 400K, 500K, 800K and 1000K. The factory default value is 1000K. The baud rate of each CAN port is independent of each other and can be set separately without affecting each other.
CAN mode	Divided into normal mode, loopback mode, monitor mode. Normal mode is used for normal communication; loopback mode and listen mode can be used for bus testing and troubleshooting.
Packing length	When the number of consecutive data frames received by the CAN port exceeds the value set by the packet length, a transmission operation will be triggered, and these data will be forwarded to the network port, with a value range of 1 to 60 frames.
Packing interval	When the intermittent time of receiving data at the CAN port exceeds this value, no matter how much data has been received, a transmission operation will be triggered, and the data will be forwarded to the network port, ranging from 0 to255ms.
Register package function	Select the sending method of the registration package; Disable: disable this function; LINK: The registration packet is only sent once when the network connection is established; DATA: The registration package is filled in front of the CAN port data each time the CAN port sends data to the network; This function is only allowed when the network working mode is UDP and

	TCP Client.
Register package content	The content of the registration package, the maximum length is 64 bytes. web is filled in in hexadecimal format.
Heartbeat packet function	Select the sending method of the heartbeat packet; Disable: disable this function; to Network: send the heartbeat packet to the network port; to CAN: send the heartbeat packet to the direction of the CAN port; This function is only allowed when the network working mode is UDP, TCP Server and TCP Client.
Heartbeat packet content	The content of the heartbeat packet is up to 64 bytes when it is sent to the network; when it is sent to the CAN port, it only supports a single frame and must be filled in according to the CAN frame format. web is filled in in hexadecimal format.
Heartbeat interval	The time interval for sending heartbeat packets, in seconds, the range is 1~65535 seconds, and the default is 30 seconds.
CAN ID filtering	Enable or disable the CAN ID filtering function. CAN ID filtering
Standard Frame ID Filtering	Allowable ID range or prohibited ID range is optional, parameter range: 0x00~0x7FF, web fills in hexadecimal format.
Extended Frame ID Filtering	Allowable ID range or prohibited ID range is optional, parameter range: 0x00~0x1FFFFFFF, web fills in hexadecimal format.
Network parameters	
Working mode	UDP, UDP Multicast, TCP Client, TCP Server, Pair Connection Master, Pair Connection Slave, Modbus TCP Slave, Relay Mode, CAN to Serial, HttpdClient, WebSocket Client and other communication methods are available. The factory default setting is UDP mode.
	When the working mode of this device is UDP, the remote device must also work in UDP mode. This device can establish a UDP connection with a remote device, and the IP address and port number of the remote device can be configured on the page.
	When the device is in UDP Multicast, the remote device must also work in UDP Multicast mode. This device can communicate with devices in the same multicast address, and the multicast address and port number of the remote device can be configured on the page.
	Whenthe device is in TCP Client mode, the remote device must work in TCP Server mode, and its IP address and port number mustbe configured, which can be configured in the corresponding options of the network connec tion. The local port number can be ignored and noconfiguration is required.
	When the device is in TCP Server mode, the remote device must work in TCP Client mode. In this mode, up to 4 remote TCP Client connections are accepted.
	When the device is in Pair Connection Master mode, the remote device must be operated in Pair Connection Slave mode, and its IP address and port number must be configured. You can connect to the corresponding

	options in the network In-progress configuration. The local port number can be ignored and no configuration is required.
	When the device is in Pair Connection Slave, the remote device must work in TCP Client mode. In this mode, up to 4 Pair Connection Master connections are accepted.
	When the device is in Relay Mode, CAN1 and CAN2 can transmit data to each other, at this time, the restart function of the device without data will be invalid. (Only MW-CANET200 supports this mode)
	When the device is in CAN to Serial mode, the device at the opposite end is required to transmitdata according to the configured protocol. This mode can convert the CAN bus data to the serial bus according to the established protocol. (Only MW-CANET300 supports this mode)
	When the working mode of this device is Modbus TCP Slave, if the Modbus RTU Over function is not enabled, the remote master device must use the Modbus TCP protocol for communication; otherwise, enable the Modbus RTU Over function, The remote master device must use the Modbus RTU protocol to communicate. This mode supports up to 4 connections.
	When the device is in HTTPD Client, the user needs to specify the address, port, method and other parameters of the remote httpd server. The device will submit the data received by the CAN port to the httpd server in the form of GET or POST. At the same time, the data sent by the httpd server can also be sent to the CAN port.
	When the working mode of the device is WebSocket Client, the user needs to specify the main parameterssuch as the address, port and method of the WebSocket server. You can also set the time interval of Ping. Keep the connection between the device and the server. The device will upload the data received by the CAN port to the WebSocket server, and can also send the data sent by the server to the CAN port.
Locol port	Local port on the network connection side
Heartbeat interval	When the network working mode is in TCP mode, the CAN server will send a TCP heartbeat detection packet at a specified interval to test whether the connection exists, and if it does not exist, the connection will be automatically disconnected, ranging from 1 to 6000s.
Overtime time	When the network working mode is in TCP mode, the CAN server will detect the idle time of the current connection and the corresponding CAN port, and will disconnect the TCP connection when it exceeds the set value.
Data mode	The network side can be set to process data according to a fixed 13-byte CAN protocol or process data according to a variable length.
Network packet loss mechanism	Divided into overtime packet loss and redundant packet loss. Timeout packet loss: If the set timeout period is exceeded, the extra unparsed bytes will be discarded, otherwise, it will be spliced with the next received data and parsed.

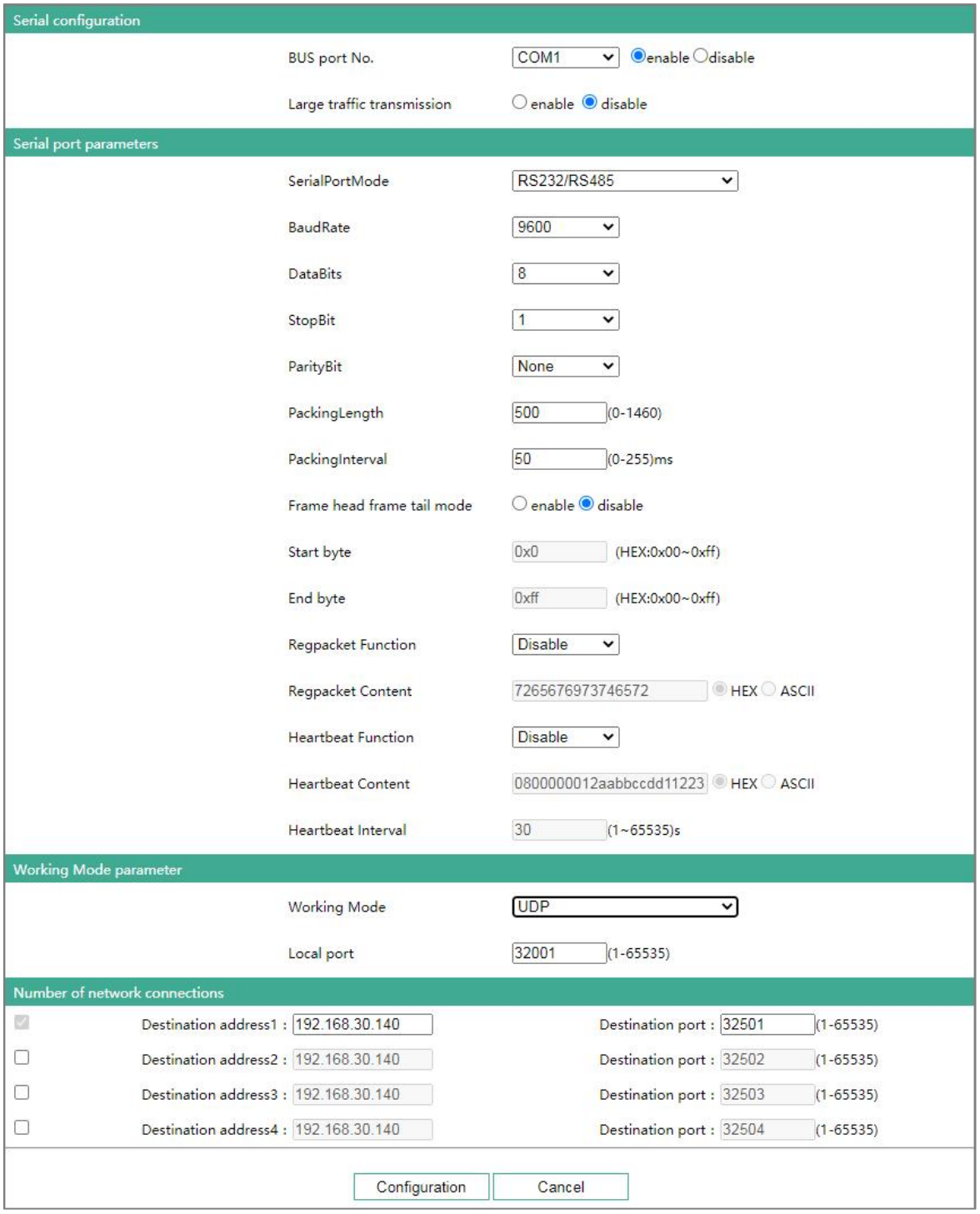
	Extra packet loss: Every time the network data is received, the unparsed ones are discarded directly.
Network packet loss timeout	It only takes effect when timeout packet loss is enabled, and the value range is 1~65535ms.
Modbus slave ID	Set the Modbus slave ID corresponding to the CAN port, value range: 1~247.
Modbus RTU Over	By default, the Modbus TCP protocol is used for network communication. After it is enabled, the Modbus RTU protocol is used for network communication.
Number of network connections	
Destination address	The IP address of the network connection peer
Destination port	The port number of the peer end of the network connection
Modbus read instruction table	
Ext	The Modbus read register corresponds to CAN commands in standard frame or extended frame format.
Rtr	The Modbus read register corresponds to the CAN command of the data frame or remote frame type.
CAN ID	The CAN ID corresponding to the Modbus read register.
Reg_addr	The Modbus read register address corresponding to the CAN command, the range is 0~127 in decimal.
Reg_len	A Modbus register occupies 2 bytes, and the value ranges from 1 to 8.
Modbus write instruction table	
Ext	The Modbus write register corresponds to CAN commands in standard frame or extended frame format.
Rtr	The Modbus write register corresponds to the CAN command of the data frame or remote frame type.
CAN ID	The CAN ID corresponding to the Modbus write register.
Reg_addr	The Modbus write register address corresponding to the CAN command ranges from 256 to 383 in decimal.
Reg_len	A Modbus register occupies 2 bytes, and the value ranges from 1 to 8.
Cmd_en	By default, the CAN command is not specified, and at this time, the corresponding CAN command is sent according to the data in the write command sent by Modbus; otherwise, it is sent to the CAN port according to the specified Cmd command.
Cmd	Cmd_en is configured to take effect when specified, and the commands sent to the CAN side are stored here.
Mode	CAN commands can be configured to be executed periodically or triggered by data, and are executed periodically by default.
Cycle	The cycle of CAN command sending takes effect when Mode is cycle execution. The value range is 1~65535, unit: millisecond.
HTTPD configuration	
HTTPD address	The IP address of the server.
HTTPD port number	The port number of the server.

HTTPD method	The server method supports GET and POST methods. Note: When using the GET method, the device sends hex data in ascii form to the server.
HTTPD request header	The server's request header.
WebSocket configuration	
WebSocket address	The IP address of the server.
WebSocket port number	The port number of the server.
WebSocket method	Method of the server.
WebSocket ping	The ping interval of the server, the value range is 0~255 seconds.

2.3.2 Serial port configuration

Only MW-CANET300 supports this communication method.

The main function of the serial Ethernet server is to carry out two-way transparent transmission of standard serial bus data (RS232, RS485, RS422) and standard Ethernet data supporting TCP/IP protocol, which is used to solve the networking problem of ordinary serial devices on the Internet. The parameters of the serial port Ethernet server can be configured on the serial port server configuration page, as shown in the figure below.



Serial server configuration parameter description

Item	Instruction
Serial number	Select the serial port to be configured and enable or disable it.
Turn on heavy traffic transmission	Depending on the amount of transferred data and connection mode, choose to enable or disable it.
Serial parameter	
Serial working mode	Support RS232/RS485 mode, you can only choose one of the two.
Baud rate	The baud rate of the serial communication, the unit is bps, the options

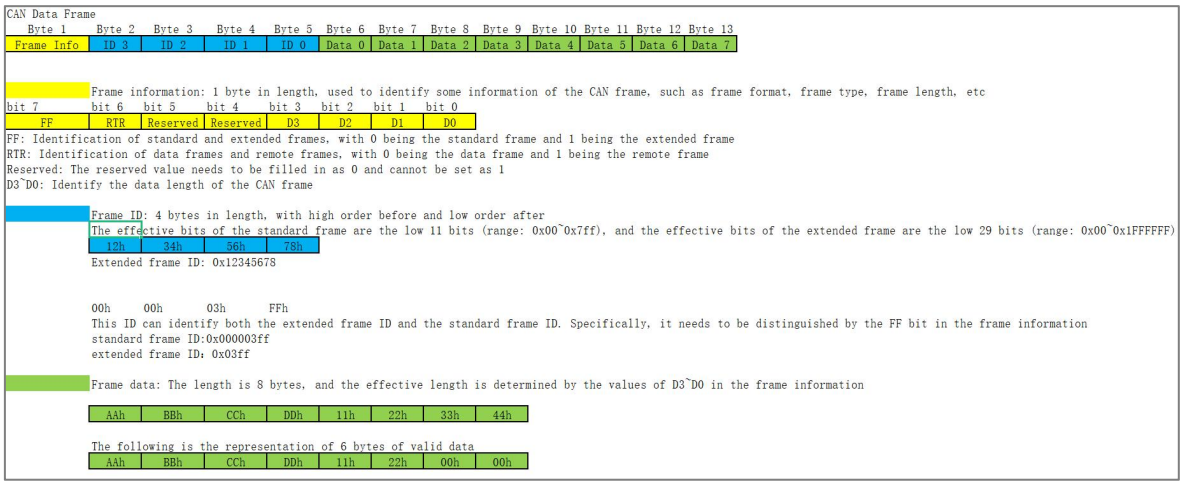
	are: 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400 and 460800. The factory default value is 9600. The baud rate of each serial port is independent of each other and can be set separately without affecting each other.
Check Digit	Select the check method, there are three options: none, odd check, and even check. The factory default setting is None. The verification method of each serial port is also independent of each other, which can be set separately without affecting each other.
Data bit	Set the number of effective data bits in serial communication, this machine supports 7 and 8 data bits.
Stop bit	When setting the stop bit length of serial communication, you can choose: 1, 2. The factory default setting is 1.
Packing length	When the length of data continuously received by the serial port exceeds the value set by the packet length, a transmission operation will be triggered, and the data will be forwarded to the network port, with a value range of 0~1460 bytes
packing interval	When the intermittent time of receiving data by the serial port exceeds this value, no matter how much data has been received, a transmission operation will be triggered, and the data will be forwarded to the network port, ranging from 0~255ms.
Frame header and frame trailer mode	After enabling this mode, the serial port will divide packets according to the start byte and end byte of the frame, and the data not between the head and tail will be discarded.
Start byte	Set the range of the starting byte of the serial port as the hexadecimal number 0x00~0xff
End byte	Set the range of the end byte of the serial port as the hexadecimal number 0x00~0xff
Registry package function	Select the sending method of the registration package; Disable: disable this function; LINK: The registration packet is only sent once when the network connection is established; DATA: The registration package is filled in front of the serial port data each time the serial port sends data to the network; This function is only allowed when the network working mode is UDP and TCP Client.
Registry Package Contents	The content of the registration package, the maximum length is 64 bytes. Users can choose to display this content in hexadecimal or ASCII.
Heartbeat function	Select the sending method of the heartbeat packet; Disable: disable this function; to COM: send the heartbeat packet to the serial port; to Network: send the heartbeat packet to the network port; This function is only allowed when the network working mode is UDP, TCP Server and TCP Client.

Heartbeat packet content	The content of the heartbeat packet, up to 64 bytes. Users can choose to display this content in hexadecimal or ASCII
Heartbeat packet interval	The time interval for sending heartbeat packets, in seconds, the range is 1~65535 seconds, and the default is 30 seconds.
RFC2217 function	Enabling this function allows users to use standard RFC2217 instructions on the network side to dynamically modify parameters such as the baud rate, data bits, stop bits, and parity bits of the serial port. This function is only allowed when the network working mode is TCP Server and TCP Client.
Network parameters	
Working mode	UDP, UDP Multicast, TCP Client, TCP Server, Modbus RTU Master, Modbus RTU Slave, Modbus ASCII Master, Modbus ASCII Slave, Pair Connection Master, Pair ConnectionSlave, Httpd Client, WebSocket Client and other communication methods are available. The factory default setting is UDP mode.
	When device is in UDP mode, the remote device must also work in UDP mode. This device can establish a UDP connection with aremote device, and the IP address and port number of the remote device can be configured on the page.
	When the device is in UDP Multicast, the remote device must also work in UDP Multicast mode. This device can communicate with devices in the same multicast address, and the multicast address and port number of the remote device can be configured on the page.
	Whenthe device is in TCP Client mode, the remote device must work in TCP Server mode, and its IP address and port number mustbe configured, which can be configured in the corresponding options of the network connec tion. The local port number can be ignored and noconfiguration is required
	When the device is in TCP Server mode, the remote device must work in TCP Client mode. In this mode, up to 4 remote TCP Client connections are accepted.
	When the working mode of this device is Modbus RTU Master, if the Modbus Over TCP function is not enabled, the remote device must work in Modbus TCP Slave mode; other wise, if the Modbus Over TCP function is enabled, the remote device must work in Modbus RTU Slave mode. This mode supports up to 4 connections
	When the device is in Pair Connection Master mode, the remote device must be operated in Pair Connection Slave mode, and its IP address and port number must be configured. You can connect to the corresponding options in the network In-progress configuration. The local port number can be ignored and no configuration is required.
	When the device is in Pair Connection Slave, the remote device must work in TCP Client mode. In this mode, up to 4 Pair Connection Master connections are accepted.

	When the device isin Relay Mode, CAN1 and CAN2 can transmit data to each other, at this time, the restart function of the device without data will be invalid. (Only MW-CANET200 supports this mode)
	When the device is in CAN to Serial mode, the device at the opposite end is required to transmit data according to the configured protocol. This mode can convert the CAN bus data to the serial bus according to the established protocol. (Only MW-CANET300 supports this mode)
	When the working mode of this device is Modbus TCP Slave, if the Modbus RTU Over function is not enabled, the remote master device must use the Modbus TCP protocol for communication; otherwise, enable the Modbus RTU Over function, The remote master device must use the Modbus RTU protocol to communicate. This mode supports up to 4 connections.
	When the device is in HTTPD Client, the user needs to specify the address, port, method and other parameters of the remote httpd server. The device will submit the data received by the CAN port to the httpd server in the form of GET or POST. At the same time, the data sent by the httpd server can also be sent to the CAN port.
	When the working mode of the device is WebSocket Client, the user needs to specify the main parameters such as the address, port and method of the WebSocket server. You can also set the time interval of Ping. Keep the connection between the device and the server. The device will upload the data received by the CAN port to the WebSocket server, and can also send the data sent by the server to the CAN port.
Locol port	Local port on the network connection side
Heartbeat interval	When the network working mode is in TCP mode, the CAN server will send a TCP heartbeat detection packet at a specified interval to test whether the connection exists, and if it does not exist, the connection will be automatically disconnected, ranging from 1 to 6000s.
Overtime time	When the network working mode is in TCP mode, the CAN server will detect the idle time of the current connection and the corresponding CAN port, and will disconnect the TCP connection when it exceeds the set value.
Modbus Over TCP	Modbus (RTU/ASCII) protocol transparent transmission enable
Network packet loss mechanism	Divided into overtime packet loss and redundant packet loss. • Timeout packet loss: If the set timeout period is exceeded, the extra unparsed bytes will be discarded, otherwise, it will be spliced with the next received data and parsed. • Extra packet loss: Every time the network data is received, the unparsed ones are discarded directly.
Modbus receive timeout	Modbus serial port receiving timeout
Modbus ID filtering	Filtering of Modbus slave ID ranges
Modbus Slave Read Ahead	The gateway automatically performs pre-reading according to the configuration in the Modbus pre-reading command table, and supports

	up to 8 items
Modbus slave polling time	The interval time when the gateway reads each item in the pre-read command table
Number of network connections	
Destination address	The IP address of the network connection peer
Destination port	The port number of the peer end of the network connection
Number of network connections in Modbus_RTU/ASCII_Master mode	
Destination address	The IP address of the network connection peer
Destination port	The port number of the peer end of the network connection
Modbus ID range	The data whose Modbus slave ID is within this range will be forwarded to the corresponding destination network address
Modbus_RTU/ASCII_Slave mode Modbus pre-read instruction table	
Device address	Modbus slave ID
Instruction	Modbus read data function code
Register address	The starting address of the slave register to be read
Number of registers	Number of slave registers to read

2.3.3 CAN data format description



Attention:

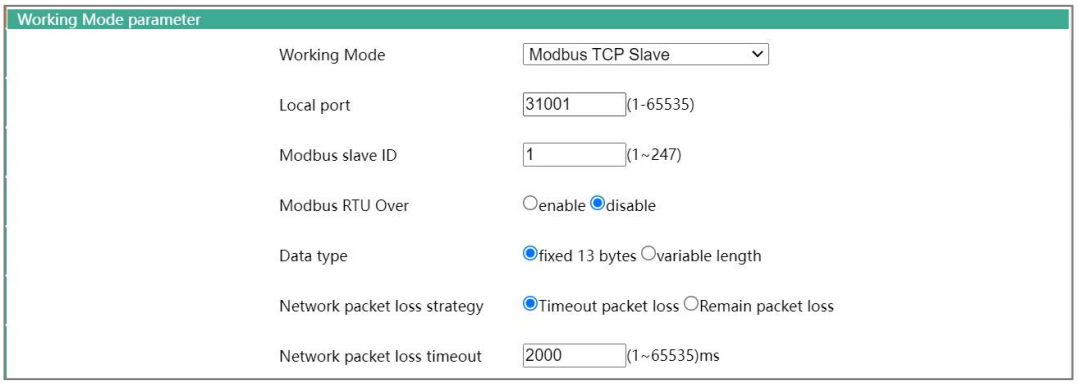
- The fixed 13 bytes in the web is transmitted according to the format shown in the figure above, and the data less than 8 bytes will be filled with 0.The variable length in the web is the opposite. When transmitting, it will be transmitted according to the actual data length, and will not be filled with 0.
- A TCP or UDP data packet is recommended not to exceed 50 frames of CAN data (that is, not to exceed 650 bytes).

2.3.4 Modbus TCP Slave function (CAN to Ethernet)

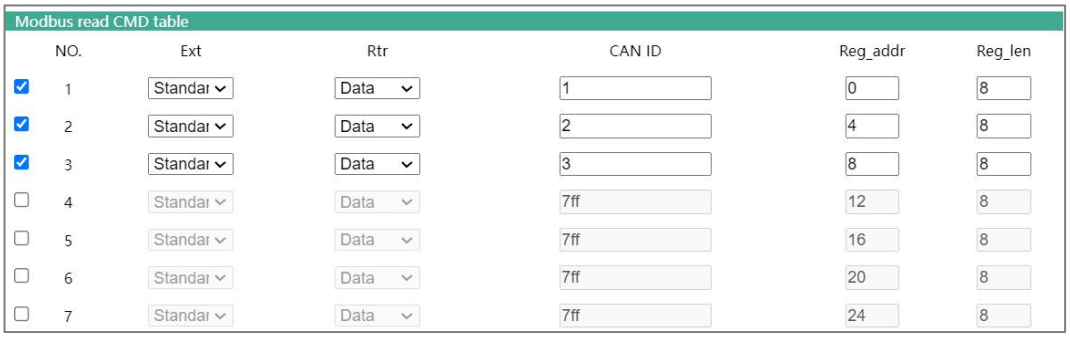
The working principle of Modbus to CAN: the CAN server acts as a Modbus slave, the network side communicates according to the Modbus protocol, and the CAN port side communicates according to the CAN bus protocol.There are two sets of register areas inside the CAN server, one is the read register area (register address range 0~127), the CAN server will store the data received by the CAN end in the read register area mapped with the Modbus read command table, When the side receives the corresponding Modbus read command, it will directly reply the data in this area to the network side, and the Modbus read command only supports 03 and 04 function codes.The other group is the write register area (register address range 256~383), the CAN server will store the data in the Modbus write command received by the network side in the write register area, and then the CAN server will follow the Modbus write command configured in the web The mapping relationship in the table is packaged into corresponding CAN commands and sent to the CAN port side. Modbus write commands only support function codes 06 and 16. Currently, the device can only support up to 32 sets of read/write commands.

Modbus to CAN example:

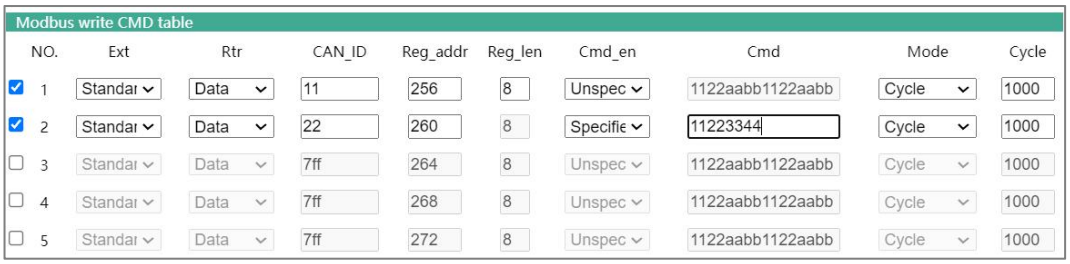
Step 1. As shown in the figure, configure CAN1 as the Modbus TCP Slave function, the local port is 31001, the slave ID is 1, and the network side communicates according to the Modbus TCP protocol.



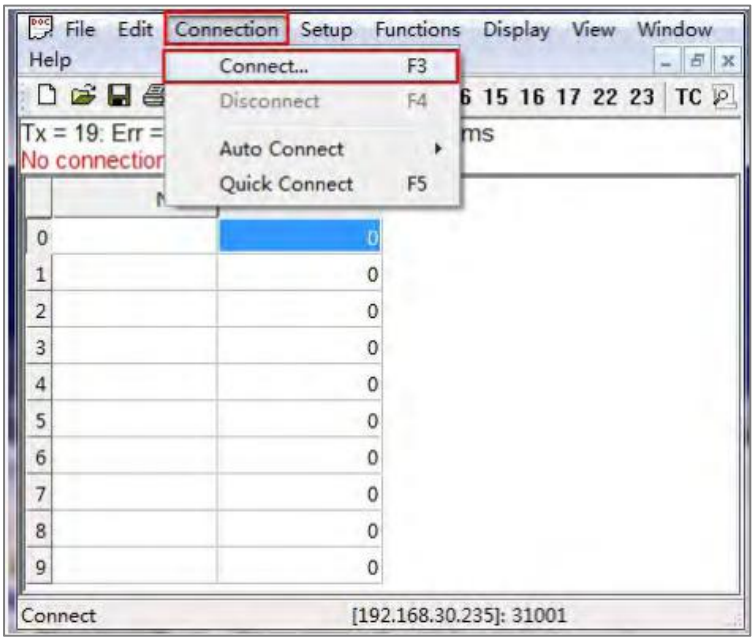
Step 2. Configure the Modbus read command table as shown in the figure. This table is the mapping relationship between the received CAN frame and the Modbus read register. The CAN data received by the device will be stored in the corresponding Modbus register. As shown above, the data in the CAN standard data frame with the CAN ID 1/2/3 will be stored in the Modbus register 0/4/8 register address respectively.



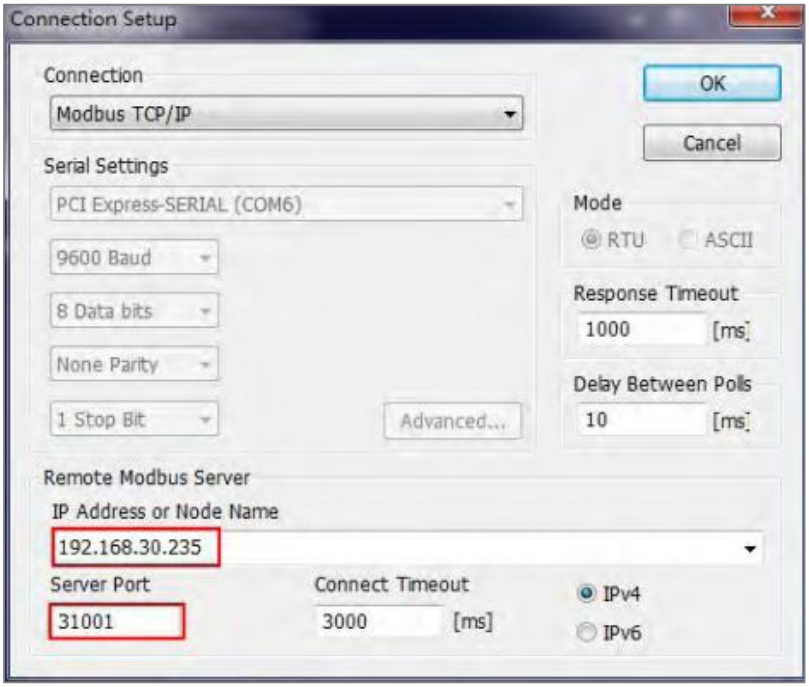
Step 3. Configure the Modbus write command table as shown in the figure. This table is the mapping relationship between the Modbus write register and the sent CAN frame. The Modbus write command received by the device will first be stored in the write command area, and then follow the Modbus write command table. Configuration, packaged into corresponding CAN frames, and sent to the CAN side. The role of the Modbus write command table is that if the CAN device does not actively send CAN data, it can read the data of the CAN device by sending a CAN frame requesting CAN data to the active device.



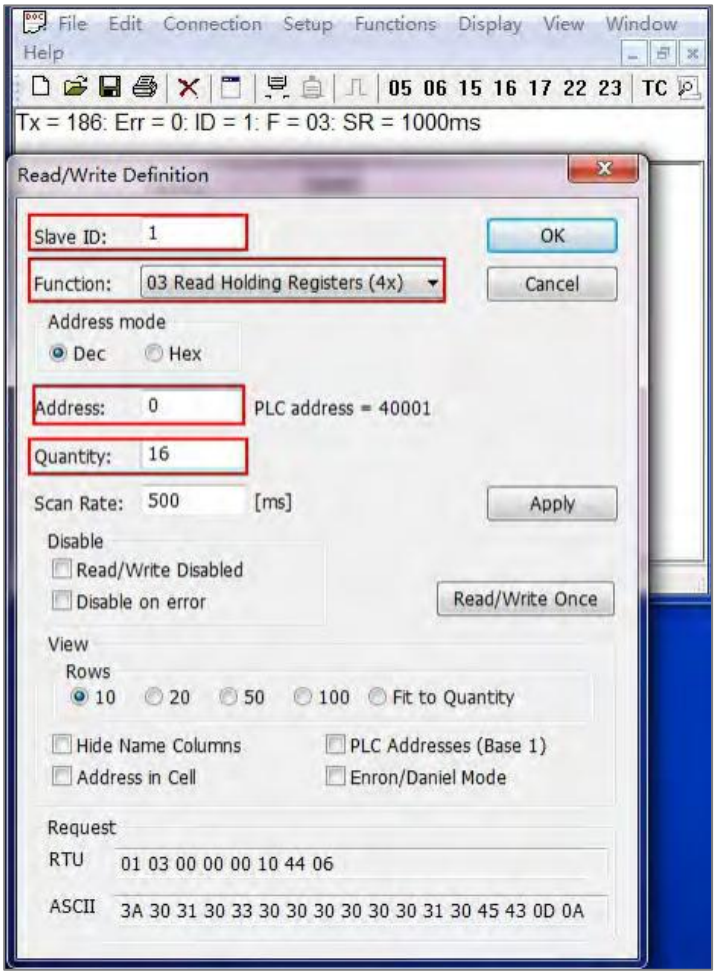
Step 4. Open a Modbus Poll software to test the data reading function. Configure the IP as the CAN server's IP address 192.168.30.230 as shown in the figure, and the port as 31001, and connect to the CAN1 port of the CAN server. As shown in the figure, configure the Slave ID in the Modbus poll software to be 1, the function code to be 03, the register address to be 0, and the number of registers to be 16. The CAN device actively sends standard data frames with CAN IDs of 1/2/3 to the CAN1 port of the CAN server. At this time, the Modbus poll software on the network side can read the corresponding value. At this point, the read command verification is complete.



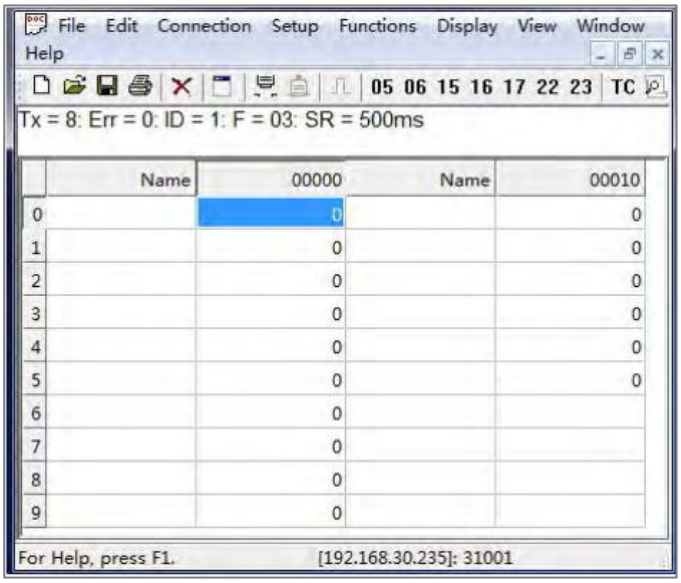
Enter the configuration page



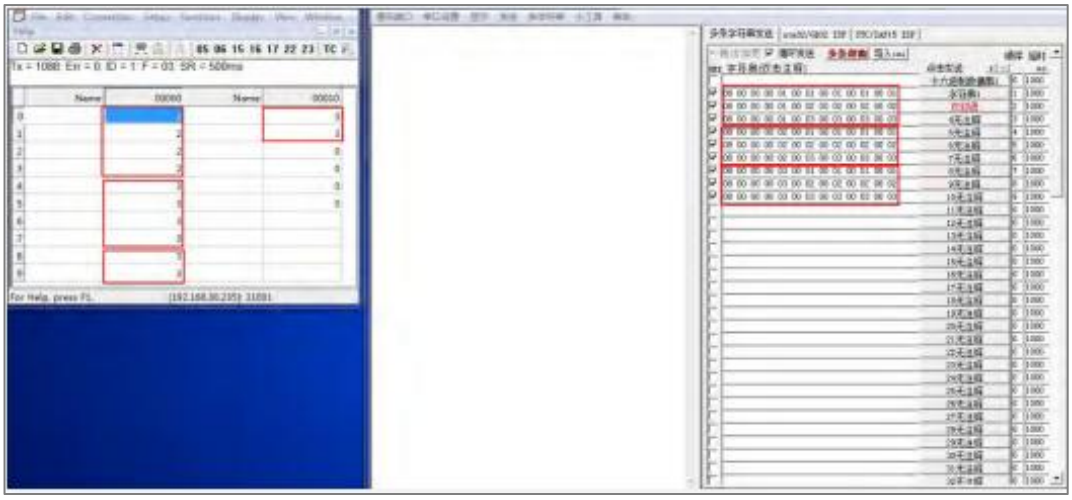
Configure network parameters



Set Modbus parameters

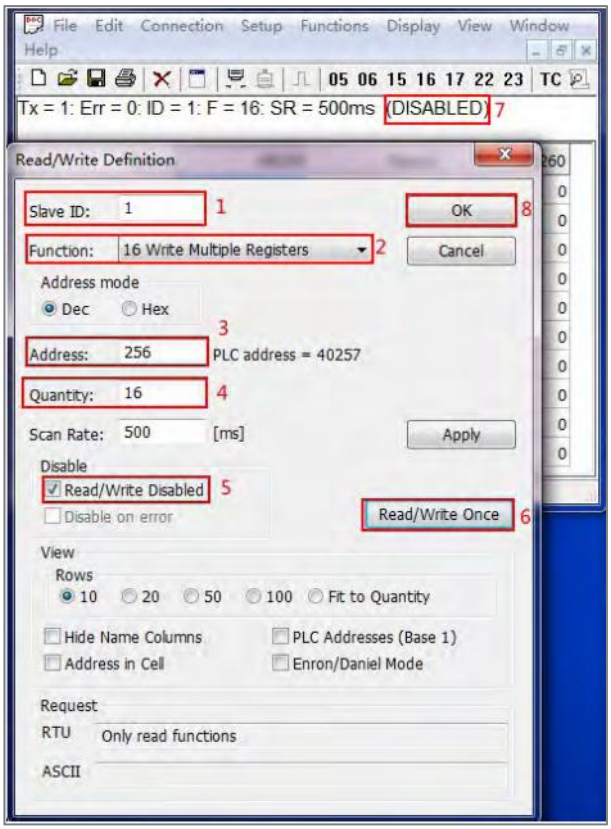


After the configuration is complete

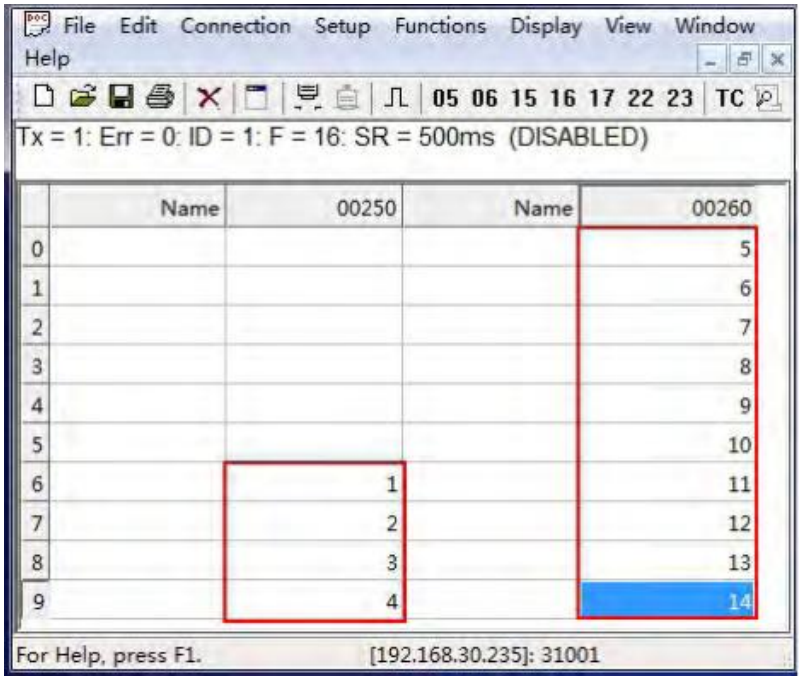


The device communicates normally

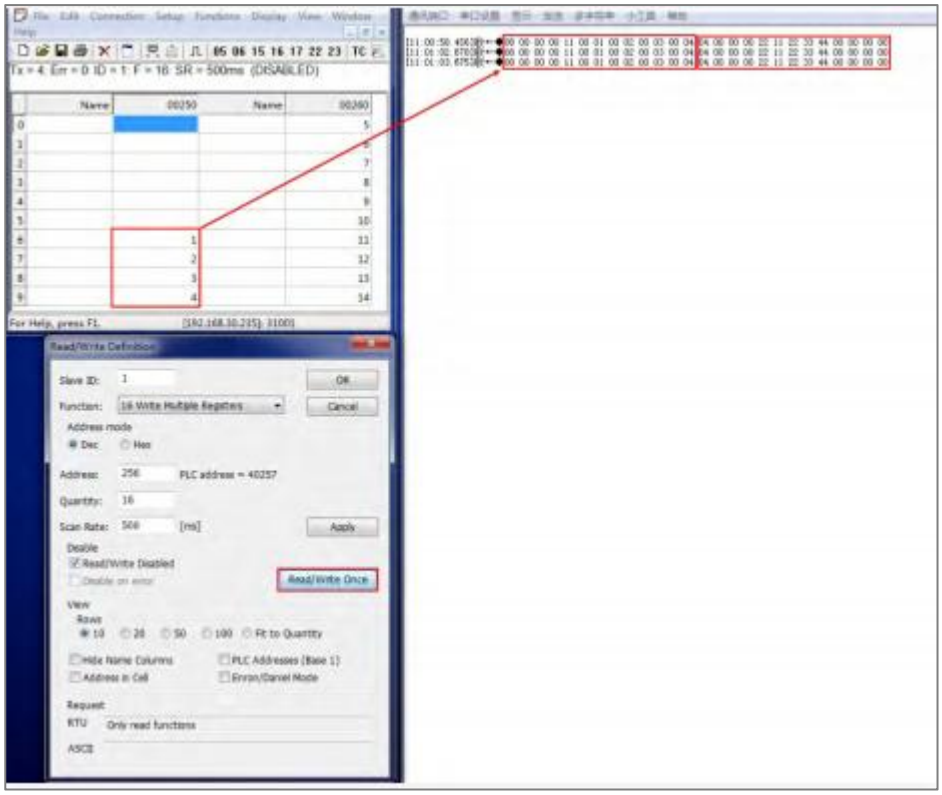
Step 5. Open a Modbus Poll software to test the data writing function. Configure the Slave ID in the Modbus poll software as shown in the figure to be 1, the function code to be 16, the register address to be 256, the number of registers to be 16, and the function of prohibiting automatic reading and writing, click Read/Write Once to take effect, click OK, the configuration is complete. Modify the data in the register as shown in the figure. As shown in the figure, click the Read/Write Once button to send the data in the register to the CAN server. After the CAN server receives the Modbus write data, it will send the corresponding CAN data to the CAN1 port. At this point, the function verification of the write command is completed.



Configure Modbus parameters

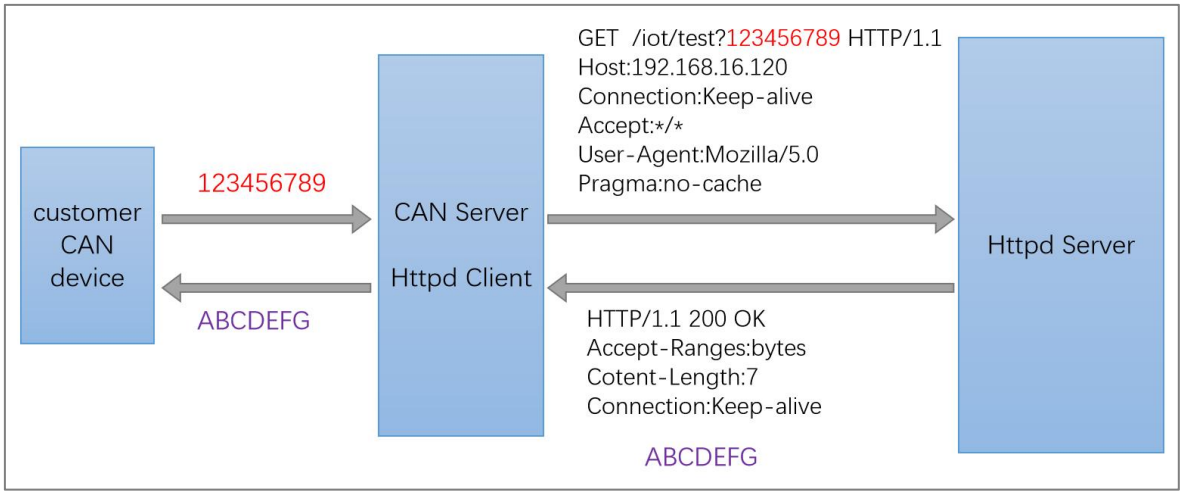


Set the register value to be written



Write register data

2.3.5 HTTPD Client function



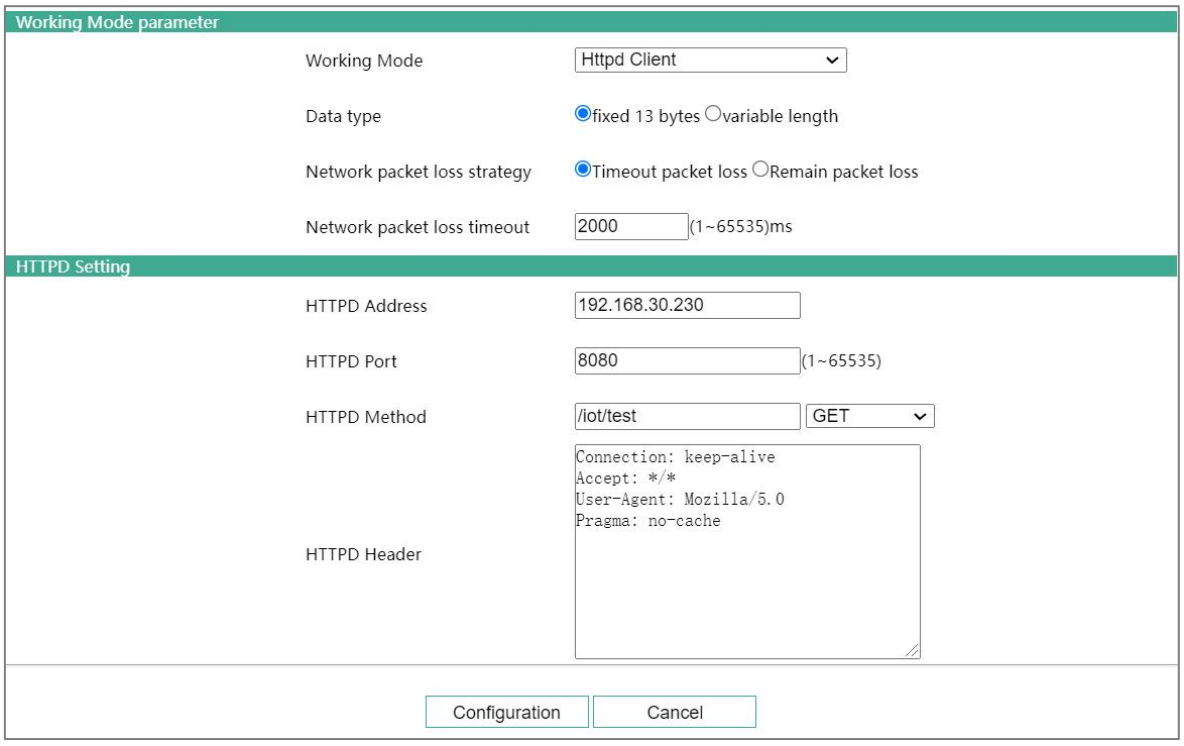
HTTPD Client Mode Introduction

This function is that the CAN server submits the data received by the CAN port to the HTTP server in the form of HTTP. If the HTTP server has data to send, the CAN server will transmit the HTTP body data to the CAN port.

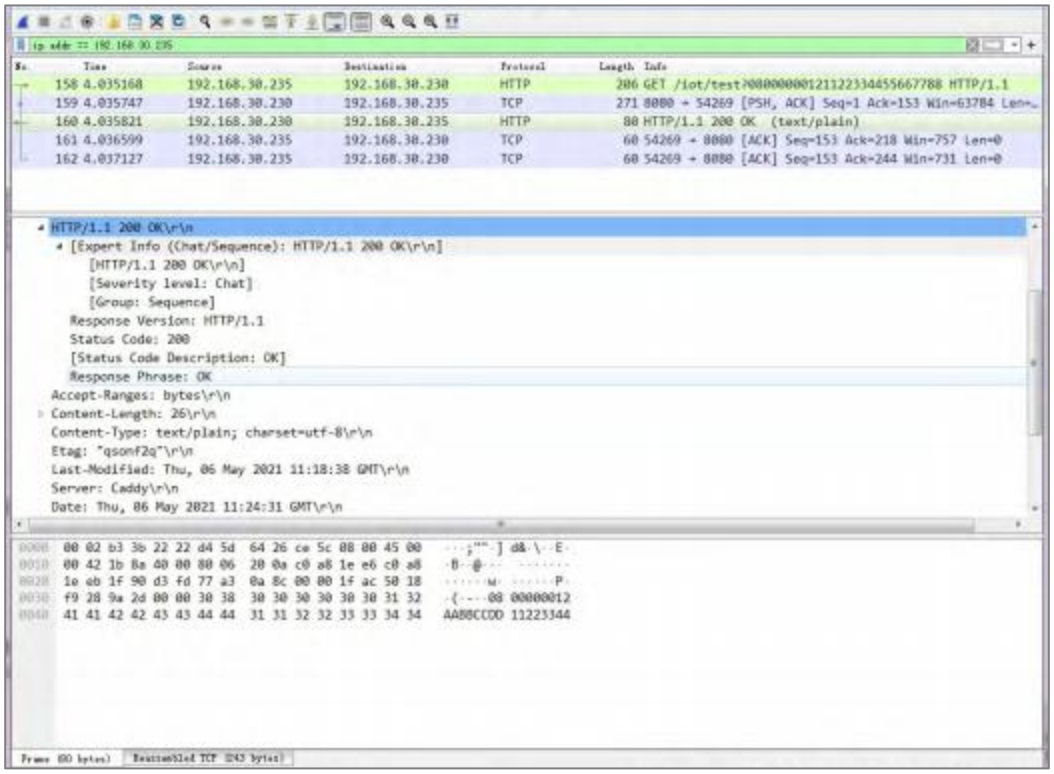
Note:
In the GET method, the CAN side transmits data in the HEX format of the CAN protocol. After passing through the CAN server, the network side transmits the data in the ascii format in hex.

Specific usage method:

1. Select "HTTPD Client" as the working mode.
2. Fill in the HTTPD address. The HTTP server address, which can be an IP address or a domain name (the ability to connect to the external network is required).
3. Fill in the HTTPD port number.
4. The HTTPD method needs to fill in the correct URL path, and select the GET or POST method as required.
5. The user fills in the HTTPD request header as required.
6. Finally click the configuration button to save the parameters.



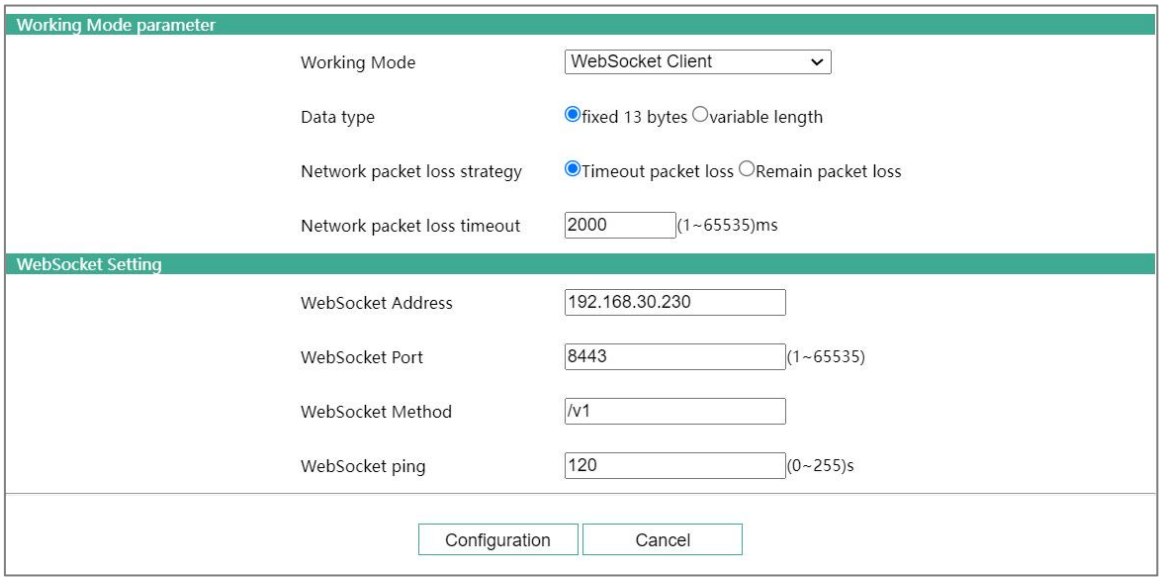
HTTPD Client Configuration Introduction



HTTPD Client communication Wireshark packet capture example

2.3.6 WebSocket Client function

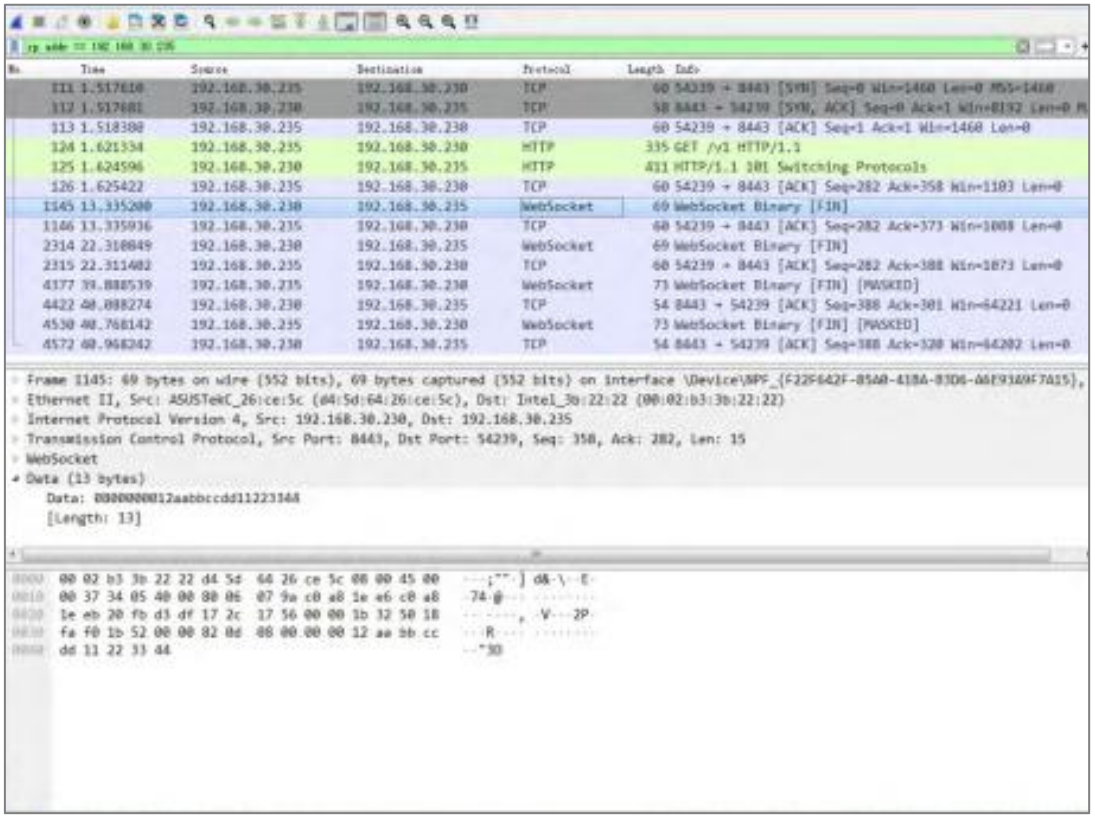
This function is the CAN server acts as a WebSocket Client, and transmits the data received by the CAN port to the WebSocket server in hexadecimal format, and the WebSocket server can also send data to the CAN port device at any time (the network side must follow the CAN protocol send data).



WebSocket Client Configuration Introduction

Specific using method:

1. Select "WebSocket Client" as the working mode;
2. Fill in the address of the WebSocket server, which can be an IP address or a domain name (the ability to connect to the external network is required);
3. Fill in the WebSocket server port number;
4. The WebSocket method needs to fill in the correct URL path;
5. The user can select the WebSocket Ping time interval according to the needs, and fill in 0 to indicate that the Ping function is not used;
6. Finally click the Configure button to save the parameters.



Example of capturing packets with Wireshark for WebSocket Client communication

2.3.7 Serial port Modbus function (serial port to Ethernet)

The following example uses software such as Modbus Poll to simulate the master, and software such as Modbus Slave to simulate the slave.

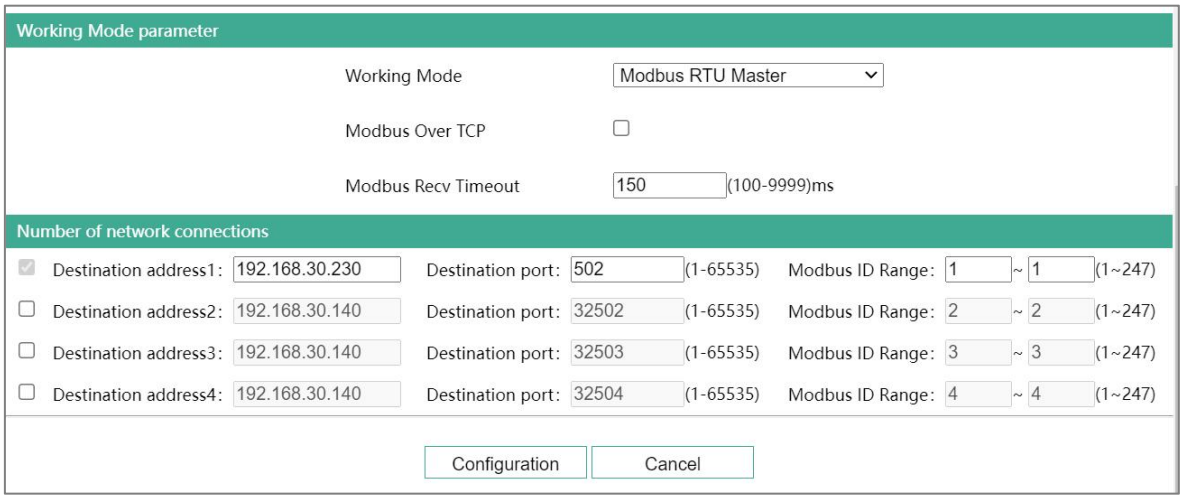
2.3.7.1 ModbusMaster

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

Configure the "serial port parameter" of the serial server to 9600-8-N-1, the working mode in"network parameters" is Modbus RTU Master, and the network address in "Network

Connections" is configured as the IP and port of the slave. The physical connection is described as follows:

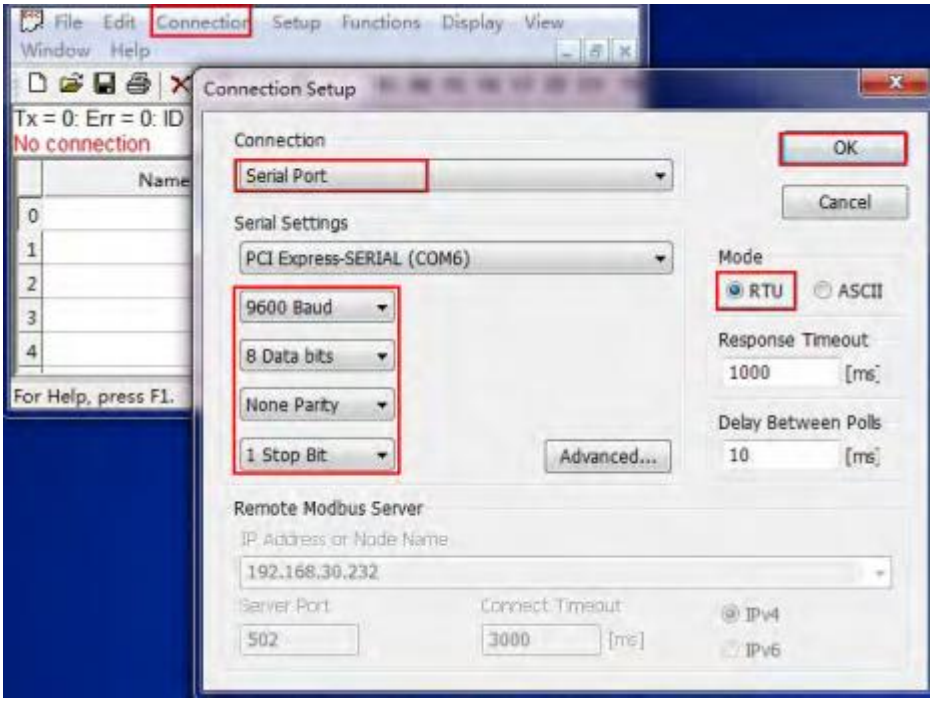
- Serial port: connect to the host
- Network port: connect to the slave



Modbus Web parameter configuration

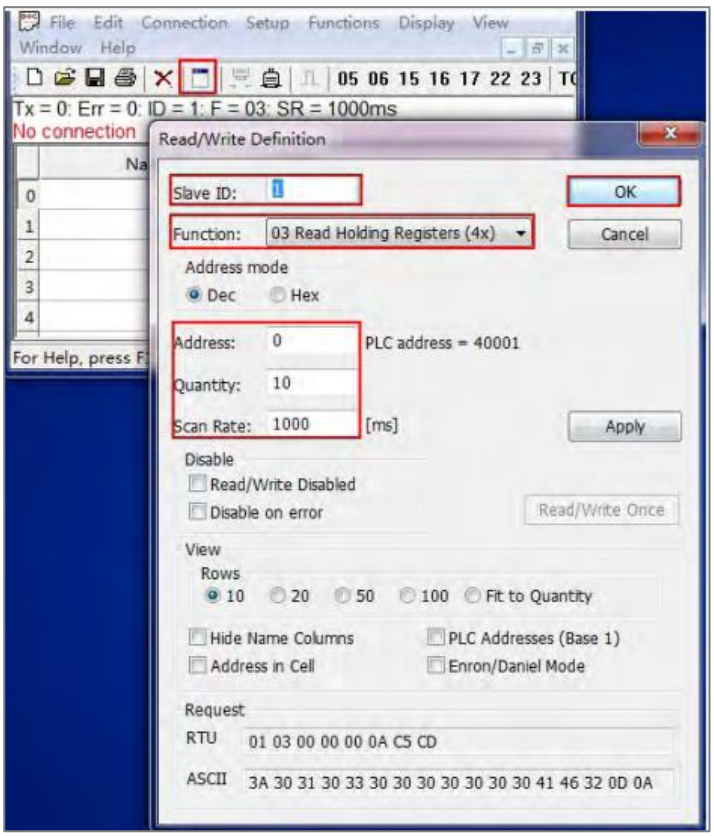
Modbus Poll software configuration:

Open the Modbus Poll software, go to "Connect" -> "Connect", and the connection parameters are configured as follows:



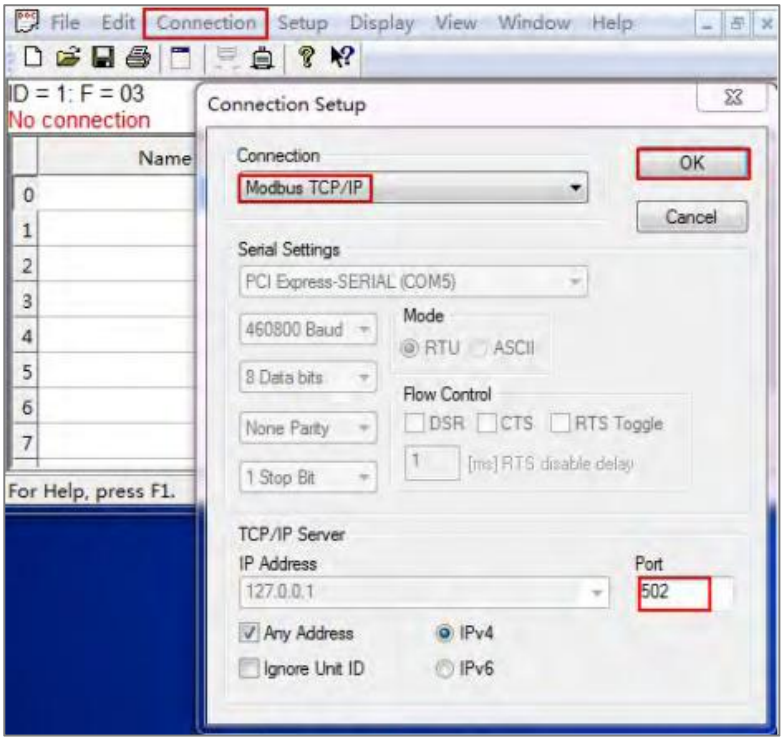
Modbus host serial port parameter configuration

Read 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.



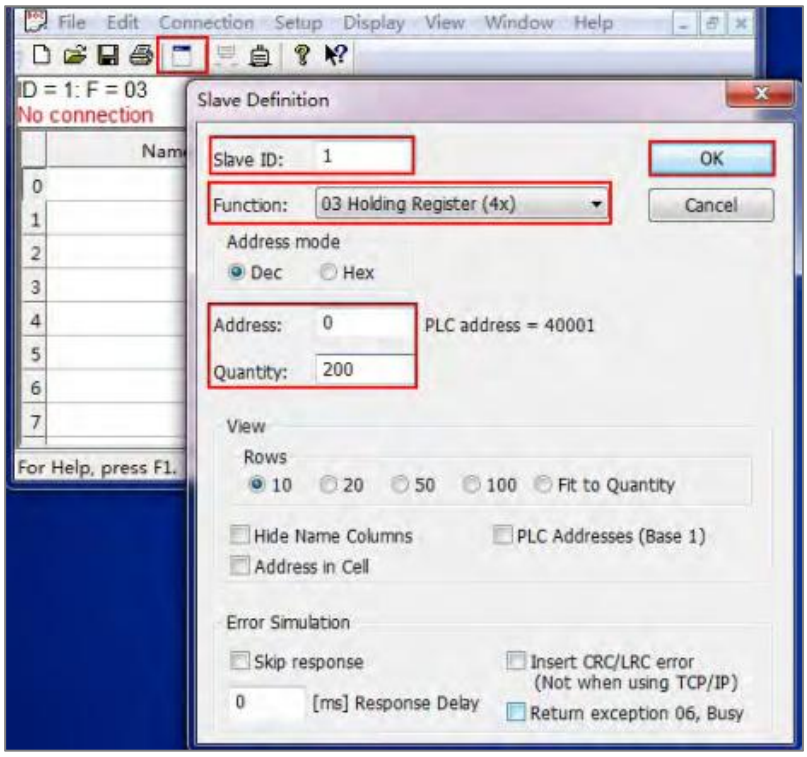
Modbus host device property definition

Open Modbus Slave software: Go to "Connect" -> "Connect", and the connection parameters are configured as follows:



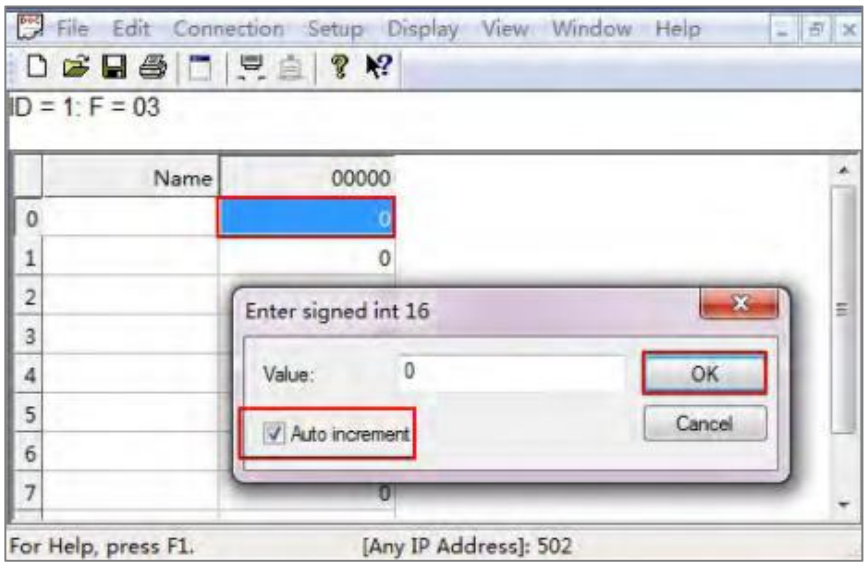
Modbus slave network connection configuration

Slave device definition configuration: the slave ID is 1, the function code is 03, the register start address is 0, and the total number of registers is 200

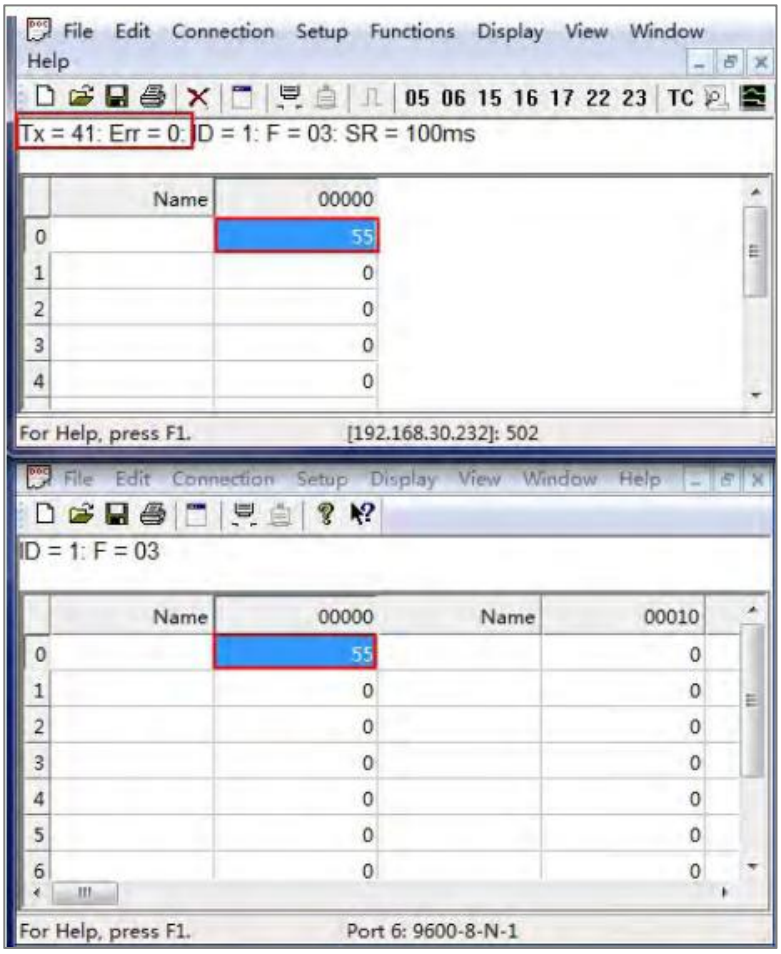


Modbus slave device property definition

Double-click the cell of Modbus Slave software and modify it to auto-increment mode, you can see that the register cell of Modbus Poll software also changes value automatically. Indicates that the device communication is normal.



Modbus slave register value auto-increment



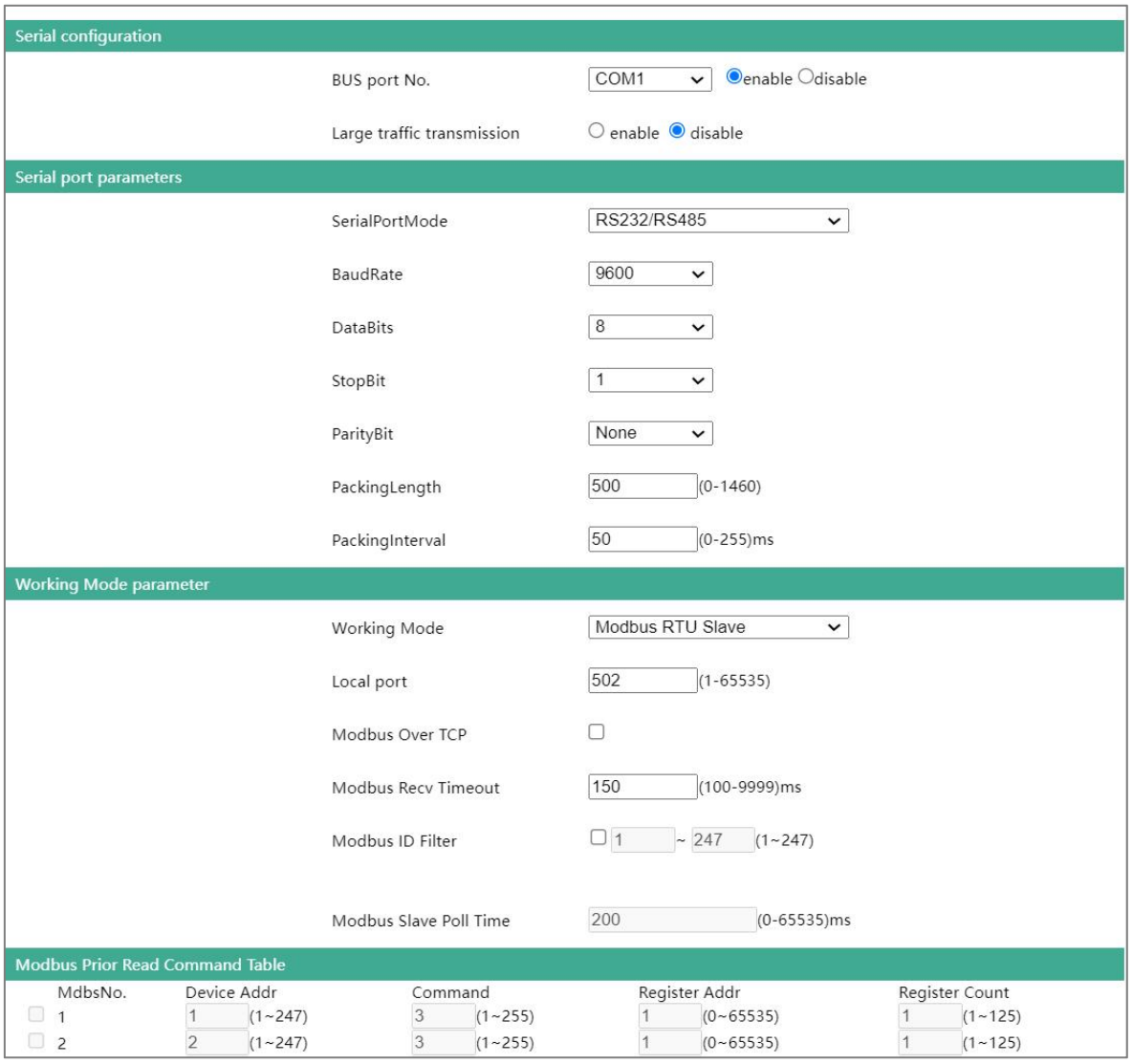
The communication is normal, and the host can read the register data of the slave through the serial port server device

2.3.7.2 Modbus Slave

Take Modbus RTU Slave as an example (the same applies to Modbus ASCII Slave):

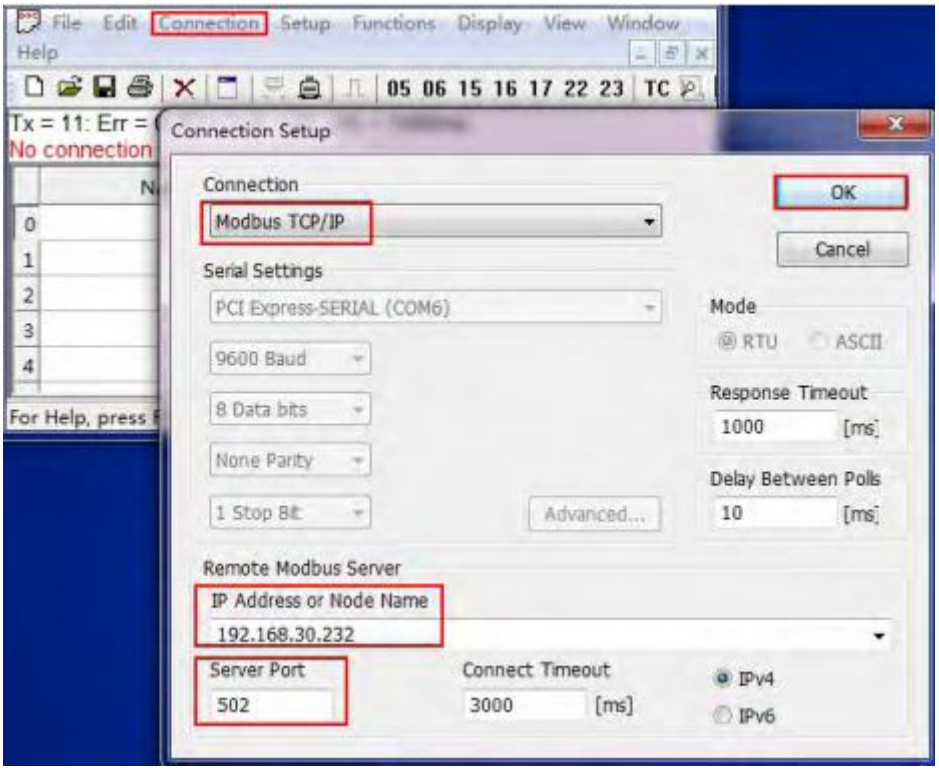
Configure the "serial port parameter" of the serial server to be 9600-8-N-1, the working mode in the "network parameter" is Modbus RTU Slave, and the local port is 502. The physical connection is described as follows:

- Network port: connect to the host
- Serial port: connect to the slave



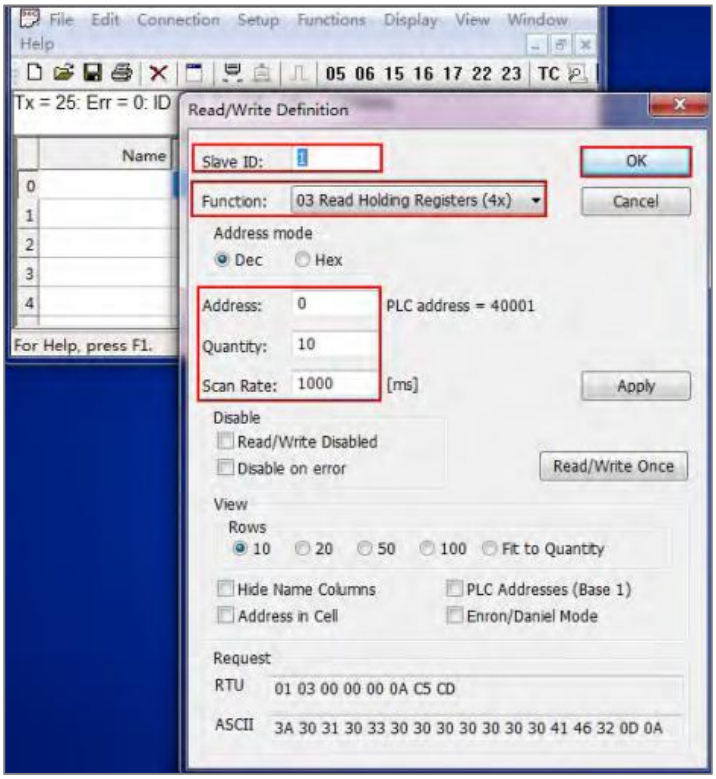
Modbus Web parameter configuration

Open Modbus Poll software: Go to "Connect" -> "Connect", and the connection parameters are configured as follows:



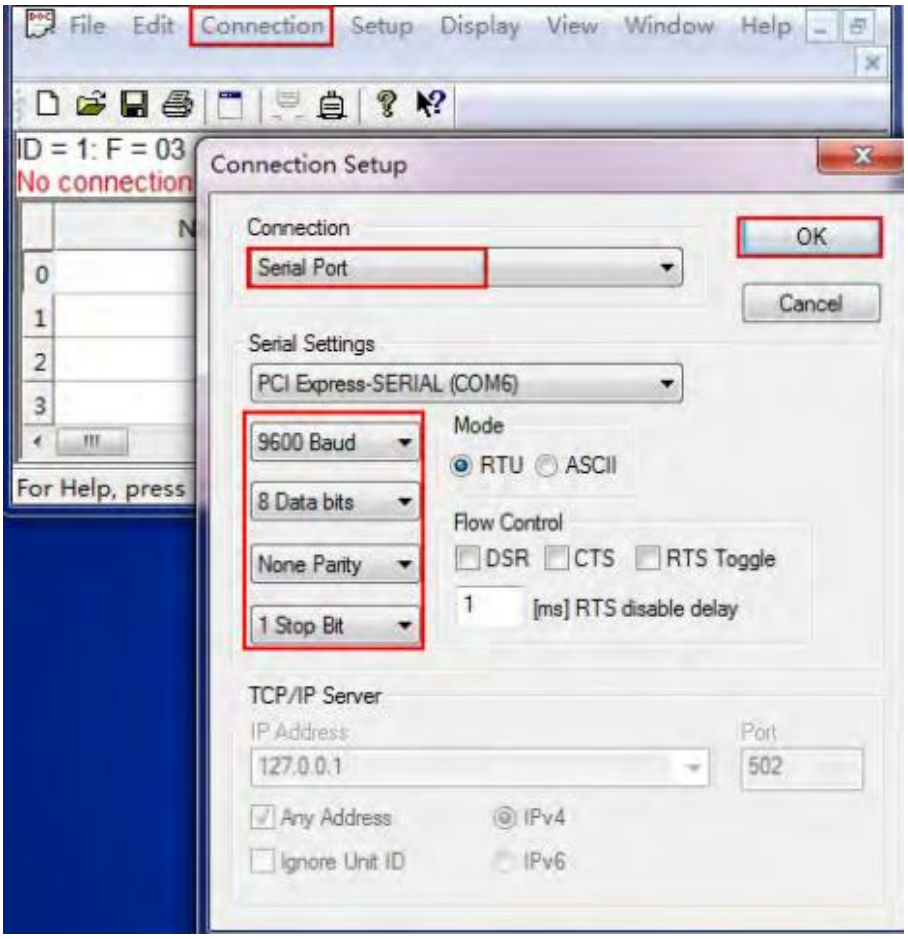
Modbus host network connection parameter configuration

Read 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.



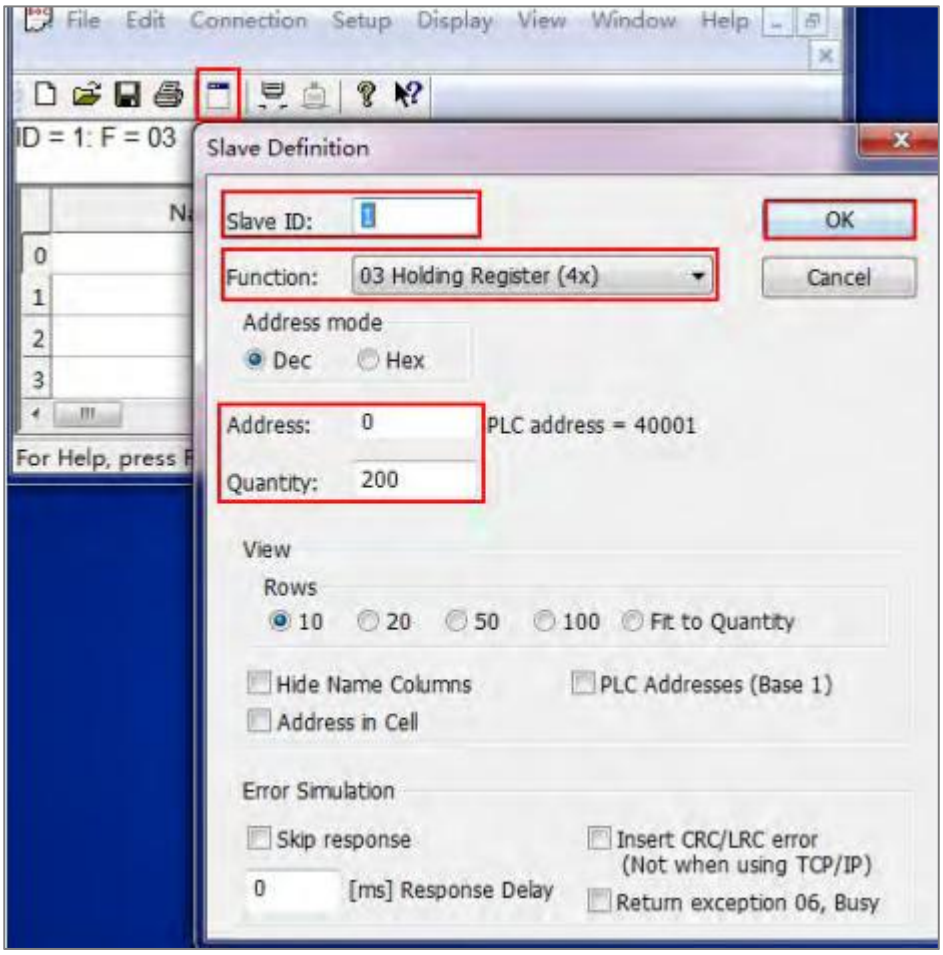
Modbus host device property definition

Open Modbus Slave software: Go to "Connect" -> "Connect", and the connection parameters are configured as follows:



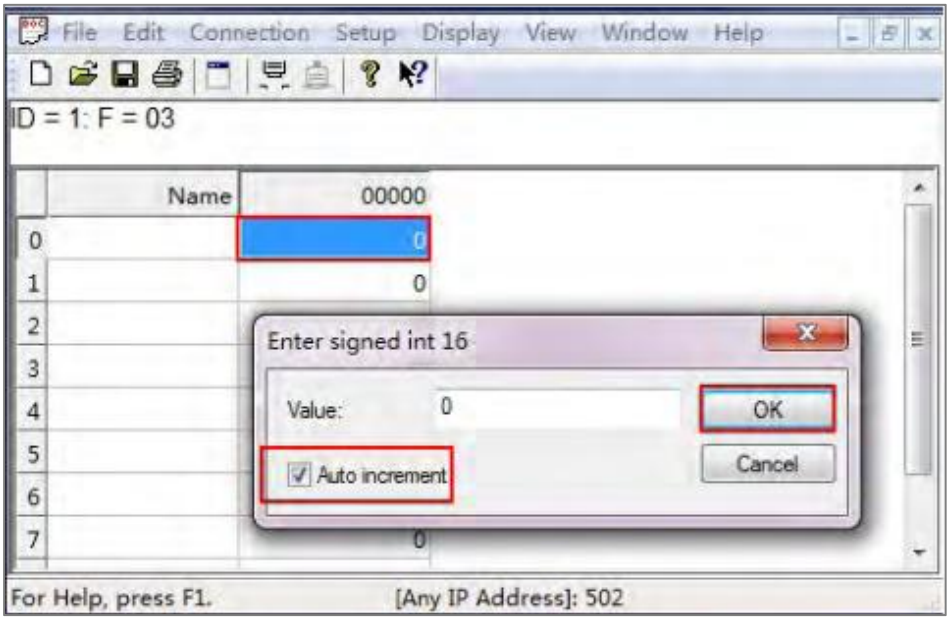
Modbus slave serial port parameter configuration

Slave device definition configuration: the slave ID is 1, the function code is 03, the register start address is 0, and the total number of registers is 200

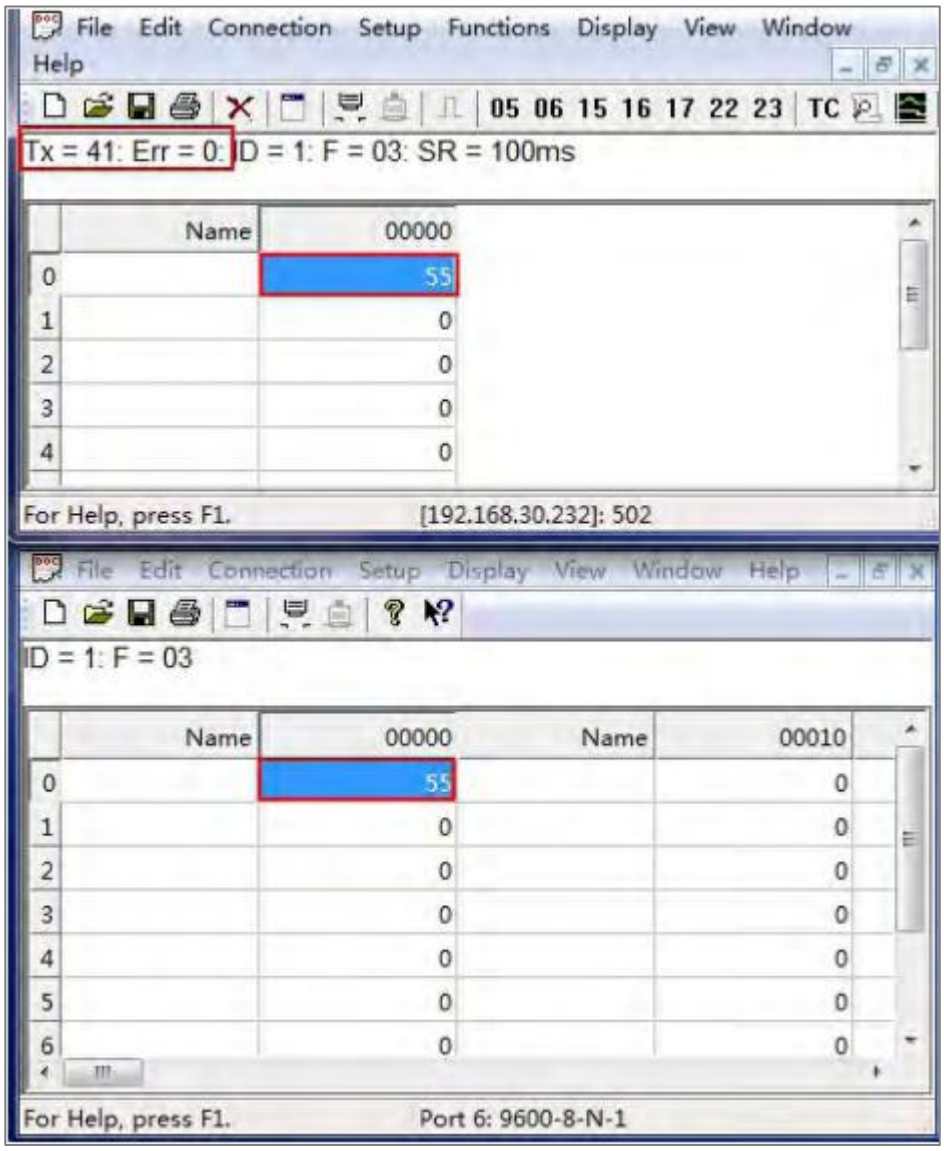


Modbus slave device property definition

Double-click the cell of Modbus Slave software and modify it to auto-increment mode, you can see that the register cell of Modbus Poll software also changes value automatically. Indicates that the device communication is normal.



Modbus slave register value auto-increment



The communication is normal, and the host can read the register data of the slave through the serial port server device.

2.3.8 CAN to serial port function.

MW-CANET300 provides four conversion modes for selection, including: transparent conversion, transparent conversion with tag, format conversion, and Modbus conversion. Parameters can be selected and set when configuring the CAN server.

- Transparent conversion:The meaning of "transparent conversion" is that the CAN server only converts the bus data in one format into the data format of another bus without adding or modifying the data. In this way, the exchange of data format is realized without changing the data content, and the CAN server is as transparent as the bus at both ends. This method will not increase the user's communication burden, but can convert the data as original, and can undertake the transmission of large-traffic data.

- Transparency with identification conversion: it is a special transparent conversion with no additional agreement. This conversion method is based on the common characteristics of common serial frames and CAN messages, so that these two different bus types can easily form the same communication network.
- Format conversion:"Format conversion" is the simplest mode of use. The data format is agreed to be 13 bytes, the fixed 13-byte serial frame data corresponds to a CAN message, and the 13-byte content includes CAN information+ ID + data. By correctly configuring frame information (the first byte of data), standard frames, extended frames, and even remote frames can be sent out flexibly. The details of standard frames, extended frames and even remote frames can be obtained by correctly parsing 13-byte serial frames.
- Modbus conversion:The meaning of "Modbus conversion" is to convert between UART data and CAN data of the Modbus protocol. Modbus protocol is a standard application layer protocol, which is widely used in various industrial control occasions. The protocol is open, has strong real-time performance, and has a good communication verification mechanism, which is very suitable for occasions that require high communication reliability.The CAN server uses the standard Modbus RTU protocol format on the serial port side, so the CAN server not only supports the user to use the Modbus RTU protocol, but the CAN server can also directly communicate with other devices that support the Modbus RTU protocol. device interface.

2.3.8.1 Transparent conversion.

In the transparent conversion mode, the CAN server immediately converts the data received from one side of the bus and sends it to the other side of the bus. In this way, the data flow is processed to maximize the speed of the CAN server and the utilization of the buffer.

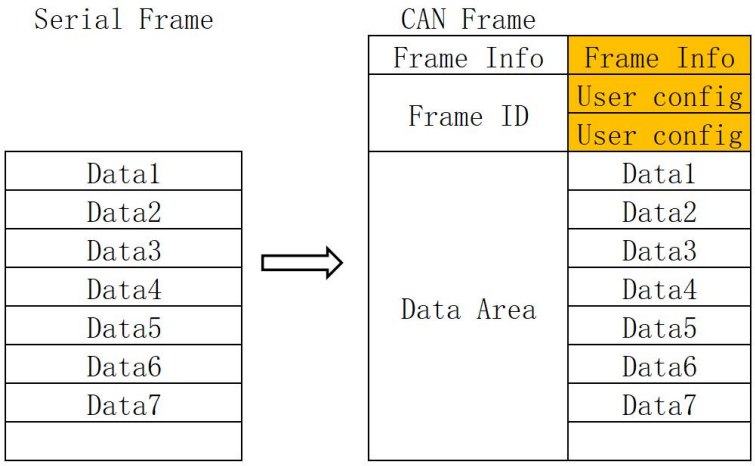
(1) Serial frame to CAN message

All the data of the serial frame is sequentially filled into the data field of the CAN message frame. The CAN server receives and converts data as soon as it detects that there is data on the serial bus.

The converted CAN message frame information (frame type part) and frame ID come from the user's prior configuration, and the frame type and frame ID remain unchanged during the conversion process. The corresponding format for data conversion is shown in the figure below.

If the line frame length of the received string is less than or equal to 8 bytes, fill characters 1 to n (n is the length of the serial frame) into the 1 to n byte positions of the data field of the CAN message in sequence (such as In Figure below, n is 7).

If the number of bytes in the serial frame is greater than 8, the processor starts from the first character of the serial frame and fills the first 8 characters into the data field of the CAN message in turn. After sending the data to the CAN bus, convert the remaining serial frame data and fill it into the data field of the CAN message until the data is converted.



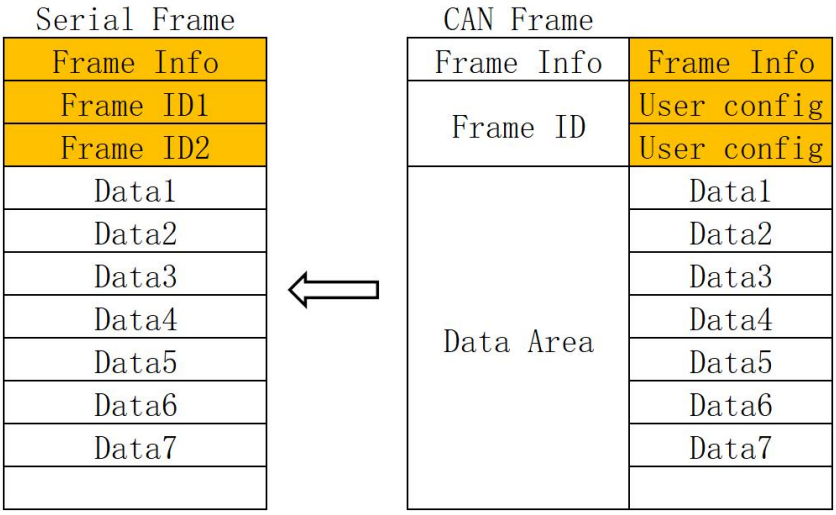
Serial frame to CAN message (transparent conversion)

(2) Convert CAN message to serial frame

For CAN bus messages, a frame is forwarded immediately after receiving the frame. The corresponding data format is shown in the figure.

When converting, all the data in the CAN message data field are converted into serial frames in sequence. If you select "Conversion" in the "Frame Information Conversion Enable" item during configuration, the CAN server will directly fill the "Frame Information" byte of the CAN message into the serial frame

If "Convert" is selected for "Frame ID Conversion Enable", then all the "Frame ID" bytes of the CAN message will be filled into the serial frame

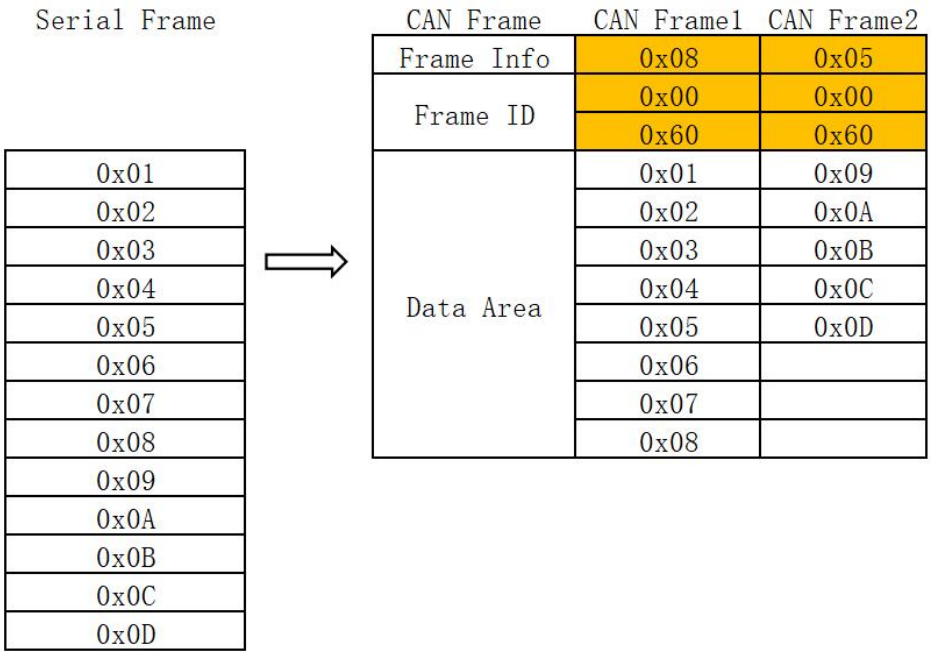


CAN message to serial frame (transparent conversion)

Conversion example:

(1) Serial frame to CAN message

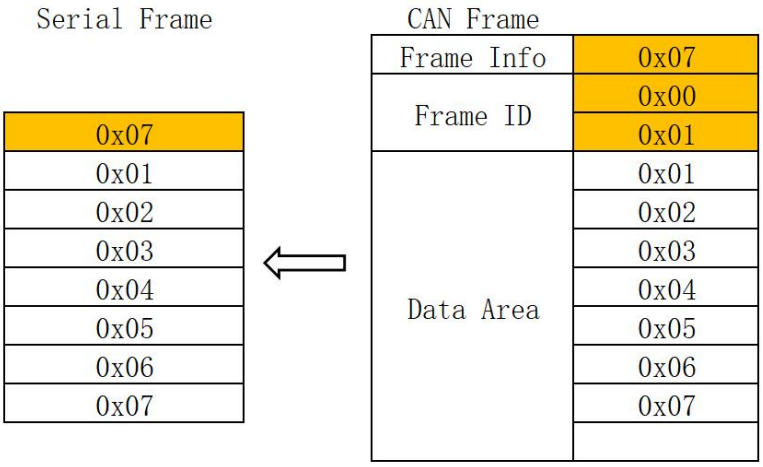
Assuming that the configured CAN message frame information is "standard frame" and the frame ID is 0x0060, the conversion format is shown in the figure.



Serial frame to CAN message (transparent conversion)

(2) Convert CAN message to serial frame

Assuming that the "frame information" conversion of the CAN message is configured, and the "frame ID" is not converted. CAN messages and converted serial frames are shown in the figure.



CAN message to serial frame (transparent conversion)

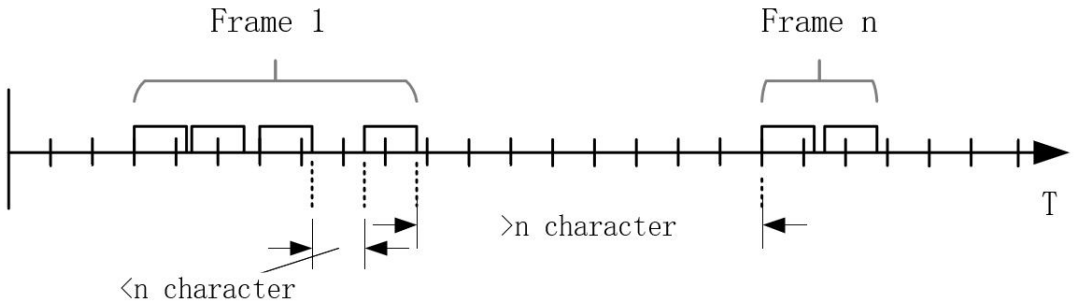
2.3.8.2 Transparency with identification conversion

1. Serial bus frame

When converting with a tag, a complete serial data frame must be obtained, and the CAN server uses the time interval between two frames as the frame division. The interval can be set by the user. The maximum length of the serial frame is the length of the buffer: 1500 bytes.The first

data detected by the CAN server in the idle state of the serial bus is used as the first character of the received frame.

During transmission, the time interval between characters in the frame must be less than or equal to the time for transmitting n characters (the value of n is configured in advance by the host computer) (the time for transmitting a character is divided by the number of bits contained in the character by the corresponding baud rate).If the CAN server does not receive a character within the transmission time of n characters or less after receiving a character, the CAN server considers the transmission of the frame to be over, and takes the character as the last character of the frame; n Characters after the character time do not belong to that frame, but to the content of the next frame. The frame format is shown in the figure.



Frame format

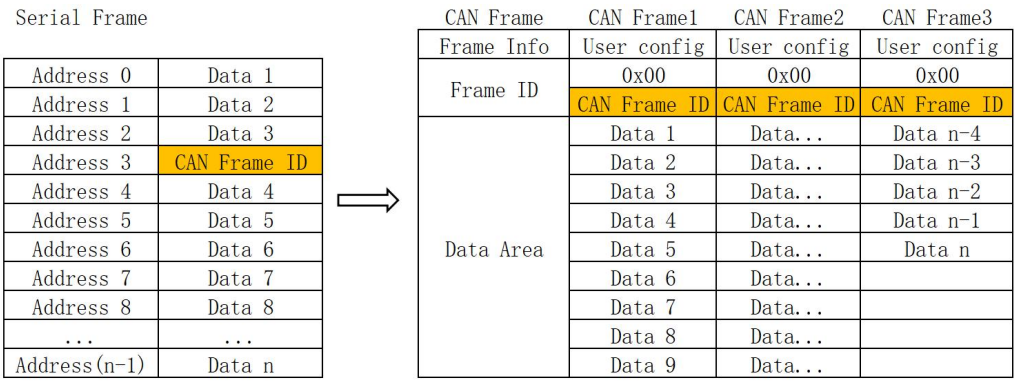
2.CAN bus frame

The format of the CAN message remains unchanged, but the corresponding CAN frame ID will also be converted into a serial frame.

(1) Serial frame to CAN message

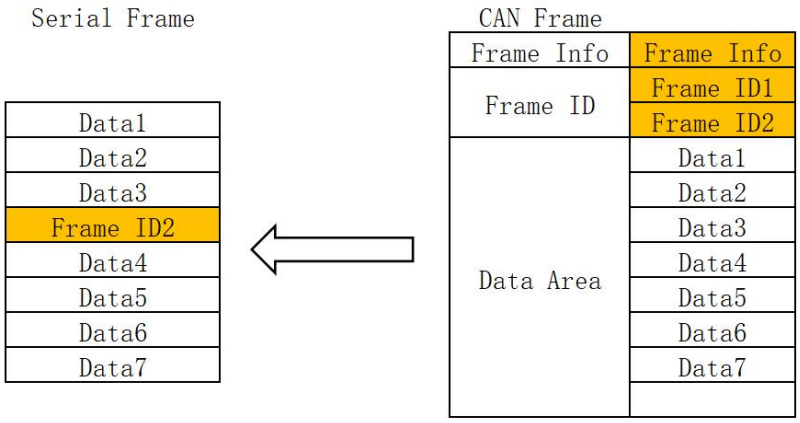
The starting address and length of the CAN with identification in the serial frame can be set by configuration. The range of start address is 0~7, and the range of length is 1~2 (standard frame) or 1~4 (extended frame).

During the conversion, all the CAN frame IDs in the serial frame are converted to the frame ID field of the CAN message according to the prior configuration (using the big-endian storage method, if the number of frame IDs carried is less than that of the CAN message frame ID number, then the low byte of the frame ID in the CAN message is filled with 0), and other data are converted sequentially, as shown in the figure.If a frame of CAN message has not converted the serial frame data, the same ID will still be used as the frame ID of the CAN message to continue converting until the conversion of the serial frame is completed.



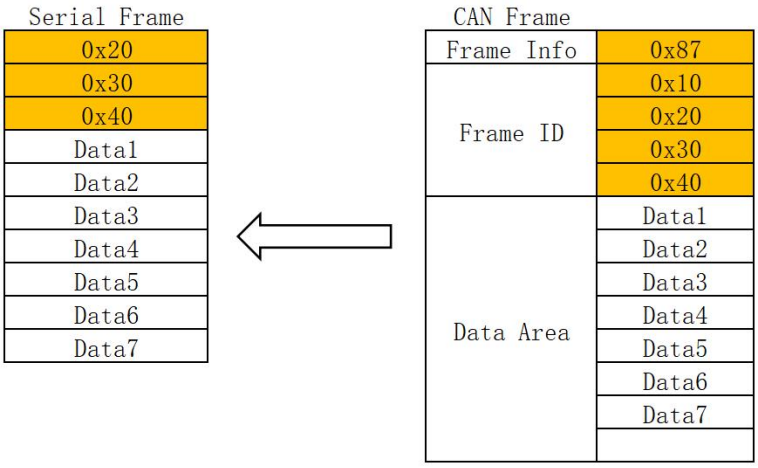
CAN message to serial frame (transparent conversion with identification)

(2) CAN message to serial frame



CAN message to serial frame (transparent conversion with identification)

Assuming that the start position of the configured CAN flag in the serial frame is 0, and the length is 3 (in the case of extended frames), the conversion of CAN messages and serial frames is shown in the figure.



Example of converting CAN message to serial frame (transparent conversion with identification)

2.3.8.3 Format conversion

As stated in the "CAN Data Format Description" above, each CAN frame contains 13 bytes, which contains CAN information + CANID + data.

In this mode, pay attention to strictly follow the 13-byte serial data format to convert successfully. First of all, ensure that the frame information is correct, the reserved bit must be 0, and the data length cannot be greater than 8, otherwise the conversion will not be performed.

Each frame is fixed at 13 bytes, if it is insufficient, it must be filled with 0. The serial data in the same serial data frame that meets the 13-byte format corresponds to a CAN message, and the serial data frame that is less than 13 bytes will not be converted. So make sure that the serial data frames to be converted are aligned in 13 bytes.

In the process of converting serial frames to CAN messages, if the data format of a certain segment of 13 bytes in the serial data frame aligned with 13 bytes is not standard, the 13 bytes will not be converted, and then converted the following data. If some CAN messages are missing after conversion, please check whether the 13-byte serial data format of the corresponding message conforms to the standard format.

2.3.8.4 Modbus mode (CAN to serial port)

Note:
The Modbus conversion function can only be used when the CAN bus device message can be edited.

Modbus protocol is a standard application layer protocol, which is widely used in various industrial control occasions. The protocol is open, has strong real-time performance, and has a good communication verification mechanism, which is very suitable for occasions that require high communication reliability. The CAN server uses the standard Modbus RTU protocol format on the serial port side, so the CAN server not only supports the user to use the Modbus RTU protocol, but the CAN server can also directly communicate with other devices that support the Modbus RTU protocol phase interface.

On the CAN side, a simple and easy-to-use segmented communication format is formulated to realize Modbus communication. The role played by the CAN server is still for protocol verification and forwarding, and supports the transmission of the Modbus protocol, rather than the master or slave of the Modbus, the user can communicate according to the Modbus protocol.

1 Serial bus frame

The serial interface adopts the standard Modbus RTU protocol, so the user frame can conform to this protocol. If the transmitted frame does not conform to the Modbus RTU format, the CAN server will discard the received frame without converting it. The Modbus RTU transmission format adopted by the CAN server is 1 start bit, 8 data bits and 1 stop bit. The maximum Modbus RTU frame length is buffer length: 2048 bytes.

2.CAN bus frame

If the devices on the CAN side want to use the Modbus protocol, a reliable transmission format needs to be defined for it. Here, a segmented protocol is used to implement it, which defines a message with a length greater than 8 bytes for Row segmentation and reassembly methods. The formulation of the segmented transfer protocol refers to the segmented message transfer protocol in DeviceNet. Segmented message format (take the extended frame as an example, the standard frame only has a different frame ID length, and the other formats are the same), the transmitted Modbus protocol content can start from the "data 2" byte, if the protocol content is large If it is less than 7 bytes, then continue to convert the rest of the protocol content according to this segmented format until the conversion finish.

The CAN bus frame format is described as follows:

CAN Extended Frame Format

Bit No.	7	6	5	4	3	2	1	0
Frame Info	FF	RTR	X	X	DLC			
Frame ID1	X	X	X	ID. 28~ID. 24				
Frame ID2	ID. 23~ID. 16							
Frame ID3	ID. 15~ID. 8							
Frame ID4	ID. 7~ID. 0							
Data1	markers	Segmentation type	Segmentation count					
Data2	Character 1							
Data3	Character 2							
Data4	Character 3							
Data5	Character 4							
Data6	Character 5							
Data7	Character 6							
Data8	Character 7							

CAN bus frame format (CAN to serial port Modbus)

- Segmented packet flag: Indicates whether the packet is a segmented packet. If this bit is 0, it means a separate packet, and it means it belongs to a frame in the segmented packet. (When the CAN message is a single frame, the value of the framing flag is 0x00).
- Segment Type: Indicates whether it is the first, middle, or last segment. Its definition is as shown in the figure:
- Segment counter: the mark of each segment, the serial number of the segment in the entire message, if it is the first segment, then the value of the counter is what. In this way, when receiving, it can be verified whether any segments have been lost.

Segment type location

Location	Meaning	Description
0	First segment	If the fragment counter contains a value of 0, then this is the first fragment in a fragmented message
1	Middle segment	Indicates that this is an middle segment
2	Final segment	Indicates that this is a finalsegment

Conversion method

In the process of converting from the serial port side to the CAN side, the CAN server will only convert when it receives a complete and correct Modbus RTU, otherwise it will take no action.

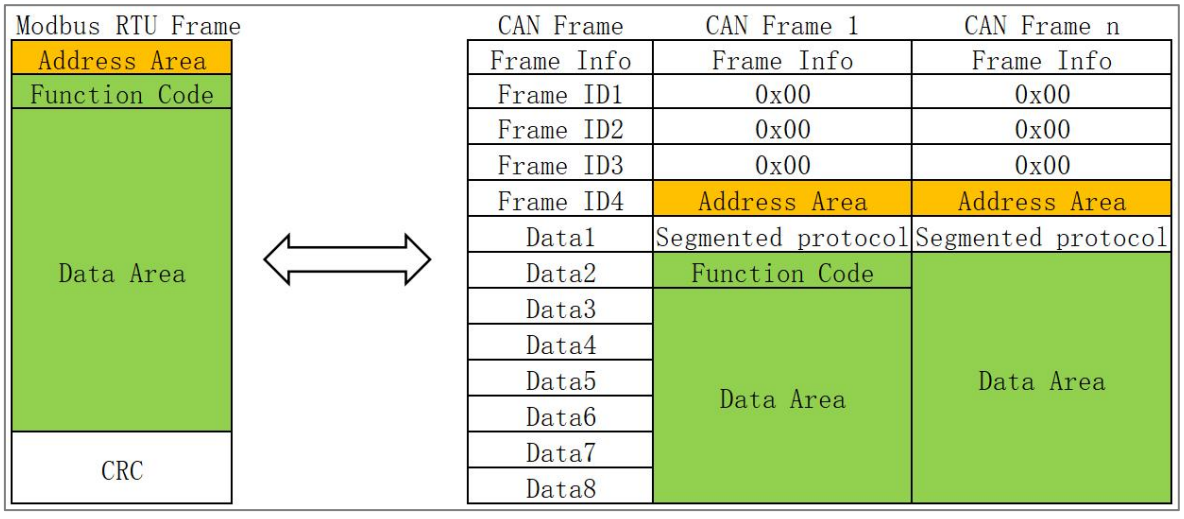
The address field of the Modbus RTU protocol is converted into ID4 (extended frame) and ID2 (standard frame) of the frame ID in the CAN message, and the identifier remains unchanged during the conversion of the frame. The CRC check byte is not converted into the CAN message, and the CAN message does not need to carry the check byte of the serial frame, because the CAN bus itself has a better check mechanism.

What is converted is the function code and address field of the Modbus RTU protocol, and they are converted in turn in the data field of the CAN message frame (starting from the second data byte, the first data byte is used by the segment protocol)), because the length of the Modbus RTU frame varies according to the function code.

A CAN message can only transmit 7 data, so the CAN server will segment the longer Modbus RTU frame into a CAN message and send it out using the above CAN segment protocol. When the user receives it on the CAN node, the function code and data field can be processed.

For the Modbus protocol data of the CAN bus, there is no need to perform a cyclic redundancy check (CRC16). The CAN server receives it according to the segmentation protocol. After receiving a frame and analyzing it, it automatically adds a cyclic redundancy check and converts it into a Modbus RTU frame. sent to the serial bus.

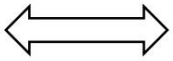
If the received data does not conform to the segmentation protocol, the group of data is discarded without conversion.



CAN to serial port Modbus

Conversion example:

Modbus RTU Frame	
Address Area	0x08
Function Code	0x11
Data Area	0x00
	0x01
	0x00
	0x02
	0x04
	0x00
	0x0A
	0x01
	0x02
CRC	0xED
	0x69



CAN Frame	CAN Frame 1	CAN Frame n
Frame Info	0x88	0x84
Frame ID1	0x00	0x00
Frame ID2	0x00	0x00
Frame ID3	0x00	0x00
Frame ID4	0x08	0x08
Data1	0x81	0xC2
Data2	0x11	0x0A
Data3	0x00	0x01
Data4	0x01	0x02
Data5	0x00	
Data6	0x02	
Data7	0x04	
Data8	0x00	

Example of CAN to serial port Modbus

2.3.8.5 Conversion direction

1. Bidirectional: The CAN server converts the data of the serial bus to the CAN bus, and also converts the data of the CAN bus to the serial bus.
2. Only serial port to CAN: Only convert the data of the serial bus to the CAN bus, but it will not convert the data of the CAN bus to the serial bus.
3. Only CAN to serial port: only convert the CAN bus data to the serial bus, but it will not convert the serial bus data to the CAN bus.

Attention:

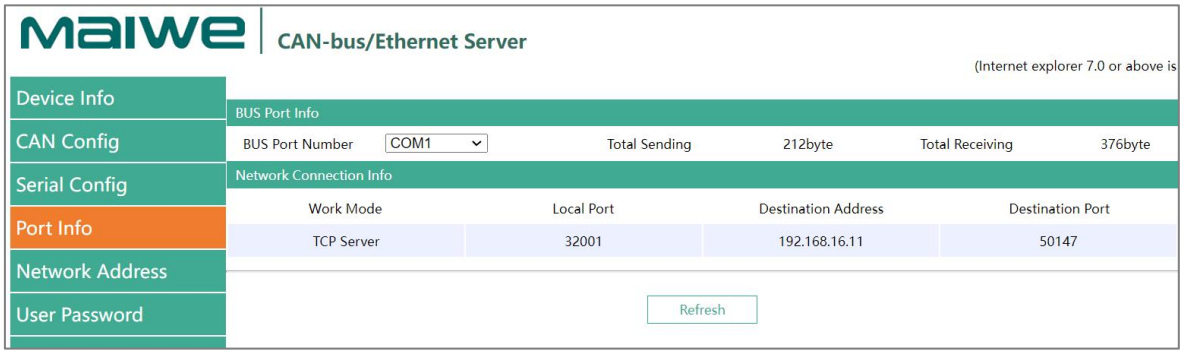
- The port numbers 80, 4500, 4800, 57050, 57051, 57850, and 57851 have already been used by the system. When configuring the network port numbers, please do not reuse them.
- This device and the peer CAN device must have the same baud rate.
- If the device is working in UDP mode, the remote device must also work in UDP mode; if the device is working in TCP Client mode, he remote device must work in TCP Server mode. If the device is working in TCP Server mode, the remote device must work in TCP Client mode.
- If the device works in UDP multicast mode, the same multicast address can only be used once, and a multicast address is not allowed to be used in different port configurations.
- When you need to use long-frame data frequently or have high requirements for data transmission, please adjust the baud rate and send interval appropriately to prevent garbled characters or packet loss caused by slow CAN port speed.
- When configuring the device, the user should ensure that the external CAN device stops sending data to the CAN server to avoid garbled characters.
- CAN to Serial mode, the baud rate of CAN and serial ports should match each other, this mode is not suitable for large and fast applications.

2.4 Port information

The port information includes two parts: bus port information and network connection information.

2.4.1 CAN port information

CAN port information page is used to display the current connection information of the CAN port as shown in the figure:

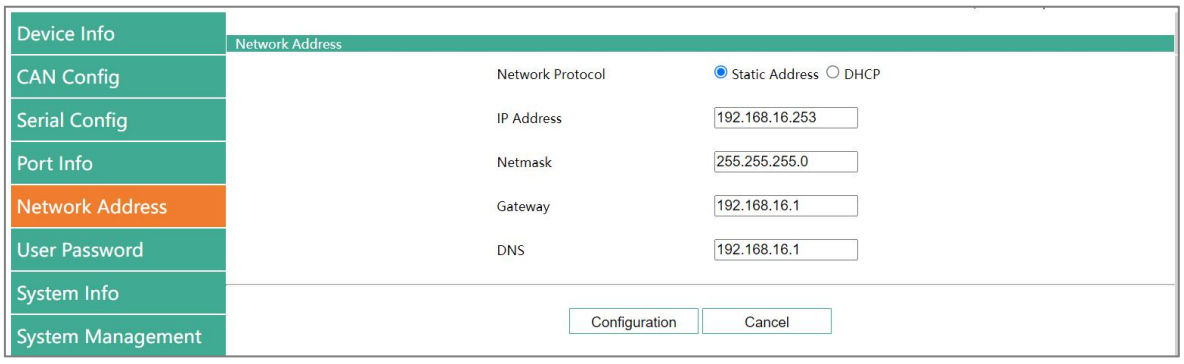


2.5 Network address

The network address includes: network IP address, subnet mask, default gateway

2.5.1 Network address

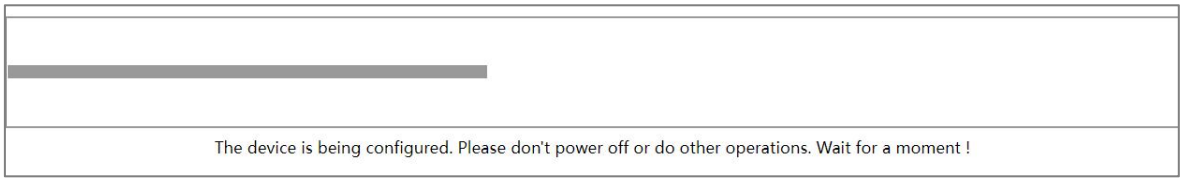
This function is to assign a specified IP address to the CAN server. The default IP address of the CAN server is 192.168.16.253. The network address configuration page is shown in the figure.



- Network protocol: Select Static Address or DHCP.
- IP address: An IP address is a 32-bit address assigned to a device connected to the Internet. An IP address consists of two fields: a network number field (net-id) and a host number field (host-id).

- Subnet mask: A mask is a 32-bit number corresponding to an IP address, some of which are 1 and others are 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 1 bit in the mask is the subnet address, and the other bits are the host address. The mask corresponding to a class A address is 255.0.0.0; the mask of a class B address is 255.255.0.0; the mask of a class C address is 255.255.255.0
- Default gateway: The default gateway in the host is usually called the default route.
- The default route (Default route) is the route selected by the router when no other route exists for the destination address in the IP packet.
- DNS: IP address of the DNS server. When the device uses a static IP address, the user needs to fill in this content. If no specific DNS server is used, generally fill in the default gateway IP address.

Whenever the user modifies the address setting, he needs to click the "Configure" button to submit it to the CAN server, and switch to a waiting page as shown in the figure.



When the progress bar on the screen is finished, the CAN server will restart the web server, and the user needs to log in again.

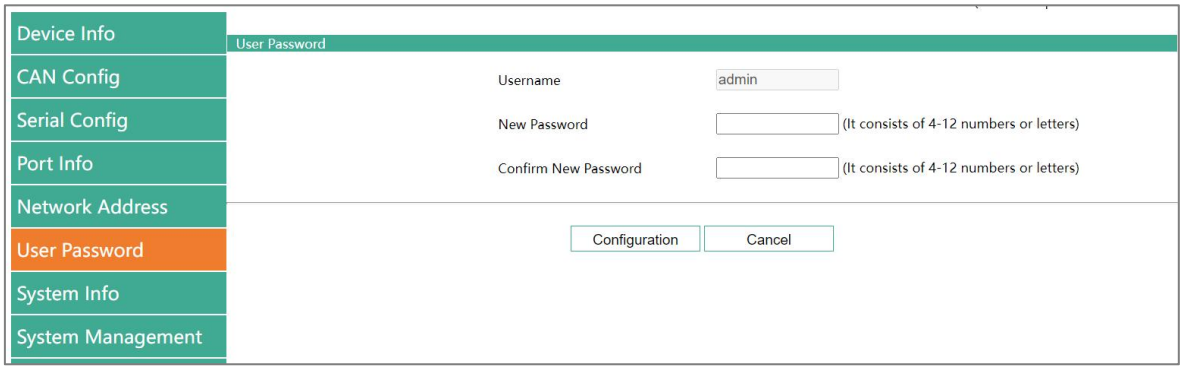
Attention:

- When entering the waiting page after configuring the IP, please do not power off or perform other operations to avoid failure to modify the IP address.
- The configured IP address and default gateway must be in the same network segment.
- If the device uses DHCP to obtain the IP address, after the device restarts, the user needs to use the company's network management assistant to search for the device to know the new IP address of the device accurately.

2.6 User password

CAN server web server provides users with three different permissions. The first is a visitor, which user can only view the current configuration of the CAN server, but cannot modify the configuration. The user name and password are both "none" and cannot be modified. The second one is an ordinary user, who can configure various functions of the CAN server parameters, the user name is admin, which cannot be modified, the initial password is "admin", which can be modified on this page. The third one is administrator, this account has the highest authority, and when the password of ordinary users is forgotten, You can use the administrator to log in and change the password of the machine. The user name is fixed as "admin", and the

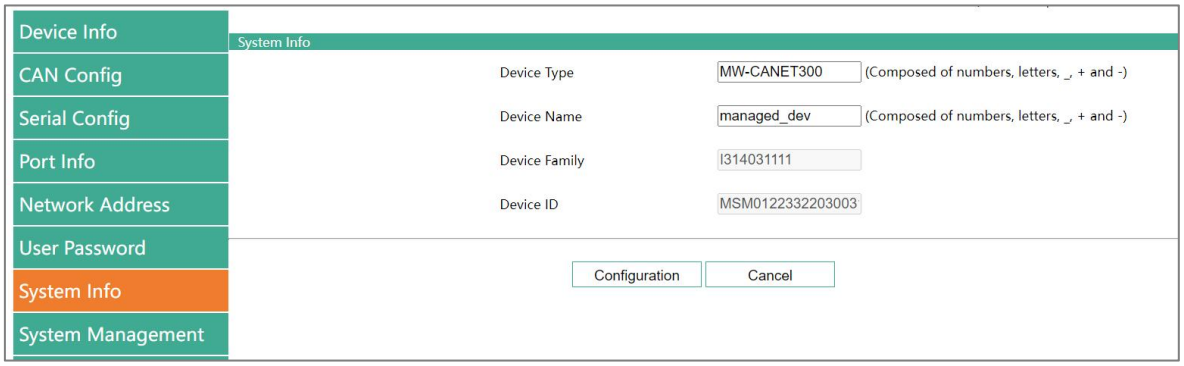
password is the last six digits of the machine's MAC address.The login password must be legal characters, consisting of 4-12 English letters (case-sensitive) and numbers. When changing the password, you need to enter it twice, and you must ensure that the passwords entered twice are the same. The page is shown in the figure.



- Username: The local username is fixed as "admin" and cannot be modified.
- New password: Set the user password for this group, which consists of 4-12 English letters (case-sensitive) and numbers.
- New password Confirmation: Re-enter the password to prevent wrong password input.

2.7 System information

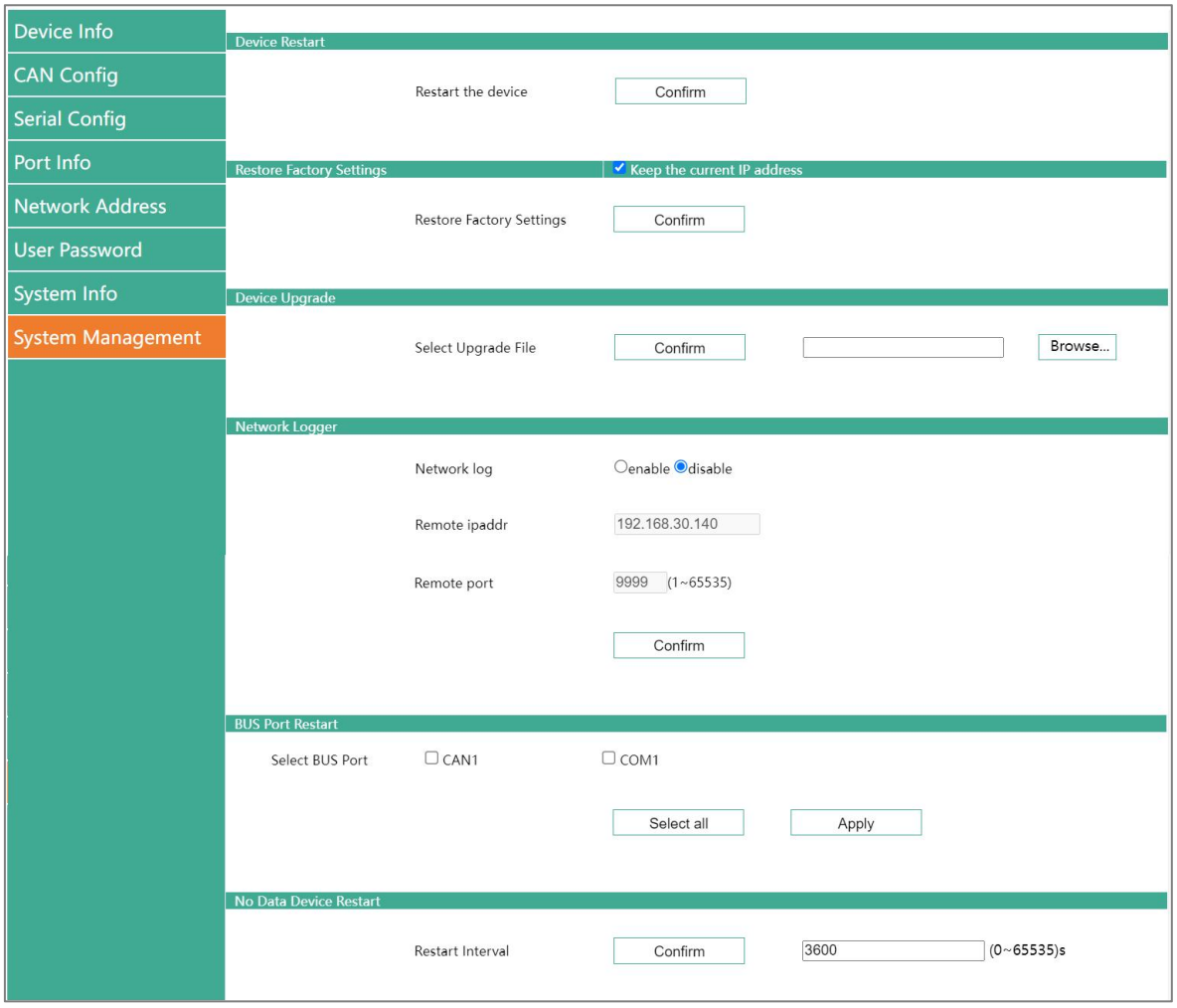
The model, name and serial number of the device can be configured on the system information page, as shown in the figure.



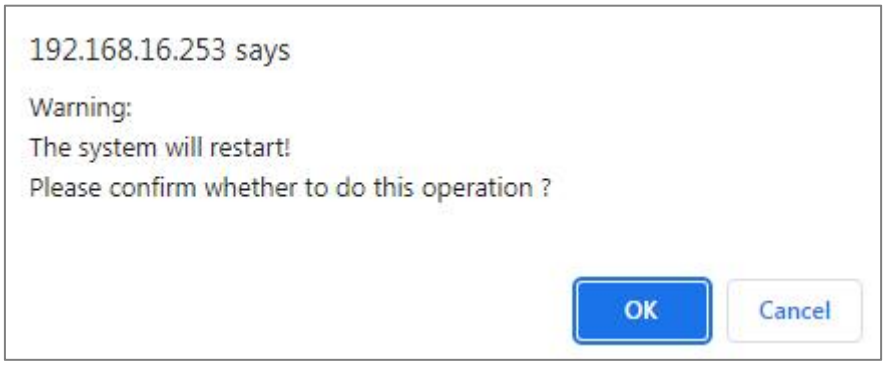
- Device model: The user can customize the model of the CAN server
- Device name: The user can customize the name of the CAN server.
- Device platform: The name of the home platform, which can not be configured by the user.
- Device ID: CAN server serial number, which can not be configured by the user.

2.8 System management

This page can perform some system operations on the CAN server, including restarting the device, restoring the factory configuration, and upgrading the device. Users are advised to use it with caution, as improper operations may damage the CAN server. The page is shown in the figure.

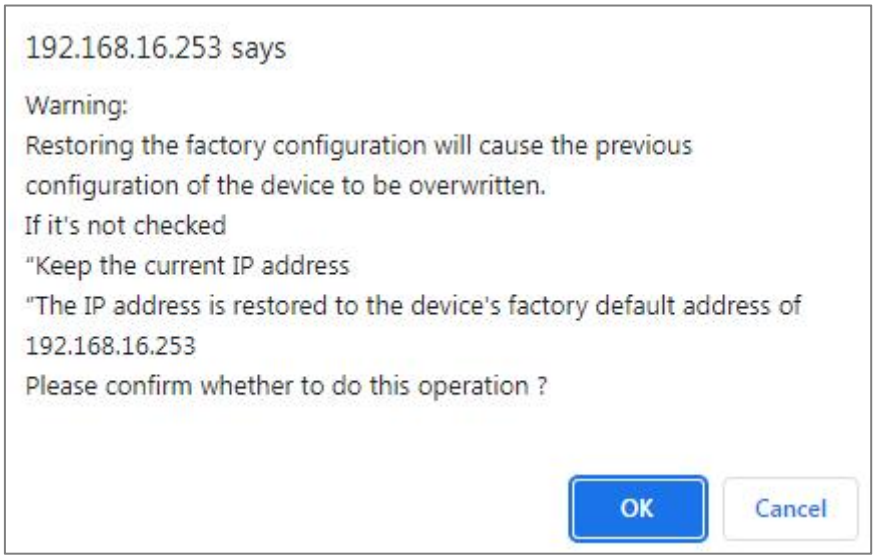


- Device restart: This function is used to restart the CAN server by software. Before the CAN server is completely restarted successfully, the device will not work and cannot forward any data packets. This kind of restart is different from the hardware reset of the power-on restart. It is only the software reset of the CAN server system, just like the "hot start" of the windows operating system. The biggest advantage of this function is to provide a function of remotely restarting the CAN server. Users can remotely restart the CAN server as long as they can access it remotely.

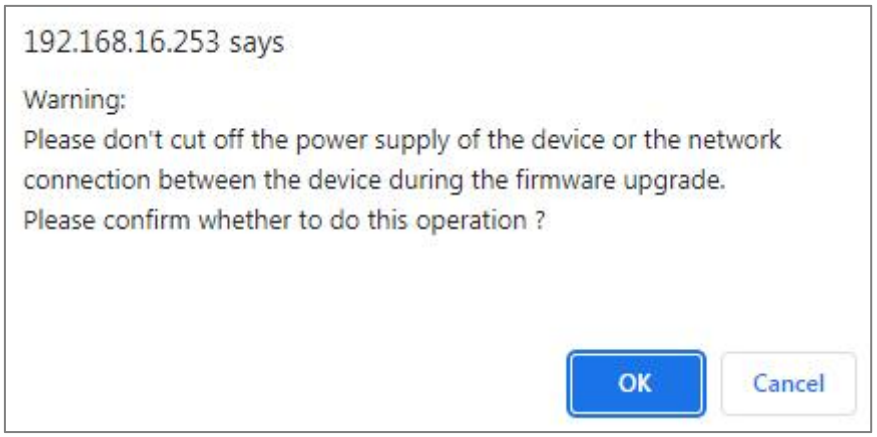


- Device restore factory configuration: This function is used to restore the CAN server to the factory settings and restart the CAN server automatically. Before the CAN server restarts successfully, the CAN server does not work and cannot forward any data packets. This

function is to restore the factory default configuration value once the user sets the wrong parameters and the CAN server does not work normally. On the right side, there is an option of "keep current IP address". When checked, the current IP address will be kept. If not checked, the IP address will also be restored to the factory default address: 192.168.16.253



- Device upgrade: This function is used to perform a system upgrade on the software of the CAN server. Users can get the upgrade program of the CAN server through email or our website. Please pay attention to the matching of the device model and version. The upgrade program will cause the upgrade to fail. After the user obtains the upgrade program, click the "Browse" button to select the upgrade program, and then click the "OK" button, a prompt box will pop up on the page, as shown in the figure, click "OK" to jump to the waiting page. After reading the progress bar in the interface, the CAN server software upgrade is completed.



- Network log: This function is used to transmit the operation log of the CAN server to the remote UDP server by UDP protocol. Users need to specify the IP address and remote port of the remote UDP server.
- Restart CAN: This function is used to soft restart single or all CAN ports without restarting the CAN server device.

- Device restart without data: This function is used for the network port of the CAN server or the CAN port without any data sending and receiving for a long time, and the CAN server will automatically restart. If the restart time is set between 0 and 59 seconds, this function will not take effect. Only when the time is set to be greater than or equal to 60 seconds, the device without data will restart take effect. The default value is 3600s, which is one hour.

Attention:

- Restoring the factory default settings will cause all the settings to return to the factory status. If you want to keep the IP, please check the "keep the current IP address" on the right, otherwise the IP address will also return to the default configuration of 192.168.16.253.
- Do not upgrade the device casually. When the device needs to be upgraded, you must make sure that the upgrade file is correct, otherwise it is easy to damage the software of the device and cause the CAN server to fail.
- Do not operate the CAN server during the upgrade process, and it is forbidden to click on the CAN server web page. If the upgrade is interrupted due to misoperation, please restart the CAN server and try again.
- Power failure is not allowed during the entire upgrade process. Power failure may cause permanent damage to the CAN server. If power failure occurs during the upgrade, please mail the product to our company immediately for possible solutions.
- When setting restart parameters for devices without data, avoid using Google Chrome 68, otherwise there will be a problem that the configuration page cannot respond.

3 Maintenance and Service

From the date of product shipment, Wuhan Maiwe Communication Co., Ltd. provides a five-year product warranty. According to the product specifications of Wuhan Maiwe Communication Co., Ltd., during the warranty period, if the product has any malfunction or functional operation failure, Wuhan Maiwe Communication Co., Ltd. will repair or replace the product for the user free of charge. However, the above commitment does not cover damage caused by improper use, accidents, natural disasters, incorrect operation or incorrect installation. In order to ensure that consumers benefit from the series of products of Wuhan Maiwe Communication Co., Ltd., help and problem solving can be obtained through the following methods:

- Internet service
- Call the technical support office
- Product repair or replacement

3.1 Internet service

Through the technical support section of Wuhan Maiwe Communication Co., Ltd. website, you can get more useful information and usage skills.

3.2 Technical support

Users of Wuhan Maiwe Communication Co., Ltd. products could call technical support office for help. Wuhan Maiwe Communication Co., Ltd. has professional technical engineers to answer your questions and help you to solve the product or usage problems ASAP.

3.3 Product repair or replacement

For product maintenance, replacement or return, in accordance with the processing procedures of Wuhan Maiwe Communication Co., Ltd., you should first contact Wuhan Maiwe Communication Co., Ltd. The technical staff of the company will confirm, and then negotiate with the sales staff of Wuhan Maiwe Communication Co., Ltd. to complete the repair, replacement or return of the product.