PUSR®
www.pusr.com

CAN/Serial to Ethernet/Fiber Optic Converter

USR-CAN528

User Manual

V2.0

Your Trustworthy Smart Industrial IoT Partner

Content

1. Introduction

The USR-CAN528 is an upgraded industrial-grade dual-channel rail-mounted CAN FD gateway independently developed by PUSR. This product features rich interfaces and powerful functionalities, offering high speed, low latency, stable performance, ease of use, and high cost-effectiveness. It can easily connect terminal devices to a CAN-BUS network.

The CAN528 has a variety of interfaces: 2 CAN ports, 1 Ethernet port, 1 SFP slot, and 1 RS485 interface. It supports the CAN FD protocol and is compatible with standard CAN 2.0A/2.0B protocols.

The CAN528 supports flexible communication. It is capable of CAN repeater, CAN to fiber optic, CAN to Ethernet, CAN to RS485, and RS485 to Ethernet conversion. It supports TCP/UDP/MQTT protocols. It features CAN to Modbus RTU/TCP conversion, capable of acting as either a master or slave station.

1.1. Features

- Supports 2 * CAN port, 1 * Gigabit Ethernet, 1 * SFP slot, 1* RS485
- Supports CAN to Ethernet, CAN to fiber optic, CAN to RS485, Ethernet to RS485, and CAN repeater
- Supports CAN FD protocol, compatible with CAN2.0A and CAN2.0B standard protocols
- Supports TCP server, TCP client, UDP server, UDP client, MQTT
- Supports SSL/TLS encryption for secure transmission
- Supports CAN(FD) to Modbus RTU (Master/Slave); CAN(FD) to Modbus TCP (Master/Slave)

Modbus

- Modbus gateway function supports 64 transmit messages and 64 receive messages, enabling polling for data collection, reporting, reading, and writing
- Supports whitelist and blacklist filtering; configurable for filtering extended frames, standard frames, and custom frame IDs
- Supports transparent conversion, transparent conversion with ID, protocol conversion, and custom protocol conversion
- Wide baud rate range: arbitration phase baud rate from 5K to 1Mbps; data phase baud rate from 100K to 5Mbps; supports custom baud rates
- Supports parameter configuration via web interface and AT commands
- Supports firmware upgrade via web interface for easier updates
- Wide operating temperature -40℃~85℃
- Built-in 120-ohm termination resistors, configurable via web interface for connection
- Supports 9-70V wide voltage input with reverse polarity protection
- Features isolation for CAN port, serial port, and network port communications, ensuring stable

operation

- Reliable hardware protection: ESD, surge, and EFT protection (Level 3)
- Hardware watchdog function for automatic reset upon system freeze, enhancing stability and reliability

1.2. Specification

Power supply	
Input voltage	DC: 9 ~ 70 V, reverse polarity protection
Physical characteristics	
Housing	Aluminum
Dimension	137.3*109.3*30 mm(Including terminal blocks)
Mounting	DIN rail mounting
Reload	Long press to restore to factory settings
Ethernet	
Ethernet port	RJ45, 100/1000Mbps, Auto MDI/MDIX
SFP	
SFP slot	100/1000Mbps
Serial port	
Pin	A, B, G
Baud rate	600-230.4K(bps)
Data bit	7, 8
Parity bit	None, Odd, Even
Stop bit	1 、2
CAN port	
Pin	CAN 1: H1, L1, GND CAN 2: H2, L2, GND
Arbitration Phase	5k~1000K bps, supports user-defined baud rate
Data Phase	100K~5Mbps, supports user-defined baud rate, only valid when CAN FD accelerate function is enabled
Terminal register	Built-in 2x 120Ω CAN bus termination resistors
EMC protection	
ESD	Air discharge: 8 kV Contact discharge: 6 kV
Surge	Power: differential mode@0.5KV, common mode@1KV CAN: common mode@1KV RS485/Ethernet: common mode@1KV
EFT	Power: 2KV RS485/CAN/Ethernet: 1KV

Isolation	Isolation for power supply, CAN ports, serial port, and Ethernet port; CAN port isolation up to 3000VDC
Indicator	
RUN	Dual-color LED Red solid on: Power on Green blinking (1Hz): Device operating normally
RS485	Dual-color LED Green blinking: indicates data reception on serial port, blinks every 200ms Red blinking: indicates data transmission on serial port, blinks every 200ms
CAN1	Dual-color LED Green blinking: indicates data reception on CAN port, blinks every 200ms Red blinking: indicates data transmission on CAN port, blinks every 200ms
CAN2	Dual-color LED Green blinking: indicates data reception on CAN port, blinks every 200ms Red blinking: indicates data transmission on CAN port, blinks every 200ms
Operating environment	
Operating Temperature	-40~85℃
Storage Temperature	-40~105℃
Operating Humidity	5% ~ 95% RH(non-condensing)
Storage Humidity	5% ~ 95% RH(non-condensing)

1.3. Indicator description

USR-CAN528 has 4 indicator lights on the front panel: RUN, RS485, CAN1, and CAN2. Users can easily observe the device status through the indicator lights, and the definition of the indicator lights is as follows.

Indicator Light	Color	Function Description
RUN	Green/Red	Red steady on: Power supply is normalGreen blinking: Blinking frequency 1HZ, the device enters working mode; at this time, the red light is off, only the green light blinks
RS485	Green/Red	Green blinking: Indicates that the serial port is receiving data, blinking once every 200msRed blinking: Indicates that the serial port is sending data, blinking once every 200ms
CAN1	Green/Red	Green blinking: Indicates that CAN1 port is receiving data, blinking once every 200msRed blinking: Indicates that CAN1

		port is sending data, blinking once every 200ms
CAN2	Green/Red	Green blinking: Indicates that CAN2 port is receiving data, blinking once every 200msRed blinking: Indicates that CAN2 port is sending data, blinking once every 200ms

1.4. Dimensions

Unit: mm

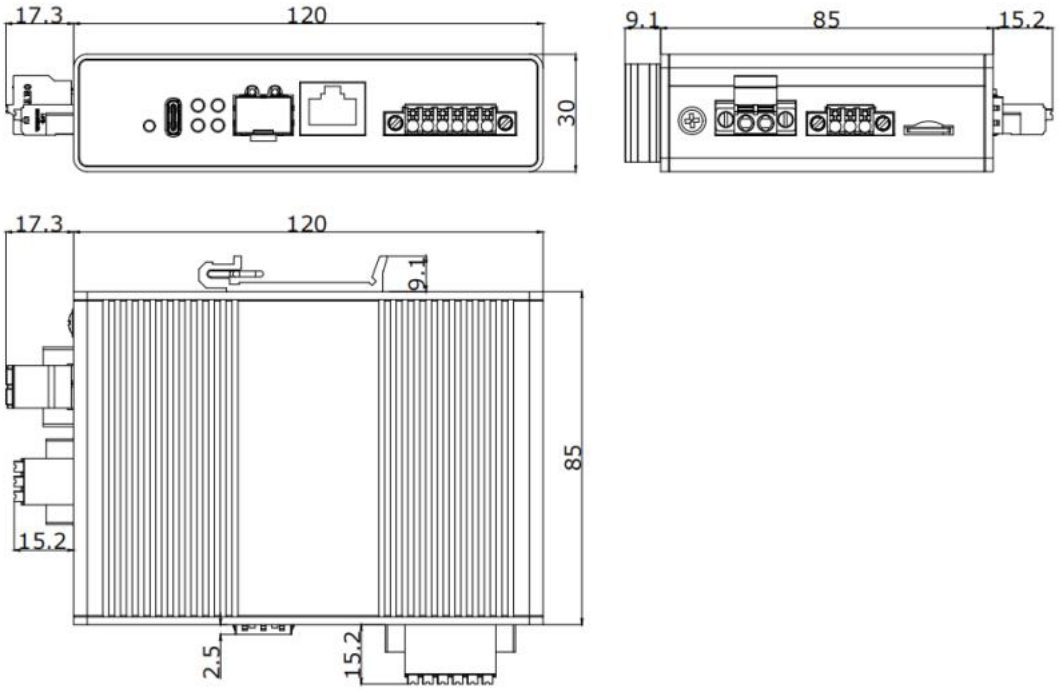
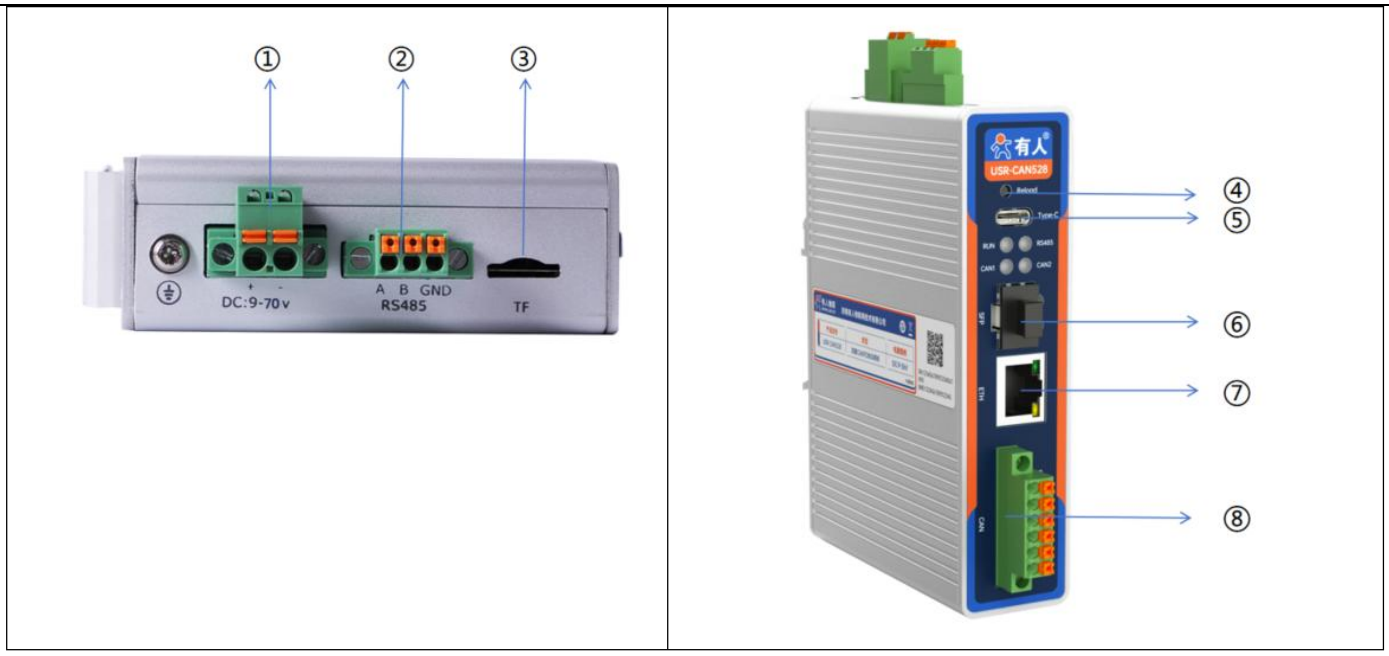


Figure 1. Dimensions of CAN528

- Aluminium enclosure, fixing holes on both sides, compatible with rail mounting brackets.
- 137.3*109.3*30mm (L*W*H, excluding rail, antenna mounts, and mounting hardware).

1.5. Hardware Interface

The interface description of USR-CAN528 is as follows:



When connecting USR-CAN528 to the CAN bus device, CAN_H must be connected to CAN_H, and CAN_L must be connected to CAN_L.

USR-CAN528 has 2 built-in 120Ω terminal resistors, acting on CAN1 and CAN2 respectively. You can select to connect the built-in 120Ω resistor of the module to the CAN bus through the built-in web page; otherwise, the 120Ω resistor will not be connected to the bus.

According to the ISO 11898 standard, to enhance the reliability of CAN-bus communication, terminal matching resistors (120Ω) are usually added at both ends of the CAN-bus network. As shown in the figure below. The value of the terminal matching resistor is determined by the characteristic impedance of the transmission cable. For example, if the characteristic impedance of the twisted pair is 120Ω, the two ends of the bus should also integrate 120Ω terminal resistors.

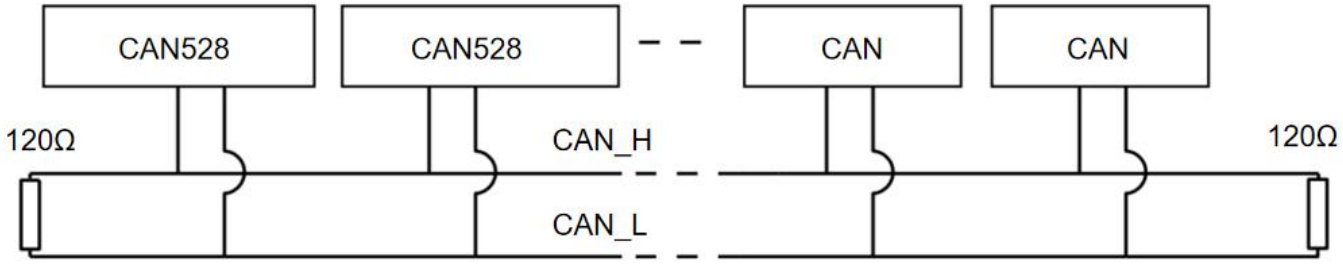


Figure 2. CAN bus connection

1.6. Login to web page

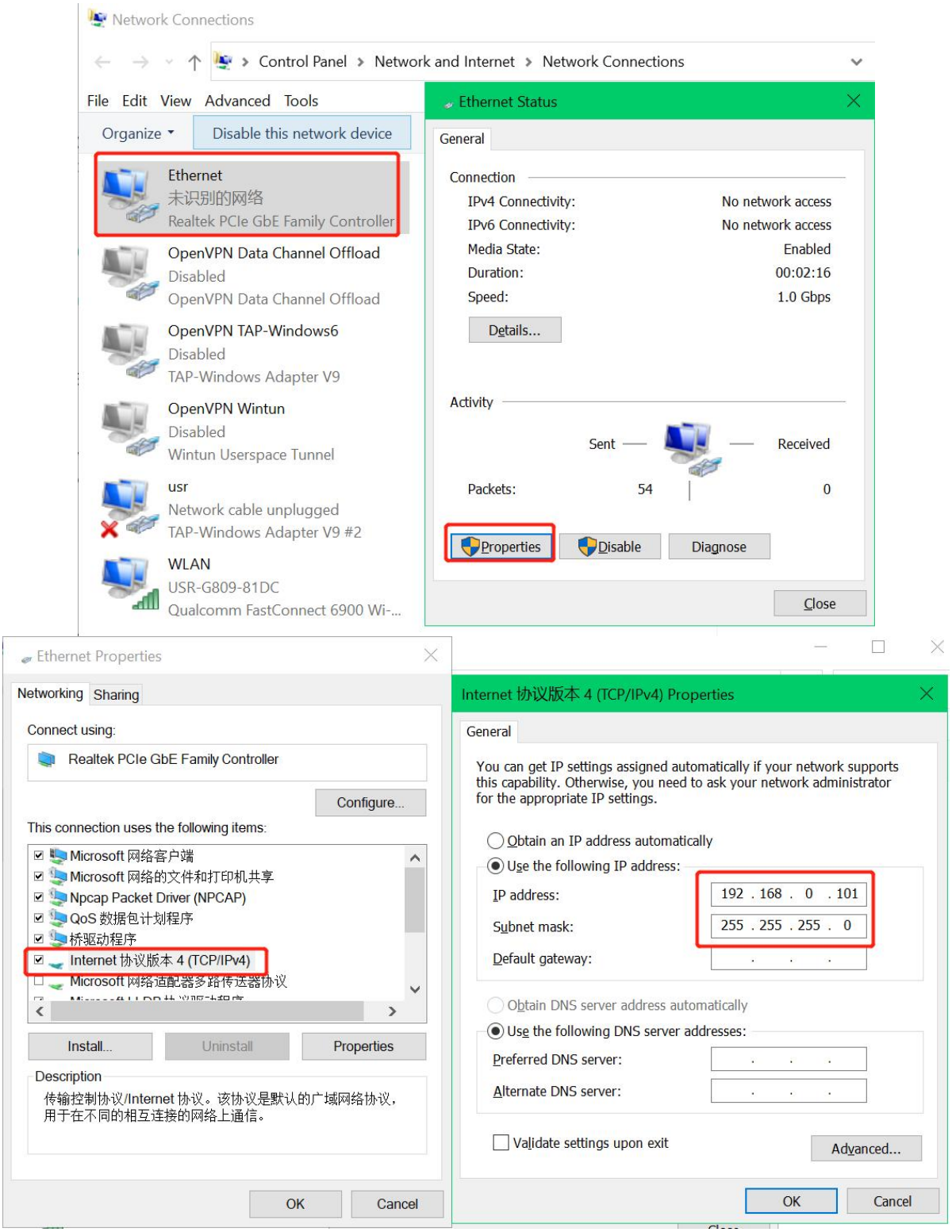
During the use of CAN528, functions and parameters can be set through the web page or AT commands. This

manual mainly describes the functions through web page configuration.

If you need to set parameters through AT command, you can refer to the AT command user manual: AT Command of USR-CAN528-V1.0.0-EN-PUSR IOT

1. Connect PC to the USR-CAN528 via network cable

2. Set the IP of PC to 192.168.0.xx, in this paper, the IP of PC is 192.168.0.101



3. Enter 192.168.0.7 in the browser's URL, the username and password are both admin.

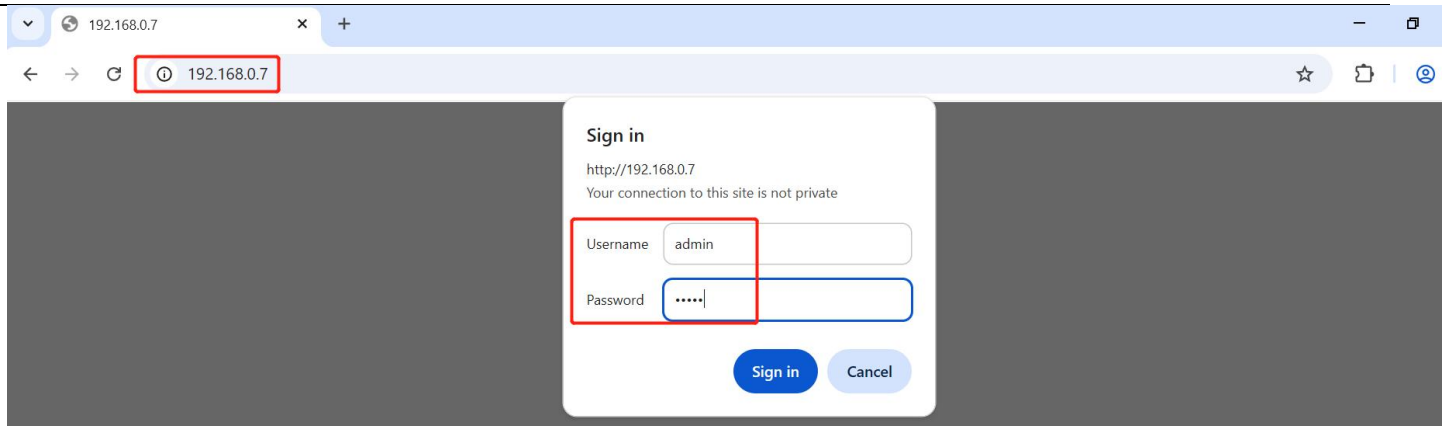


Figure 3. Login page

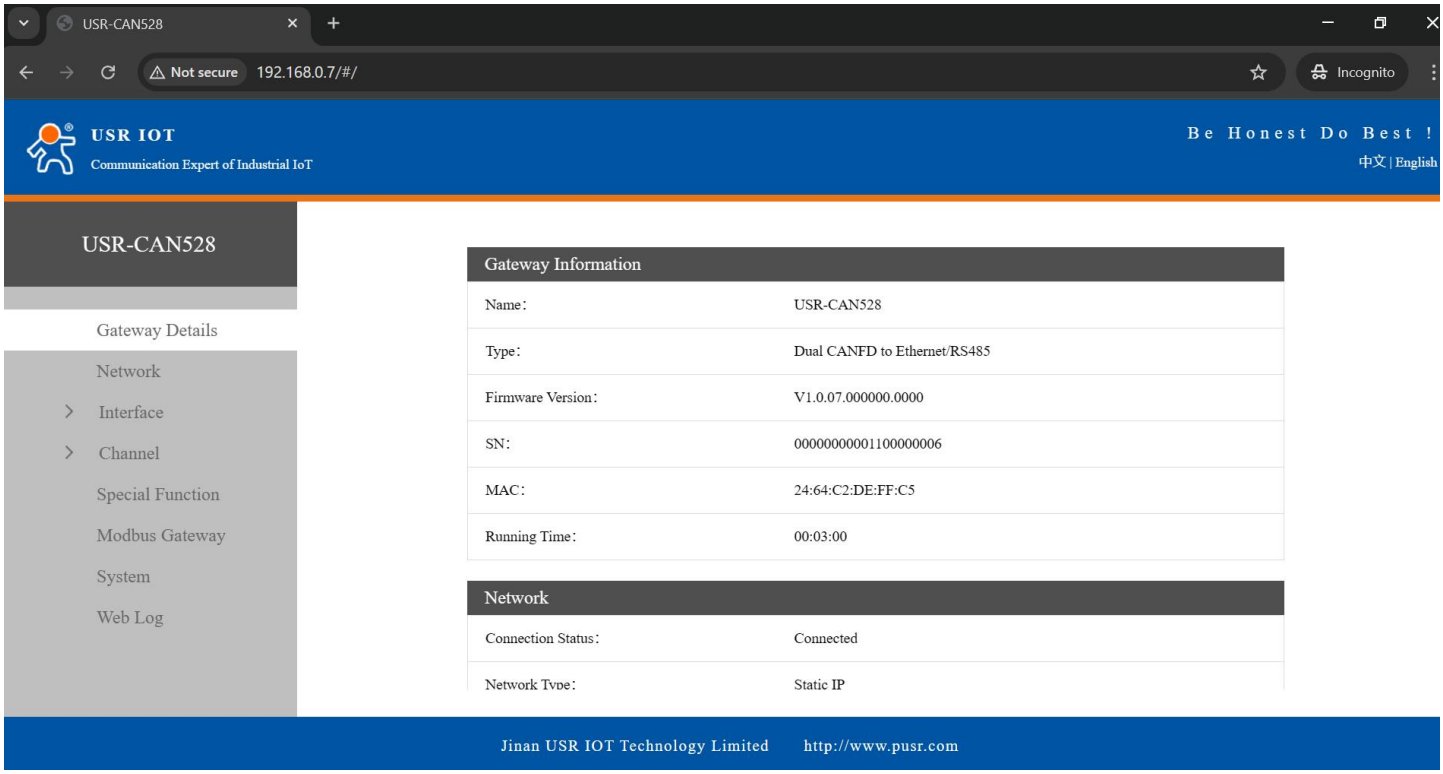


Figure 4. Login successfully

2. Function overview

To help you have a systematic understanding of the functions of CAN528, the functions of CAN528 are briefly introduced first.

CAN528 is equipped with two CAN ports, one Ethernet port (network), one optical fiber port (network), and one RS485 port. The two CAN ports have exactly the same functions and do not affect each other.

(1) Supports Modbus gateway functionality:

Supports CAN to Modbus RTU master/RTU slave/TCP master/TCP slave

First priority function: Enable the Modbus gateway function, and the data transparent transmission function

will be disabled. Note: To switch from Modbus gateway function to transparent transmission, you must disable the Modbus gateway enable.

(2) Supports the following channels for data transmission:

- Convert to network (Ethernet/Fiber):

  CAN1-->SocketA/SocketB/SocketA+SocketB/MQTT

  CAN2-->SocketA/SocketB/SocketA+SocketB/MQTT

  RS485-->SocketA/SocketB/SocketA+SocketB/MQTT

  among：

  SocketA supports TCP Server, TCP Client, UDP Server, and UDP Client

  SocketB supports both TCP Client and UDP Client

- CAN to RS485
- CAN relay

The modes include CAN to Network, CAN to RS485, and CAN Relay (exclusive). Only one channel can be used for direct transmission at any time. To switch from CAN to RS485 or CAN Relay mode back to CAN (RS485) to Network mode, the CAN to RS485 or CAN Relay enable must be disabled.

2.1. Gateway details

View the basic parameters of CAN528 through the built-in web page. You can check the gateway information and network connection details, as shown in the figure below.

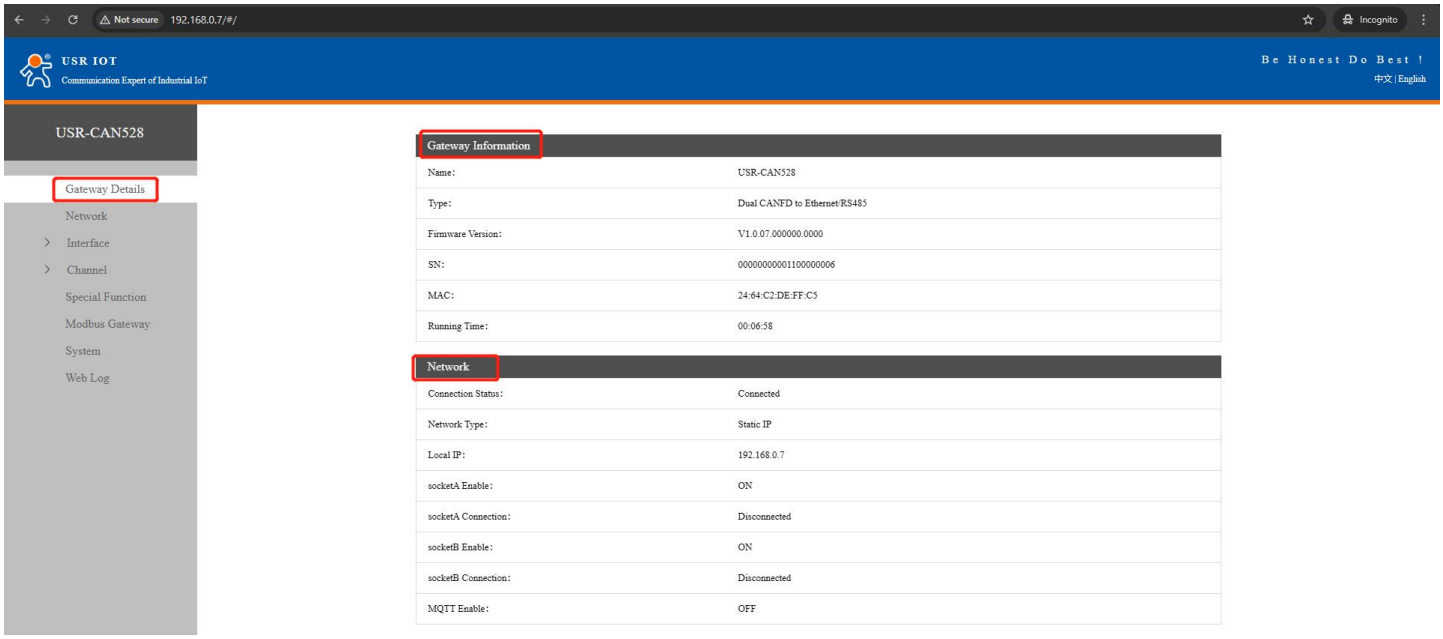


Figure 5. Gateway information overview

2.2. Network settings

The CAN528 features dual Gigabit Ethernet ports, including one standard Ethernet port and one fiber optic interface. Both ports can be used for data transmission as network interfaces.

Ethernet primarily uses RJ45 ports to connect to external networks via Ethernet cables. The communication requires configuration parameters including the host's IP address, subnet mask, gateway, and DNS server.

Network port parameters can be obtained through two methods: dynamic DHCP allocation or static configuration.

The DHCP dynamic acquisition method dynamically retrieves IP addresses, subnet masks, gateway addresses, and DNS server addresses from routers. This approach prevents IP address duplication within local networks while simplifying configuration procedures. It is ideal for scenarios where a one-to-one correspondence between IP addresses and modules is not required.

Static configuration requires users to manually enter local addresses, subnet masks, gateway addresses, and DNS server addresses. It is suitable for scenarios where IP addresses need to be bound to devices. However, when configuring static IP addresses, ensure that each device's IP address within the local network does not conflict with others, as this may affect usage.

The product uses a static IP address by default, with the default IP address being 192.168.0.7.

Note: The CAN528 cannot be configured for DHCP when directly connected to a computer, as most computers lack IP address allocation capability.

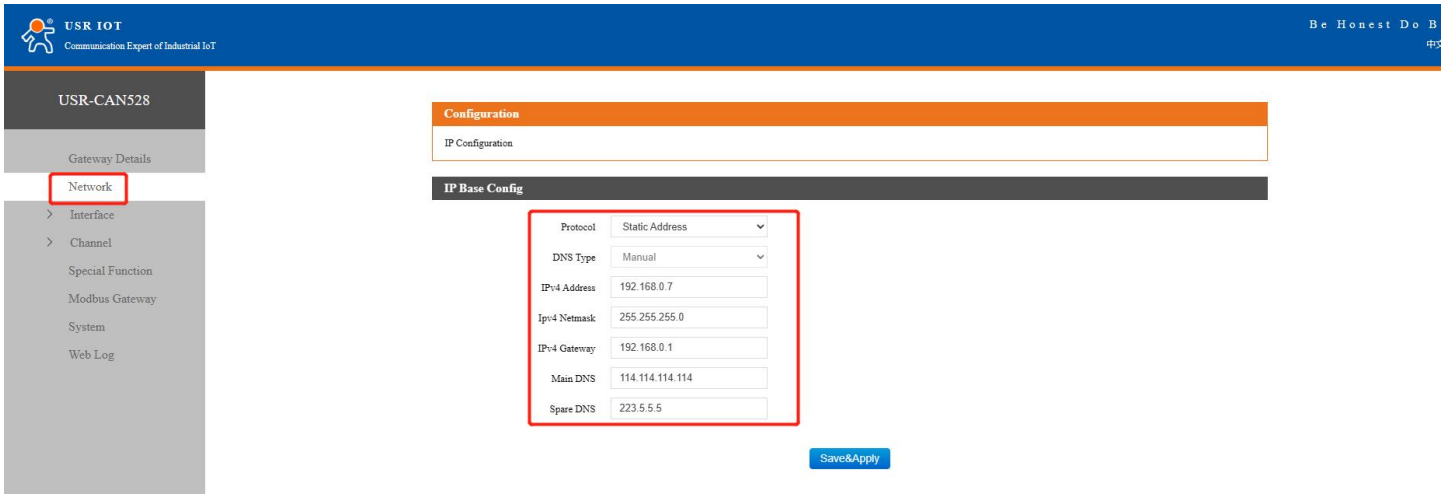


Figure 6. Network settings

Parameter description:

- IP acquisition methods: static IP (manual configuration) and dynamic allocation (DHCP). In dynamic allocation mode, the CAN528 must be connected to devices with IP allocation capabilities, such as routers or Layer 3 switches.

- DNS retrieval methods: Automatic and manual. For static IP addresses, the DNS address must be configured manually and cannot be retrieved automatically.
- Local address: the device's IP address, which is the network identifier for the device. The default is 192.168.0.7.
- Subnet mask: Used to calculate the subnet address of the host's IP address. The default value is 255.255.255.0.
- Gateway address: An IP address that routes traffic between networks. The local IP must be in the same subnet as the gateway IP, with the default value being 192.168.0.1.
- DNS server address: The DNS server address is divided into primary and backup DNS servers, which are used to resolve domain names.

3. CAN settings

The CAN528 features two independent CAN interfaces with identical configuration parameters, which will not be repeated here. During data transmission, whether the data channel is CAN-to-network or CAN-to-serial, the CAN basic configuration, data filtering, and CAN conversion settings remain fully functional.

3.5.1.basic configuration

3.1. Basic settings

Basic CAN port parameter configuration: All data transmitted through CAN ports—whether for CAN-to-serial, CAN-to-network, or CAN relay—requires setting these fundamental parameters.

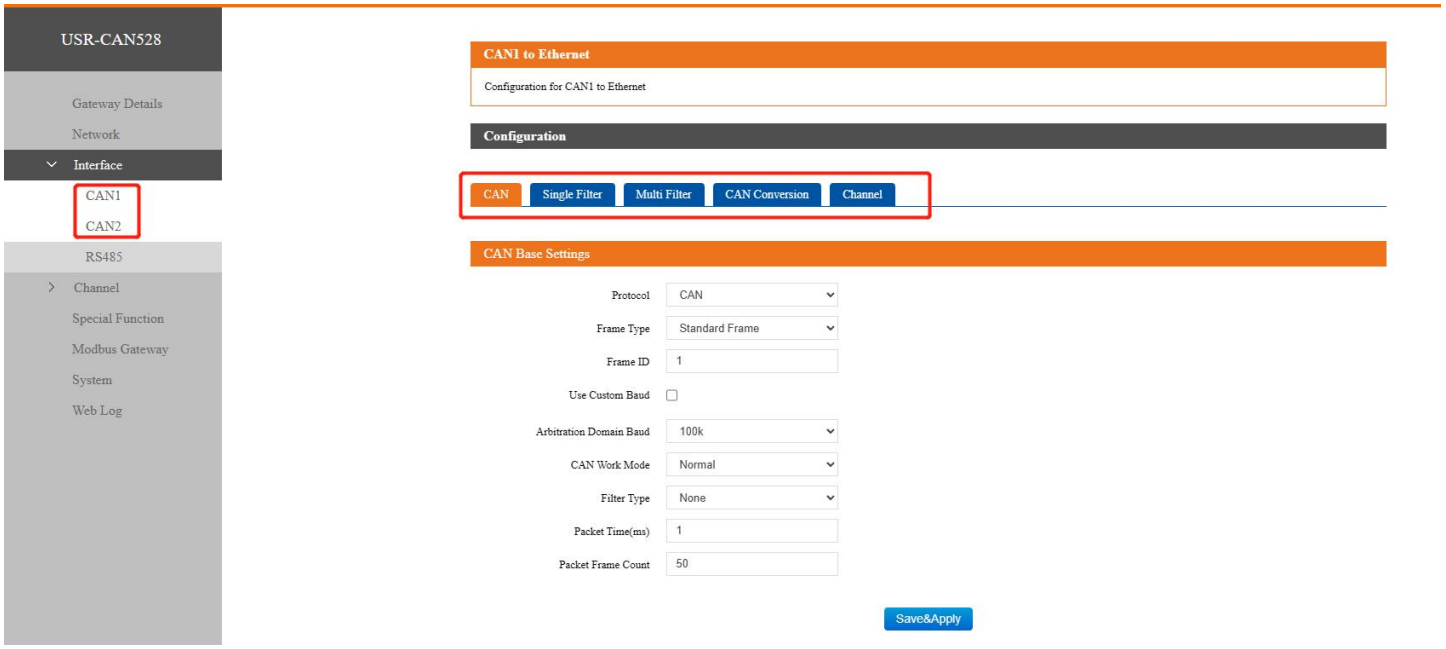


Figure 7. Basic settings of CAN port

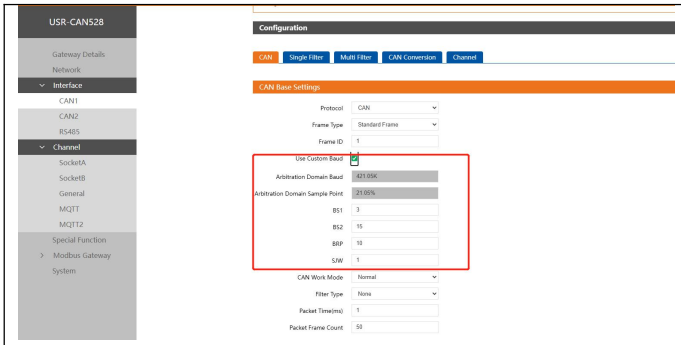
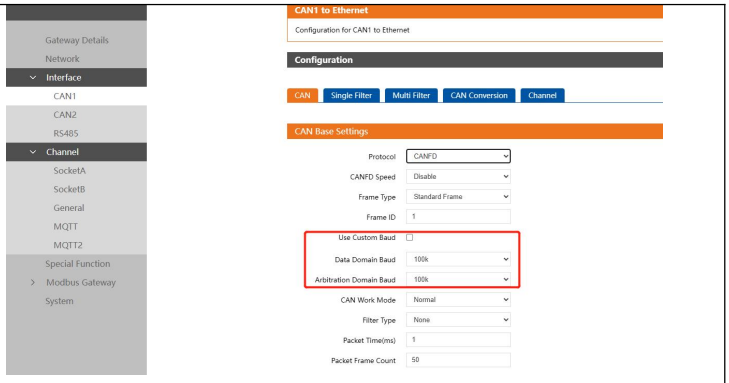
Protocol: Supports CAN or CANFD mode. When selected as CAN, the transponder converts serial data into CAN packets; when selected as CANFD, it converts data into CANFD packets.

CANFD acceleration: This parameter enables the baud rate switching feature of CANFD, which is only active in CANFD mode.

Frame type: The frame type of the CAN message during conversion, with standard and extended frame types available.

CAN ID: Hexadecimal format. Range: 0~7FF (standard frame), 0~1FFFFFFF (extended frame)

Baud rate: Includes arbitration domain and data domain baud rates. Arbitration domain baud rate range: 5Kbps to 1000Kbps, default 100Kbps. Data domain baud rate range: 100Kbps to 5Mbps. The data domain baud rate is only available when CANFD acceleration is enabled. Custom baud rate is supported.

	
Use custom baud rate: Configure the baud rate sampling points. The baud rate calculation formula is: baud = CLK / [(BS1+ BS2+1)] * (BRP)，CLK=80M Parameter value: Arbitration domain: BRP (1~128), BS1 (2~256), BS2 (CAN: 2~128/CANFD: 1~32), SJW (1~512) Data fields: BRP (0~15), BS1 (2~31), BS2 (0~15), SJW (1~32)	Do not use custom baud rate Select common baud rates by dropping down

CAN operating modes: supports three modes: normal, listen-only, and loopback.

- Under normal mode, it can receive and send data normally.
- In silent mode, the CAN port operates in passive listening mode without responding.
- In loopback mode, the transmitted data is both received by the system and relayed to the CAN bus, but cannot be sent to the internal module. This mode is primarily used for testing.

Filter mode: The CAN528 supports ID filtering to selectively receive CAN bus data, minimizing network load.

Disabled by default, it offers single-group or multi-group filtering. For details, see Section 3.5.2 Data Filtering.

CAN packing mechanism:

Since network data is transmitted in data frames, CAN data must be packaged into frame data for efficient network transmission. The CAN528 module performs data packaging based on both packaging time and frame count.

Note: When handling large data volumes or operating socketA, socketB, and MQTT simultaneously, increase the packetization interval appropriately.

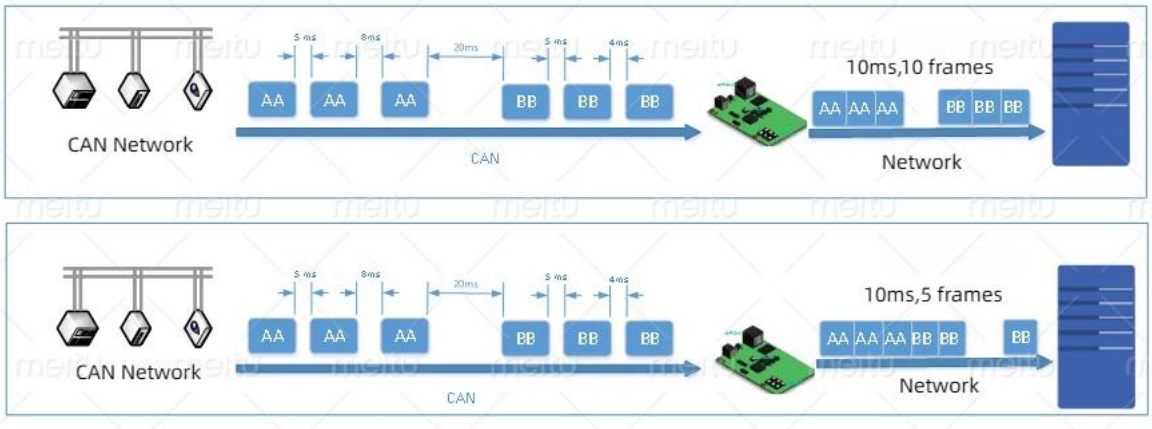


Figure 8. CAN packing instructions

The CAN packetization mechanism is based on either the subframe time or the number of subframes, and packets are transmitted when either condition is met.

Subcontract time: Default is 1 ms. Setable range: 1~254.

Subcontract frame count: Default is 50 frames, adjustable from 1 to 50.

3.2. Signal filter

In the CAN basic configuration filter mode, select the data filtering method (single group or multiple groups). Configuring data filtering allows you to choose which data to receive, effectively reducing the CAN bus load.

Single-group filtering:

Filter mode: Select from standard frame, extended frame, standard frame, or extended frame for the whitelist. Select the whitelist to allow only configured frame ID groups to pass. Select the blacklist to block all frame ID groups.

Minimum frame ID: The minimum frame ID for configuring filter group ID in HEX format. Select standard frames (range: 0~7FF) or extended frames (range: 0~1FFFFFFF)

Maximum frame ID: The maximum frame ID for the filter group ID, in HEX format. Select standard frames (range: 0~7FF) or extended frames (range: 0~1FFFFFFF).

Note: The maximum frame ID must be greater than or equal to the minimum frame ID. Otherwise, the

configuration will fail.

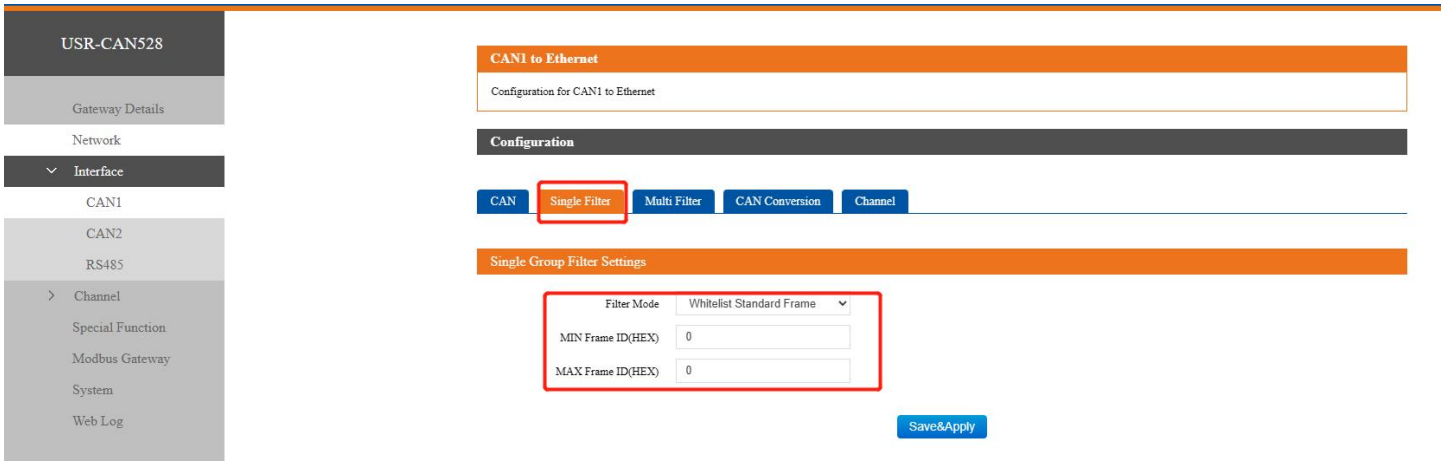


Figure 9. Single filter settings

3.3. Multiple filter

Multiple group filtering:

The CAN528 supports filtering multiple frame ID data sets.

Filter type: Allowlist, Blocklist. Select Allowlist to allow only configured frame ID group data to pass. Select Blocklist to block all configured frame ID group data.

Filter range: Standard frame, Extended frame, Custom. When selecting Custom, you can create new frame ID groups, supporting up to 32 groups. You can customize the frame type to filter, minimum frame ID, and maximum frame ID.

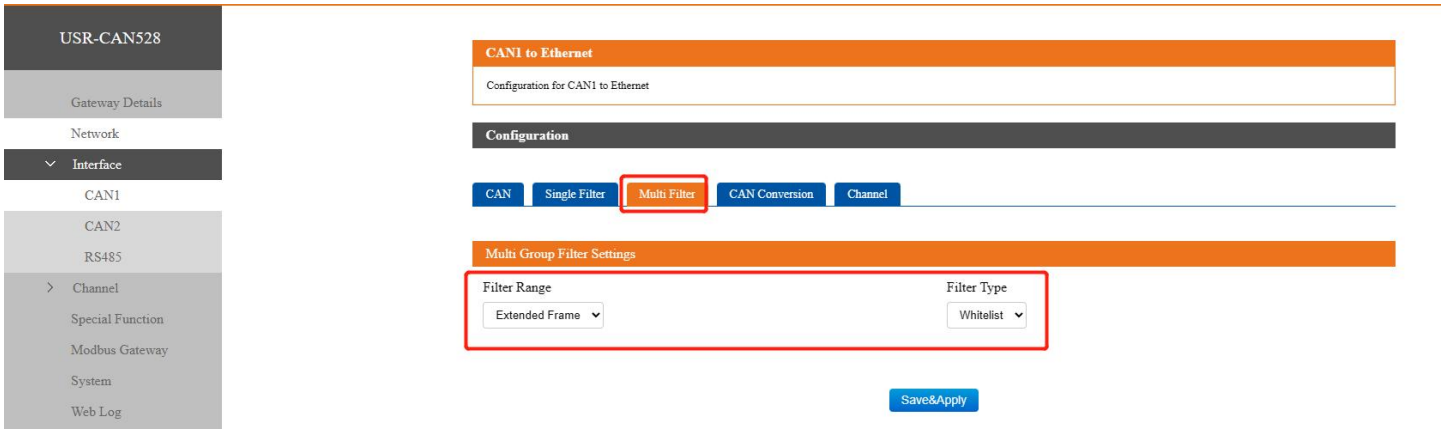


Figure 10. Multiple filter settings

3.4. CAN conversion

Direction of conversion: By selecting the conversion direction, you can eliminate unnecessary data interference on the bus side. There are three conversion directions:

- Bidirectional: The converter transmits data from the network to the CAN bus and vice versa.
- Network-to-CAN: Transforms data from the network to the CAN bus, but not vice versa.
- CAN-to-network port only: Transforms CAN bus data to network traffic, but not vice versa.

Conversion modes: Supports transparent conversion, transparent band ID conversion, standard protocol conversion, and custom frame header/tail conversion. Each mode has different conversion rules, enabling mutual conversion between serial frame information and CAN (FD) frame information.

3.4.1. Transparent Conversion

In transparent conversion mode, CAN528 receives data from one bus and immediately transmits it to the other bus without adding or modifying the data. This achieves data format exchange without altering the content, making the converter transparent to both buses.

The CAN message frame information (frame type section) and frame ID are pre-configured by the user and remain unchanged during conversion. The user can choose whether to enable frame ID or frame information, enabling the conversion of frame information and frame ID.

Enable frame information: Only effective in transparent conversion. When selected, the converter adds the frame information of CAN (FD) messages to the first byte of the serial frame. If not selected, the frame information of CAN (FD) is not converted.

Enable Frame ID: Only active in transparent conversion mode. When selected, the converter adds the CAN (FD) frame ID before the serial frame data and after the frame information (if enabled). If not selected, the CAN (FD) frame ID is not converted.

This method does not increase the user's communication load, and can convert data in real time without any distortion, thus capable of handling large-volume data transmission.

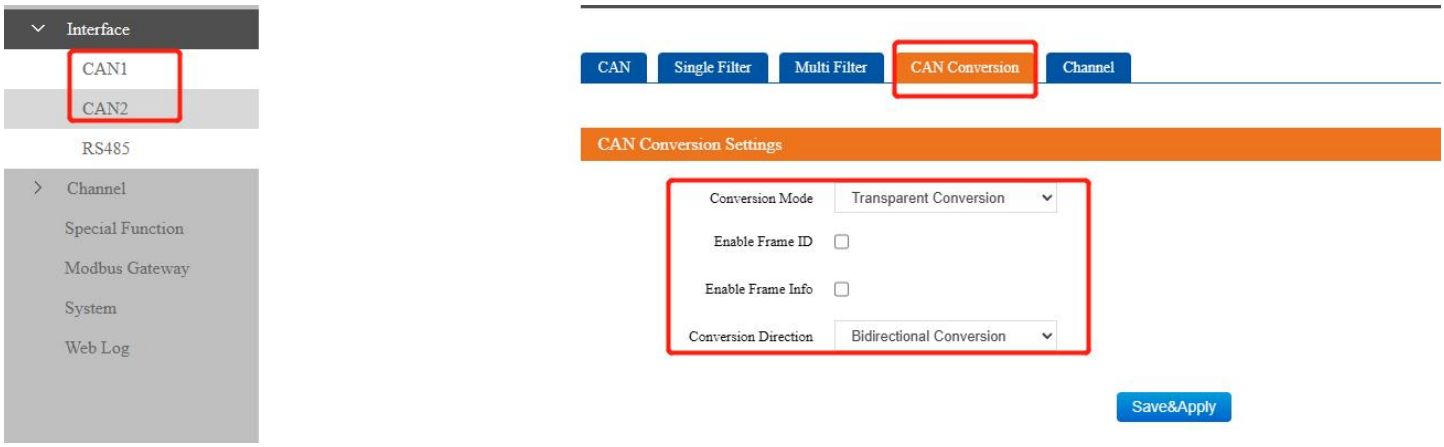
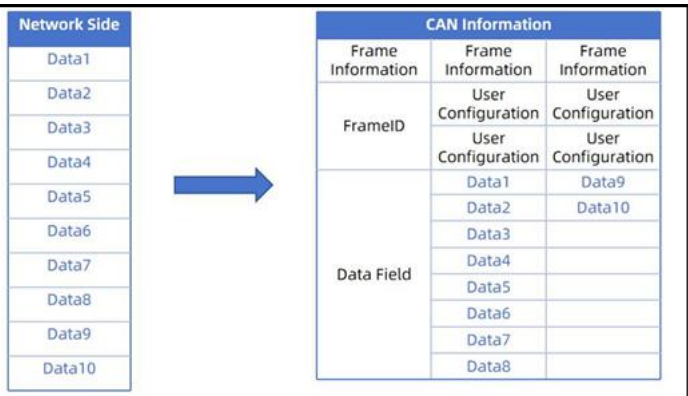


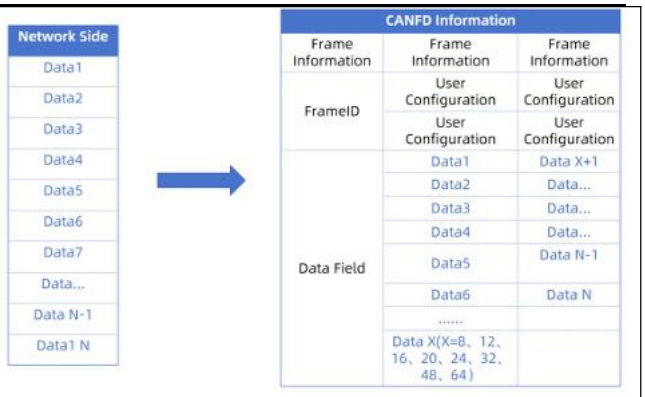
Figure 11. Transparent Conversion Configuration

The transparent conversion data transmission is shown as follows:

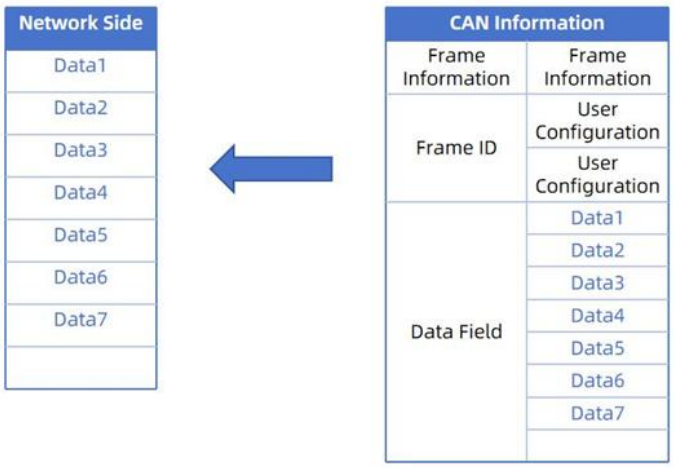
Ethernet to CAN conversion	Ethernet Data to CANFD

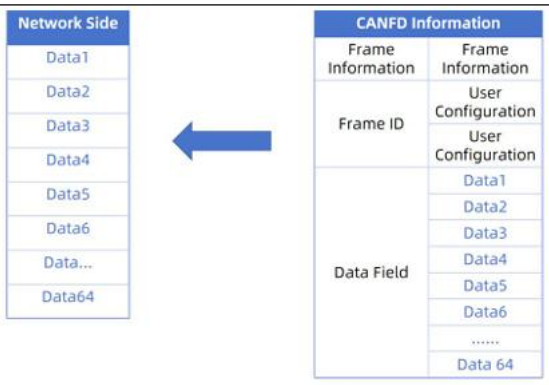


CAN changeEthernet data (disabled frame information/frame ID)



CANFD changeEthernet data (enable frame information/enable frame ID)





3.4.2. Zona pellucida ID conversion

The transparent band ID conversion is a specialized technique for transparent conversion without requiring additional protocols. This method converts the "address" in Ethernet data into the identifier field of a CAN (FD) message, where both the starting position and length of the frame ID in Ethernet data can be configured. The converter extracts this frame ID during conversion and fills it into the frame ID field of the CAN (FD) message, serving as the ID for the CAN (FD) message when forwarding the Ethernet data. Similarly, when converting a CAN message back to Ethernet data, the CAN (FD) message ID is also converted and placed at the corresponding position in the Ethernet data.

Clearing Belt ID Length: This parameter applies exclusively to clearing belt ID conversion. When serial data is converted to CAN (FD) frames, it defines the starting byte of the CAN (FD) frame ID within the serial frame. The frame ID length can accommodate 1-2 bytes for standard frames (corresponding to CAN (FD) frame IDs ID1 and ID2) or 1-4 bytes for extended frames (ID1, ID2, ID3, and ID4). The ID length is 11 bits for standard frames and 29 bits for extended frames.

Clear zone ID position: This setting applies only to clear zone ID conversion. It indicates the starting byte offset of the CAN (FD) frame ID within the serial frame when converting serial data to CAN (FD) frames.

In this way, the converter can adapt to the user's custom protocol to the maximum extent.

Note: In this conversion mode, the 'CAN ID' field in the software's CAN parameter settings becomes invalid, as the transmitted identifier (frame ID) is populated by data from the aforementioned Ethernet data.

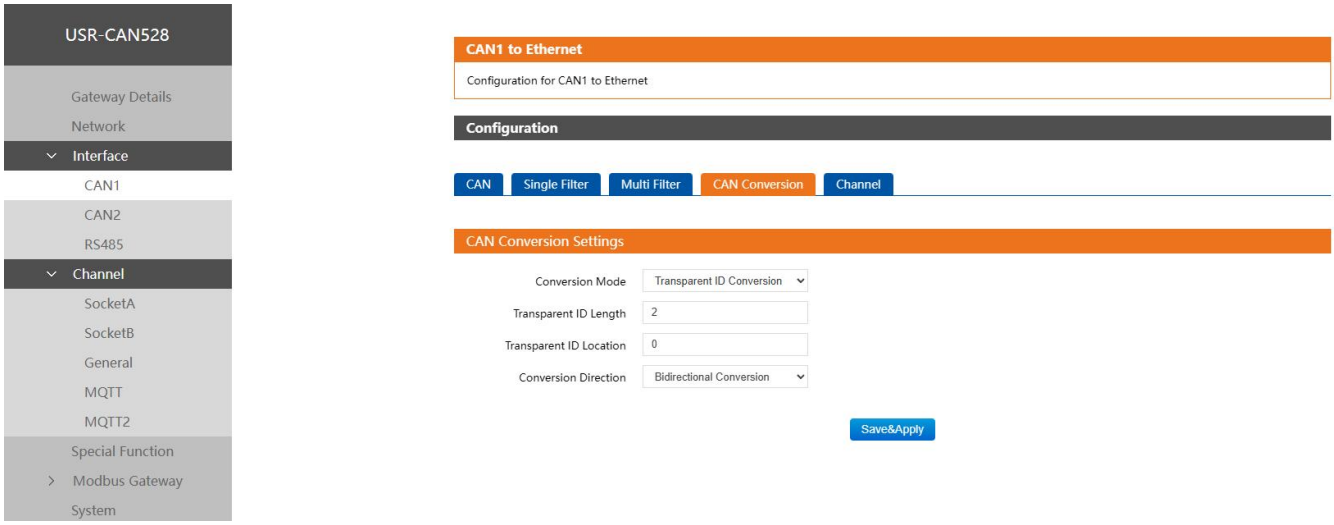


Figure 12. Zona pellucida ID conversion configuration

The data transmission process for coriolis force ID conversion in the zona pellucida is illustrated as follows:

Ethernet to CAN conversion

Address	Ethernet Data
0	Data1
1	Data2 (ID1)
2	Data3 (ID2)
3	Data4
4	Data5
5	Data6
6	Data7
7	Data8
...	...
N-2	Data N-1
N-1	Data N

CAN Information		
Frame Information	Frame Information	Frame Information
Frame ID	ID1 (0x00)	ID1 (0x00)
	ID2 (0x00)	ID2 (0x00)
	Data2 (ID3)	Data2 (ID3)
	Data3 (ID4)	Data3 (ID4)
Data Field	Data1	Data...
	Data4	Data N-1
	Data5	Data N
	Data6	
	Data7	
	Data8	
	Data9	
	Data10	

Ethernet Data to CANFD

Address	Ethernet Data
0	Data1
1	Data2 (ID1)
2	Data3 (ID2)
3	Data4
4	Data5
5	Data6
6	Data7
7	Data8
...	...
N-2	Data N-1
N-1	Data N

CANFD Information		
Frame Information	Frame Information	Frame Information
Frame ID	ID1 (0x00)	ID1 (0x00)
	ID2 (0x00)	ID2 (0x00)
	Data2 (ID3)	Data2 (ID3)
	Data3 (ID4)	Data3 (ID4)
Data Field	Data1	Data...
	Data4	Data N-1
	Data5	Data N
	Data6	
	Data7	
	.....	
	Data65	
	Data66	

CAN to Ethernet data

Ethernet Data
Data1
FrameID3
FrameID4
Data2
Data3
Data4
Data5
Data6
Data7
Data8

CAN Information	
Frame Information	Frame Information
Frame ID	FrameID1
	FrameID2
	FrameID3
	FrameID4
Data Field	Data1
	Data2
	Data3
	Data4
	Data5
	Data6
	Data7
	Data8

CANFD to Ethernet data

Ethernet Data
Data1
FrameID3
FrameID4
Data2
Data3
Data4
Data5
Data6
.....
Data64

CANFD Information	
Frame Information	Frame Information
Frame ID	FrameID1
	FrameID2
	FrameID3
	FrameID4
Data Field	Data1
	Data2
	Data3
	Data4
	Data5
	Data6
	.....
	Data64

3.4.3. Standard protocol conversion

The standard CAN frame format consists of 13 bytes per frame, including: frame header (1 byte), frame ID (4 bytes), and data frame (8 bytes).

The standard CANFD frame format consists of 69 bytes, including: CANFD frame information (1 byte), frame ID (4 bytes), and data frame (64 bytes).

By properly configuring frame information (the first byte of data), you can flexibly transmit standard frames, extended frames, or even remote frames. Correct parsing of serial frames yields detailed information about standard frames, extended frames, and remote frames.

Pay attention to：

In this conversion mode, the 'frame ID' and 'frame type' in the basic configuration are invalid, as the frame ID sent is filled with the frame ID data from the serial frame, and the frame type is determined by the frame information in the serial frame.

In this mode, strict adherence to the standard serial data format is required for successful conversion. Frame information must be error-free, with all reserved bits set to zero. CAN frames have a fixed length of 13 bytes, while CANFD frames are 69 bytes. Any missing data must be padded with zeros; otherwise, transmission will fail.

The standard CAN frame format is as follows:

CAN fixed format (1 CAN frame contains 13 bytes)		
frame information	frame ID	frame data
1Byte	4Byte	8Byte

Fixed CANFD format (1 CANFD frame contains 69 bytes)		
frame information	frame ID	frame data
1Byte	4Byte	64Byte

Frame information: 1 byte in length, used to identify frame information: frame type and frame length.

Bit7	Bi6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FF	RTR	EDL	BRS	DLC3	DLC2	DLC1	DLC0

FF: The flag for standard and extended frames, where 1 indicates an extended frame and 0 indicates a standard frame.

RTR: A flag to distinguish between remote frames and data frames, where 1 indicates a remote frame and 0 indicates a data frame. In CAN FD mode, the flag can only be 0.

EDL: CAN and CANFD identifiers, where 0 indicates CAN and 1 indicates CANFD.

BRS: Bit rate switching enable flag. 0 means no rate conversion, 1 means rate conversion is allowed. This bit is valid only for CANFD and should be 0 for CAN.

DLC3~DLC0: Data Length Bits, indicating the data length of the CAN (FD) frame.

Frame ID: 4 bytes, with high bits first and low bits last. The standard frame valid bit is 11 bits, and the extended frame valid bit is 29 bits.

12h	34h	56h	78h

Extended frame ID: 0x12345678

00h	00h	01h	23h

This ID can represent either an extended frame ID or a standard frame ID.

Extended frame ID: 0x00000123

Standard frame ID: 0x0123

Extended frame and standard frame IDs are distinguished by frame information

Frame data: 8 bytes in CAN mode, 64 bytes in CAN FD mode. If the data is incomplete, pad with 00.

3.4.4. Custom Frame Header and Footer Conversion

To facilitate user convenience in using CAN-bus, the serial frame format is adapted to the CAN frame format. A frame's start and end, namely the "frame header" and "frame footer," are defined in the serial frame, which can be configured by the user.

The serial frame format must comply with the specified frame format; otherwise, it cannot be transmitted correctly. A serial frame must include: frame header, data length, data field, and frame tail.

The header and trailer are customized by the client, each 1 byte.

Custom frame header: Only applies when the frame header and footer are converted. Users can customize the serial frame header. Length: 1 byte.

Custom frame end: Only applies when the frame header and end are converted. Users can customize the serial frame end. Length: 1 byte.

Data length refers to the byte length of a data field. To ensure proper transmission, the data length must match the frame tail data; otherwise, the frame will be discarded. For example, if the frame header is configured as AA and the frame tail as FF, a serial frame AA 03 01 02 04 FF can be transmitted normally. However, if a serial frame is sent as AA 03 01 02 03 04 FF, and the data field 01 02 03 is followed by 04 instead of the frame tail FF, the frame will be discarded and cannot be transmitted.

Similar to transparent conversion, in custom protocol conversion, CAN ID and CAN type must be configured

manually. Frame headers, trailers, and data lengths are not converted into CAN frames.

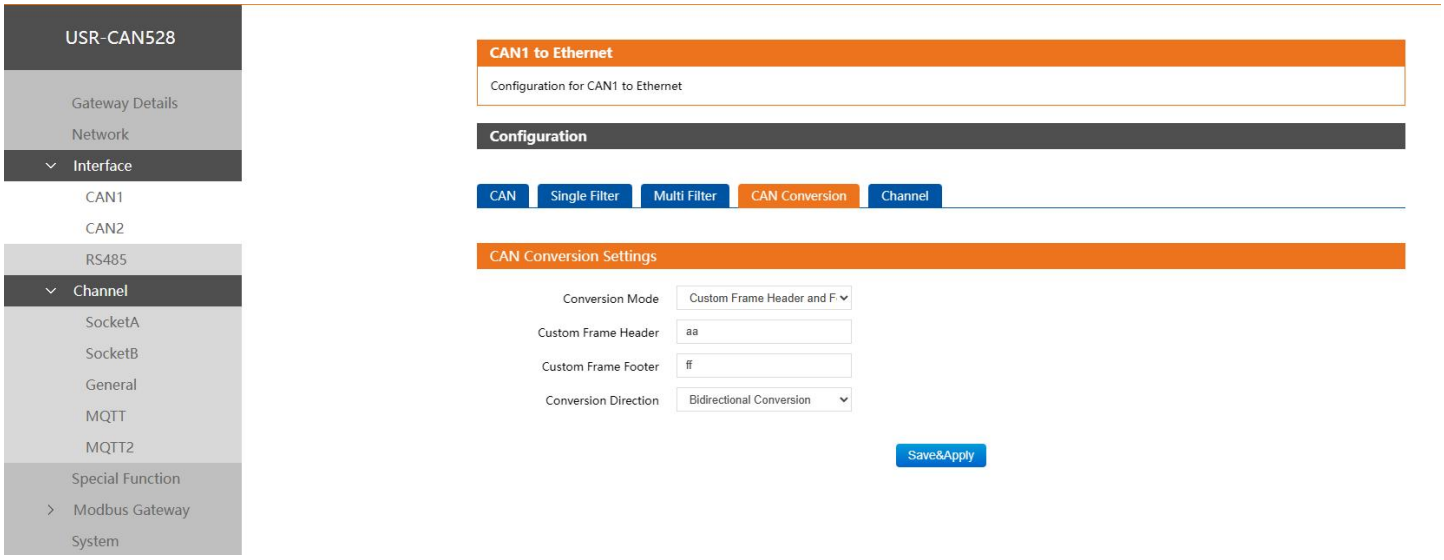
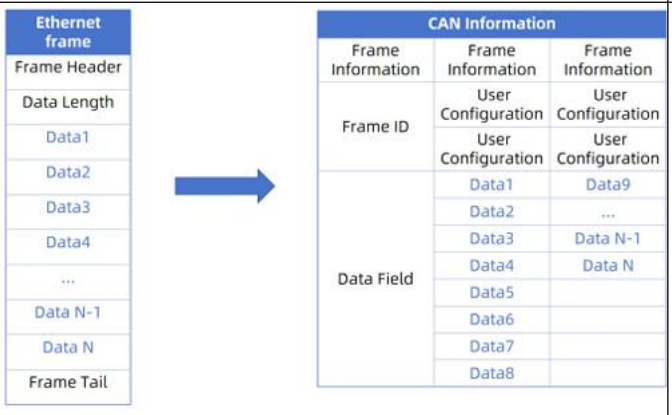


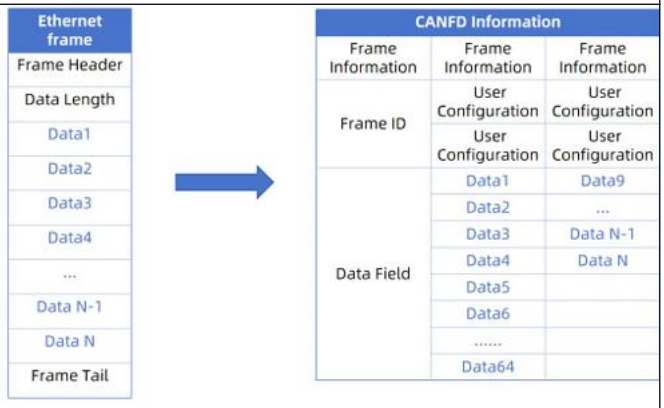
Figure 13. Custom Frame Header and Footer Conversion Configuration

The diagram illustrates the data transmission process for converting custom frame headers and trailers as follows:

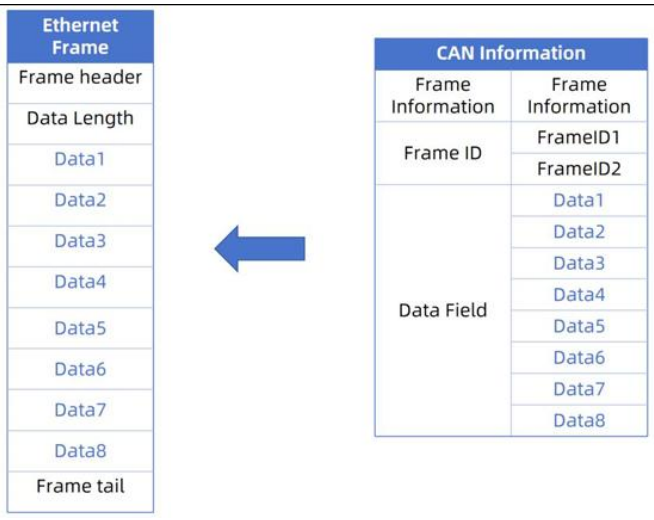
Ethernet to CAN conversion



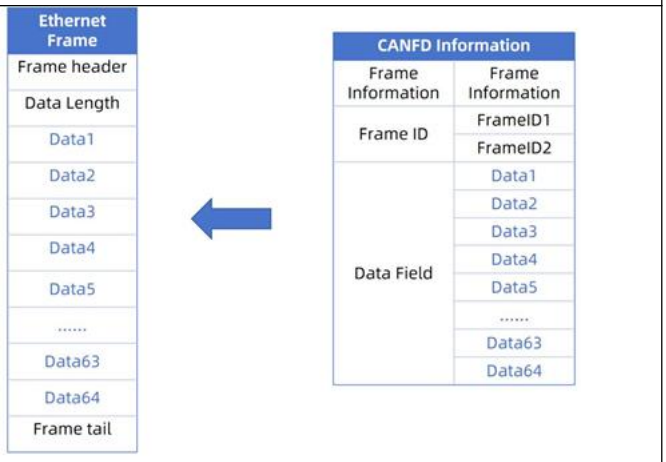
Ethernet Data to CANFD



CAN to Ethernet data



CANFD to Ethernet data



3.4.5. Socket channel

When converting CAN to network transmission, the CAN528 data channel must be selected, and data will be transmitted through the designated channel.

Optional: CAN to SocketA/SocketB/SocketA+SocketB (CAN1/CAN2 identical)

For CAN-to-MQTT conversion, perform the configuration separately on the MQTT interface. Refer to MQTT Functions for details.

Figure 14. Socket channel

3.5. RS485 settings

When data requires serial port transmission—whether converting CAN to serial or serial to network—configuration of the serial port's basic parameters is required.

- Baud rate: 600~230400bps, default 115200bps
- Data bits: 7, 8, default 8
- Check bits: No parity, odd parity, even parity, default is no parity
- Stop position: 1, 2, default 1
- Packaging length (bytes): 256~1024 bytes. During data transmission, serial port data is packaged according to the set length.

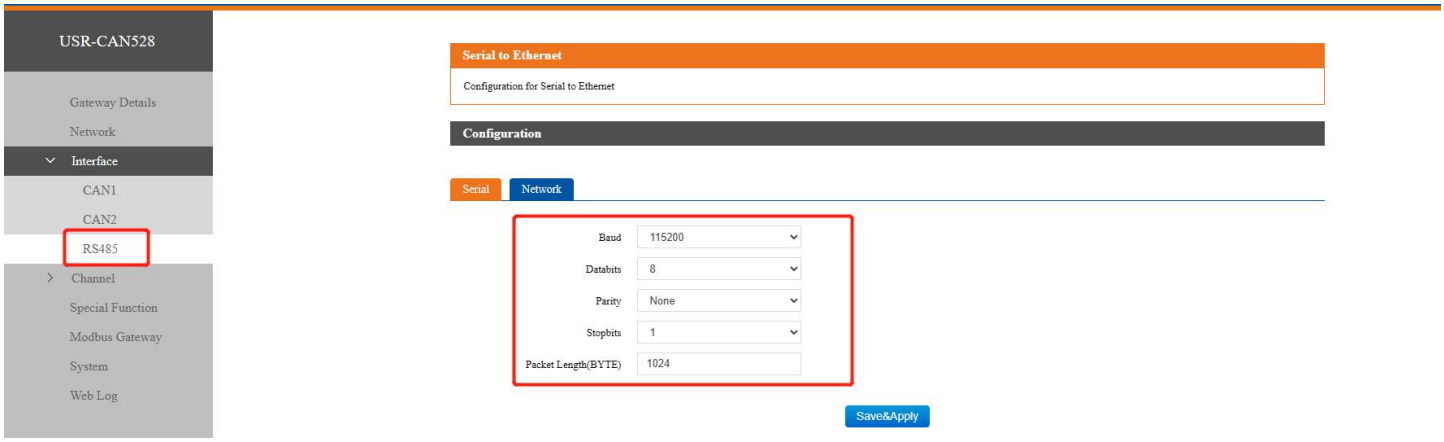


Figure 15. Basic settings of RS485

4. Socket Settings

To enable CAN/Serial-to-network conversion, configure CAN/Serial parameters, select a network transmission channel, and set channel parameters. The system supports three channels: two Socket channels and one MQTT channel. Socket A supports TCP Client, TCP Server, UDP Client, and UDP Server modes, while Socket B supports TCP Client and UDP Client modes.

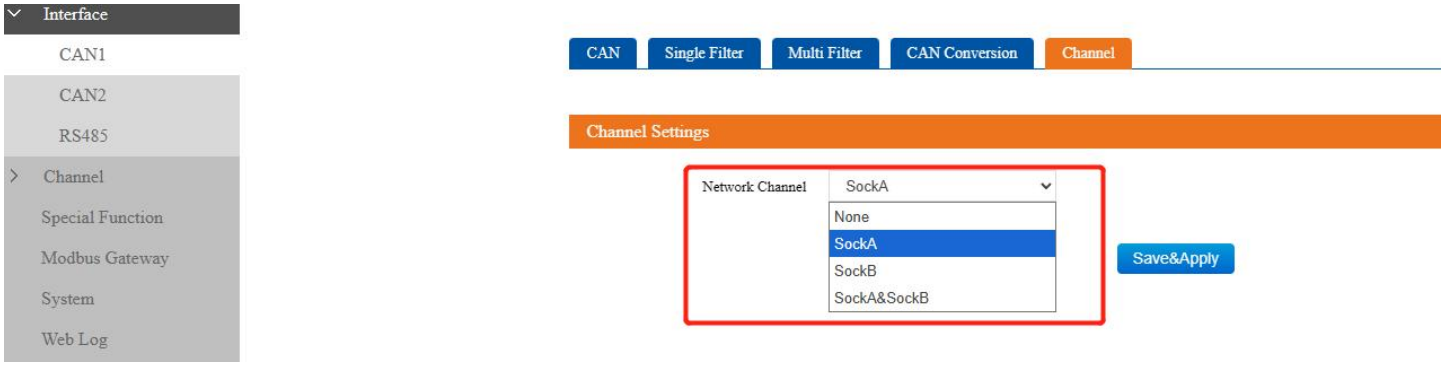


Figure 16. Socket channel of CAN Port

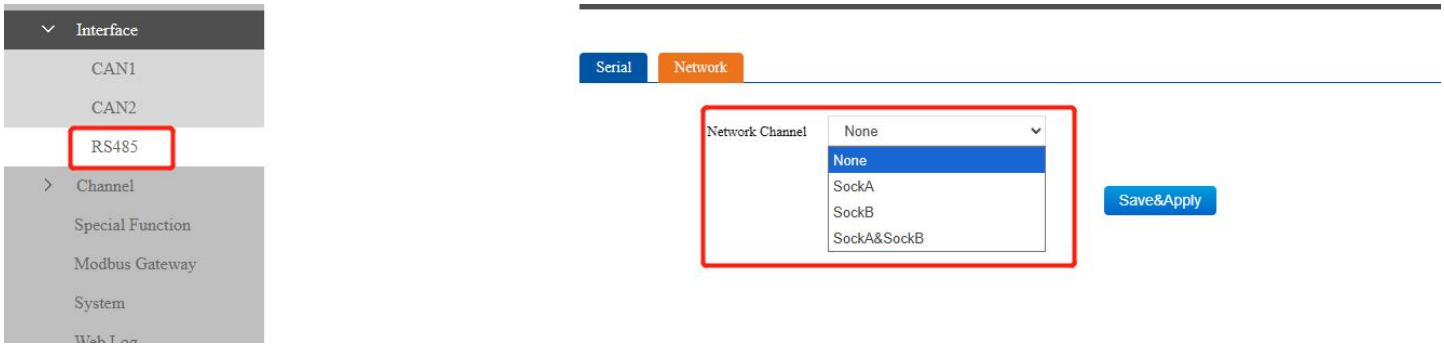


Figure 17. Socket channel of RS485

4.1. Socket Function

4.1.1. TCP Client

(1) Description:

The TCP Client provides client connections for TCP network services. It actively initiates connection requests to servers and establishes connections to enable data exchange between serial ports and servers. It is commonly used for data interaction between devices and servers and is the most common method of network communication.

1)The TCP Client establishes client connections for TCP network services. It initiates connections and connects to servers to facilitate data exchange between serial ports and servers. As per TCP protocol specifications, the TCP Client distinguishes between established and disconnected connections to ensure reliable data transmission.

2)To use CAN528 as a TCP client, you must establish a connection to a TCP server. Key parameters to note

include the target IP address or domain name and the target port number. The target IP can be either a local device within the same LAN, an IP address from a different LAN, or an IP address across public networks. If connecting to a server across public networks, the server must have a public IP address or domain name.

3)When configured as a TCP client, CAN528 initiates connections to the target port on the specified IP address and does not accept any other connection requests.

4)When using CAN528 as a TCP client, set its local port to 0. This allows CAN528 to access the server through a random port, preventing connection failures caused by the server blocking reconnection requests due to abnormal status. If the local port is not 0, the device must wait 2 minutes before establishing a connection with the server.

5)Disable data reconnection: Enable data reconnection without a set duration. The timer starts from the last received TCP data. When the duration expires, the device reconnects to the TCP server.

6)Within the same LAN, if CAN528 is configured as a static IP, ensure its IP address and gateway are on the same subnet and the gateway IP is correctly set. Otherwise, communication will fail.

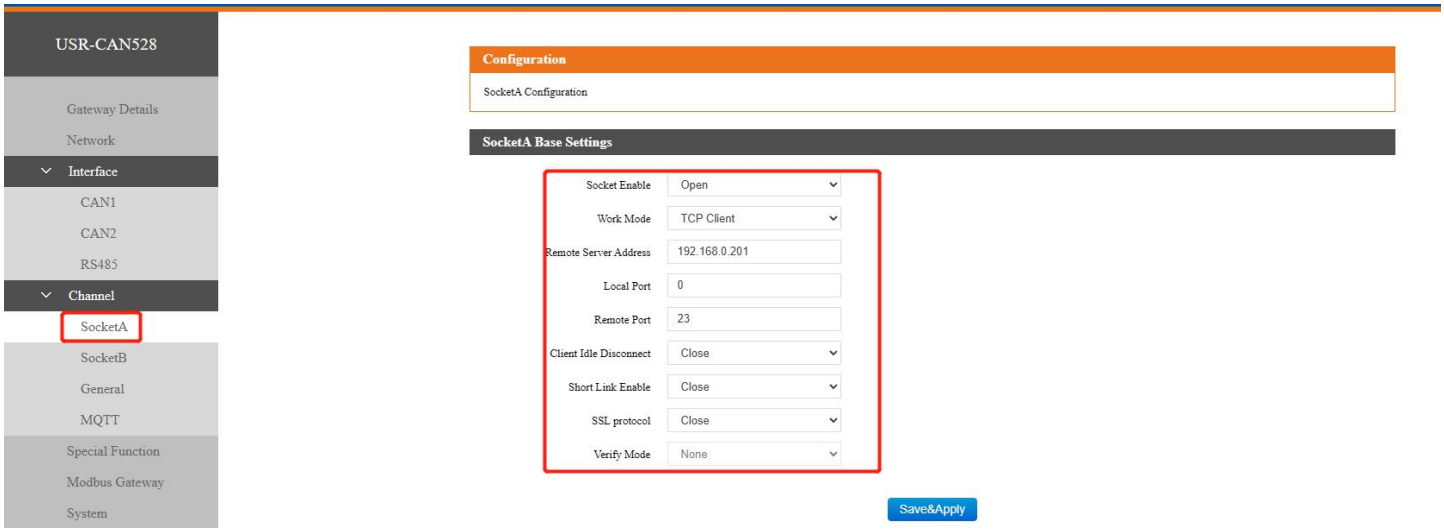


Figure 18. TCP client Settings

(2) Short connection

TCP short connections are primarily designed to conserve server resources, typically deployed in many-to-one scenarios. This mechanism ensures all established connections are valid, eliminating the need for additional filtering mechanisms.

The TCP short connection feature operates in TCP Client mode. When enabled, it automatically disconnects if no data is received from the serial or network port within a set time. By default, this feature is disabled, but the disconnect time can be configured between 2 and 255 seconds, with a default of 3 seconds. The configuration diagram is shown below:

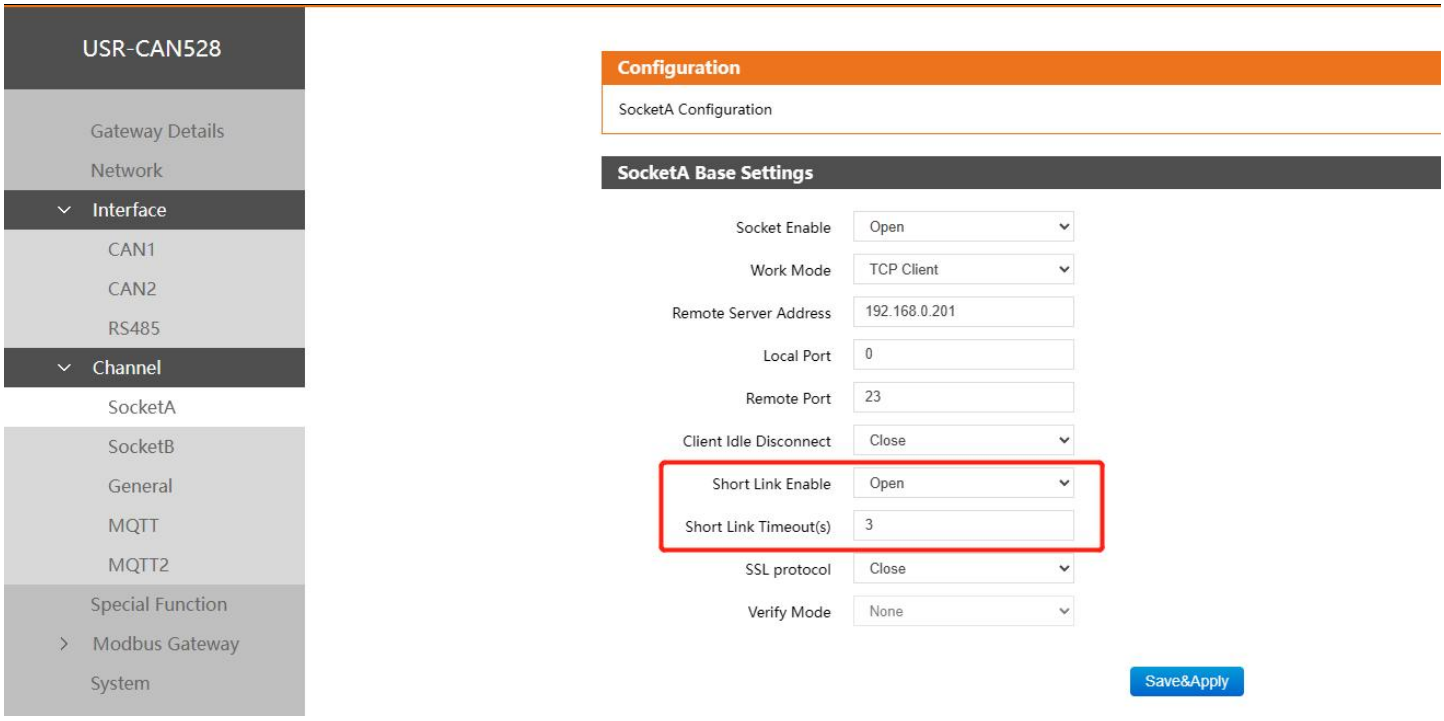


Figure 19. Short connection setup

(3) TCP+SSL encryption

The CAN528 operates in TCP Client mode with SSL encryption support. It offers two SSL versions (TLS 1.0 and TLS 1.2) and provides authentication options including no certificate verification, server certificate verification, and mutual certificate authentication.

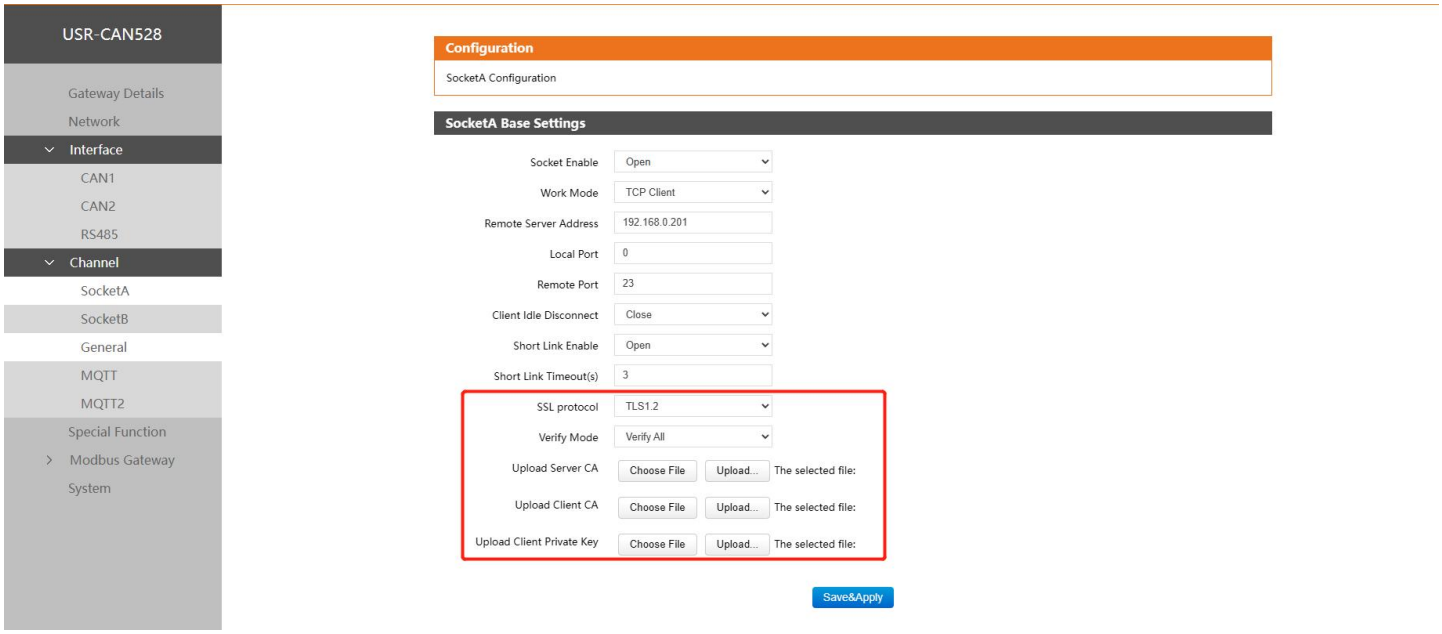


Figure 20. SSL encryption diagram

4.1.2. TCP Server

(1) Description:

A TCP Server is a device that operates in this mode. In TCP Server mode, the CAN528 listens on its local port, accepting connection requests and establishing connections for data communication. Upon receiving data, the CAN528 simultaneously transmits it to all connected client devices. The TCP Server mode also features a KeepAlive function to monitor connection integrity in real time.

This protocol is typically used for communication with TCP clients within a local area network (LAN). It is suitable for scenarios where there is no server in the LAN and multiple computers or mobile devices request data from the server. Similar to TCP Client, it distinguishes between connected and disconnected states to ensure reliable data exchange.

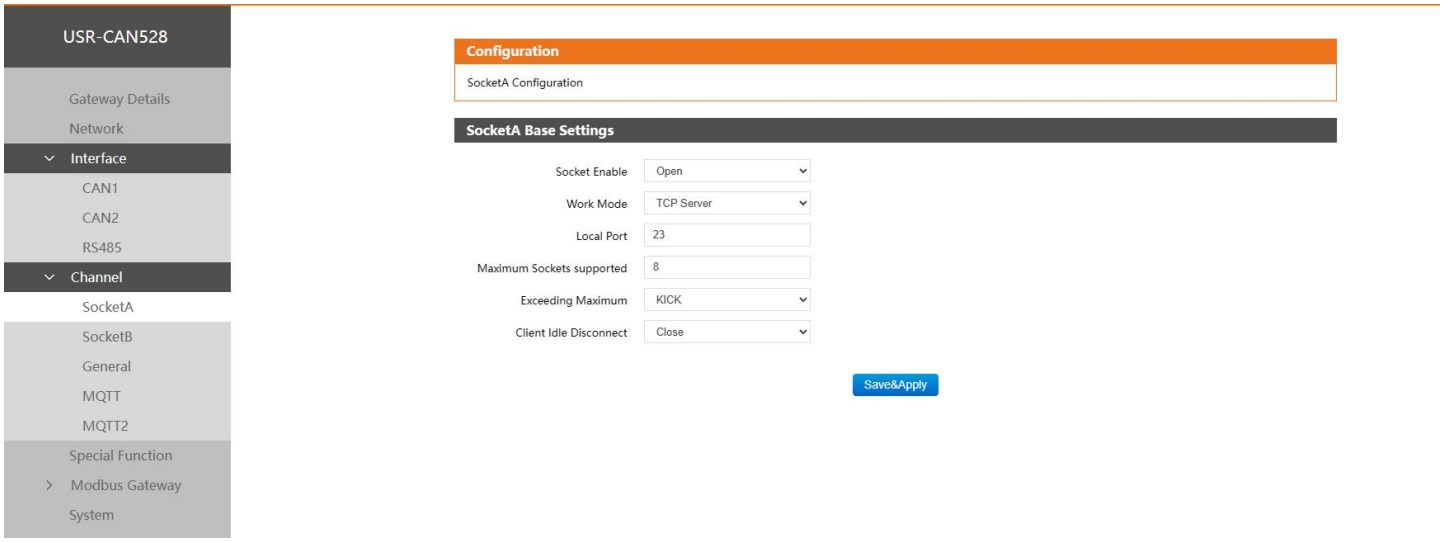


Figure 21. TCP Server Settings

(2) Custom Client connections

The CAN528 supports up to 14 TCP Server connections. The maximum number can be customized to meet customer requirements for optimal usability.

When the number of Client connections exceeds the user-defined limit, new connections will replace old ones by default. Alternatively, you can configure the system to prevent new connections from replacing old ones.

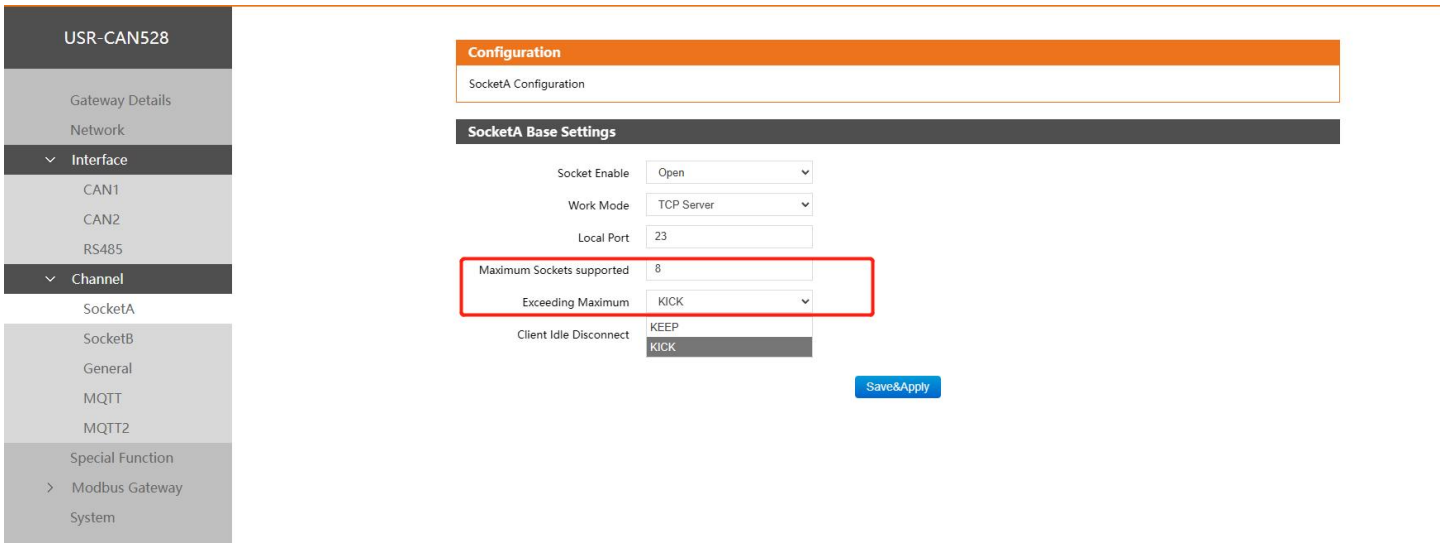


Figure 22. Maximum connection limit

4.1.3. UDP Client

This operational model operates under the UDP protocol. As a connectionless transport protocol, UDP Client provides transaction-oriented, simple, and unreliable data transmission services. It requires no connection establishment or termination, only specifying IP addresses and ports to send data to the recipient. It is typically used in scenarios where packet loss rate is not a concern, data packets are small and transmitted at high frequency, and the data must reach a specified IP address.

1)In UDP Client mode, the CAN port only communicates with the target port of the specified IP address.

2)In this mode, setting the destination address to 255.255.255.255 enables UDP full subnet broadcast functionality while allowing data reception. It supports intra-subnet broadcasting, such as xxx.xxx.xxx.255 broadcast mode.

3)Supports UDP multicast functionality. Multicast enables one-to-many connections between data senders and receivers, allowing multiple recipients to join the same multicast group and share a single IP address. Group members are dynamic, and their joining or leaving does not affect existing members. The valid address range for multicast groups is 224.0.0.2-239.255.255.255.

4)Supports UDP source judgment. When enabled, it allows you to receive data from the entire network segment (e.g., 255.255.255.255), within a specific subnet (e.g., 192.168.0.255), or from a designated IP (e.g., 192.168.0.201) based on configuration.

- The network-wide broadcast does not verify the data source IP address, but only checks if the source port matches the target port. If they match, the network data is transmitted; if not, it is discarded.
- The in-band broadcast checks if the data source port matches the target port and verifies whether the IP is within the same subnet. Data meeting these criteria is output; otherwise, it is discarded.
- For standard UDP communication, the system verifies the source port and IP address. If the data matches the target port and IP address, it is transmitted; otherwise, it is discarded.

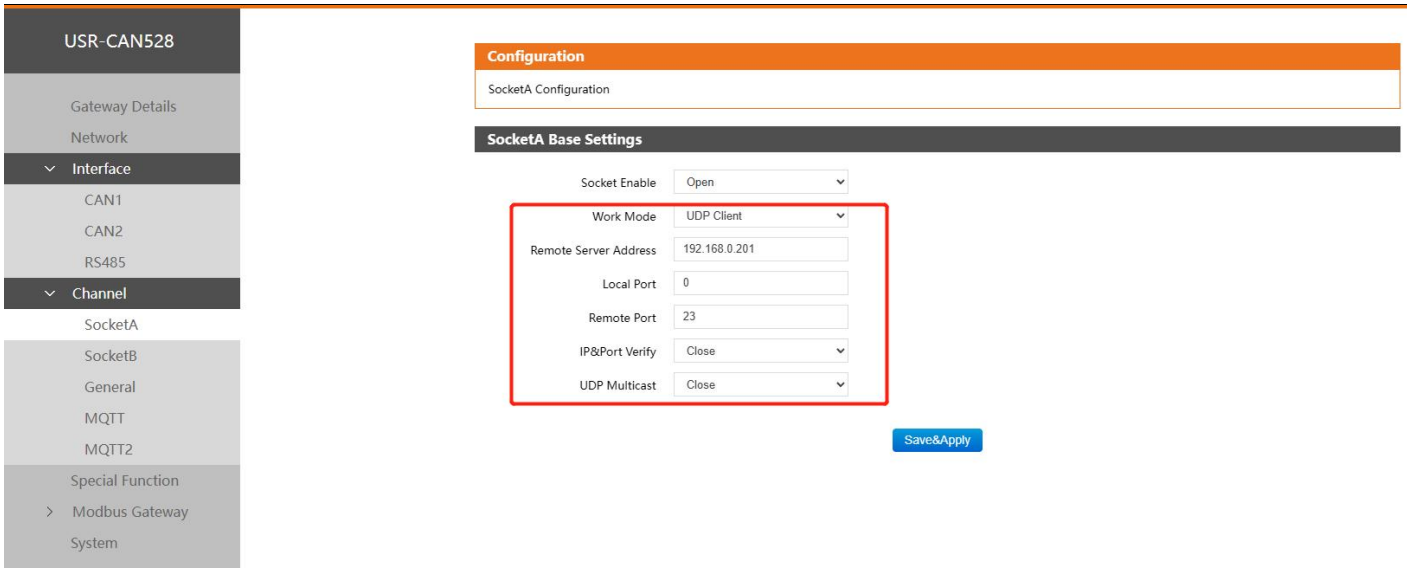


Figure 23. UDP client settings

4.1.4. UDP Server

A UDP Server operates without source IP verification in standard UDP. Upon receiving a UDP packet, it replaces the destination IP with the source IP and port number, then transmits data to the most recently connected IP and port. This mode is typically employed when multiple network devices need to communicate with a module, and TCP is not suitable for high-speed, high-frequency data transfer scenarios.

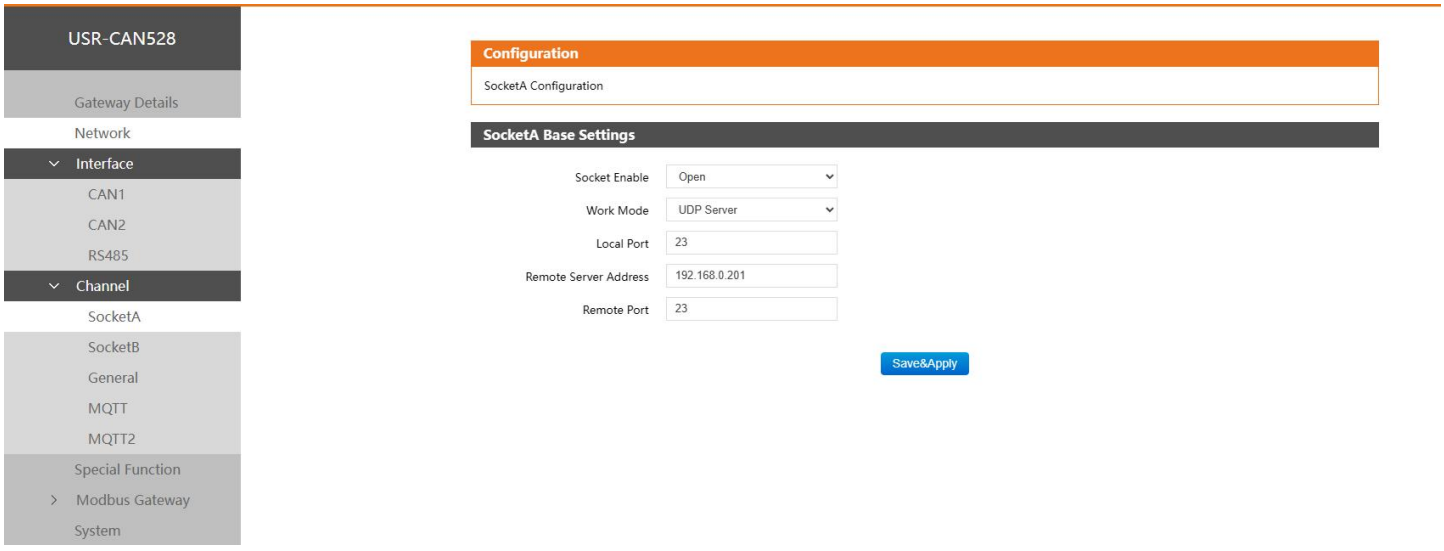


Figure 24. UDP server settings

4.2. MQTT settings

MQTT is a client-server-based message publishing/subscription transmission protocol. Its lightweight, simple, open, and easy-to-implement features make it widely applicable.

The MQTT functionality of CAN528 offers remarkable flexibility, supporting configuration of nearly all protocol-related

connection parameters, as well as publish and subscribe topic parameters.

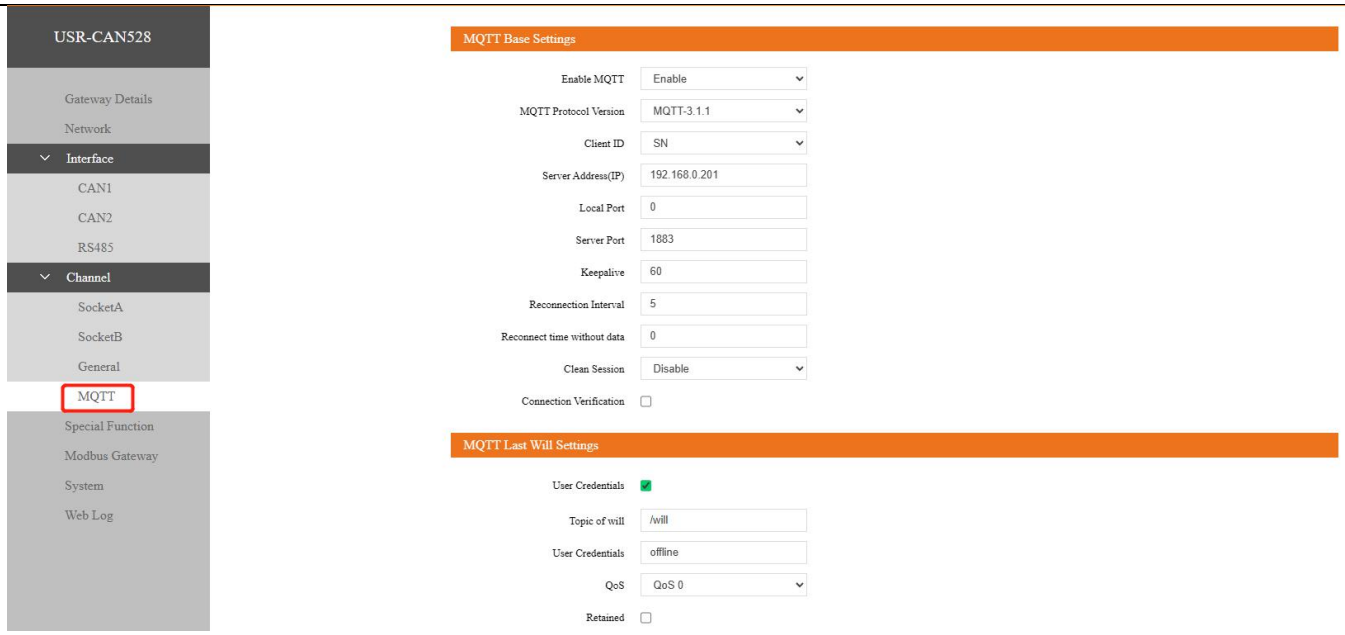
MQTT's publish topic mode supports three configurations: pure broadcast mode (where CAN1, CAN2, and RS485 can broadcast data to all topics), topic distribution mode (using identifiers to deliver data to specific topics), and custom topic publish mode (targeting specific topics via CAN1, CAN2, or RS485).

MQTT's topic mode supports two configurations: pure push mode (where subscribed data is pushed to CAN1, CAN2, and RS485) and push-with-topic mode (where subscribed data is pushed to CAN1, CAN2, and RS485 with the topic).

4.2.1. Basic Configuration

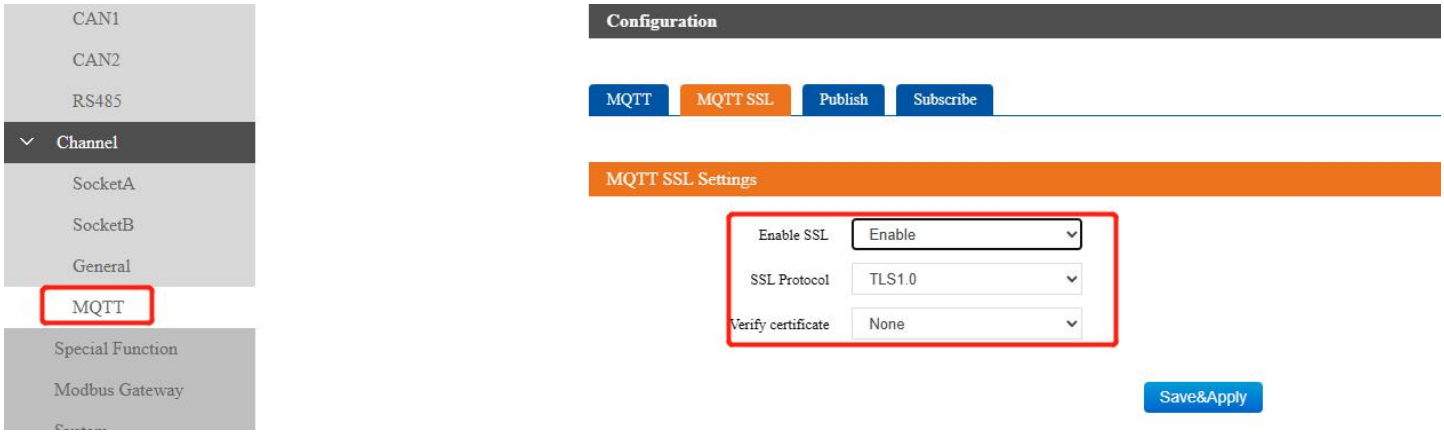
Overview: The device supports MQTT connections for nearly all configuration items, offering high flexibility. It allows reconnection interval configuration to adapt to different MQTT servers. SSL encryption is supported with options for TLS 1.0 and TLS 1.2, and authentication modes include no certificate verification, one-way authentication, and two-way authentication.

- MQTT enabled: You can choose to enable or disable the MQTT Client service
- MQTT protocol version: Devices can choose between MQTT-3-1 and MQTT-3-1-1 versions
- Client ID: MQTT client identifier, supports SN, MAC, and custom
- Server domain (IP): MQTT server domain (IP) address
- Local port number: 0~65535. The port number assigned to the device's MQTT socket. Enter 0 to assign a random local port number.
- Remote port: 0~65535. MQTT server connection port
- Keepalive: 0~65535. MQTT protocol heartbeat time
- Reconnect interval: 1 to 65535. The interval between MQTT disconnections and reconnections.
- No data reconnection time: 0, 60~65535. The device will not receive MQTT protocol frames until the no data time expires, triggering a reconnection to the MQTT server. 0 means disabled.
- Enable session cleanup: A flag for the MQTT protocol connection to control session lifetime
- Connection verification: Enabling this option allows you to set the MQTT connection username and password. Disabling it will reset them to default values.
- Username: MQTT connection username
- Password: Password for MQTT connection
- Testament: MQTT connection flag. When the network connection is closed, the server must publish this testament message. Clients subscribed to this testament topic will receive the set testament.



4.2.2. SSL encryption of MQTT

The CAN528 operates in MQTT mode with SSL encryption support, offering SSL encrypted connections. It supports both TLS 1.0 and TLS 1.2 protocols, and provides authentication modes including no certificate verification, server certificate verification, and mutual certificate verification.



4.2.3. Topic publish

Publishing a topic allows other MQTT clients to subscribe to it and exchange data. Up to 16 topics can be published. Topic publishing supports custom mode, transparent mode, and topic distribution mode. Other detailed parameter configurations are described below.

4.2.3.1. Custom publish

Custom mode is simple and quick to use. Users don't need to configure data publishing parameters on the web page. Just enable the custom mode switch, select the corresponding port number, click save, and restart the device to achieve MQTT data transmission.

Custom data release message format: <top>, <qos>, <retained>, <payload>

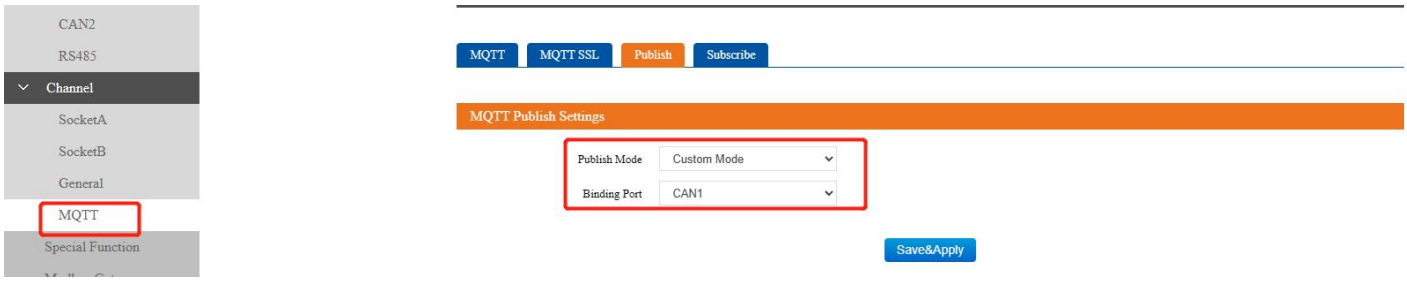
<top>: Publish the topic name.

<qos>: Quality of service for message publishing. Enter 0, 1, or 2. 0: QOS0 (transmitted once at most), 1: QOS1 (transmitted at least once), 2: QOS2 (transmitted only once).

<retained>: Flag for retaining message topics. Set to ON (retains message bit 1) or OFF (retains message bit 0).

<payload>: Publish the topic message content.

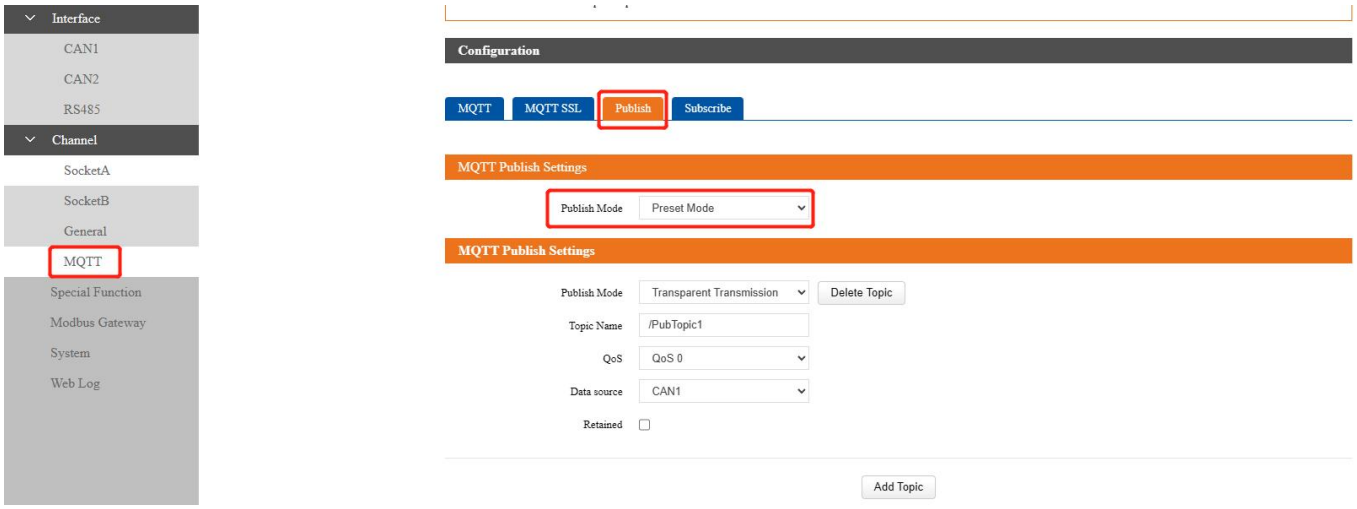
For example: publishtopicname, 0, on, payloaddata



4.2.3.2. Transparent transmission

The device transmits data received from ports to associated topics in a transparent manner, supporting up to 16 topic publications. Each topic can be configured individually and bound to different ports. When data is received on a port, the device pushes the data to the topic bound to that port.

- Transmission mode: Transparent mode
- Title: Set publication title
- QOS: Message quality for the published topic. Options: QOS0 (transmitted once at most), QOS1 (transmitted at least once), QOS2 (transmitted only once)
- Data source: The port bound to this release topic, such as CAN1, CAN2, or RS485
- Retained message: The MQTT message retention flag indicates whether the server stores this application message and its Quality of Service (QoS).



4.2.3.3. Topic distribution

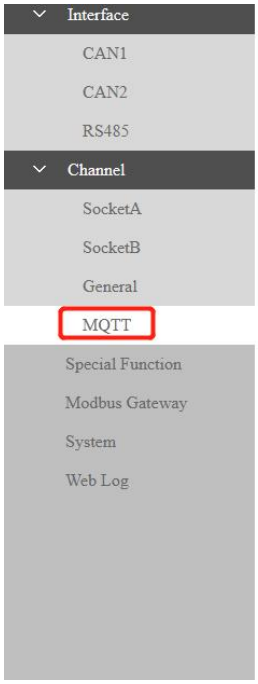
When configuring a topic, you can define its unique identifier and assign it to different device ports. Upon receiving port data, the device will push it to the associated topic based on the identifier. When the topic binds to the port, the device will send the data to the MQTT server. Devices allow multiple topics to share the same

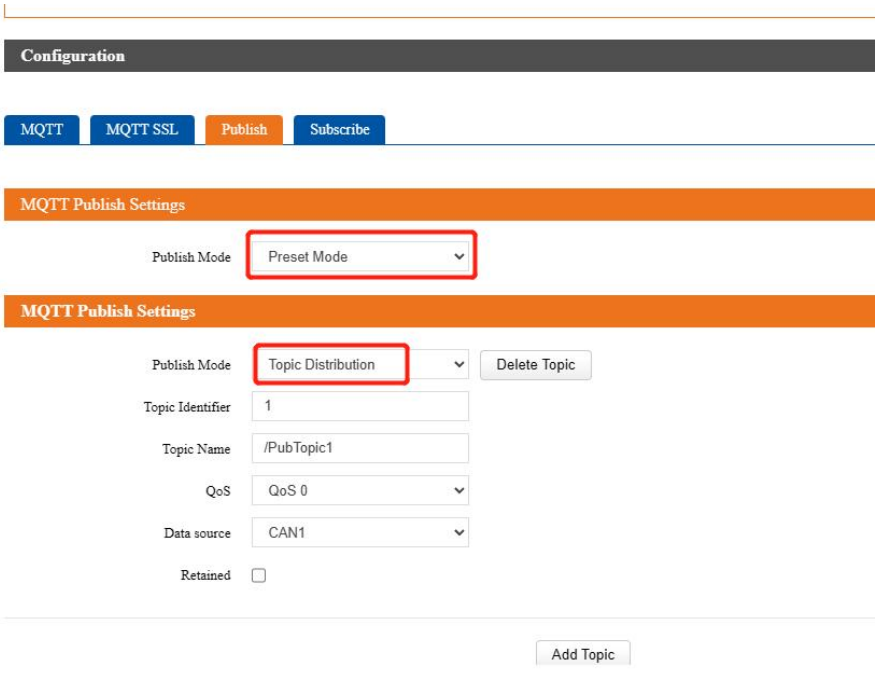
identifier, enabling data to be pushed to all topics sharing the same identifier. The identifier and payload are separated by commas.

Port data sending: <topic_id>, <payload> corresponds to the MQTT topic subscribed to the payload data

Note: In this mode, the CAN conversion mode cannot be set to standard protocol conversion. Standard protocol conversion uses fixed CAN frame formats, which may cause transmission failure.

- Transmission Mode: Theme Distribution Mode
- Theme ID: The identifier for this theme
- Topic name: Publish topic name
- QOS: Message quality for the published topic. Options: QOS0 (transmitted once at most), QOS1 (transmitted at least once), QOS2 (transmitted only once)
- Data source: The port bound to this release topic, such as CAN1, CAN2, or RS485
- Retained message: The MQTT message retention flag indicates whether the server stores this application message and its Quality of Service (QoS).





4.2.4. Topic Subscribe

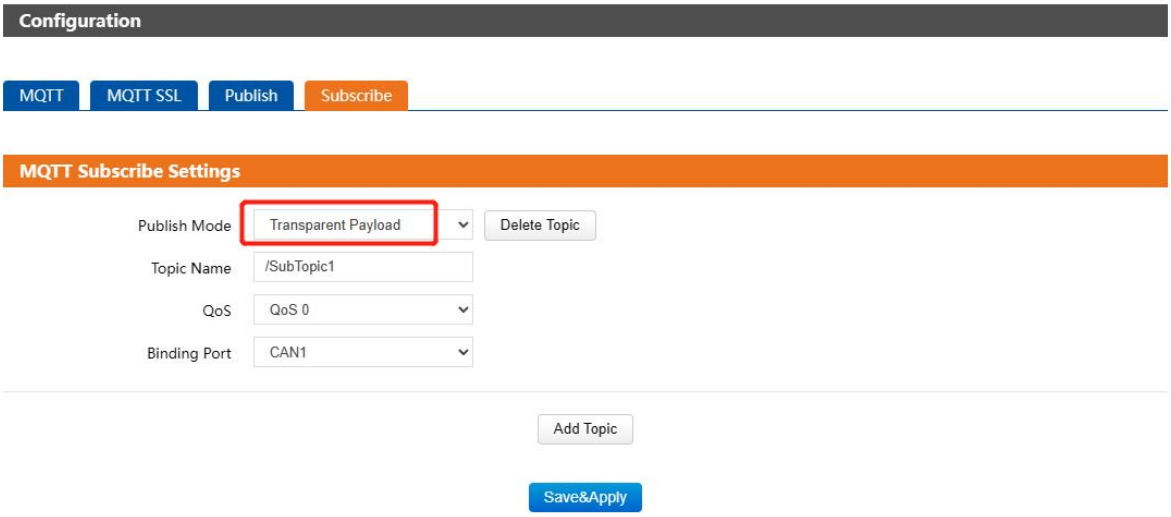
Topic subscription allows two or more clients to communicate by subscribing to topics from other clients, supporting up to 16 topics. It supports transparent mode and topic-based distribution mode. Detailed parameter configurations are described below.

4.2.4.1. Transparent mode

Description: Transmits received data to associated topics in transparent mode. Supports up to 16 topic publications, each configurable and bindable to different ports. Topics send data to their bound ports upon receiving it.

- Transmission mode: Transparent mode

- Subject name: Subscription subject name
- QOS: Quality of subscription topic messages. Optional maximum transmission: QOS0; minimum transmission: QOS1; single transmission: QOS2
- Port: The port number for this subscription topic. Options include CAN1, CAN2, and RS485.

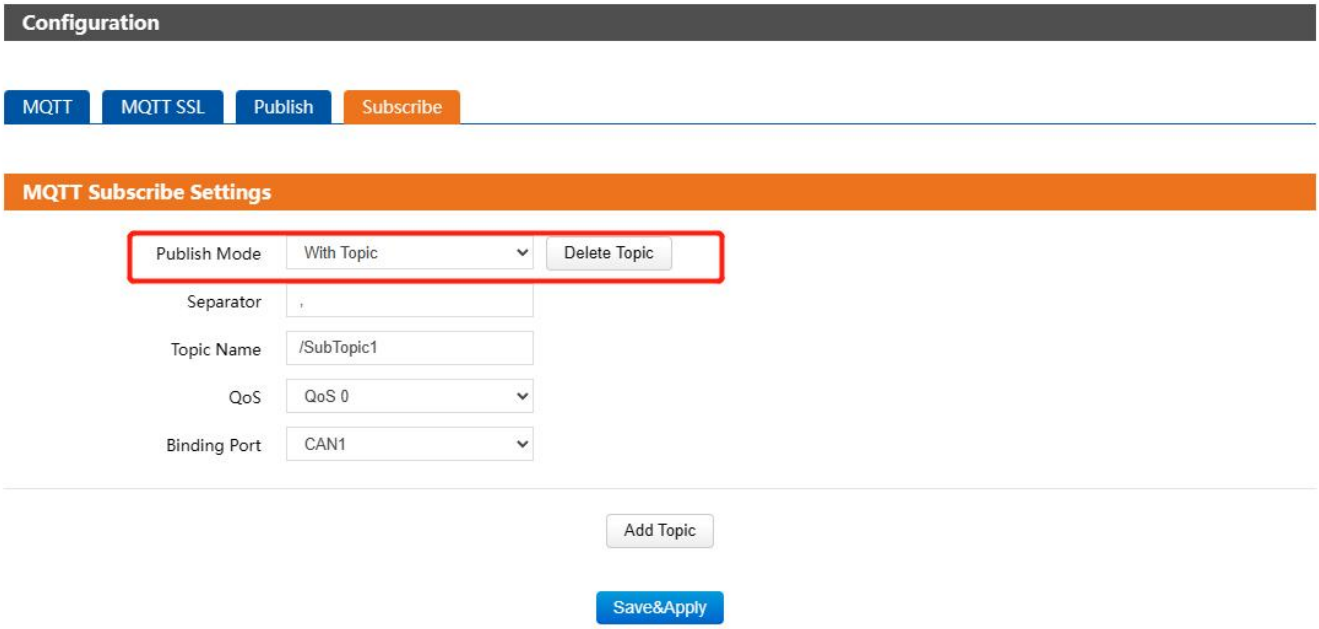


4.2.4.2. Topic-based distribution mode

Topic-based delivery adds a subject to the message received by the subscriber and delivers it with the payload. The subject and data are separated by a delimiter. Each topic can be configured individually and bound to different serial ports. After receiving data, the topic sends the data to the bound port.

MQTT transmits payload data normally, with the default format received by the bound port being: <topic identifier><separator><payload>.

- Transmission mode: Themed distribution mode
- Separator: The separator between the subject and application data
- Subject name: Subscription subject name
- QOS: Quality of subscription topic messages. Optional maximum transmission: QOS0; minimum transmission: QOS1; single transmission: QOS2
- Port: The port number for this subscription topic. Options include CAN1, CAN2, and RS485.



5. Advanced Function

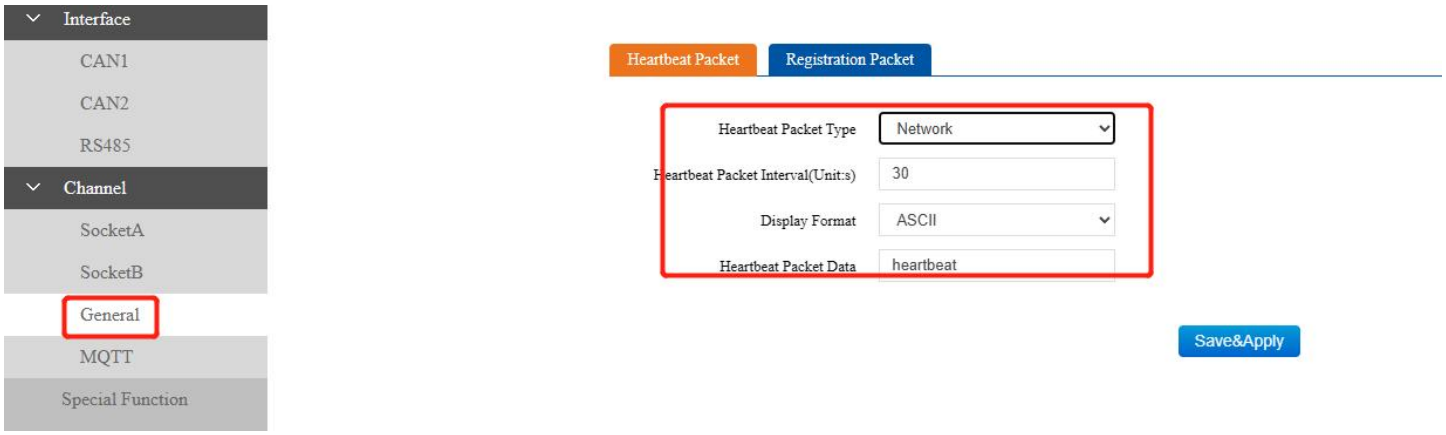
5.1. Heartbeat packet

In network transparent mode, users can configure CAN528 to transmit heartbeat packets. These packets are categorized into network heartbeat, serial port heartbeat, and CAN port heartbeat, with only one type active at any given time. CAN1 and CAN2 can operate simultaneously.

The heartbeat packet supports both HEX and ASCII formats. The heartbeat period can be set to 1 to 65535 seconds.

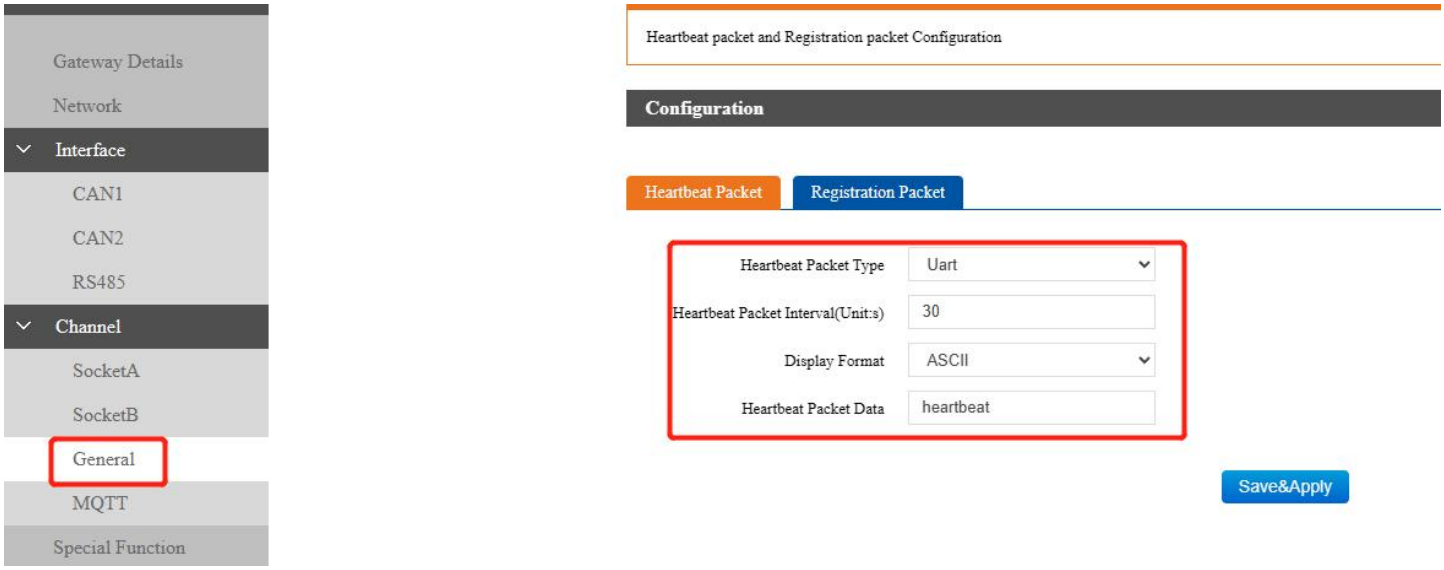
(1) Network heartbeat packets:

When the network client has no data, it periodically sends requests to the network server to maintain connection. This feature is only available in TCP Client and UDP Client modes.



(2) Serial heartbeat packet:

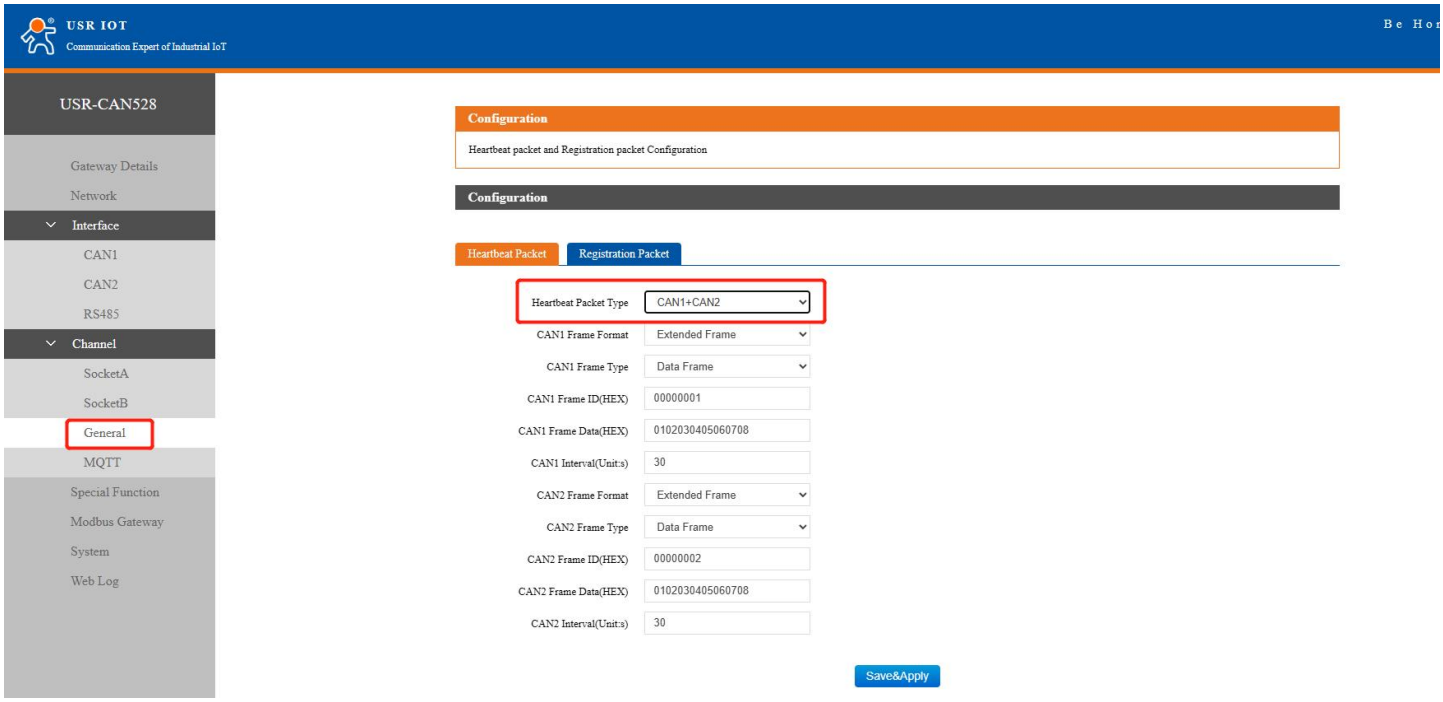
It can be used as a fixed query command, transmitted to the serial port via heartbeat packets. The maximum size of a serial port heartbeat packet is 40 bytes.



(3) CAN port heartbeat packet:

It can be configured as a fixed query command and transmitted to CAN via heartbeat packets. The content must comply with CAN format, where CAN frame format, frame type, and frame ID are all configurable.

It can be applied to CAN1 or CAN2 individually, or both CAN1 and CAN2 simultaneously.



5.2. Registration packet

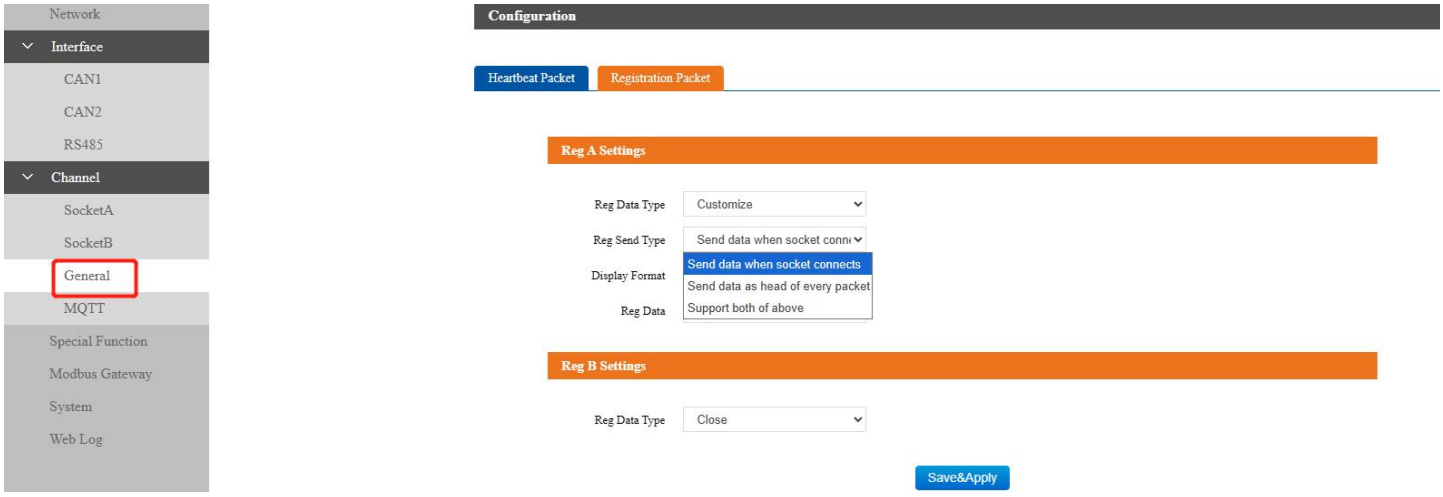
In network transparent mode, users can configure the registration package function. The registration package can serve as an identifier for the server to recognize the data source device or as a password to obtain server function authorization.

The registration packet can be configured to either send the registration packet when establishing a

connection with the server, carry the registration packet during transmission, or activate both simultaneously. Connection-based transmission refers to sending the registration packet during TCP or UDP connection establishment, while packet-carrying transmission involves appending the registration packet data at the front of each data packet as a single unit. The registration packet may contain either a MAC address or custom registration data, with the custom registration data being limited to a maximum of 40 bytes.

The registration package applies only to TCP client and UDP client modes. It is not available for TCP server and UDP server.

The registration packets for socketA and socketB are independent.

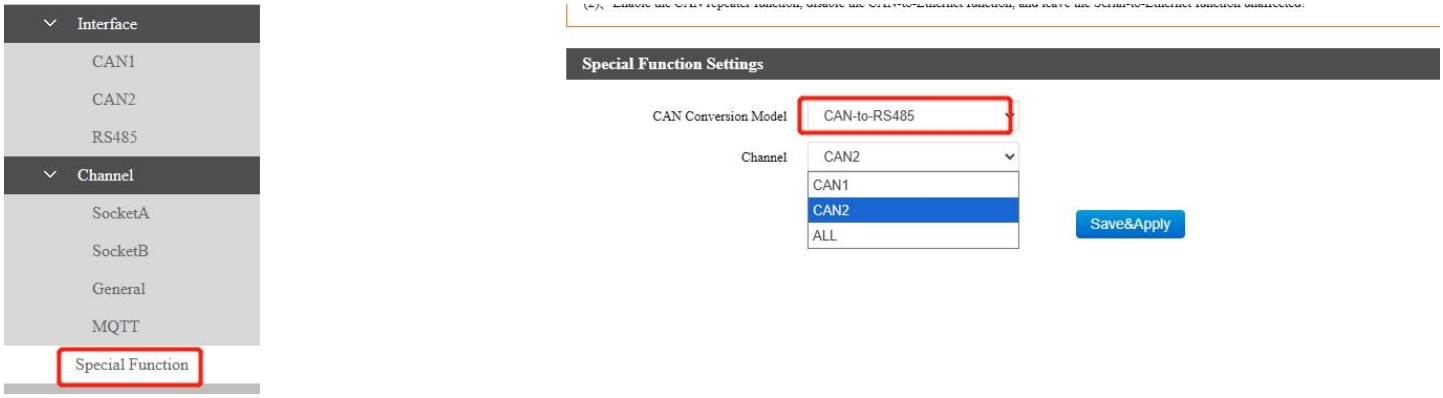


5.3. CAN to RS485 conversion

The CAN528 supports CAN-to-serial port conversion. When this function is activated, the CAN-to-network and serial-to-network functions for the corresponding CAN port will be disabled.

The setup is straightforward: simply enable CAN-to-serial port conversion, select the target CAN port, configure its parameters and the serial port settings, and your communication is ready to go.

The basic configuration, data filtering, and CAN conversion modes in CAN port parameters can all be applied to the CAN-to-serial port.



5.4. CAN repeater

The CAN relay function is mainly to solve the contradiction of transmission speed and transmission distance of CAN network and to expand the number of communication network.

The CAN528 integrates CAN relay functionality. Once the CAN parameters are configured, it only requires power supply to operate. It can convert baud rates when different baud rates are configured.

In relay mode, all functions including CAN-to-network, CAN-to-serial, and serial-to-Ethernet conversion will be disabled. Data received by CAN1 will be transmitted directly through CAN2, and vice versa.

6. Modbus Gateway

This device employs a straightforward approach to enable interoperability between CAN and Modbus devices. It allows for both data transmission from Modbus devices to the CAN bus for centralized management and control of Modbus device data via the CAN bus.

The device can function as either a Modbus master or slave. It supports CAN-to-Modbus TCP (master/slave) and CAN-to-Modbus RTU (master/slave) conversion.

When the Modbus gateway function is enabled, data transmission is limited to the CAN and Modbus devices, and all functions related to data transmission are disabled.

6.1. Basic Settings of Modbus Gateway

- When the CAN528 is configured in Modbus TCP master mode, the TCP client must be enabled, and the slave's IP address and port number should be configured.
- When CAN528 operates in Modbus TCP slave mode, the TCP server must be enabled and the port number configured for CAN528 as the exclusive slave. The device contains 2,400 preconfigured registers, allowing external master devices to write to these registers at any time within the range of 0 to 2,399.
- When the CAN528 is configured in Modbus RTU master mode, it can support up to 255 Modbus RTU slave devices.

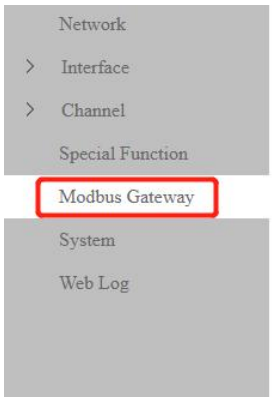
- When the CAN528 is configured in Modbus RTU slave mode, it contains 2400 preconfigured registers. The external master device can write to these registers at any time, with a range of 0 to 2399.

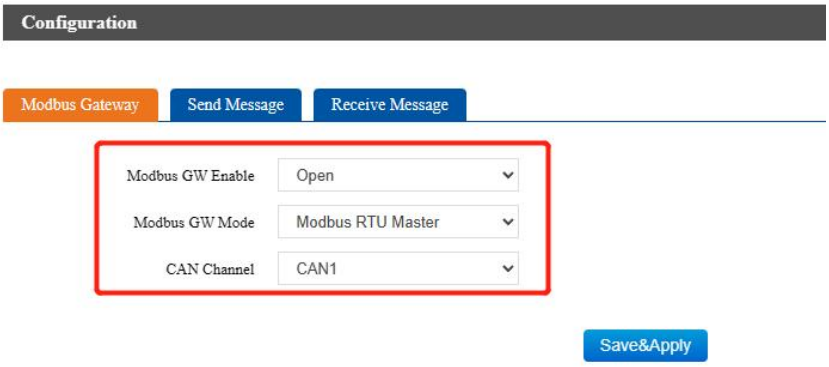
Steps:

Enable Modbus gateway and select its operating mode

Configure CAN basic parameters and corresponding socket or serial port parameters

Configure sending and receiving messages on demand



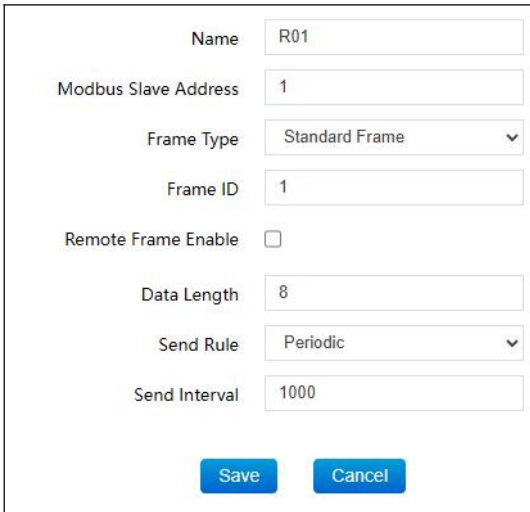
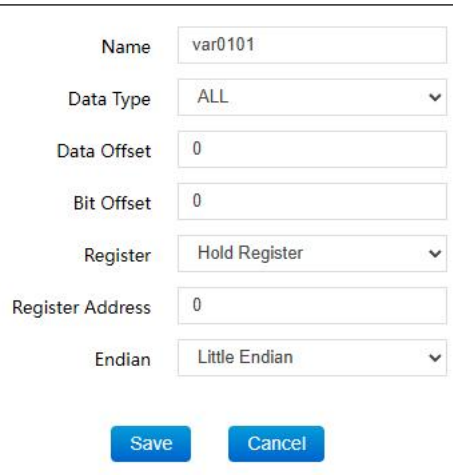


6.2. Send Massages

By configuring device messages, the system actively collects Modbus data and maps it to corresponding positions in CAN (FD) data frames, then transmits them as CAN frames according to predefined rules. It supports up to 64 message configurations, with each message allowing up to 32 variable data entries mapped to specific CAN frame positions.

You can either configure and send messages directly in the web page or use a.csv file for importing and exporting point tables, which is suitable for scenarios with many collection points and mapped points.

You can add, delete, or edit configured messages and variables through the web page.

	
message parameter	variable parameter

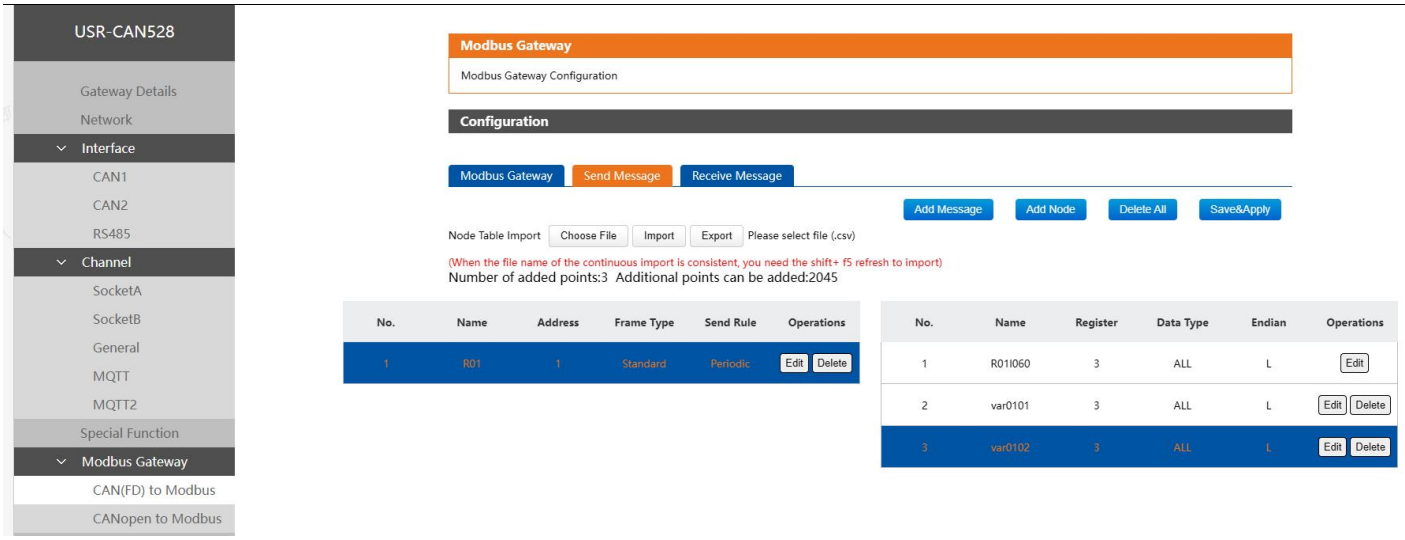


Table 1. Parameter information

parameter	meaning	scope	Corresponding to the.csv document
message parameter			
Message name	The name of the message to be sent is not mapped to CAN data and is used for mnemonic purposes only.	Supports English or numbers	English or numbers
slave address	When the device acts as the master station, enter the address of the peer slave. When the device acts as a slave, enter its address as a slave.	1~255 Only supports a single slave	1~255
frame type	Frame type of the transmitted message	standard frame extended frame	Standard Frame--A Extend Frame--B
frame ID	Frame ID of the transmitted message	Hexadecimal, hex format 0~07FF (Standard Frame) 0~1FFFFFFF (Extended Frame)	0~07FF (Standard Frame) 0~1FFFFFFF (Extended Frame)
Supports remote frames	Confirm whether the frame is a remote frame. This option is invalid when the CAN type is selected as CAN FD.	Yes/No	Yes--1 No--0
DL	Send frame data segment length: up to 8 bytes for CAN frames and up to 64 bytes for CAN FD frames. Note: CAN FD frames must be set to a length that can be encoded by DLC.	0~8 0~8 、 12 、 16 、 20 、 24、32、48、64	0~8 0~8、12、16、20、24、32、48、64
Send rules	Mode to trigger device to send CAN messages. Periodic send: Report at the set interval	periodic transmission Send changes	Send cycle-1 Change sent-2

	Change report: Report any data point change in this group of messages One-time send: Send once after establishing the connection Frame ID Trigger: Trigger transmission upon receiving the specified frame ID	Send once frame ID trigger	Send once--3 Frame ID Trigger--4
send frame ID	Active in frame ID trigger mode, refers to the frame ID of the CAN frame that triggers this message transmission.	Hexadecimal, hex format 0~07FF (Standard Frame) 0~1FFFFFFF (Extended Frame)	0~07FF (Standard Frame) 0~1FFFFFFF (Extended Frame)
Send frame type	Active in frame ID trigger mode, indicating the frame type of the CAN frame that triggers this message transmission.	standard frame extended frame	Standard Frame--A Extend Frame--B
Send cycle	When the sending rule is periodic, set the periodic sending time When the sending rule is change-based, this sets the cycle for checking Modbus data changes. Data is sent if changes occur within the cycle, and not sent if no changes occur. The wait time for a single send when the sending rule is set to single send	0~65535ms	0~65535ms
variable parameter			
name	The variable name is not mapped to CAN data and is used for mnemonic purposes only.	Supports English or numbers	English or numbers
data type	The mapped data size. ALL: Full frame data BIT: bit BYTE: 1 byte WORD: 2 bytes Word: 4 bytes QWORD: 8 bytes	ALL BIT BYTE WORD DWORD QWORD	ALL BIT BYTE WORD DWORD QWORD
byte offset	Select which byte to start from in the CAN message data segment to sequentially map Modbus register data to the CAN message data segment. When the operation size is ALL, the offset is invalid. Note: In CAN mode, if the setting is greater than 8, 8 will	CAN：0~8 CANFD：0~64	CAN：0~8 CANFD：0~64

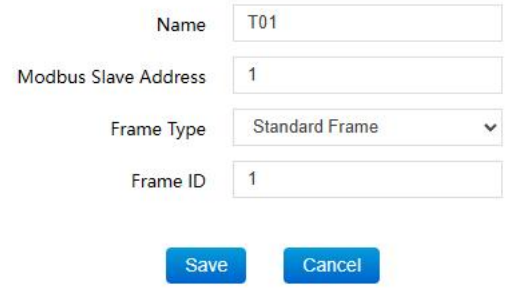
	take effect.		
bit offset	Select the starting byte and bit from the CAN message data segment to sequentially map Modbus register data to the CAN message data segment. This only applies when the data type bit BIT is active.	0~8	0~8
Register type	Modbus register type When the data type is BIT, 01 02 is supported When the data type is BYTE, WORD, DWORD, or QWORD, 03 04 is supported.	coil register (01) Discrete register (02) Keep register (03) Input register (04)	01 02 03 04
register address	The starting address of the message data in the device or Modbus slave register	0~65534	0~65534
byte order	Modbus data storage mode	main aspects small end	main aspects --B small end --S

6.3. Receive Messages

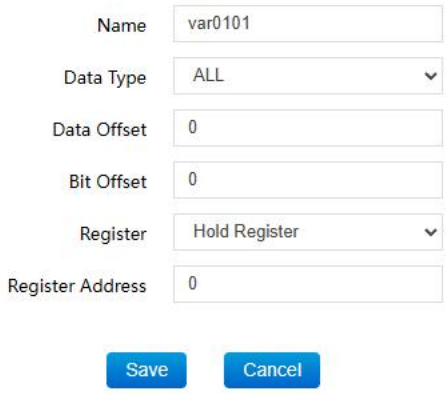
By configuring device packet reception, CAN (FD) data frames can be used to write required data segments from CAN packets into Modbus slave registers. The system supports up to 64 configured reception packets, with each packet allowing up to 32 variable data entries.

You can configure message reception directly in the web page or use.csv files for point table import and export, which is suitable for scenarios with many collection points and mapped points.

You can add, delete, or edit configured messages and variables through the web page.



message parameter



variable parameter

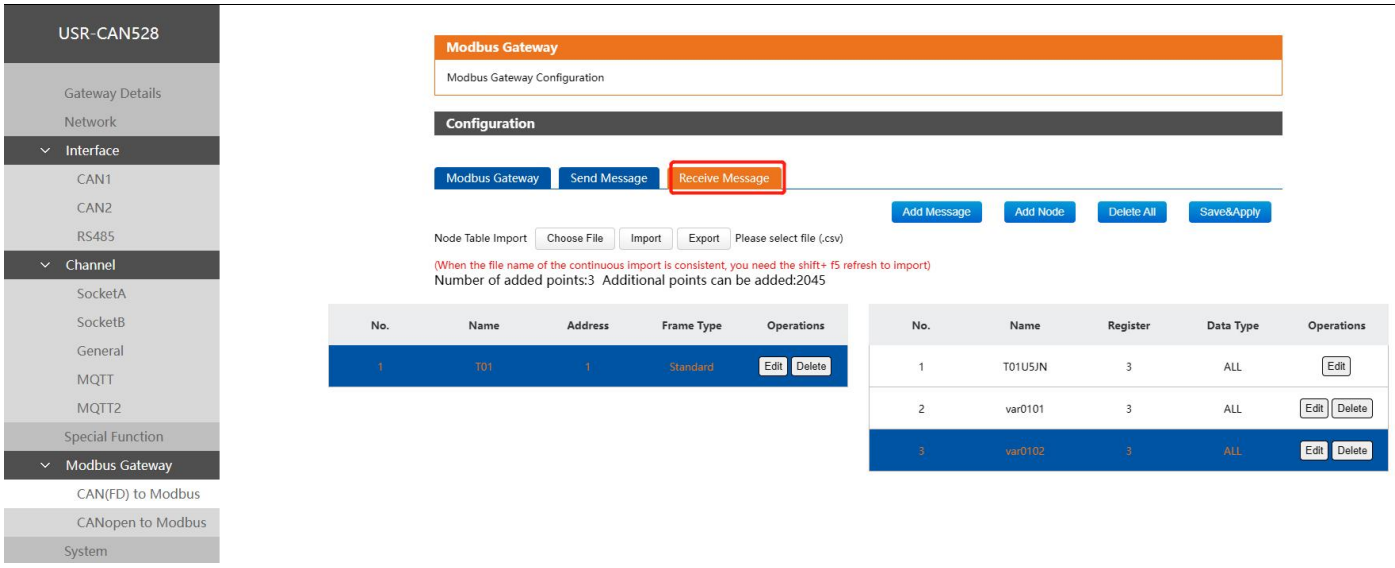


Table 2. Parameter information

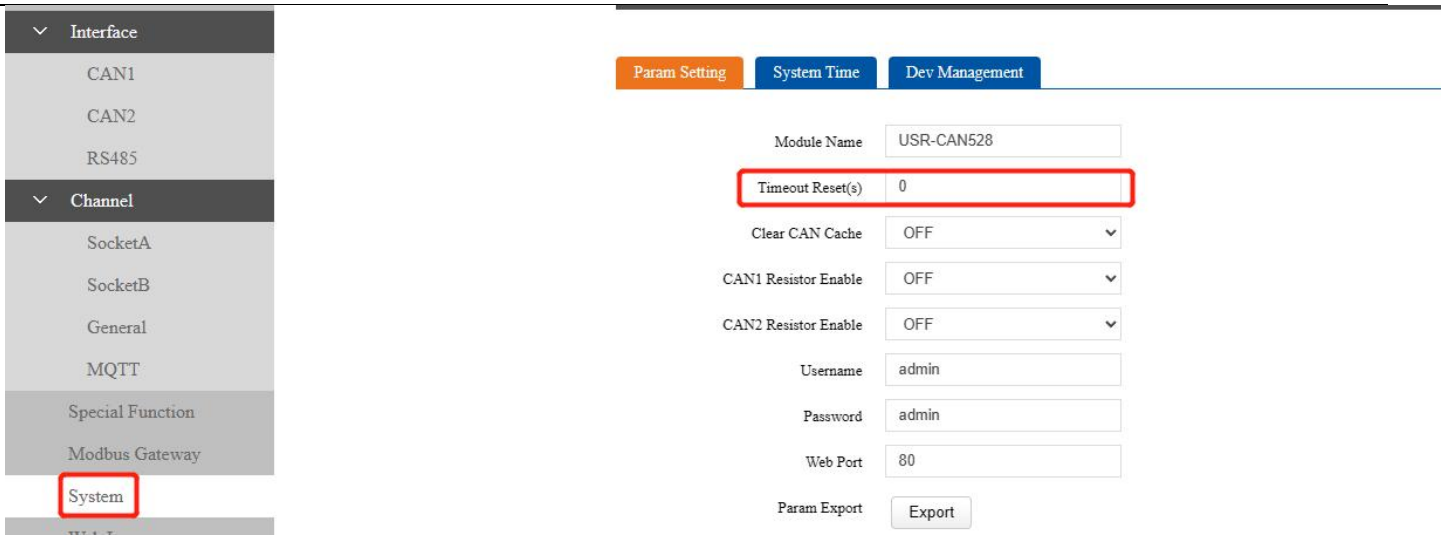
parameter	meaning	scope	Corresponding to the.csv document
message parameter			
name	The name of the received message is not mapped to CAN data and is used for mnemonic purposes only.	Supports English or numbers	English or numbers
slave address	When the device acts as the master station, enter the address of the peer slave. When the device acts as a slave, enter its address as a slave.	1~255 Only supports a single slave	1~255
frame type	frame type of received message	standard frame extended frame	Standard Frame--A Extend Frame--B
frame ID	Frame ID of received packet	Hexadecimal, hex format 0~07FF (Standard Frame) 0~1FFFFFFF (Extended Frame)	0~07FF (Standard Frame) 0~1FFFFFFF (Extended Frame)
variable parameter			
name	The variable name is not mapped to CAN data and is used for mnemonic purposes only.	Supports English or numbers	English or numbers
data type	The mapped data size. ALL: Full frame data BIT: bit BYTE: 1 byte WORD: 2 bytes	ALL BIT BYTE WORD DWORD	ALL BIT BYTE WORD DWORD

	Word: 4 bytes QWORD: 8 bytes	QWORD	QWORD
offset	Select the starting byte of the CAN message data segment and sequentially map the received data segments to the Modbus slave register. The offset is invalid when the operation size is ALL. Note: In CAN mode, if the setting is greater than 8, 8 will take effect.	CAN：0~8 CANFD：0~64	CAN：0~8 CANFD：0~64
bit offset	Select the starting byte and bit from the CAN message data segment to sequentially map the received CAN message to the Modbus slave register. This only applies when the data type bit BIT is active.	0~8	0~8
Register type	Modbus register type Support coil register when the data type is BIT Supports keeping registers when the data type is BYTE, WORD, DWORD, or QWORD.	Coil registers (01,05,0F) Maintain registers (03,06,10)	01 03
register address	The starting address of the message data in the device or Modbus slave register	0~65534	0~65534

7. System

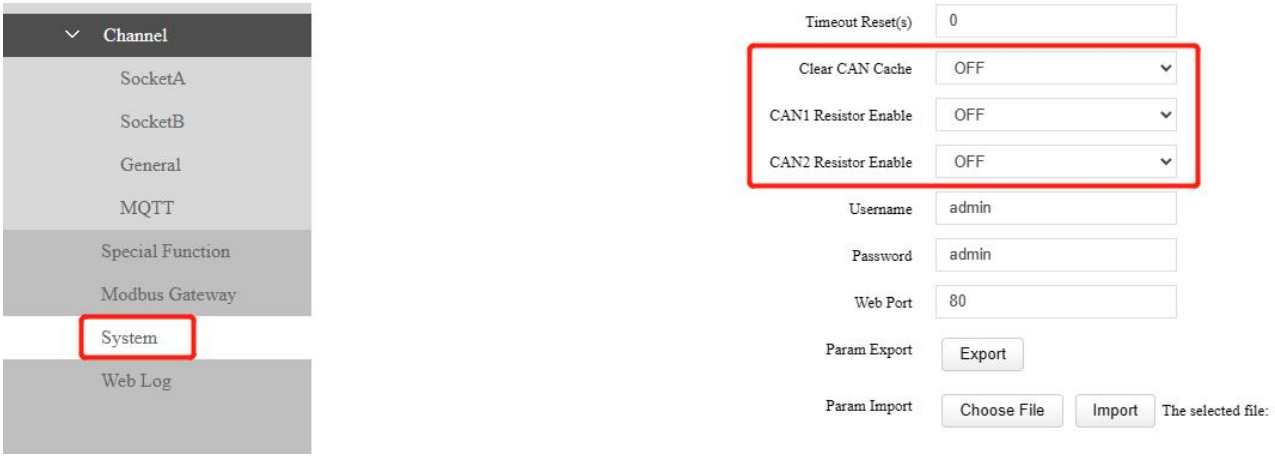
7.1. Timeout restart

The timeout restart function ensures long-term stable operation of the CAN528. When the network port fails to receive data for an extended period or the network remains inactive, the CAN528 will restart after a preset timeout to prevent communication disruptions. The default timeout duration ranges from 60 to 65535 seconds, with a default value of 3600 seconds. If the timeout is set to less than 60 seconds, the function is disabled by default.



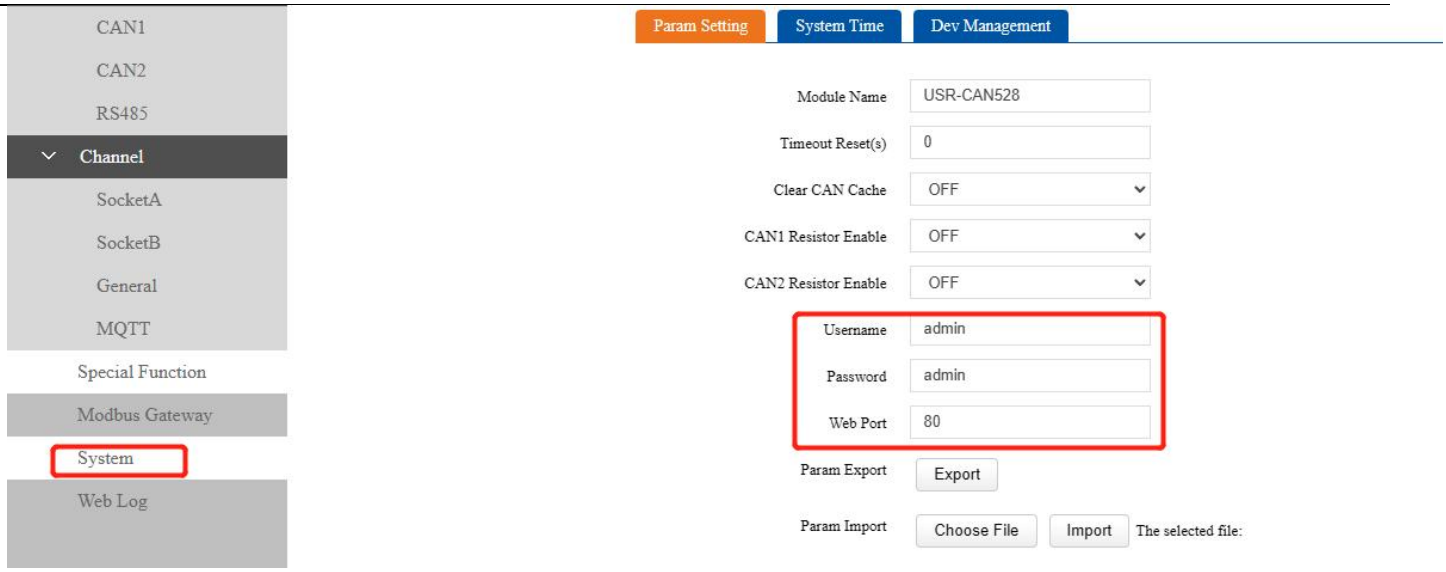
7.2. CAN cache clearing

When a TCP connection is not established, data received through the CAN port is temporarily stored in a buffer with a maximum capacity of 200 frames. After the TCP connection is established, the CAN port can choose whether to clear the buffer data based on client requirements. By default, the buffer data remains uncleaned. However, this feature becomes disabled when the short connection function is enabled in TCP Client mode. The configuration diagram is shown below:



7.3. Login password

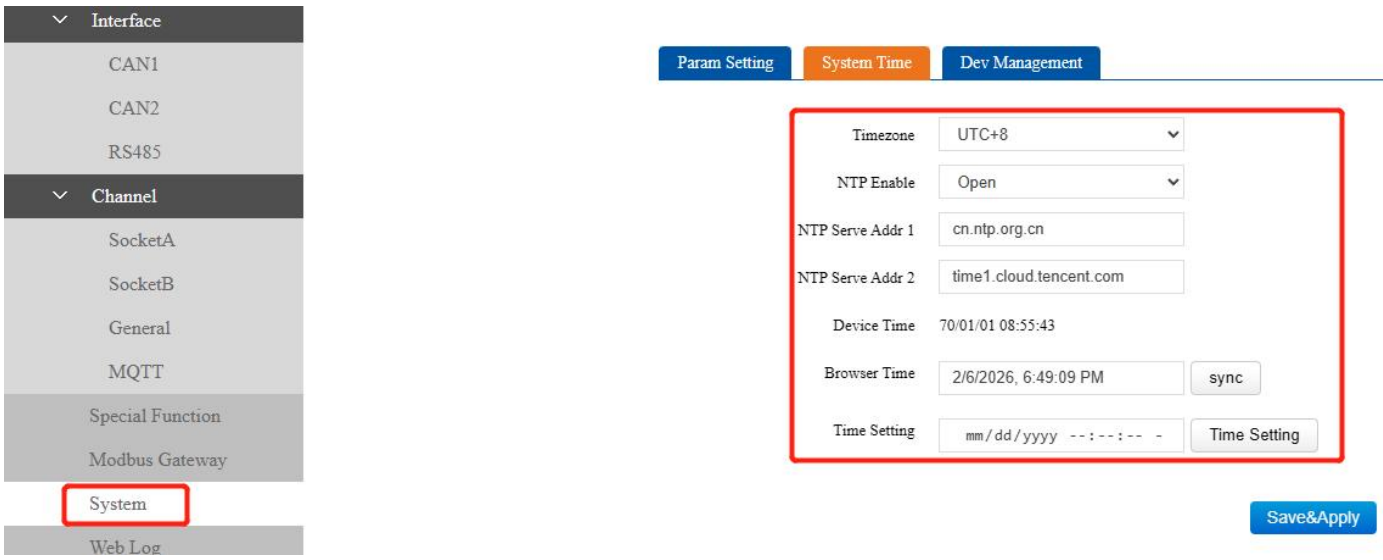
Users can set the username, password and port number for web login on this interface.



7.4. Time Settings

Get NTP time from the NTP server. This feature primarily serves edge computing and other time-related functions.

- NTP time zone: Configure NTP time zone. Default time zone: UTC+8
- NTP enable: Enable via web configuration. Disabled by default
- NTP server address: Configure the NTP server address
- Browser time: Synchronize the current browser time



7.5. Reset

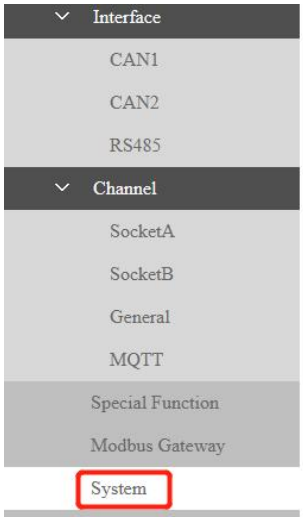
Reset: Reset CAN528 to factory settings

Hardware factory reset: The module supports hardware factory reset. After powering on, press the Reload button, hold it for 3-15 seconds, then release to restore factory settings.

Restore factory settings: You can restore factory settings through the web.

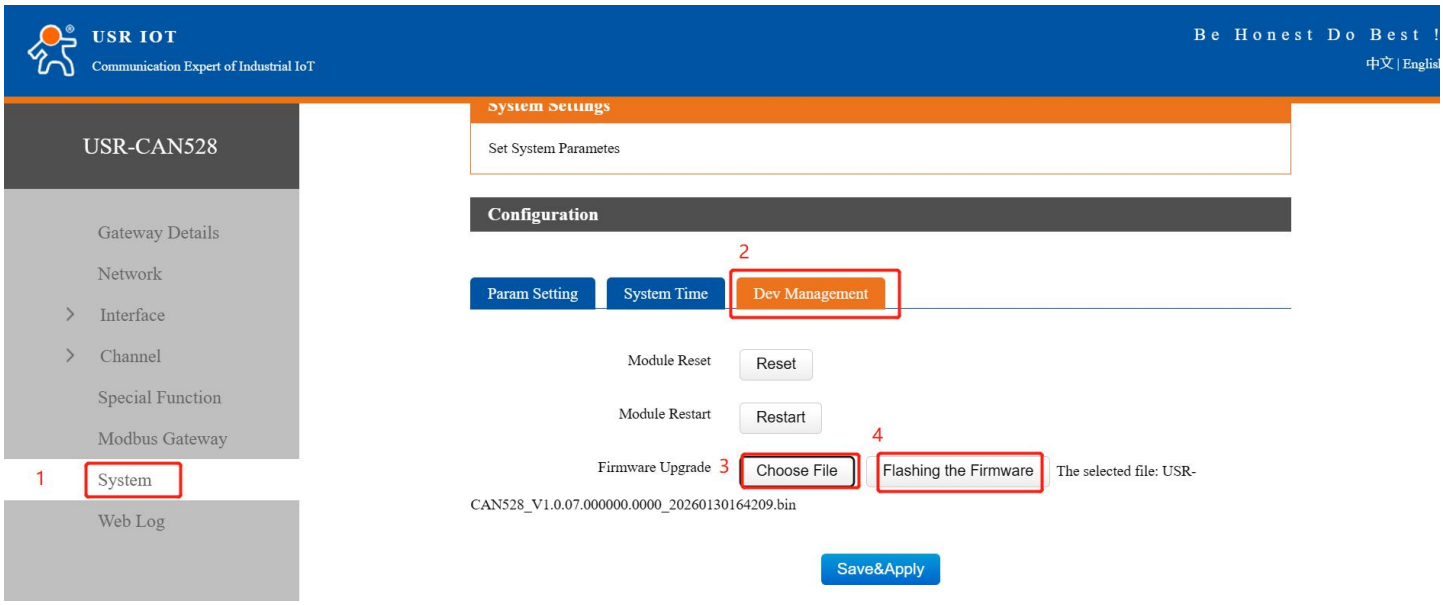
AT command to reset factory settings: In AT command mode, send AT+CLEAR followed by a return character.

The device will reset to factory settings upon receiving a valid +OK response.



7.6. Firmware upgrading

System -- Dev Management -- Choose file -- Flashing firmware



8. Contact Us

Jinan USR IOT Technology Limited

Address : Floor 12 and 13, CEIBS Alumni Industrial Building, No. 3 Road of Maolingshan, Lixia District, Jinan, Shandong, China

Official website: https://www.pusr.com

Official shop: https://shop.usriot.com

Technical support: http://h.usriot.com/

Email : sales@usriot.com

Tel : +86-531-88826739

Fax : +86-531-88826739-808

9. Disclaimer

The information in this document provided in connection with Jinan USR IoT technology ltd. and/or its affiliates' products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of USR IoT products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, USR IoT AND/OR ITS AFFILIATES ASSUME NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL USR IoT AND/OR ITS AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF USR IoT AND/OR ITS AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. USR IoT and/or its affiliates make no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. USR IoT and/or its affiliates do not make any commitment to update the information contained in this document.

Your Trustworthy Smart IOT Partner

PUSR®
www.pusr.com

Official Website: www.pusr.com

Official Shop: shop.usriot.com

Technical Support: h.usriot.com

Inquiry Email: inquiry@usriot.com

Skype & WhatsApp: +86 13405313834

Click to view more: Product Catalog & Facebook & Youtube