PUSR®
www.pusr.com

Communication Expert of
Industrial IOT

CANFD To Ethernet

USR-CAN315

User Manual

Be Honest & Do Best

Your Trustworthy Smart Industrial IoT Partner

Functional characteristics

- CAN (FD) and Ethernet bidirectional conversion
- Supports CANFD protocol and is compatible with CAN2.0a and CAN2.0b standard protocols
- Support transparent conversion, transparent band id conversion, standard protocol conversion, modbus conversion, and custom protocol conversion
- Support tcp server, tcp client, UDP server, UDP client
- Supports only receiving extended frames, only receiving standard frames, and receiving custom frame id
- Support 32 sets of custom frame id filtering to avoid data interference
- Wide bandwidth range, arbitration domain baud rate: 5k~1mbps; data domain baud rate 100k~5mbps
- Supports custom baud rates
- Support heartbeat packets: Network heartbeat packets, CAN port heartbeat packets
- Support registration package: Connection sending, data carrying, full registration
- Support normal, only listen and loop three working modes
- Support CAN(FD) To Modbus TCP（Master/Slave）
- Supports 64 transmission messages and 64 reception messages
- Support upper computer parameter configuration
- Support network at command configuration
- Support the upgrade of firmware on the host computer, and the firmware update is more convenient
- CAN withstand high and low temperature, -40℃~85℃ stable operation
- It comes with a 120 ohm terminal resistor
- Supports 9-36v wide voltage input and has anti-reverse protection
- Reliable hardware protection, static protection, surge, pulse group three levels of protection
- Hardware watchdog function, automatic restart when the system crashes, and the module is more stable and reliable

Directory

1. Product overview

1.1. Product introduction

USR-CAN315/316 is a rail-mounted CAN (FD) to Ethernet/RS485/RS232 device independently developed by the iot company. This series of products features high speed, low latency, stable performance, ease of use, and excellent cost-effectiveness. It supports five data conversion modes: Transparent conversion, transparent with id conversion, standard protocol conversion, modbus conversion, and custom conversion. Among these, CAN316 supports interconnection between CAN devices and serial devices, while CAN315 supports interconnection between CAN devices and network interfaces. The device supports the CANFD protocol and is compatible with the standard CAN 2.0a/2.0b protocol.

This series of products adopts industrial-grade design standards, operating stably at-40℃ to 85℃. It supports a wide voltage range of 9 to 36v for terminal power supply. The baud rate range is broad, with arbitration domain baud rates ranging from 5k to 1mbps; data domain baud rates from 100k to 5mbps. Custom baud rates are supported, which CAN be calculated using the baud rate calculator on the host computer. The product supports at commands and parameter configuration via host software, making it easy to use. It comes with a built-in 120Ω resistor, which CAN be quickly connected to the CAN-bus bus through a dip switch. The product also comes with a mounting rail for convenient and quick installation.

In order to meet the needs of more customers, there are two main specifications available.

This manual mainly introduces the product functions of the web version of USR-CAN315.

Tab 1 USR-CAN316/315 specification selection table

Model	Edition	Specific description
USR-CAN316	Serial port version	Realize CAN (FD) to RS485/RS232, two-way data conversion
USR-CAN315	Network port version	Implement CAN (FD) to Ethernet, two-way data conversion

1.2. Technical parameter

Tab 2 Basic parameters of the product

Classify	Parameter	Numeric value
Essential parameter	Working voltage	Dc 9~36v, recommended to use 12v 1a
	Size	110*27*76.1mm
	Way to install	Railway installation
	Reload key	Long press to restore factory settings
	Pilot lamp	Power,work,net,CAN
Interface	CAN port specification	One CAN port, supports CANFD and is compatible with CAN 2.0a/2.0b
	CAN port baud rate	Arbitration domain baud rate: 5k~1mbps; data

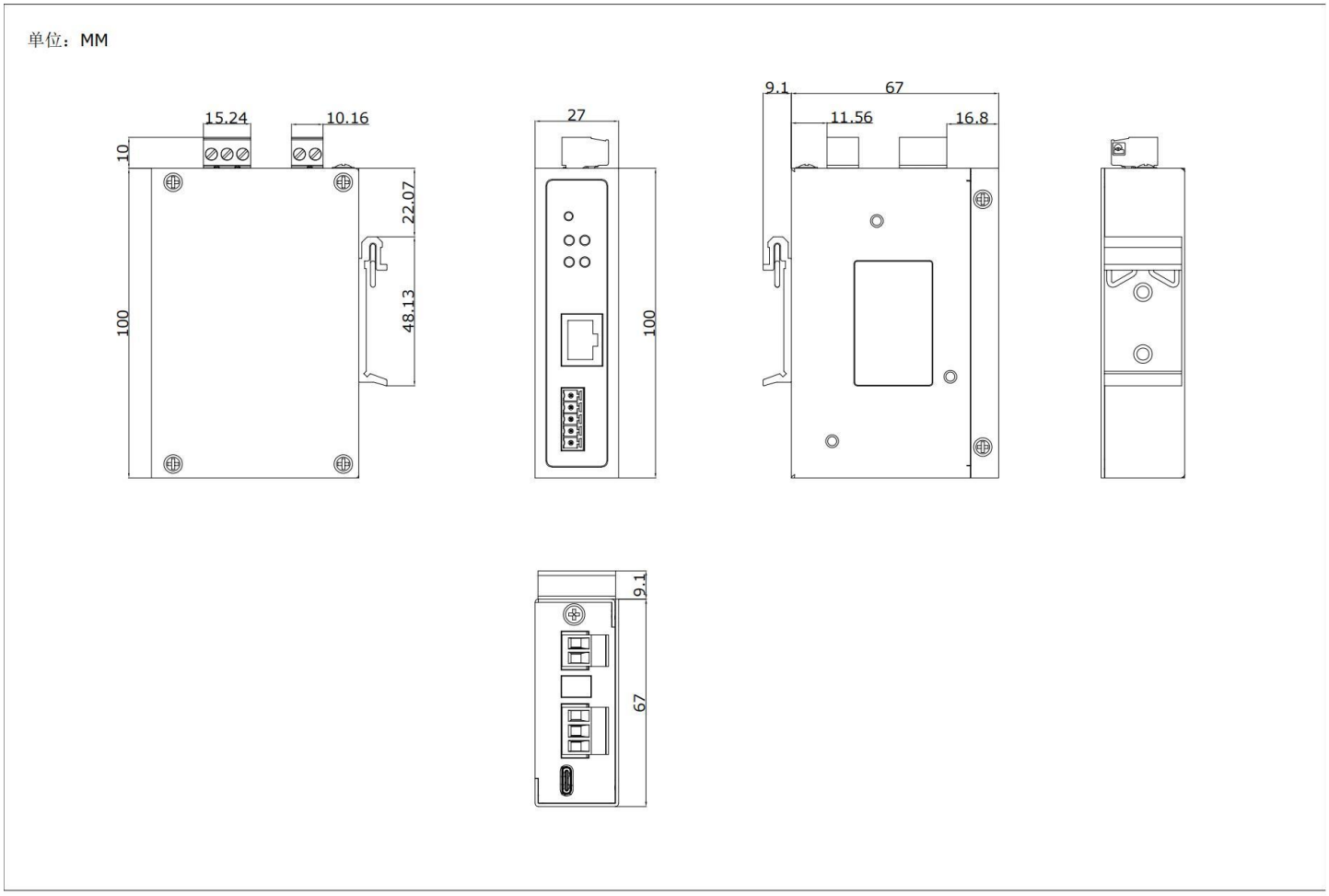
parameters		domain baud rate 100k~5mbps
	Terminal resistance	Built-in 2 CAN bus 120Ω terminal resistor. By using the code control resistor access, turn any code to on, and connect a 120Ω resistor in parallel. Turn both dials to on and connect two 120Ω resistors in parallel
	Network port specification	Rj45,10/100mbps, cross-connect adaptive
Work environment	Working temperature	-40~85℃
	Storage temperature	-40~105℃
	Working humidity	5~95% RH (no condensation)
	Storage humidity	5~95% RH (no condensation)
Software function	Networking protocol	Tcp Server,Tcp Client,UDP Server,UDP Client
	Twin channel socket	Support
	Conversion mode	Transparent conversion, transparent band id conversion, standard conversion, modbus conversion, custom frame header and tail conversion
	CAN Id	Support standard frame and extended frame
	Frame id filtering	Supports only standard frames, only remote frames, and custom input frame id (up to 32 sets)
	Pack frame time	Supports customizing the number of frames and the time of packing
	Change direction	Supports bidirectional conversion, only network port to CAN, and only CAN to network port
	Work pattern	Normal, loop, listen only
	Firmware upgrade	Support the upper computer to upgrade firmware
	Parameter configuration	At command, upper computer software configuration
	Heartbeat	Support network heartbeat packet and CAN port heartbeat packet
	Registration package	Supports custom, mac registration package; CAN

		choose connection transmission, data carrying, full registration
Protection parameters	Electrostatic protection	Air discharge 8kv, contact discharge 6kv
	EFT/Burst	Power circuit 2kv; Network port and CAN port circuit 1kv
	Surge Immunity Test	The differential mode of the power circuit is 1kv, the common mode is 2kv; the common mode of the CAN port circuit is 2kv; and the network port circuit is 1kv

2. Hardware parameters

2.1. Size description

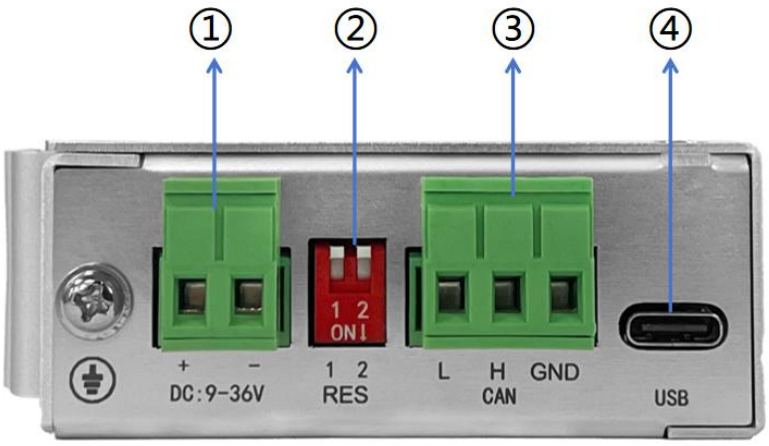
Machine size (including terminals and rails): 110*27*76.1mm



Pic1 USR-CAN315 standard size diagram

2.2. Interface description

The USR-CAN315 interface is described as follows.



Pic2 Interface specification

Tab3 terminal wiring definition

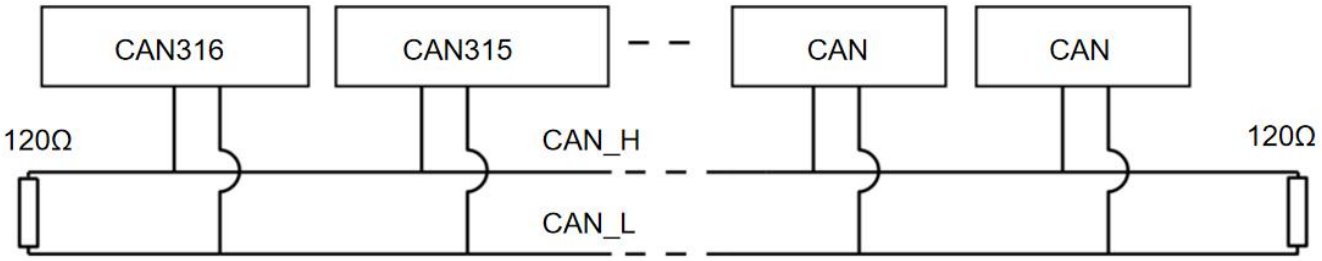
Order number	Interface name	Function declaration
1	Dc 9-36v +	Power supply interface, dc 9-36v positive power supply
	Dc 9-36v -	Power supply interface, dc 9-36v power negative pole
2	Res 1	Terminal resistance 1,120Ω. Turn it down to on and connect the resistance in parallel to the CAN bus. Default OFF.
	Res 2	Terminal resistance 2,120Ω. Turn down to on and connect the resistance in parallel to the CAN bus. Default OFF.
3	CAN l	CAN interface, CAN_l signal line connection end
	CAN h	CAN interface, CAN_h signal line connection end
	CAN gnd	CAN interface, CAN ground signal line connection end
4	USB	Standard type-c interface, through which firmware upgrade CAN be performed

<explain>

When the USR-CAN315 is connected to the CAN bus, CAN_h should be connected to CAN_h and CAN_l to CAN_l.

Res is the terminal resistance selection. If any code is turned to on, the 120Ω resistance inside the module is incorporated into the CAN bus; otherwise, the 120Ω resistance is not connected to the bus.

According to ISO 11898 standards, to enhance the reliability of CAN-bus communication, two endpoints of the CAN-bus network typically need to be equipped with termination matching resistors (120Ω), as shown in the figure below. The size of the termination matching resistor is determined by the characteristic impedance of the transmission cable. For example, if the characteristic impedance of the twisted pair is 120Ω, then both endpoints on the bus should also integrate a 120Ω termination resistor.



Pic3 CAN bus connection

2.3. Instructions for indicator lights

The USR-CAN315 has four indicator lights: Power, work, net and CAN. Users CAN easily observe the status of the device through the indicator lights as defined below.

Tab4 indicators rules

Pilot lamp	Pigment	Function declaration
Power	Red	The power is always on and off
Work	Green	Flicker: The device is running normally, frequency 1s; Stroboscopic: Enter the passive error state of CAN bus; Changchang: CAN bus is running abnormally
Net	Green/red	Green light flashing: Indicates that data is being received on the serial port Red light flashing: Indicates that data is being sent on the serial port
CAN	Green/red	Green light flashing: Indicates that data is being received on the CAN port Red light flashing: Indicates that data is being sent on the CAN port

3. Product function

3.1. Function configuration description

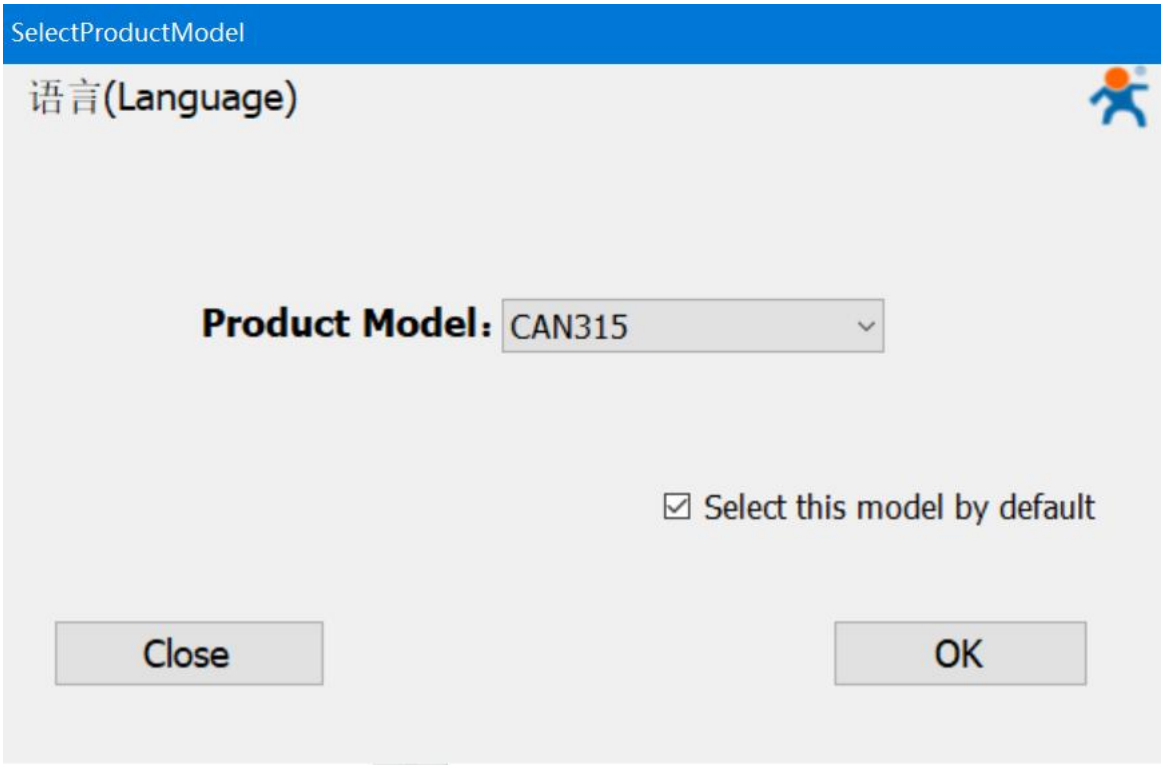
CAN315 supports the configuration parameters of the host computer and also supports the configuration of network at

instructions.

Parameters CAN be configured and queried through at instructions. For details of specific at instructions, see CANFD protocol converter at instruction set

The configuration of the upper computer is simple and easy to use. The following describes the configuration parameters of the upper computer. Please read this description in detail.

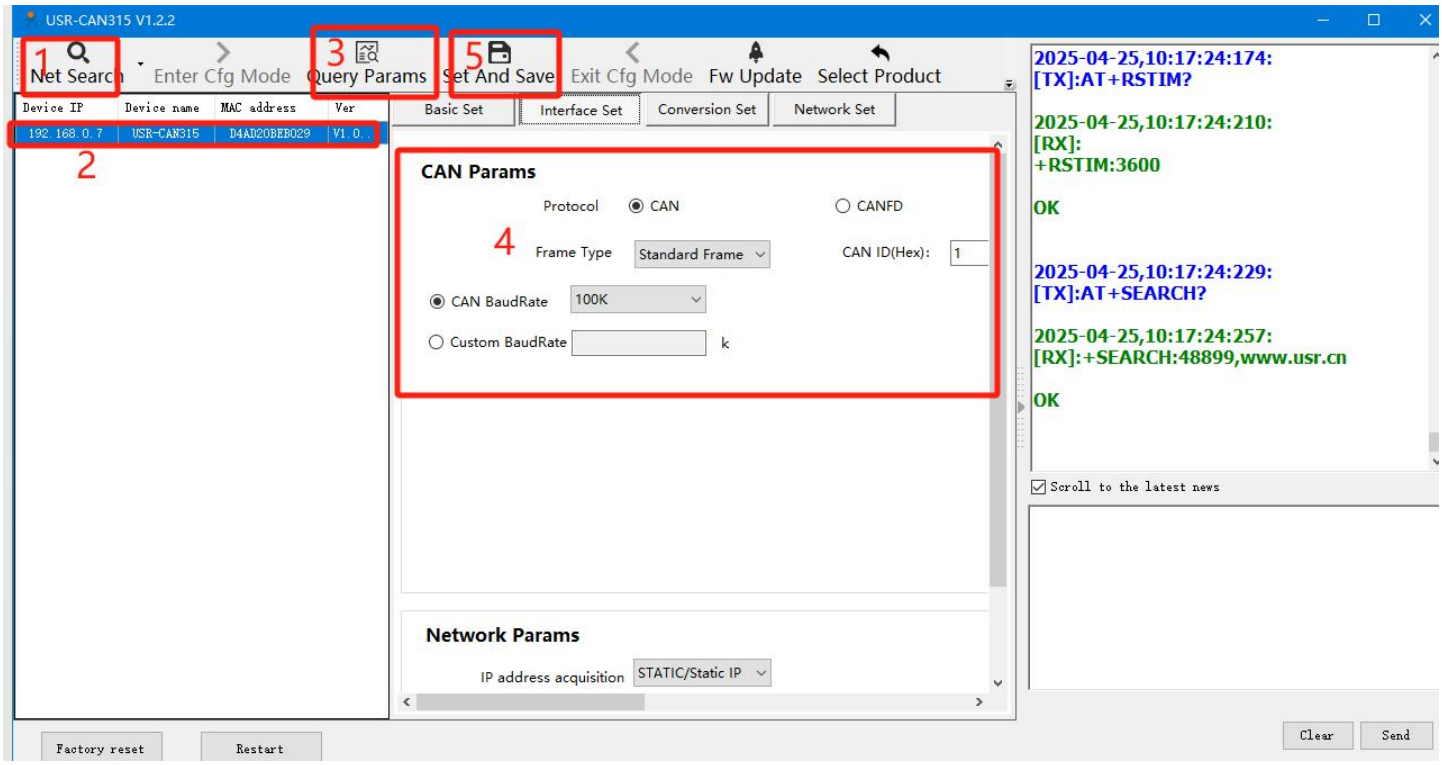
（1）Download the upper computer from the official website, and first select the model after opening it. CAN315 you CAN check the default login for this model.



Pic4 Choose the model

（2）

- First, do a web search for the device.
- Select the device and click read parameters.
- After all the current parameters are read, parameter configuration is performed.
- After the configuration is complete, click settings parameters
- Click restart, and all parameter configurations take effect



Pic5 Connect network configuration

3.2. CAN parameter

Protocol: Supports CAN or CANFD mode. When selected as CAN, the transponder will forward serial data into CAN message; when selected as CANFD, the transponder will forward data into CANFD message.

CANFD acceleration: This parameter enables the baud rate switching function of CANFD and is only effective in CANFD mode.

Frame type: The frame type of CAN message during conversion. Standard frame and extended frame are optional.

CAN id: 16-bit, hex format. Range: 0~7ff (standard frame), 0~1fffffff (extended frame)

Baud rate:

（1）In CAN mode: The range is 5kbps~1000kbps, and the default rate is 100kbps. You CAN directly select common baud rates: Custom baud rates are supported.

（2）In CANFD mode, the arbitration domain baud rate and data domain baud rate are divided. The arbitration domain baud rate enables the range: 5kbps~1000kbps, default is 100kbps. The data domain baud rate enables the range: 100kbps~5mbps. Only after CANFD acceleration is enabled, the data domain baud rate takes effect.

（3）Baud rate value: In both CAN and CANFD modes, the conventional baud rate recommended by cia CAN be directly set via the host computer. For more flexible usage, you CAN select the custom baud rate option. After selecting the custom baud rate, users CAN use a baud rate calculator to determine the desired baud rate value.

（4）Bit rate calculator:

Due to the purpose of custom baud rate, it is mainly for more flexible configuration of sampling point information. However, if the sampling point information is inconsistent or out of the allowable error range, normal communication CANnot be achieved, manifesting as bidirectional disconnection, or only one-way reception or transmission. Therefore, the baud rate calculator primarily enumerates the sampling point information. The calculation methods for arbitration domain baud rate and data domain baud rate sampling points are both:

Smp = 100% * (bs1 + 2)/((bs2+2) + (brp + 1))

Baud = clk / [(bs1+2) + (bs2 + 1)] * (brp+1)

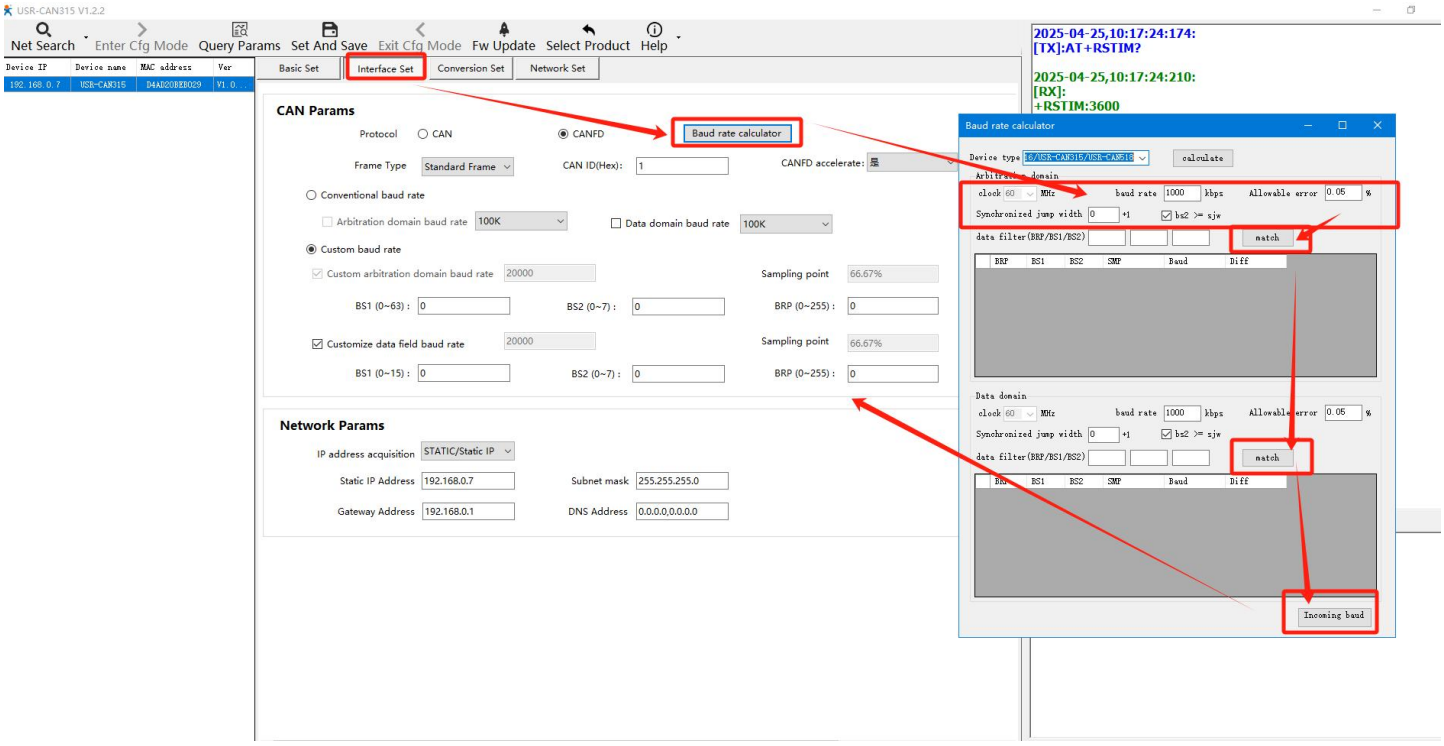
The parameter value of the baud rate sampling point is:

Arbitration domain: Brp (0~255), bs1 (0~63), bs2 (0~7), sjw (0~15)

Data fields: Brp (0~255), bs1 (0~15), bs2 (0~7), sjw (0~7)

Take the arbitration rate as an example:

1) Set the required baud rate value and allowable error, where the clock (clk) is fixed at 60mhz and does not need to be set
2) Set the appropriate synchronous jump width (sjw). If you need to perform synchronous jump, select bs2>=sjw, which directly affects the value range of bs2. For example, if sjw is set to 5, the range of bs2 changes from 0~7 to 5~7
3) By checking the match, you CAN calculate all the sample point information that CAN be matched within the current baud rate and error value
4) Select the information of the sampling point needed, click the input baud rate, and upload the sampling point information
5) If there is too much information about the sampling point, you CAN input a certain value of brp/bs1/bs2 to filter the data and accurately select the required sampling point information



Pic6 CAN parameter configuration

3.3. Network port parameters

(1) IP address acquisition:

An IP address is the identity of a module in a lan and is unique in a lan, so it CANnot be duplicated with other devices in the lan. There are two ways to obtain an ip address for CAN315: Static IP and DHCP.

- Static state ip

Static ip requires manual setting by the user. During the setting process, pay attention to writing ip, subnet mask and gateway at the same time. Static ip is suitable for scenarios that require statistics of ip and devices and correspond to each other.

Advantages: Devices that CAN not be assigned ip addresses CAN be searched through the full network segment broadcast mode

Disadvantages: Different lan segments CAN not carry out normal tcp/UDP communication

- Dhcp

The main function of dhcp is to dynamically obtain ip address, gateway address, dns server address and other information from the gateway host, so as to avoid the cumbersome steps of setting ip address. It is suitable for scenarios where there are no requirements for ip and it is not required that ip corresponds to modules one by one.

Advantages: Access to routers and other dhcp server devices CAN communicate directly, reducing the trouble of setting ip address, gateway address and subnet mask;

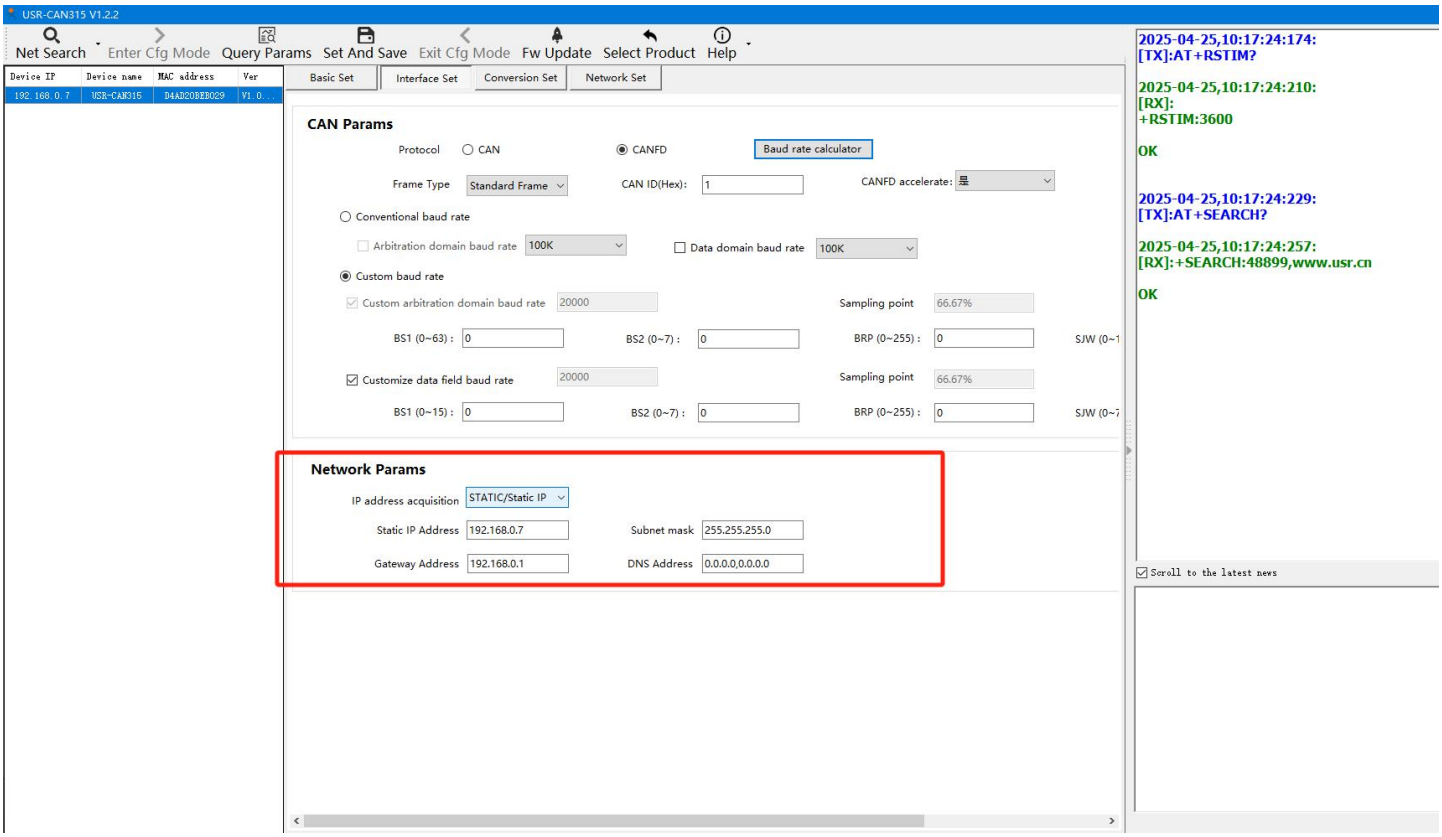
Disadvantages: If you connect to a network without dhcp server, such as directly connected to a computer, CAN315 will not work properly.

（2）Subnet mask:

A subnet mask is a 32-bit address used to mask part of an ip address to distinguish between network and host identifiers, indicating whether the ip address belongs to a local area network or a remote network. The subnet mask CANnot exist alone; it must be used in conjunction with the ip address. A commonly used class c subnet mask is 255.255.255.0, which allows for 2 to the power of 8 minus 2, or 256-2=254 ip addresses within a subnet. Typically, all host addresses are either all 0s or all 1s (in binary), which has special signifiCANce.

（3）Gateway address:

Gateway address refers to the network number of the network where the current ip address of the module is located. If you connect to an external network through a router or other devices, the gateway is the ip address of the router. If you set it incorrectly, you CANnot correctly access the external network. If you do not connect to a router or other devices, you do not need to set it, and it is default.



Pic7 Network port parameter configuration

3.4. Conversion Function

3.4.1. Conversion Parameter

Conversion mode: Supports transparent conversion, transparent id conversion, standard protocol conversion, modbus protocol conversion, and custom frame header and trailer conversion. Each mode has different conversion rules, and CAN realize the mutual conversion of serial frame information and CAN (FD) frame information. See chapter 4 for the specific description of conversion modes.

Conversion direction: By selecting the conversion direction, data interference on the bus side that does not need to be converted CAN be eliminated. There are three conversion directions as follows:

- Bidirectional: The converter converts data from the network to the CAN bus and converts data from the CAN bus to the network.
- Only network port to CAN: Only the data from the network is converted to the CAN bus, and the data from the CAN bus is not converted to the network.
- Only CAN to network port: Only converts data from the CAN bus to the network, and does not convert data from the network to the CAN bus.

Enable frame information: Only effective in transparent conversion. When this item is selected, the converter will add the frame information of CAN (FD) message to the first byte of the serial frame when working. When this item is not selected, the frame information of CAN (FD) is not converted.

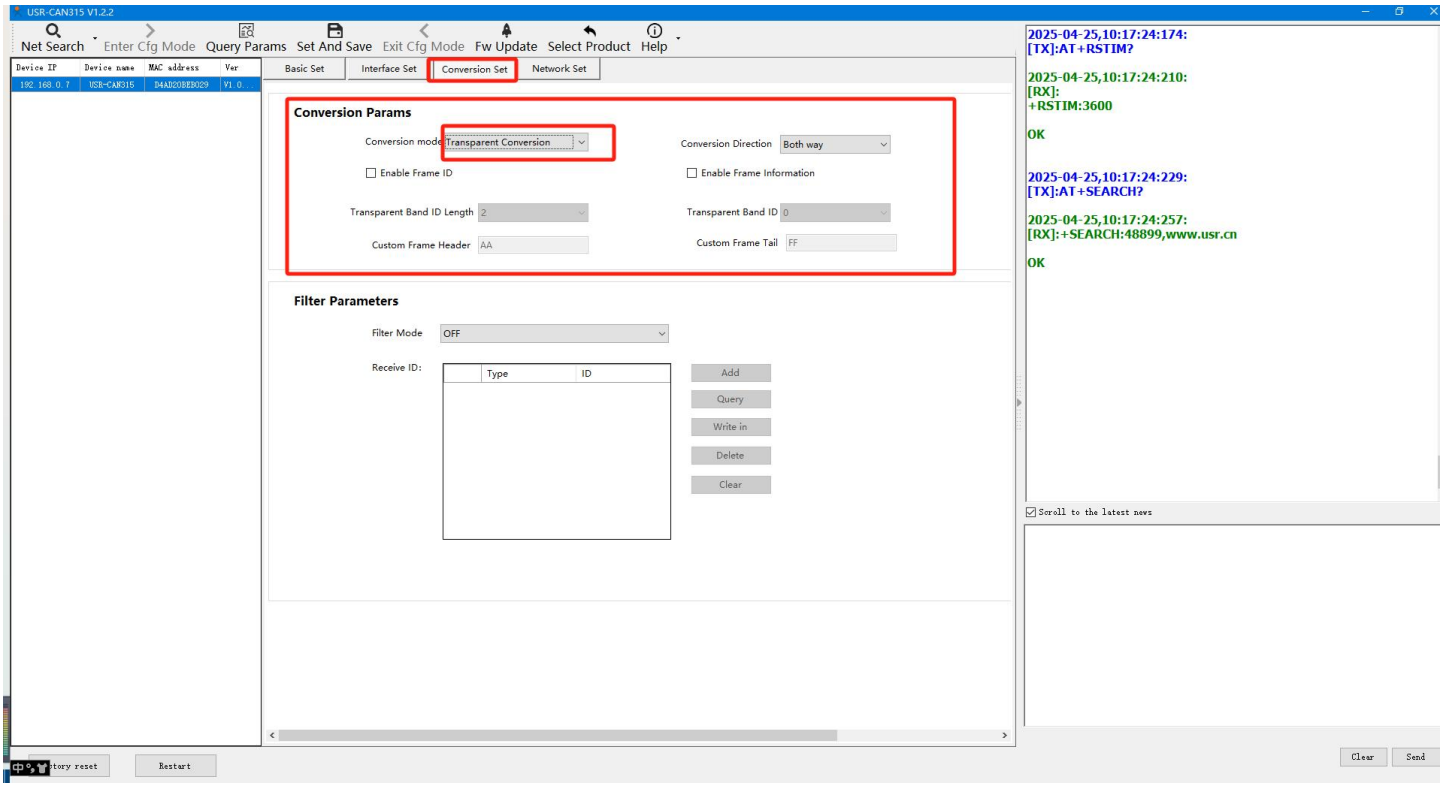
Enable frame id: Only effective in transparent conversion. When selected, the converter adds the CAN (FD) frame id before the frame data of the serial frame and after the frame information (such as enable frame information) when working. When not selected, the CAN (FD) frame id is not converted.

Transparent band id length: Only effective when the transparent band id is converted. When serial data is converted into a CAN (FD) message, the start byte of the frame id in the CAN (FD) message is the length of the frame id in the serial frame. In standard frames, the frame id CAN be padded with 1 to 2 bytes, corresponding to id1 and id2 of the CAN (FD) message. In extended frames, it CAN be padded with 1 to 4 bytes, corresponding to id1, id2, id3, and id4 of the CAN (FD) message. The id is 11 bits in standard frames and 29 bits in extended frames.

Transparent band id location: Only effective when the transparent band id is converted. When serial data is converted into CAN (FD) message, the start byte of the frame id of the CAN (FD) message is the offset position in the serial frame.

Custom frame header: Only effective when the custom frame header and frame tail are converted. The user CAN customize the serial frame header. Length: 1 byte.

Custom frame tail: Only effective when the custom frame head and tail are converted. The user CAN customize the serial frame tail. Length: 1 byte.



Pic8 Diagram of configuration of conversion parameters

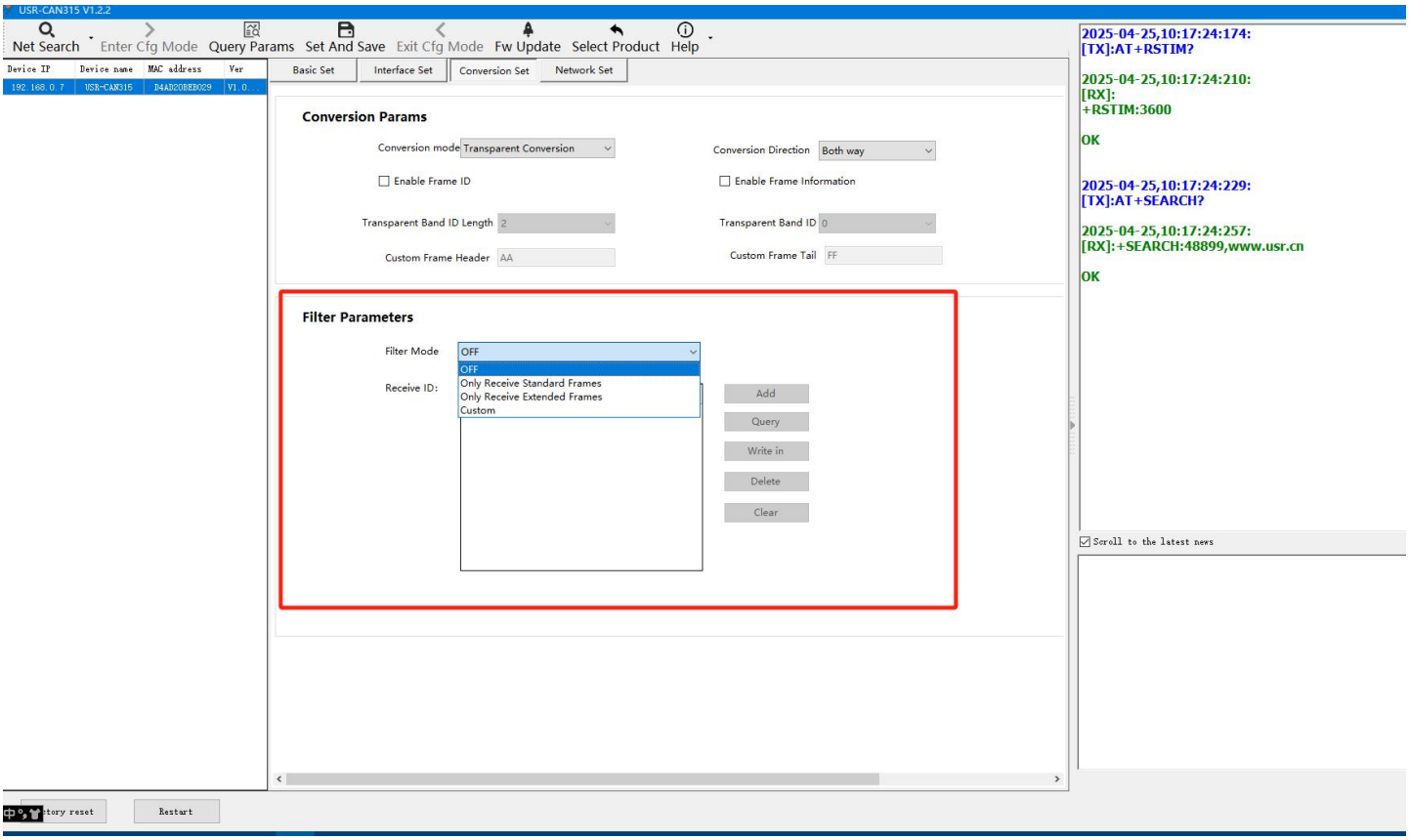
3.4.2. Filter Function

CAN315 it has the function of filtering id, which CAN filter CAN bus data and selectively receive it. In this way, the network load of the self-network CAN be reduced to the maximum extent.

There are three types of filtration:

- Only extended frames are received
- Only standard frames are received
- User-defined

Only extended frames and only standard frames are selected by configuration, which is as follows:



Pic9 Filter settings

In the custom mode, users CAN add the id they need to receive by themselves, and up to 32 groups CAN be set.

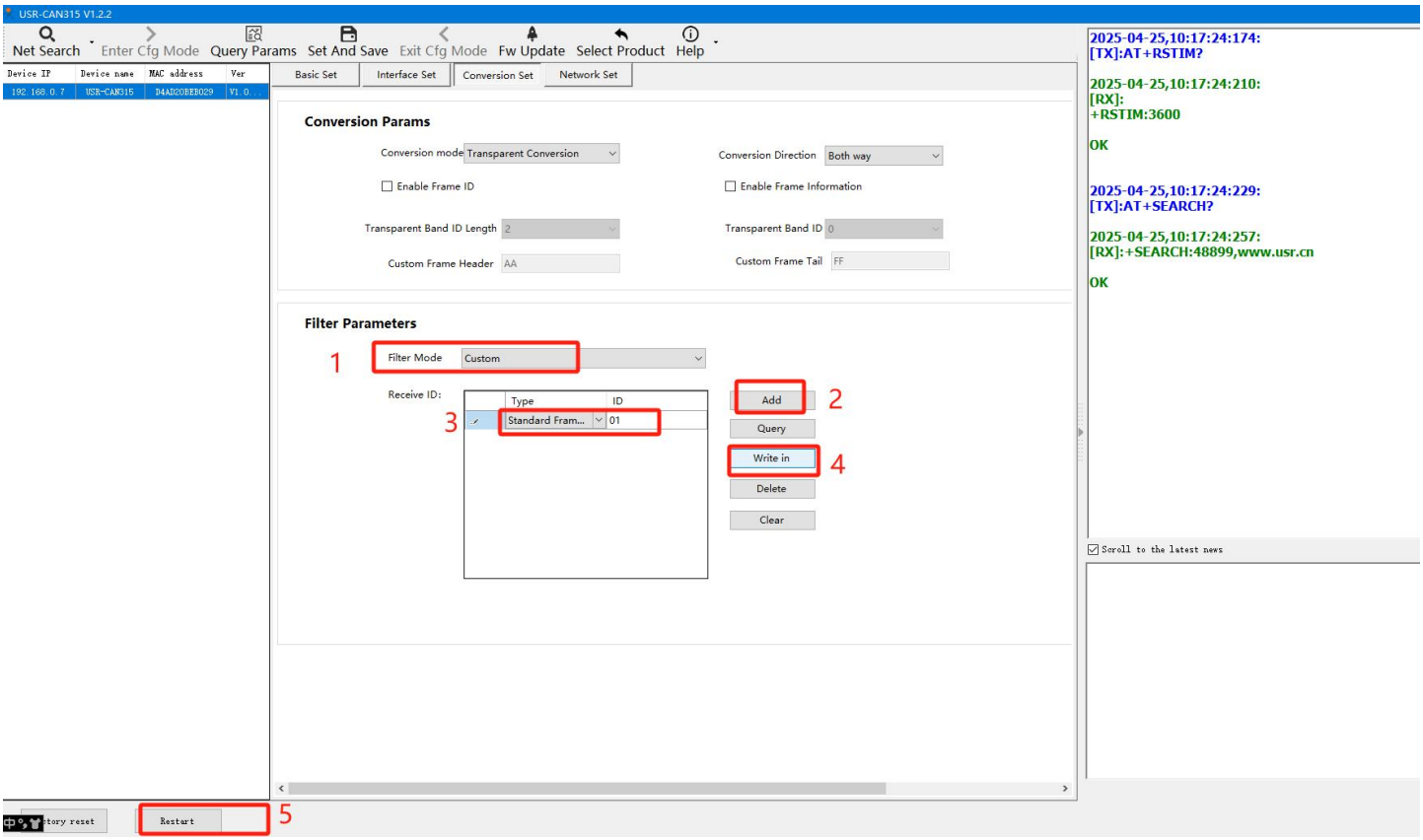
The configuration is as follows:

- Select the device you want to set up in the web search
- The filter mode is selected as custom
- Click add information and enter the id to be received. You CAN choose extended frame or standard frame for each group. Standard frame range: 0~7ff, extended frame range: 0~1fffffff
- Click write to restart and save the parameters

You CAN click to query all the current filter ids

Click delete to delete the selected id

Click to clear the list, you CAN delete all the current id



Pic10 Customize frame id configuration

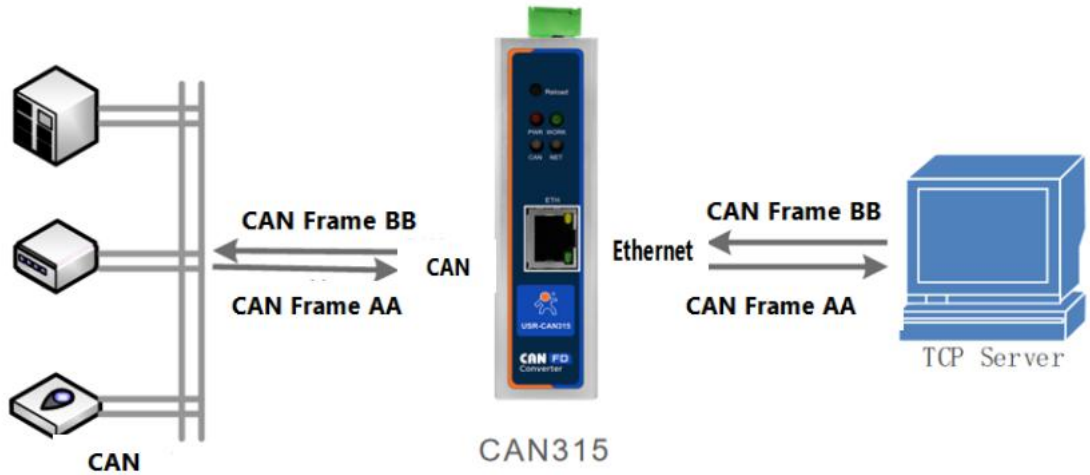
3.5. Socket function

CAN315 supports the simultaneous use of two socket. Socketa working modes are tcp client, tcp server, UDP client and UDP server. Socketb working modes are tcp client and UDP client. The specific introduction is as follows:

3.5.1. Tcp client pattern

(1) model description:

Tcp client provides client connections for tcp network services. It actively initiates connection requests to the server and establishes connections to realize the interaction between serial data and server data. It is usually used for data interaction between devices and servers, and is the most commonly used networking communication mode.



Pic11 Tcp client model description

1) Tcp client provides client connections for tcp network services. It actively initiates connections and connects to the

server to facilitate the exchange of serial data and server data. According to the relevant provisions of the tcp protocol, tcp client has a distinction between connection establishment and disconnection, ensuring reliable data exchange.

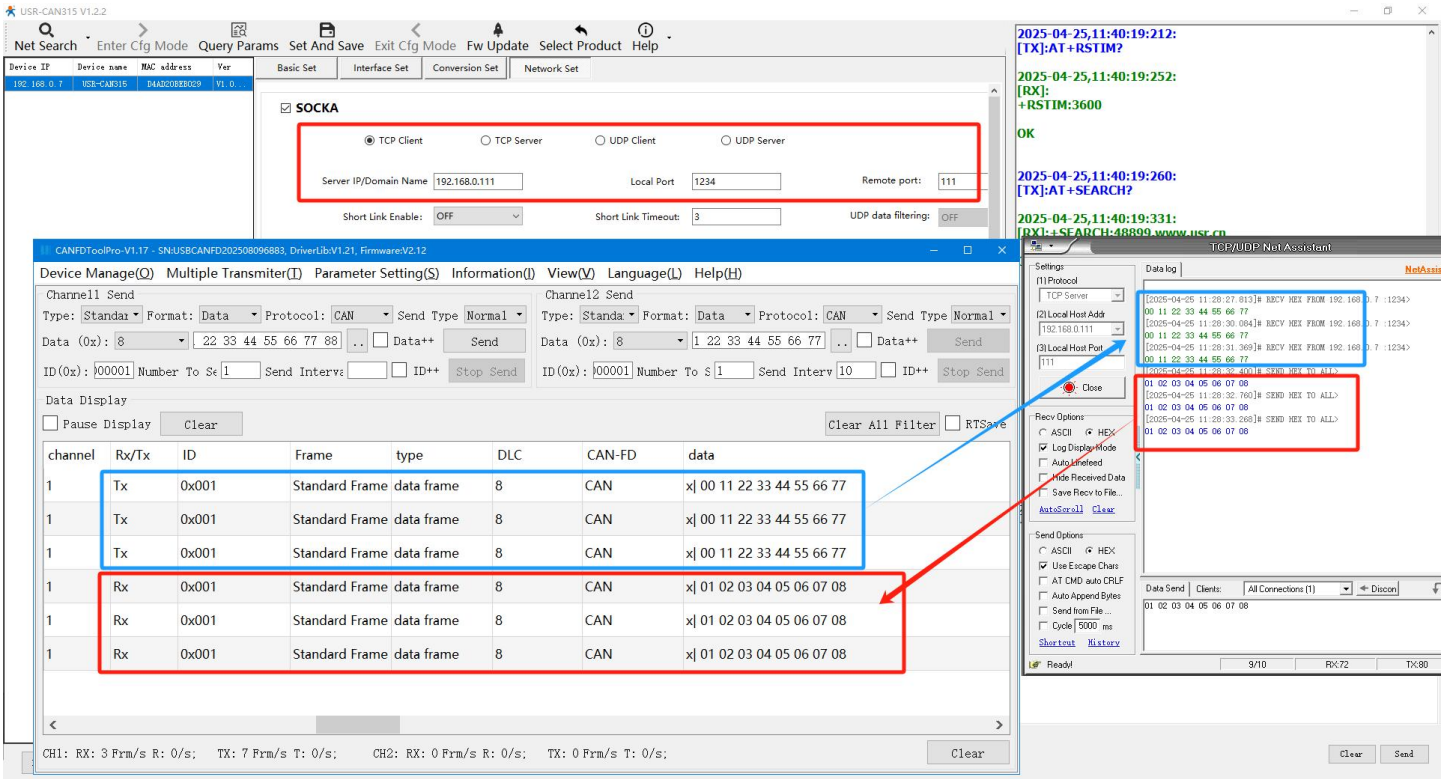
2) CAN315 to do tcp client, you need to connect to tcp server. Parameters to pay attention to: Target ip/domain and target port number. The target ip CAN be a device within the same local network or an ip address from a different local network, or even an ip across public networks. If connecting to a server across public networks, the server must have a public ip or domain name.

3) CAN315 when doing tcp client, it will actively connect to the target port of the target ip and will not accept other connection requests.

4) CAN315 to do tcp client, it is recommended to set the local port number of CAN315 to 0, so that CAN315 CAN access the server with a random port number. This CAN solve the problem that the server incorrectly determines the connection status and blocks the reconnection request sent by CAN315, resulting in the failure of reconnection.

5) This mode has the function of actively identifying connection anomalies. After the connection is established, keepalive keep-alive probe packets will be sent at an interval of about 15s. If there are any abnormal interruptions in the connection, they will be detected immediately, and the CAN315 will disconnect the original connection and reconnect.

6) In the same lan, if CAN315 is set as static ip, please keep the ip and gateway of CAN315 in the same subnet, and correctly set the gateway ip, otherwise normal communication will not be possible.

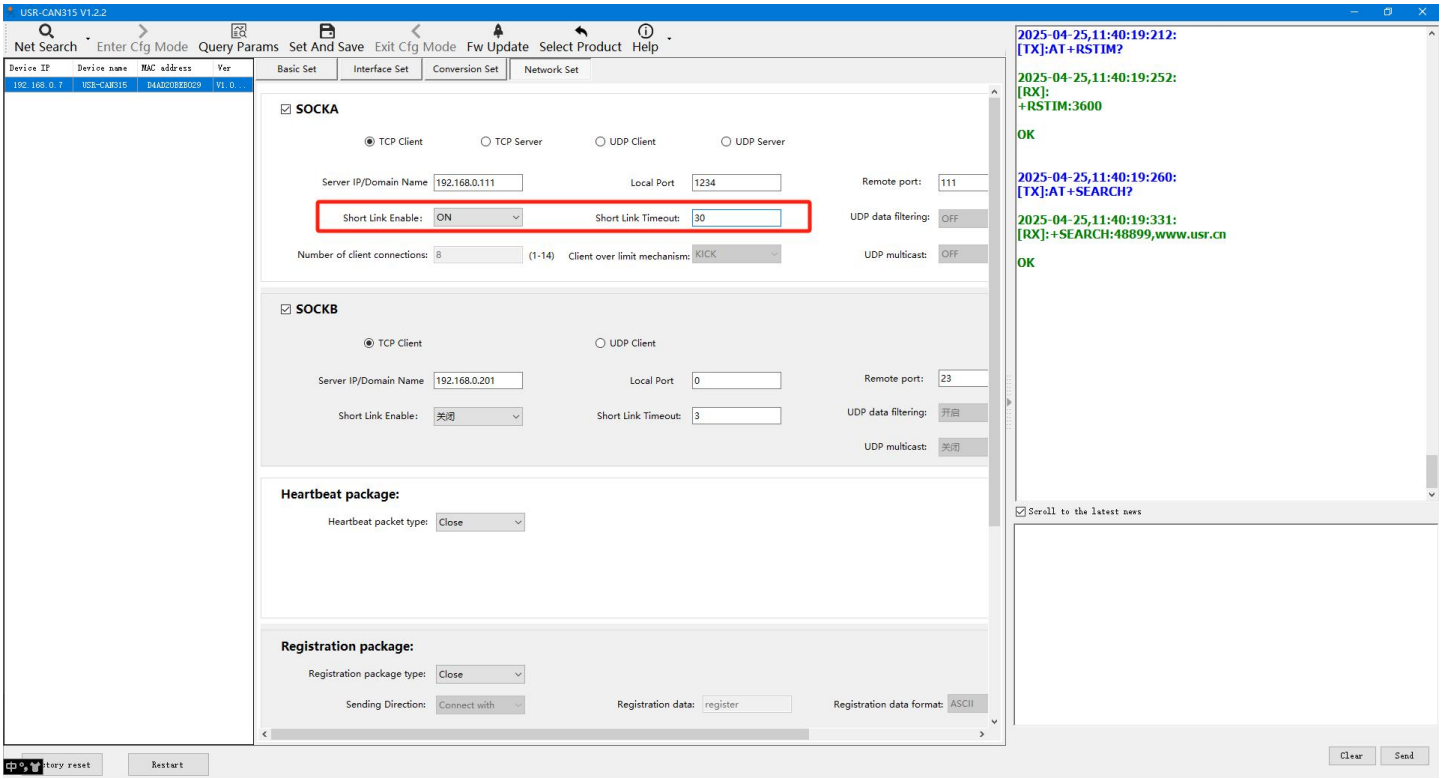


Pic12 Tcp client settings and data transmission diagram

(2) short connection

The use of short tcp connections is mainly to save server resources and is generally used in point-to-multipoint scenarios. Using short connections ensures that all existing connections are useful and do not require additional control means for filtering.

The short connection function of tcp is applied in tcp client mode. After enabling the short connection function, when sending information, if no data is received from the serial or network port within the set time, the connection will automatically disconnect. This function is initially disabled. The disconnection time CAN be set after enabling the function, with a range of 2 to 255 seconds, and the default setting is 3 seconds. The schematic diagram for setting it up is as follows:

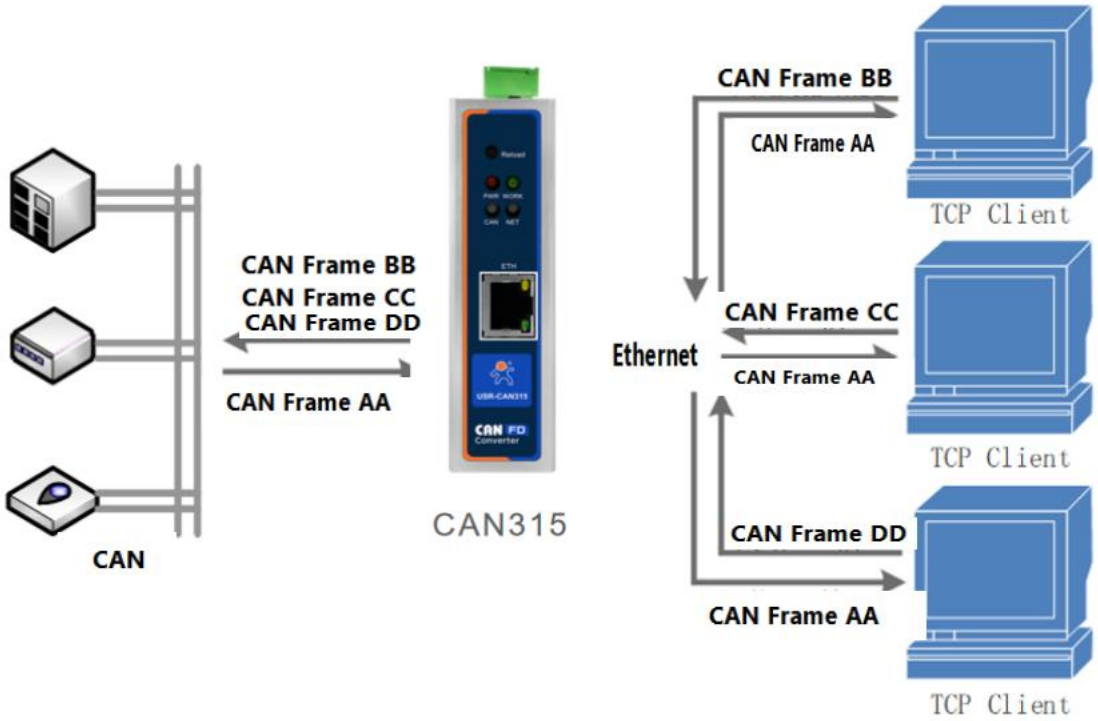


Pic13 Short connection setting diagram

3.5.2. Tcp Server Pattern

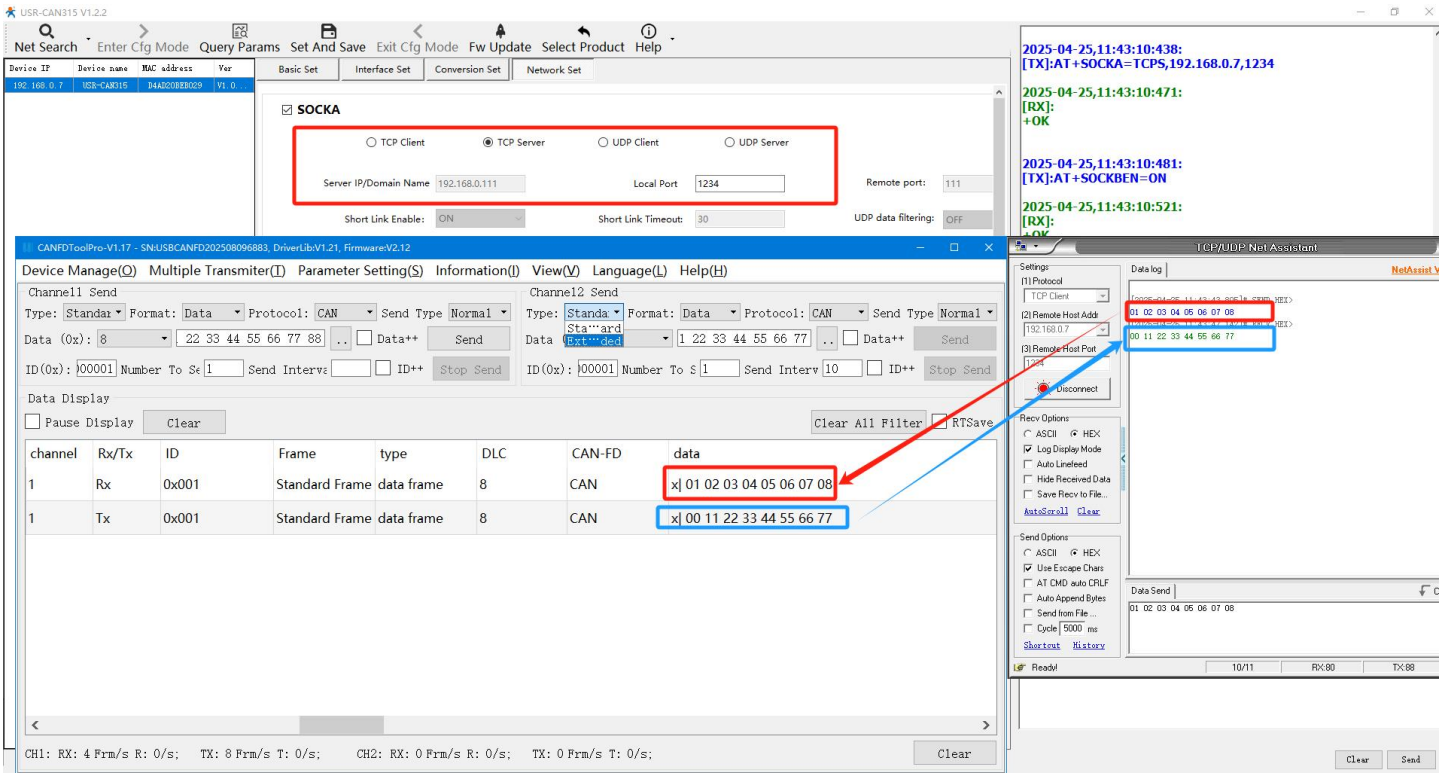
(1) model description:

Tcp server is the tcp server. In tcp server mode, CAN315 listens on its local port and accepts connection requests to establish a connection for data communication. When the CAN315 CAN port receives data, it simultaneously sends the data to all client devices that have established connections with CAN315. Similarly, tcp server mode also has keepalive functionality to monitor the integrity of connections in real time.



Pic14 Tcp server model description

It is usually used for communication with tcp clients within a lan. It is suitable for scenarios where there is no server in the lan and multiple computers or mobile phones request data from the server. Like tcp client, it has the difference between connection and disconnection to ensure reliable data exchange.



Pic15 Tcp server settings and data transmission diagram

(2) customize the number of client connections

CAN315 when doing tcp server, the number of client that CAN be connected is up to 14. The maximum value CAN be set according to customer needs for easy use by customers.

When the number of client connections is greater than the maximum value set by the user, the default new

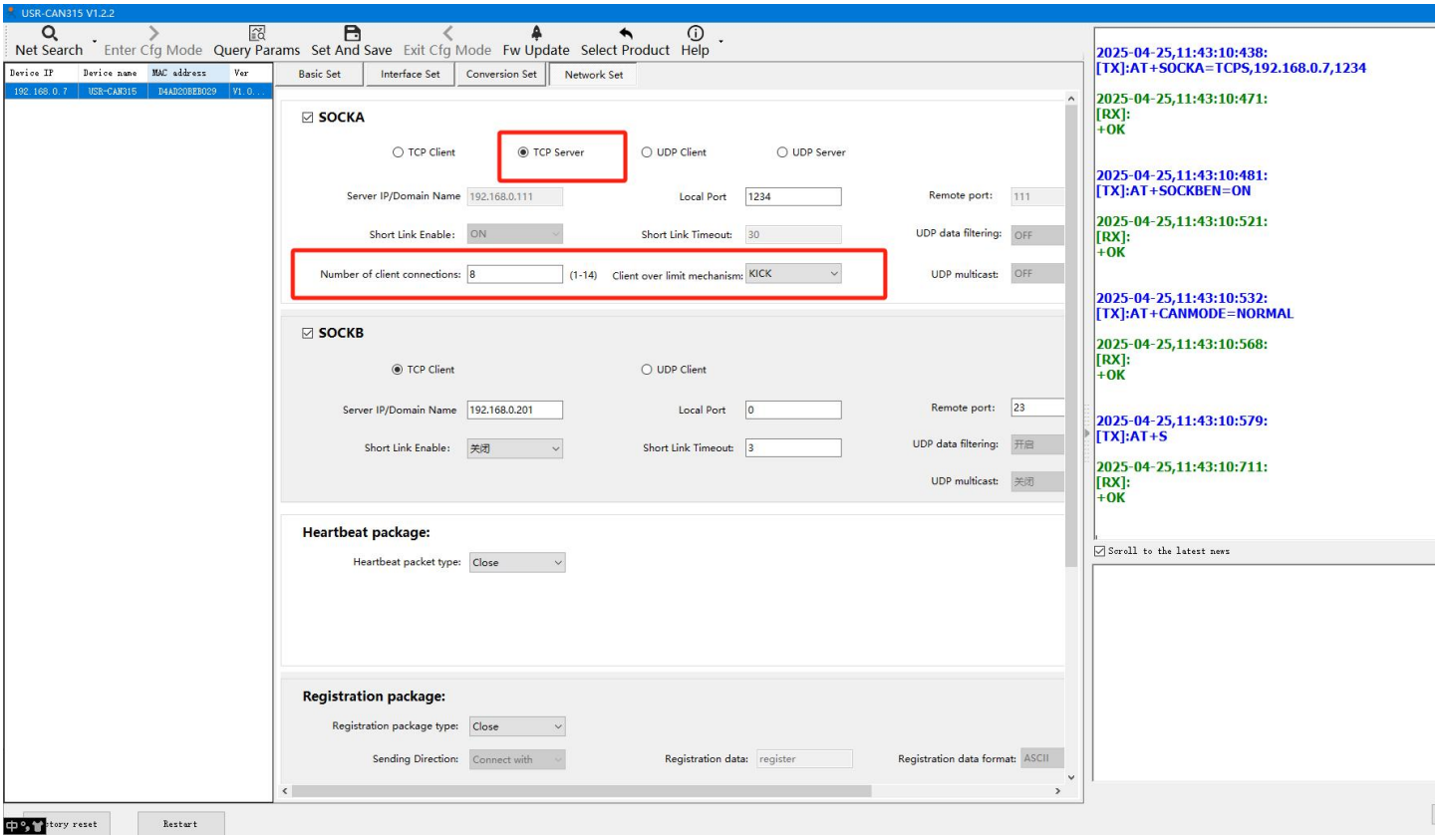
connection will replace the old connection, or you CAN set the new connection not to kick out the old connection.

Processing beyond the number of connections:

Kick: Kick off the old connection and connect to the new one.

Keep: Keep the existing connection and kick off the new one.

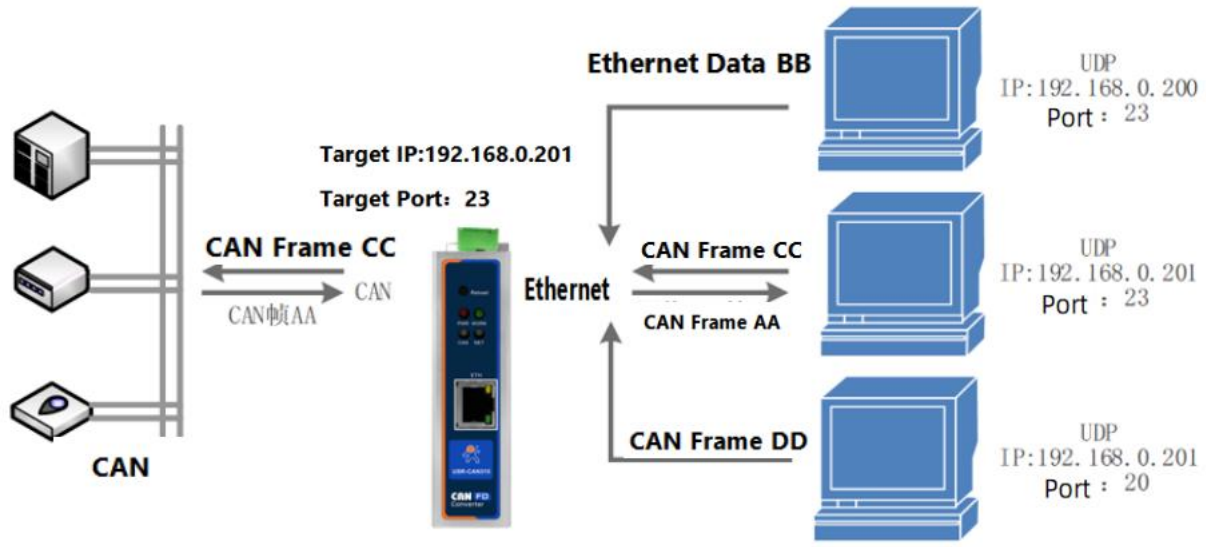
The schematic diagram is as follows:



Pic16 Set up the schematic diagram

3.5.3. UDP client Pattern

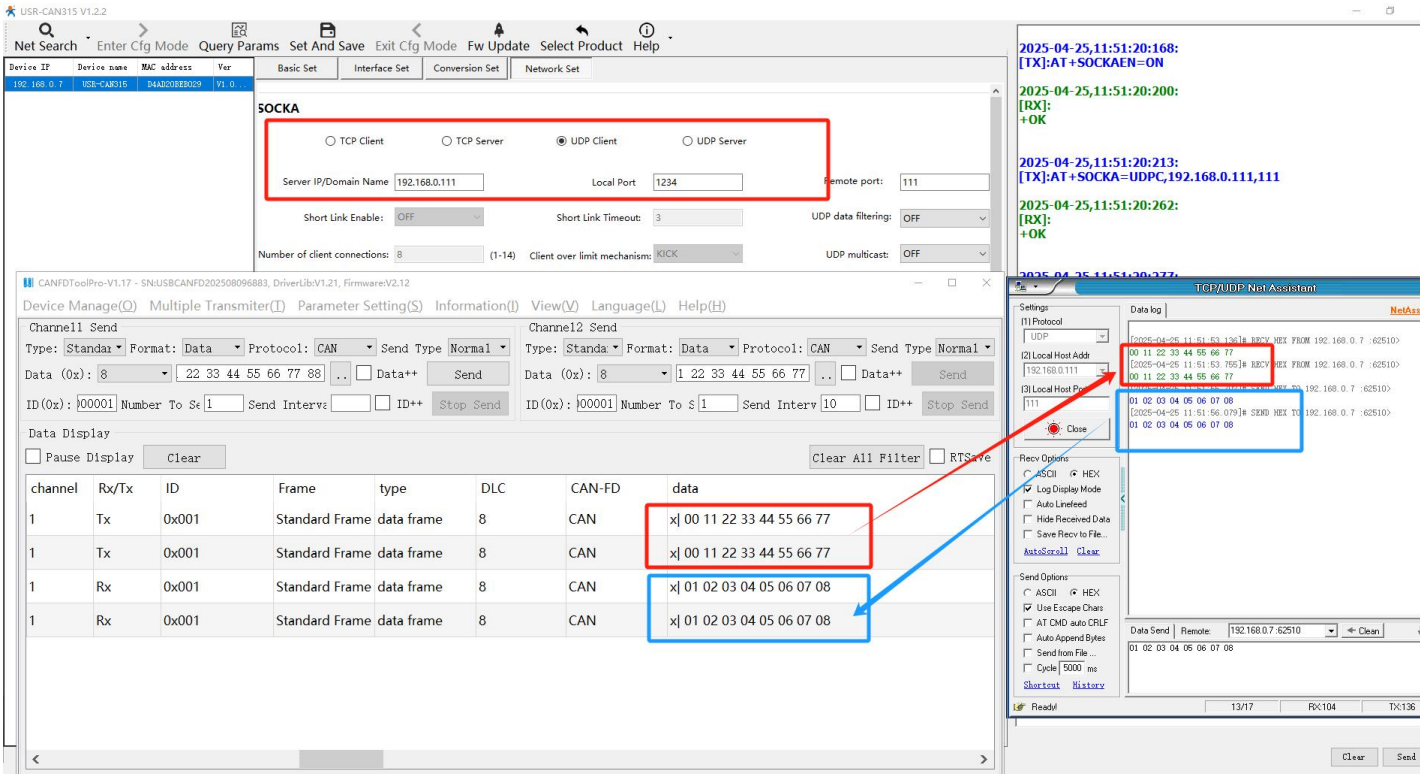
This work mode is subordinate to the UDP protocol. UDP client a connectionless transmission protocol that provides a simple, unreliable message delivery service for transactions. There is no connection establishment or termination; data CAN be sent to the recipient by setting up an ip address and port. It is typically used in scenarios where packet loss is not critical, data packets are small, and transmission frequency is high, with data being sent to a specific ip address.



Pic17 UDP client model description

1) In UDP client mode, the CAN port will only communicate with the target port of the target ip in the direction of the network.
2) In this mode, if the target address is set to 255.255.255.255, the effect of UDP broadcast across the entire network segment CAN be achieved; at the same time, broadcast data CAN also be received; broadcast within the network segment, such as xxx.xxx.xxx.255, is supported.
3) Supports UDP multicast functionality. Using multicast enables one-to-many connections between data senders and receivers. Multiple receivers CAN join the same multicast group, sharing the same ip address. Members in the multicast group are dynamic; the addition or removal of a member does not affect the existing multicast group. The valid address range for multicast groups is 224.0.0.2-239.255.255.255.
4) Supports the UDP data filtering function, which CAN be enabled through the upper computer configuration or at command. After enabling, it CAN select the data to be received from the whole network segment (e.g., 255.255.255.255), the data within the segment (e.g., 192.168.0.255), or the data to be passed through the specified ip (e.g., 192.168.0.201) according to the configuration.

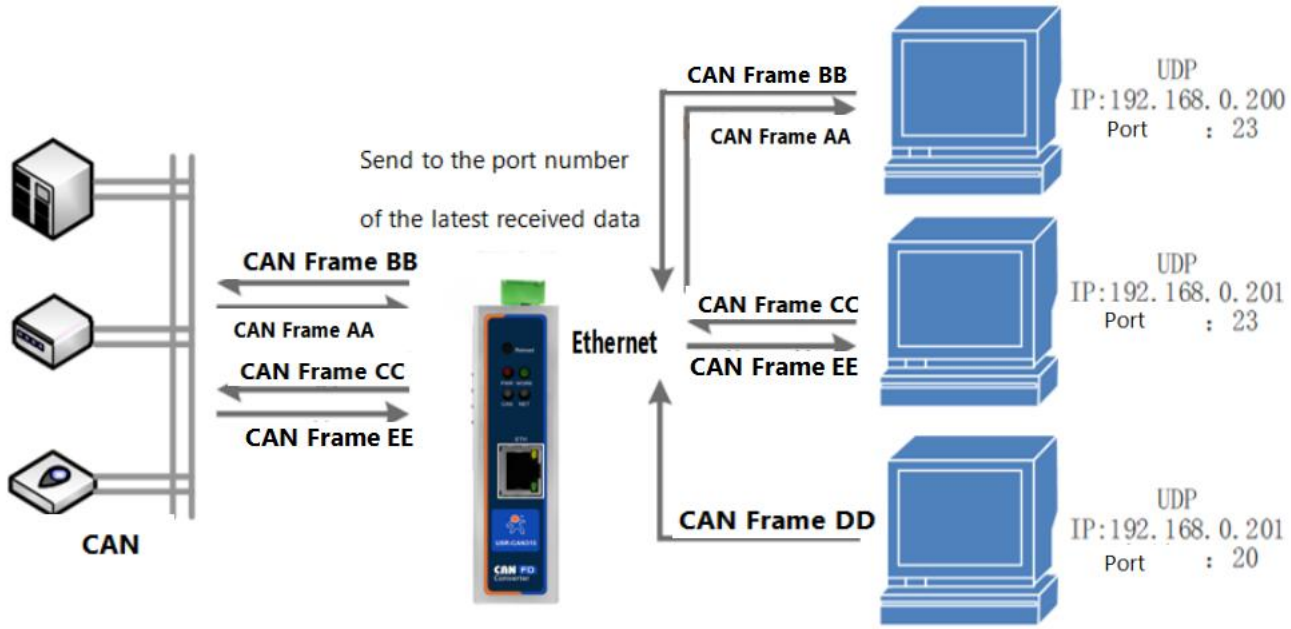
- The whole network segment broadcast does not judge the source ip, but only judges whether the source port is the same as the target port. If they are the same, the network data output is discarded; if they are different, the network data is discarded.
- The broadcast in the section determines whether the source port and target port are the same, and also determines whether the ip is an ip in the section. If the data meets the conditions, it is output; otherwise, it is discarded.
- For normal UDP communication, the data output is judged to be the same as the source port and ip and the target port and ip, otherwise it is discarded.



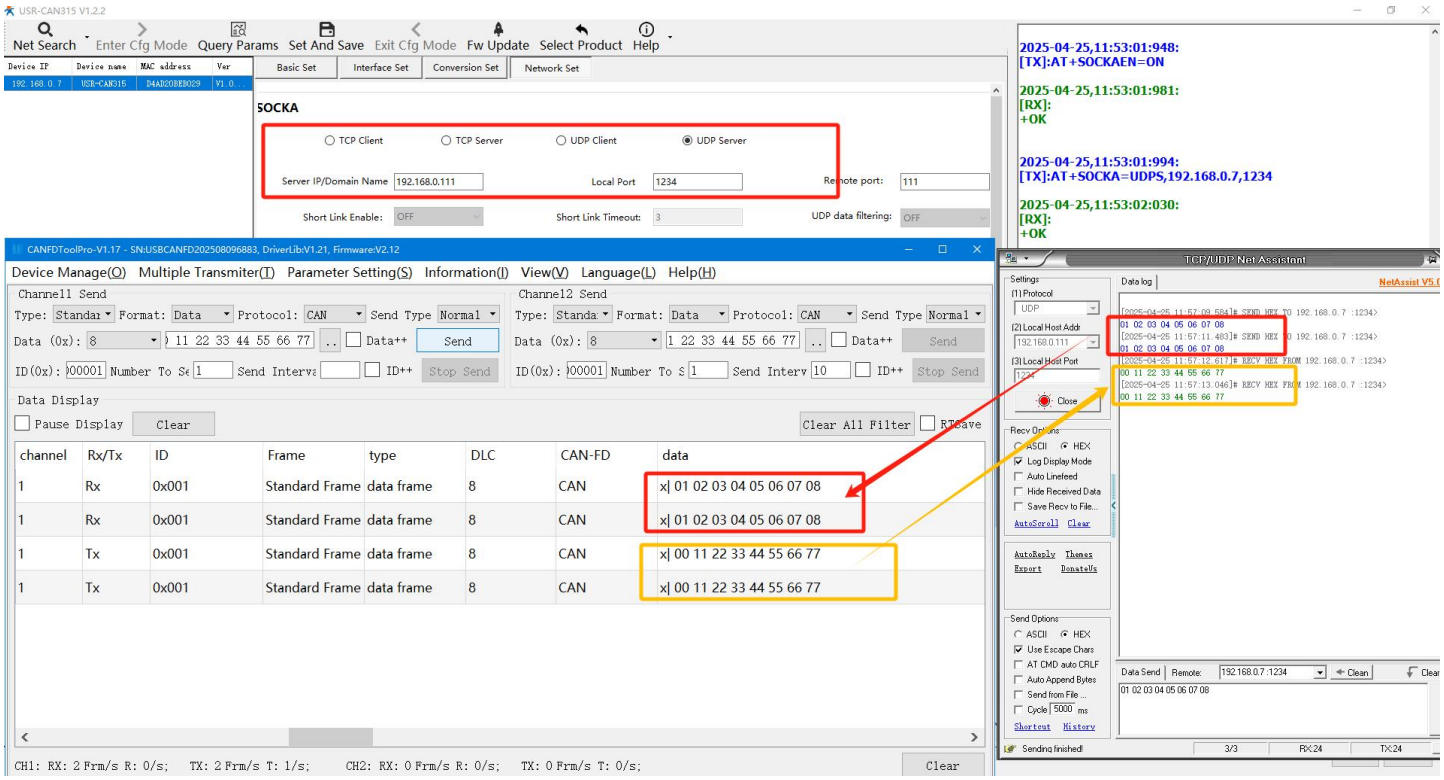
Pic18 UDP client setting and data transmission diagram

3.5.4. UDP server Pattern

UDP server refers to the approach that, based on regular UDP, does not verify the source ip address. After receiving each UDP packet, it changes the destination ip to the data source ip and port number, and when sending data, it sends to the most recent communication ip and port number. This mode is typically used in scenarios where multiple network devices need to communicate with modules and do not want to use tcp due to high speed and frequency requirements.



Pic19 UDPserver pattern



Pic20 UDP server settings and data transmission diagram

3.6. Modbus Gateway Function

This device enables simple interconnection between CAN and Modbus devices, supporting data transmission from Modbus to CAN for unified management and CAN-based control of Modbus device data.

The device can be used as both Modbus master and Modbus slave. CAN315 supports CAN to Modbus TCP (master/slave); CAN316 supports CAN to Modbus RTU (master/slave).

Note：Unlike Modbus protocol conversion in data conversion, this method does not require Modbus and CAN side data to be editable.

3.6.1. Enable Modbus configure

（1）Enter the parameter configuration state: network search--> select the device--> read the parameter--> enter the parameter configuration state.

（2）Configure CAN parameters. According to the following CAN devices, configure CAN protocol, frame type, and CAN baud rate.

（3）Set network parameters.

- When CAN315 is a **TCP client**, it supports CAN (FD) to Modbus TCP master station. It supports the access of a unique TCP slave station.

  Configure server IP: IP address of the unique slave

  Remote port: The port number of the sole slave

- When CAN315 works as a **TCP server**, it supports CAN (FD) to **Modbus TCP slave**. The device is preset with 2400 registers, and the external master station can write the preset registers at any time, ranging from 0 to 2399. It supports function codes 03,06 and 10.

- **Local port：** CAN315 is the port number of the slave

Note: This function is only applicable to the TCP mode of SOCKA.

（3） Enable Modbus Gateway and select the working mode of Modbus Gateway.

（4） Send and receive packets on demand.

3.6.2. Send message configuration

By configuring the device to send messages, you can actively collect Modbus data and map it to the corresponding positions in the CAN data frame. The data is then sent as CAN frames according to predefined rules. Up to 64 message configurations are supported, with each message capable of adding up to 32 variable data points, which are mapped to the corresponding data positions in the CAN frame.

The configuration of the upper computer is as follows:

（1）Enter the parameter configuration state: network search--> select the device--> read the parameter--> enter the parameter configuration state.

（2）Click New message to configure the required sending message content. Select the message and click New variable to configure the mapping point parameter.

（3）If there is an adjustment, you can delete a single message or a single point. You can also click delete all to directly delete all messages.

（4）After the point configuration is completed, click the configuration data to save and restart.

Parameter information introduction:

parameter	meaning	range	.csv
Message parameters			
Slave Address	When the device is the master station, fill in the address of the opposite end; When the device acts as a slave, fill in the address of the device when it acts as a slave;	1~255	1~255
Message name	The name of the message to be sent is not mapped to the CAN data and is used only for mnemonics	Support English or number	Correspond to English or number
frame type	The frame type of the message being sent	Standard frame Expand the frame	Standard frame--A Expand the frame--B
frame ID	The frame ID of the message being sent	hex form ; 0~07FF(Standard frame) 0~1FFFFFFF(Expand the frame)	0~07FF(Standard frame) 0~1FFFFFFF(Expand the frame)
remote frame	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
Data Length	Send the length of the frame data segment, up to 8 bytes for CAN frame and up to 64 bytes for CAN FD frame. Note: CANFD frames need to be set to the length that DLC can encode	0~8 0~8、12、16、20、24、32、48、64	0~8 0~8、12、16、20、24、32、48、64
Send rules	(1)The mode in which the trigger device sends CAN messages. (2)Periodic sending: report according to the set period time; (3)Change sending: the group of messages is reported when any data point in the group changes; (4)Single send: a single send after the connection is established; (5)Frame ID trigger: Triggered to send after receiving the specified frame ID	Periodic send; Changes sent; Send once; Frame ID triggered;	Periodic send--1 Changes sent--2 Send once--3 Frame ID triggered--4
Trigger frame ID	It takes effect in frame ID trigger mode, which refers to the frame ID of the CAN frame that triggers the sending of this message;	hex form; 0~07FF(Standard frame) 0~1FFFFFFF(Expand the frame)	0~07FF(Standard frame) 0~1FFFFFFF(Expand the frame)
Trigger frame	It takes effect in frame ID trigger mode and refers to the	Standard frame	Standard frame--A

type	frame type of the CAN frame that triggers the transmission of this message;	Expand the frame	Expand the frame--B
periodic time	When the sending rule is periodic, it is the periodic sending time; When the sending rule is changed, it checks the period of Modbus data change. That is, in this period, if the group data changes, it will send; if the data does not change, it will not send; When the sending rule is a single send, the waiting time for that single send	0~65535ms	0~65535ms
variable parameter			
variable name	The name of this variable is not mapped to the CAN data and is used only for mnemonic purposes	Support English or digital	Correspond to English or number
Data size	The size of the mapped data. ALL:Whole frame data BYTE：1 Byte WORD：2 Byte DWORD：4 Byte QWORD：8 Byte	ALL BYTE WORD DWORD QWORD	ALL BYTE WORD DWORD QWORD
offset	Select the starting byte in the CAN message data segment to map Modbus register data sequentially. The offset is invalid when the operation size is ALL.	CAN：0~8 CANFD：0~64	CAN：0~8 CANFD：0~64
Register type	Modbus Register type	03 04	03 04
Register address	Start address of the transmitted message data in the Modbus slave's registers	0~65534	0~65534
Byte order	Modbus data storage mode	Big - Endian Little - Endian	Big - Endian--B Little - Endian--S

3.6.3. Receive message configuration

By configuring the device to receive messages, you can write the data segments required by CAN messages into the registers of the Modbus slave through CAN (FD) data frames. Up to 64 messages can be configured for receiving messages, and each message can add up to 32 variable data.

The configuration of the upper computer is as follows:

（1）Enter the parameter configuration state: network search--> select the device--> read the parameter--> enter the parameter configuration state.

（2）Add new messages and configure the required content of the received messages. Select the message and click

Add variable to configure the mapping point parameters.

（3）If there is an adjustment, you can delete a single message or a single point. You can also click delete all to directly delete all messages.

（4）After the point configuration is completed, click the configuration data to save and restart.

Parameter information introduction:

parameter	meaning	Range	.csv corresponding
Message parameters			
Message name	The name of the received message is not mapped to the CAN data and is used only for mnemonics;	Support English or digital	Correspond to English or number
frame type	The frame type of the received message;	Standard frame Expand the frame	Standard frame--A Expand frame--B
frame ID	The frame ID of the received message;	Hex form ; 0~07FF(Standard frame) 0~1FFFFFFF(Expand the frame)	0~07FF(Standard frame) 0~1FFFFFFF(Expand the frame)
Slave Address	When the device is the master station, fill in the address of the opposite end; When the device is a slave, fill in the address of the device as a slave;	1~255	1~255
variable parameter			
Message numbering	The sequence number of the message is not mapped to the CAN data and is used only to distinguish messages.	1~32	1~32
variable name	The name of this variable is not mapped to the CAN data and is used only for mnemonic purposes.	Support English or digital	Correspond to English or number
Data size	The size of the mapped data. ALL:Whole frame data BYTE：1 byte WORD：2 byte DWORD：4 byte QWORD：8 byte	ALL BYTE WORD DWORD QWORD	ALL BYTE WORD DWORD QWORD
offset	Select which byte in the CAN message data segment to start with, and map the received CAN message data segment to the registers of the Modbus slave in sequence. When the operation size is ALL, the offset is invalid.	CAN：0~8 CANFD：0~64	CAN：0~8 CANFD：0~64
Register type	Modbus Register type	03	03

Register address	The sent message data is at the starting address of the register on the device or Modbus slave	0~65534	0~65534

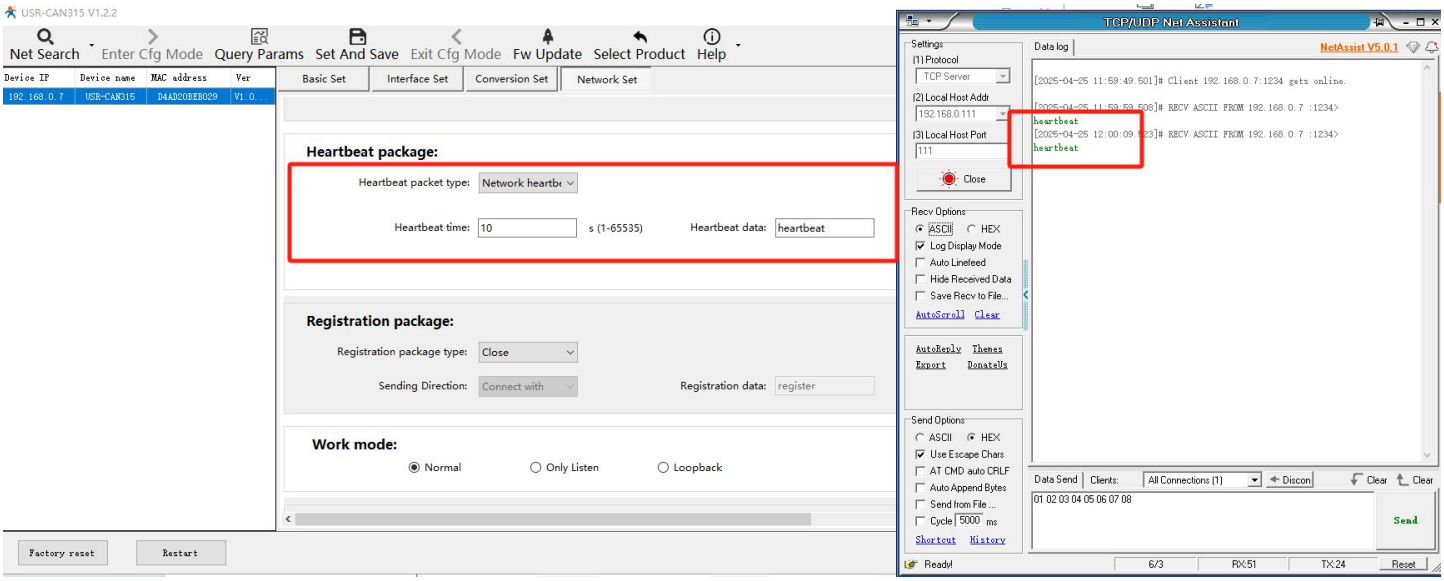
3.7. Special Features

3.7.1. Heartbeat

In the network transmission mode, users CAN choose to let CAN315 send heartbeat packets. Heartbeat packets are divided into network heartbeat packets and CAN port heartbeat packets, only one of which is effective at the same time. They CAN also be sent to CAN port devices.

Network heartbeat packet:

When there is no data on the network side, it sends to the network server side at regular intervals. The main purpose is to keep the connection with the server. It only works in tcp client and UDP client modes. It supports both hex and ascii formats.



Pic21 Network heartbeat packet setting and data transmission diagram

CAN port heartbeat packet:

It CAN be used as a fixed query command and sent to CAN through heartbeat packets. The content must conform to CAN format. CAN frame format, frame type and frame id CAN be configured.

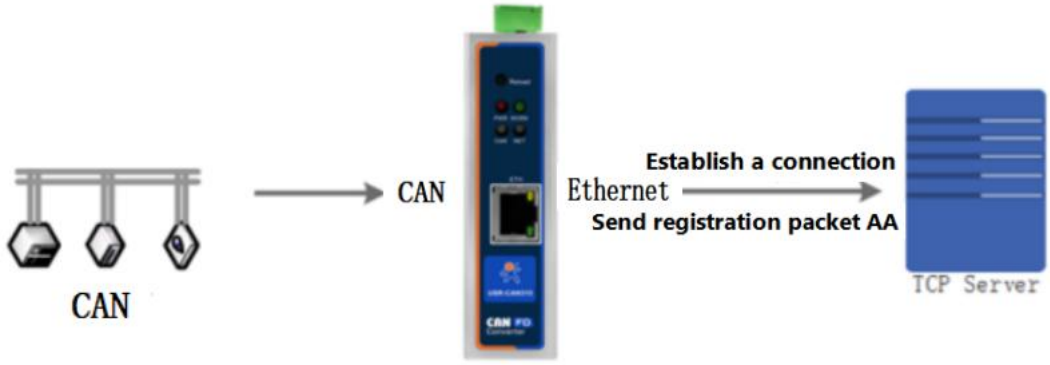
3.7.2. Registration Package

In the network transmission mode, users CAN configure the registration package function. The registration package CAN be used as an identifier for the server to identify the data source device, or as a password to obtain the server function authorization.

Registration packets CAN be configured to send either connection-based or carry-based registration packets, and they CAN also take effect simultaneously. Connection-based sending refers to sending when establishing a tcp connection or UDP connection, while carry-based sending means appending the registration packet data at the very beginning of each data packet, making it a single data packet. The data in the registration packet CAN optionally include a mac address or custom registration data, with the maximum length of custom registration packet settings being 40 bytes.

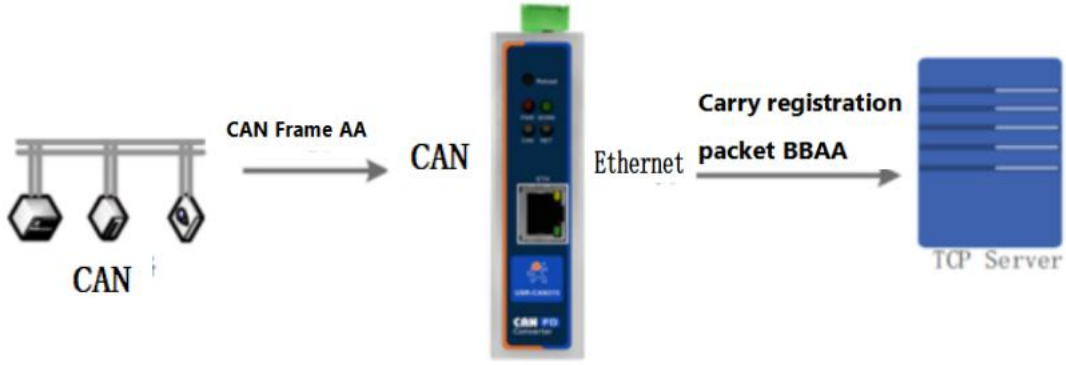
The registration package is only applicable to the tcp client and UDP client modes, and there is no registration package under the tcp server and UDP server modes.

Establishing a connection to send a registration packet is mainly used to connect to a server that needs to be registered. The application diagram is as follows:

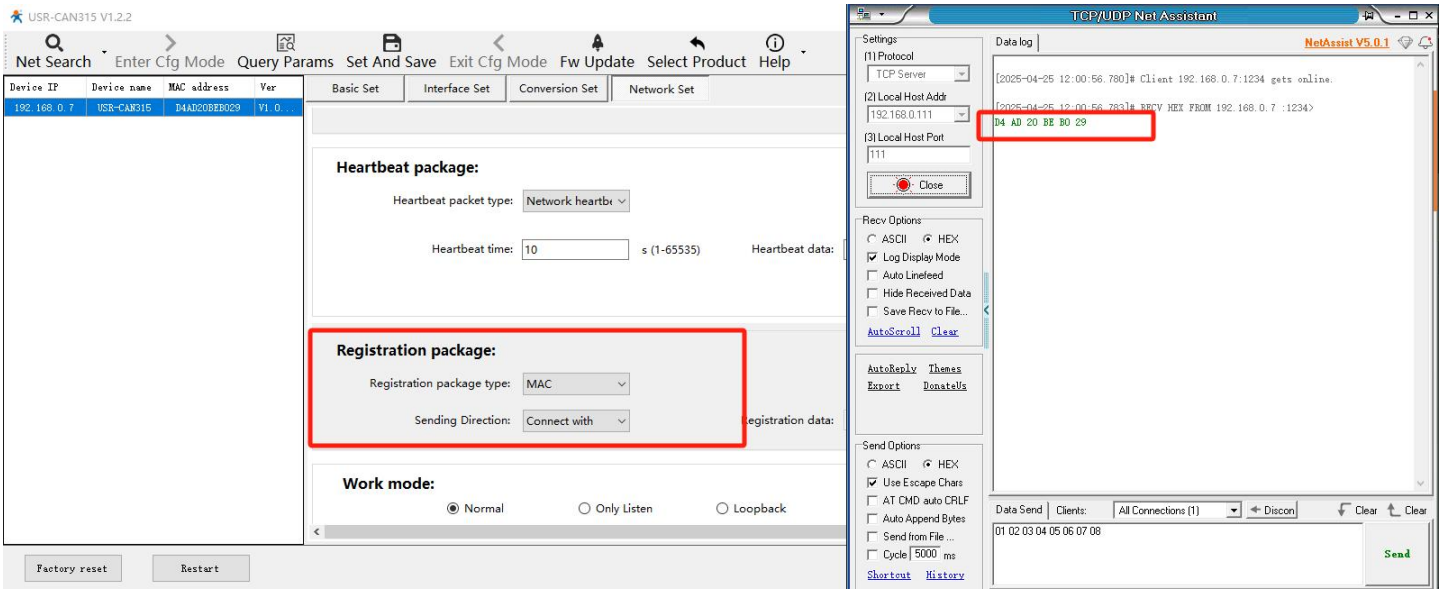


Pic22 Diagram of connection sending registration packet

Data carrying registration package: The data is connected to the registration package at the front end of the data, which is mainly used for protocol transmission. The application diagram is as follows:



Pic23 Attach a schematic diagram of the registration package



Pic24 Registration package setting and data transmission diagram

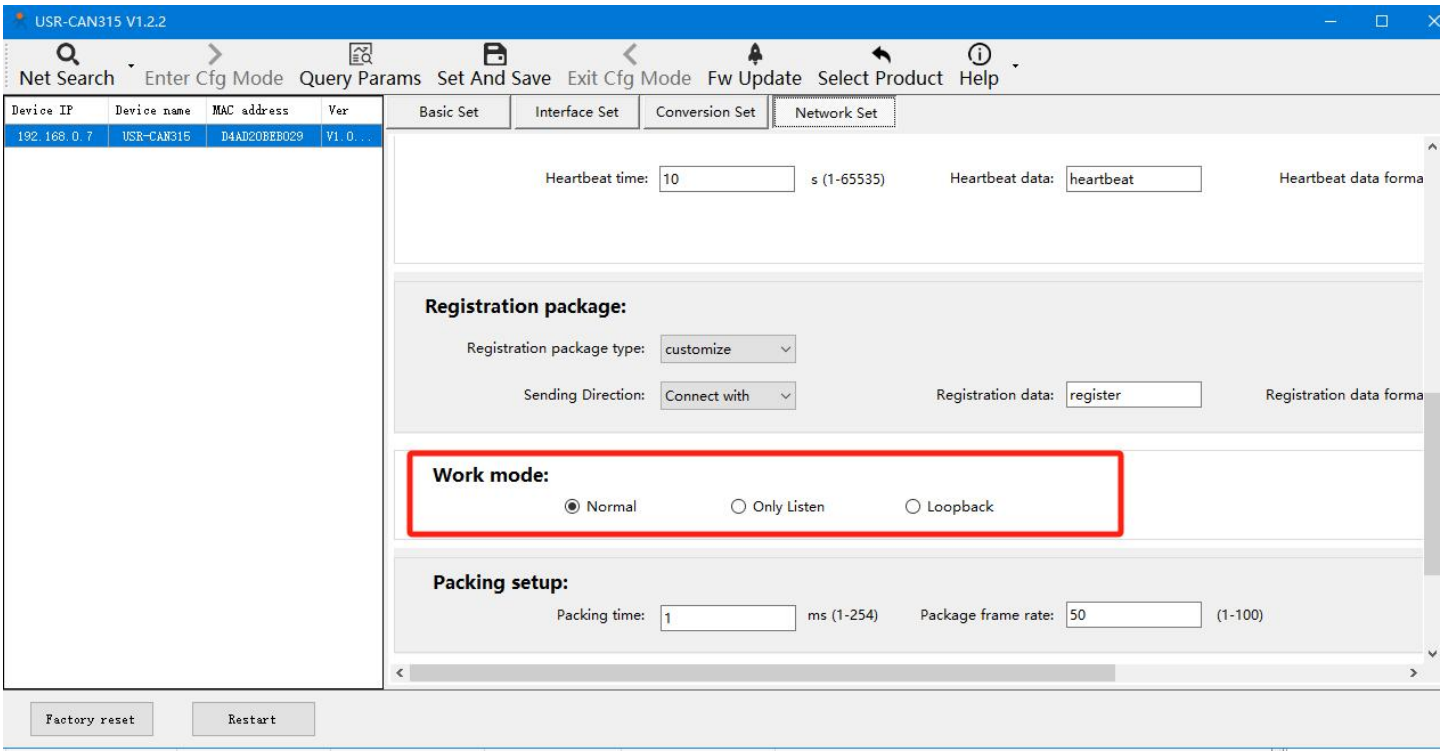
3.7.3. Can Work Pattern

CAN supports three modes of operation: Normal, listen only and loopback.

In normal mode, data CAN be received and sent normally;

Only in listen mode, the CAN port works in listen mode and does not respond;

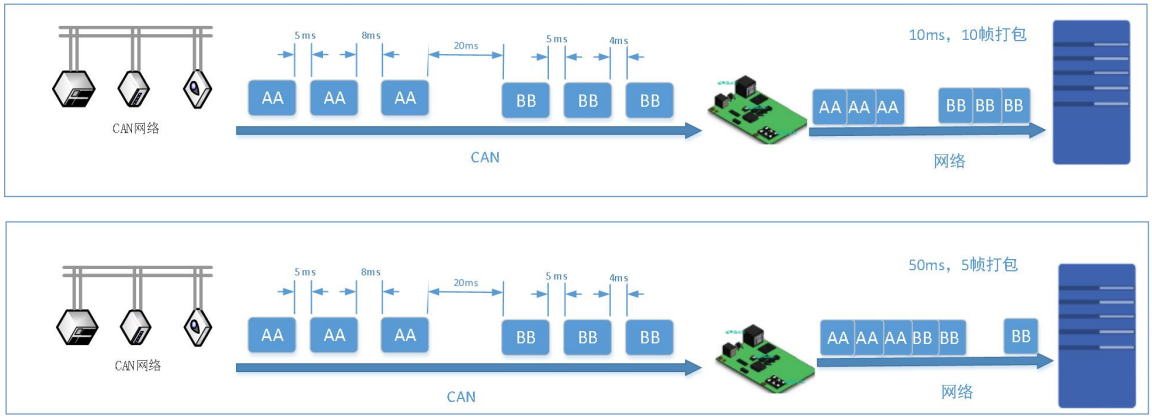
In the loopback mode, the data sent will be received by itself and also transmitted to the CAN bus, but the data CANnot be sent to the module. This mode is mainly used for testing.



Pic25 Work mode setting diagram

3.7.4. Can Packaging Mechanism

Since the data on the network is transmitted in units of data frames, it is necessary to send the CAN data frame data to the network end, so that the data CAN be transmitted more efficiently and quickly. CAN315 CAN pack the data received by CAN according to the packing time and the number of packed frames.

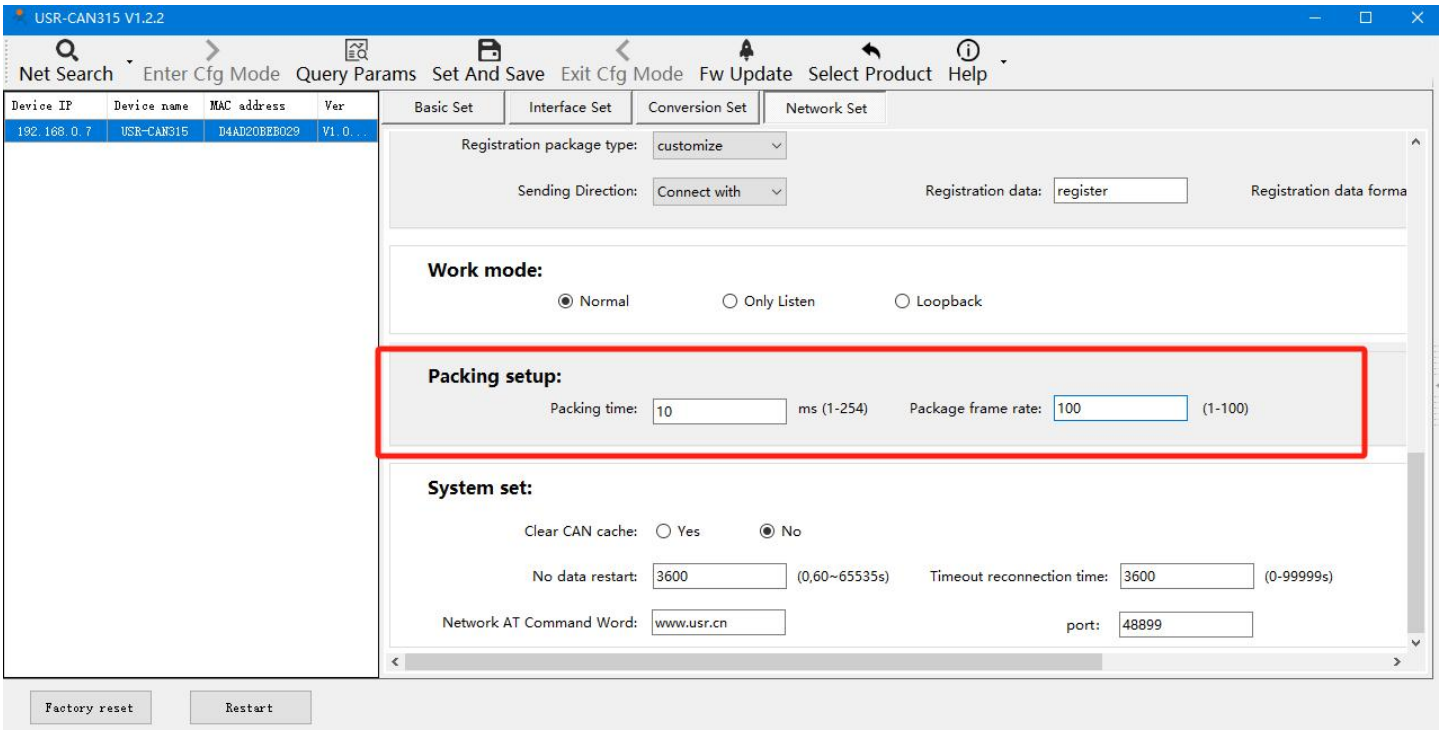


Pic26 CAN packaging instructions

The CAN packaging mechanism is based on the packaging time and packaging length. When either of them meets the requirements, the package is sent.

Packaging time: The default is 10ms, which CAN be set in the range of 1~254.

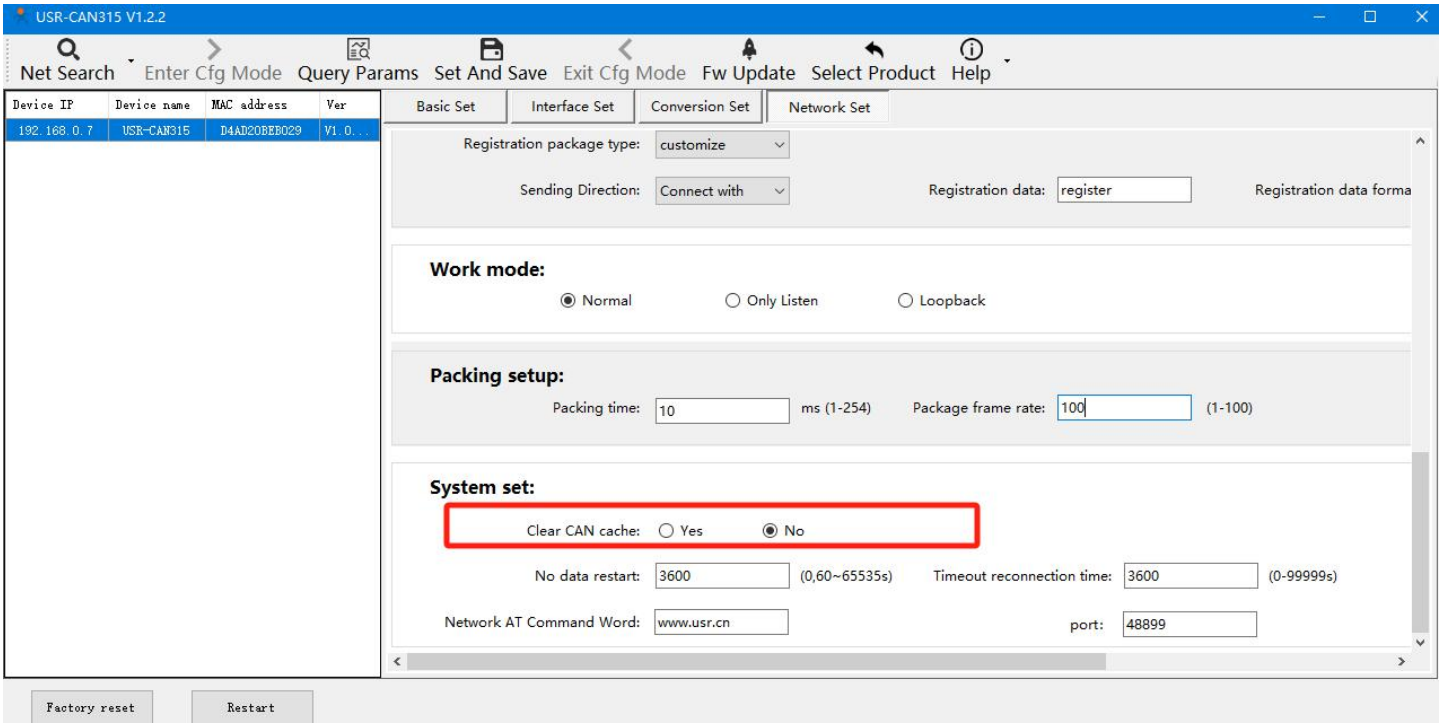
Pack length: The default is 100 frames. It CAN be set from 1 to 100.



Pic27 Subcontract time setting diagram

3.7.5. Clear Can Cache Data

When a tcp connection is not established, data received by the CAN port will be placed in the buffer, with a maximum of 200 frames in the buffer. Once a tcp connection is established, the CAN port CAN set whether to clear the buffer data based on customer needs. This function is default non-clearing. In tcp client mode, when short connection function is enabled, the clear buffer data function becomes ineffective. The configuration diagram is as follows:



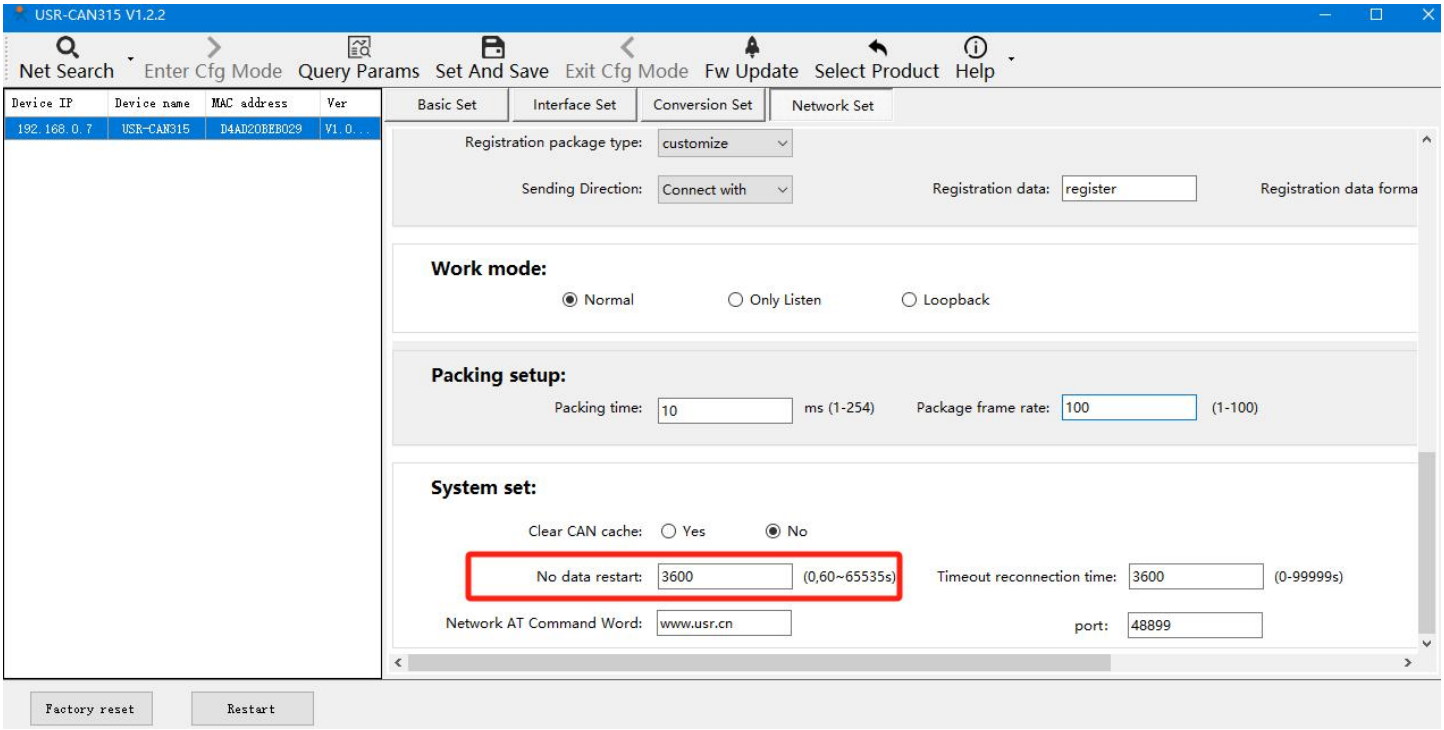
Pic28 Clear the cache data settings diagram

3.7.6. No Data Restart

The data-free reboot (timeout reboot) function is primarily used to ensure the long-term stable operation of the CAN315.

When the network port fails to receive data for an extended period or the network does not receive data for a prolonged time, the CAN315 will reboot after exceeding the set time, thus preventing abnormal conditions from affecting communication. The normal working time for this function is set between 60 to 65535 seconds, with a default value of 3600 seconds. If the setting time is less than 60 seconds, it defaults to zero, effectively disabling the function.

The schematic diagram is as follows:



Pic29 No data restart function

3.8. Firmware upgrade

Support the device to easily achieve firmware upgrade through the upper computer. If you need firmware upgrade, please seek technical support to obtain the latest firmware, do not operate at will. For details of firmware upgrade, see CANFD series product firmware upgrade manual.

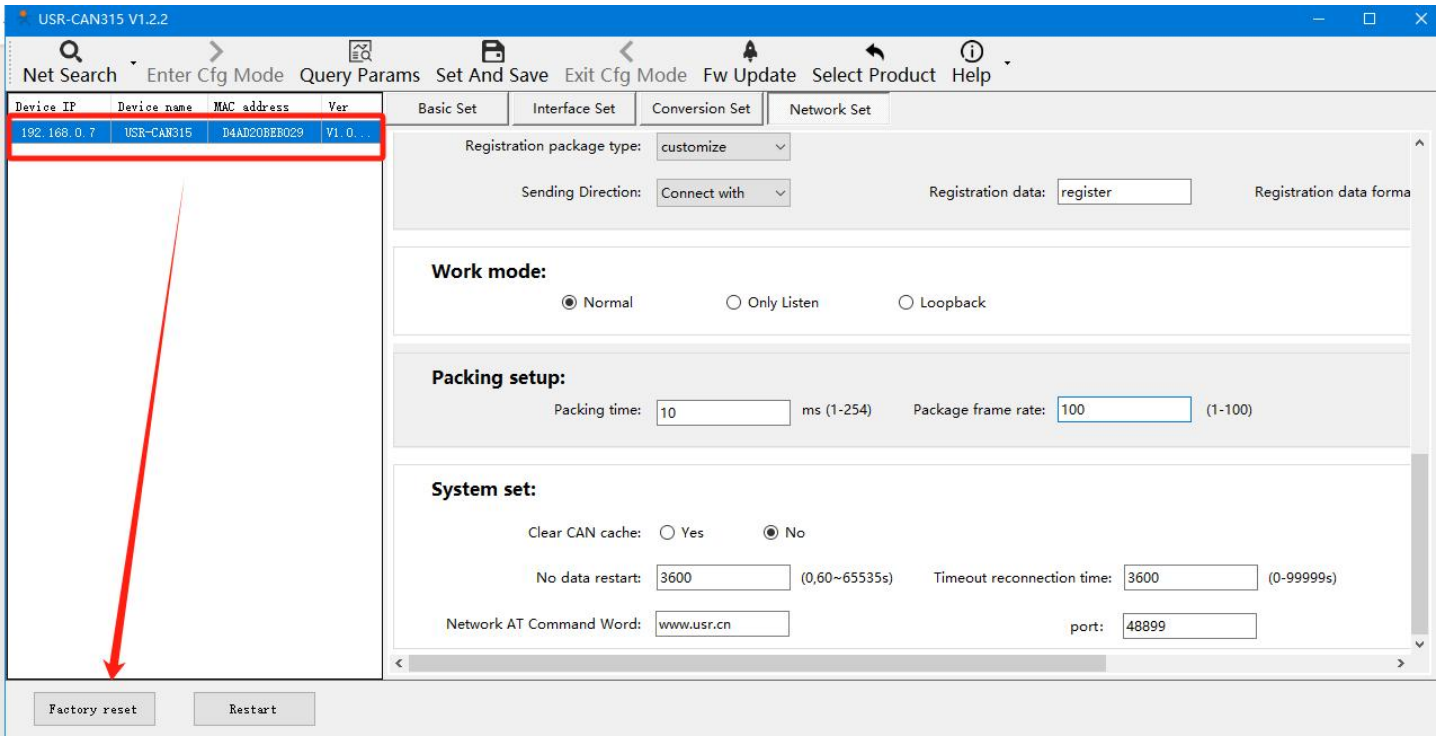
3.9. Factory data reset

Hardware restore to factory settings: The module CAN restore to factory settings through hardware. After power on, press the reload button, keep the reload pressed state and release it after 3-15s, and the hardware will be restored to factory settings.

Software restore factory settings: The software CAN be restored to factory settings by setting the software.

At command to restore factory settings: In at command mode, send the command at+clear, press enter, and receive the correct reply +ok, then the factory settings are restored.

Set software settings:



Pic30 Set up the software diagram

4. Conversion mode example

The device supports five conversion modes: Transparent conversion, transparent with id conversion, standard protocol conversion, modbus protocol conversion, and frame header and footer conversion. All five modes apply under tcp server, tcp client, UDP server, and UDP client protocols, making data conversion more flexible. Below are detailed examples of three conversion modes.

4.1. Transparent conversion

Under the transparent conversion method, the CAN315 immediately converts and sends data received from one bus to the other without adding any data or making any modifications. This not only achieves the exchange of data formats but also preserves the content of the data, making the converter appear transparent to both buses.

The frame information (frame type part) and frame id of CAN message are configured by the user in advance, and the frame type and frame id remain unchanged during the conversion. The user CAN choose whether to convert the frame information and frame id.

In this way, the communication burden of users will not be increased, but the data CAN be converted in real time, and the transmission of data with large traffic CAN be undertaken.

4.1.1. Ethernet to CAN (FD) —— transparent conversion

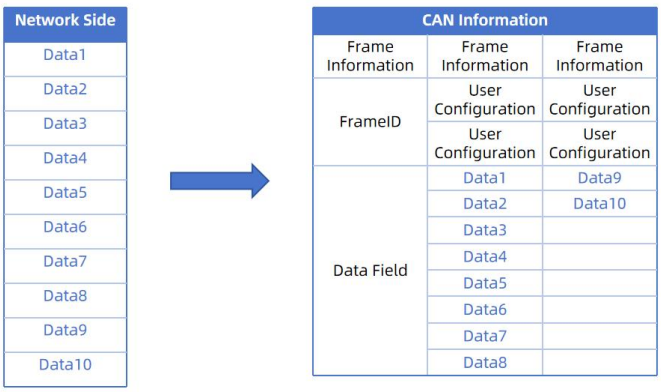
Ethernet frame to CAN message: All data of the serial frame is sequentially filled into the data field of the CAN message frame. The converter immediately receives and converts the data as soon as it detects data on the serial bus. Frame information (frame type part) and frame id need to be preconfigured.

CAN mode: Maximum 8 bytes per frame.

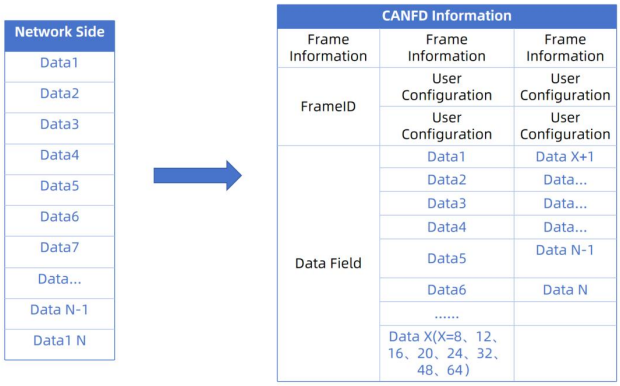
CANFD mode: The maximum data length of each frame is 64 bytes. Note: When the data length exceeds 8 bytes, it must be in line with the dlc that CANFD CAN encode to ensure accurate conversion, that is, the length is 12,16,20,24,32,48,64, otherwise the converter will automatically split it into several CANFD messages of appropriate length.

For example: The data length is 58, and the dlc of CANFD CANnot encode a length of 58; the closest it CAN be is 48. Therefore, a CANFD message with a data length of 48 is converted. The remaining data length is 10, and the closest dlc encoding length is 8. Thus, a CANFD message with a data length of 8 is converted. Finally, the remaining 2 bytes of data are converted into a CANFD message with a data length of 2. In summary, a 58-byte serial frame will be split into 3 CANFD messages.

Ethernet data to CAN



Ethernet data to CANFD



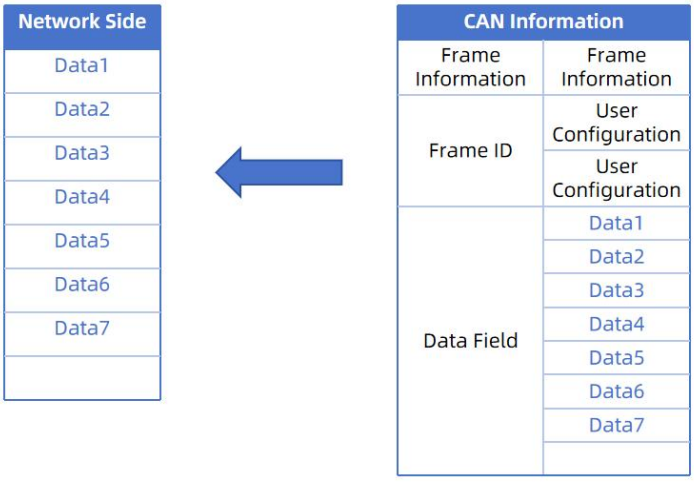
4.1.2. CAN (FD) to Ethernet —— transparent conversion

For CAN bus messages, a frame of CAN (FD) message is also received and immediately forwarded.

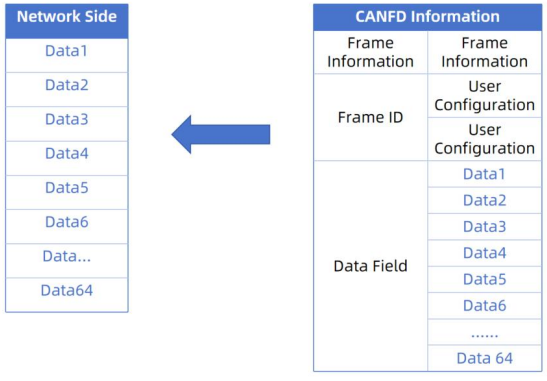
If the enable frame information is enabled, the converter adds the frame information of the CAN (FD) message to the first byte of the serial frame when it is working. The frame information of the CAN (FD) is not converted if it is not selected.

If the enable frame id is enabled, the converter will add the CAN (FD) message frame id to the serial frame frame data before the frame information (such as enable frame information) when it is working.

CAN To Ethernet data (no enabled frame information/enabled frame id)



CANFD To Ethernet data (enable frame information / enable frame id is enabled)



4.2. Transparent band id conversion

Transparent label conversion is a special use of transparent conversion, without an additional protocol. This method converts the "Address" In Ethernet data to the identifier field of a CAN (FD) message, where the frame id CAN be configured at both the starting position and length within the Ethernet data. The converter extracts this frame id during the conversion process and fills it into the frame id field of the CAN (FD) message, serving as the id for the CAN (FD) message when the Ethernet data is forwarded. Similarly, when converting a CAN message back to Ethernet data, the id of the CAN (FD) message is also converted and placed at the corresponding position in the Ethernet data.

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" Setting of the CAN parameter settings in the configuration software is invalid because the identifier sent (frame id) is filled with data from the Ethernet data above.

4.2.1. Ethernet to CAN (FD) —— transparent band id conversion

Specify the CAN (FD) frame type and the starting address and length of the "Frame id" Of the CAN (FD) message in the Ethernet data. The starting address ranges from 0 to 7; the length ranges from standard frame: 1 to 2, extended frame: 1 to 4.

During the conversion, according to the pre-configured configuration, the "Frame id" Of the CAN (FD) message in Ethernet data will be fully converted into the frame id domain of the CAN (FD). If the configured transparent id length is shorter than the frame type id length of the CAN (FD) message, the high byte of the frame id in the CAN (FD) message will be padded with 0.

Ethernet data to CAN

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 is converted 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

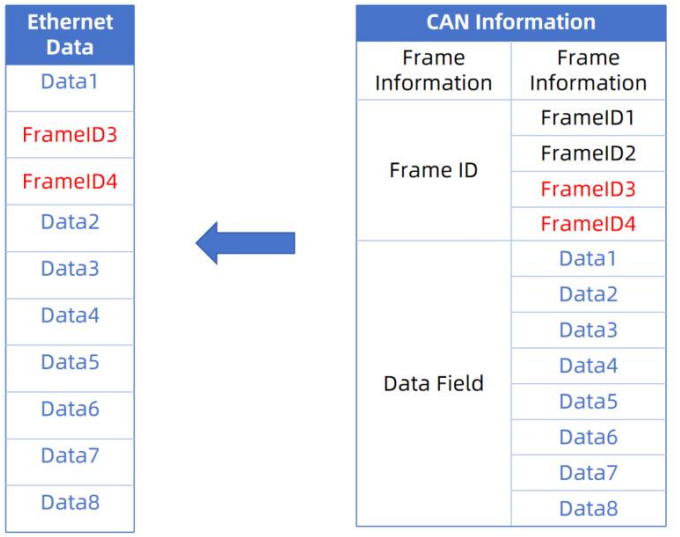
CANFD Information		
Frame Information	Frame Information	Frame Information
Frame ID	ID1 (0x00)	ID1 (0x00)
	ID2 (0x00)	ID2 (0x00)
	Data2 (ID3)	Data2 (ID3)
	Data3 (ID4)	Data3 (ID4)
Data Field	Data1	Data...
	Data4	Data N-1
	Data5	Data N
	Data6	
	Data7	
	.....	
	Data65	
	Data66	

4.2.2. CAN (FD) is transparent to Ethernet —— with id conversion

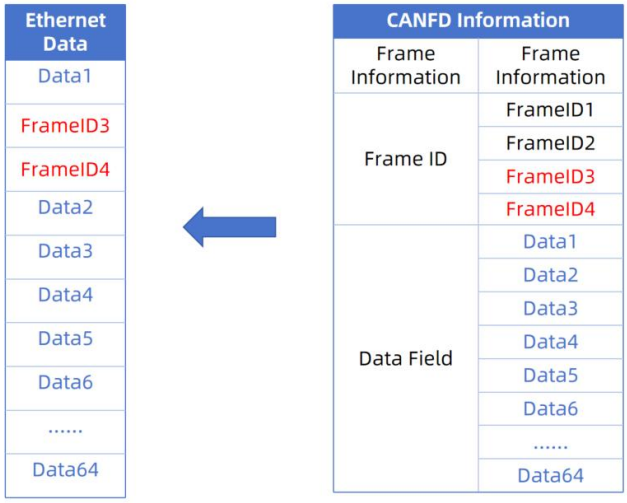
For CAN (FD) packets, a frame is forwarded immediately upon receipt. Each time the frame is forwarded, the id of the received CAN (FD) packet is converted according to the position and length of the CAN (FD) frame id in the Ethernet data that is pre-configured. Other data are forwarded in sequence.

Note: If the length of CAN (FD) frame id is greater than the length of the set transparent band id, only the lower byte is converted to the corresponding position in the serial frame. For example, if the CAN frame id is 01020304 and the length of the set transparent band id is 2, only 0304 is converted to the corresponding position in the serial frame.

CAN To Ethernet data



CANFD To Ethernet data



4.3. Standard protocol conversion

Standard CAN frame format, each CAN frame contains 13 bytes, 13 bytes content includes CAN frame information (1 byte) + frame id (4 bytes) + data frame (8 bytes).

Standard CANFD frame format, each CANFD frame contains 69 bytes, 69 bytes content includes CANFD frame information (1 byte) + frame id (4 bytes) + data frame (64 bytes).

By properly configuring the frame information (the data in the first byte), you CAN flexibly send standard frames, extended frames, and even remote frames. By properly parsing the serial frame, you CAN get the details of the standard frame, extended frame, and even remote frame.

Pay attention to:

（1）In this conversion mode, the "CAN id" And "Frame type" Of the configuration software are invalid, because the frame id sent at this time is filled by the frame id data in the serial frame mentioned above, and the frame type is determined by the frame information in the serial frame.

（2）In this mode, the serial data format must be strictly followed to successfully convert. It is necessary to ensure that the frame information is correct and the reserved bit is zero. CAN frame is fixed length of 13 bytes, CANFD frame is fixed length of 69 bytes, and the frame must be padded with 0 if it is less than 69 bytes, otherwise it CANnot be transmitted.

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

CANFD fixed format (one 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, frame length.

Bit7	Bi6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Ff	Rtr	Edl	Brs	Dlc3	Dlc2	Dlc1	Dlc0

Ff: The identification bit of standard frame and extended frame. 1 is the extended frame, and 0 is the standard frame.

Rtr: Identifier of remote frame and data frame. 1 is for remote frame, 0 is for data frame, and CAN only be 0 in CANFD mode.

Edl: CAN and CANFD identification, 0 for CAN, 1 for CANFD.

Brs: Port rate switching enable identifier, 0 is not to convert the rate, 1 is to convert the variable rate, only valid when CANFD. This bit should be 0 in CAN.

Dlc3~dlc0: Data length bits, indicating the data length of the CAN (FD) frame.

Frame id: 4 bytes in length; high bits first, low bits last. The standard frame is 11 bits effective and the extended frame is 29 bits effective.

12h	34h	56h	78h

Extended frame id: 0x12345678

00h	00h	01h	23h

This id CAN be used to represent both extended frame id and standard frame id

Extended frame id: 0x00000123

Standard frame id: 0x0123

Extended frame and standard frame id are distinguished by frame information

Frame data: CAN mode length is 8 bytes, CANFD mode length is 64 bytes, and 00 must be added if it is insufficient.

4.4. Modbus protocol conversion

Modbus protocol conversion CAN convert standard modbus data protocol into specific CAN (FD) data format. This kind of conversion generally requires that CAN (FD) bus device message CAN be edited.

On the CAN side, a simple and easy-to-use segmented communication format has been established to implement modbus communication. The converter still plays the role of protocol verification and forwarding. It supports the transmission of modbus protocols, not the host or slave of modbus, allowing users to communicate according to the modbus protocol.

Pay attention to:

（1）In this conversion mode, the "CAN id" Item of the "CAN parameters" Item in the configuration software is invalid because the frame id sent at this time is filled by the address field in the modbus rtu serial frame.

4.4.1. Frame format

（1）Serial frame

The serial interface uses the standard modbus protocol, so the user frame conforms to this protocol.

(2) CAN frame

CAN side has designed a set of segment protocol format, which defines a segment and the method of reassembly, as shown below. The CAN frame information (remote frame or data frame; standard frame or extended frame) is set by configuration software.

In CAN mode, the content of the transmitted modbus protocol CAN start from "Data 2" Bytes. If the content of the protocol is greater than 7 bytes, the remaining protocol content will continue to be converted in this segmentation format until the conversion is complete.

Bit Number	7	6	5	4	3	2	1	0
Frame Information	FF	RTR	EDL	BRS	DLC(Data Length)			
FrameID1	X	X	X	ID.28-ID.24				
FrameID2	ID.23-ID.16							
FrameID3	ID.15-ID.8							
FrameID4	ID.7-ID.0（Modbus RTU Address Code）							
Data1	Segment Marker	Segment Type	Segment Counte					
Data2	Character1							
Data3	Character2							
Data4	Character3							
Data5	Character4							
Data6	Character5							
Data7	Character6							
Data8	Character7							

Pic31 Segmented protocol in CAN mode

In CANFD mode, the content of the transmitted modbus protocol CAN start from "Data2" Bytes. If the content of the protocol is greater than 63 bytes, the remaining protocol content will continue to be converted in this segmentation format until the conversion is complete.

Bit Number	7	6	5	4	3	2	1	0
Frame Information	FF	RTR	EDL	BRS	DLC(Data Length)			
FrameID1	X	X	X	ID.28-ID.24				
FrameID2	ID.23-ID.16							
FrameID3	ID.15-ID.8							
FrameID4	ID.7-ID.0 (Modbus RTU Address Code)							
Data1	Segment Marker	Segment Type	Segment Counte					
Data2	Character1							
Data3	Character2							
Data4	Character3							
Data5	Character4							
Data...	.....							
Data63	Character62							
Data64	Character63							

Pic32 Segmented protocol in CANFD mode

Data 1 is a segment control information (1 byte, 8 bit), which means as follows:

- Segmentation mark: Occupies 1 bit (bit7), indicating whether the message is a segmentation message. If this bit is 0, it indicates a separate message; if it is 1, it indicates a frame belonging to the segmented message.
- Segment type: 2 bits (bit6, bit5) are used to indicate the type of the message in the segment message:

Place value	Meaning
00	First paragraph
01	Intermediate segmentation
10	The last paragraph

- Segment counter: Occupies 5 bits (bit4~bit0), indicating the number of this segment in the whole message. If it is the nth segment, then the value of the counter is n. In this way, it CAN be verified whether any segment has been lost during reception.

4.4.2. Conversion mode

(1) modbus to CAN (FD):

The address field of the modbus protocol is converted into the id4 (extended frame) or 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 to the CAN message, and the CAN message does not need to have the check byte of the serial frame, because the CAN bus itself has a good check mechanism.

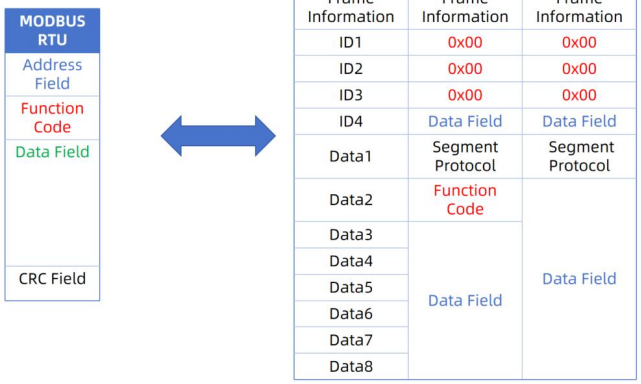
The conversion involves the protocol content of modbus, the function code, and the data field. During the conversion, they are sequentially transferred into the data field of a CAN message frame (starting from the second data byte; the first data byte is used for segmenting protocols). Since the length of modbus frames varies depending on the function code, and a single CAN message frame CAN only transmit 7 bytes, while a single CANFD message frame CAN transmit 63 bytes, the converter segments longer modbus frames and converts them into CAN messages, which are then transmitted using the aforementioned CAN segment protocol. Users CAN receive and process the function code and data field at CAN nodes.

(2) CAN (FD) to modbus:

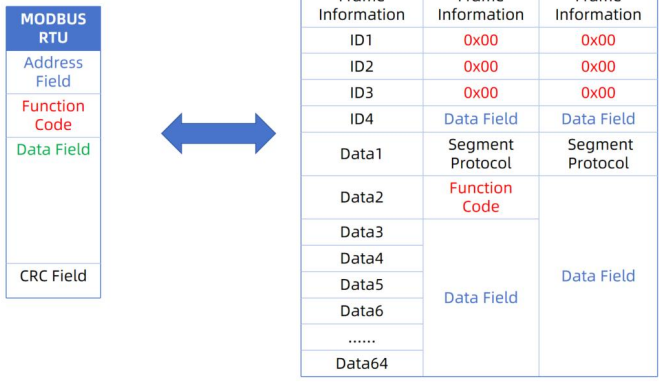
For CAN, the bus modbus protocol data does not need to do cyclic redundancy check (crc16). The converter receives according to the segmented protocol, and automatically adds cyclic redundancy check (crc16) after receiving a frame and parsing, and converts it into modbus frame to send to the serial bus.

Note: If the received data does not conform to the segmentation protocol, the group of data will be discarded and not converted.

In CAN mode, the modbus protocol is converted



In CANFD mode, the modbus protocol is converted



Take CAN mode as an example:

Serial port send: 01 03 14 00 0a 00 00 00 00 00 14 00 00 00 00 00 17 00 2c 00 37 00 c8 4e 35

01 as the modbus address code, it is converted into CAN's id.7-id.0, and the last two bytes (4e 35) are the crc check of modbus rtu, which is not converted.

CAN port reception:

Frame 1 CAN message: 81 03 14 00 0a 00 00 00

Frame 2 CAN message: A2 00 00 14 00 00 00 00

Frame 3 CAN message: A3 00 17 00 2c 00 37 00

Frame 4 CAN message: C4 c8

4.5. Customize the frame header and frame tail

In order to make it convenient for users to use CAN-bus, the serial frame format is close to the CAN frame format. The start and end of a frame in the serial frame, namely "Frame head" And "Frame tail", are specified, which CAN be configured by users themselves.

4.5.1. Ethernet to CAN (FD) —— custom frame header and tail conversion

The serial frame format must conform to the specified frame format, otherwise it CAN not be transmitted correctly. The serial frame must contain: Frame header, data length, data domain, and frame tail.

The frame header and frame tail are customized by the customer, 1 byte.

Data length refers to the byte length of the data field. The data length must match the frame end data for correct transmission; otherwise, it will be discarded. For example: If the frame header is configured as aa and the frame end as ff, a serial frame aa 03 01 02 04 ff CAN be transmitted normally. If the serial frame sends aa 0301 02 0304 ff, and after the data field 01 02 03 comes 04 instead of the frame end ff, the frame will be discarded and CAN not be transmitted.

In transparent conversion and custom protocol conversion, CAN id and CAN types need to be configured by themselves. Frame header, frame tail, and data length are not converted into CAN frames.

Ethernet data to CAN

Ethernet frame
Frame Header
Data Length
Data1
Data2
Data3
Data4
...
Data N-1
Data N
Frame Tail

CAN Information		
Frame Information	Frame Information	Frame Information
Frame ID	User Configuration	User Configuration
	User Configuration	User Configuration
Data Field	Data1	Data9
	Data2	...
	Data3	Data N-1
	Data4	Data N
	Data5	
	Data6	
	Data7	
	Data8	

Ethernet data to CANFD

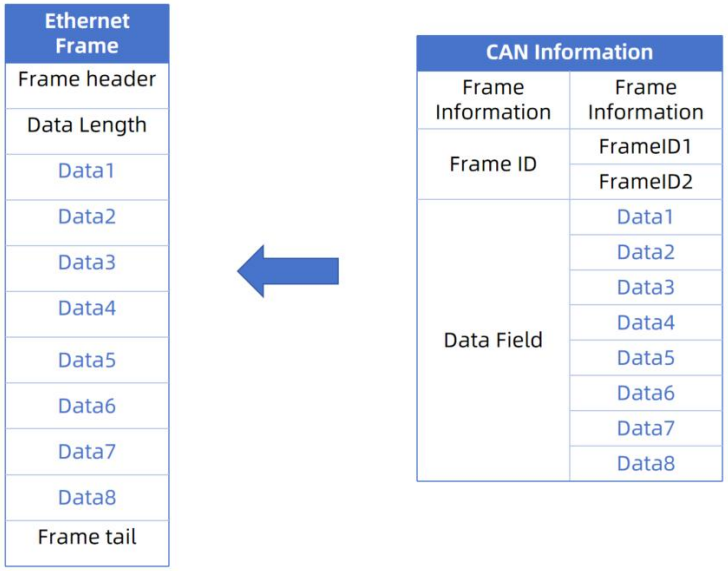
Ethernet frame
Frame Header
Data Length
Data1
Data2
Data3
Data4
...
Data N-1
Data N
Frame Tail

CANFD Information		
Frame Information	Frame Information	Frame Information
Frame ID	User Configuration	User Configuration
	User Configuration	User Configuration
Data Field	Data1	Data9
	Data2	...
	Data3	Data N-1
	Data4	Data N
	Data5	
	Data6	
	.....	
	Data64	

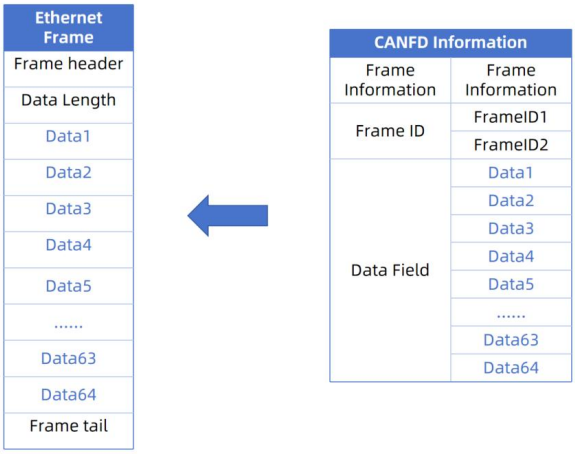
4.5.2. Ethernet to CAN (FD) —— custom frame header and tail conversion

The CAN (FD) bus message receives one frame and forwards one frame. The module will convert the data in the CAN message data field one by one, and automatically add frame header, frame length, frame information and other data to the serial frame, which is actually the reverse form of serial frame to CAN message.

CAN To Ethernet data



CANFD To Ethernet data



5. Contact way

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

6. Disclaimer

This document provides information about the USR-CAN315 series of products. This document does not grant any intellectual property rights and does not grant any intellectual property rights by express or implied terms, nor by prohibiting speech or other means. Except for the liability stated in the sales terms and conditions of its products, our company assumes no other responsibilities. Furthermore, our company makes no express or implied warranties regarding the sale and/or use of this product, including its suitability for specific purposes, merchantability, or liability for infringement of any patent, copyright, or other intellectual property rights. Our company may modify the product specifications and descriptions at any time without prior notice.

7. Update history

File version	Update content	Refresh time
V1.0	First edition	2025-06-10

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