PUSR®
www.pusr.com

Communication Expert of
Industrial IOT

CANFD To RS485/RS232

USR-CAN316

User Manual

Be Honest & Do Best

Your Trustworthy Smart Industrial IoT Partner

Functional characteristics

- CAN (FD) and RS485/RS232 bidirectional conversion
- Supports CANFD protocol and is compatible with CAN2.0A and CAN2.0B standard protocols
- Support transparent conversion, transparent band ID conversion, protocol conversion, MODBUS conversion, and custom protocol conversion
- 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
- Multi-master multi-slave function, using a single CAN interface instead of multiple 485
- Supports custom baud rate, which can be calculated by the upper computer baud rate calculator
- Support upper computer parameter configuration
- Support serial AT command configuration
- Supports CAN (FD) to Modbus RTU (master/slave)
- Supports 64 sending messages and 64 receiving messages
- Support the upgrade of firmware on the host, 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 someone IoT. This series of products features high speed, low latency, stable performance, easy use, and high 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 and 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, making installation convenient and quick.

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

This manual mainly introduces the product functions of the serial port version of USR-CAN316.

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 9~36 V, 12V 1A is recommended
	Size	110*27*76.1mm
	Way To Install	Railway installation
	Reload Key	Long press to restore factory Settings

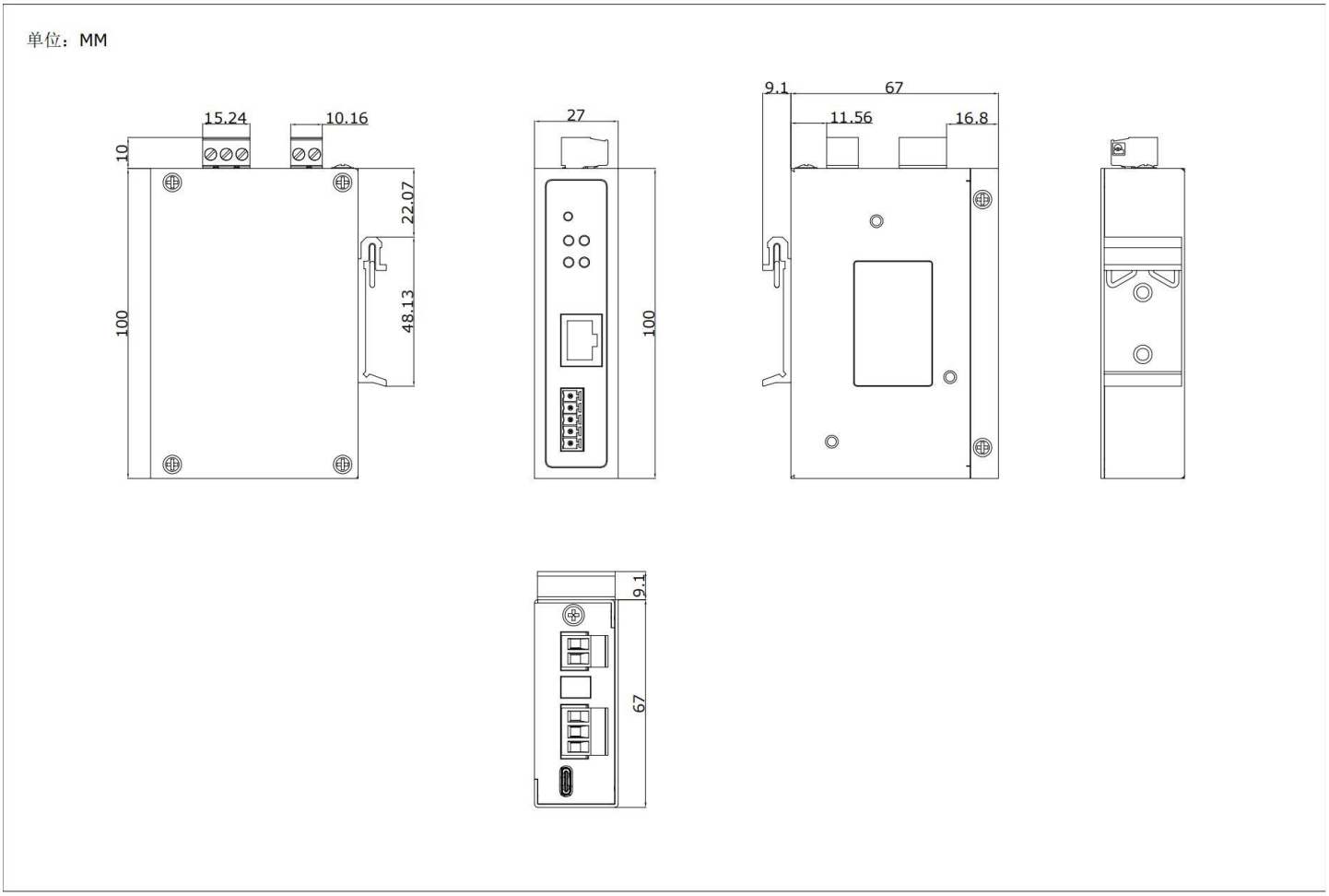
	Pilot Lamp	POWER,WORK,COM,CAN
Interface Parameters	Can Port Specification	One CAN port, supports CANFD, compatible with CAN 2.0A/2.0B
	Can Port Baud Rate	Arbitration domain baud rate: 5K~1Mbps; data domain baud rate 100K~5Mbps
	Terminal Resistance	Built-in 2 CAN bus 120Ω terminal resistor. By using the code control resistor access, turn any one of the codes to ON, and connect a 120Ω resistor in parallel. Turn both dials to ON and connect two 120 ohm resistors in parallel
	Serial Port Specifications	Support one-way RS485 or RS232 access
	Serial Port Baud Rate	600—230.4K(bps)
	Data Bit	8
	Stop Bit	1、2
	Check Bit	Supports None, Odd, Even
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	CanFD Quicken	Support CANFD acceleration mode
	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 extended frames, and custom input frame ID (up to 32 sets)

	serial port packet length	Supports customizing the length of subpackages
	Change Direction	Supports bidirectional conversion, only serial port to CAN, and only CAN to serial port
	Firmware Upgrade	Support the upper computer to upgrade firmware
	Parameter Configuration	AT command, upper computer software configuration
Protection Parameters	Electrostatic Protection	Air discharge 8KV, contact discharge 6KV
	EFT/Burst	Power Circuit 2Kv, RS485/RS232 Circuit 1Kv, Can Circuit 1Kv
	Surge Immunity Test	Power supply circuit: Differential mode 1 kV, common mode 2 kV. CAN circuit: Common mode 2 kV.

2. Hardware parameters

2.1. Size description

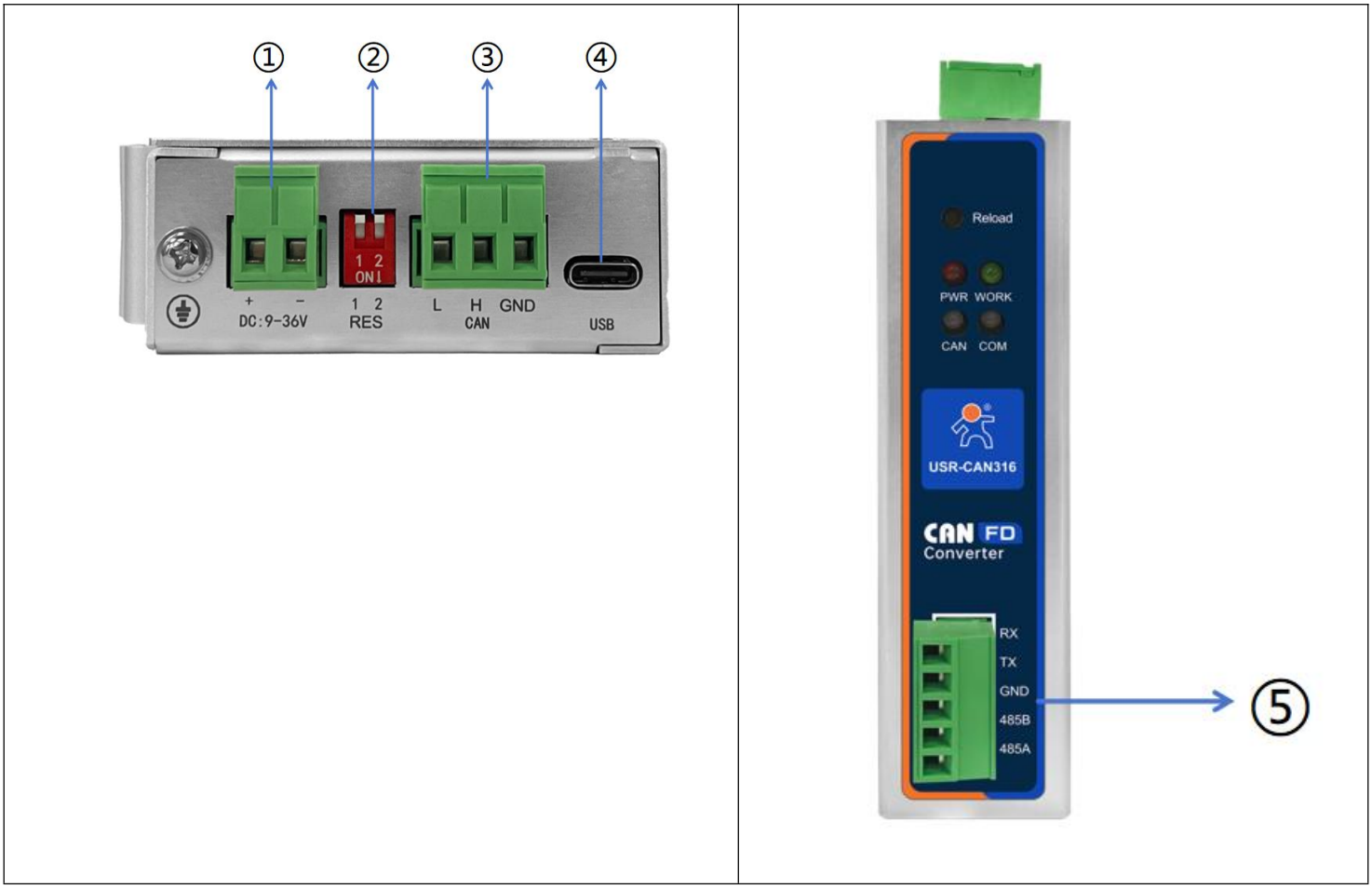
Machine size (including terminals and rails): 110*27*76.1mm



Pic1 USR-CAN316 standard size diagram

2.2. Interface description

The USR-CAN316 interface is described as follows.



Pic2 Interface Specification

Tab 3 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
2	RES 1	Terminal resistance 1,120Ω. Turn it down to ON and connect the resistance in parallel to the CAN bus. Default ON
	RES 2	Terminal resistance 2,120Ω. Turn it down to ON and connect the resistance in parallel to the CAN bus. Default ON
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

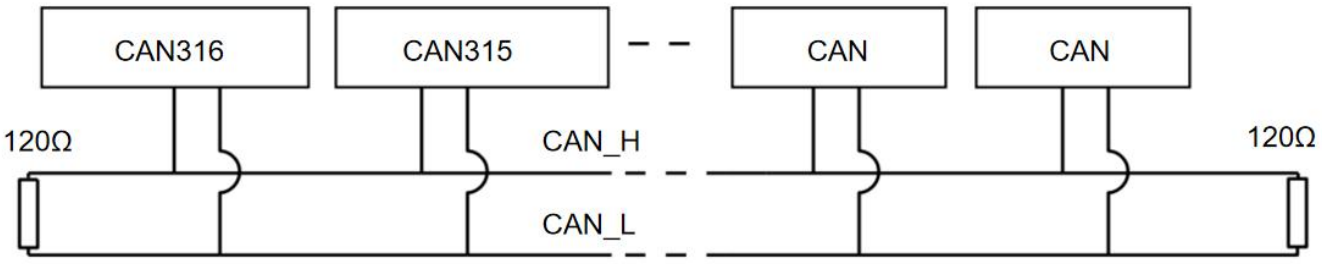
		be performed
5	485A	Serial port interface, RS485 A
	485B	Serial port interface, RS485 B
	GND	Serial port interface, digital ground
	RX	Serial port interface, RS232 receiving end
	TX	Serial port interface, RS232 transmitter

<explain>

When the USR-CAN316 is connected to the CAN bus, CAN_H should be connected to CAN_H and CAN_L to CAN_L.

RES is used for terminal resistance selection. If you dial any code to ON, the 120Ω resistor inside the module will be incorporated into the CAN bus; otherwise, the 120Ω resistor will not be connected to the bus.

According to ISO 11898 standards, to enhance the reliability of CAN-bus communication, two endpoints on 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-CAN316 has four indicator lights: POWER, WORK, COM and CAN. Users can easily observe the status of the device through the indicator lights, which are defined as follows.

Tab 4 Indicator light rules

Pilot Lamp	Pigment	Function Declaration
PWR	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
		Green light flashing: indicates that data is being

Com	Green/Red	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

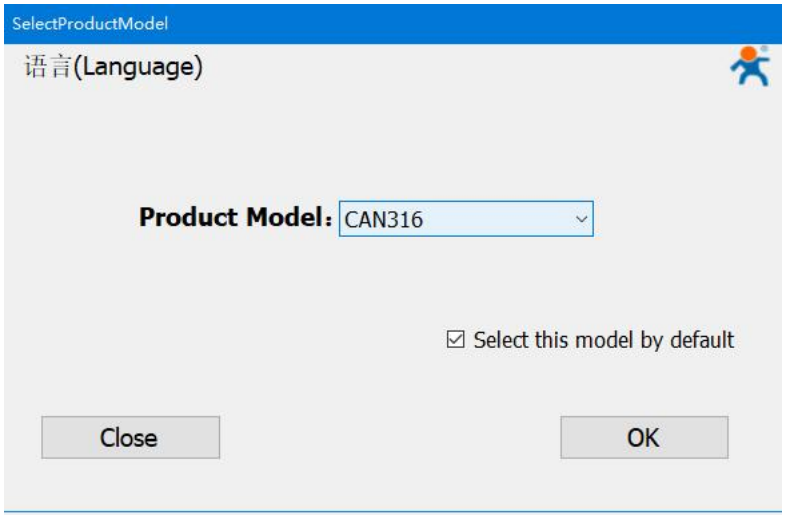
3.1. Function configuration description

CAN316 Supports upper computer configuration parameters and AT command configuration.

Parameters can be configured and queried through AT instructions. For details of specific AT instructions, see CANFD Protocol Converter AT Instruction Set--USR-CAN316

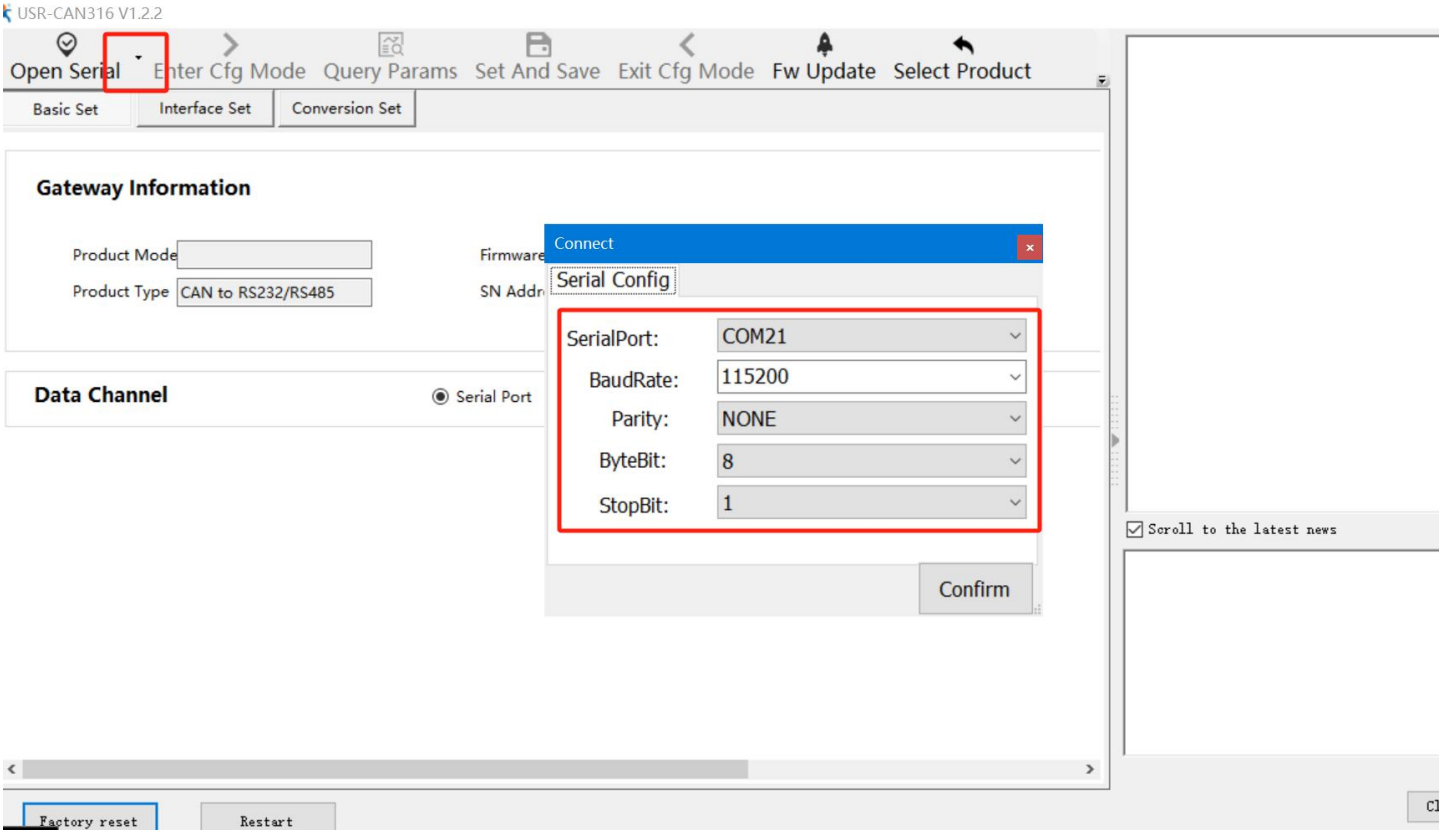
The configuration of the upper computer is simple and easy to use. The following details the configuration parameters of the upper computer.

（1）Download the upper computer from the official website. After opening it, select the model first. For example, CAN316, you can check the default login for this model.



Pic4 Choose the model

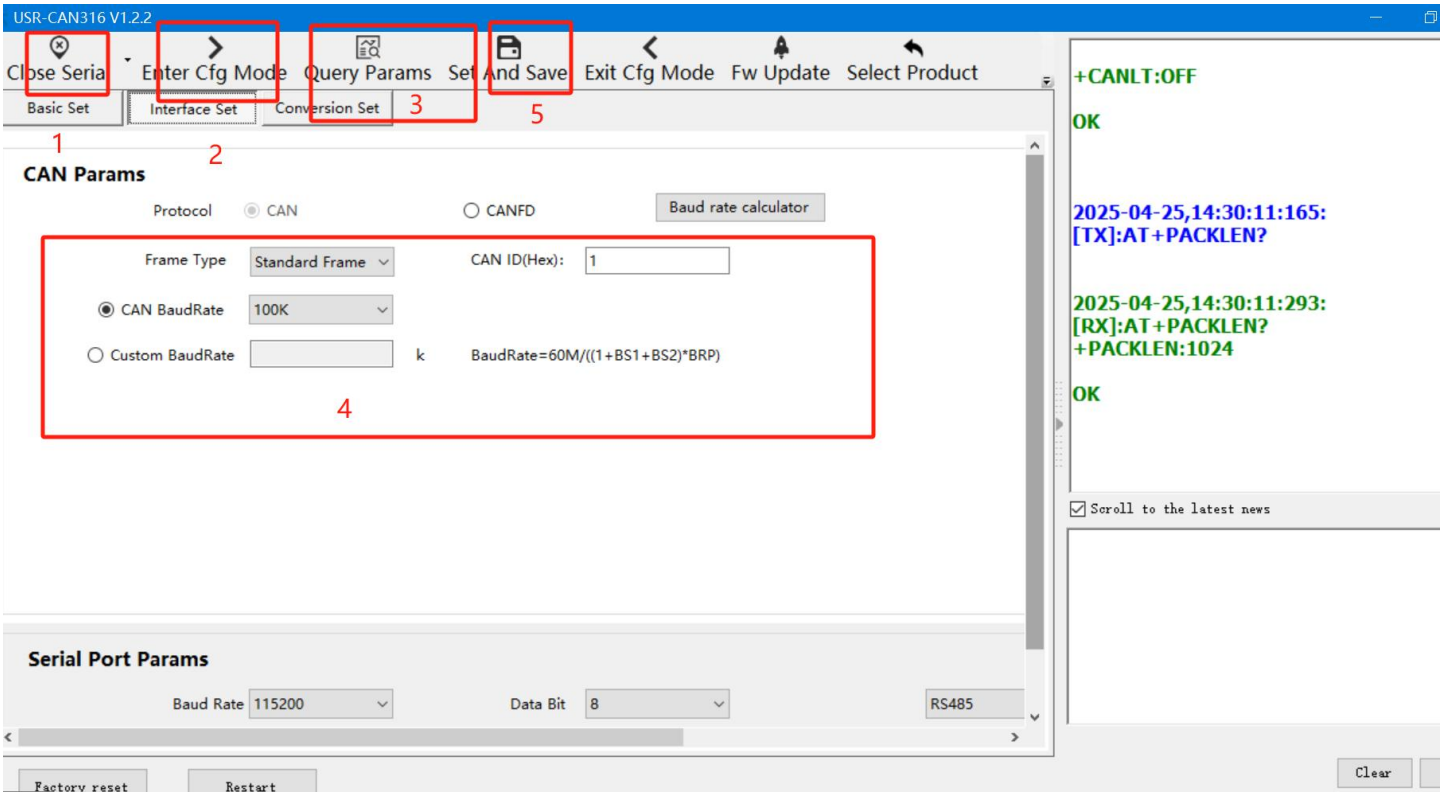
（2）First, the serial port configuration is performed. Click the arrow on the right side of the serial port to open it and configure the serial port parameters. Confirm that the serial port name, baud rate, parity, data bits and stop bits are consistent with the serial port parameters of the CAN316 connected. After the configuration, click confirm.



Pic5 Serial port parameter configuration

（3）Perform parameter configuration:

- Open the serial port, and the message interface on the right displays: [Info]: COM95 serial port has been opened
- Click to enter the configuration state
- Click to read the parameters
- After all the current parameters are read, parameter configuration is performed
- After the configuration is complete, click Settings parameters
- Click restart to make the parameter configuration effective



Pic6 Parameter configuration

3.2. CAN parameter

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

CANFD acceleration: This parameter enables the baud rate switching function of CANFD, which only works 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 is 100kbps. Common baud rates can be directly selected: 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 sample point information. However, if the sample 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 sample point information. The calculation methods for arbitration domain baud rate and data domain baud rate sample points are both:

SMP = 100% * （BS1 + 2） / （(BS2+2) + (BRP + 1)）

baud = CLK / [(BS1+2) + (BS2 + 1)] * (BRP+1)

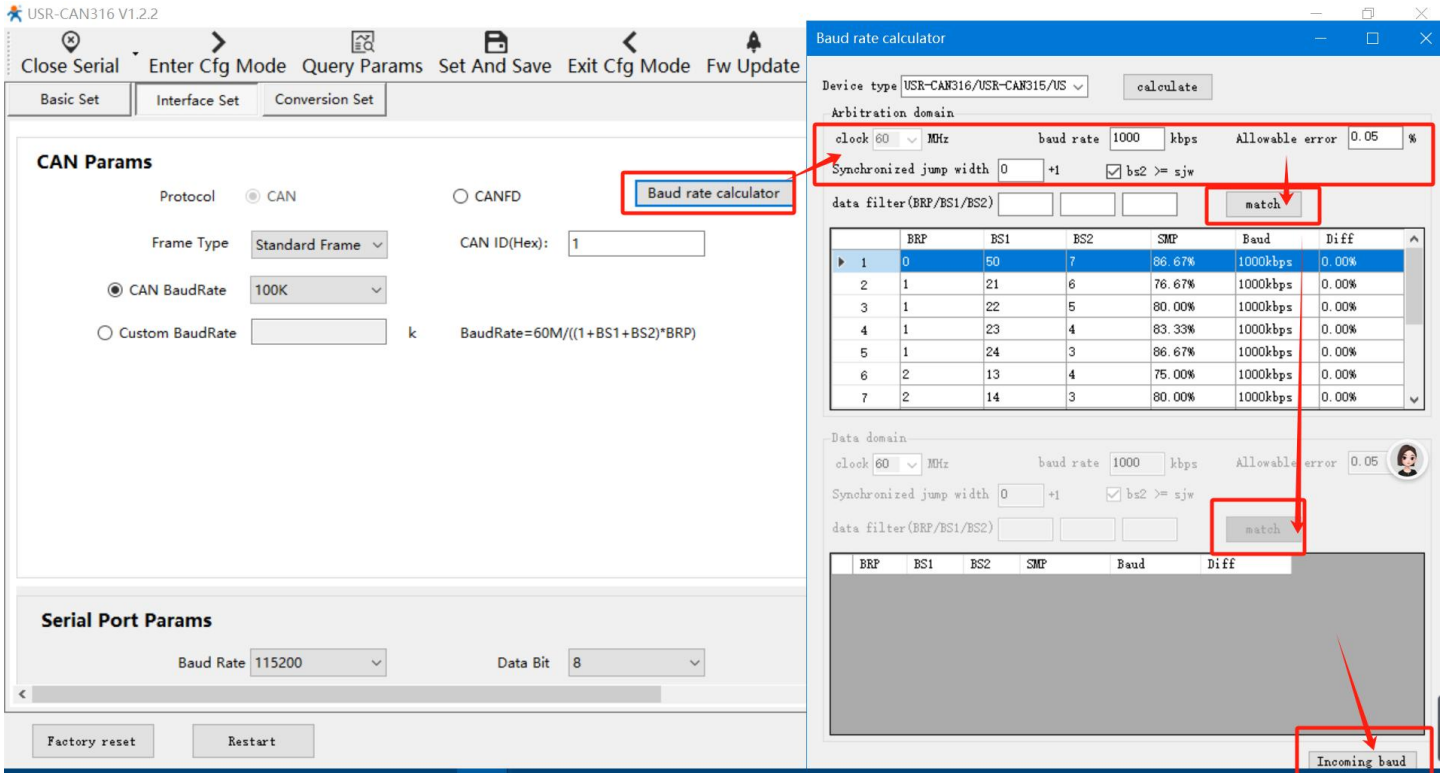
The parameters of the Baud rate sampling point are:

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 can be left out
2) Set the appropriate synchronous jump width (SJW). If synchronous jump is required, 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

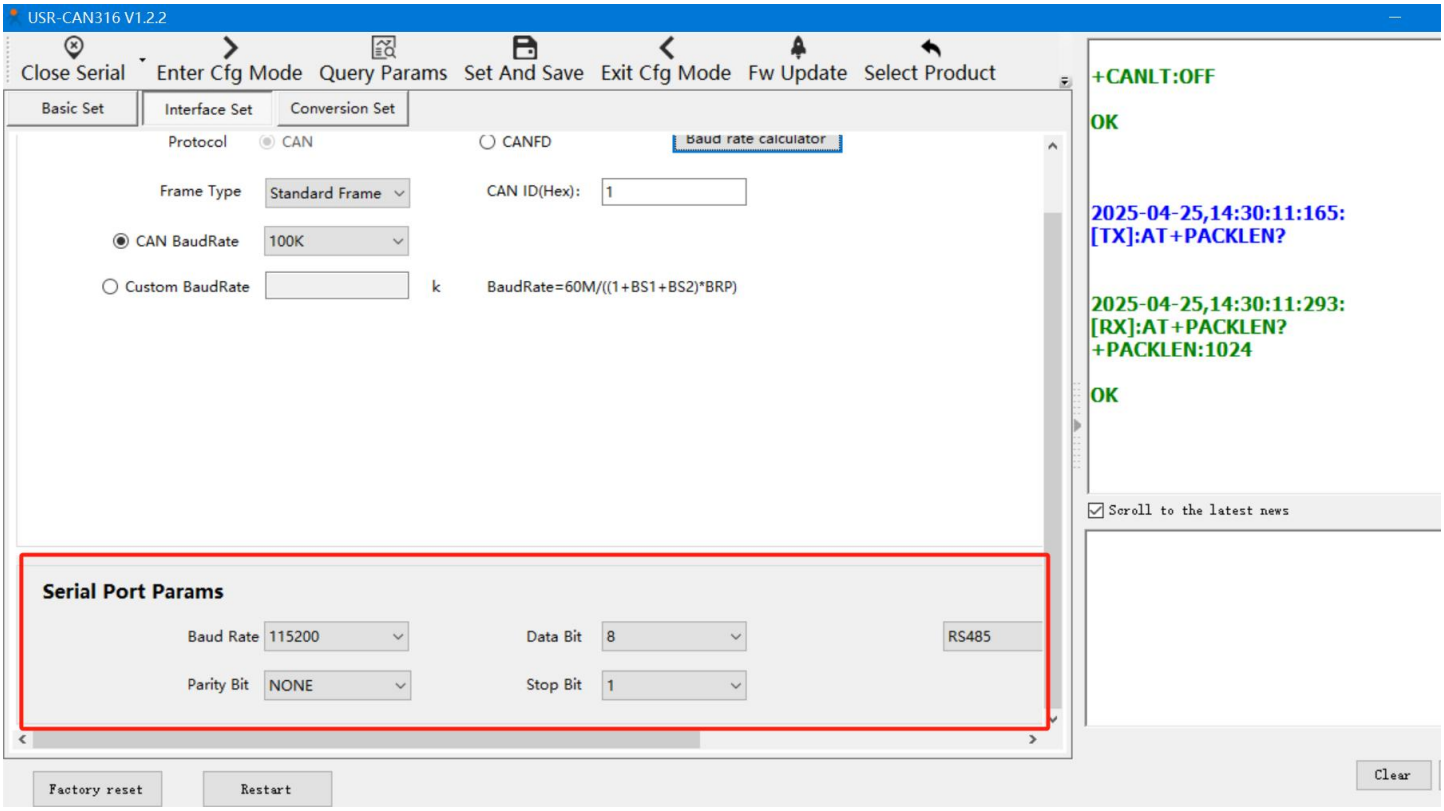


Pic7 Bit rate calculator

3.3. Serial port parameters

- Baud rate: 600~230400bps, default 115200bps
- Data bits: 8
- Check digit: NONE, EVEN, ODD, default is None
- Stop position: 1, 2, default 1

- Mode switching: RS485/RS232 mode is optional, and cannot be used at the same time



Pic8 Serial 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, which can realize the mutual conversion between serial frame information and CAN (FD) frame information. See Chapter 4 for the specific description of the conversion mode.

Conversion direction: By selecting the conversion direction, data interference from 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 serial bus to the CAN bus and vice versa.
- Serial to CAN only: converts data from the serial bus to the CAN bus, but does not convert data from the CAN bus to the serial bus.
- Only CAN to serial: only the data of CAN bus is converted to serial bus, and the data of serial bus is not converted to 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. If not selected, the frame information of CAN (FD) is not converted.

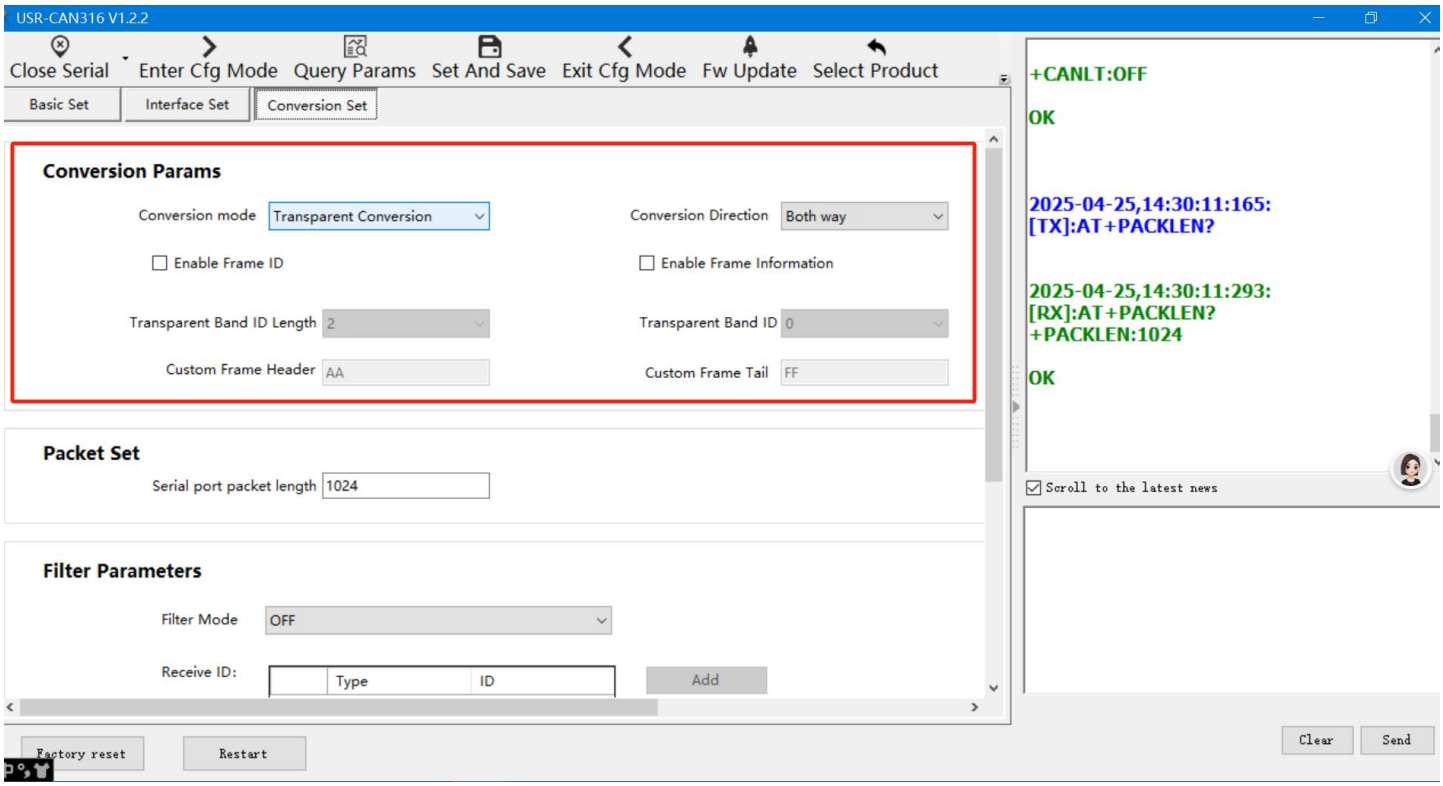
Enable Frame ID: Only effective in transparent conversion. When selected, the converter adds the CAN (FD) frame ID to the frame data of the serial frame before the frame information (such as enable frame information) when working. If not selected, the CAN (FD) frame ID is not converted.

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.



Pic9 Diagram of configuration of conversion parameters

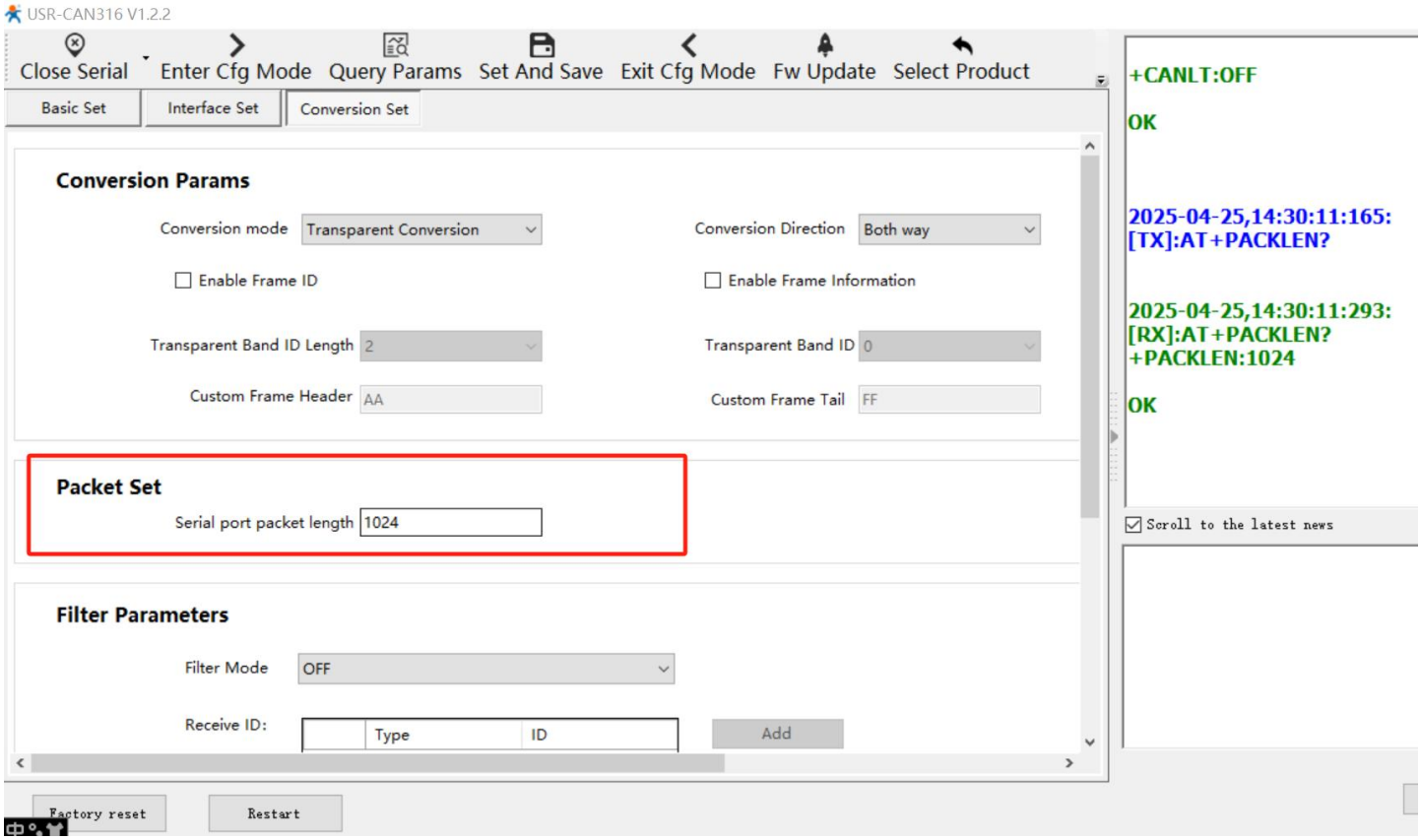
3.4.2. Subcontracting Settings

CAN316 Supports custom serial frame packet length. If a serial frame data packet contains a large amount of data and needs to be forwarded to the CAN network, the serial frame packet length can be customized to send the serial frame data in packets. The range can be set from 512 to 1024 bytes.

Note: Due to the standard protocol conversion rules, CAN frames must be 13 bytes, and CANFD frames must be 69 bytes. Therefore, in CAN mode, when the number of bytes in a serial frame exceeds 1014 bytes (the largest multiple of 13 within the range), the serial packet length must be set to a multiple of 13; otherwise, the excess part cannot be correctly transmitted. If the number of bytes in a serial frame does not exceed 1014 bytes, setting the serial frame packet length greater than 1014 bytes is sufficient.

Similarly, in CANFD mode, when the number of bytes in a serial frame exceeds 966 (the largest multiple of 69 within the range), the serial frame packet length must be set to a multiple of 69; otherwise, the excess part cannot be transmitted

correctly. If the number of bytes in a serial frame does not exceed 966, setting the serial frame packet length to be greater than 966 is sufficient.



Pic10 Subcontract configuration diagram

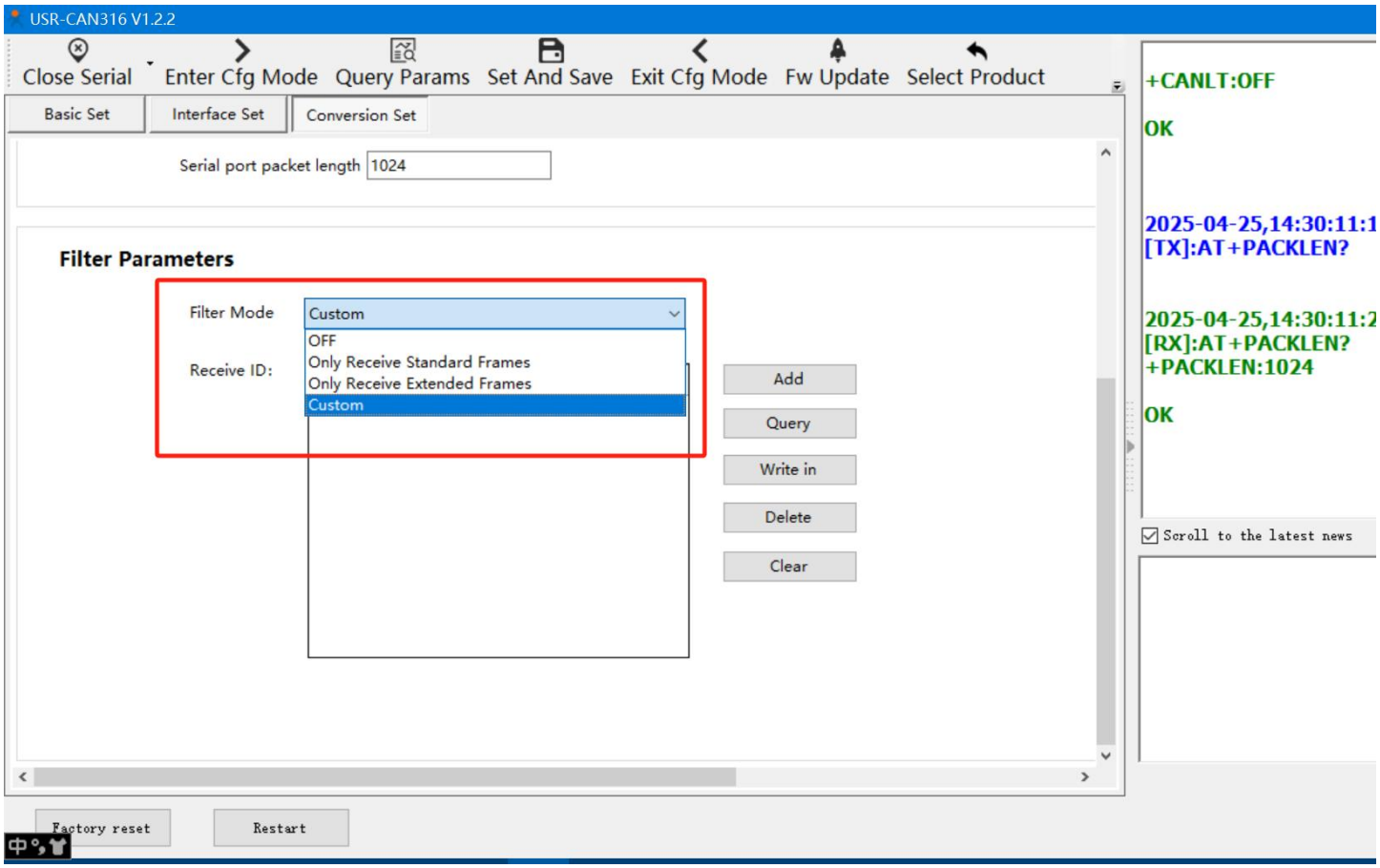
3.4.3. Filter function

CAN316 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. The configuration method is as follows:



Pic11 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.

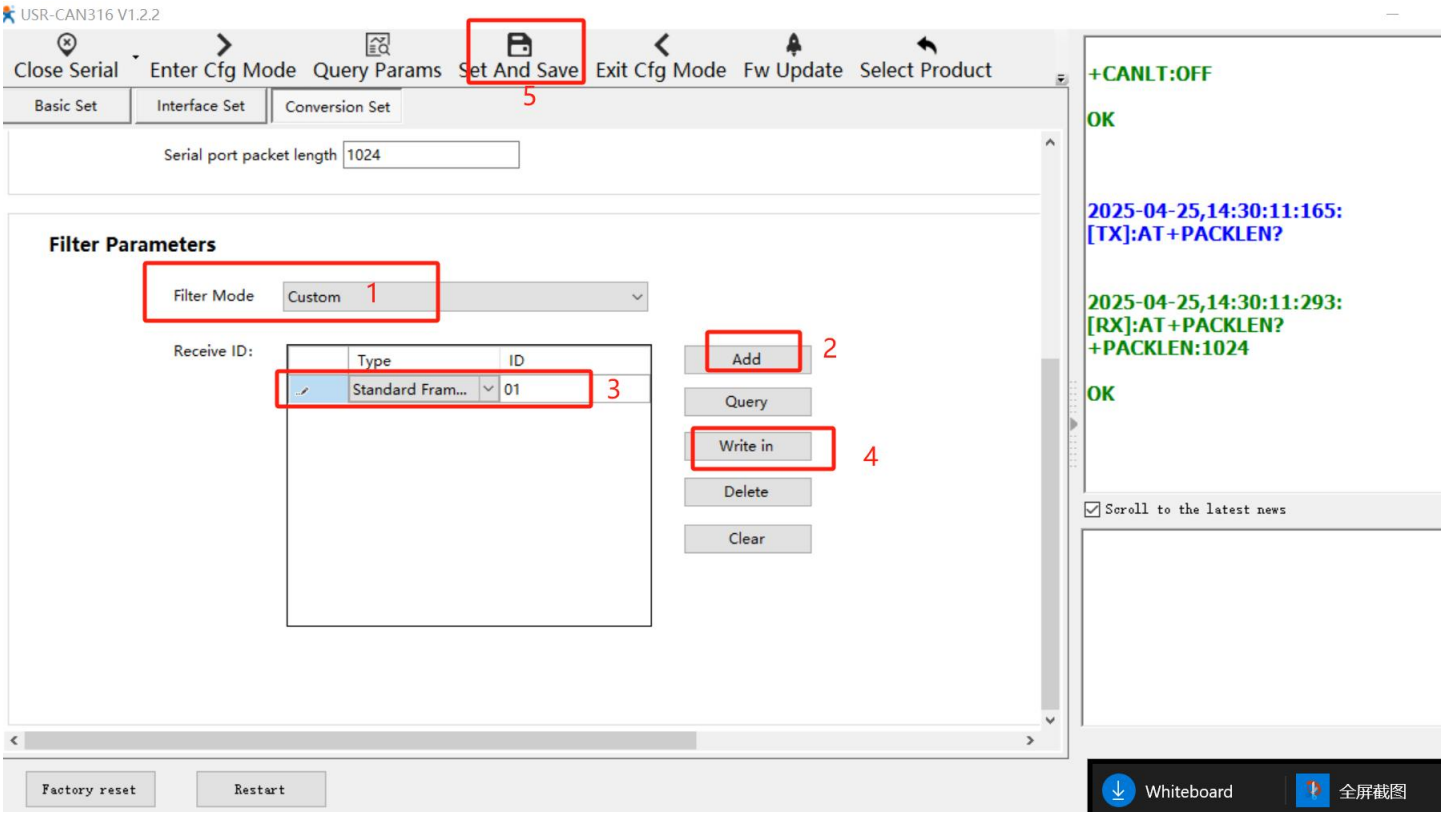
The configuration is as follows:

- The upper computer enters the parameter configuration state, and the filter mode is selected as custom
- Click Add Information and enter the ID you want to receive. 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



Pic12 Customize frame ID configuration

3.5. Modbus gateway function

This device adopts a simple way to realize the interconnection between CAN devices and Modbus devices. It can not only realize the data transmission of Modbus devices to the CAN bus for unified management, but also realize the control of Modbus devices through the CAN bus.

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).

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

3.5.1. Enable Modbus configuration

（1）Enter the parameter configuration state: serial port configuration--> select the corresponding serial port, open the serial port--> enter the configuration state--> read the parameters.

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

（3）Enable the Modbus Gateway and select its operating mode. When the device acts as a Modbus RTU host, it can support up to 255 Modbus RTU slave devices. When the device acts as a Modbus RTU slave, it has 2400 preset registers, which can be written to by external master devices at any time, ranging from 0 to 2399. The device supports reading with function code 03, writing to a single register with function code 06, and writing to multiple registers with function code 10.

（4）Send and receive packets on demand

3.5.2. 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.

This device provides a quick upper-level software that can easily configure and send messages. You can either directly configure in the upper-level machine or import/export point tables through.csv files. It is suitable for scenarios with numerous collection points and mapped points. The configuration method of the upper-level machine is as follows:

（1） Enter the parameter configuration state: open the serial port--> enter the configuration state--> read the parameters--> select Modbus working mode.

（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	Periodic send;	Periodic send--1

	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	Changes sent; Send once; Frame ID triggered;	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 type	It takes effect in frame ID trigger mode and refers to the frame type of the CAN frame that triggers the transmission of this message;	Standard frame Expand the frame	Standard frame--A 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

3.5.3. 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.

This device provides a user-friendly upper-level software that allows for easy configuration of receiving messages. You can either configure directly within the upper-level machine or import and export point tables via.csv files. It is ideal for scenarios with a large number of collection points and mapped points. The configuration method for the upper-level machine is as follows:

（1）Enter the parameter configuration state: open the serial port--> enter the configuration state--> read the parameters--> select Modbus working mode.

（2）Click Add message to configure the required content of the received message. Select the message and click Add 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 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;	1~255	1~255

	When the device is a slave, fill in the address of the device as a slave;		
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

3.6. 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, please refer to CANFD series product Firmware Upgrade Manual.

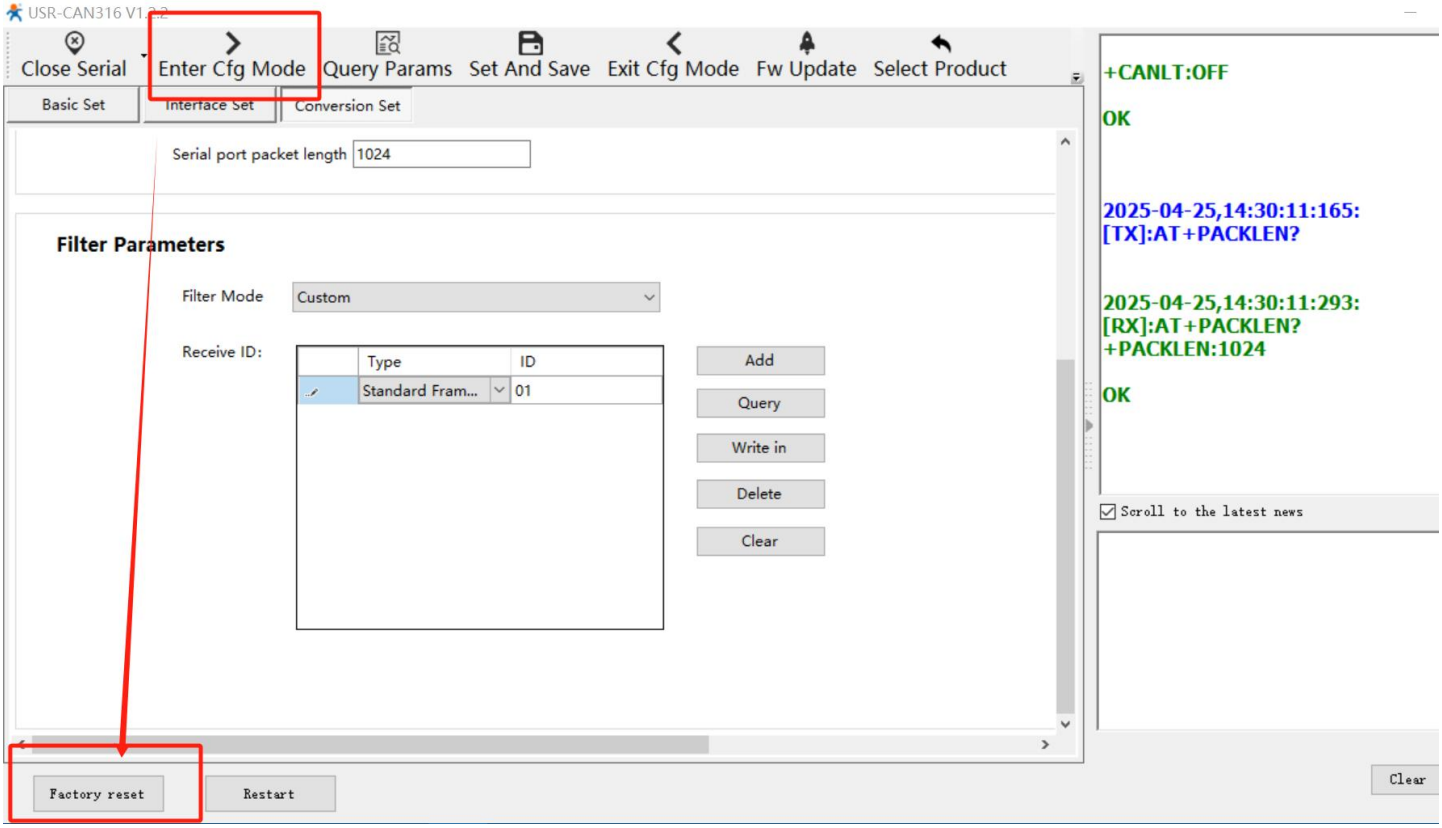
3.7. Factory data reset

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+RELD, press Enter, and receive the correct reply +OK, then the factory Settings are restored.

Set software Settings:



Pic13 Set up the software diagram

4. Conversion mode example

4.1. Transparent conversion

Under the transparent conversion method, CAN316 immediately converts and sends data received from one bus to the other side without adding any data or making any modifications to the data. 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 CAN (FD) message frame information (frame type part) and frame ID 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, and the data can be converted in real time, which can support the transmission of data with large traffic.

4.1.1. Serial frame to CAN (FD) —— transparent conversion

Serial 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.

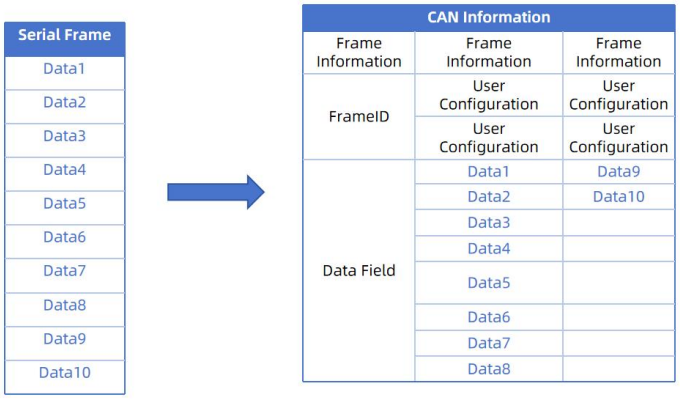
CAN mode: maximum 8 bytes per frame.

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 of 12,16,20,24,32,48, and 64, otherwise the converter will automatically split it into several CANFD messages of appropriate length.

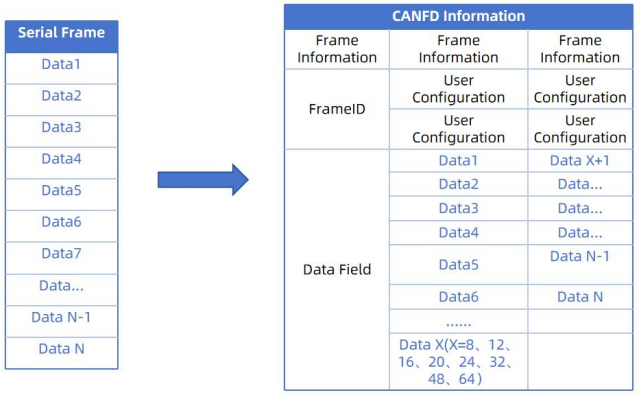
For example: the data length is 58, and the DLC of CANFD cannot encode a length of 58; the closest it can be is 48. Therefore, a CANFD message with a data length of 48 is converted. The remaining data length is 10, and the closest DLC encoding length is 8. Thus, a CANFD message with a data length of 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 is split into 3 CANFD messages.

Serial frame to CAN



Serial frame to CANFD



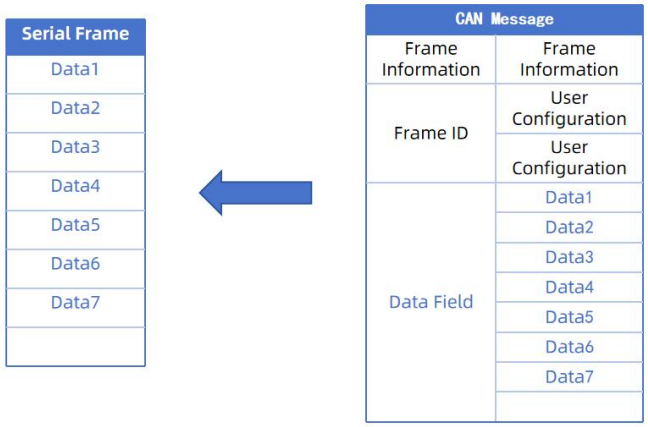
4.1.2. CAN (FD) to serial frame —— transparent conversion

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

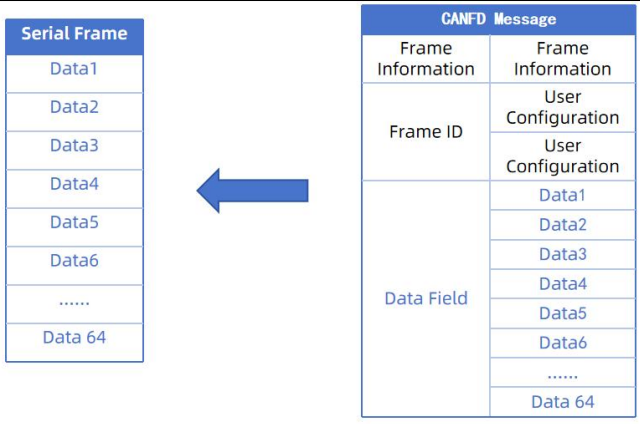
If the enable frame information is enabled, the converter adds the frame information of the CAN (FD) message to the first byte of the serial frame when it is working. The frame information of CAN (FD) is not converted if it is not selected.

If the enable frame ID is enabled, the converter will add the CAN (FD) message frame ID before the serial frame frame data and after the frame information (such as enable frame information) when it works.

CAN serial frame (no enabled frame information/ enabled frame ID)



CANFD to serial frame (enable frame information / enable frame ID is enabled)



4.2. Transparent band ID conversion

The transparent label conversion is a special use of transparent conversion, without an additional protocol. This method can convert the "address" in a serial frame to the identifier field of a CAN (FD) message, where the start position and length of the serial frame ID can be configured. The converter extracts this frame ID during the conversion and fills it into the frame ID field of the CAN (FD) message, serving as the ID for the forwarded CAN (FD) message. Similarly, when converting a CAN (FD) message to a serial frame, the ID of the CAN (FD) message is also converted and placed in the corresponding position of the serial frame.

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 (frame ID) sent at this time is filled with data from the serial frame mentioned above.

4.2.1. Serial frame to CAN (FD) —— Transparent band ID conversion

Set the starting address and length of the "frame ID" of the CAN (FD) message in the CAN (FD) frame type and serial frame. 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 settings, the "frame ID" of the CAN (FD) message in the serial frame is fully converted into the frame ID field of the CAN (FD) message. 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 is padded with 0.

Serial frame to CAN

Serial Frame Address	Serial Frame 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 Message		
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	

Serial frame to CANFD

Serial Frame Address	Serial Frame 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 Message		
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) to serial frame —— transparent band ID conversion

For CAN (FD) messages, a frame is forwarded immediately upon receipt. Each time the frame is forwarded, the ID of the received CAN (FD) message is converted according to the position and length of the CAN (FD) frame ID in the serial frame. 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.

The CAN transmits serial frames

Serial Frame 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 converts serial frames

Serial Frame Data
Data1
FrameID3
FrameID4
Data2
Data3
Data4
Data5
Data6
.....
Data64

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

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, 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 has a fixed length of 13 bytes, and CANFD frame has a fixed length of 69 bytes. If it is less than this, 0 must be added, 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 8 bytes, CANFD mode length 64 bytes, less must be supplemented with 00.

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							

Pic14 Segmented protocol in CAN mode

In CANFD mode, the content of the transmitted Modbus protocol can start from "data 2" 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 completed.

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							

Pic15 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 or not. If this bit is 0, it indicates that the message is a separate message; if it is 1, it indicates that it belongs to a frame in the segmented message.
- Segment type: 2 bits (Bit6, Bit5) are used to indicate the type of this 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 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.

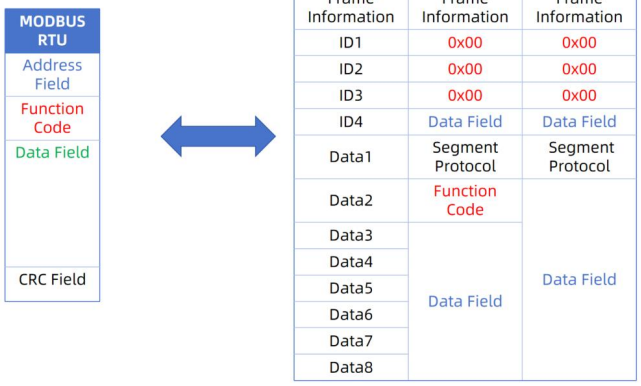
The conversion involves the protocol content of Modbus, the function code, and the data field. During conversion, they are sequentially transferred into the data field of a CAN message frame (starting from the second data byte, with the first data byte used for segmenting the protocol). Since the length of Modbus frames varies depending on the function code, and a single CAN message frame can only transmit 7 data bytes, while a single CANFD message frame can only transmit 63 data bytes, the converter segments longer Modbus frames into CAN messages and sends them 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:

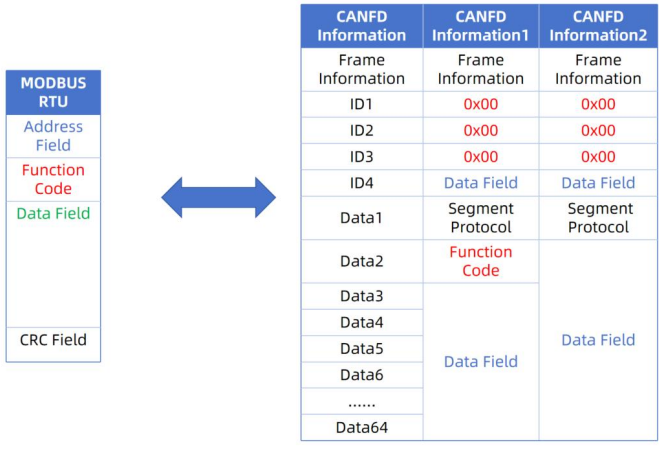
For CAN, the bus Modbus protocol data does not need to do cyclic redundancy check (CRC16). The converter receives according to the segmented protocol. After receiving a frame and parsing, it automatically adds cyclic redundancy check (CRC16) and converts it into a Modbus frame to send to the serial bus.

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. Serial frame to CAN (FD) —— Custom frame header and tail conversion

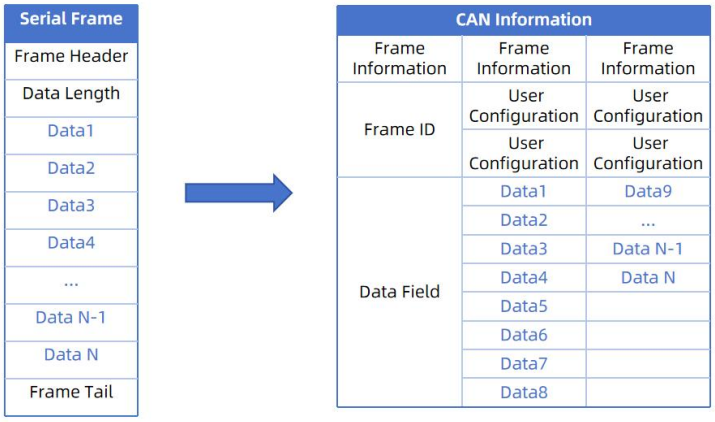
The serial frame format must conform to the specified frame format, otherwise it cannot be transmitted correctly. The serial frame must contain: frame header, data length, data domain, and frame tail.

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

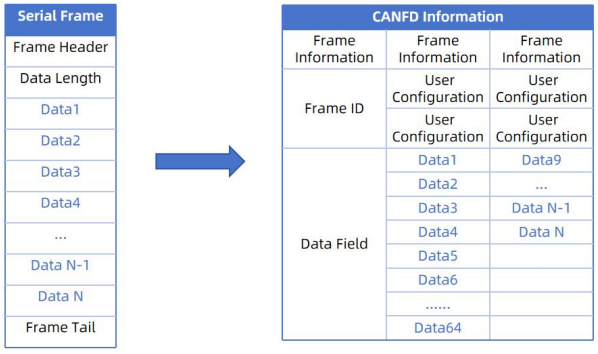
Data length refers to the byte length of the data field. The data length must match the frame end data for correct transmission; otherwise, it will be discarded. For example: if the frame header is configured as AA and the frame end as FF, a serial frame AA 03 01 02 04 FF can be transmitted normally. 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 cannot 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 to CAN frames.

Serial frame to CAN



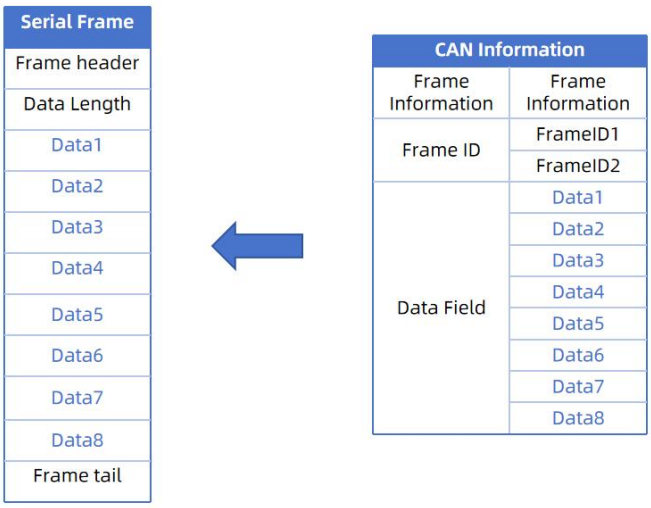
Serial frame to CANFD



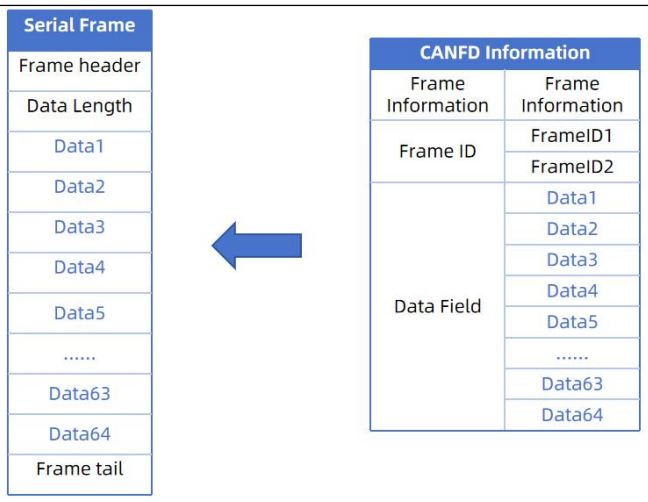
4.5.2. Serial frame to CAN (FD) —— Custom frame header and tail conversion

The CAN (FD) bus message forwards a frame as soon as it receives a frame. The module will convert the data in the data field of the CAN message 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.

The CAN transmits serial frames



CANFD converts serial frames



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-CAN316 series of products. This document does not grant any intellectual property rights and does not explicitly or implicitly grant any intellectual property rights, nor does it prohibit any form of speech or other means to grant such rights. Except for the liability stated in the terms and conditions of sale for 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 but not limited to the 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