CAN FD TO RS232/RS485

From Waveshare Wiki
Jump to: navigation, search

Overview

Introduction

The CAN FD TO RS232/RS485 is a DIN-rail CAN FD to RS232/RS485 bidirectional converter, featuring high-speed, low-latency, stable performance, ease of use, and cost-effectiveness. It supports five data conversion modes: transparent conversion, transparent conversion with ID, standard protocol conversion, Modbus protocol conversion, and custom protocol conversion. It enables interconnection between CAN and serial devices, compatible with CAN FD and CAN 2.0A/2.0B protocols.
It features industrial-grade design, supports -40°C ~ 85°C stable working, 9V ~ 36V wide power input, 5Kbps~1Mbps CAN FD arbitration domain

CAN FD TO RS232/RS485

(https://www.waveshare.com/product/can-fd-to-rs232-rs485.htm)

RS232/RS485, CAN FD

baud rate, and 100Kbps~5Mbps CAN FD data domain baud rate.

It supports parameter configuration via host computer, built-in 120Ω terminal resistors (switchable via DIP switches), supports DIN-rail mounting, easy to configure and install.

Specifications

Classification	Parameter	Value
Basic parameters	Working voltage	DC 9V ~ 36V
	Size	110 × 27 × 85.6 (mm)
	Installation	DIN-rail installation
	Reload button	Long press to restore factory settings
	Indicator light	POWER, WORK, COM and CAN
Interface parameters	CAN port specification	1-ch CAN port, supports CAN FD, compatible with CAN 2.0A/2.0B
	CAN port baud rate	Arbitration domain baud rate: 5Kbps ~ 1Mbps Data domain baud rate: 100Kbps ~ 5Mbps
	Terminal resistor	Built-in 2 × 120Ω terminal resistors Switching either DIP switch to ON connects one 120 Ω resistor in parallel Switching both to ON connects two 120Ω resistors in parallel
	Serial port specification	Supports 1-ch RS485 or RS232
Working 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	Modbus gateway	CAN FD to Modbus RTU (Master/Slave)
	Send message	Supports up to 64 transmit messages
	Receive message	Supports up to 64 receiving messages
	Protocol conversion	Transparent conversion, transparent conversion with ID, standard protocol conversion, Modbus protocol conversion, and custom protocol conversion
	CAN ID	Supports standard frame and extended frame
	Frame ID filtering	Supports only standard frames, only extended frames, and custom input frame ID (up to 32 groups)
	Change direction	Supports bidirectional conversion, only serial port to CAN, and only CAN to serial port
	Firmware upgrade	Supports the host computer for firmware upgrade
	Parameter configuration	AT command and host computer software configuration
Protection parameters	ESD protection	Air discharge: 8 KV, Contact discharge: 6 KV
	Eft/Burst	Power circuit: 2KV; RS485/RS232 circuit: 1KV; CAN circuit: 1KV
	Surge immunity test	Power circuit: 1KV differential mode, 2KV common mode; CAN circuit: 2KV common mode

Hardware Description

Indicator Light Description

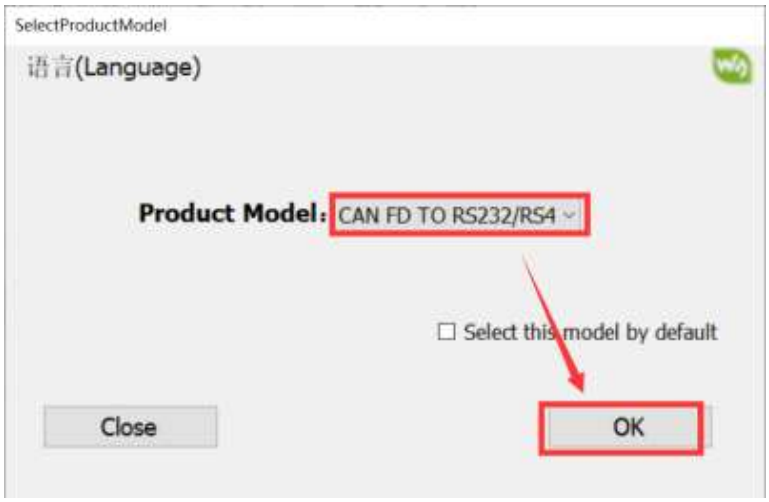
Indicator	Color	Function description
POWER light	Red	Always on when powered on, turns off when powered off
STA light	Green	Flickering: Device operating normally, frequency 1s; Strobe: Enters the CAN bus passive error state; Always on: CAN bus running abnormally
COM light	Green/Red	Green flickering: Indicates that the serial port is receiving data Red flickering: Indicates that data is being sent from the serial port
CAN light	Red	Green flickering: Indicates that the CAN port is receiving data Red flickering: Indicates that data is being sent from the CAN port

Software Operation

- Note: Take the transmission mode as an example

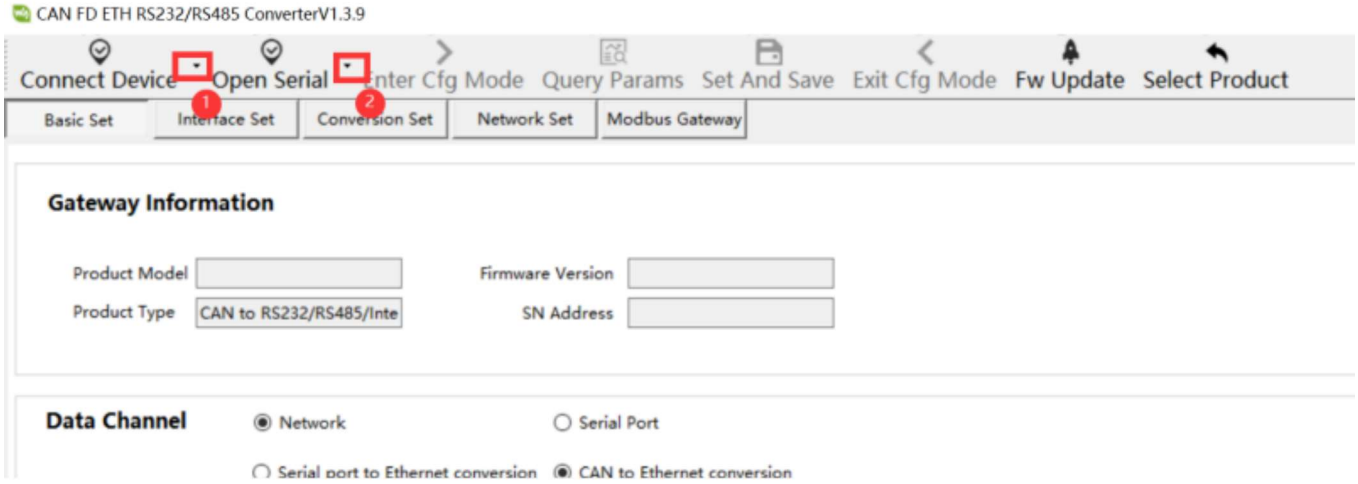
CAN FD Converter Basic Parameter Configuration

- Open WS-CAN-FD-Converter and select CAN FD TO RS232/RS485. (The software defaults to English and can be modified to Chinese)



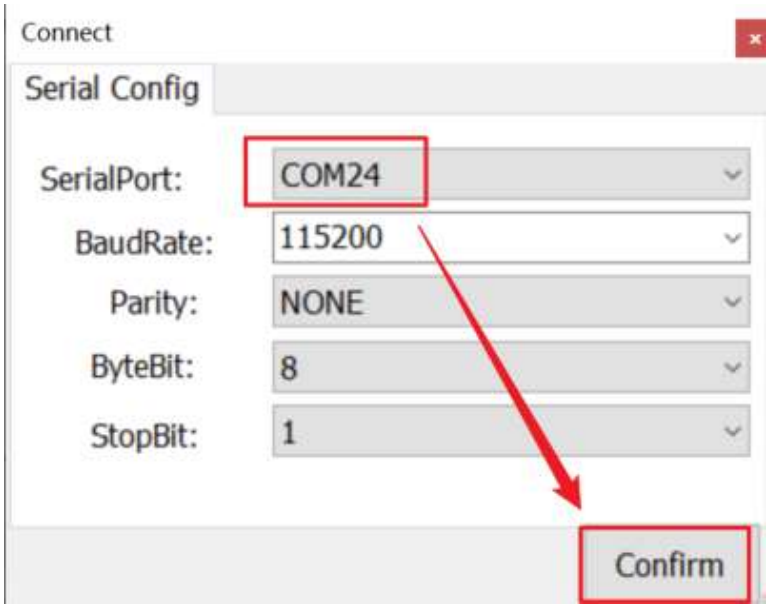
(/wiki/File:CAN_FD_TO_RS232_RS485_Converter_RS232_communication_test_wiring_wiki.png)

- Click the triangle next to Open Serial to select the serial port configuration, then click the small triangle next to the Open Serial to modify the serial port configuration



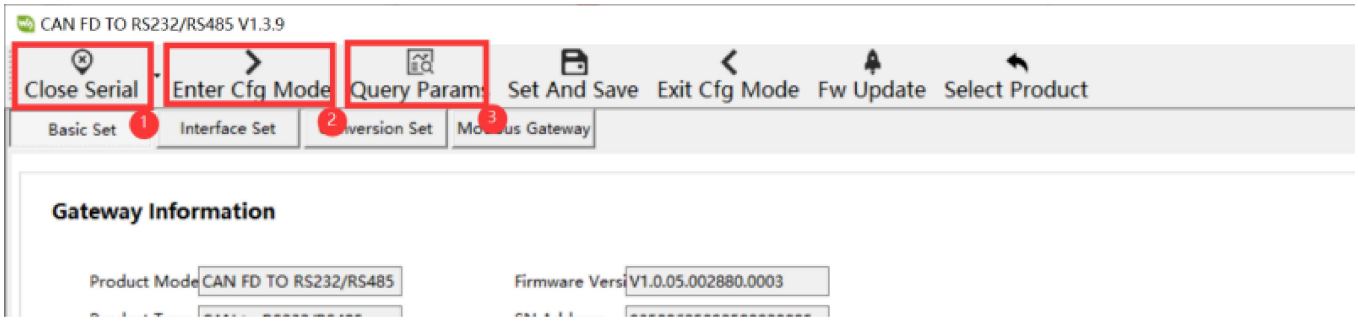
(/wiki/File:CAN_FD_ETH_RS232_RS485_Converter_Serial_Port_Config_wiki_2_1.png)

- Modify the serial port number to the actual serial port number of USB TO RS232/485



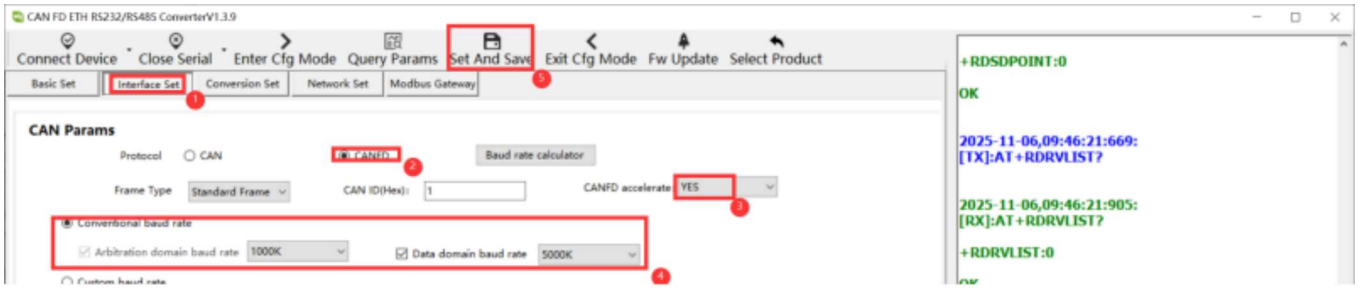
(/wiki/File:CAN_FD_ETH_RS232_RS485_Converter_Serial_Port_Config_wiki_3.png)

- After configuring the serial port parameters, click on "Open Serial", "Enter Cfg Mode", and "Query Params" in sequence. (If there is an error prompt in the message bar on the right, click "Query Params" again until no error occurs before proceeding to the next step)



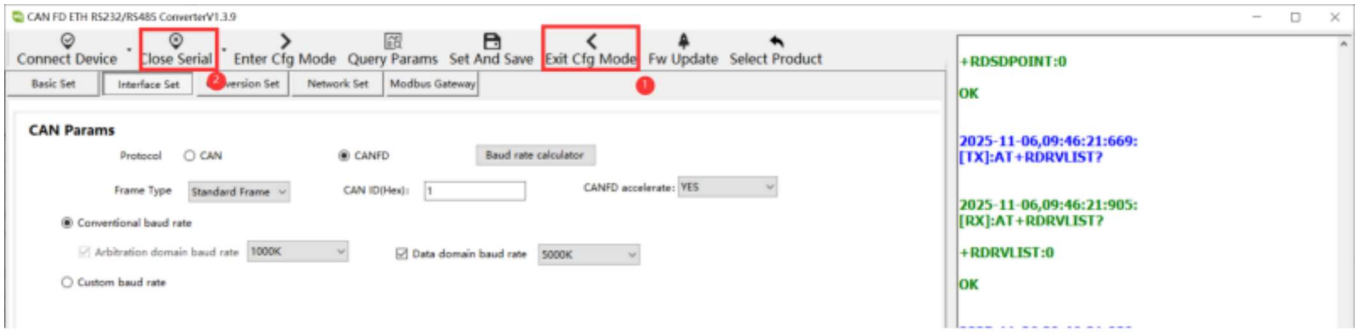
(/wiki/File:CAN_FD_ETH_RS232_RS485_Converter_Serial_Port_Config_wiki_4.png)

- Click to set CANFD mode, enable CANFD acceleration, Arbitration domain baud rate as 1000k, Data domain baud rate as 5000k, and then click Set and Save



(/wiki/File:CAN_FD_ETH_RS232_RS485_Converter_Serial_Port_Config_wiki_5.png)

- After confirming that there is no error prompt on the right command, click Exit Cfg Mode and Close Serial to end the device configuration. The command on the right has an error and needs to be saved or configured again



(/wiki/File:CAN_FD_ETH_RS232_RS485_Converter_Serial_Port_Config_wiki_6.png)

Example Demonstration

Connection Description

CAN

1. The CAN-H of CAN FD TO RS232/RS485 is connected to the CAN1-H of USB-CAN-FD
2. The CAN-L of CAN FD TO RS232/RS485 is connected to the CAN1-L of USB-CAN-FD

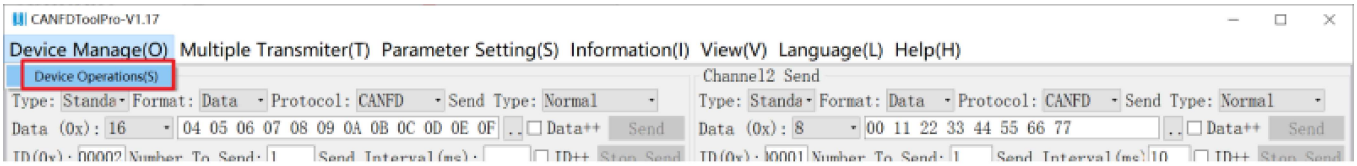
RS232

1. The RS232-GND of CAN FD TO RS232/RS485 is connected to the RS232-GND of USB TO RS232 (B)
2. The RS232-RX of CAN FD TO RS232/RS485 is connected to the RS232-TX of USB TO RS232 (B)
3. The RS232-TX of CAN FD TO RS232/RS485 is connected to the RS232-RX of USB TO RS232 (B)

(/wiki/File:800px-CAN_FD_TO_RS232_RS485_Converter_RS232_communication_test_wiring_wiki.png)

USB_CANFD TOOL Communication Software Configuration

- Open the test software "USB_CANFD TOOL.exe" and click Device Operations

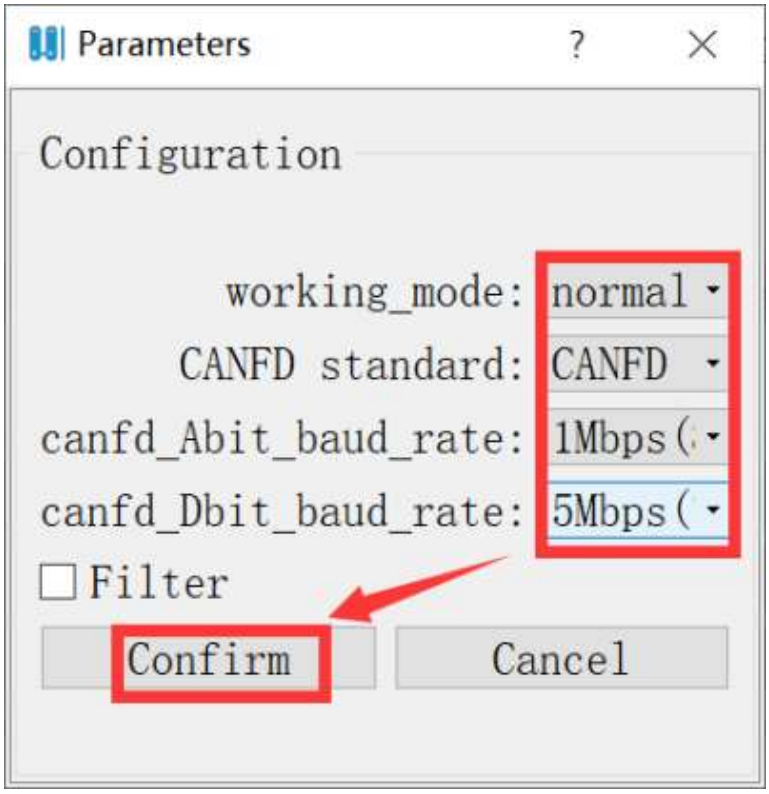


(/wiki/File:CAN_FD_ETH_RS232_RS485_Converter_USB_CANFD_TOOL_Communication_Software_Configuration_wiki_1.png)

- Click "Open Device" next to the Index, then click "Start" next to USBCANFD- 200U

(/wiki/File:CAN_FD_ETH_RS232_RS485_Converter_USB_CANFD_TOOL_Communication_Software_Configuration_wiki_2.png)

- Select "Normal" for working mode, "CANFD ISO" for CANFD standard, "1Mbps" for arbitration domain baud rate, "5Mbps" for data domain baud rate, and finally click "Confirm"



(/wiki/File:CAN_FD_ETH_RS232_RS485_Converter_USB_CANFD_TOOL_Communication_Software_Configuration_wiki_3.png)

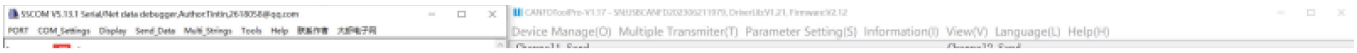
SSCOM Software Settings

- Open the test software "SSCOM.exe", select the COM port (COM5) connected to the computer via serial port (select according to specific circumstances), choose CAN FD TO RS232/RS485 serial port with default baud rate 115200, and click to open the serial port.

(/wiki/File:CAN_FD_ETH_RS232_RS485_Converter_SSCOM_Software_Configuration_Serial_Port_wiki_1.png)

Interface Transmission and Reception Communication Test

- USB_CANFD TOOL: Click to send message, send data 55 55 66 66 77 77 88 88. SSCOM software: Check HEX hexadecimal to receive data 55 55 66 66 77 77 88 88.
- SSCOM software: Click to send message, send data 11 11 22 22 33 33 44 44. USB_CANFD TOOL: Check HEX hexadecimal to receive data 11 11 22 22 33 33 44 44.



(/wiki/File:CAN_FD_ETH_RS232_RS485_Converter_Network_port_communication_test_wiki_1.png)

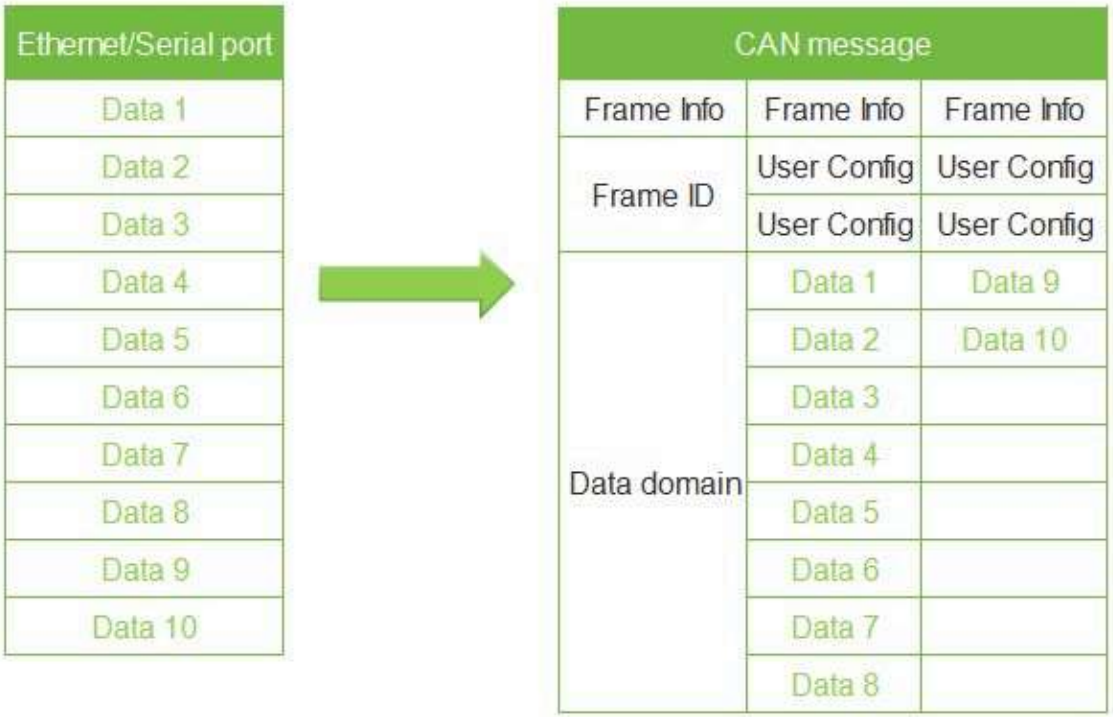
Instructions for Conversion Modes

Transparent Conversion

- In the transparent conversion mode, the data received from one bus is immediately converted and sent to the other bus, without attaching data or making any modifications to the data. This not only realizes the exchange of data formats without changing the data content, and the converter is transparent to the buses at both ends.
- The CAN message frame information (frame type part) and frame ID are pre-configured by the user and remain unchanged during the conversion process. The user can choose whether to convert the frame information and frame ID.
- This method does not increase the communication burden on users, but can convert data in real-time and handle the transmission of large amounts of data.

Serial Port to CAN(FD)

- Serial Port to CAN message: All data from serial frame is sequentially filled into the data field of CAN message frame. As soon as the converter detects data on the serial bus, it immediately receives and converts it. The frame information (frame type section) and frame ID need to be pre-configured.
  - CAN mode: Maximum 8 bytes per frame of data.
  - CANFD mode: Maximum 64 bytes per frame of data. Note: When the data length exceeds 8 bytes, it must be encoded according to the length of CANFD's DLC to ensure accurate conversion, i.e., the length is 12, 16, 20, 24, 32, 48, 64, otherwise the converter will automatically split the CANFD messages into several CANFD messages of appropriate length.
  - Serial Port to CAN



(/wiki/File:CAN_FD_ConverterTransparent_transmission_ETH2CAN.jpg)

- Serial Port to CAN FD

(/wiki/File:CAN_FD_ConverterTransparent_transmission_ETH2CANFD1.jpg)

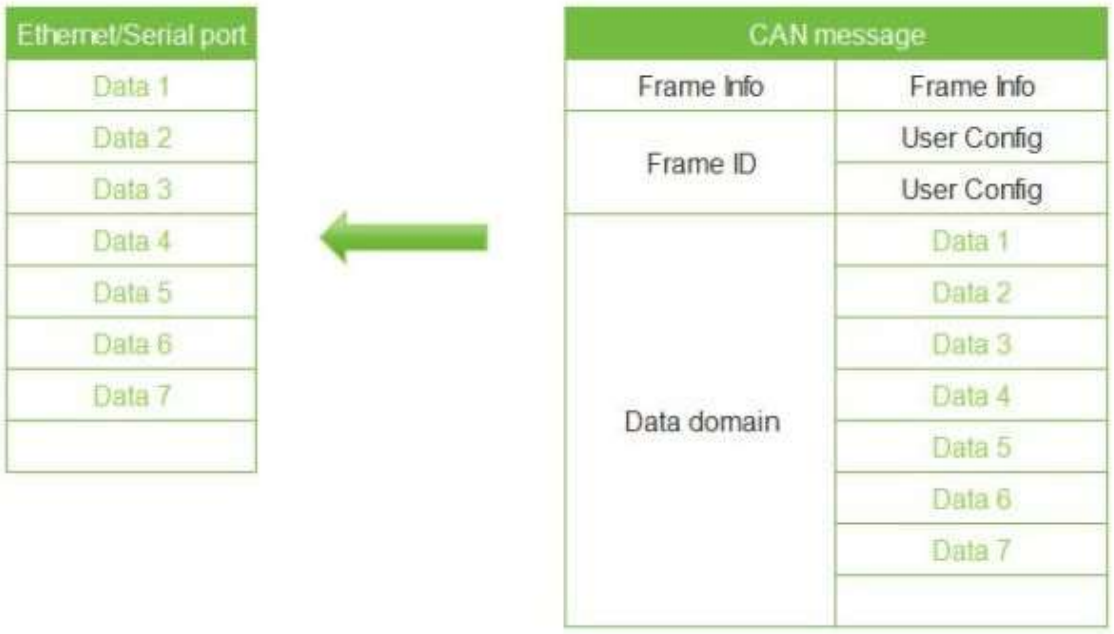
CAN(FD) to Serial Port

- Similarly, for CAN bus messages, one received CAN(FD) message will be immediately forwarded.
- If the configuration enables frame information, the converter will add the frame information of the CAN (FD) message to the first byte of the serial frame during operation. Frame information for CAN(FD) is not converted when unchecked.

- If the configuration enables the frame ID, the converter will add the frame ID of the CAN (FD) message before the frame data of the serial frame and after the frame information (such as enabling the frame information) during operation

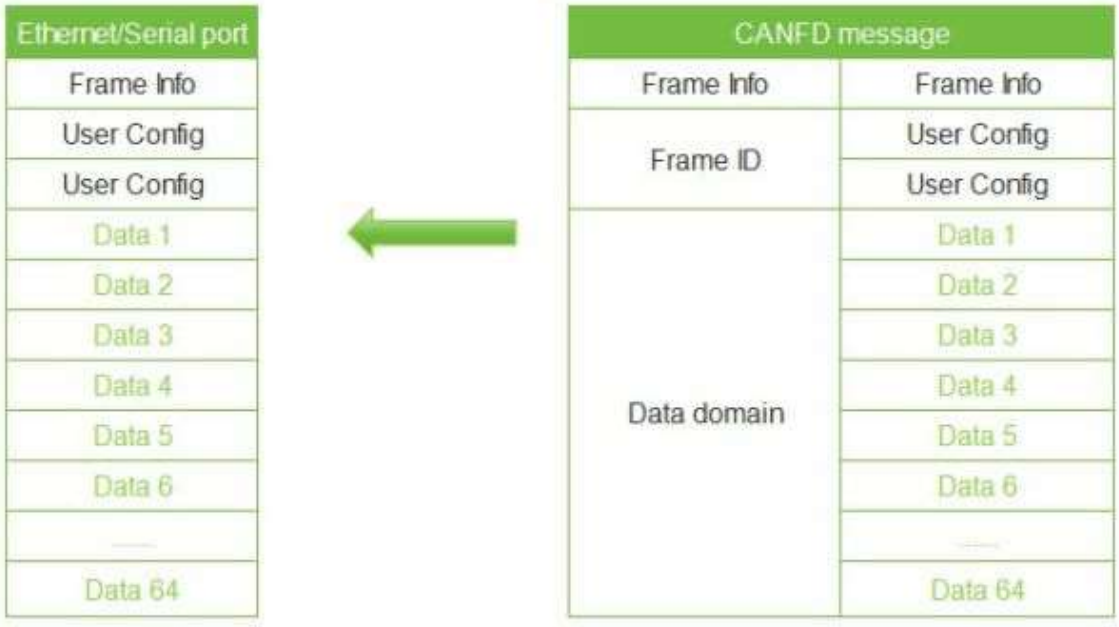
CAN to Serial Port data (enable frame information/enable frame ID not enabled) CANFD to Serial Port data (enable frame information/enable frame ID enabled)

- - CAN to Serial Port (enable frame information/enable frame ID not enabled)



(/wiki/File:CAN_FD_ConverterTransparent_transmission_CAN2ETH.jpg)

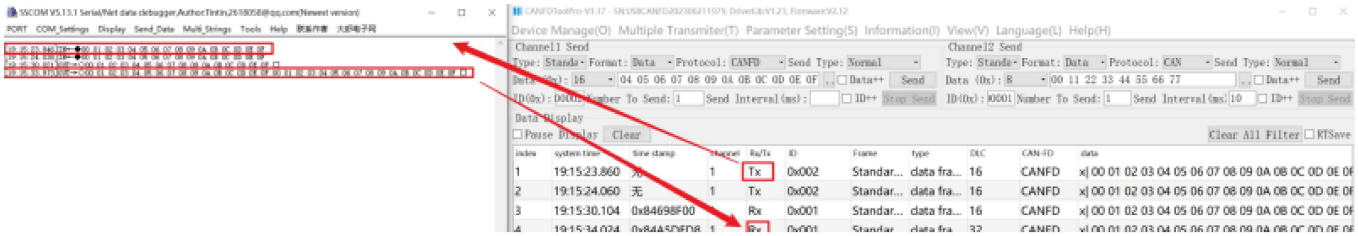
- - CAN FD to Serial Port (enable frame information/enable frame ID enabled)



(/wiki/File:CAN_FD_ConverterTransparent_transmission_CANFD2ETH.jpg)

Conversion Examples

Serial port:



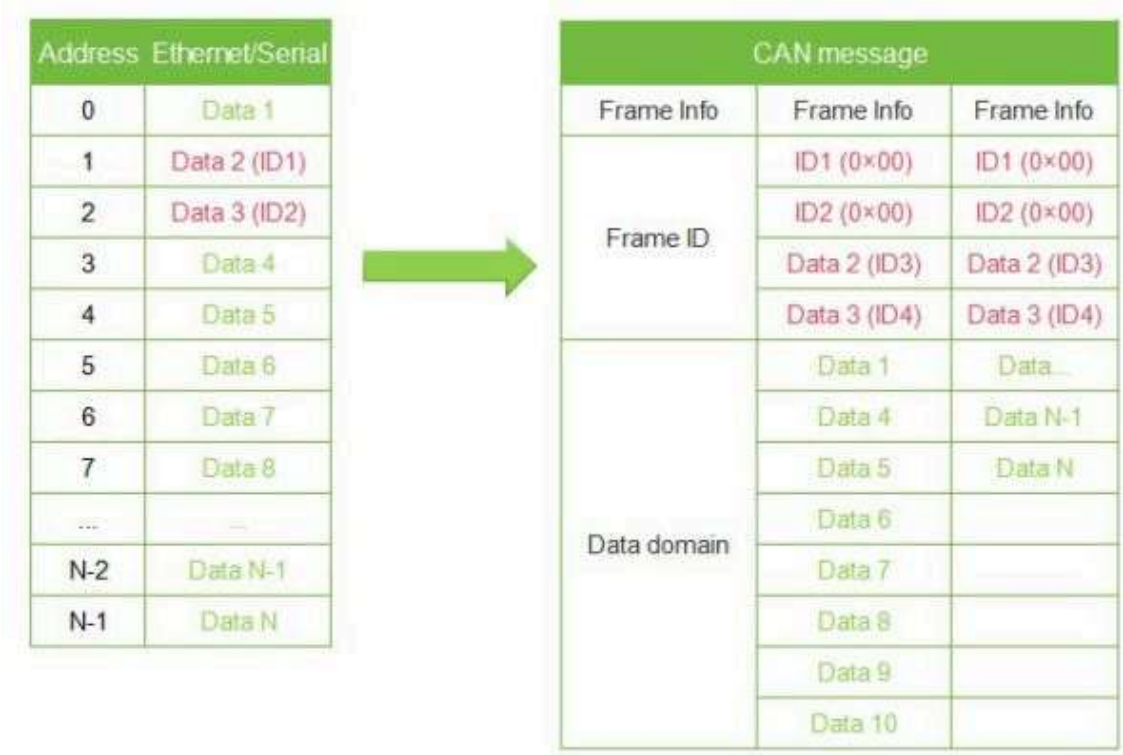
(/wiki/File:CAN_FD_Converter_Serial_port_conversion_mode_test_1.png)

Transparent Conversion with ID

- Transparent conversion with ID is a special use of transparent conversion and does not add any protocol. This method can convert the "Address" in serial port data into the ID field of CAN (FD) messages, where the starting position and length of the frame ID in serial port data can be configured. The converter will extract this frame ID during conversion and fill it in the frame ID field of the CAN (FD) message as the ID of the CAN (FD) message when forwarding serial port data. Similarly, when CAN messages are converted to serial port data, the CAN(FD) message ID is also converted to the corresponding position in the serial port data.
- In this way, the converter adapts to the user's custom protocol to the maximum.
- Note: In this conversion mode, the "CAN ID" setting option of the configuration software is invalid because the identifier (frame ID) sent at this time is filled with data from the above serial port data.

Serial Port to CAN(FD)

- Configure the starting address and length of the "Frame ID" for CAN (FD) frame types and all CAN (FD) messages carried in serial port data. The range of the starting address is 0~7; the length range is standard frame: 1~2, and the extended frame: 1~4.
- The "Frame ID" of the CAN(FD) messages in the serial port data will be converted according to the pre-configured settings and mapped to the Frame ID field of the CAN(FD). If the length of the configured transparent conversion with ID is less than the frame ID length of the frame type in the CAN(FD) message, the high byte of the frame ID in the CAN(FD) message is padded with 0.
- For CAN (FD) messages, one frame is immediately forwarded upon receipt. During each forwarding, the ID in the received CAN (FD) message is converted based on the pre-configured position and length of the CAN (FD) frame ID in the serial port data. Other data is forwarded in sequence.
- Note: If the CAN(FD) frame ID length is greater than the set transparent conversion with ID length, only the low bytes are taken and converted to the corresponding position in the serial port frame. For example: If the CAN frame ID is 01020304 and the transparent conversion with ID length is set to 2, then only 0304 is converted to the corresponding position in the serial frame.
    - Serial Port to CAN

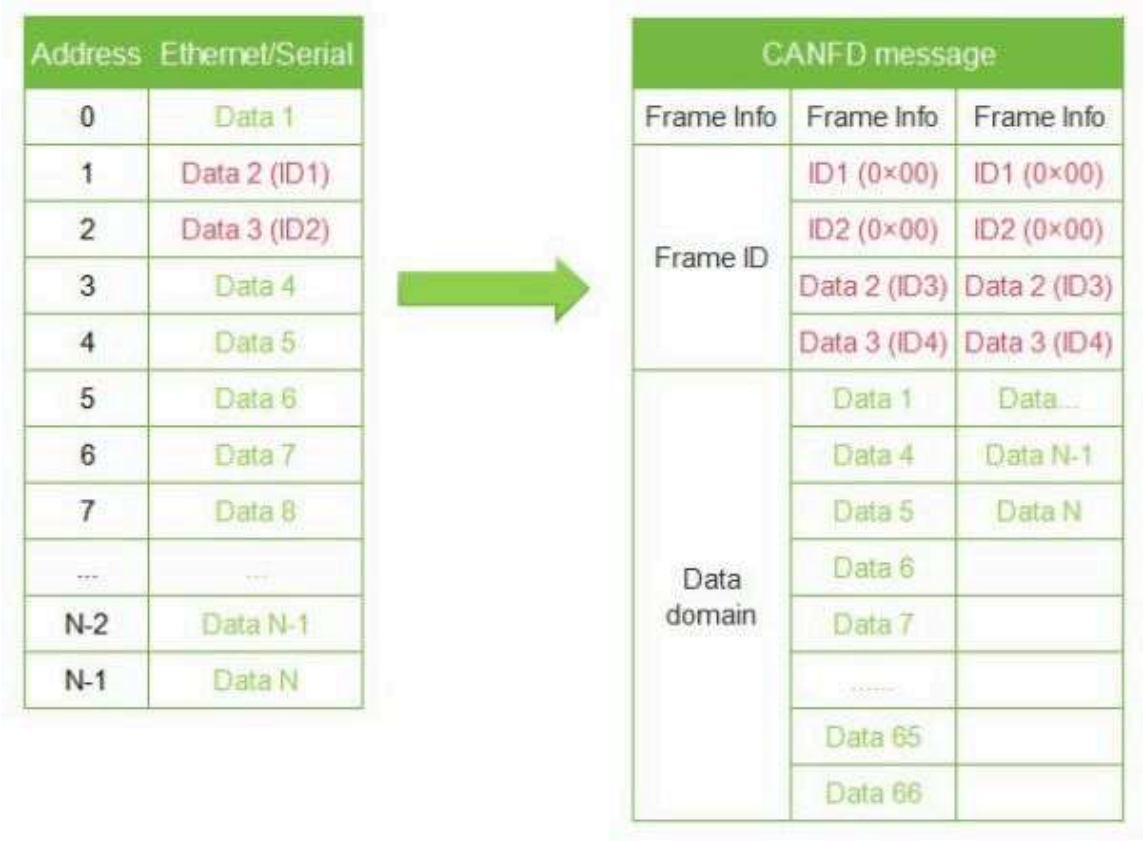


Address	Ethernet/Serial
0	Data 1
1	Data 2 (ID1)
2	Data 3 (ID2)
3	Data 4
4	Data 5
5	Data 6
6	Data 7
7	Data 8
...	...
N-2	Data N-1
N-1	Data N

CAN message		
Frame Info	Frame Info	Frame Info
Frame ID	ID1 (0×00)	ID1 (0×00)
	ID2 (0×00)	ID2 (0×00)
	Data 2 (ID3)	Data 2 (ID3)
	Data 3 (ID4)	Data 3 (ID4)
Data domain	Data 1	Data...
	Data 4	Data N-1
	Data 5	Data N
	Data 6	
	Data 7	
	Data 8	
	Data 9	
	Data 10	

(/wiki/File:600px-CAN_FD_ConverterTransparent_transmission_with_id_ETH2CAN1.jpg)

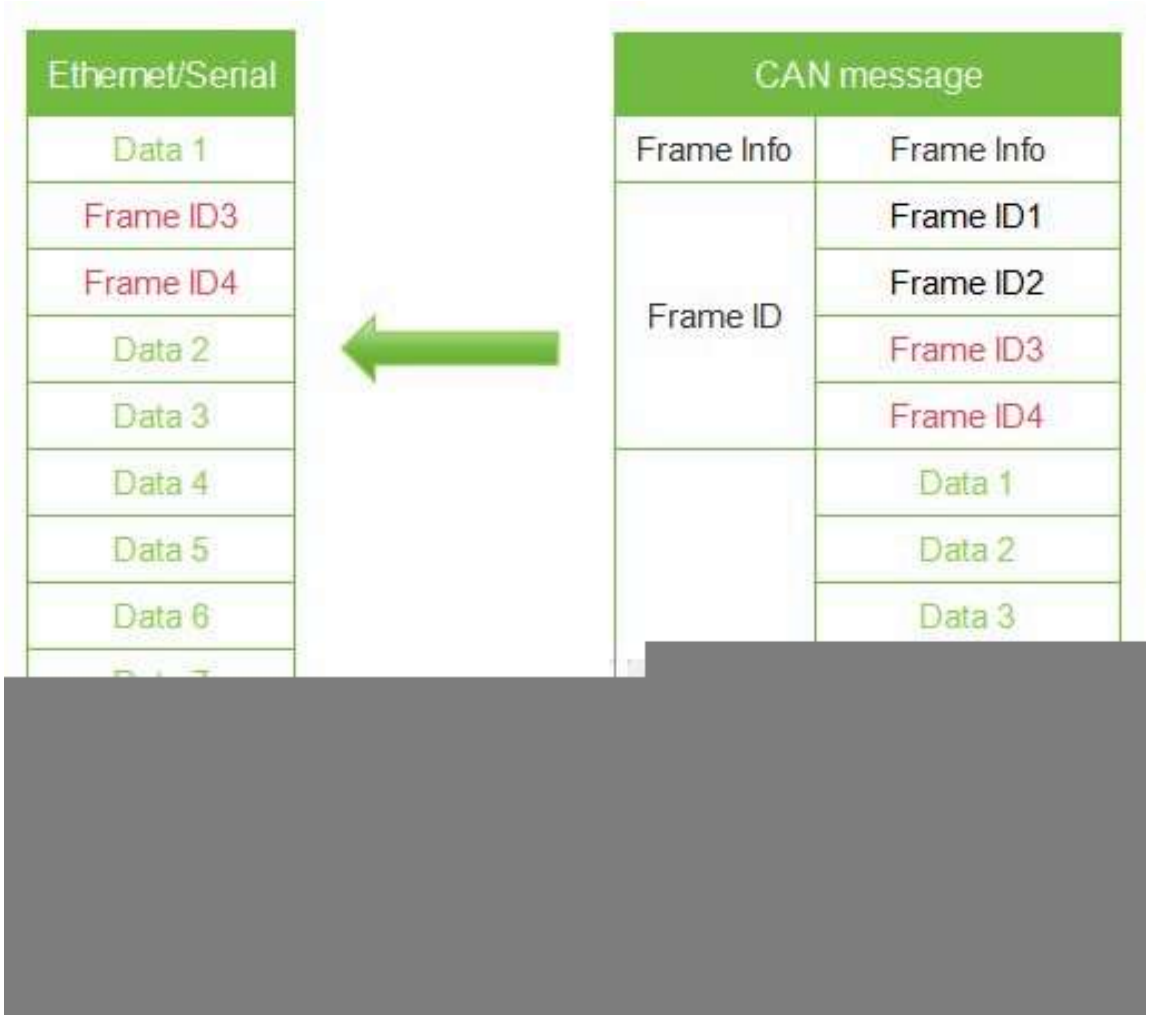
- Serial Port to CAN FD



(/wiki/File:CAN_FD_ConverterTransparent_transmission_with_id_ETH2CANFD.jpg)

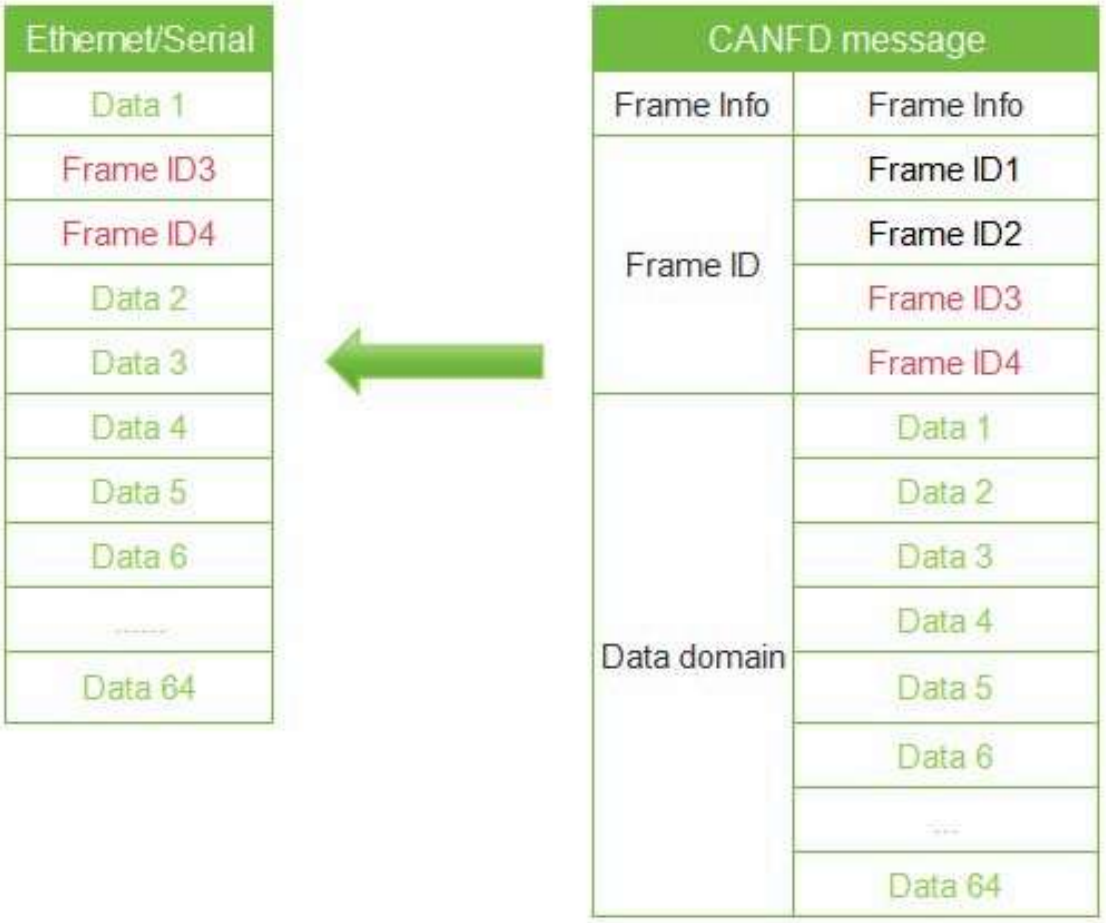
CAN(FD) to Serial Port

- For CAN (FD) messages, one frame is immediately forwarded upon receipt. During each forwarding, the ID in the received CAN (FD) message is converted based on the pre-configured position and length of the CAN (FD) frame ID in the serial port data. Other data is forwarded in sequence.
- Note: If the CAN(FD) frame ID length is greater than the set transparent conversion with ID length, only the low bytes are taken and converted to the corresponding position in the serial port frame. For example: If the CAN frame ID is 01020304 and the transparent conversion with ID length is set to 2, then only 0304 is converted to the corresponding position in the serial frame.
- - CAN to Serial Port



(/wiki/File:CAN_FD_ConverterTransparent_transmission_with_id_CAN2ETH11.jpg)

- CAN FD to Serial Port



(/wiki/File:CAN_FD_ConverterTransparent_transmission_with_id_CANFD2ETH.jpg)

Conversion Examples

Serial port:

(/wiki/File:800px-CAN_FD_Converter_Serial_port_conversion_mode_test_2.png)

Standard Protocol Conversion

- Standard CAN frame format, each CAN frame contains 13 bytes, including CAN frame information (1 byte) + frame ID (4 bytes) + data frame (8 bytes).
- Standard CANFD frame format, each CANFD frame contains 69 bytes, including CANFD frame information (1 byte) + frame ID (4 bytes) + data frame (64 bytes).
- By correctly configuring the frame information (the first byte of data), it is possible to flexibly transmit standard, extended and even remote frames. Details of standard frames,

extended frames, or even remote frames can be obtained by correctly parsing serial frames.

- Note:
- (1) In this conversion mode, the "CAN ID" and "Frame Type" configured in the software are invalid because the frame ID sent at this time is populated by the frame ID data in the serial frame above, and the frame type is determined by the frame information in the serial frame.
- (2) In this mode, the standard serial data format must be strictly followed to successfully convert. It is necessary to ensure that the frame information is correct and that the reserved bits are zero. CAN frames have a fixed length of 13 bytes, CANFD frames have a fixed length of 69 bytes. If the length is not sufficient, it must be padded with 0s, otherwise it cannot be transmitted.

Standard CAN Frame Format

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

CANFD fixed format (1 CAN frame contains 69 bytes)		
Frame info	Frame ID	Frame data
1Byte	4Byte	64Byte

(/wiki/File:600px-CAN_FD_Converter_Graphic_illustration_of_standard_protocol_conversion_1.jpg)

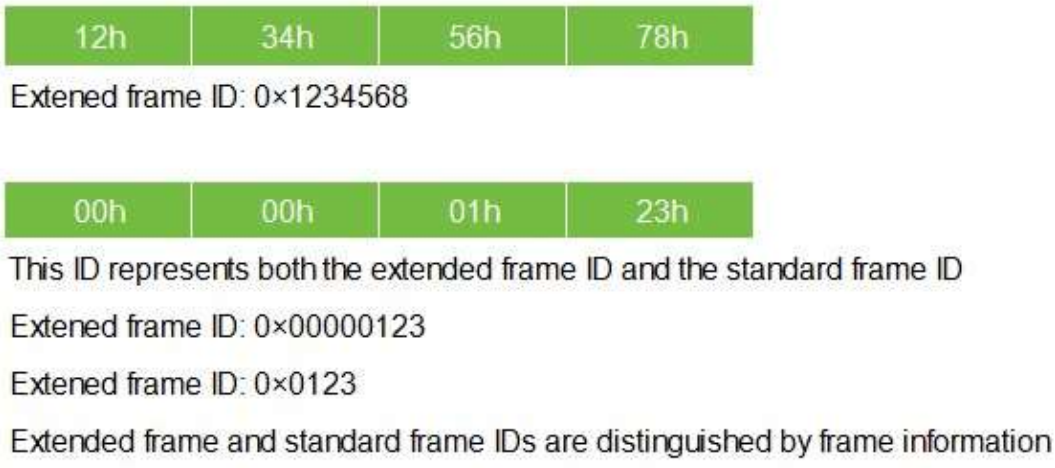
Frame information: 1 byte in length, used to identify frame information: frame type, frame length.

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FF	RTR	EDL	BR5	DLC3	DLC2	DLC1	DLC0

(/wiki/File:600px-CAN_FD_Converter_Graphic_illustration_of_standard_protocol_conversion_2.jpg)

```
FF: The identification bit for standard frame and extended frame, 1 for extended frame, 0
for standard frame.

RTR: The identification bit for remote frame and data frame, 1 for remote frame, 0 for da
ta frame, and can only be 0 in CANFD mode.

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

BRS: Baud rate switches enable identification, 0 for no rate conversion, 1 for variable r
ate conversion, valid only when CANFD. This bit should be 0 when CAN.

DLC3~DLC0: Data length bit, identifying the data length of the CAN(FD) frame.
```

Frame ID: 4 bytes long; high byte first, low byte last. The standard frame has 11 valid bits, and the extended frame has 29 valid bits.



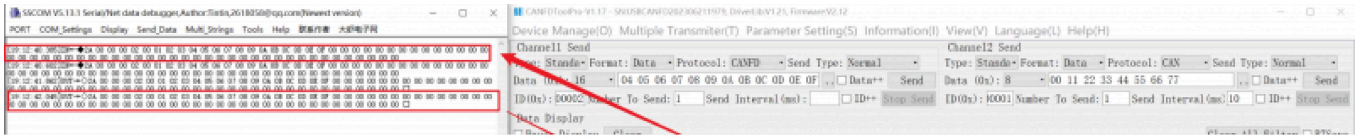
(/wiki/File:600px-

CAN_FD_Converter_Graphic_illustration_of_standard_protocol_conversion_3.jpg)

Frame data: In CAN mode, the length is 8 bytes, and in CANFD mode, the length is 64 bytes; if insufficient, it must be padded with 00.

Conversion Examples

Serial port:



(/wiki/File:800px-CAN_FD_Converter_Serial_port_conversion_mode_test_3.png)

Modbus Protocol Conversion

The Modbus protocol conversion can convert the standard Modbus data protocol into a specific CAN(FD) data format. This conversion generally requires the CAN(FD) bus device messages to be editable.

On the CAN side, a simple and user-friendly segmented communication format has been developed to implement Modbus communication. The role of the converter remains protocol validation and forwarding. It supports the transmission of Modbus protocol, instead of the host or slave of Modbus, and the user can communicate according to the Modbus protocol.

Note:

(1) In this conversion mode, the "CAN ID" of the "CAN Parameter" item of the configuration software is invalid because the frame ID sent at this time is populated by the address field in the Modbus RTU serial frame.

Frame Format

Modbus frame format:

Network port: Modbus TCP protocol, serial port: Modbus RTU protocol, so user frames only need to comply with this protocol.

CAN frame:

A segmentation protocol format is designed on the CAN side, which defines a segmentation

and reorganization method, as shown below. Among them, the CAN frame information (remote frame or data frame; standard frame or extended frame) is set through configuration software.

In CAN mode, the transmitted Modbus protocol content can start from the "Data 2" byte. If the protocol content is greater than 7 bytes, then the remaining protocol content will continue to be converted according to this segmented format until the conversion is completed.

Bit No	7	6	5	4	3	2	1	0
Frame info	FF	RTR	EDL	BRS	DLC (Data length)			
Frame ID1	X	X	X	ID.28-ID.24				
Frame ID2	ID.23-ID.16							
Frame ID3	ID.15-ID.8							
Frame ID4	ID.7-ID.0 (Address code of Modbus)							
Data 1	Segment marker	Segment type	Segment counter					
Data 2	Character 1							
Data 3	Character 2							
Data 4	Character 3							
Data 5	Character 4							
Data 6	Character 5							
Data 7	Character 6							
Data 8	Character 7							

(/wiki/File:CAN_FD_Converter_Graphic_illustration_of_Modbus_Protocol_Conversion_1.jpg)

In CANFD mode, the transmitted Modbus protocol content can start from the "Data 2" byte. If the protocol content is greater than 63 bytes, then the remaining protocol content will continue to be converted according to this segmented format until the conversion is completed.

Bit No.	7	6	5	4	3	2	1	0
Frame info	FF	RTR	EDL	BRS	DLC (Data length)			
Frame ID1	X	X	X	ID.28-ID.24				
Frame ID2	ID.23-ID.16							

(/wiki/File:600px-CAN_FD_Converter_Graphic_illustration_of_Modbus_Protocol_Conversion_2.jpg)

Data 1 is the segment control information (1 byte, 8 bits), which means the following:

- Segment mark: Occupies 1 bit (Bit7), indicating whether this message is a segmented message. This bit is 0 to indicate a standalone message, and 1 to indicate a frame belonging to a segmented message.
- Segment type: Occupies 2 bits (Bit6, Bit5), used to indicate the type of this message in segmented messages:
- Segment counter: Occupies 5 bits (Bit4~Bit0), indicating the sequence number of the segment in the entire message. If it is the nth segment, then the value of the counter is n. This allows you to verify that any segments are missing at the time of receipt.

Bit value	Meaning
00	First segment
01	Middle segment
10	Last segment

(/wiki/File:CAN_FD_Converter_Graphic_illustration_of_Modbus_Protocol_Conversion_3_1.jpg)

Modbus RTU 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 this frame.

CRC check bytes do not need to be converted into CAN messages, and the CAN message does not need to include the check bytes of the serial frame, because the CAN bus itself has a good check mechanism.

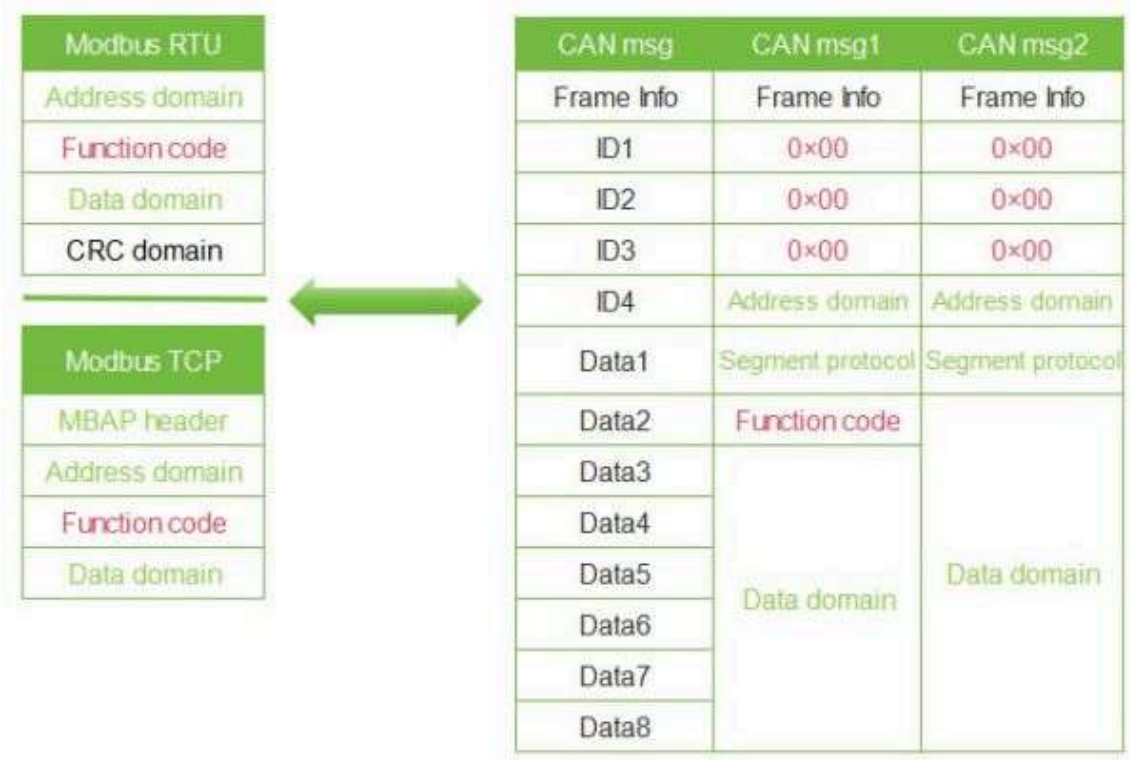
The conversion involves the Modbus protocol content—function code and data field—where they are sequentially converted into the data field of a CAN message frame (starting from the second data byte, with the first data byte used for segmented protocol). Since the length of Modbus frame varies depending on the function code, while a CAN message frame can only transmit 7 data, and a CANFD message frame can only transmit 63 data. Therefore, the converter will convert the longer Modbus frame segments into CAN messages and send them using the CAN fragmentation protocol described above. Users can process the function code and data field when receiving them on the CAN node.

CAN(FD) to Modbus RTU

For Modbus protocol data on the CAN bus, there is no need to perform cyclic redundancy check (CRC16). The converter receives it according to the segmentation protocol and automatically adds cyclic redundancy check (CRC16) after parsing one frame, then convert it into a Modbus frame and send it to the serial bus.

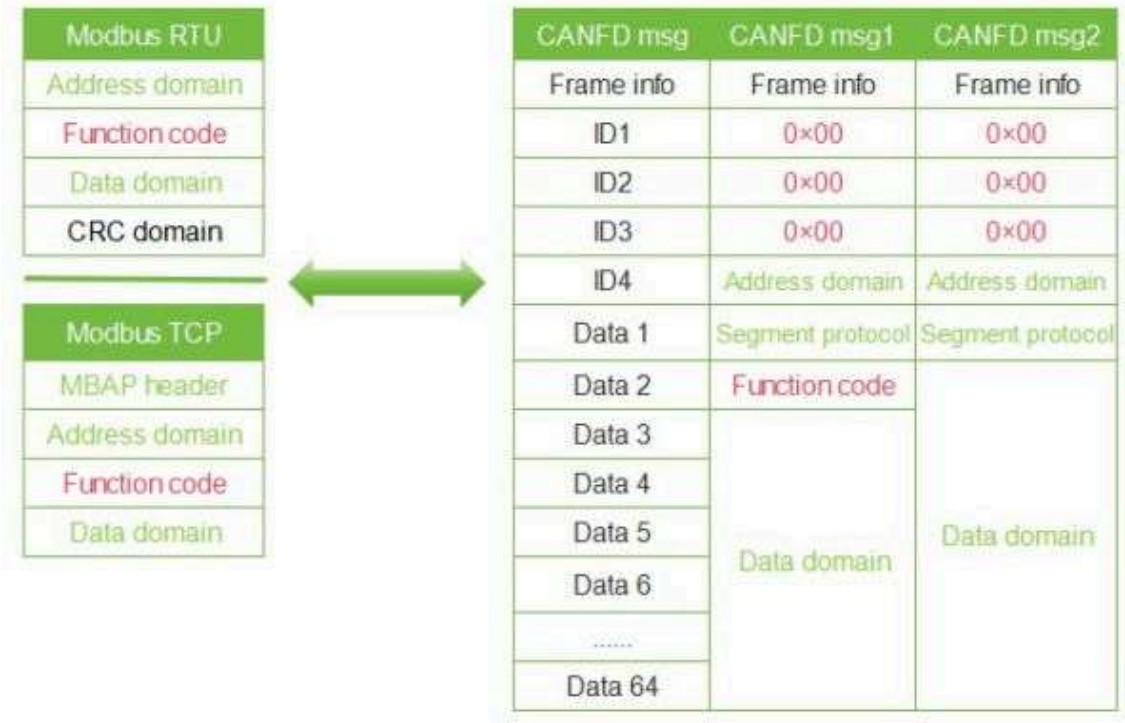
Note: If the received data does not comply with the segmentation protocol, discard this group of data without conversion.

CAN:



(/wiki/File:CAN_FD_Converter_Graphic_illustration_of_Modbus_Protocol_Conversion_4.jpg)

CAN FD:



(/wiki/File:CAN_FD_Converter_Graphic_illustration_of_Modbus_Protocol_Conversion_5.jpg)

Conversion Examples

Serial port:

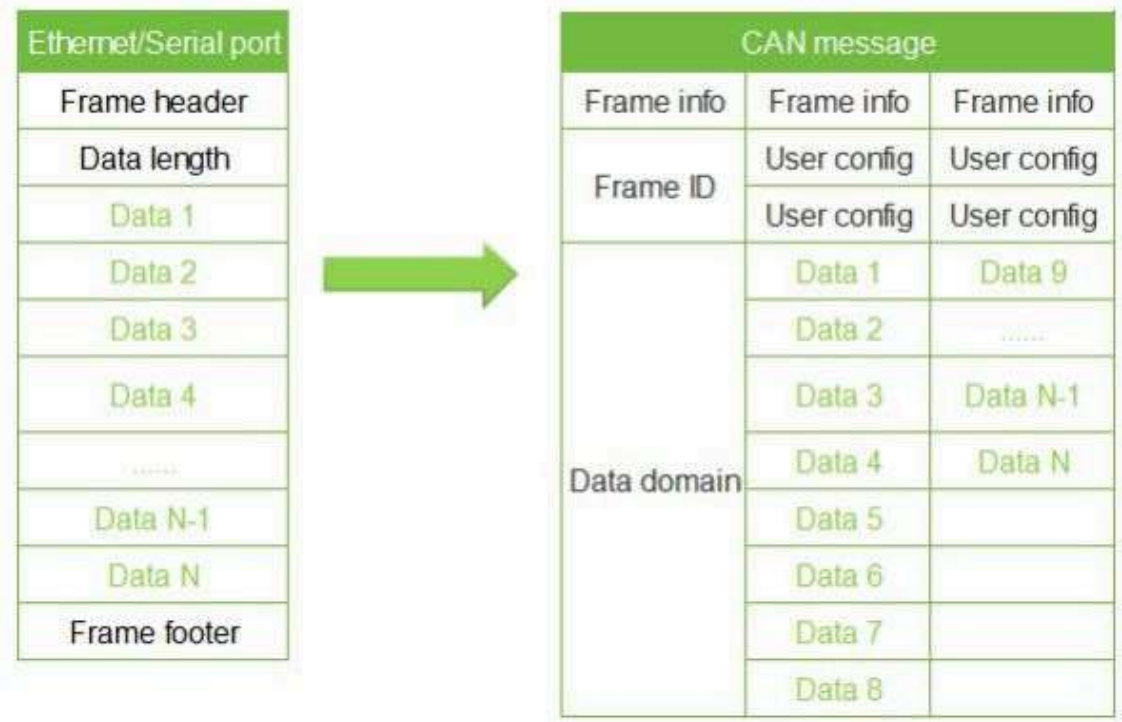
(/wiki/File:800px-CAN_FD_Converter_Serial_port_conversion_mode_test_4.png)

Custom Protocol Conversion

To make it convenient for users to use CAN-bus, the serial frame format is aligned with the CAN frame format. In the serial frame, the start and end of a frame are specified, namely the "frame header" and "frame footer", which can be configured by the user.

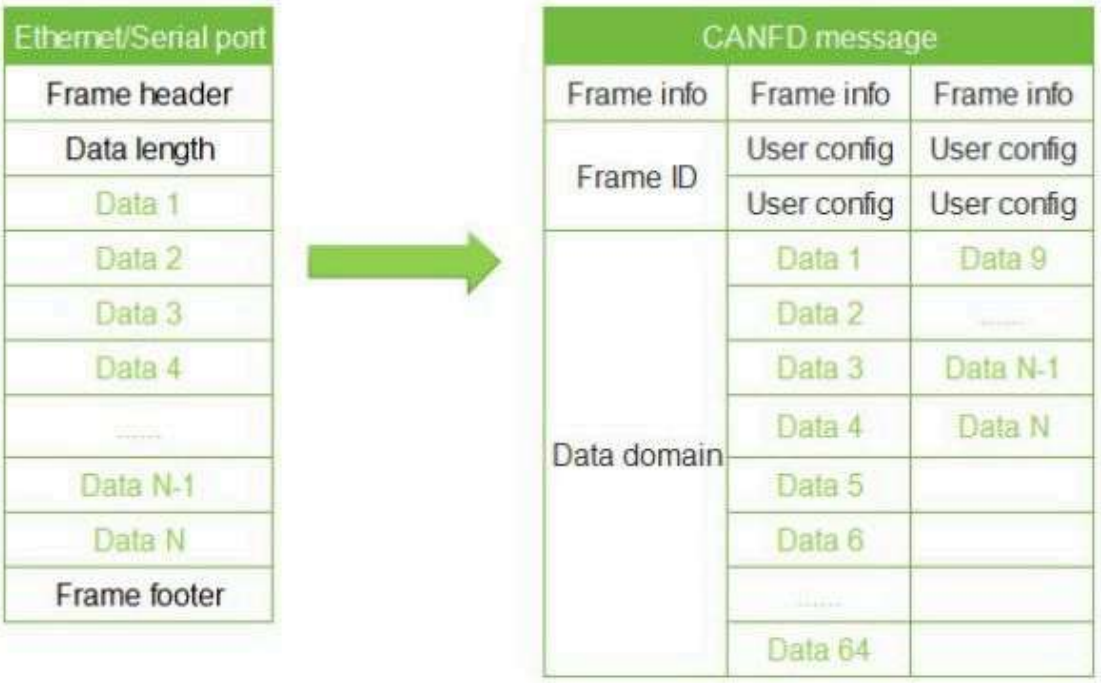
Serial Port to CAN(FD)

The serial frame format must comply with the specified frame format, otherwise it cannot be transmitted correctly. A serial frame must contain: header, data length, data field, and frame footer. The frame header and footer are customized by the user and consist of 1 byte.
Data length refers to the byte length of a data field. The data length and frame end data must match to be transmitted correctly, otherwise they are discarded. For example, the frame header is configured as AA and the frame footer is configured as FF. For the serial frame AA03010204FF, it can be transmitted normally. If the serial frame AA0301020304FF is to be sent, and the data field 010203 is followed by 04 instead of the frame footer FF, the frame is discarded and cannot be transmitted.
As with transparent conversion, the CANID and CAN types need to be configured by yourself in custom protocol conversion. The frame header, frame footer, and data length are not converted into CAN frames.
CAN:



(/wiki/File:600px-CAN_FD_Converter_Graphic_illustration_of_Custom_frame_header_and_trailer_conversion_1.jpg)

CAN FD:

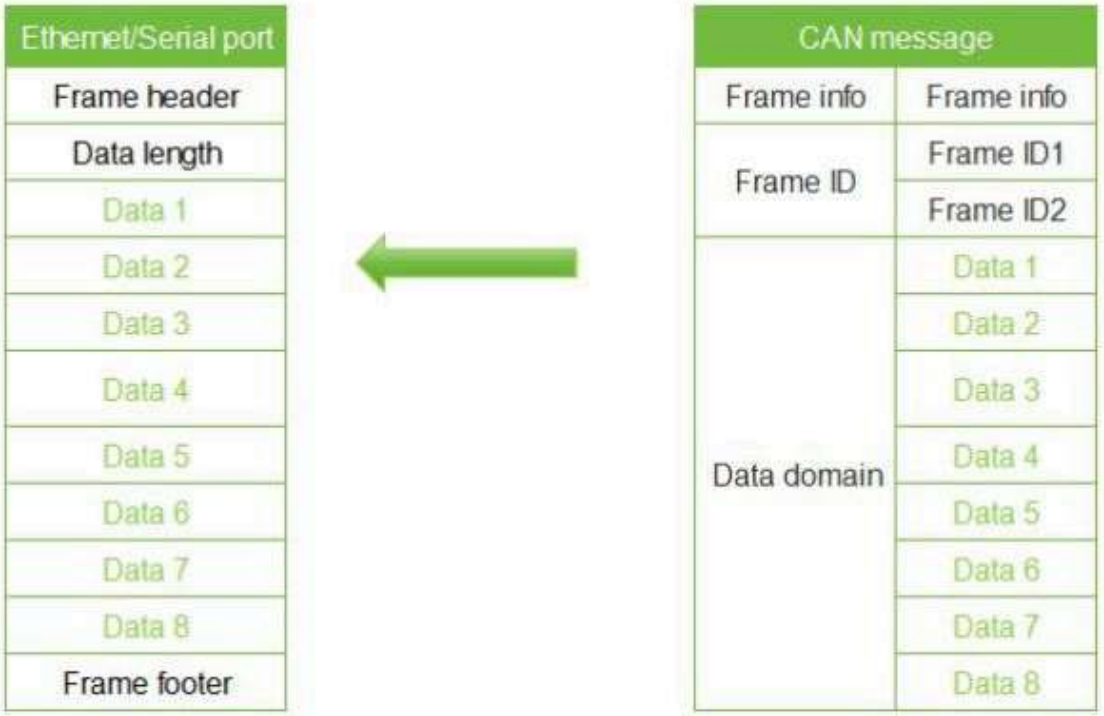


(/wiki/File:600px-CAN_FD_Converter_Graphic_illustration_of_Custom_frame_header_and_trailer_conversion_2.jpg)

Serial Port to CAN(FD)

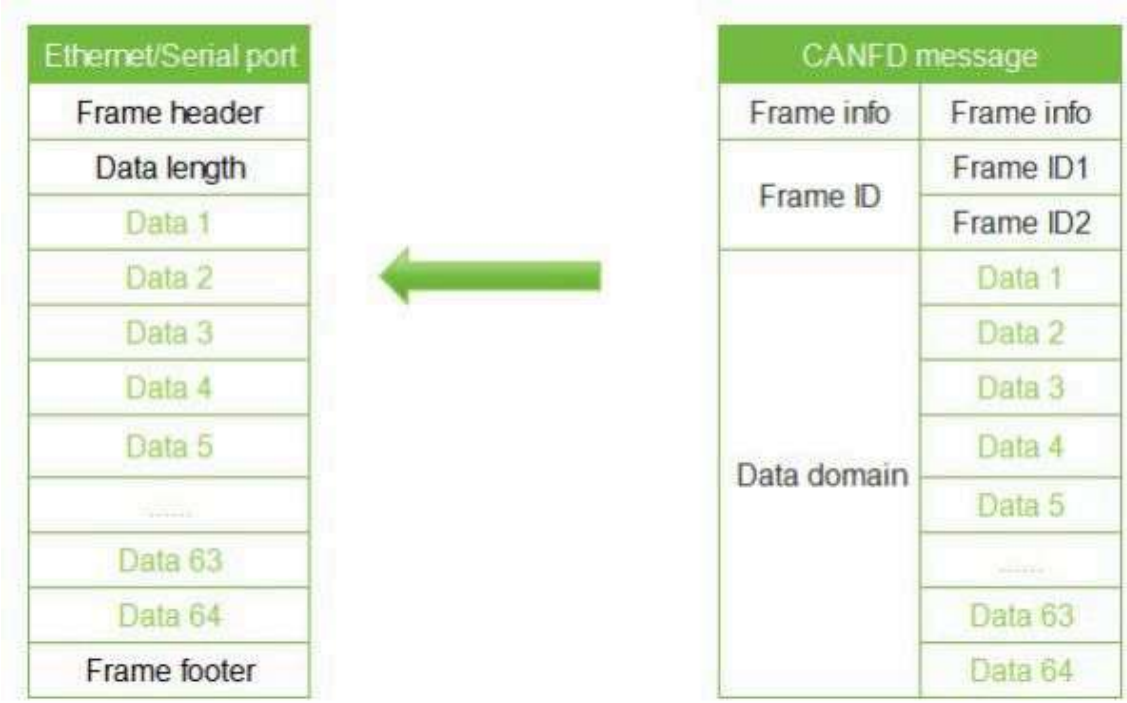
The CAN(FD) bus message forwards one frame for every frame received. The module will sequentially convert the data in the CAN message data field, while automatically adding frame header, frame length, frame information, and other data to the serial frame. This is essentially the reverse form of converting a serial frame to a CAN message.

CAN:



(/wiki/File:600px-CAN_FD_Converter_Graphic_illustration_of_Custom_frame_header_and_trailer_conversion_3.jpg)

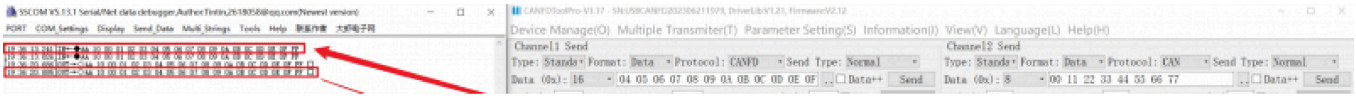
CAN FD:



(/wiki/File:600px-CAN_FD_Converter_Graphic_illustration_of_Custom_frame_header_and_trailer_conversion_4.jpg)

Conversion Examples

Serial port:



(/wiki/File:800px-CAN_FD_Converter_Serial_port_conversion_mode_test_5.png)

Resources

Software

- Configuration software WS-CAN-FD-Converter (https://files.waveshare.com/wiki/CAN-FD-ETH-RS232-RS485-Converter/tool/WS-CAN-FD-Converter_v1.3.9.0.exe)
- Sscom software (https://files.waveshare.com/wiki/common/Sscom5.13.1.zip)

FAQ

Question: After configuring the device, the CAN side sends a message, but the device does not respond. After the serial port side sends a message, the device replies directly with err -1 (ASCII code). How should this be handled?

Answer:

This situation usually occurs when you do not exit the configuration state after configuring the device, and you need to use the configuration software to exit the configuration mode

Question: What should I do if the configuration cannot be saved?

Answer:

Check if there are any missing configuration items

Support

Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our

support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.
Working Time: 9 AM - 6 PM GMT+8 (Monday to Friday)

Submit Now (https://service.waveshare.com/)

Retrieved from "https://www.waveshare.com/w/index.php?title=CAN_FD_TO_RS232/RS485&oldid=109149 (https://www.waveshare.com/w/index.php?title=CAN_FD_TO_RS232/RS485&oldid=109149)"