

# CANFD COM-100IE

RS232/485/422-CAN(FD)-BUS Intelligent Protocol converter

UM01010101 V1.02 Date: 2019/08/20

Product User Manual

Category	Contents
Keywords	RS232, CAN, conversion, baud rate
Description	This converter quickly connects RS232/485/422 communication equipment to the CAN(FD)-BUS field bus

# CANFDNET-400U

High-performance Four-channel CANFD Bus-to-Ethernet Converter User Manual User Manual

---

## Revision History

Version	Date	Reason
V1.00	April 11, 2018	Created
V1.01	March 15, 2019	Updated the document header and footer and "Sales and Service Network" content and added the "Disclaimer" content
V1.02	August 19, 2019	Added the product mechanical dimensions

## Contents

1. Functions .....	1
1.1 Overview.....	1
1.2 Features.....	1
1.3 Typical Applications .....	1
2. Hardware .....	3
2.1 Appearance .....	3
2.2 Interfaces.....	3
2.3 Indicators .....	4
2.4 Buttons.....	5
2.5 CAN Bus Connection .....	5
2.6 Mechanical Dimensions .....	6
3. Configuration Tool Installation and Description .....	8
3.1 Entering Configuration Mode.....	8
3.2 PC Software Configuration.....	8
3.2.1 Conversion Parameters .....	9
3.2.2 Serial Port Parameters.....	11
3.2.3 CAN Parameters .....	11
3.2.4 Filter Parameters.....	14
3.2.5 Button Description.....	14
4. Conversion Modes .....	16
4.1 Transparent Conversion .....	17
4.1.1 Frame Formats.....	17
4.1.2 Conversion Methods .....	17
4.1.3 Conversion Examples .....	20
4.2 Transparent Conversion with Flags.....	23
4.2.1 Frame Formats.....	23
4.2.2 Conversion Methods .....	24
4.2.3 Conversion Examples .....	27
4.3 Format Conversion .....	30
4.4 Modbus Conversion.....	32
4.4.1 Frame Formats.....	32
4.4.2 Conversion Methods .....	34
4.4.3 Conversion Examples .....	36
5. Quick Instructions .....	38
5.1 Configuring the Converter .....	38
5.1.1 Entering Configuration Mode .....	38
5.1.2 Connecting the Converter by Using the Configuration Tool .....	38
5.1.3 Setting the Save Parameters .....	39
5.2 Communication Test .....	39
6. Firmware Upgrade .....	41
7. Disclaimer .....	43

## 1. Functions

### 1.1 Overview

CANFDCOM-100IE intelligent protocol converter quickly connects RS-232/485/422 communication equipment to the CAN(FD)-BUS field bus.

The converter supports 1200-921600bps serial port baud rate, 50k-5Mbps CAN (FD)-bus communication rate.

The converter supports four conversion modes: transparent conversion, transparent conversion with flags, format conversion, and Modbus conversion.

The CANFDCOM-100IE converter provides configuration software, which can flexibly set the operating parameters of the CANFDCOM-100IE.

### 1.2 Features

- ◆ Realize bidirectional data communication between CAN (FD)-bus and RS-232/485/422;
- ◆ Support CAN2.0A and CAN2.0B protocols, and comply with ISO-119898 specification;
- ◆ Integrate one CAN-bus communication interface, and support user-defined communication baud rate;
- ◆ CANFDCOM-100IE integrates one 3-wire RS-232 communication interface, one 2-wire RS-485 communication interface, and one 4-wire RS-422 communication interface. The communication rate can be set between 1,200-921,600 bps;
- ◆ Provides four data conversion modes: transparent conversion, transparent conversion with flags, format conversion, and Modbus conversion;
- ◆ The CAN-bus circuit adopts 2,500 V DC electrical isolation;
- ◆ Can be used in environments with safety and explosion-proof requirements;
- ◆ Operating temperature: -40°C to +85°C;
- ◆ Operating power: less than 2 W;

### 1.3 Typical Applications

- Coal mine telecommunication;
- PLC equipment networking;
- The existing RS-232/485/422 equipment is connected to the CAN (FD)-bus network;
- Extended standard RS-232/485/422 network communication length;
- The PLC equipment is connected to the CAN(FD)-bus network;
- Gateway bridge between CAN(FD)-bus and serial bus;
- Industrial field network data monitoring;
- CAN (FD) teaching application remote communication;
- CAN (FD) industrial automation control system;

# CANFDNET-400U

High-performance Four-channel CANFD Bus-to-Ethernet Converter User Manual User Manual

---

- Low-speed CAN (FD) network data acquisition data analysis;
- CAN(FD)-bus application system such as the intelligent building control data broadcasting system;

# CANFDNET-400U

## 2. Hardware

### 2.1 Appearance

Figure 2.1 shows the appearance.



Figure 2.1 CANFDCOM-100IE appearance

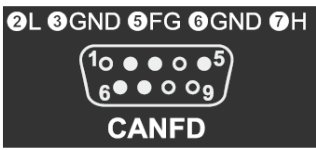

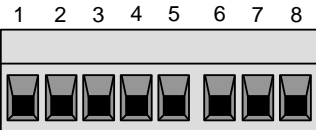
### 2.2 Interfaces

The CANFDCOM-100IE converter has three user interfaces: one CAN (FD)-bus interface, one 485/422 interface, one RS-232 interface. Table 2.1 lists the interface pin

# CANFDNET-400U

definitions.

Table 2.1 Interface description

Type	Schematic Diagram	Pin definitions	Pin description
DB9 socket		1: NC	Dangling
		2: CAN_L	CAN data transceiving differential inverted signal
		3, 6: CAN_GND	CAN isolated ground
		4, 8: NC	Dangling
		5: Shield	Shielding ground
		7: CAN_H	CAN data sending and receiving differential positive phase signal
		9: NC	Dangling
RS232 terminal (female)		2:TXD	RS232 send
		3:RXD	RS232 receive
		5:RS232_GND	RS232 isolated ground
		1, 4, 6, 7, 8, 9:NC	Dangling
OPEN8 terminal		1:485A_R+	485A/422 receive positive phase signal
		2:485B_R-	485B/422 receive inverted signal
		3:485\422_GND	485\422 isolated ground
		4:422_T+	422 send a positive phase signal
		5:422_T-	422 send inverted signal
		6:GND	Power input negative
		7:EARTH	Ground
		8:9V-48V	Power input positive

## 2.3 Indicators

There are four LEDs on the converter to indicate the operating status of the CANFDNET-400U converter. Table 2.2 lists the functions.

Table 2.2 LED indicators

Identification	Function	Status	Description
PWR	Converter power indicator	The indicator is on	The converter power supply is working properly
		The indicator is off	The converter power supply is not powered

# CANFDNET-400U

		is off	
CAN	CAN communication status indicator	Flashing red light	There is an error in CAN communication, and an error frame is received
		Green flashing light	The CAN side is transmitting data properly
COM	RS-232/485/422 communication status indication	Flashing red light	Baud rate mismatch error occurs in serial communication
		Green flashing light	The serial port side is transmitting data properly
CFG	Configuration status indicators	The indicator is off	In conversion working mode, normal data forwarding
		Normally on	In configuration mode. You can configure parameters and perform other operations
		Flash	In BootLoader, firmware upgrade in progress

## 2.4 Buttons

Table 2.3 Button operations

Operation	Function
ON	Make the converter in normal working state and perform normal data forwarding
CFG	Put the converter in the configured state. Normal data conversion is impossible in the configured state. Only in the configuration state can the host computer or AT commands be used for parameter configuration

## 2.5 CAN Bus Connection

The physical layer mainly transmits inter-device signals, converts various information into transmittable physical signals (usually electrical or optical signals), and transmits these signals to other target devices. For this purpose, CAN-bus has made detailed regulations on signal levels, cables and connectors used for communication.

Two standards are released after CAN-bus was standardized by ISO, namely ISO11898-2 (125kbps-1Mbps high-speed communication standard) and ISO11898-3 (low-speed communication standard less than 125 kbps).

The data link layer is the same for both high-speed CAN and fault-tolerant CAN except for the physical layer. The CAN transceiver determines the bus level based on the voltage difference between the two cables. This transmission method is called differential transmission. There are only two possibilities for the level signal transmitted on the cable, namely the dominant level and the recessive level. The dominant level indicates logic 0, and the recessive level indicates logic 1. Table 2.4 lists the high-speed CAN electrical characteristics.



Table 2.4 CAN-BUS interface specifications

parameter		Minimum	Typical value	Maximum	Unit
Communication baud rate		5k		5M	bps
Number of nodes				110	pcs
Dominant level (logic 0)	CANH	2.75	3.5	4.5	V
	CANL	0.5	1.5	2	
Recessive level (logic 1)	CANH	2	2.5	3	
	CANL	2	2.5	3	
Differential level	Dominant (logic 0)	1.2	2	3.1	
	Recessive (logic 1)	-0.5	0	0.05	
Maximum withstand voltage of the bus pin		-18		18	
Instantaneous voltage of the bus		-100		+100	
Isolation voltage (DC)		2500			V

The CAN bus adopts balanced transmission. ISO11898-2 stipulates: In the high-speed CAN, a 120 ohm terminal resistor needs to be connected to the network terminal node to eliminate signal reflection on the bus and avoid signal distortion. Figure 2.2 shows the high-speed CAN network topology.

The device has a built-in 120-ohm terminal resistance, which can be configured to be turned on or off by using the CANFDCOM configuration tool. For operation details, see 3.2.3.

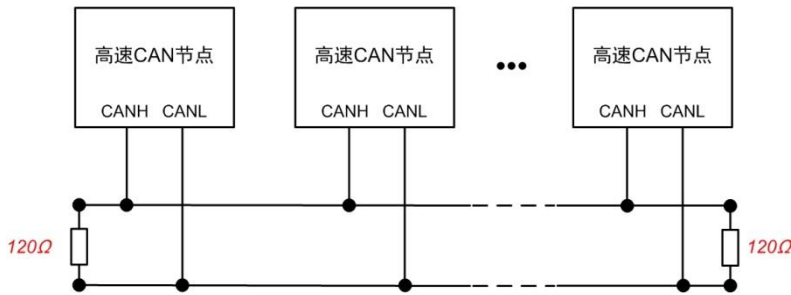


Figure 2.2 High-speed CAN network topology

Note: The bus communication distance and communication rate are related to the field application and can be designed according to the actual application and related standards. The CAN-Bus cable can use ordinary twisted pair, shielded twisted pair or standard bus communication cable. In long-distance communication, the terminal resistance value needs to be selected according to the communication distance, cable impedance and number of nodes.

## 2.6 Mechanical Dimensions

Figure 2.3 shows the mechanical dimensions (unit: mm).

# CANFDNET-400U

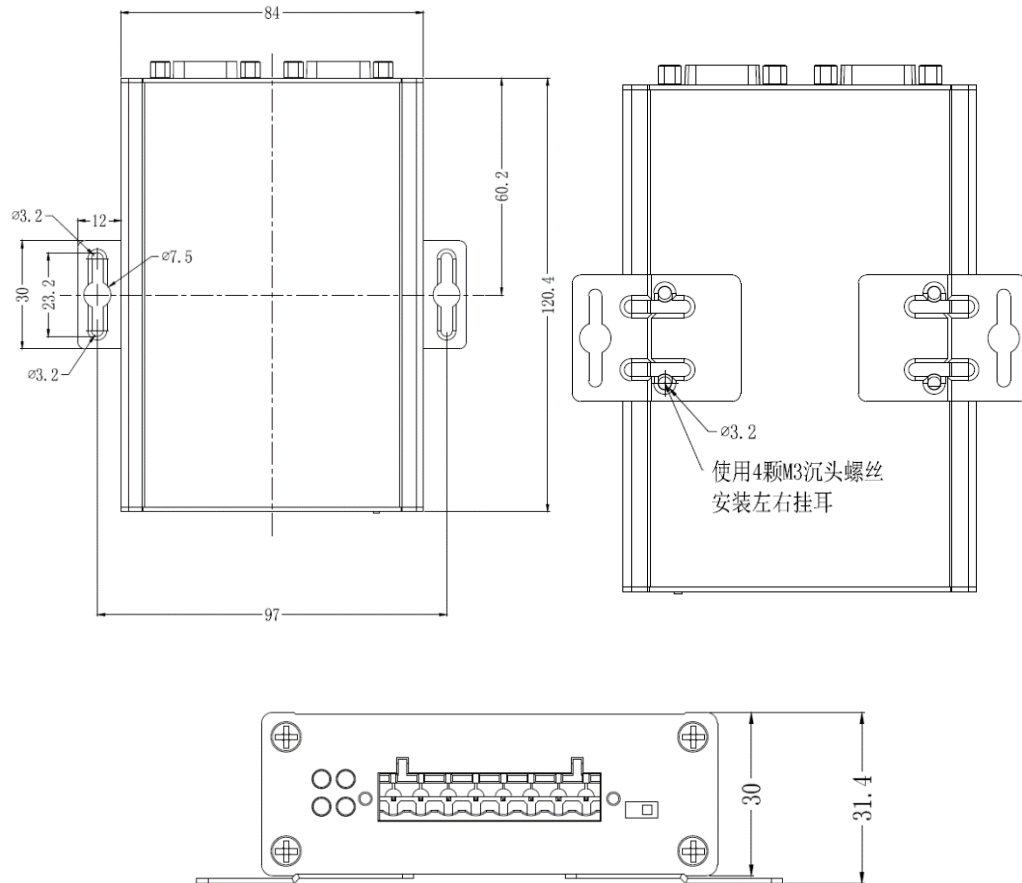


Figure 2.3 Mechanical dimensions

## 3. Configuration Tool Installation and Description

The operating parameters of CANFDCOM-100IE support two configuration methods. The first is to use the host computer software, and the second is to use AT commands. For how to use AT command for configuration, see the *CANFDCOM Series AT Command Instructions*. Both configuration methods require the repeater to enter configuration mode.

### 3.1 Entering Configuration Mode

To put the converter into configuration mode, a dedicated configuration switch is set, as shown in Figure 3.1. Set the switch to the "CFG" position. The system is in the configuration state, and the CFG indicator is on. The converter can be configured. **After configuring the device, set the "configuration switch" to "ON"! In normal operation.**



Figure 3.1 Configuration switch

### 3.2 PC Software Configuration

The configuration software name of the CANFDCOM-100IE converter is "CANCOM-100IE configuration". The setting software is included on the CD-ROM delivered with the product. Figure 3.2 shows the software interface.



Figure 3.2 Configuration software interface

After the converter enters the configuration mode, the parameters can be set by using the software; otherwise, the software will consider the converter not connected. After setting the configuration parameters, click Write Configuration to write the parameters to the device for curing. The following describes the main configuration parameters in detail with reference to the configuration software.

### 3.2.1 Conversion Parameters

Conversion parameters refer to parameters such as the conversion rule direction of the converter. Figure 3.2 shows the conversion parameter interface.

Conversion mode: includes four selectable conversion modes: transparent conversion, transparent conversion with flags, format conversion, and Modbus mode.

Conversion direction:

Bidirectional: The converter converts the data of the serial bus to the CAN bus, and also converts the data of the CAN bus to the serial bus.

Only serial port to CAN: Only convert the data of the serial bus to the CAN bus, without converting the data of the CAN bus to the serial bus.

Only CAN to serial port: Only convert the data of the CAN bus to the serial bus, without converting the data of the serial bus to the CAN bus.

[Note]: By selecting the conversion direction, the data interference on the bus side that does not need conversion can be eliminated.

Allow CAN frame information to be forwarded into serial frames:

This parameter is **only used in "transparent conversion" mode**. When this parameter is selected, the converter will add the frame information of the CAN (CANFD) message to the first byte of the serial frame when it is working. When this parameter is not selected,

the frame information of CAN (CANFD) will not be converted.

Allow CAN frame flags to be forwarded into serial frames:

This parameter is **only used in "transparent conversion" mode**. When this option is selected, the converter will add the frame ID of the CAN (CANFD) message before the frame data of the serial frame and after the frame information (if frame information conversion is allowed). When this parameter is not selected, the frame ID of CAN (CANFD) is not converted.

Position of the CAN frame identifier in the serial frame:

This parameter is only used in transparent conversion with flags. When the serial port data is converted into CAN (CANFD) messages, the offset address of the start byte of the frame ID of the CAN (CANFD) message in the serial frame and the length of the frame ID (see 4.2 Transparent Conversion with Flags).

[Note]: The frame ID length can be filled with 1 to 2 bytes in the standard frame, corresponding to ID1 and ID0 of the CAN (CANFD) message respectively. When extending the frame, 1 to 4 bytes of ID3, ID2, ID1 and ID0 can be filled.

The ID of the standard frame is 11 bits, and the ID of the extended frame is 29 bits.

Time interval (ms) between serial frames:

This parameter is used **only in transparent conversion and transparent conversion with flags**. This parameter is the minimum time interval between two serial frames when the user sends serial frames to the converter. The interval is in "milliseconds". The value is 0-500 ms.

[Note]: This time cannot be less than the time to transmit a single character; otherwise, the repeater will be automatically set to an appropriate value (two characters of time). The frame interval time sent by the user is preferably greater than the set time; otherwise, the frame conversion may be incomplete.

The meaning of "time to transmit a single character" is: at the corresponding baud rate, the time required for the serial port to transmit one character (10 bits); that is, divide 10 by the corresponding baud rate.

For example: at a baud rate of 9,600 bps, the "number of characters in the serial frame interval" is 4, and the "time to transmit a single character (10 bits per character)" is  $(10/9600)s$ , the obtained actual time interval between serial frames is:  $(10/9600)*4=4.17(ms)$ ; that is, the time interval between two serial frames is at least 4.17 ms.

CAN frame forwarding interval (ms):

This parameter is valid in all modes. This parameter is the delay time between forwarding CAN (CANFD) messages from the repeater. If this parameter is greater than 0,

# CANFDNET-400U

the repeater will buffer the data received by the serial port, and forward the data by message at a time based on the preset delay time. The value is 0-500 ms. **Note: The repeater cache is limited. If the preset delay time is too long, it may cause too much data accumulation at the serial port, which results in overflow and loss.**

## 3.2.2 Serial Port Parameters

As shown in Figure 3.3, this option mainly sets serial port related parameters.



Figure 3.3 Serial port parameter settings

**Baud rate:** The baud rate of the serial port ranges between 1,200 bps and 921,600 bps.

**The data is:** between 7 and 8 bits.

**Stop bit:** 1 or 2.

**Parity:** no parity, even parity, or odd parity.

**Mode switching:** RS232 mode, RS485 mode, or RS422 mode.

## 3.2.3 CAN Parameters

As shown in Figure 3.4, this option mainly sets CAN related parameters.



Figure 3.4 CAN parameter setting

CANFD standard: supports ISO standard and Bosch standard

Protocol: This parameter affects the transmission on the CAN side. CAN or CANFD can be selected. When CAN is selected, the repeater forwards the serial port data into CAN messages; when CANFD is selected, the repeater forwards the data as CANFD messages.

Frame Type: frame type of the CAN (CANFD) message at the time of conversion. The options include standard frames and extended frames.

CANFD acceleration: This parameter enables the baud rate switching function of CANFD.

CAN ID: In "transparent conversion" mode, the forwarded CAN (CANFD) message uses this ID. In other modes, the serial port data has an ID parameter, so this parameter is invalid.

Terminal resistance enable: This parameter enables the built-in CAN bus 120-ohm terminal resistance of the device.

Baud rate: includes the arbitration domain baud rate and data domain baud rate. In CAN mode, the baud rate is determined by the arbitration domain baud rate, while the data domain baud rate is invalid. In CANFD mode, if CANFD acceleration is enabled, the data domain baud rate is valid. CAN bus baud rate. In addition to the standard baud rate recommended by CIA in the list (sampling point 75-83.5%, SJW = 2, 3), a "custom baud rate" option is also given. After selecting a custom baud rate, use the baud rate calculator to calculate the desired baud rate value and fill it in the custom baud rate box as needed. The baud rate calculator method is as follows:

1. As shown in Figure 3.5, set the arbitration domain baud rate in ①, and select the appropriate synchronization jump width. The clock is fixed at 60 MHz;
2. Set the baud rate parameter of the data domain in ②, and select the appropriate

synchronization jump width. The clock is fixed at 60 MHz;

3. After setting, click the calculation button at ③ to list the calculation results of the corresponding baud rate parameters;
4. Select the arbitration domain baud rate for the appropriate sampling point. If the value is selected, the blue background color appears, indicating that the value is selected, as shown in ④;
5. Select the data domain baud rate for the appropriate sampling point. If the value is selected, the blue background color appears, indicating that the value is selected, as shown in ⑤;
6. Click the copy button at ⑥ to copy the value of the custom baud rate. Paste this value into the custom baud rate text box.



Figure 3.5 Steps for using the baud rate calculator



### 3.2.4 Filter Parameters



Figure 3.6 Baud parameter settings

CANFDCOM-100IE has the capability of hardware acceptance filtering and **selective reception**, which minimizes the network load from the network. When setting the acceptance filter, switch to the Filter Settings tab, as shown in Figure 3.6. Select the "Enable Filter" option to enable the acceptance filter function. The filtering mode of CANFDCOM-100IE is whitelist filtering. After filtering is enabled, only CAN (CANFD) messages within the ID range of each filtering item in the filtering table are received. **Note: If filtering is enabled, but there is no filtering item in the filtering table, all packets will be filtered at this time.** The maximum number of filter items can be set to 64.

Example: As shown in Figure 3.6, the acceptance criteria frame IDs are set to 0x08 and 0x12. Extended frame group IDs are 0x55 to 0x66, and standard frame group IDs are 0x22 to 0x66. The repeater only receives standard frame CAN (CANFD) messages with IDs 0x08, 0x12, 0x22-0x66 and extended frame CAN (CANFD) messages with IDs 0x55-0x66.

### 3.2.5 Button Description

Export configuration: Export its parameters to a parameter file, and save it.

Import configuration: Import the parameter file, and use the parameters in the parameter file.

Restore default: Restore its parameters to their default values.

Firmware upgrade: Provides the function of upgrading the CANFDCOM-100IE firmware.

Get configuration: Read out the parameters of the converter and display them on the host computer software.

# CANFDNET-400U

High-performance Four-channel CANFD Bus-to-Ethernet Converter User Manual User Manual

---

Save configuration: After the parameters are set, click this button to write the configuration parameters into the converter.

## 4. Conversion Modes

CANFD-100IE converter is an intelligent protocol converter that supports conversion between serial port and CAN and conversion between serial port and CANFD. Before use, you need to set the CAN type to be converted to the normal CAN or CANFD. The converter provides four conversion modes, including transparent conversion, transparent conversion with flags, format conversion, and Modbus conversion. The parameters can be selected and set during converter configuration. Note: In the same conversion mode, the CAN type is different, and the conversion method will change. For example, in "transparent conversion", for ordinary CAN, the serial port receives a maximum of eight bytes and must be converted into a CAN message. For CANFD, a maximum of 64 bytes of data can be received and converted into CANFD messages.

"Transparent conversion" means that the converter just converts the bus data of one format to the data bus of another format without appending or modifying the data. This realizes the exchange of data format without changing the data content. The converter is transparent to the bus at both ends. This method will not increase the user's communication burden, but can convert the data as it is in real time, and can undertake the transmission of data with a large flow.

**"Transparent conversion with flags"** is a special transparent conversion, with no protocol attached. This conversion method is based on the common characteristics of common serial frames and CAN (CANFD) messages, so that the two bus types can easily form the same communication network. This method can convert the "address" in the serial frame into the identification field of the CAN (CANFD) message. The starting position and length of the serial frame "address" in the serial frame can be configured. Therefore, in this way, the converter can adapt to the user's custom protocol to the greatest extent.

**"Format conversion"** is one of the simplest usage patterns. The data is in a fixed length. The fixed length of CAN is 13 bytes, and the fixed length of CANFD is 69 bytes. That is, when the CAN type is set to ordinary CAN, the fixed serial frame data of 13 bytes corresponds to a CAN message. When it is CANFD, the fixed 69-byte serial frame data corresponds to one CANFD message. The fixed serial port frame content includes frame information (1 byte) + ID (4 bytes) + data (8 bytes for CAN, 64 bytes for CANFD). By correctly configuring the frame information (the data of the first byte), when the CAN type is ordinary CAN, standard frames, extended frames and even remote frames of CAN can be flexibly sent. When it is CANFD, it can send CANFD standard frame and extended frame. Details of standard frames, extended frames and even remote frames can be obtained by correctly parsing fixed-byte serial frames.

**"Modbus conversion"** refers to the conversion between the UART data of the Modbus protocol and the CAN data. The Modbus protocol is a standard application layer protocol, which is widely used in various industrial control scenarios. The protocol is open, real-time, and has a good communication verification mechanism. It is ideal for scenarios with high communication reliability requirements. The converter uses the standard

Modbus RTU protocol format on the serial port side. Therefore, the converter not only allows users to use the Modbus RTU protocol, but the converter can also directly interface with other devices that support the Modbus RTU protocol. On the CAN or CANFD side, an easy-to-use segmented communication format is developed to realize Modbus communication. The role of the converter is still protocol verification and forwarding. It supports Modbus protocol transmission, not Modbus master or slave host. The user can communicate according to the Modbus protocol.

In "transparent conversion" and "format conversion", a byte of frame information is used to identify some information of the CAN frame, such as type, format, and length. Table 4.1 lists the frame information format.

Table 4.1 Frame format

FF	RTR	EDL	BRS (CANFD valid)	L3	L2	L1	L0
0: Standard frame 1: Extended frame	0: Data frame 1: Remote frame	0: CAN 1: CANFD	0: Disable acceleration 1: Enable acceleration	The length of the identification data is equivalent to the DLC			

## 4.1 Transparent Conversion

In transparent conversion mode, the converter receives the data on one side of the bus and immediately converts it and sends it to the other side of the bus. In this way, the data flow is processed, which maximizes the speed of the converter and improves the buffer utilization. The reason is that the converter is converting and sending at the same time as receiving, which frees up the buffer that can be received.

### 4.1.1 Frame Formats

#### 1. Serial bus frame

It can be a data stream or data with a protocol. According to the user-defined CAN type, it is decided to forward to CAN or CANFD message. The frame information and frame ID forwarded to the CAN (CANFD) message are defined by the user in advance.

#### 2. CAN (CANFD) bus frame

The format of CAN (CANFD) message frame remains unchanged. The frame information and frame ID need to be enabled by configuration, that is, whether to forward the frame information and frame ID to the serial frame.

### 4.1.2 Conversion Methods

In transparent conversion mode, CAN and CANFD messages are converted in roughly the same way. The main difference is that a CANFD message can be converted into more serial frame data.

#### 1. Serial frame to CAN (CANFD) message

All the data of the serial frame is sequentially filled into the data field of the CAN (CANFD) message frame. The converter receives and converts data as soon as it detects

# CANFDNET-400U

data on the serial bus. The converted CAN (CANFD) message frame information (frame type part) and frame ID come from the user's prior configuration. Moreover, the frame type and frame ID remain unchanged during the conversion.

Table 4.2 lists the corresponding format of data conversion in CAN mode. If the length of the received serial line frame is less than or equal to 8 bytes, the converter will time out based on the time interval between serial frames configured by the user. If no more data is received after the timeout, fill the characters 1 to n (n indicates the serial frame length) into the 1 to n byte positions of the data field of the CAN message in sequence (for example, n is 8 in Table 4.2)..

If the number of bytes of the serial frame is greater than 8, the repeater fills the data field of the CAN message with 8 characters for the first time, starting from the first character of the serial frame. After the data is sent to the CAN bus, the remaining serial frame data is converted and filled into the data field of the CAN message until the data is converted.

Table 4.2 Converting serial frames to CAN messages (transparent mode)

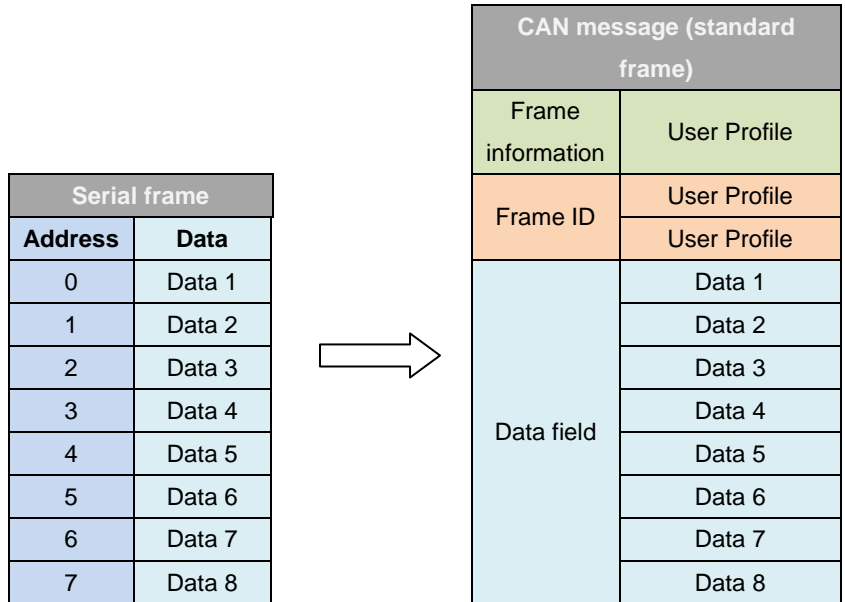
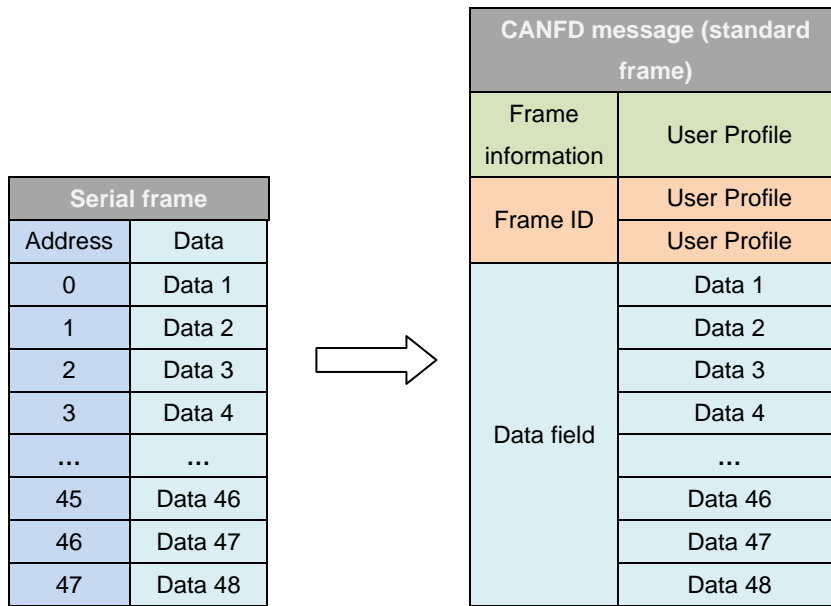


Table 4.3 lists the data conversion formats in CANFD mode. If the length of the received serial line frame is less than or equal to 64 bytes, the converter will time out based on the time interval between serial frames configured by the user. If no more data is received after the timeout, fill the characters 1 to n (n is the serial frame length) into the 1 to n byte positions of the data field of the CAN message in sequence (for example, n is 48 in Table 4.3). **Note: In CANFD mode, if the length of the serial frame exceeds 8 bytes, it must conform to the length that CANFD's DLC can encode to ensure accurate conversion; that is, the length is 12, 16, 20, 24, 32, 48, 64. Otherwise, the converter will automatically split into several CANFD messages of appropriate length. For example, if the serial frame length is 62, CANFD's DLC cannot be encoded to indicate that the length is 62, and the closest value can only be 48. Therefore, a CANFD message with a data length of 48 is converted. The remaining data length is 14. CANFD's DLC cannot be encoded to indicate**

a length of 14, and the closest value can only be 12. Therefore, a CANFD message with a data length of 12 is split. Finally, the remaining 2 bytes of data are converted into a CANFD message with a data length of 2. To sum up, the 62-byte serial frame will be split into three CANFD messages.

If the number of bytes of the serial frame is greater than 64, the processor fills the data field of the CANFD message with 64 characters for the first time, starting from the first character of the serial frame. After the data is sent to the CANFD bus, the remaining serial frame data is converted and filled into the data field of the CANFD message until the data is converted.

Table 4.3 Conversion of serial frames into CANFD messages (transparent conversion)



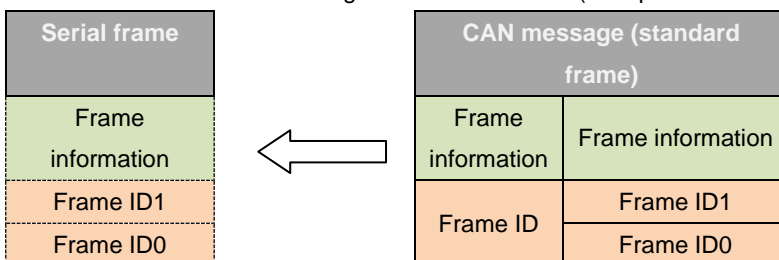
## 2. Converting CAN (CANFD) messages to serial frames

For CAN (CANFD) bus messages, one frame is immediately forwarded upon receiving one frame. For CAN messages, the corresponding data format is listed in Table 4.4. For CANFD messages, the corresponding data format is listed in Table 4.5.

In the conversion, convert all the data in the CAN (CANFD) message data field into serial frames in sequence. If "Enable" is selected for "frame information conversion enable" during configuration, the converter will directly fill the "frame information" byte of the CAN (CANFD) message into the serial frame.

If "Enable" is selected for "Enable Frame ID Conversion", the "Frame ID" bytes of CAN (CANFD) messages are also filled into serial frames.

Table 4.4 Conversion of CAN messages into serial frames (transparent conversion)



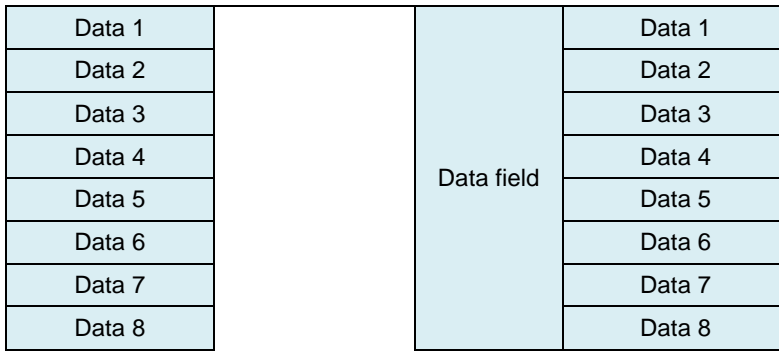
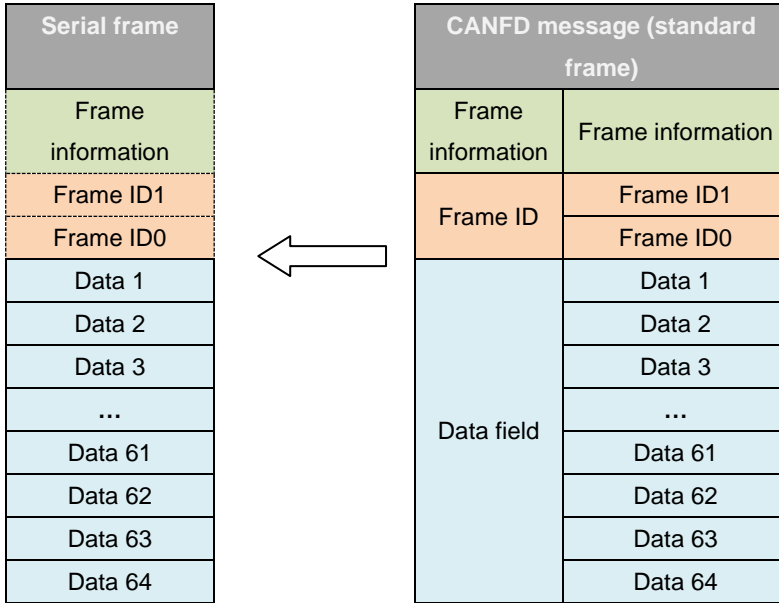


Table 4.5 Conversion of CANFD messages into serial frames (transparent conversion)

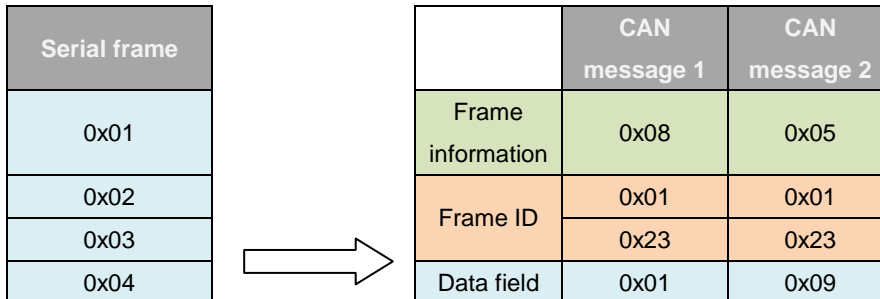


### 4.1.3 Conversion Examples

#### 1. Serial frames to CAN messages

Suppose the CAN type is ordinary CAN, the frame information is "standard frame", the frame ID1 and ID0 are "0x01, 0x23" respectively, the serial frame data is 0x01-0x0C, the conversion format is listed in Table 4.6. The frame ID of the CAN message is 0x0123 (user configuration), the frame information is a standard frame (user configuration), and the data part in the serial frame will be converted into the CAN message without any modification.

Table 4.6 Example of converting serial frames to CAN messages (transparent mode)



0x05		0x02	0x0A
0x06		0x03	0x0B
0x07		0x04	0x0C
0x08		0x05	
0x09		0x06	
0x0A		0x07	
0x0B		0x08	
0x0C			

## 2. Converting serial frames to CANFD messages

Suppose the configuration CAN type is CANFD, the frame information is "standard frame", the baud rate acceleration is enabled, the frame ID1 and ID0 are "0x01, 0x23" respectively, the serial frame data is 0x01-0x46, a total of 70 data, the conversion format is listed in Table 4.7. The frame ID of the CAN message is 0x0123 (configured by the user), the frame information is a standard frame (configured by the user), the EDL bit corresponding to the CANFD message is 1, the BRS bit corresponding to baud rate acceleration is 1, and the data in the serial frame part will be converted into CANFD message without any modification. Because there are 70 bytes of data and the maximum data size of one CANFD message is 64 bytes, it is divided into two CANFD messages.

Table 4.7 Example of converting serial frames to CANFD messages (transparent mode)

Serial frame		CANFD message 1	CANFD message 2
0x01	Frame information	0x3F	0x36
0x02		0x01	0x01
0x03	Frame ID	0x23	0x23
0x04		0x01	0x41
0x05	Data field	0x02	0x42
0x06		0x03	0x43
0x07		0x04	0x44
...		0x05	0x45
0x42		0x06	0x46
0x43		...	
0x44		0x3E	
0x45		0x3F	
0x46		0x40	

## 3. Converting CAN messages to serial frames

This example configures the CAN message "frame information" conversion and "frame ID" conversion. Assuming that the frame ID1 in the CAN message is 0x01, the frame ID0 is 0x23, and the data is 0x12, 0x34, 0x56, 0x78, 0xab, 0xcd, 0xef, 0xff, the CAN message and the converted serial frame are listed in Table 4.8.



Table 4.8 Example of converting CAN messages into serial frames (transparent conversion)

Serial frame	CAN message (standard frame)	
0x08	Frame information	0x08
0x01	Frame ID	0x01
0x23		0x23
0x12	Data field	0x12
0x34		0x34
0x56		0x56
0x78		0x78
0xAB		0xAB
0xCD		0xCD
0xEF		0xEF
0xFF		0xFF

As listed in Table 4.8, the frame information (0x08) of the CAN message is converted to the first byte (0x08) in the serial frame, and the ID 0x0123 in the CAN message is converted to the second byte (0x01) and the third byte (0x23) in the serial frame. The data field of the CAN message will be converted into the data part of the serial frame without any modification.

#### 4. Converting CANFD messages to serial frames

The CAN type configured in this example is CANFD, and the conversion of "frame information" and "frame ID" of the message is enabled. Assuming that the frame ID1 in the CANFD message is 0x01, the frame ID0 is 0x23, and the data is 0x01-0x40, a total of 64 data. Table 4.9 lists the CANFD message and the converted serial frame .

Table 4.9 Example of converting CANFD messages into serial frames (transparent conversion)

Serial frame	CANFD message (standard frame)	
0x3F	Frame information	0x3F
0x01	Frame ID	0x01
0x23		0x23
0x01	Data field	0x01
0x02		0x02
0x03		0x03
0x04		0x04
...		...
0x3E		0x3E
0x3F		0x3F
0x40		0x40

As listed in Table 4.9, the frame information (0x38) of the CANFD message is

converted to the first byte (0x38) in the serial frame, and the ID: 0x0123 in the CANFD message is converted to the second byte (0x01) and the third byte (0x23) in the serial frame in turn.. The data field of the CANFD message will be converted into the data part of the serial frame without any modification.

## 4.2 Transparent Conversion with Flags

Transparent conversion with flags is a special transparent conversion. The ID information of the CAN (CANFD) message is carried in the serial frame, and CAN (CANFD) messages with different IDs can be sent as required. It helps users set up their own network conveniently by using the converter, and use the self-defined application protocol. This method automatically converts the ID information in the serial frame into the frame ID of the CAN (CANFD) bus. Just tell the converter in the configuration that the ID information is at the start and length of the serial frame. The converter extracts the frame ID and fills it in the frame ID field of the CAN (CANFD) message during conversion as the ID of the CAN (CANFD) message when the serial frame is forwarded. When the CAN (CANFD) message is converted into a serial frame, the ID of the CAN (CANFD) message is also converted into the corresponding position of the serial frame.

Note: In this conversion mode, the "CAN ID" of the "CAN parameter" item of the configuration software is invalid, because the identifier (frame ID) sent at this time is filled with the ID information data in the above-mentioned serial frame.

### 4.2.1 Frame Formats

#### 1. Serial bus frame

In conversion with flags, the complete serial data frame must be obtained. The converter uses the time interval between two frames as the frame division. In addition, the interval can be set by the user. Figure 4.1 shows the frame format. The maximum length of the serial frame is the length of the buffer: 5,000 bytes.

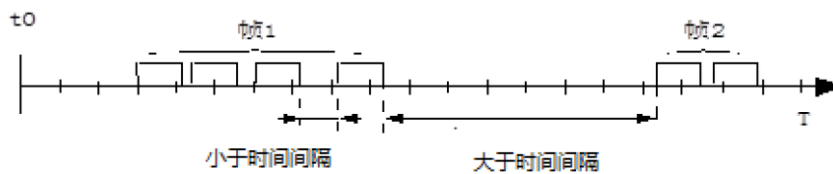


Figure 4.1 Serial frame time format (transparent conversion with flags)

The first data detected by the converter while the serial bus is idle is taken as the first character of the received frame. During transmission, the time interval between characters in the same frame must be less than the time interval between serial frames set by the user.

After the converter receives a character, if no character is received within the time interval between serial frames set by the user, the converter considers the frame transmission to be over and takes this character as the last character of the frame. The characters after the timeout do not belong to this frame, but to the next frame.

## 2. CAN (CANFD) bus frame

The format of the CAN (CANFD) message remains unchanged, and the corresponding frame ID in the CAN (CANFD) message will also be converted into the serial frame, but the frame information will not be converted.

### 4.2.2 Conversion Methods

#### 1. Serial frame to CAN (CANFD) message

The **starting address** and **length** of the "frame ID" of the CAN (CANFD) message in the serial frame can be set. The range of the start address is 0 to 7, and the range of the length is 1 to 2 (standard frame) or 1 to 4 (extended frame).

During conversion, the CAN (CANFD) message "frame ID" in the serial frame is correspondingly converted into the frame ID field of the CAN (CANFD) message according to the prior configuration. (If the number of frame IDs carried is less than the number of frame IDs in the CAN (CANFD) message, the high byte of the frame ID in the CAN (CANFD) message is filled with 0.) Other data are converted sequentially. If a frame of CAN (CANFD) message has not finished converting the serial frame data, the same ID is still used as the frame ID of the CAN (CANFD) message to continue converting until the serial frame conversion is completed.

Table 4.10 lists serial frames converted to CAN, and Table 4.11 lists serial frames converted to CANFD. **Note: The frame ID range of the standard frame is 0x000-0x7ff, which respectively indicate frame ID1 and frame ID0. The frame ID1 is the high byte. The frame ID range of the extended frame is: 0x00000000-0x1fffffff, which respectively indicate frame ID3, frame ID2, frame ID1, and frame ID0, of which frame ID3 is the high byte.**

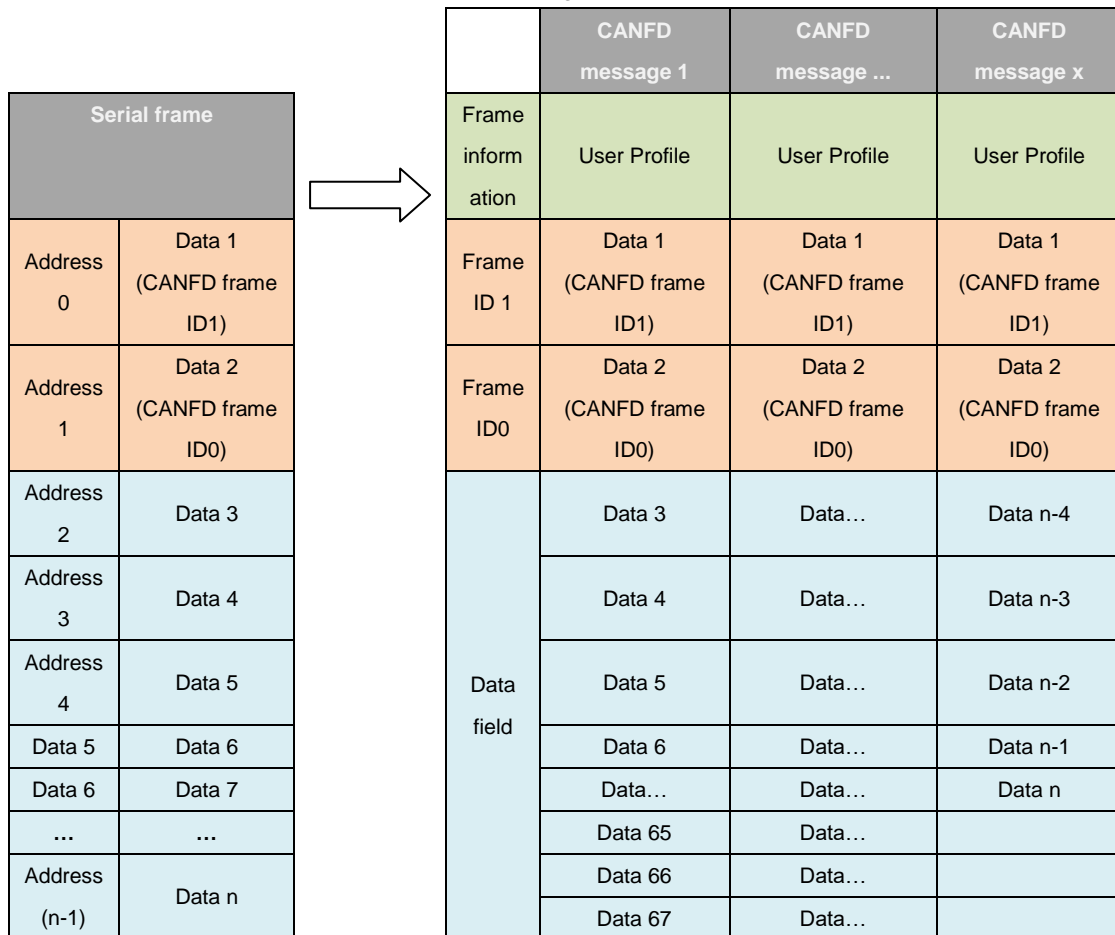
Table 4.10 Serial frames converted to CAN standard frame messages (transparent with flags)

Serial frame			CAN message 1	CAN message...	CAN message x
		Frame information	User Profile	User Profile	User Profile
Address 0	Data 1 (CAN frame ID1)	Frame ID 1	Data 1 (CAN frame ID1)	Data 1 (CAN frame ID1)	Data 1 (CAN frame ID1)
Address 1	Data 2 (CAN frame ID0)	Frame ID0	Data 2 (CAN frame ID0)	Data 2 (CAN frame ID0)	Data 2 (CAN frame ID0)
Address 2	Data 3	Data field	Data 3	Data...	Data n-4
Address 3	Data 4		Data 4	Data...	Data n-3

# CANFDNET-400U

Address 4	Data 5		Data 5	Data...	Data n-2
Data 5	Data 6		Data 6	Data...	Data n-1
Data 6	Data 7		Data 7	Data...	Data n
...	...		Data 8	Data...	
Address (n-1)	Data n		Data 9	Data...	
			Data 10	Data...	

Table 4.11 Converting serial frames into CANFD standard frame messages (transparent conversion with flags)



## 2. Converting CAN (CANFD) messages to serial frames

For CAN messages, a frame is forwarded immediately upon receipt. In each forwarding, the ID in the received CAN message is converted based on the position and length of the pre-configured CAN frame ID in the serial frame. Other data are forwarded

# CANFDNET-400U

sequentially. Assuming that the preset CAN type is ordinary CAN, the message is a standard frame, the "frame ID" start address is 0, and the length is 2, the conversion method is listed in Table 4.12. Assuming that the preset CAN type is CANFD, the message is a standard frame, the "frame ID" start address is 0, and the length is 2, the conversion method is shown in Table 4.13. Note: Whether it is a serial frame or a CAN message, its frame format (standard frame or extended frame) should meet the pre-configured frame format requirements; otherwise, the communication may fail.

Table 4.12 Converting CAN standard frame messages into serial frames (transparent conversion with flags)

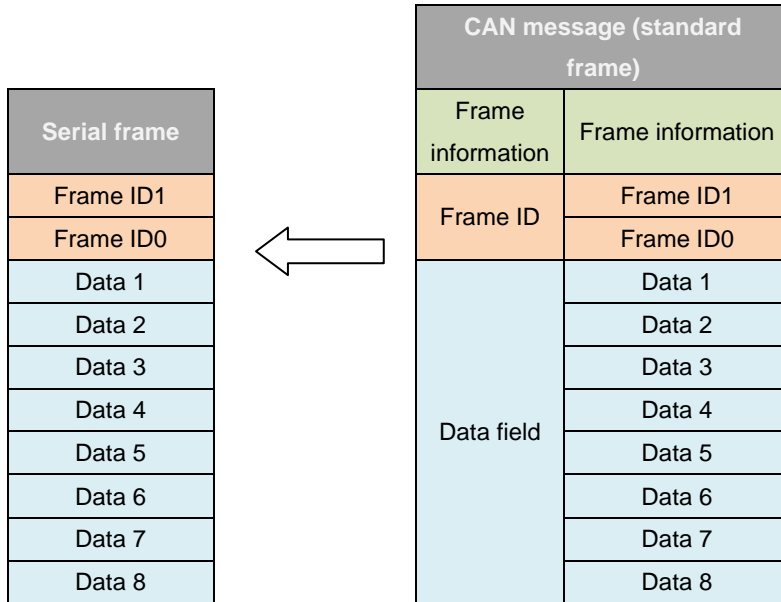
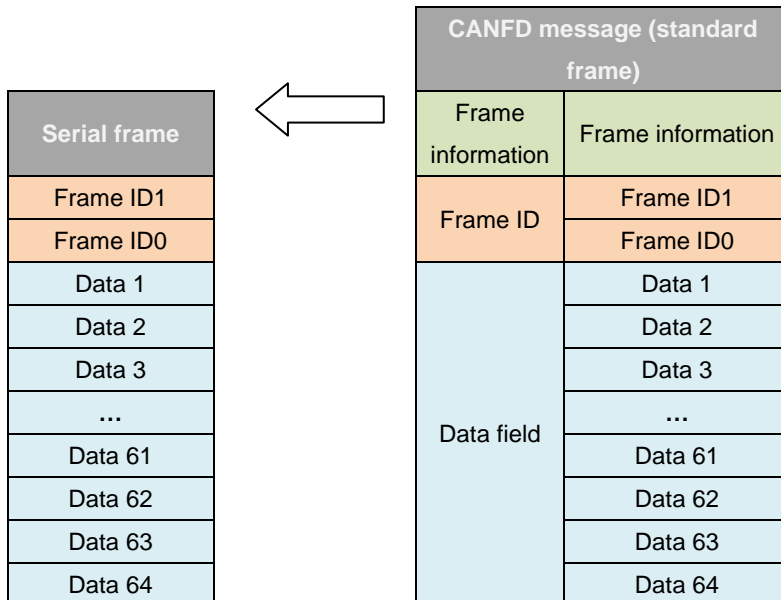


Table 4.13 Converting CANFD standard frame messages into serial frames (transparent conversion with flags)



## 4.2.3 Conversion Examples

### 1. Serial frames to CAN messages

Assuming that the starting address of the CAN logo in the serial frame is 2, the length is 3 (in the case of an extended frame), the data sent by the serial frame is 0x00-0x0E, the results of the serial frame conversion into CAN messages are shown in Table 4.14. Among them, two frames of CAN messages are converted with the same ID.

Table 4.14 Example of converting serial frames into CAN messages (transparent conversion with flags)

Serial frame			CAN message 1	CAN message 2	
Address 0	0x00	➔	Frame information	0x88	0x84
Address 1	0x01		Frame ID 3	0x00	0x00
Address 2	0x02		Frame ID 2	0x02	0x02
Address 3	0x03		Frame ID1	0x03	0x03
Address 4	0x04		Frame ID0	0x04	0x04
Data 5	0x05		Data field	0x00	0x0B
Data 6	0x06			0x01	0x0C
Address 7	0x07			0x05	0x0D
Address 8	0x08			0x06	0x0E
Address 9	0x09			0x07	
Address 10	0x0A			0x08	
Address 11	0x0B			0x09	
Address 12	0x0C		0x0A		
Address 13	0x0D				
Address 14	0x0E				

As shown in Table 4.14, because the pre-configured CAN message is an extended frame, the frame ID in the serial frame-to-CAN message is 4 bytes. Because the starting address in the configuration serial frame is 2 and the length is 3, three bytes are taken continuously from the address 2 in the serial frame, which are 0x02, 0x03, 0x04 in this example. During conversion, these three bytes are filled into frame ID2, frame ID1, frame ID0 of CAN message in turn, but frame ID3 is filled with 0x00. The rest of the data in the serial frame will be converted into the data field of the CAN message without any modification.

### 2. Converting serial frames to CANFD messages

Assuming that the starting address of the CANFD flag in the serial frame is 2 and the length is 3 (in the case of extended frame), the data sent by the serial frame are 0x00-0x45 respectively, a total of 70 data. Table 4.15 lists the serial frames converted to CAN messages. Among them, two frames of CAN messages are converted with the same ID.

Table 4.15 Example of converting serial frames to CANFD messages (transparent conversion with flags)

Serial frame			CANFD message 1	CANFD message 2		
Address 0	0x00	→	Frame information	0xBF	0xB3	
Address 1	0x01		Frame ID 3	0x00	0x00	
Address 2	0x02		Frame ID 2	0x02	0x02	
Address 3	0x03		Frame ID 1	0x03	0x03	
Address 4	0x04		Frame ID 0	0x04	0x04	
Data 5	0x05		Data field		0x00	0x43
Data 6	0x06				0x01	0x44
Address 7	0x07				0x05	0x45
Address...	...				0x06	
Address 64	0x40				...	
Address 65	0x41				0x40	
Address 66	0x42				0x41	
Address 67	0x43				0x42	
Address 68	0x44					
Address 69	0x45					

As shown in Table 4.15, because the CAN type is configured as CANFD and the frame format is extended frame, the frame ID in the serial frame converted to CANFD messages is 4 bytes. Because the starting address in the configuration serial frame is 2 and the length is 3, three bytes are taken continuously from the address 2 in the serial frame, which are 0x02, 0x03, 0x04 in this example. During conversion, these three bytes are filled into frame ID2, frame ID1, frame ID0 of CAN message in turn, but frame ID3 is filled with 0x00. The rest of the data in the serial frame will be converted into the data field of the CANFD message without any modification.

### 3. Converting CAN messages to serial frames

Assume that the start address of the configured CAN identifier in the serial frame is 2, the length is 3 (in the case of extended frame), the frame ID of the CAN message is 0x00123456, and the data is 0x00-0x07. Table 4.16 lists the CAN messages and the results of conversion to serial frames.

Table 4.16 Converting CAN extended frame messages into serial frames (transparent conversion with flags)

Serial frame			CAN message (extended frame)	
Address 0	0x00	←	Frame information	0x88
Address 1	0x01		Frame ID	0x00
				0x12
				0x34

# CANFDNET-400U

Address 2	0x12	Data field	0x56
Address 3	0x34		0x00
Address 4	0x56		0x01
Data 5	0x02		0x02
Data 6	0x03		0x03
Address 7	0x04		0x04
Address 8	0x05		0x05
Address 9	0x06		0x06
Address 10	0x07		0x07

As shown in Table 4.16, in this mode, it is obvious that the frame information of CAN messages will not be converted into serial frames. Since the start address in the configured serial frame is 2, the length is 3 (in the case of extended frames). Therefore, during conversion, the frame ID2, frame ID1, and frame ID0 in the CAN message are sequentially converted to the serial frame as the contents of address 2, address 3, and address 4. The data field of the CAN message will be sequentially converted to the unfilled part of the serial frame without any modification.

#### 4. Converting CANFD messages to serial frames

Assume that the start address of the configured CANFD identifier in the serial frame is 2, the length is 3 (in the case of extended frame), the frame ID of the CANFD message is 0x00123456, and the data is 0x00-0x3F. Table 4.17 lists the CANFD messages and the results of conversion into serial frames.

Table 4.17 Converting CANFD extended frame messages into serial frames (transparent conversion with flags)

Serial frame		←	CANFD message (extended frame)	
Address 0	0x00		Frame information	0xBF
Address 1	0x01		Frame ID	0x00
Address 2	0x12			0x12
Address 3	0x34			0x34
Address 4	0x56			0x56
Data 5	0x02		Data field	0x00
Data 6	0x03			0x01
Address...	...			0x02
Address 64	0x3D			0x03
Address 65	0x3E	...		
Address 66	0x3F	0x3D		
		0x3E		
		0x3F		

As shown in Table 4.17, in this mode, it is obvious that the frame information of CANFD messages will not be converted into serial frames. Since the start address in the



configured serial frame is 2, the length is 3 (in the case of extended frames). Therefore, during conversion, the frame ID2, frame ID1, and frame ID0 in the CANFD message are sequentially converted to the serial frame as the content in address 2, address 3, and address 4. The data field of the CAN message will be sequentially converted to the unfilled part of the serial frame without any modification.

### 4.3 Format Conversion

According to the preset CAN type, the format conversion is divided into CAN format conversion and CANFD format conversion. The difference is that the fixed number of bytes in the corresponding format is different. CAN format conversion with fixed 13 bytes indicates a CAN frame, while CANFD format conversion with fixed 69 bytes indicates a CANFD frame. As shown in Table 4.18, each CAN frame contains 13 bytes. The content of 13 bytes includes CAN frame information + frame ID + frame data. As shown in Table 4.19, each CANFD frame contains 69 bytes. The 69-byte content includes CANFD frame information + frame ID + frame data.

Note: In this conversion mode, the preset CAN ID is invalid, because the identifier (frame ID) sent at this time is filled with the frame ID data in the above-mentioned format serial frame. The configured frame type is also invalid, and the frame type is determined by the frame information in the format serial frame.

Table 4.18 CAN fixed format serial frame

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

Table 4.19 CANFD fixed format serial frame

CANFD fixed format serial frame (69 bytes)		
Frame information	Frame ID	Frame data
1Byte	4Byte	64Byte

The frame information is shown in Table 4.20, with a length of 1 byte. It is used to identify some information of the CAN (CANFD) message, such as type and length. Standard frames, extended frames and even CAN remote frames can be flexibly sent by correctly configuring the frame information (the first byte of data). Details of standard frames, extended frames and even remote frames can be obtained by correctly parsing fixed-byte format serial frames.

Table 4.20 Frame information

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

- FF: identifier of standard frame and extended frame. 0 indicates a standard frame, while 1 indicates an extended frame;
- RTR: identifier of remote frame and data frame. 0 indicates a data frame, while 1

indicates a remote frame. It can only be 0 for CANFD;

- EDL: CAN and CANFD identifier. 0 indicates the CAN, while 1 indicates the CANFD;
- BRS: baud rate switching enable flag. 0 indicates no rate conversion, while 1 indicates variable rate conversion, only valid for CANFD; this bit should be 0 in CAN;
- DLC3-DLC0: identifies the data length of the CAN (CANFD) message.

The length of the **frame ID** is 4 bytes. The valid bits of the standard frame are 11 bits, and the valid bits of the extended frame are 29 bits.

Table 4.21 Frame ID example

Standard frame ID 0x12345678				Standard frame ID 0x3FF			
0x12	0x34	0x56	0x78	0x00	0x00	0x03	0xFF

When the **frame data** is converted in CAN format, the length is fixed to 8 bytes. In CANFD format conversion, the length is fixed to 64 bytes. The effective length is determined by the value of DLC3-DLC0. When the valid data is less than the fixed length, add 0 to the fixed length. As listed in Table 4.22, it is the representation of 6-byte valid data during CAN format conversion. The last two bytes are filled with 0s.

In this mode, pay attention to the serial data format in strict accordance with the fixed byte format. Table 4.23 and Table 4.24 shows the reference examples for CAN mode conversion. It is similar in CANFD mode, except that the frame data is fixed 64 bytes. In the conversion, first ensure correct frame information. The data length must be correct; otherwise, no conversion will occur.

Each frame of CAN format conversion is fixed to 13 bytes, while each frame of CANFD format conversion is fixed to 69 bytes. If it is insufficient, it must be filled with 0. The serial data satisfying the fixed byte format in the same serial data frame corresponds to a CAN (CANFD) message. Serial data frames that are less than fixed bytes will not be converted. Therefore, make sure that the serial data frame being converted is aligned with the corresponding fixed byte. If the number of fixed bytes is insufficient, a timeout of about 100 ms will occur. If fixed bytes of data have not been received after the timeout, the previously received data will be cleared.

In the conversion of serial frames to CAN (CANFD) messages, if the data format of a certain segment of fixed byte in the serial data frame aligned with fixed bytes (fixed 13 bytes for CAN and 69 bytes for CANFD) is not standard, fixed byte lengths will not be converted, and subsequent data will be converted. If some CAN messages are missing after conversion, check whether the fixed-byte-length serial data format of the corresponding message is not the standard format.

Table 4.22 Data frame

CAN format conversion 6-byte valid data representation							
0x11	0x22	0x33	0x44	0x55	0x66	0x00	0x00

Table 4.23 Example of converting CAN formats to extension frames

The following example is a CAN extended data frame. The ID is 0x12345678, which

contains the frame representation of 8 bytes of valid data (0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88)												
0x8	0x1	0x3	0x5	0x7	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8
8	2	4	6	8	1	2	3	4	5	6	7	8

Table 4.24 Example of CAN format conversion standard frame

The following example is a CAN standard data frame. The ID is 0x3FF, frame representation containing 6 bytes of valid data (0x11, 0x22, 0x33, 0x44, 0x55, 0x66)												
0x0	0x0	0x0	0x0	0xF	0x1	0x2	0x3	0x4	0x5	0x6	0x0	0x0
6	0	0	3	F	1	2	3	4	5	6	0	0

## 4.4 Modbus Conversion

The Modbus protocol is a standard application layer protocol, which is widely used in various industrial control scenarios. The protocol is open, real-time, and has a good communication verification mechanism. It is ideal for scenarios with high communication reliability requirements.

The converter uses the standard Modbus RTU protocol format on the serial port side. Therefore, the converter not only allows you to use the Modbus RTU protocol, but the converter can also directly interface with other devices that support the Modbus RTU protocol.

On the CAN (CANFD) side, an easy-to-use segmented communication format is developed to realize Modbus communication. The role of the converter is still protocol verification and forwarding. It supports Modbus protocol transmission, not Modbus master or slave host. The user can communicate according to the Modbus protocol.

Note: In this conversion mode, the preset CAN (CANFD) ID is invalid, because the identifier (frame ID) sent at this time is filled with the address field in the Modbus RTU serial frame.

### 4.4.1 Frame Formats

#### 1. Serial bus frame

The serial interface adopts the standard Modbus RTU protocol, so the user frame only needs to conform to this protocol. If the transmitted frame does not meet the Modbus RTU format, the converter discards the received frame without converting it.

The Modbus RTU transmission format adopted by the converter is 1 start bit, 8 data bits, and 1 stop bit. The maximum length of the Modbus RTU frame is the buffer length: 5,000 bytes.

#### 2. CAN (CANFD) bus frame

If the device on the CAN (CANFD) side adopts the Modbus protocol, a reliable transmission format needs to be defined for it. A segmentation protocol is used here, which defines a method for segmenting and reorganizing information whose length is greater than the maximum data length of a CAN (CANFD) message.

The formulation of the segmented transmission protocol refers to the transmission protocol of segmented messages in DeviceNet. For CAN messages, the segmented

# CANFDNET-400U

message format is shown in Table 4.25 (using the CAN extended frame as an example, the standard frame only has a different length of the frame ID, and other formats are the same). The transmitted Modbus protocol content can then start with the "Data 2" byte. If the protocol content is greater than 7 bytes, the remaining protocol content will continue to be converted in this segmented format until the conversion is completed.

Table 4.25 CAN extended frame segmentation message format

Bit number	7	6	5	4	3	2	1	0
Frame information	FF	RTR	0x00	0x00	DLC (data length)			
Frame ID3	X	X	X	ID.28 - ID.24				
Frame ID 2	ID.23 - ID.16							
Frame ID1	ID.15 - ID.8							
Frame ID0	ID.7 - ID.0							
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							

For CANFD messages, the segmented message format is shown in Table 4.25 (using the CANFD extended frame as an example, the standard frame only has a different length of the frame ID, and other formats are the same). The transmitted Modbus protocol content can then start with the "Data 2" byte. If the protocol content is greater than 63 bytes, the remaining protocol content will continue to be converted in this segmented format until the conversion is completed.

Table 4.26 CANFD extended frame segmentation message format

Bit number	7	6	5	4	3	2	1	0
Frame information	FF	RTR	EDL	BRS	DLC (data length)			
Frame ID3	X	X	X	ID.28 - ID.24				
Frame ID 2	ID.23 - ID.16							
Frame ID1	ID.15 - ID.8							

Frame ID0	ID.7 - ID.0		
Data 1	Segment marker	Segment type	Segment counter
Data 2	Character 1		
Data 3	Character 2		
Data 4	Character 3		
Data...	Character...		
Data 62	Character 61		
Data 63	Character 62		
Data 64	Character 63		

- **Fragmented packet flag:** indicates whether the packet is a fragmented packet. A 0 for this bit indicates a separate message, while a 1 indicates a frame within the segmented message.
- **Segment type:** indicates the first, middle or last segment. Table 4.27 lists the value definitions.

Table 4.27 Segment type bit values

Bit value	Meaning	Description
0	First segment	If the segment counter contains the value 0, this is the first segment in a series of segments.
1	Middle segment	This is an intermediate segment
2	Last segment	Mark the last segment

- **Segment counter:** The flag of each segment indicates the sequence number of the segment in the entire message. The value of the counter is the segment number. This will allow you to verify whether any segments are missing when receiving. A total of 5 bits are used, and the range is 0-31.

#### 4.4.2 Conversion Methods

During the conversion from the serial port side to the CAN (CNAFD) side, the converter will only perform the conversion after receiving a complete and correct Modbus RTU.

Table 4.28 shows the CAN mode conversion, and Table 4.29 shows the CANFD mode conversion. The address field of the Modbus RTU protocol is converted into ID0 of the frame ID in the CAN message. The flag does not change during the frame conversion.

Table 4.28 CAN communication frame mutual conversion format (Modbus mode)

Modbus RTU frame		CAN message	CAN message 1	CAN message x
Address field	↔	Frame information	Frame information	Frame information
Function Code		Frame ID3	0x00	0x00
Data field		Frame ID 2	0x00	0x00
		Frame ID1	0x00	0x00
		Frame ID0	Address field	Address field
		Data 1	[Segmentation protocol usage]	[Segmentation protocol usage]
		Data 2	Function Code	Data field
		Data 3	Data field	
		Data 4		
	Data 5			
	Data 6			
	Data 7			
CRC field		Data 8		

Table 4.29 CANFD communication frame mutual conversion format (Modbus mode)

Modbus RTU frame		CANFD message	CANFD message 1	CANFD message x
Address field	↔	Frame information	Frame information	Frame information
Function Code		Frame ID3	0x00	0x00
Data field		Frame ID 2	0x00	0x00
		Frame ID1	0x00	0x00
		Frame ID0	Address field	Address field
		Data 1	[Segmentation protocol usage]	[Segmentation protocol usage]
		Data 2	Function Code	Data field
		Data 3	Data field	
		Data 4		
	Data...			
	Data 62			
	Data 63			
CRC field		Data 64		

The CRC check byte of the Modbus RTU frame is not converted into the CAN (CANFD) message, and the CAN (CANFD) message does not need to have the check byte of the serial frame, because the CAN (CANFD) bus itself has an excellent verification mechanism.

The converted data is the function code and data field of the protocol content of

# CANFDNET-400U

Modbus RTU. In conversion, convert them in turn in the data field of the CANFD message frame (starting from the second data byte, the first data byte is used by the segmentation protocol). The length of the Modbus RTU frame varies according to the function code. Because the first byte is used for segment identification, only seven pieces of data can be transmitted in one frame of CAN message. A CANFD message can only transmit 63 bytes of data in one frame. The converter will segment the longer Modbus RTU frame into CAN(CANFD) messages and send them using the above-mentioned CAN(CANFD) segment protocol. When you receive them on the CAN (CANFD) node, you can reassemble the data according to the segmentation protocol, extract the function code, and process the data field.

For the Modbus protocol data of CAN (CANFD) bus, there is no need to perform cyclic redundancy check (CRC16), and the converter receives data by using the segmented protocol. After receiving a frame, it will automatically add a cyclic redundancy check (CRC16), convert data into a Modbus RTU frame and send it to the serial bus. If the received data does not conform to the segmentation protocol, the group of data is discarded without conversion.

### 4.4.3 Conversion Examples

When the CAN mode is configured and the frame type is set to extended frame, as shown in Table 4.30, if the Modbus RTU frame is converted into a CAN message, the address field 0x08 is directly filled into the frame ID0, and other frame IDs are filled with 0x00; that is, the CAN ID is 0x08. Keep this frame ID unchanged during the conversion of this Modbus frame.

If the data in the function code + data field is less than 7 bytes, a single message can be sent, as shown in Table 4.31. When one frame of CAN message cannot process one frame of Modbus message, the CAN message adopts segmentation protocol.

The "data 1" of each CAN message is used to fill the segment information (0x81, 0xC2), which is not converted into the ModbusRTU frame, and is only used as the frame format to confirm the frame information. The value of the function code and data field is filled in the data 2-8 of the CAN message in turn.

The CANFD mode is similar to CAN, except that the segmented protocol is used to divide the message when the data exceeds 63 bytes.

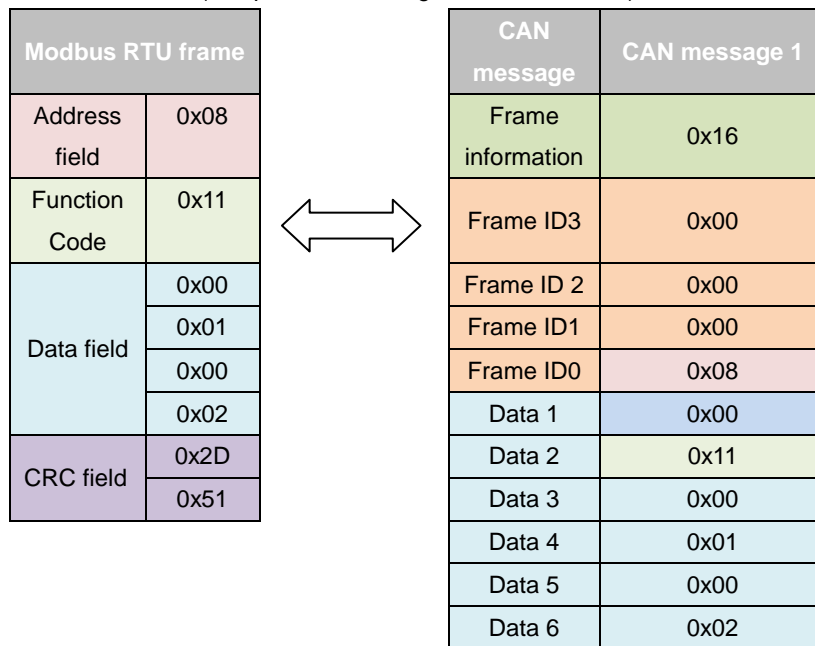
Table 4.30 Example of the mutual conversion format for the CAN extended frame communication frame (Modbus mode segmented message)

Modbus RTU frame		CAN message	CAN message 1	CAN message 2
Address field	0x08	Frame information	0x88	0x84
Function Code	0x11	Frame ID3	0x00	0x00
Data field	0x00	Frame ID 2	0x00	0x00
	0x01	Frame ID1	0x00	0x00
	0x00	Frame ID0	0x08	0x08

# CANFDNET-400U

	0x02	Data 1	0x81	0xC2
	0x04	Data 2	0x11	0x0A
	0x00	Data 3	0x00	0x01
	0x0A	Data 4	0x01	0x02
	0x01	Data 5	0x00	
	0x02	Data 6	0x02	
CRC field	0xED	Data 7	0x04	
	0x69	Data 8	0x00	

Table 4.31 Example of mutual conversion format of the CAN extended frame communication frame  
(independent message in Modbus mode)





## 5. Quick Instructions

### 5.1 Configuring the Converter

#### 5.1.1 Entering Configuration Mode

After the device is powered on, set the switch to the "CFG" position, as shown in Figure 5.1. The system enters the configuration state, and the CFG indicator is on. The converter can be configured.



Figure 5.1 Entering the configuration mode

#### 5.1.2 Connecting the Converter by Using the Configuration Tool

In the configuration mode, connect the PC and the converter by using a serial cable.



Figure 5.2 Connecting a device

Run the "CANFDNET-100IE Configuration" host software, as shown in Figure 5.2. Select the PC serial port connected to the converter, "COM15" in the example. Click the "Connect Device" button. After a successful connection, the "Connect Device" button becomes a "Disconnect" button.

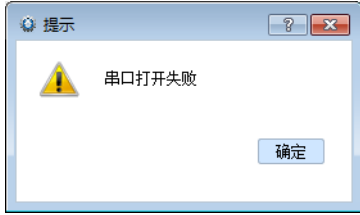


Figure 5.3 Serial port occupied



Figure 5.4 Device verification failed

If the "Failed to open serial port" message appears, as shown in Figure 5.3, the serial port of the currently selected PC is unavailable or already occupied. If the "Device connection verification failed" message appears, check whether the converter has entered configuration mode (if the converter is working in "normal operation" mode, the software will also display the "Device connection verification failed" message), and check whether the serial port on the selected PC is connected.

### 5.1.3 Setting the Save Parameters

After the connection is successful, configure the converter parameters. After selecting the required parameters, click the "Save Configuration" button to save the parameters to the converter, as shown in Figure 5.5. **After the configuration is complete, click the "Disconnect" button, or close the configuration tool. Otherwise, the configuration tool will occupy the serial port.**



Figure 5.5 Saving the configuration

## 5.2 Communication Test

After the configuration is complete, set the "Configuration Switch" to "ON"! In this way, you can start the normal working mode. In the normal working mode, the converter can start normal conversion.

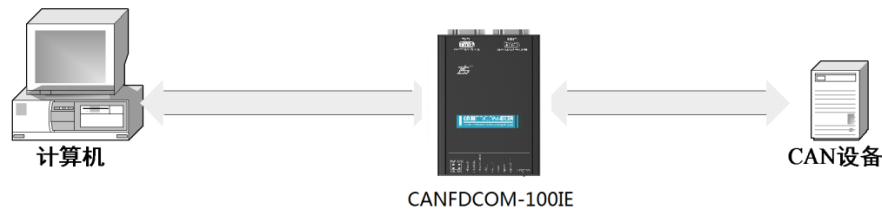


Figure 5.6 Communication test structure

As shown in Figure 5.6, the test requires a CAN (CANFD) device in addition to a PC to receive or transmit data. Note: In the same CAN-bus bus, the CAN baud rate of the CAN device and the CANFDCOM-100IE converter must be the same, and the baud rate of the serial port must be the same. Otherwise, the communication fails and the indicator flashes red.

Conduct data conversion tests. Use the serial port debugging software to send data by using the same serial port baud rate as the converter. Observe whether the data received by the CAN device is consistent with the data sent. You can also send data from the CAN device to the converter, and observe whether the data received by the serial port software is consistent with the sent data.

If there is data transmission on one side of the bus, and the transmission is normal, the indicator of the bus will flash green.

## 6. Firmware Upgrade

To improve the device maintainability, CANFDCOM-100IE provides the IAP firmware upgrade function. In the upgrade, use the RS232 port for firmware upgrade. There are two upgrade methods:

Method 1: Upgrade from the configuration mode of the application. The method is the most simple and convenient. The upgrade can be completed by simply pressing a few buttons on the host computer software, but you need to ensure that the application program can run properly; otherwise, only method 2 can be used.

First, connect the RS232 port, and set the configuration switch to "CFG" to enter the configuration mode. In configuration mode, the configuration indicator is always on. The upgrade can be completed by sending the firmware bin file by using the host computer configuration tool. The procedure is as follows:

Step 1: Click the firmware upgrade button below the host computer configuration tool. The firmware upgrade window appears, as shown in Figure 6.1;

Step 2: Select the path of the firmware file to be upgraded;

Step 3: Click the upgrade button. The configuration indicator will flash rapidly, indicate that the firmware is being upgraded. At the same time, the COM indicator is solid on, indicating that the data is transmitted properly. After the progress bar reaches 100% and the upgrade completion confirmation box is displayed, the upgrade is completed. At this time, the configuration indicator changes from fast flashing to steady light, indicating that the upgrade is successful. If the actual situation is none of the preceding symptoms, the upgrade is abnormal. Check whether the communication interface is normal and whether the selected firmware bin file path is correct. If the upgrade exception damages the application, method 1, which is simple, can no longer be used, and only method 2 can be used.

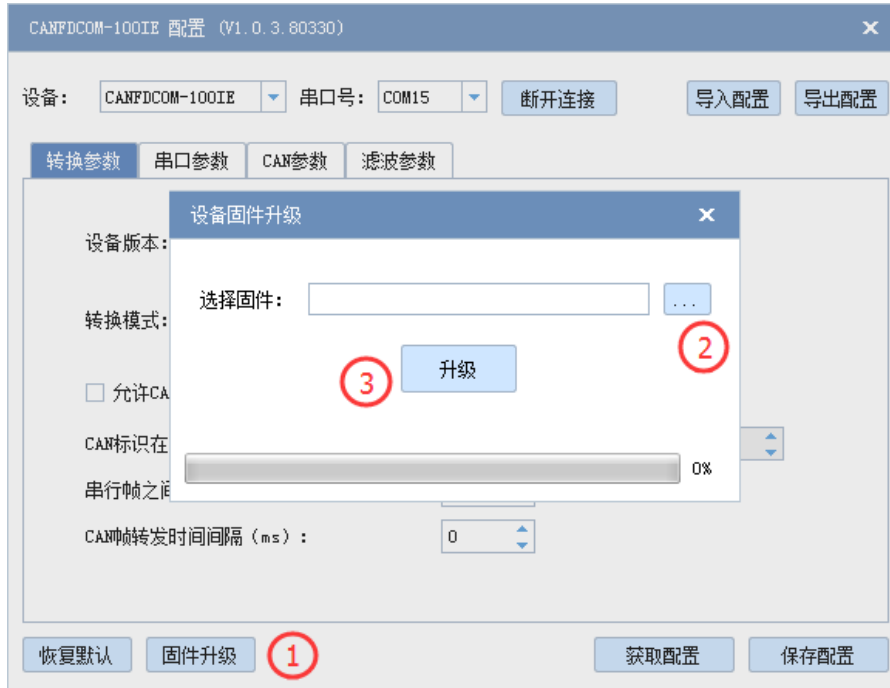


Figure 6.1 Firmware Upgrade window

Method 2: After entering the Bootloader configuration mode by operating the configuration switch, connect to the configuration software of the host computer, and perform upgrade using method 1. Figure 3.1 shows the configuration switch. In this way, as long as the Bootloader program is not damaged, the upgrade can be performed, but the operation is troublesome.

When the configuration switch is set to ON, that is, in the normal working mode, the device will not stay in the Bootloader mode after being reset or powered on again, and will immediately run the application. When the configuration switch is set to the CFG side, the device reset will stay in the Bootloader mode for 1 second. During this second, if the configuration indicator is flashing fast, the device stays in Bootloader mode. Turn the switch to ON within this second. At this time, the time spent in Bootloader mode will increase by 1 second. In this 1 second, turn the switch back to the CFG side to enter the Bootloader configuration mode. At this time, the configuration indicator will keep flashing slowly (on for 1 second, off for 1 second), indicating that it has been in the Bootloader configuration mode. Simply speaking, set the configuration switch to CFG, turn the switch to ON within 1 second after the device is powered on again, and then set the switch to CFG within 1 second.

In Bootloader mode, if you do not want to perform upgrade, set the configuration switch to ON to enter the normal working mode. The premise is that the application is not damaged. Therefore, the configuration switch cannot be toggled during the upgrade (in the process of downloading the firmware from the host computer configuration software). This action will terminate the upgrade. If the application has not been upgraded, it will not work properly. The only way to return to Bootloader mode and upgrade the firmware properly.

## 7. Disclaimer

Based on the principle of providing better service for users, Guangzhou ZLG Electronics Co., Ltd. ("ZLG Electronics") will try to present detailed and accurate product information in this manual. However, due to the effectiveness of this manual within a particular period of time, ZLG Electronics does not guarantee the applicability of this document at any time. ZLG Electronics shall reserve the right to update this manual without prior notice. To get the latest version, please visit the official website of ZLG Electronics regularly or contact ZLG Electronics. Thank you!

# CANFDNET-400U

High-performance Four-channel CANFD Bus-to-Ethernet Converter User Manual User Manual

---

Dreams come true with professionalism and dedication.

**Guangzhou ZLG  
Electronics Co., Ltd.**

For more details, please visit [www.zlg.cn](http://www.zlg.cn)  
Welcome to call the national service hotline **400-888-4005**

