| Category | Contents |
|---|---|
| Keywords | CANCOM-100IE+ CAN-bus RS-232 RS-485 RS-422 |
| Description | CAN-bus intelligent converter |

Revision History

| Version | Date | Description |
|---|---|---|
| V0.90 | June 22, 2017 | Created |
| V1.00 | August 17, 2017 | Completed the manual |
| V1.01 | March 23, 2018 | Modified and improved the description of the Modbus protocol |
| V1.02 | August 2, 2018 | Completed the Modbus example table |
| V1.03 | January 21, 2019. | Added Modbus conversion function description and usage conditions |
| V1.04 | March 12, 2019 | Updated the document header and footer and "Sales and Service Network" content and add the "Disclaimer" content |

# Contents

## 1. Functions

### 1.1 Overview

The CANCOM-100IE+ intelligent protocol converter can quickly connect RS-232/485/422 communication equipment to the CAN-bus field bus.

The converter supports 600-230,400 bps rate, 5Kbps-1Mbps CAN-bus communication rate.

The converter provides four data conversion modes: transparent conversion, transparent conversion with flags, format conversion, and Modbus conversion (the Modbus conversion function can be used only when the CAN bus device message can be modified).

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

### 1.2 Features

- Realizes bidirectional data communication between CAN-bus and RS-232/485/422;
- Supports CAN2.0A and CAN2.0B protocol, and comply with ISO/DIS 11898 specification;
- Integrates one CAN-bus communication interface, and supports user-defined communication baud rate;
- 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 600-230,400 bps;
- Provides four data conversion modes: transparent conversion, transparent conversion with flags, format conversion, and Modbus conversion;
- Supports CAN-bus virtual PC serial port application;
- 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;
- Existing RS-232/485 equipment connected to the CAN-bus network;
- Extended standard RS-232/485 network communication length;
- PLC equipment connected to the CAN-bus network for communication;
- Gateway bridge between CAN-bus and serial bus;
- Industrial field network data monitoring;

- Remote communication for CAN teaching application;
- CAN industrial automation control system;
- Low-speed CAN data acquisition data analysis;
- CAN-bus application systems, such as intelligent building control data broadcasting system.

## 2. Hardware

### 2.1   Product Appearance

Figure 2.1 shows the appearance.
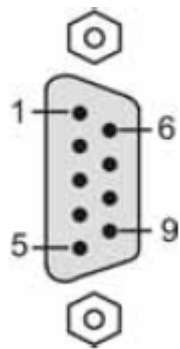


Figure 2.1 CANCOM-100IE+ appearance

### 2.2   Interfaces

The CANCOM-100IE+ converter has two user interfaces. One is a CAN-bus interface, and the other is an RS-232/485/422 interface. The interface pins are defined as follows.

#### 2.2.1  RS-232 Interface Pin Definitions

For CANCOM-100IE+, the RS-232 port is a standard DB9 socket, and the pin definition conforms to the RS-232 specification. A three-wire connection is used here, as shown in Figure 2.2.

| Pin No. | Pin Name | Pin Definition |
|---|---|---|
| 1 | -- | No connection |
| 2 | TXD | Data sender |
| 3 | RXD | Data receiver |
| 4 | -- | No connection |
| 5 | GND | Signal ground |
| 6 | -- | No connection |
| 7 | -- | No connection |
| 8 | -- | No connection |
| 9 | -- | No connection |

Figure 2.2 RS-232 interface pin definition

### 2.2.2 Interface Definitions

Figure 2.3 shows the interface definition. Pin 1 marked "PWR" is connected to the external +9V to +26V DC power supply (operating power is lower than 2 W), as shown in the table.

| Pin No. | Pin Name | Pin Definition |
|---|---|---|
| 1 | CAN_L | CANL signal cable connector |
| 2 | CAN_H | CANH signal cable connection |
| 3 | NC | Null |
| 4 | EARTH | Ground |
| 5 | RS485A | Differential positive level |
| 6 | RS485B | Differential positive and |
| 7 | RS422A | Differential positive level |
| 8 | RS422B | Differential negative level |
| 9 | PWR- | Negative power supply (with |
| 10 | PWR+ | Positive power supply (with |

Figure 2.3 CAN interface pin definition

Note: When the converter is used as a CAN-bus network terminal, a 120-ohm resistor (accessory) needs to be connected between the two pins.

## 2.3 Indicators

The three indicators on the converter are used to indicate the operating status of the CANCOM-100IE+ converter. Table 2.1 describes the indicators.

Table 2.1 Indicator definition

| Indicator | Colour | Function | Description |
|---|---|---|---|
| POWER | Red | Converter power indicator | If the indicator is on, the converter power supply is working properly |
| COM | Green | RS-232/485/422 communication status indication | If the indicator is flashing, the serial port side is transmitting data |
| CAN | Green | CAN communication status indication | If the indicator is flashing, the CAN side is transmitting data |

After normal power-on, the "POWER" indicator lights up immediately.

When the power-on self-test of the converter is completed, both the "COM" indicator and the "CAN" indicator are off.

When there is data transmission on the serial port side, the "COM" indicator flashes.

When there is data transmission on the CAN side, the "CAN" indicator flashes; when there is no data, it turns off .

If an error occurs on CAN bus communication, the "CAN" indicator is solid on.

## 2.4 CAN Bus Connection

When the CANCOM-100IE+ converter is connected to the CAN bus, CANL is connected to CANL, and CANH is connected to CANH.

According to the ISO 11898 specification, to enhance the reliability of CAN-bus communication, the two endpoints of the CAN-bus network usually need to add a terminal matching resistor (120 ohm), as shown in Figure 2.4. The terminal matching resistance is determined by the characteristic impedance of the transmission cable. For example, if the characteristic impedance of the twisted pair is 120 ohm, the two terminals on the bus should also integrate a 120-ohm terminal resistance.

The internal circuit of the CANCOM-100IE+ converter does not integrate a 120-ohm terminal resistor (the terminal resistor is delivered along with the product). When the CANCOM-100IE+ converter is used as a terminal device, a 120-ohm terminal resistance can be connected between the CAN interface pins of the CANCOM-100IE+ converter.
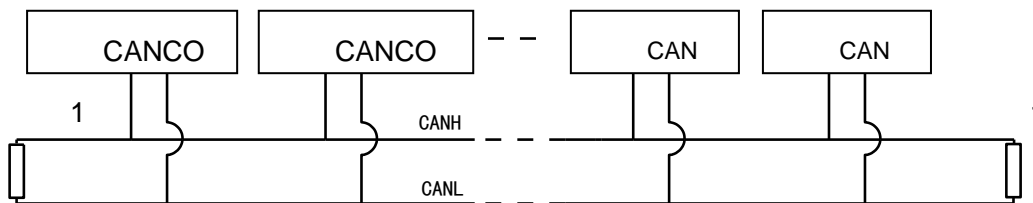


Figure 2.4 CAN bus connection

Note: The CAN communication cable can be a twisted pair and shielded twisted pair. If the communication distance exceeds 1 km, the cross-sectional area of the cable should be greater than 1.0

$mm^2$. The specific specifications should be determined based on the distance. Generally, it should be appropriately increased with the distance.

## 3. Configuration

Because there are many communication parameters for CAN-bus and RS-232/485/422 serial ports, the CANCOM-100IE+ converter also opens most of the parameters, allowing users to set them to meet the needs of practical applications. CANCOM-100IE+ converter configuration includes the converter conversion method, serial port parameters and CAN-bus parameters. Parameters are configured by using the dedicated configuration software, without hardware jumper configuration.

Before normal use, pre-configure the conversion parameters of the CANCOM-100IE+ converter; if the parameters are not configured, the CANCOM-100IE+ converter executes the parameters that were successfully configured last time (if no parameters have ever been configured, the converter executes the default configuration parameters ).

### 3.1 Entering the Configuration Mode

To put the converter into configuration mode, there is a dedicated configuration switch, as shown in the figure. Set the switch to the "CFG" position. The system will be in the configuration state. Run the host computer configuration software, select the serial port number, and click "Connect Device" to set parameters. After configuring the device, set the "configuration switch" to "ON"! In normal operation.



Figure 3.1 Configuration switch

### 3.2 Software Description

The configuration software name of CANCOM-100IE+ converter is "CANCOM-100IE+ V1.00.exe". The setting software is included on the CD-ROM delivered with the product. Figure 3.2 shows the 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 the configuration is completed in a certain conversion mode, 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.



Figure 3.2 Configuration interface

### 3.2.1 Conversion Parameters

Conversion parameters refer to parameters such as the conversion rule direction of the converter. Figure 3.1 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 message to the first byte of the serial frame when it is working. When this parameter is not selected, the frame information of CAN 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 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 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 messages, the offset address of the start byte of the frame ID of the CAN 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 ID2 of the CAN message respectively. When extending the frame, 1 to 4 bytes of ID1, ID2, ID3 and ID4 can be filled.

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

Time interval between serial frames:

This parameter is only used in 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 time interval is in units of "time to transmit a single character". The time set here is 2 to 10 characters.

[Note]: "Serial Frame Time Interval Characters" can be set only in "Transparent Conversion with Flags" mode. The actual time interval of the user frame must be consistent with the setting (the frame interval time sent by the user is preferably greater than the preset 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.

### 3.2.2 Serial Port Parameters

Baud rate: The baud rate of the serial port ranges between 600 bps and 230,400 bps.

The data is between 5 and 8 bits.

Stop bit: 1 or 2.

Parity check: no check, even check, odd check, forced to 0, forced to 1.

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

### 3.2.3 CAN Parameters

Baud rate: CAN bus baud rate. In addition to the standard baud rates recommended by CIA (75%-83.5% sampling point, SJW = 2, 3) in the list, a "custom option" is also given. If you require customization, contact us or refer to the baud rate calculation software.

Frame Type: frame type of the CAN message at the time of conversion. Standard frame or extended frame.

Filter: Multiple filter modes can be set (refer to 3.2.4).



Figure 3.3 CAN parameters

### 3.2.4 Example for the Setting of the Acceptance Filter

CANCOM-100IE+ has the capability of hardware acceptance filtering. The selective reception minimizes the network load from the network. When setting the acceptance filter, switch to the filter setting tab, and select the enable acceptance filter function before the "Enable filter" option. CANCOM-100IE+ configures a storage space of 2 KB for

acceptance filtering. The storage space occupied by various ID resources cannot exceed this value. Table 3.1 lists the resources and quantities occupied by various ID types.

Table 3.1 Resource occupied by various types of acceptance filter IDs

| ID type | Occupied resources (Byte) |
|---|---|
| Standard frame single ID filtering | 2 |
| Standard frame group ID filtering | 4 |
| Extended frame single ID filtering | 4 |
| Extended frame group ID filtering | 8 |

Example: Set the acceptance standard frame ID to 0x08, 0x12, and the extended frame group ID to 0x55 to 0x66. The configuration is as follows, as shown in Figure 3.4.



Figure 3.4 Filter parameter settings

### 3.2.5 Buttons

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.

Default: Restore its parameters to their default values.

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

Read current configuration: Read out the current parameters of the converter and display them on the current panel.

Read all configuration: Read out all the parameters of the converter and display them on the panel.

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

## 4. Conversion Example

CANCOM-100IE+ converter is an intelligent protocol converter. 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.

"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 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 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 format is 13 bytes; that is, the serial frame data of fixed 13 bytes corresponds to a CAN message. The 13-byte content includes CAN information + ID + data. Standard frames, extended frames and even 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 13-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 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.

## 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.

2. CAN bus frame

The format of CAN message frame remains unchanged.

### 4.1.2 Conversion Methods

1. Serial frames to CAN messages

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

The converted CAN 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. Figure 4.1 sows the corresponding format of data conversion.

If the line frame length of the received string is less than or equal to 8 bytes, 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 (as shown in Figure 4.1, n is 7).

If the number of bytes of the serial frame is greater than 8, the processor 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.
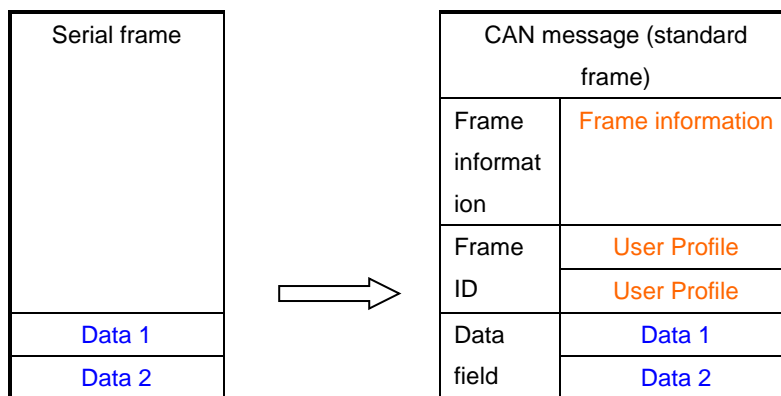
| Serial frame | | CAN message (standard frame) | |
|---|---|---|---|
| | | Frame information | Frame information |
| | | Frame ID | User Profile |
| | | | User Profile |
| Data 1 | | Data field | Data 1 |
| Data 2 | | | Data 2 |

| | | | |
|---|---|---|---|
| Data 3 | | Data 3 | |
| Data 4 | | Data 4 | |
| Data 5 | | Data 5 | |
| Data 6 | | Data 6 | |
| Data 7 | | Data 7 | |
| | | | |

Table 4.1 Converting serial frames to CAN messages (transparent mode)

2.  Converting CAN messages to serial frames

For CAN bus messages, one frame is immediately forwarded upon receiving one frame. Figure 4.2 shows the data format.

During the conversion, all the data in the data field of the CAN message are sequentially converted into serial frames. If "Convert" is selected for "frame information conversion enable" during configuration, the converter will directly fill the "frame information" byte of the CAN message into the serial frame.

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

| Serial frame | CAN message (standard frame) | |
|---|---|---|
| Frame information | Frame information | Frame information |
| Frame ID1 | Frame ID | Frame ID1 |
| Frame ID 2 | | Frame ID 2 |
| Data 1 | Data field | Data 1 |
| Data 2 | | Data 2 |
| Data 3 | | Data 3 |
| Data 4 | | Data 4 |
| Data 5 | | Data 5 |
| Data 6 | | Data 6 |
| Data 7 | | Data 7 |
| | | |

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

### 4.1.3 Conversion Examples

1.  Serial frames to CAN messages

Assuming that the configured frame information converted into CAN message is "standard frame", and frame ID1 and ID2 are "0x00, 0x60" respectively, the conversion format is shown in Figure 4.3.

| Serial frame | | | CAN message 1 | CAN message 2 |
|---|---|---|---|---|
| | | Frame information | 0x08 | 0x05 |
| | | Frame ID 1 | 0x00 | 0x00 |
| | | Frame ID 2 | 0x60 | 0x60 |
| 0x01 | | Data field | 0x01 | 0x09 |
| 0x02 | | | 0x02 | 0x0A |
| 0x03 | | | 0x03 | 0x0B |
| 0x04 | | | 0x04 | 0x0C |
| 0x05 | | | 0x05 | 0x0D |
| 0x06 | | | 0x06 | |
| 0x07 | | | 0x07 | |
| 0x08 | | | 0x08 | |
| 0x09 | | | | |
| 0x0A | | | | |
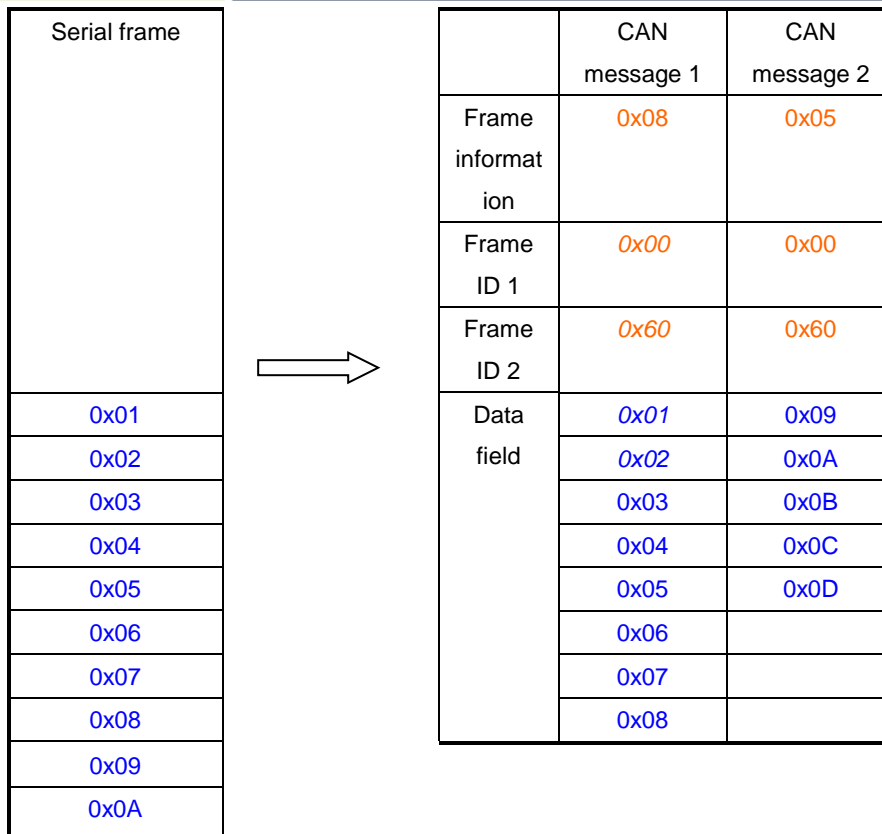
Continued

| |
|---|
| 0x0B |
| 0x0C |
| 0x0D |

Table 4.3 Example of converting serial frames to CAN messages (transparent mode)

2. Converting CAN messages to serial frames

It is configured as "frame information" conversion of CAN messages, and "frame ID" is not converted. The figure shows CAN messages and converted serial frames.
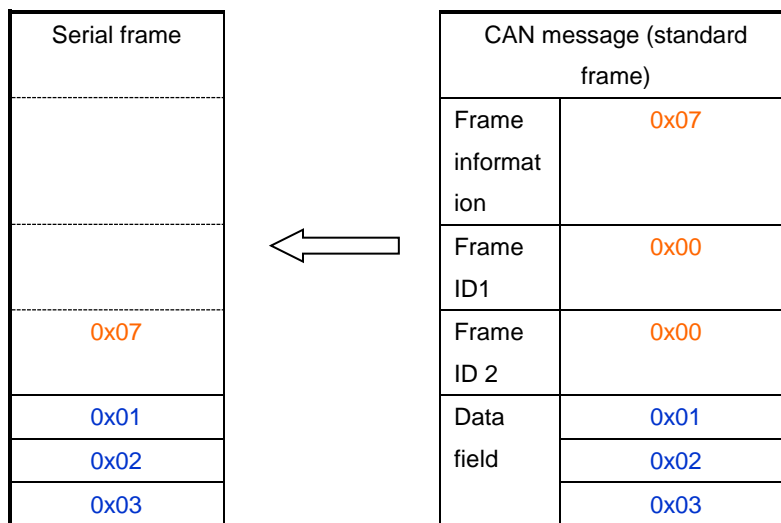
| Serial frame | | CAN message (standard frame) | |
|---|---|---|---|
| | | Frame information | 0x07 |
| | | Frame ID1 | 0x00 |
| 0x07 | | Frame ID 2 | 0x00 |
| 0x01 | | Data field | 0x01 |
| 0x02 | | | 0x02 |
| 0x03 | | | 0x03 |

| | | | |
|---|---|---|---|
| 0x04 | | 0x04 | |
| 0x05 | | 0x05 | |
| 0x06 | | 0x06 | |
| 0x07 | | 0x07 | |
| | | | |

Table 4.4 Example of converting CAN messages into serial frames (transparent mode)

## 4.2 Transparent Conversion with Flags

Transparent conversion with flag is a special transparent conversion. 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 address information in the serial frame into the frame ID of the CAN bus. Just tell the converter in the configuration that the address 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 message during conversion as the ID of the CAN message when the serial frame is forwarded. When the CAN message is converted into a serial frame, the ID of the CAN 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 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. The maximum length of the serial frame is the length of the buffer: 2,048 bytes.

The first data detected by the converter while the serial bus is idle is taken as the first character of the received frame. The time interval between characters in the frame must be less than or equal to the time for transmitting n characters (the value of n is pre-configured by the host computer) (the time for transmitting a character is divided by the number of bits contained in the character by the corresponding baud rate).

If the converter does not receive any more characters within the transmission time less than or equal to n characters after receiving a character, the converter considers the transmission of this frame to be over, and takes this character as the last character of this frame; characters after n character time do not belong to the frame, but the content of the next frame. Figure 4.5 shows the frame format.
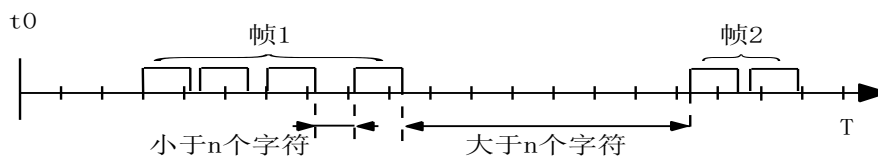
Figure 4.5 Serial frame time format (transparent conversion with flags)

2. CAN bus frame

The format of the CAN message remains unchanged, except that the corresponding frame ID of the CAN will also be converted into the serial frame.

### 4.2.2 Conversion Methods

1. Serial frames to CAN messages

The starting address and length of the CAN ID 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 frame ID in the serial frame is correspondingly converted into the frame ID field of the CAN message based on the prior configuration.(The method of big-endian storage is adopted. If the number of frame IDs is less than the number of frame IDs of the CAN message, the low byte of the frame ID in the CAN message is filled with 0). The other data are converted sequentially, as shown in Figure 4.6.

If a frame of CAN message has not finished converting the serial frame data, the same ID is still used as the frame ID of the CAN message to continue conversion until the serial frame conversion is completed.

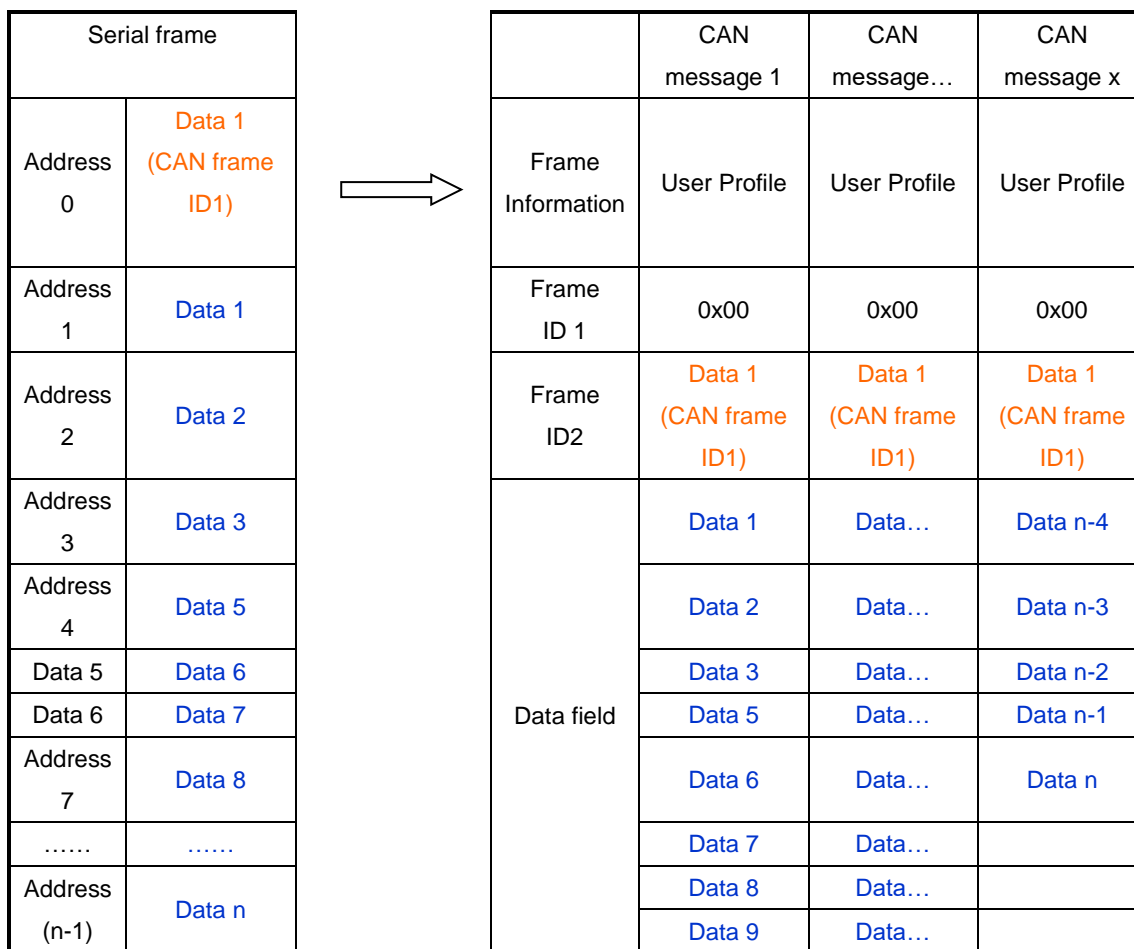| Serial frame | | | CAN message 1 | CAN message… | CAN message x |
|---|---|---|---|---|---|
| Address 0 | Data 1 (CAN frame ID1) | Frame Information | User Profile | User Profile | User Profile |
| Address 1 | Data 1 | Frame ID 1 | 0x00 | 0x00 | 0x00 |
| Address 2 | Data 2 | Frame ID2 | Data 1 (CAN frame ID1) | Data 1 (CAN frame ID1) | Data 1 (CAN frame ID1) |
| Address 3 | Data 3 | Data field | Data 1 | Data… | Data n-4 |
| Address 4 | Data 5 | | Data 2 | Data… | Data n-3 |
| Data 5 | Data 6 | | Data 3 | Data… | Data n-2 |
| Data 6 | Data 7 | | Data 5 | Data… | Data n-1 |
| Address 7 | Data 8 | | Data 6 | Data… | Data n |
| …… | …… | | Data 7 | Data… | |
| Address (n-1) | Data n | | Data 8 | Data… | |
| | | | Data 9 | Data… | |

Table 4.6 Converting serial frames to CAN messages (transparent conversion with flags)

2. Converting CAN 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 in sequence, as shown in Figure 4.7.

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.
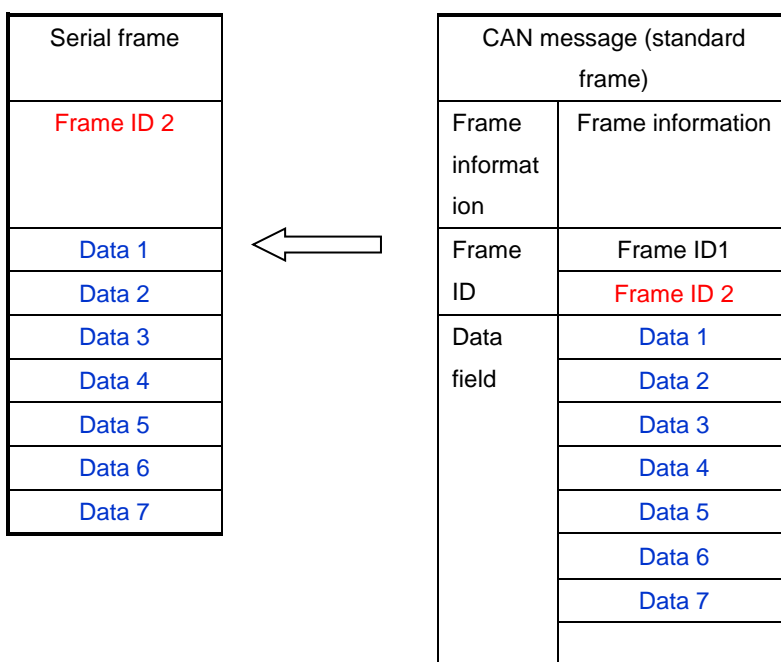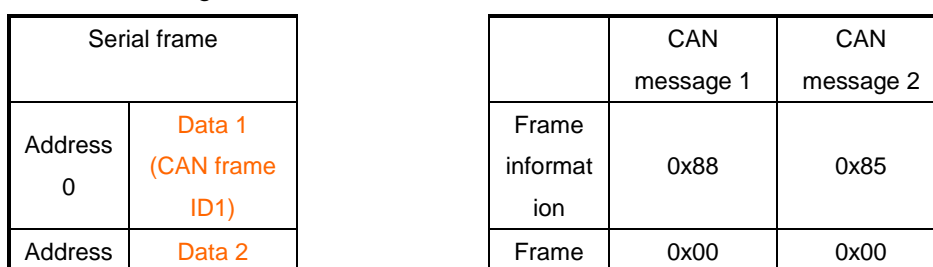
| Serial frame |
|---|
| Frame ID 2 |
| Data 1 |
| Data 2 |
| Data 3 |
| Data 4 |
| Data 5 |
| Data 6 |
| Data 7 |

| CAN message (standard frame) | |
|---|---|
| Frame information | Frame information |
| Frame ID | Frame ID1 |
| | Frame ID 2 |
| Data field | Data 1 |
| | Data 2 |
| | Data 3 |
| | Data 4 |
| | Data 5 |
| | Data 6 |
| | Data 7 |
| | |

Table 4.7 Converting CAN 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 identifier in the serial frame is 0 and the length is 3 (in the case of extended frame), the serial frame and the result of converting it into a CAN message are shown is shown in Figure 4.8. Among them, two frames of CAN messages are converted with the same ID.

| Serial frame | | | CAN message 1 | CAN message 2 |
|---|---|---|---|---|
| Address 0 | Data 1 (CAN frame ID1) | Frame information | 0x88 | 0x85 |
| Address | Data 2 | Frame | 0x00 | 0x00 |

| | | | | | |
|---|---|---|---|---|---|
| 1 | (CAN frame ID2) | | ID 1 | | |
| Address 2 | Data 3 (CAN frame ID3) | | Frame ID 2 | Data 1 (CAN frame ID1) | Data 1 (CAN frame ID1) |
| Address 3 | Data 1 | | Frame ID3 | Data 2 (CAN frame ID2) | Data 2 (CAN frame ID2) |
| Address 4 | Data 2 | | Frame ID4 | Data 3 (CAN frame ID3) | Data 3 (CAN frame ID3) |
| Data 5 | Data 3 | | | Data 1 | Data 9 |
| Data 6 | Data 4 | | | Data 2 | Data 10 |
| Address 7 | Data 5 | | | Data 3 | Data 11 |

Continued

| | | | | | |
|---|---|---|---|---|---|
| Address 8 | Data 6 | | Data field | Data 4 | Data 12 |
| Address 9 | Data 7 | | | Data 5 | |
| Address 10 | Data 8 | | | Data 6 | |
| Address 11 | Data 9 | | | Data 7 | |
| Address 12 | Data 10 | | | Data 8 | |
| Address 13 | Data 11 | | | | |
| Address 14 | Data 12 | | | | |

Table 4.8 Example of converting serial frames to CAN messages (transparent conversion with flags)

2.  Converting CAN messages to serial frames

Assuming that the start address of the configured CAN identifier in the serial frame is 0 and the length is 3 (in the case of an extended frame), the CAN message and the result of converting it into a serial frame are shown in Figure 4.9.
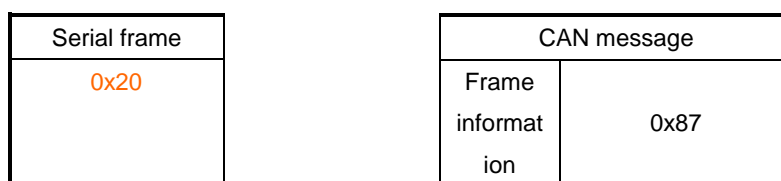
| Serial frame | | CAN message | |
|---|---|---|---|
| 0x20 | | Frame information | 0x87 |

| | 0x30 | | | 0x10 |
|---|---|---|---|---|
| | 0x40 | | Frame | 0x20 |
| | Data 1 | | ID | 0x30 |
| | Data 2 | | | 0x40 |
| | Data 3 | | | Data 1 |
| | Data 4 | | | Data 2 |
| | Data 5 | | | Data 3 |
| | Data 6 | | Data | Data 4 |
| | Data 7 | | field | Data 5 |
| | | | | Data 6 |
| | | | | Data 7 |
| | | | | |

Figure 4.9 Example of converting CAN messages to serial frames (transparent conversion with flags)

## 4.3 Format Conversion

CANCOM-100IE+ data conversion format. As shown in Figure 4.10, each CAN frame contains 13 bytes, and the content of 13 bytes includes CAN information, ID, and data.

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 frame ID data in the above-mentioned 13-byte serial frame. The "frame type" of the "CAN parameter" item of the configuration software is also invalid. The frame type is determined by the frame information in the 13-byte serial frame.

Standard frames, extended frames and even 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 13-byte serial frames.

In this mode, strictly follow the serial data format of 13 bytes for successful conversion. Refer to the example, as shown in Figure 4.11. Make sure that the frame information is correct, the reserved bits must be zero, and the data length cannot be greater than 8; otherwise, no conversion will be performed.

Each frame is fixed at 13 bytes. If it is insufficient, it must be filled with 0. In the same serial data frame, the serial data satisfying the 13-byte format corresponds to a CAN message; the serial data frame with less than 13 bytes is not converted. Therefore, make sure that the serial data frame to be converted is 13 bytes.

During the conversion from a serial frame to a CAN message, if a certain 13-byte data format is not standard in a 13-byte-aligned serial data frame, the 13-byte data will not be converted, and the following data will be converted. If some CAN messages are missing after conversion, check whether the 13-byte serial data format of the corresponding message is not the standard format.

1个CAN帧包含13个字节
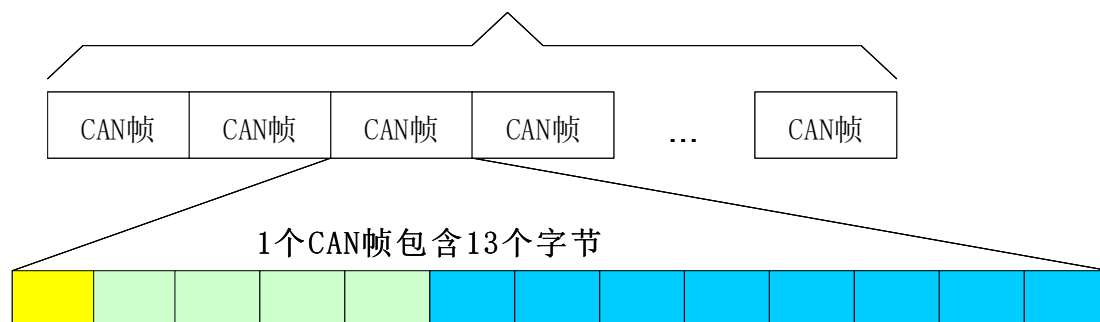
帧信息：长度1个字节，用于标识该CAN帧的一些信息，如类型、长度等

| Bit7 | | | | | | | Bit0 |
|------|-----|------|------|----|----|----|----|
| FF | RTR | 保留 | 保留 | D3 | D2 | D1 | D0 |

FF：标准帧和扩展帧的标识，1为扩展帧，0为标准帧。
RTR：远程帧和数据帧的标识，1为远程帧，0为数据帧。
保留值为0，不可写入1。
D3～D0 ：标识该CAN帧的数据长度。
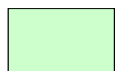
帧ID：长度4个字节，标准帧有效位是11位，扩展帧有效位是29位。

| 低字节 | | | 高字节 |
|------|-----|-----|------|
| 12h | 34h | 56h | 78h |

如上为扩展帧ID号
0x12345678的表示方式

| 低字节 | | | 高字节 |
|------|-----|-----|------|
| 00h | 00h | 03h | FFh |

如上为标帧ID号
0x3FF的表示方式
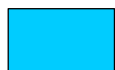
帧数据：长度8个字节，有效长度由帧信息的D3～D0的值决定。

| DATA1 | | | | | | | DATA8 |
|------|-----|-----|-----|-----|-----|-----|------|
| 11h | 22h | 33h | 44h | 55h | 66h | 77h | 88h |

如上为8个字节有效数据
的表示方式

| DATA1 | | | | | | | DATA8 |
|------|-----|-----|-----|-----|-----|-----|------|
| 11h | 22h | 33h | 44h | 55h | 66h | 00h | 00h |

如上为6个字节有效数据
的表示方式

Figure 4.10 Format conversion method

以下例子是一个扩展数据帧，ID为0x12345678，包含8个字节
数据（11h, 22h, 33h, 44h, 55h, 66h, 77h, 88h）的帧的表示方式

| 88h | 12h | 34h | 56h | 78h | 11h | 22h | 33h | 44h | 55h | 66h | 77h | 88h |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|

以下例子是一个标准数据帧，ID为0x3ff，包含6个字节数据
（11h, 22h, 33h, 44h, 55h, 66h）的帧的表示方式

| 06h | 00h | 00h | 03h | FFh | 11h | 22h | 33h | 44h | 55h | 66h | 00h | 00h |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|

Note: Each frame is fixed at 13 bytes; if it is insufficient, it must be filled with 0.

Figure 4.11 Data frame example

## 4.4 Modbus Conversion

Note: The Modbus conversion function can only be used when the CAN bus device message can be modified.

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 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 "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 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: 2,048 bytes.

2. CAN bus frame

If the device on the CAN side uses the Modbus protocol, a reliable transmission format needs to be defined. A segmentation protocol is used here, which defines a method for segmenting and reassembling information longer than 8 bytes.

The formulation of the segmented transmission protocol refers to the transmission protocol of segmented messages in DeviceNet. The format of the segmented message is shown in Table 4.1 (Using the extended frame as an example, the standard frame is only the length of the frame ID is different, 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.

The CAN bus frame format is described as follows:

Table 4.1 CAN2.0B extended frame format

| Bit number | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Frame information | FF | RTR | X | X | DLC (data length) | | | |
| Frame ID1 | X | X | X | ID.28 - ID.24 | | | | |
| Frame ID 2 | ID.23 - ID.16 | | | | | | | |
| Frame ID3 | ID.15 - ID.8 | | | | | | | |
| Frame ID4 | 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 | | | | | | | |

Fragmented packet flag: indicates whether the packet is a fragmented packet. This bit is 0 for a separate message, 1 for a frame in the segmented message. Note: When the CAN message is a single frame, the segment flag value is 0x00.

Segment type: indicates the first, middle or last segment. Table 4.2 lists the value definitions.

Table 4.2 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: flag of each segment, indicating 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.

### 4.4.2 Conversion Methods

During the conversion from the serial port side to the CAN side, the converter will only perform the conversion when it receives a complete and correct Modbus RTU; otherwise, it does not respond.

As shown in Figure 4.12, the address field of the Modbus RTU protocol is converted into the ID4 (extended frame) or ID2 (standard frame) of the frame ID in the CAN message. The flag remains unchanged during the frame conversion.

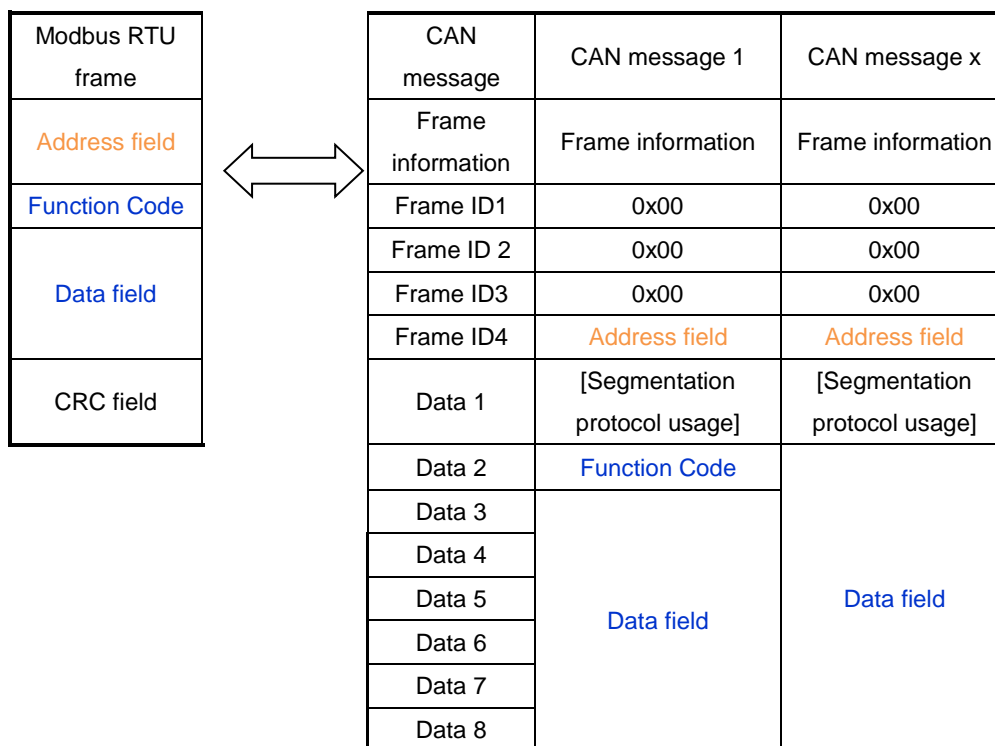| Modbus RTU frame | CAN message | CAN message 1 | CAN message x |
|---|---|---|---|
| Address field | Frame information | Frame information | Frame information |
| Function Code | Frame ID1 | 0x00 | 0x00 |
| | Frame ID 2 | 0x00 | 0x00 |
| Data field | Frame ID3 | 0x00 | 0x00 |
| | Frame ID4 | Address field | Address field |
| | Data 1 | [Segmentation protocol usage] | [Segmentation protocol usage] |
| CRC field | Data 2 | Function Code | |
| | Data 3 | | |
| | Data 4 | | |
| | Data 5 | Data field | Data field |
| | Data 6 | | |
| | Data 7 | | |
| | Data 8 | | |

Figure 4.12 Mutual conversion format of the communication frame (Modbus mode)

The CRC check byte is not converted into the CAN message, and the CAN message does not need to have the check byte of the serial frame, because the CAN bus itself has

an excellent check mechanism.

The converted data is the protocol content of Modbus RTU - function code and data field of the . In conversion, convert them in turn in the data field of the CAN 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. The CAN message frame can only transmit seven pieces of data. Therefore, the converter will segment the longer Modbus RTU frame into CAN messages and send them using the CAN segment protocol. You can take the function code and data field for processing when receiving on the CAN node.

For the Modbus protocol data of CAN 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.
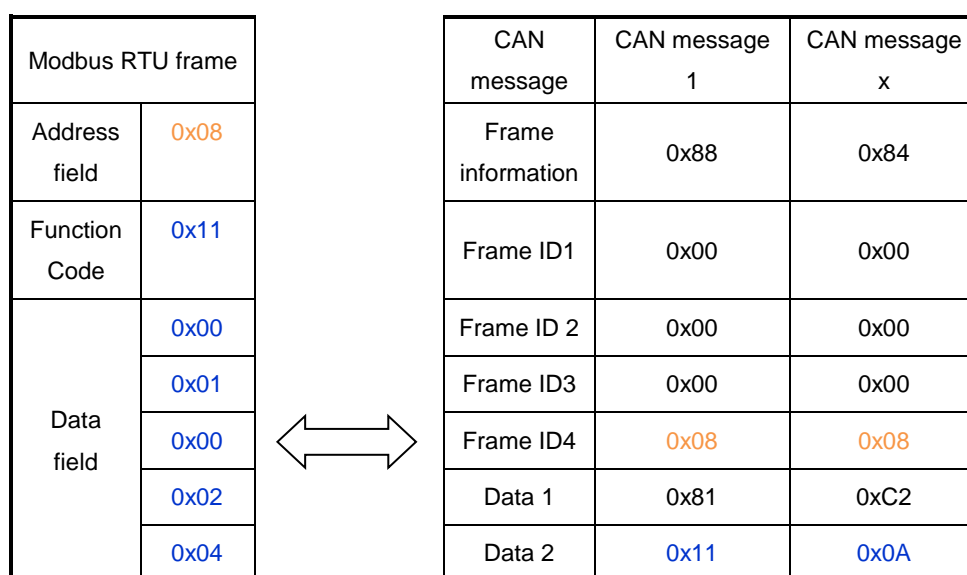
If the received data does not conform to the segmentation protocol, the group of data is discarded without conversion.

### 4.4.3  Conversion Examples

In the case of extended frame configuration, as shown in Figure 4.13, when the Modbus RTU frame is converted into a CAN message, the address 0x08 is directly filled into the frame ID4, and other frame IDs are filled with 0x00. Keep this frame ID unchanged while converting this frame.

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 Modbus RTU 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.

| Modbus RTU frame | | | CAN message | CAN message 1 | CAN message x |
|---|---|---|---|---|---|
| Address field | 0x08 | | Frame information | 0x88 | 0x84 |
| Function Code | 0x11 | | Frame ID1 | 0x00 | 0x00 |
| Data field | 0x00 | | Frame ID 2 | 0x00 | 0x00 |
| | 0x01 | | Frame ID3 | 0x00 | 0x00 |
| | 0x00 | ⟺ | Frame ID4 | 0x08 | 0x08 |
| | 0x02 | | Data 1 | 0x81 | 0xC2 |
| | 0x04 | | Data 2 | 0x11 | 0x0A |

| | | | | Data 3 | 0x00 | 0x01 |
|---|---|---|---|---|---|---|
| | 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 | |

Figure 4.13 Example of mutual conversion format of communication frames (Modbus mode)

## 5. Firmware Upgrade

To improve the device maintainability, CANCOM-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 5.1;

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 flashes, 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 5.1 Firmware upgrade

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.

## 6. Device Test

### 6.1 Device Preparation

After the "Configuration Switch" is turned on, connect the power supply again. The converter enters "Configuration Mode" (the CFG indicator is on). At this time, connect the PC to the converter by using the serial cable.

Run the "Smart Protocol Converter Configuration.exe" configuration software, select the PC serial port connected to the converter, and click the "Open Serial Port" button. If the software starts successfully, the following configuration parameters are open and can be changed and set.

If the "Current serial port unavailable" message appears, the serial port of the currently selected PC is unavailable or already occupied. If the "Device not connected" message appears, check that the converter has entered configuration mode (if the converter is operating in "Normal" mode, the software will also display the "No device connected" message), and note that the selected PC serial port is connected.

### 6.2 Communication Test

After disconnecting the "configuration switch", power on the device again. The converter enters "normal operation" mode. Use serial port debugging software for communication test.
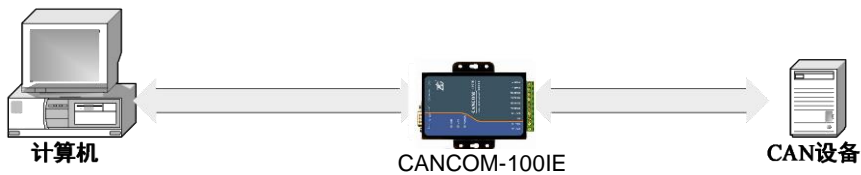


计算机       CANCOM-100IE       CAN设备

Figure 6.1 Communication test structure

Figure 6.1 shows the connection. The test requires a CAN device in addition to a PC to receive or send data. Note: In the same CAN-bus bus, the baud rate of CAN device and CANCOM-100IE+ converter must be the same. Use the serial port debugging software to select the same serial port communication baud rate as the converter, and observe whether the data received by the CAN device is consistent with the sent data. 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, the indicator of the bus will flash.

## 7. Installation Dimensions

The external dimension of CANCOM-100IE+ is 110 mm x 88 mm, and the shell has fixing holes.

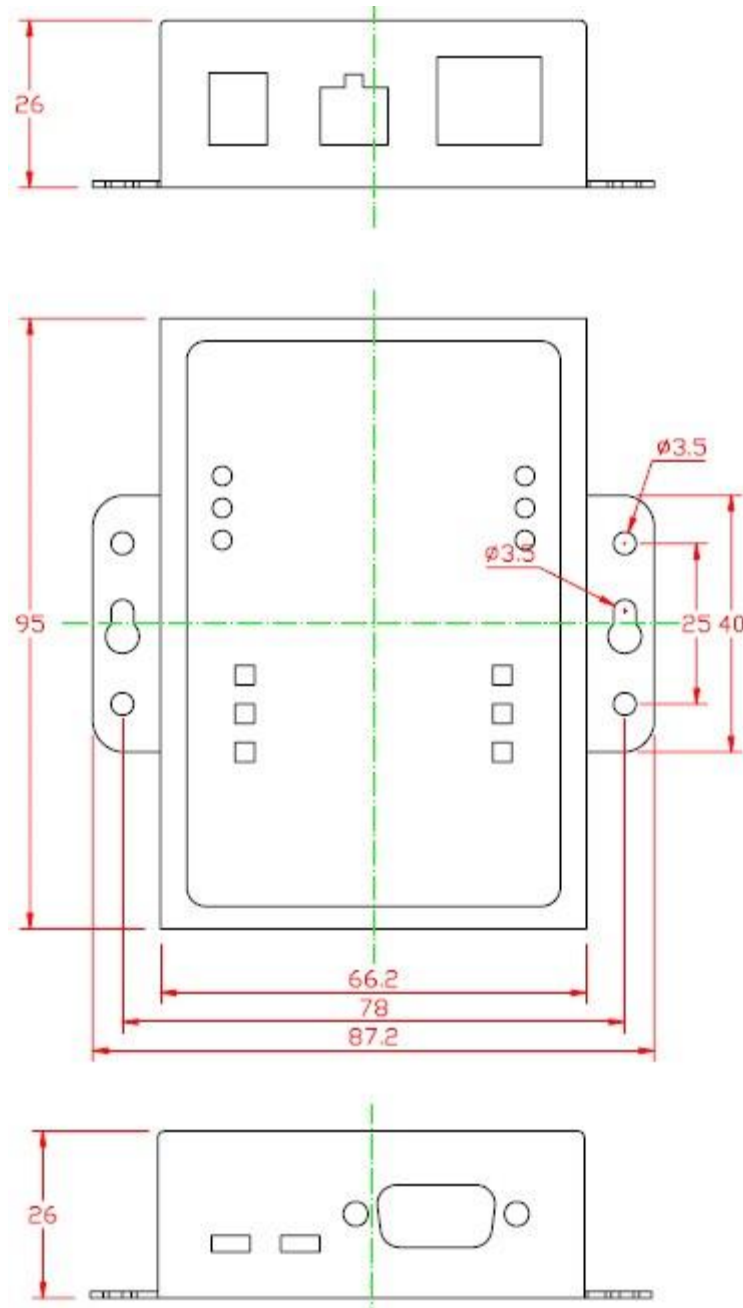Figure 7.1 shows the mounting holes on the device.



Figure 7.1 CANCOM-100IE+ installation dimensions

## 8. 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!

Dreams come true with professionalism and dedication.

**Guangzhou ZLG**

**Electronics Co., Ltd.**

For more details,
please visit
**www.zlg.cn**

Welcome to call the
national service hotline
**400-888-4005**