Chengdu Ebyte Electronic Technology Co.,Ltd

# Wireless Modem

## User Manual

E810-DTU(CAN-RS485) Product Specification
E810-DTU(CAN-RS485) Intelligent Protocol Converter




All rights to interpret and modify this manual belong to
Chengdu Ebyte Electronic Technology Co., Ltd.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# Contents

Disclaimer and Copyright Notice ..... 3
1 Introduction ..... 4
1.1 Brief Introduction ..... 4
1.2 Features ..... 4
1.3 Application ..... 4
2 Product Specification Parameters and Characteristics ..... 5
2.1 Basic Parameters ..... 5
2.2 Factory Default Parameters ..... 5
3 Introduction to Hardware Parameter Design ..... 6
3.1 Introduction to the design ..... 6
3.2 Dimension drawing ..... 8
3.3 Connection method ..... 9
3.3.1 RS485 Connection method ..... 9
3.3.2 CAN Connection method ..... 10
4 Description of the mode ..... 10
4.1 Operating mode ..... 10
4.2 Data conversion method ..... 10
4.2.1 Data conversion method ..... 11
4.2.2 Transparent transmission band information mode ..... 13
4.2.3 Protocol mode ..... 16
4.2.4 Modbus mode ..... 18
5 Operational Instructions ..... 21
5.1 Access to command configuration instructions ..... 21
5.2 Overview of Instructions ..... 21
5.3 Instruction Error Codes ..... 22
5.4 Command List ..... 22
5.5 Instruction Details ..... 22
5.5.1 AT test command ..... 22
5.5.2 AT+CANFLT Querying/Setting CAN Filter Information ..... 22
5.5.3 AT+CAN Querying/Setting CAN Transmission Parameter Information ..... 23



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

5.5.4 AT+EXAT Exit AT Command ..... 23
5.5.5 AT+E Query/Set Command Display Mode ..... 24
5.5.6 AT+MODBUSID Query/set device MODBUS address ..... 24
5.5.7 AT+MODE Querying/Setting the Device Operating Mode ..... 24
5.5.8 AT+MID Query Device Name ..... 25
5.5.9 AT+RESTORE Restore Factory Settings ..... 25
5.5.10 AT+REBT Reboot Device ..... 25
5.5.11 AT+UARTPKT Query/Set Serial Port Packet Information ..... 26
5.5.12 AT+UART Querying/Setting Serial Port Parameters ..... 26
5.5.13 AT+VER Querying Device Version Information ..... 26
6 FAQ ..... 27
6.1 Equipment is fragile ..... 27
6.2 Unable to set up a successful command ..... 27
6.3 Failure to use after parameter change ..... 27
Revision history ..... 27
About us ..... 28



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# Disclaimer and Copyright Notice

The information in this document, including the URL addresses for reference, is subject to change without notice. this document is provided “as is” without warranty of any kind, including any warranty of merchantability, fitness for a particular purpose, or non-infringement, and any warranty that any proposal, specification, or sample is referred to elsewhere. this document disclaims all liability, including liability for infringement of any patent, arising out of the use of the information in this document. This document does not grant any license, express or implied, by estoppel or otherwise, to use any intellectual property.

The test data in this document are from tests conducted in the laboratory of EBYTE, and the actual results may vary slightly.

All trade names, trademarks and registered trademarks mentioned herein are the property of their respective owners and are hereby acknowledged.

The final explanation right belongs to Chengdu Ebyte Electronic Technology Co.,Ltd.

Notes:

The contents of this manual are subject to change due to product version upgrade or other reasons. Chengdu Ebyte Electronic Technology Co.,Ltd. reserves the right to make changes to the contents of this manual without notice or prompting. EBYTE makes every effort to provide accurate information in this manual, however, EBYTE does not warrant that the contents of this manual are error-free, and all statements, information and recommendations in this manual do not constitute any warranty, express or implied.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# 1 Introduction

## 1.1 Brief Introduction




E810-DTU(CAN-RS485) is a small intelligent protocol conversion product independently developed by Chengdu Yibert Electronic Technology Co. E810-DTU(CAN-RS485) is a small intelligent protocol conversion product developed by Chengdu Yibert Electronic Technology Co. The product is powered by a wide voltage range from 8V to 28V, and it integrates one CAN-BUS interface and one RS485 interface, which can realise the bidirectional conversion of two different protocols of RS485 and CAN data. The product supports serial commands to configure device parameters and operating modes, and supports four data conversion modes, including transparent conversion, transparent conversion with information, protocol conversion, and Modbus RTU conversion. At the same time, E810-DTU (CAN-RS485) intelligent protocol converter has the characteristics of compact size, easy to install, etc., and it has a very high cost performance in CAN-BUS product development and data analysis applications, and it is a reliable assistant for engineering applications, project debugging and product development.

## 1.2 Features

- Supports bi-directional conversion of data between RS485 and CAN.
- Supports transparent conversion, transparent with information conversion and protocol conversion.
- Supports Modbus RTU protocol conversion.
- Supports two CAN identifier sending methods, which can be specified by fixed configuration or serial frame data.
- Supports RS485 interface parameter configuration.
- Supports 1200~115200 range parameter RS485 baud rate setting.
- Support AT command parameter configuration.
- Support AT command and hardware restore factory settings.
- With power indicator, status indicator and other status indication.

## 1.3 Application

- CAN-BUS network for industrial control, etc.
- Automotive, railway equipment networking
- Security and fire protection networks
- Underground remote communication
- Public address system
- Car park equipment control
- Intelligent home, intelligent building

copyright@2025，Chengdu Ebyte Electronic Technology Co.,Ltd.


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# 2 Product Specification Parameters and Characteristics

## 2.1 Basic Parameters

| Main parameters | Specification |
|---|---|
| Supply Voltage | 8V to 28V, 12V or 24V power supply recommended (over 28V may permanently damage the device) |
| Operating current | 38.2762mA@12V（RS485） |
| Operating temperature | -40℃～85℃，industrial grade |
| Interface Type | RS485：1*5*3.81，Crimping method; CAN: 1*4*3.81, crimping method |
| Communications level | 3.3V, if connected to 5V, level conversion required |
| Operating humidity | 10%～90%，Relative humidity, non-condensing |

## 2.2 Factory Default Parameters

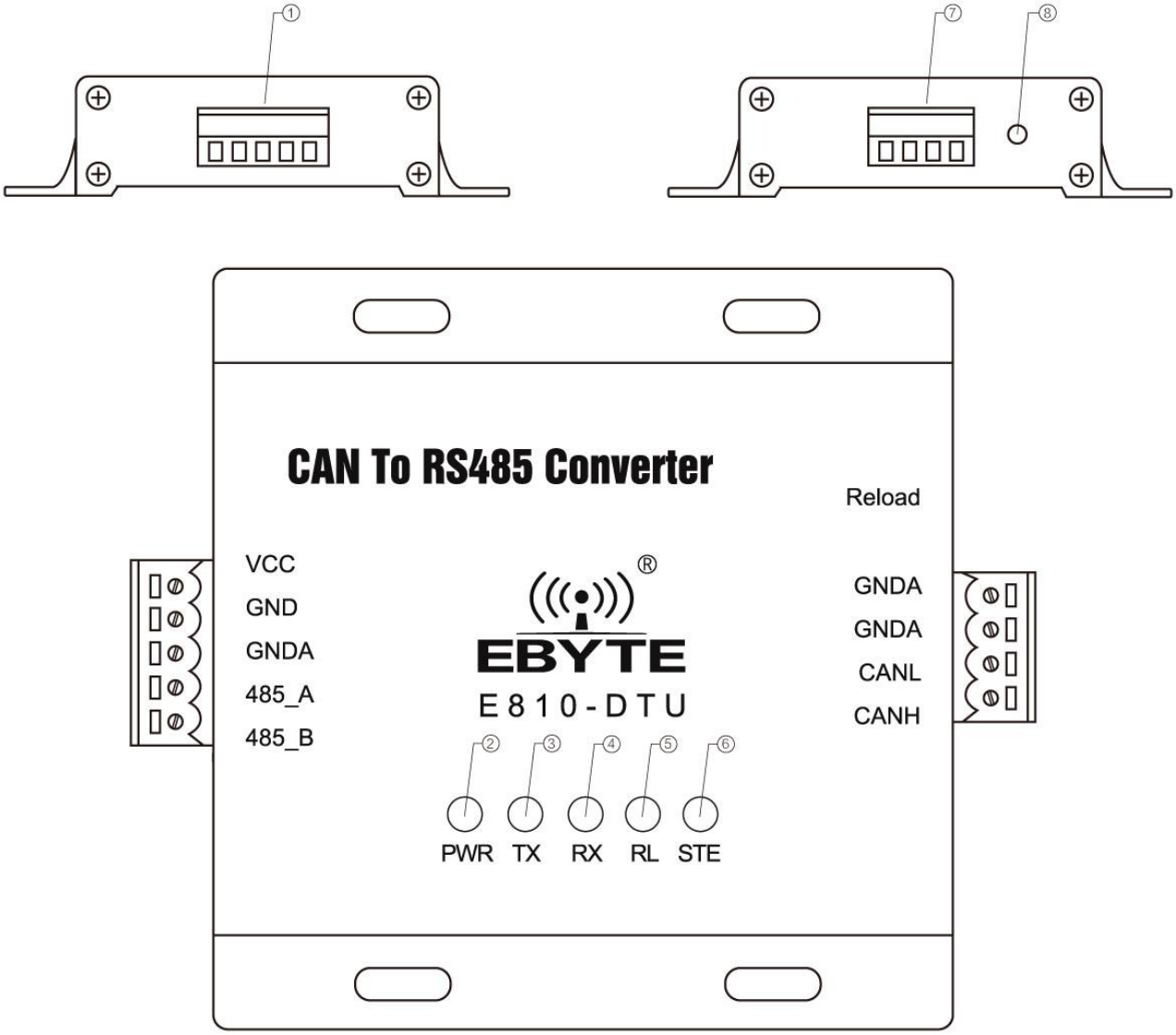
| | | |
|---|---|---|
| RS485 | Serial port baud rate | 115200 bps |
| | Parity checking | None |
| | Data bit | 8 |
| | Stop-bit | 1 |
| | Flow control | Close |
| Can | CAN baud rate | 100K bps |
| | CAN ID number | 0x00000000 |
| | Flow control | Close |
| Default Operating Mode | Transparent transfer mode | Receive all data types |
| Default Device Address | Modbus device address | Default address is 1 |
| Fractional Packet Parameters | Time | 10ms |
| | Bytes | 1000byte |

copyright@2025，Chengdu Ebyte Electronic Technology Co.,Ltd.


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# 3 Introduction to Hardware Parameter Design

## 3.1 Introduction to the design




| Pin No. | Pin Name | Pin Usage |
|---|---|---|
| 1 | VCC | Power supply, default 8-28V (standard 5V version can be customised), 12V/24V recommended. |
| 2 | GND | Power ground |
| 3 | GNDA | RS485 common, external to other devices RS485 device GND. |
| 4 | 485_A | RS485 bus data A |
| 5 | 485_B | RS485 bus data B |
| 6 | PWR | Power indicator |



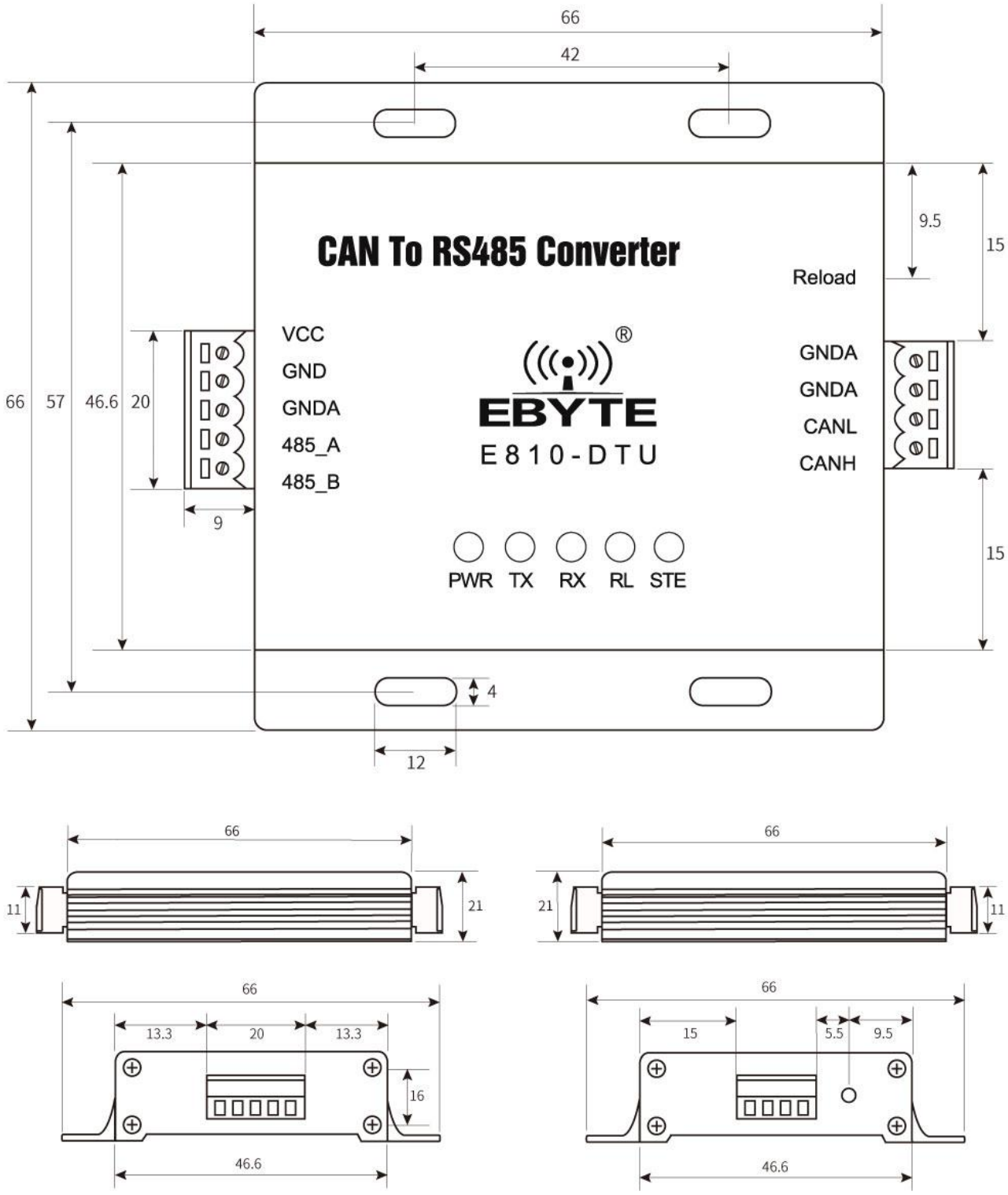
Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

| | | |
|---|---|---|
| 7 | TX | Serial Send Indicator |
| 8 | RX | Serial Receive Indicator |
| 9 | RL | Restore factory settings indicator, long press 5-10S device to restore production settings. |
| 10 | STE | Status indicator |
| 11 | GNDA | Signal reference ground |
| 12 | GNDA | Signal reference ground |
| 13 | CAN-H can bus H | CAN communication interface |
| 14 | CAN-L can bus L | CAN communication interface |

copyright@2025，Chengdu Ebyte Electronic Technology Co.,Ltd.


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 3.2 Dimension drawing

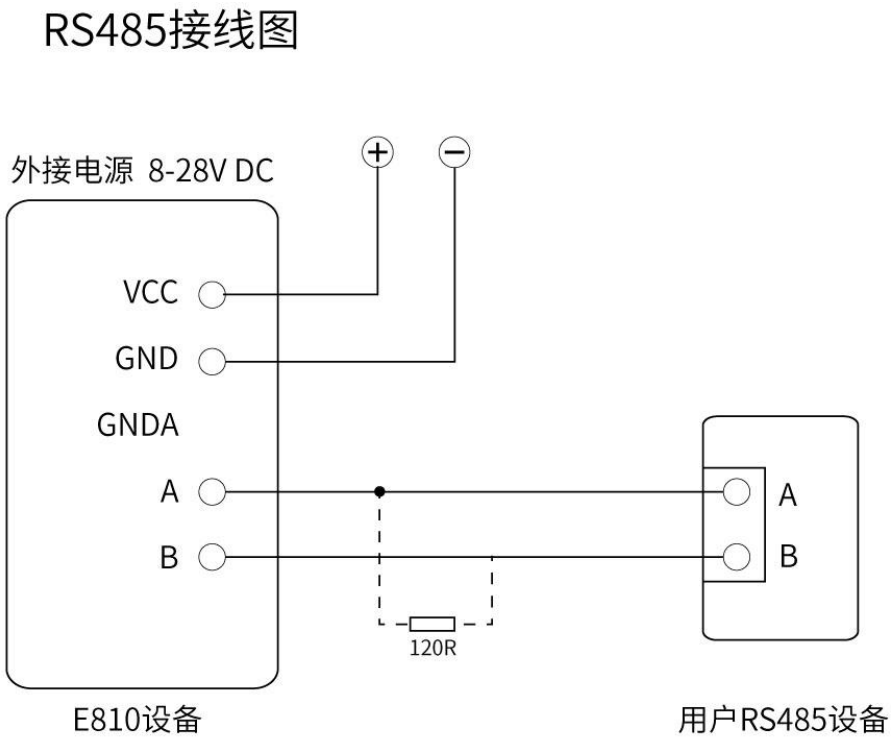





Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 3.3 Connection method

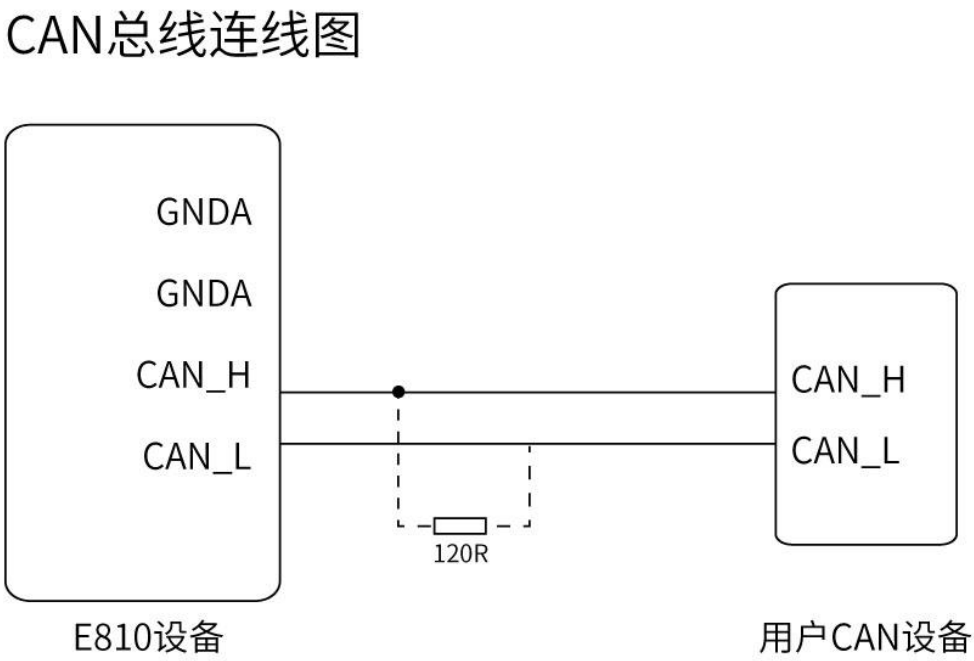
### 3.3.1 RS485 Connection method






Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 3.3.2 CAN Connection method




# 4 Description of the mode

## 4.1 Operating mode

The E810-DTU (CAN-RS485) device has two operating modes, normal mode and configuration mode.

| Mode | Function |
|---|---|
| Normal mode | The device is shipped from the factory in a generic mode and works normally on power-up. |
| Configuration mode | The mode in which the device can be configured, see Chapter 5 for details on how to enter configuration mode. |

## 4.2 Data conversion method

E810-DTU (CAN-RS485) device supports four types of data conversion: transparent conversion, transparent with information conversion, protocol conversion and MODBUS conversion. Support CAN and RS485 bi-directional conversion.

| Data conversion method | Conversion direction |
|---|---|
| Transparent conversion | CAN to RS485 Bidirectional Conversion |



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

| | |
|---|---|
| Transparent tape information conversion | CAN to RS485 Bidirectional Conversion |
| Protocol conversion | CAN to RS485 Bidirectional Conversion |
| MODBUS conversion | CAN to RS485 Bidirectional Conversion |

### 4.2.1 Data conversion method

Transparent conversion: the converter will be a format of the bus data as is converted to another bus data format, without additional data and do not modify the data. This not only achieves the exchange of data format but also does not change the data content, for both ends of the bus converter as 'transparent', so it is transparent conversion.

The E810-DTU (CAN-RS485) can convert the valid data received from the CAN bus to the RS485 bus output without any change. Similarly, the device can convert valid data received from the RS485 bus to the CAN bus output unchanged. Transparent conversion between RS485 and CAN is possible.

1.Serial frames to CAN messages

The entire data of the serial frame is sequentially filled into the data field of the CAN telegram frame. The converter receives and converts the data as soon as it detects it on the serial bus. The frame type and frame ID of the converted CAN message come from the user's pre-configuration and remain unchanged during the conversion process. The corresponding format for data conversion is shown in Figure 4.1.

If the length of the received serial frame is less than or equal to 8 bytes, the characters 1 to n (n is the length of the serial frame) are filled in the data field of the CAN telegram in the order of 1 to n byte positions (e.g. n is 7 in Figure 4.1). If the number of bytes in the serial frame is greater than 8, the processor fills the data field of the CAN telegram with the first 8 characters, starting with the first character of the serial frame. After sending the data to the CAN bus, the remaining serial frame data is converted and filled into the data field of the CAN telegram until it has been converted.

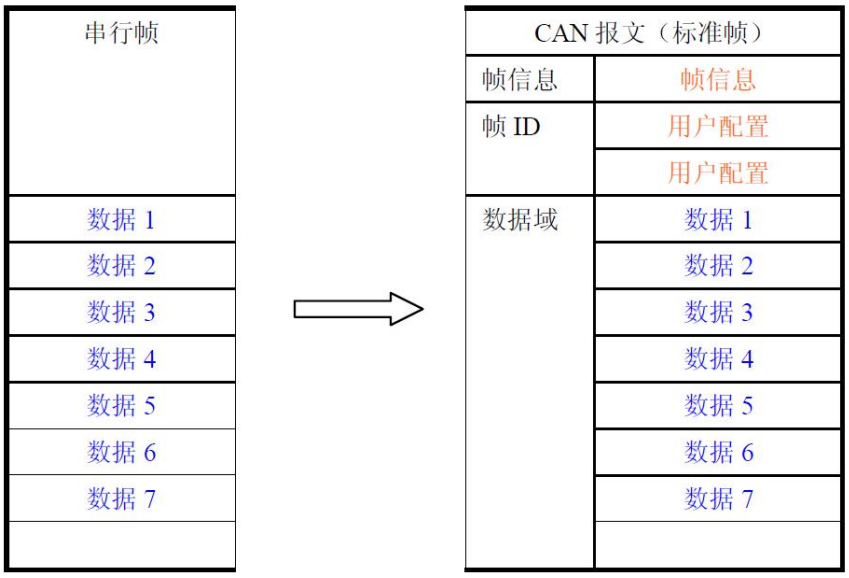



Figure 4.1 Conversion of serial frames into CAN messages (transparent conversion)

Conversion example:

Assuming that the frame information configured for conversion to CAN messages is 'Standard Frame' and the frame ID (ID1, ID2) is set to 0060, then the conversion format is shown in Figure 4.2.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

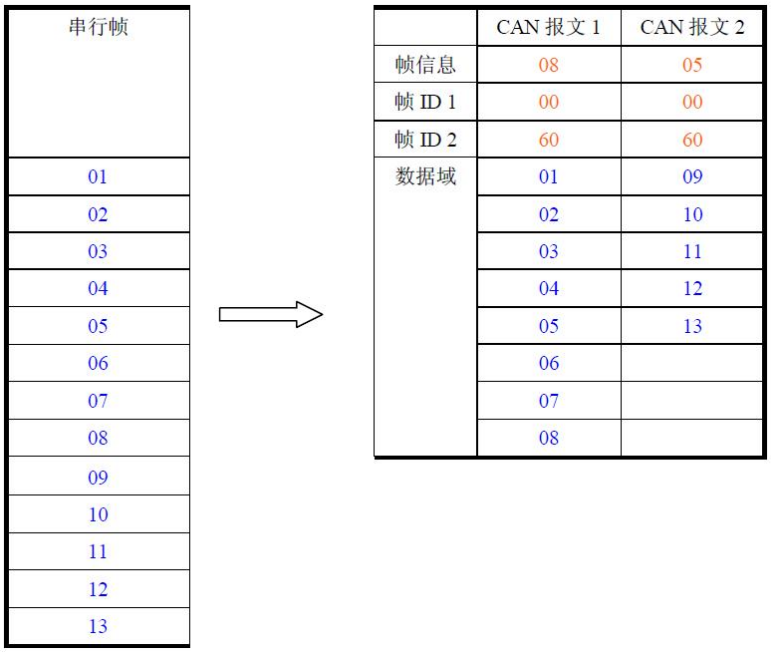



Figure 4.2 Serial frame to CAN message example (transparent conversion)

2. CAN messages to serial frames

The data field of the CAN telegram frame is filled with all data in sequence in the serial frame data. The converter receives and converts the data as soon as it detects it on the CAN bus.

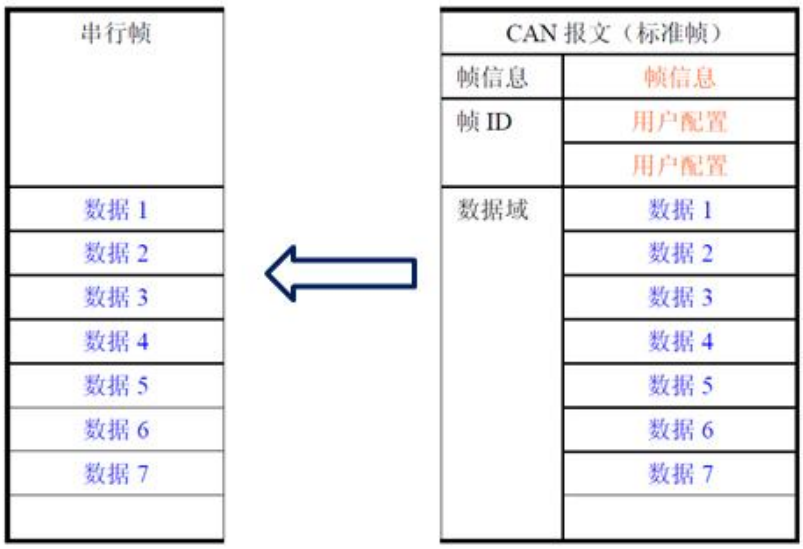



Figure 4.3 Conversion of CAN messages into serial frames (transparent conversion)

Conversion example:

Assuming that the CAN message frame information is ‘standard frame’ and the frame IDs (ID1, ID2) are set to 0060, the conversion format is shown in Figure 4.4.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

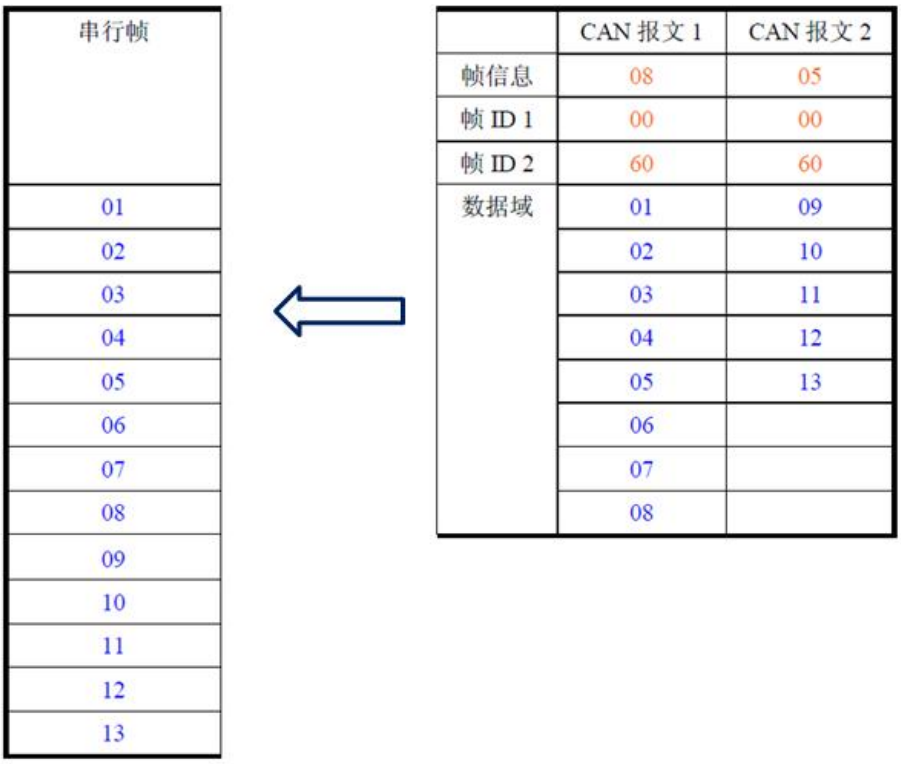



Figure 4.4 Example of CAN message to serial frame (transparent conversion)

## 4.2.2 Transparent transmission band information mode

Transparent conversion is a special use of transparent conversion and does not have a protocol attached. This conversion method is based on the common characteristics of the usual serial frames and CAN messages and makes it easy to set up the same communication network with these two different bus types.

In this way, the receiver of the CAN bus can add the frame information and frame ID of the received CAN message to the converted serial frame. In this way, the receiver can clearly see the CAN message information of the sender and can use this information in a more flexible way.

Conversion method:

1.Serial frames to CAN messages

With transparent conversion, the entire data of the serial frame is sequentially filled into the data field of the CAN telegram frame. The converter receives and converts the data as soon as it detects it on the serial bus. The frame type and frame ID of the converted CAN message come from the user's pre-configuration and remain unchanged during the conversion process. The corresponding format for data conversion is shown in Figure 4.1.

If the length of the received serial frame is less than or equal to 8 bytes, the characters 1 to n (n is the length of the serial frame) are filled in the data field of the CAN telegram in the order of 1 to n byte positions (e.g. n is 8 in Figure 4.5).

If the number of bytes in the serial frame is greater than 8, the processor fills the data field of the CAN telegram with the first 8 characters, starting with the first character of the serial frame. After sending the data to the CAN bus, the remaining serial frame data is converted and filled into the data field of the CAN telegram until it has been converted.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

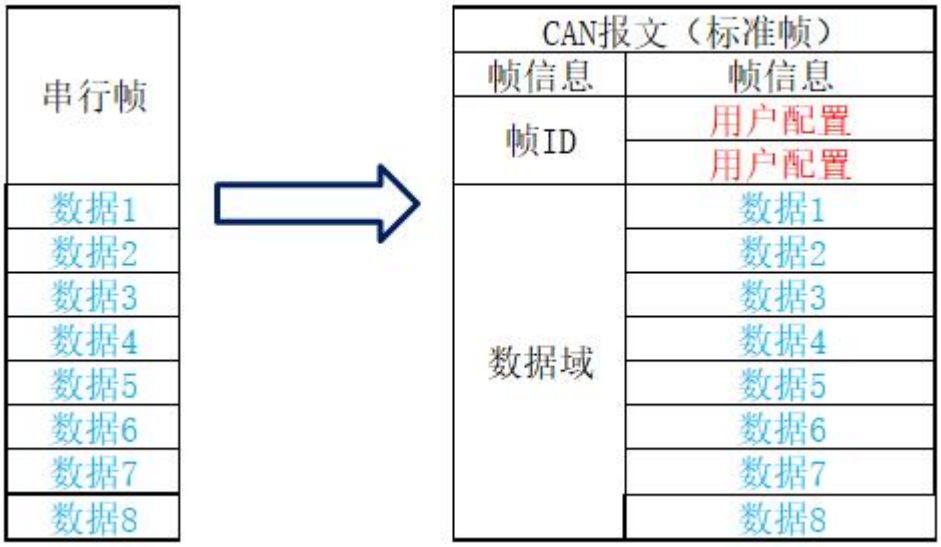



Figure 4.5 Conversion of serial frames into CAN messages (pass-through band information)

Conversion example:

Assuming that the frame information configured for conversion to CAN messages is ‘Standard Frame’ and the frame IDs (ID1, ID2) are set to 0060, the conversion format is shown in Figure 4.6.

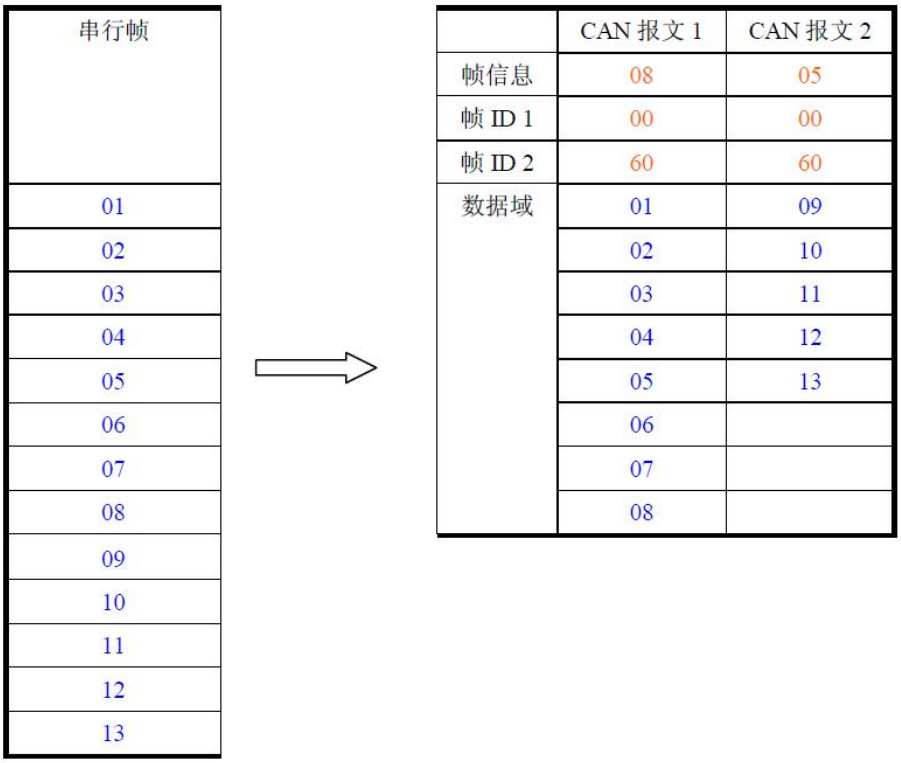



Figure 4.6 Serial frame to CAN message example (pass-through band information conversion)

2.CAN message to serial frame

As soon as the converter detects data on the CAN bus, it receives and converts the data, and converts one frame when it receives a CAN message, and adds the CAN frame information and the frame ID to the serial frame every time it converts (this conversion is the same as that of the following protocol mode of converting a CAN message to a serial



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

frame, please refer to the protocol mode for details). This is shown in the figure below

Please note: The frame format (standard or extended) of both serial and CAN telegrams should comply with the pre-configured frame format requirements when applying them, otherwise the communication may not work properly.

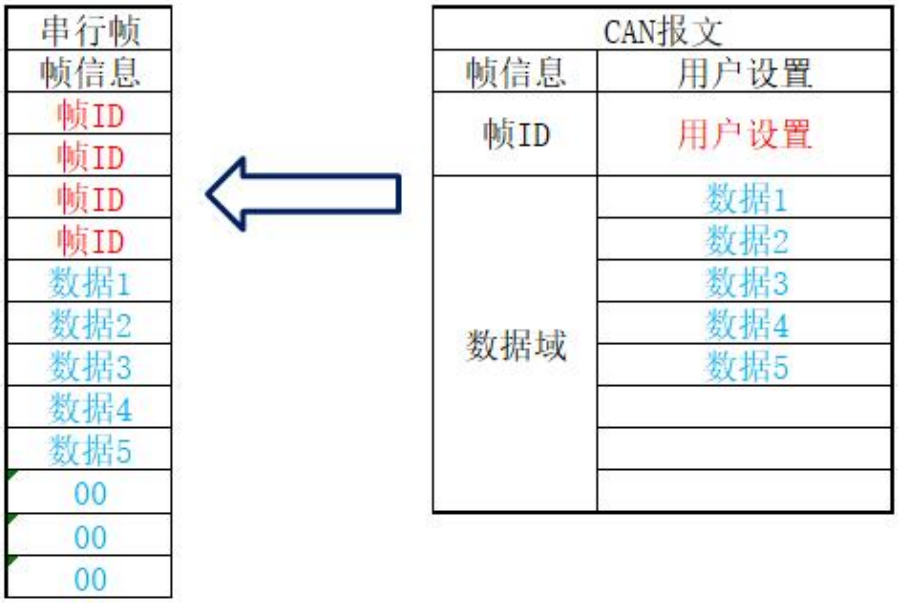



Figure 4.7 CAN message to serial frame (transparent band information conversion)

CAN transmission:

Frame format: Extended frame

Frame type: Data frame

ID ： 0x12345678

Data : AAh BBh CCh DDh EEh

Serial frame reception: 85 12 34 56 78 AA BB CC DD EE 00 00 00

0x85 means the frame format is extended frame, the frame type is data frame and the data length is 5.

The last four bits indicate the CAN ID 12345678.

The last 8 bits are the data area, the valid length is 5, the rest of the bits are zero-complemented.

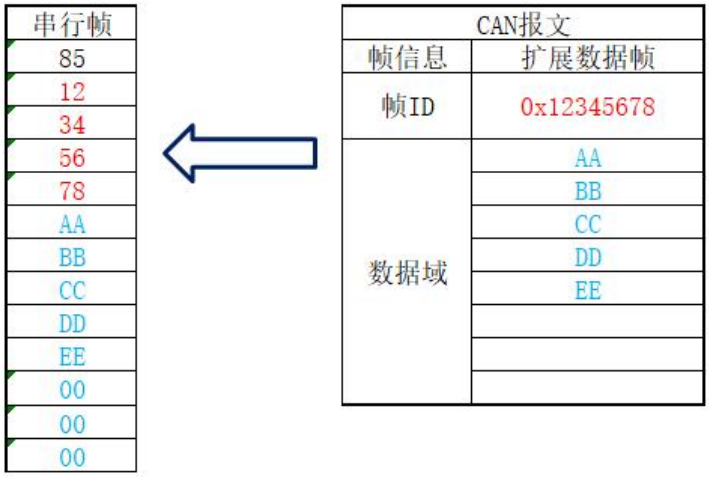


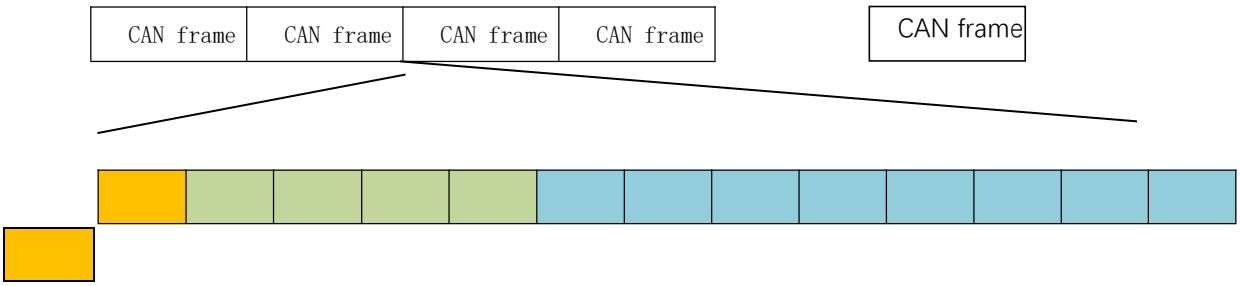

Figure 4.8 Example of CAN message to serial frame (transparent band information conversion)



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.2.3 Protocol mode

E810-DTU (CAN-RS485) device data conversion format is shown below, each CAN frame contains 13 bytes, and the content of 13 bytes includes CAN frame information + frame ID + frame data.




**Frame Information:** 1 byte in length, used to identify some information of CAN frame, such as type, length, etc.



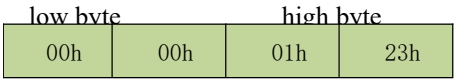

BIT7 BIT0

**FF:** Identifier bits for standard and extended frames, 1 for extended frames, 0 for standard frames.
**RTR:**Identifier bits for remote frames and data frames, 1 for remote frames and 0 for data frames.
**Reserve:** The value 0 is reserved and 1 cannot be written.

**D3～D0：** The Data Length bit identifies the data length of this CAN frame.
**frame ID：** Length 4 bytes, Standard Frame Valid Bits 11 bits, Extended Frame Valid Bits 29 bits.






Extended frame ID number as above

Representation of 0x12345678

Standard frame ID number as above

Representation of 0x123

**Frame Data:** 8 bytes in length, the effective length is determined by the value of D3~D0 in the frame information.



DATA1 DATA8




The above is a representation of 8 bytes of valid data.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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

The above is a representation of 5 bytes of valid data.

Example:

The following example is a representation of an extended data frame with frame ID 0x11223344 containing 8 bytes of valid data (11h,22h,33h,44h,55h,66h,77h,88h).

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

The following example is a representation of a standard data frame with frame ID 0x789 containing 5 bytes of valid data (12h,34h,56h,78h,90h).

| 05h | 00h | 00h | 07h | 89h | 12h | 34h | 56h | 78h | 90h | 00h | 00h | 00h |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

**Please note: Each frame is fixed at 13 bytes, and any shortfall must be made up to 0, otherwise it will result in a communication error.**

Example:

1.CAN to serial frame:

CAN transmission:

Frame format: Extended frame

Frame type: Data frame

ID ： 0x12345678

Data : AAh BBh CCh DDh EEh

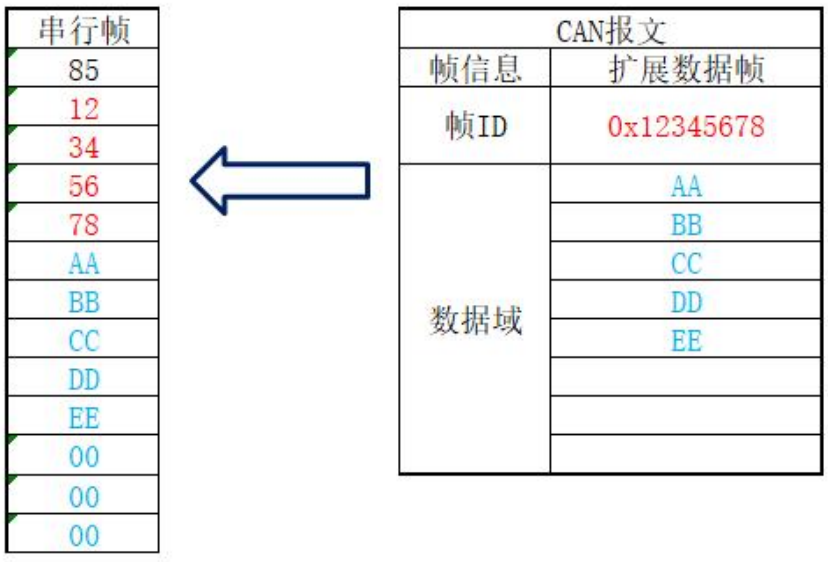
Serial frame reception: 85 12 34 56 78 AA BB CC DD EE 00 00 00

0x85 means the frame format is extended frame, the frame type is data frame and the data length is 5.

The last four bits indicate the CAN ID 12345678.

The last 8 bits are the data area, the valid length is 5, and the rest of the bits are zero-complemented.

As shown in the figure below:




2. Serial frames to CAN.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

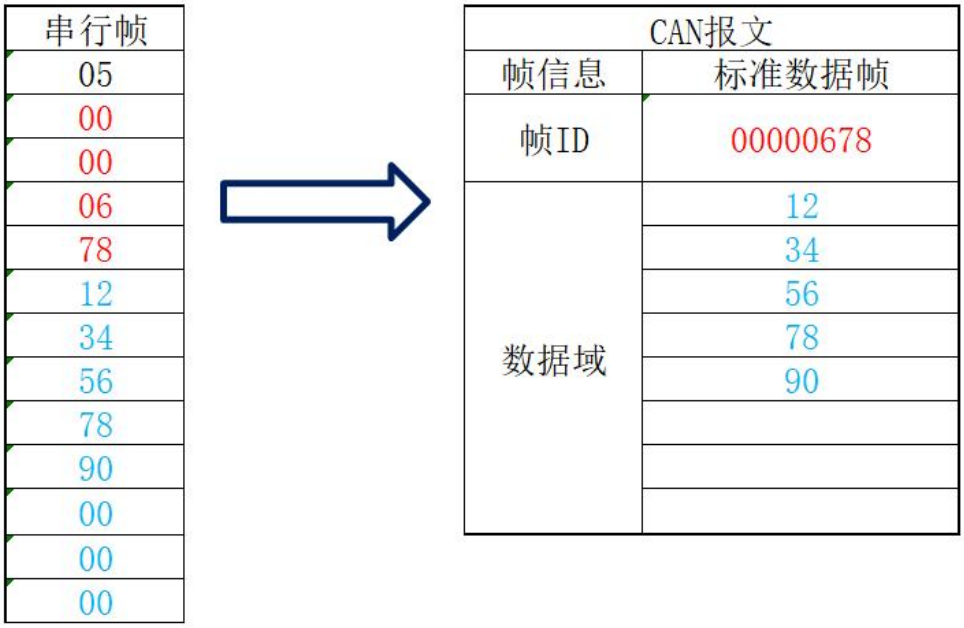
Serial frame transmission: 05 00 00 06 78 12 34 56 78 90 00 00 00 00

0x05 means that the frame format is standard, the frame type is data frame, and the data length is 5.

00 00 06 78 means ID 0678

12 34 56 78 90 00 00 00 00 is the data area, the valid length is 5

As shown in the figure below:




## 4.2.4 Modbus mode

Modbus conversion mode, RTU conversion mode supported The E810-DTU(CAN-RS485) device receives and responds to commands sent to it (via UART).

The E810-DTU(CAN-RS485) device supports two types of Modbus commands:

Read register (function code 03), Write multiple registers (function code 16).

There is an internal buffer in the converter device to cache the received CAN frame data, and there are 64 levels of buffer according to address 0~63. The buffer address starts from 0 to 63, and can continuously cache 8 frames of CAN data (8 bytes of data per frame, 64 bytes in total). When the first frame of CAN data is received, the CAN frame data is stored in address 0, and the CAN frame data received later is stored in increasing order according to the address. If the 64-level cache is full, the newly received CAN frame data will be stored in address 0 and overwrite the original number, following FIFO.

Read register (function code 03):

Send command:

[Device address] [Command number 03 (0x03)] [Start register address high 8 bits] [Low 8 bits] [Number of registers to read high 8 bits] [Low 8 bits] [High 8 bits for CRC check] [Low 8 bits for CRC check]

The read format only allows to read 00 08 from address 00 00 (one-time read 8 bytes that is a frame of data, i.e., read the data of address 00 00 - 00 07), after successful reading, the 8 bytes of data will be cleared, and the data after its address will be shifted forward by 8 data.

For example:

The device is in MODBUS mode and the CAN bus receives 4 frames of data:



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

First frame: 0x01 0x02 0x03 0x04 total 4 bytes of data

Second frame: 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F total 6 bytes of data

Third frame: 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 total 8 bytes of data

Fourth frame: 0XAA 0XBB 0XCC Total 3 bytes of data

They are stored in the MODBUS cache address:

(the address without data is 0x00)

| 缓存地址 | 数据 | 缓存地址 | 数据 |
|---|---|---|---|
| 0x31 | 0x00 | 0x63 | 0x00 |
| 0x30 | 0x00 | 0x62 | 0x00 |
| 0x29 | 0x00 | 0x61 | 0x00 |
| 0x28 | 0x00 | 0x60 | 0x00 |
| 0x27 | 0x00 | 0x59 | 0x00 |
| 0x26 | 0xCC | 0x58 | 0x00 |
| 0x25 | 0xBB | 0x57 | 0x00 |
| 0x24 | 0xAA | 0x56 | 0x00 |
| 0x23 | 0x88 | 0x55 | 0x00 |
| 0x22 | 0x77 | 0x54 | 0x00 |
| 0x21 | 0x66 | 0x53 | 0x00 |
| 0x20 | 0x55 | 0x52 | 0x00 |
| 0x19 | 0x44 | 0x51 | 0x00 |
| 0x18 | 0x33 | 0x50 | 0x00 |
| 0x17 | 0x22 | 0x49 | 0x00 |
| 0x16 | 0x11 | 0x48 | 0x00 |
| 0x15 | 0x00 | 0x47 | 0x00 |
| 0x14 | 0x00 | 0x46 | 0x00 |
| 0x13 | 0x0F | 0x45 | 0x00 |
| 0x12 | 0x0E | 0x44 | 0x00 |
| 0x11 | 0x0D | 0x43 | 0x00 |
| 0x10 | 0x0C | 0x42 | 0x00 |
| 0x09 | 0x0B | 0x41 | 0x00 |
| 0x08 | 0x0A | 0x40 | 0x00 |
| 0x07 | 0x00 | 0x39 | 0x00 |
| 0x06 | 0x00 | 0x38 | 0x00 |
| 0x05 | 0x00 | 0x37 | 0x00 |
| 0x04 | 0x00 | 0x36 | 0x00 |
| 0x03 | 0x04 | 0x35 | 0x00 |
| 0x02 | 0x03 | 0x34 | 0x00 |
| 0x01 | 0x02 | 0x33 | 0x00 |
| 0x00 | 0x01 | 0x32 | 0x00 |

Write this instruction to a special tool (Modbus CRC 16 calculator) with the instruction: 01 03 00 00 00 08 (in a slave device with address 01, read the 8 bars of data starting from address 0000), calculate the CRC checksum value and add it to the end of the instruction. Send this instruction via UART: 01 03 00 00 00 00 08 44 0C , when the slave receives this instruction, it will return the cached value inside the slave conversion device. (When no new CAN frame data is received, the buffer value read by the host is all zeros).

Return command:

[Device address] [Command number 03] [Number of bytes returned] [Data 1] [Data 2]... [Data n] [Higher 8 bits of CRC check] [Lower 8 bits of CRC check]

As:

The slave receives the instruction 01 03 00 00 00 00 08 44 0C,, then returns:

01 03 10 00 01 00 02 00 03 00 04 00 00 00 00 00 00 00 00 00 1F 9F

After reading is completed, the 8 bytes of data are cleared and the data after its address is shifted forward 8 data. This is shown in the figure below:



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

| 缓存地址 | 数据 | 缓存地址 | 数据 |
|---|---|---|---|
| 0x31 | 0x00 | 0x63 | 0x00 |
| 0x30 | 0x00 | 0x62 | 0x00 |
| 0x29 | 0x00 | 0x61 | 0x00 |
| 0x28 | 0x00 | 0x60 | 0x00 |
| 0x27 | 0x00 | 0x59 | 0x00 |
| 0x26 | 0x00 | 0x58 | 0x00 |
| 0x25 | 0x00 | 0x57 | 0x00 |
| 0x24 | 0x00 | 0x56 | 0x00 |
| 0x23 | 0x00 | 0x55 | 0x00 |
| 0x22 | 0x00 | 0x54 | 0x00 |
| 0x21 | 0x00 | 0x53 | 0x00 |
| 0x20 | 0x00 | 0x52 | 0x00 |
| 0x19 | 0x00 | 0x51 | 0x00 |
| 0x18 | 0xCC | 0x50 | 0x00 |
| 0x17 | 0xBB | 0x49 | 0x00 |
| 0x16 | 0xAA | 0x48 | 0x00 |
| 0x15 | 0x88 | 0x47 | 0x00 |
| 0x14 | 0x77 | 0x46 | 0x00 |
| 0x13 | 0x66 | 0x45 | 0x00 |
| 0x12 | 0x55 | 0x44 | 0x00 |
| 0x11 | 0x44 | 0x43 | 0x00 |
| 0x10 | 0x33 | 0x42 | 0x00 |
| 0x09 | 0x22 | 0x41 | 0x00 |
| 0x08 | 0x11 | 0x40 | 0x00 |
| 0x07 | 0x00 | 0x39 | 0x00 |
| 0x06 | 0x00 | 0x38 | 0x00 |
| 0x05 | 0x0F | 0x37 | 0x00 |
| 0x04 | 0x0E | 0x36 | 0x00 |
| 0x03 | 0x0D | 0x35 | 0x00 |
| 0x02 | 0x0C | 0x34 | 0x00 |
| 0x01 | 0x0B | 0x33 | 0x00 |
| 0x00 | 0x0A | 0x32 | 0x00 |

This leads to the next read register command.

Write multiple registers (function code 16): when successfully written, the written data is sent to the CAN bus.

Send command:

[Device address] [Command number 16 (0x10)] [Address of register to be placed high 8 bits] [low 8 bits] [Number of data high 8 bits] [Number of data low 8 bits] [Number of data to be placed high 8 bits] [low 8 bits] [.....] [.....] [ CRC checksum high 8 bits] [CRC checksum low 8 bits]

As:

Write 5 bytes of data to be sent out via the write command: 01 10 00 00 00 00 05 0A 00 11 00 22 00 33 00 44 00 55 , write this command to a special tool (Modbus CRC 16 calculator), calculate the CRC checksum value and add it to the end of the command: 01 10 00 00 00 00 05 0A 00 11 00 22 00 33 00 44 00 55 47 84, sent in HEX format, you can write 5 bytes of data, successful writing, will be written to the CAN bus, that is, at this time the CAN bus will send 5 bytes of data: 11,22,33,44,55. CAN message, frame information, frame ID for the user to configure in advance.

Note: The register address of this command is only allowed to be 00 00, and it is not allowed to send such command format as: 01 10 00 01 00 05 0A 00 11 00 22 00 33 00 44 00 55.

It is not allowed to send commands such as: 01 10 00 01 00 05 0A 00 00 11 00 22 00 33 00 44 00 55:

Write 8 bytes of data to be sent out through the write command: 01 10 00 00 00 00 08 10 00 00 11 00 00 22 00 33 00 44 00 55 00 66 00 77 00 88, write this command to the special tool (Modbus CRC 16 Calculator), calculate the CRC check value and add it to the end of the command: 01 10 00 00 00 00 08 10 00 11 00 22 00 33 00 44 00 55 00 66 00 77 00 88 FE D5, sent in HEX format, you can write 8 bytes of data, successful writing, will be written to the CAN bus, that



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

is, at this time the CAN bus will send 8 bytes of data: 0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88. In the CAN message, frame information, frame ID is pre-configured by the user.

Command formats such as 01 10 00 00 00 00 09 12 00 11 00 22 00 33 00 44 00 00 55 00 66 00 77 00 88 00 99 are not allowed.

# 5 Operational Instructions

## 5.1 Access to command configuration instructions

Note: The mode indicator blinks at 1Hz in the general operating mode; when entering the command configuration mode, the indicator blinks at 5Hz, and the configured parameters take effect after reset, whether it is a software entry or a hardware entry.

Timing of switching from other modes to command mode (software entry):

Send ‘+++’ continuously through UART, after the device receives ‘+++’, the device starts to start with a 3-second timeout, and if it receives any AT command within the timeout period, it will successfully switch to the configuration mode (after entering, the indicator light of the device blinks at a frequency of 5Hz). 5Hz). At this time, you can configure the parameters, the parameter list and parameter description please see the following introduction.

After the parameters are successfully configured, send the command ‘AT+ EXAT’ through UART, and the device will return ‘+OK’ after receiving the command, and exit the configuration mode to the mode before configuration at the same time. Then short the reset RESET pin, the parameter will take effect. You can also use the command ‘AT+REBT’ to reset the device after the parameters are successfully configured with the AT command.

## 5.2 Overview of Instructions

AT instruction refers to the instruction set for users to transfer commands with the device via UART in command mode, and the format of AT instruction will be explained in detail later. Switch to configuration mode after successful power-up, you can set up the device via UART.

Timing of switching from pass-through mode to command mode:

The serial device sends ‘++++’ to the device continuously, after the device receives ‘++++’, it starts to start in 3 seconds timeout, if it receives any AT instruction within the timeout, it switches to the configuration mode successfully.

Finally, send the command ‘AT+ EXAT’ through UART, after receiving the command, the device will return ‘+OK’ and exit the configuration at the same time. Then short the reset RESET pin, the parameters take effect. You can also use the command ‘AT+REBT’ to reset the device after successfully configuring the parameters with the AT command.

Note: After configuring the parameters, the parameters must be reset before they take effect.

Description： <CR>：ASCII code 0x0D

<LF>: ASCII code 0x0A

copyright@2025，Chengdu Ebyte Electronic Technology Co.,Ltd.


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 5.3 Instruction Error Codes

| Error code | Instruction |
|---|---|
| -1 | Invalid command format |
| -2 | Invalid order |
| -3 | Invalid Operators |
| -4 | Invalid arguments |
| -5 | The operation is not allowed |

## 5.4 Command List

| AT | Instruction |
|---|---|
| CANFLT | Query/set CAN filter information |
| CAN | Query/set CAN transmit parameter information |
| EXAT | Exit AT command |
| E | Query/set AT command display |
| MODBUSID | Query/set device MODBUS address |
| MODE | Query/set device operating mode |
| MID | Query device information |
| RESTORE | Restore device parameters to factory settings instruction |
| REBT | Device reset command |
| UARTPKT | Query/set device serial port sub-packet parameters |
| UART | Query/set device serial port parameters |
| VER | Query device version information |

## 5.5 Instruction Details

### 5.5.1 AT test command

Function: Test command

Format: Set

Send：AT<CR>

Return: <CR><LF>+OK<CR><LF>

Example: AT<CR>

### 5.5.2 AT+CANFLT Querying/Setting CAN Filter Information

Function: Query/set CAN filter information



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Format: Query

Send: AT+CANFLT<CR>

Return: <CR><LF>+OK=<id,mask id,mode><CR><LF>

Setting

Send: AT+ CANFLT =<id,mask id,mode><CR>

Return: <CR><LF>+OK<CR><LF>

Parameters:

id: filter id, used in conjunction with filter mask id, this parameter only takes effect in user-defined filter mode.

mask id: filter mask id, used in conjunction with id, this parameter only takes effect in user-defined mode filtering.

mode: Filtering mode There are 8 modes:

OFF Receive all types of

EDTF Receive only extended data frames

ERTF: Extended Remote Frames only

NRTF Receives only standard remote frames

NDTF Receives only standard data frames

ETF Receives extended frames only

NTF Standard frames only

UARF User-defined filtering mode (i.e. standard CAN identifier mask bit mode, please refer to CAN related information to check the use, need to set the identifier match value (i.e. id in the parameter), and mask code (i.e. mask id in the parameter))

Example: AT+CANFLT=0,0,OFF <CR>.

## 5.5.3 AT+CAN Querying/Setting CAN Transmission Parameter Information

Function: To query/set CAN transmit parameter information.

Format: Query

Send：AT+CAN<CR>

Return: <CR><LF>+OK=<baud,id,mode><CR><LF>

Setting

Send: AT+ CANFLT =<baud,id,mode><CR>

Return: <CR><LF>+OK<CR><LF>

Parameters:

baud: CAN baud rate Unit: kbps Range: 6, 10, 20, 50, 100, 120, 125, 150, 200, 250, 400, 500, 600, 750, 1000

id: Send frame ID (identifier) NDTF mode: 0-7FF EDTF mode: 0-1FFFFFFF

mode: Transmission mode 2 modes: NDTF Standard data frame transmission

EDTF Extended data frame

Example: AT+CAN=100,70,NDTF <CR

## 5.5.4 AT+EXAT Exit AT Command

Function: Exit AT command

Format: Set



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Send: AT+EXAT<CR>

Return: <CR><LF>+OK<CR><LF>

Example: AT+EXAT<CR>

## 5.5.5 AT+E Query/Set Command Display Mode

Function: To query/set the command recall mode.

Format: Query

Send: AT+E<CR>

Return: <CR><LF>+OK=<sw><CR><LF>

Setting

Send: AT+E=<sw><CR>

Return: <CR><LF>+OK<CR><LF>

Parameters:

sw AT command showback switch Total two states:

ON Turns on the display, displaying back the commands entered under AT command.

OFF Turns off the display, the input command is not displayed in AT command mode.

Example: AT+E=ON<CR>

## 5.5.6 AT+MODBUSID Query/set device MODBUS address

Function: Query/set device MODBUS address.

Format: Query

Send：AT+ AT+MODBUSID <CR>

Return: <CR><LF>+OK=<id><CR><LF>

Setting

Send: AT+ AT+MODBUSID =<id><CR>

Return: <CR><LF>+OK<CR><LF>

Parameters:

id: MODBUS ID Range: 0-255

Example: AT+MODBUSID=9 <CR>

## 5.5.7 AT+MODE Querying/Setting the Device Operating Mode

Function: To query/set the working mode of the device.

Format: Query

Send：AT+MODE <CR>

Return: <CR><LF>+OK=<mode><CR><LF>

Setting

Send: AT+MODE =<mode><CR>

Return: <CR><LF>+OK<CR><LF>



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameter:

mode: device working mode There are 4 modes:

TRANS Transparent transmission mode

TPRTL Transparent with information mode

MODBUS MODBUS mode

PROTOL protocol mode

Example: AT+MODE=TRANS <CR

## 5.5.8 AT+MID Query Device Name

Function: Queries the device name.

Format: Query

Send: AT+MID<CR>

Return: <CR><LF>+OK=<name><CR><LF>

Parameter: AT+MID<CR>:

name Device name

Example: AT+MID=E810-TTL<CR>

## 5.5.9 AT+RESTORE Restore Factory Settings

Function: Restore Factory Settings

Format: Setting

Send: AT+RESTORE<CR>

Return: <CR><LF>+OK<CR><LF>

Parameters: None

Example：AT+ RESTORE <CR>

<Note>: After this command is executed correctly, please reboot the device with AT+REBT command.

## 5.5.10 AT+REBT Reboot Device

Function: Reboot the device

Format: Setting

Send: AT+REBT<CR>

Return: <CR><LF>+OK<CR><LF>

Parameter: None

Example：AT+ REBT <CR>

<Note>: After this command is executed correctly, the device reboots and will exit AT mode.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 5.5.11 AT+UARTPKT Query/Set Serial Port Packet Information

Function: Query/Set the serial port sub-packet information.

Format: Query

Send: AT+UARTPKT<CR>

Return: <CR><LF>+OK=<size,time><CR><LF>

Setting

Send: AT+ UARTPKT=<size,time><CR>

Return: <CR><LF>+OK<CR><LF>

Parameters:

size Packing length, 0, 4-1000 bytes, 0 is off, both 0 is default packing parameter.

time Packaging time, 0, 10~1000 milliseconds, 0 is off, 0 is default.

Example: AT+UARTPKT=1000,10<CR

## 5.5.12 AT+UART Querying/Setting Serial Port Parameters

Function: Query/Set serial port parameters.

Format: Query

Send: AT+UART<CR>

Return: <CR><LF>+OK=<baud,data,stop,parity,flowctrl><CR><LF>

Setting

Send: AT+UART=<baud,data,stop,parity,flowctrl><CR>

Return: <CR><LF>+OK<CR><LF>

Parameters:

baud Baud rate, support any baud rate from 300 to 115200.

data Data 8 bits

stop Stop 1 or 2 bits

parity Checksum bits ODD (odd parity), EVEN (even parity), NONE (no parity)

flowctrl flow control bit NFC (no hardware flow control)

## 5.5.13 AT+VER Querying Device Version Information

Function: Query device version information

Format: Query

Send：AT+VER<CR>

Return: <CR><LF>+OK=<ver><CR><LF>

Parameters:

ver Device version number

Example: AT+VER<CR>

copyright@2025，Chengdu Ebyte Electronic Technology Co.,Ltd.


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# 6 FAQ

## 6.1 Equipment is fragile

- Please check the power supply to ensure that it is between the recommended supply voltages, exceeding the maximum value can cause permanent damage to the device.
- Please check the stability of the power supply, the voltage should not fluctuate significantly and frequently.
- Please ensure that the installation and use of the process of anti-static operation.
- Please ensure that the humidity during installation and use should not be too high, some components are humidity-sensitive devices.
- If there is no special demand, it is not recommended to use in too high or too low temperature.

## 6.2 Unable to set up a successful command

- Please check whether the instruction format is correct .
- Please check the indicator blinking frequency, the indicator blinking frequency is 5Hz in the command configuration state .
- Please check the mode used by the instruction.

## 6.3 Failure to use after parameter change

- Please check whether the serial port baud rate and CAN baud rate are set correctly.
- Please check whether the data format is in accordance with the regulations.
- Please check whether the computer serial port is dead.
- Please read the instruction carefully.

# Revision history

| Version | Revision date | Description | Issued by |
|---|---|---|---|
| 1.2 | 2025-05-22 | Revision | Blue |

copyright@2025，Chengdu Ebyte Electronic Technology Co.,Ltd.


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# About us

Technical support: service@cdebyte.com

Documents and RF Setting download link: www.cdebyte.com

Thank you for using Ebyte products! Please contact us with any questions or suggestions: info@cdebyte.com

---

Web: www.cdebyte.com

Address: B5 Mould Park, 199# Xiqu Ave, High-tech District, Sichuan, China

Chengdu Ebyte Electronic Technology Co.,Ltd.

copyright@2025，Chengdu Ebyte Electronic Technology Co.,Ltd.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000