



EBYTE

成都亿佰特电子科技有限公司

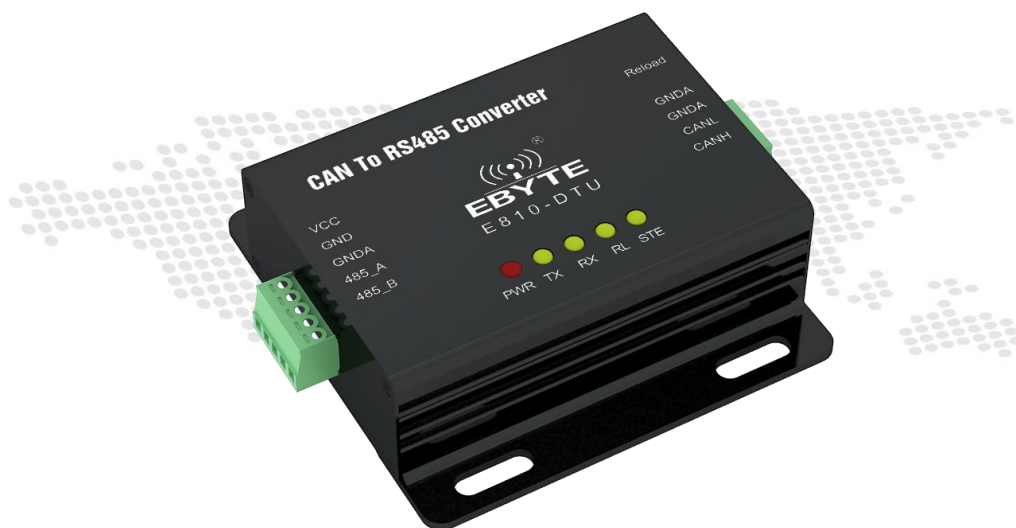
Chengdu Ebyte Electronic Technology Co.,Ltd.

Wireless Modem

用户使用手册

E810-DTU (CAN-RS485) 产品规格书

E810-DTU (CAN-RS485) 智能协议转换器



本说明书可能会随着产品的改进而更新，请以最新版的说明书为准
成都亿佰特电子科技有限公司保留对本说明中所有内容的最终解释权及修改权

目录

第一章 概述	4
1.1 简介	4
1.2 功能特点	4
1.3 应用场景	4
第二章 产品规格参数及特性	5
2.1 基本参数	5
2.2 出厂默认参数	5
第三章 硬件参数设计介绍	6
3.1 设计介绍	6
3.2 尺寸图	8
3.3 连接方法	9
3.3.1 RS485连接方法	9
3.3.2 CAN连接方法	10
第四章 模式说明	10
4.1 工作模式	10
4.2 数据转换方式	10
4.2.1 透明转换模式	11
4.2.2 透传带信息模式	14
4.2.3 协议模式	16
4.2.4 Modbus模式	18
第五章 操作指令	21
5.1 进入指令配置说明	21
5.2 指令概述	21
5.3 指令错误码	21

5.4 指令列表	22
5.5 指令详情	22
5.5.1 AT 测试指令	22
5.5.2 AT+CANFLT 查询/设置CAN过滤器信息	22
5.5.3 AT+CAN 查询/设置CAN发送参数信息	23
5.5.4 AT+EXAT 退出AT指令	23
5.5.5 AT+E 查询/设置指令回显模式	23
5.5.6 AT+MODBUSID 查询/设置设备MODBUS地址	24
5.5.7 AT+MODE 查询/设置设备工作模式	24
5.5.8 AT+MID 查询设备名称	24
5.5.9 AT+RESTORE 恢复出厂设置	25
5.5.10 AT+REBT 重启设备	25
5.5.11 AT+UARTPKT 查询/设置串口分包信息	25
5.5.12 AT+UART 查询/设置串口参数	25
5.5.13 AT+VER 查询设备版本信息	26
第六章 常见问题	26
6.1 设备易损坏	26
6.2 无法成功指令设置	26
6.3 参数更改后使用失败	26
修订历史	27
关于我们	27

第一章 概述

1.1 简介

E810-DTU (CAN-RS485) 是成都亿佰特电子科技有限公司自主研发的一款小型智能协议转换产品。产品使用 8V~28V 宽电压供电，集成 1 路 CAN-BUS 接口和 1 路 RS485 接口，可实现 RS485、CAN 两种不同协议数据之间的双向转换。该产品支持串口指令配置设备参数和工作模式，支持透明转换、透明带信息转换、协议转换、Modbus RTU 转换等四种数据转换模式。同时，E810-DTU (CAN-RS485) 智能协议转换器具有体积小、方便安装等特点，在 CAN-BUS 类产品开发、数据分析应用中具有极高的性价比，是工程应用、项目调试及产品开发的可靠助手。



1.2 特点功能

- 支持RS485与CAN之间数据双向转换。
- 支持透明转换、透明带信息转换和协议转换。
- 支持Modbus RTU 协议转换。
- 支持两种CAN标识符发送方式，可通过固定式配置或串行帧数据指定。
- 支持RS485接口参数配置。
- 支持 1200~115200 范围内参数RS485波特率设置。
- 支持AT指令参数配置。
- 支持AT指令和硬件恢复出厂设置。
- 拥有电源指示灯，状态指示灯等多种状态指示。

1.3 应用场景

- 工业控制等CAN-BUS网络
- 汽车、铁路设备联网
- 安防、消防网络
- 地下远程通讯
- 公共广播系统
- 停车场设备控制
- 智能家居、智能楼宇

第二章 产品规格参数及特性

2.1 基本参数

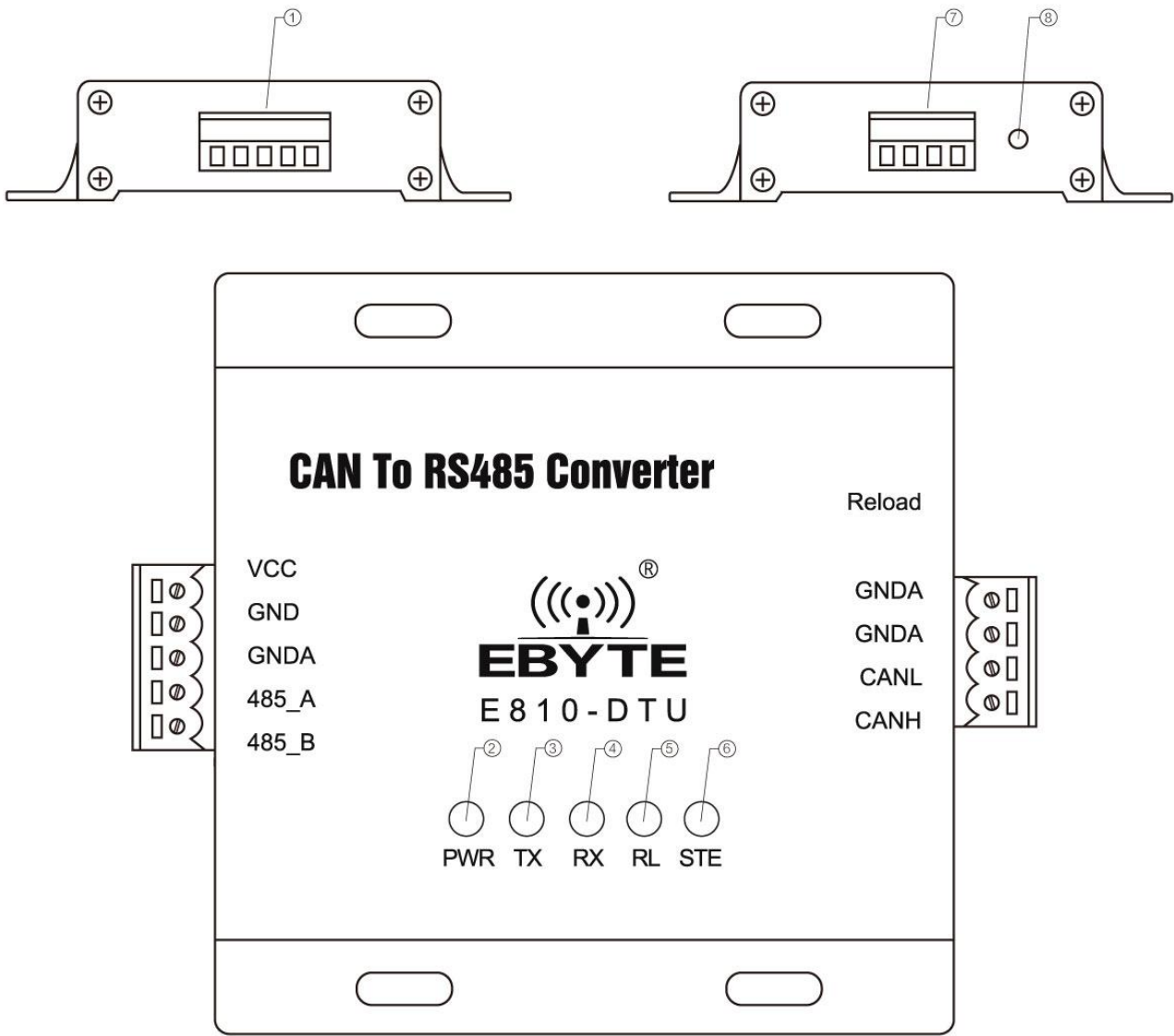
主要参数	规格
电源电压	8V~28V，推荐12V或者24V供电（超过28V可能会永久损坏设备）
工作电流	38.2762mA@12V（RS485）
工作温度	-40℃~85℃，工业级
接口类型	RS485：1*5*3.81，压线方式；CAN：1*4*3.81，压线方式
通信电平	3.3V，若接5V，需电平转换
工作湿度	10%~90%，相对湿度，无冷凝

2.2 出厂默认参数

RS485	串口波特率	115200 bps
	奇偶校验	无
	数据位	8
	停止位	1
	流控	关
CAN	CAN波特率	100K bps
	CAN ID号	0x00000000
	流控	关
默认工作模	透传模式	接收所有数据类型
默认设备地	Modbus设备地址	默认地址为1
帧分包参数	时间	10ms
	字节数	1000byte

第三章 硬件参数设计介绍

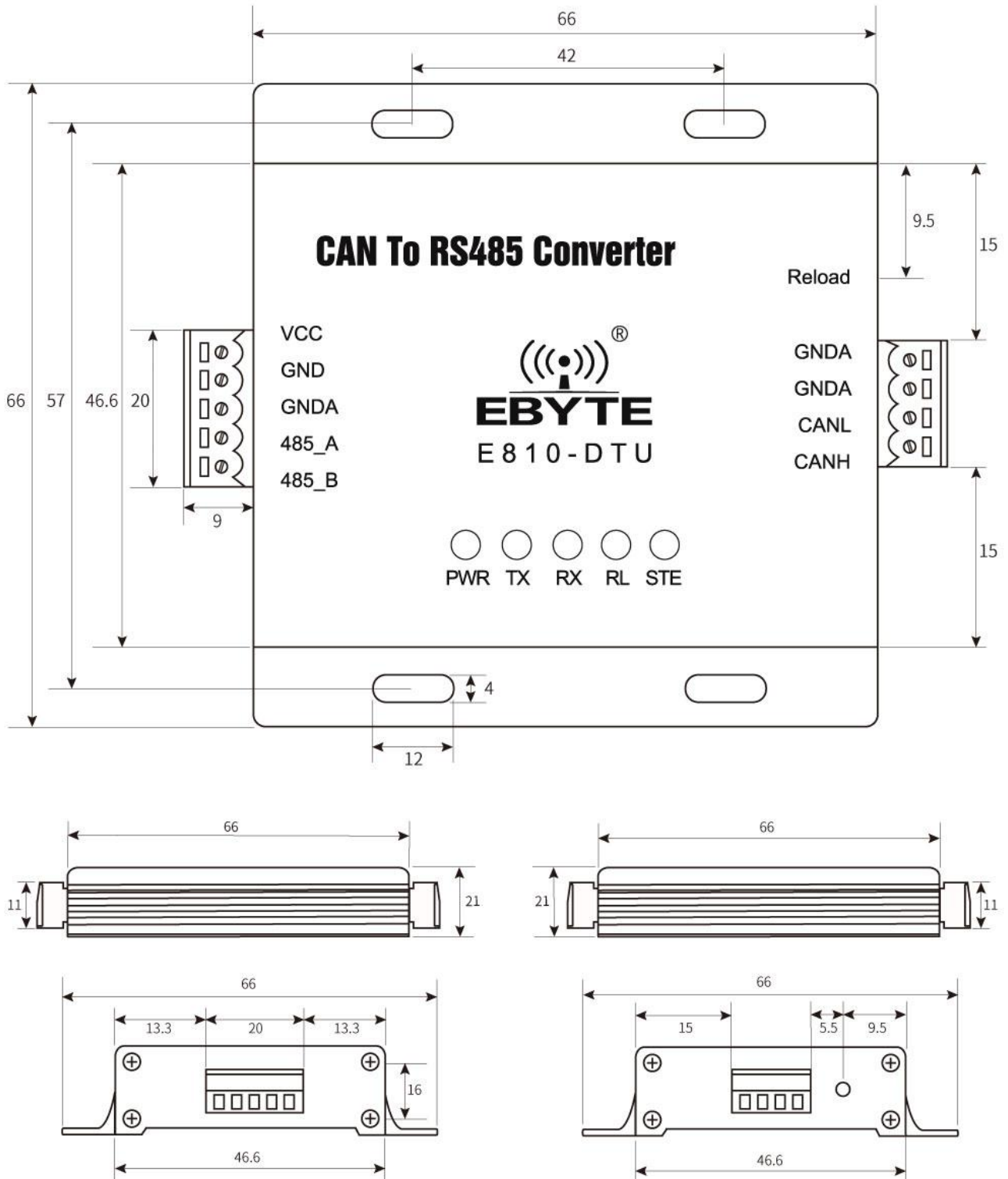
3.1 设计介绍



引脚序号	引脚名称	引脚用途
1	VCC	电源，默认8-28V（可定制标准5V版本），建议12V/24V
2	GND	电源地
3	GNDA	RS485公共端，外接其它设备RS485设备的GND
4	485_A	RS485总线数据A
5	485_B	RS485总线数据B
6	PWR	电源指示灯
7	TX	串口发送指示灯

8	RX	串口接收指示灯
9	RL	恢复出厂设置指示灯，长按5-10S设备恢复出厂设置
10	STE	状态指示灯
11	GNDA	信号参考地
12	GNDA	信号参考地
13	CAN-H can总线H	CAN通信接口
14	CAN-L can总线L	CAN通信接口

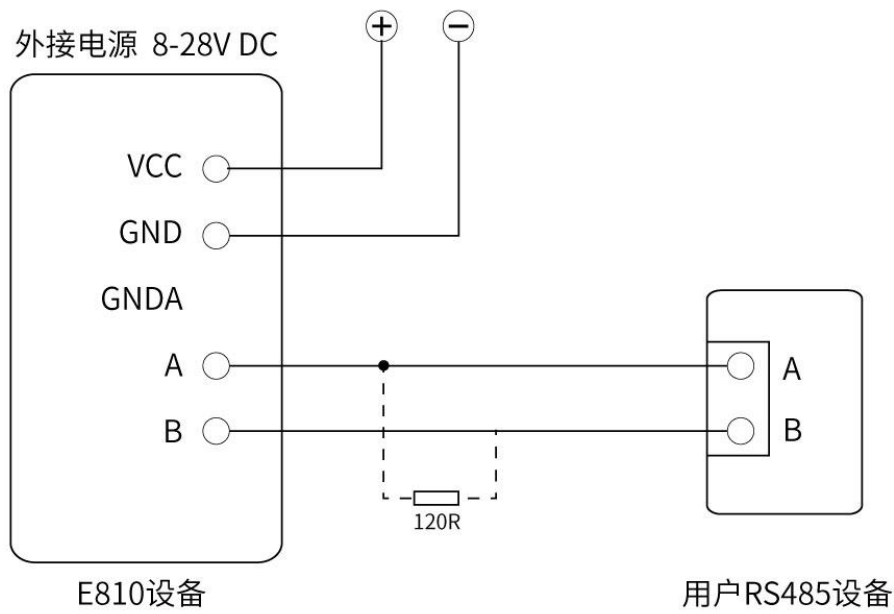
3.2 尺寸图



3.3 连接方法

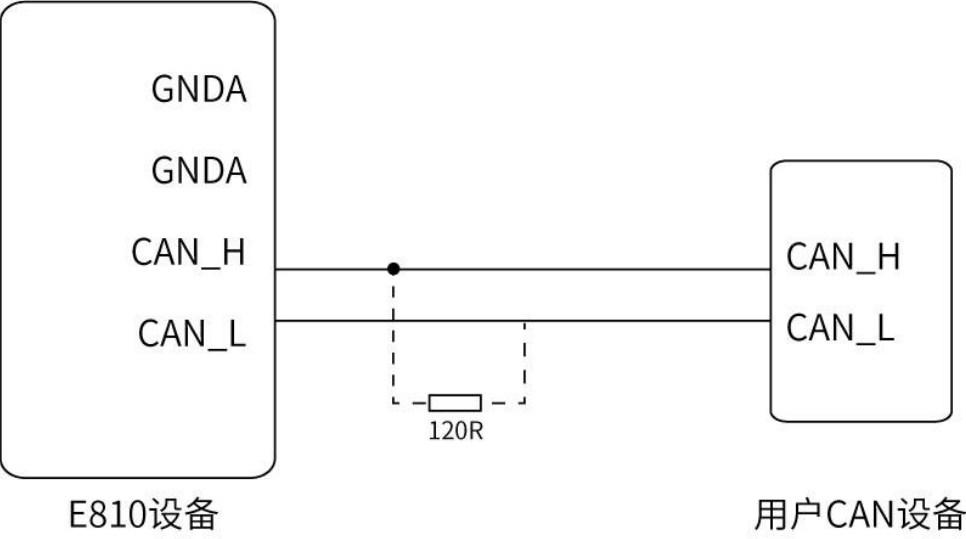
3.3.1 RS485连接方法

RS485接线图



3.3.2 CAN连接方法

CAN总线连线图



第四章 模式说明

4.1 工作模式

E810-DTU (CAN-RS485) 设备有两种工作模式，正常模式和配置模式。

模式	功能
正常模式	设备出厂的通用模式，上电即正常工作。
配置模式	设备可以进行配置的模式，进入配置模式方法详情见第五章。

4.2 数据转换方式

E810-DTU (CAN-RS485) 设备支持四种数据转换方式：透明转换、透明带信息转换、协议转换和 MODBUS 转换。支持 CAN 与 RS485 的双向转换。

数据转换方式	转换方向
透明转换	CAN 与 RS485 双向转换
透明带信息转换	CAN 与 RS485 双向转换
协议转换	CAN 与 RS485 双向转换
MODBUS 转换	CAN 与 RS485 双向转换

4.2.1 透明转换模式

透明转换：转换器将一种格式的总线数据原样转换成另一种总线的数据格式，不附加数据也不对数据做修改。这样既实现了数据格式的交换又没有改变数据内容，对于两端的总线来说转换器如同“透明”的一样，故为透明转换。

E810-DTU(CAN-RS485) 设备可以将 CAN 总线收到的有效数据原封不动的转换到 RS485 总线输出。同样的，设备也可以将 RS485 总线收到的有效数据原封不动的转换到 CAN 总线输出。实现 RS485 与 CAN 的透明转换。

1. 串行帧转 CAN 报文

串行帧的全部数据依序填充到 CAN 报文帧的数据场里。转换器一检测到串行总线上有数据后就立即接收并转换。转换成的 CAN 报文的帧类型和帧 ID 来自用户事先的配置，并且在转换过程中帧类型和帧 ID 一直保持不变。数据转换对应格式如图 4.1 所示。

如果收到的串行帧长度小于等于 8 字节，依序将字符 1 到 n（n 为串行帧长度）填充到 CAN 报文的数据域的 1 到 n 个字节位置（如图 4.1 中 n 为 7）。如果串行帧的字节数大于 8，那么处理器从串行帧首个字符开始，第一次取 8 个字符依次填充到 CAN 报文的数据域。将数据发至 CAN 总线后，再转换余下的串行帧数据填充到 CAN 报文的数据域，直到其数据被转换完。

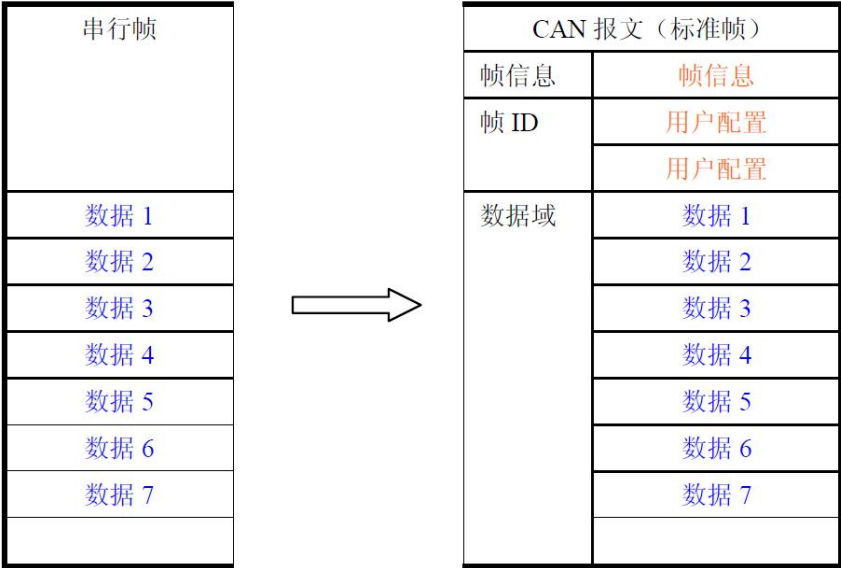


图 4.1 串行帧转换成 CAN 报文（透明转换）

转换示例：

假设置配的转换成 CAN 报文帧信息为“标准帧”，帧 ID（ID1，ID2）设置为 0060，那么转换格式如图 4.2 所示。

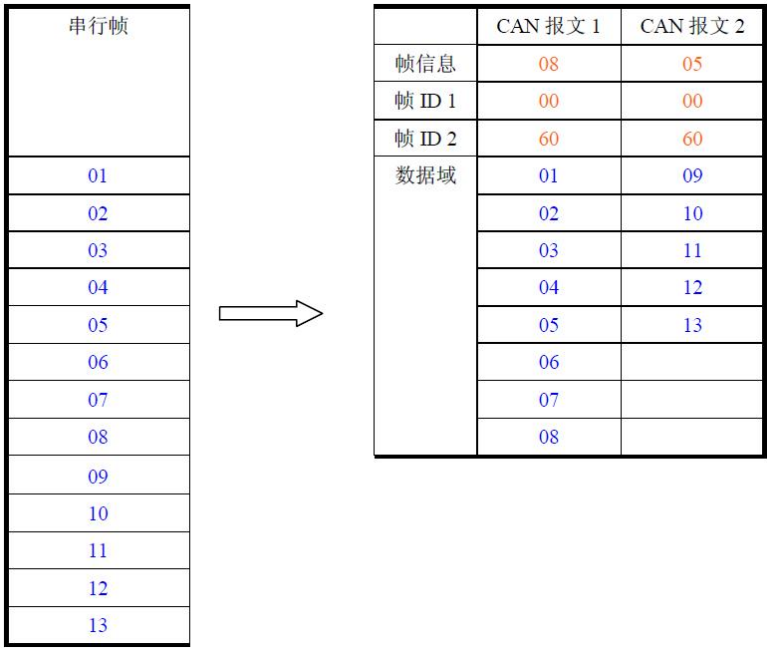


图 4.2 串行帧转 CAN 报文示例（透明转换）

2. CAN 报文转串行帧

CAN 报文帧的数据场里全部数据依序填充到串行帧数据里面。转换器一检测到 CAN 总线上有数据后就立即接收并转换。

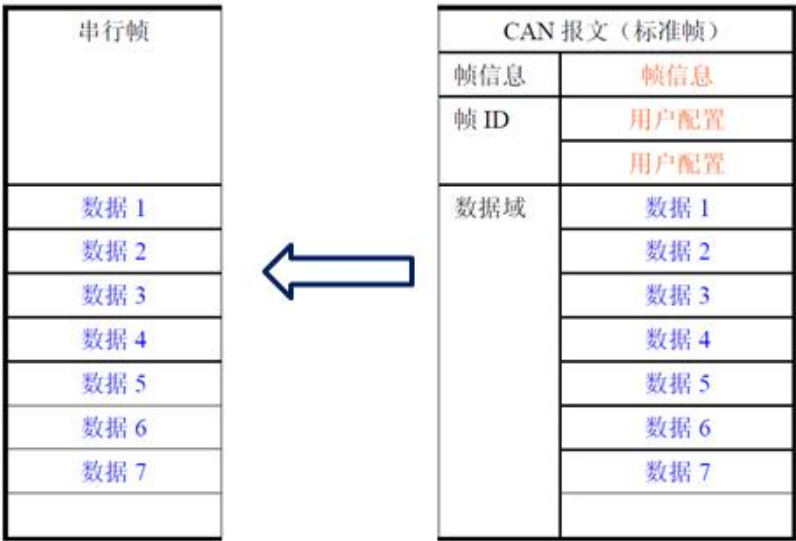


图 4.3 CAN 报文转换成串行帧（透明转换）

转换示例：

假设 CAN 报文帧信息为“标准帧”，帧 ID（ID1，ID2）设置为 0060，那么转换格式如图 4.4 所示。



图 4.4 CAN 报文转串行帧示例（透明转换）

4.2.2 透传带信息模式

透传带信息转换是透明转换的一种特殊的用法，也不附加协议。这种转换方式是根据通常的串行帧和 CAN 报文的共有特性，使这两种不同的总线类型也能轻松的组建同一个通信网络。

该方式下，CAN 总线接收方能将收到的 CAN 报文的帧信息与帧 ID，添加到转换后的串行帧中，在这种方式下，接收方能够清晰看到发送方的 CAN 报文信息，从而可以根据这些信息作出更灵活的运用。

转换方式：

1. 串行帧转 CAN 报文

同透明转换，串行帧的全部数据依序填充到 CAN 报文帧的数据场里。转换器一检测到串行总线上有数据后就立即接收并转换。转换成的 CAN 报文的帧类型和帧 ID 来自用户事先的配置，并且在转换过程中帧类型和帧 ID 一直保持不变。数据转换对应格式如图 4.1 所示。

如果收到的串行帧长度小于等于 8 字节，依序将字符 1 到 n（n 为串行帧长度）填充到 CAN 报文的数据域的 1 到 n 个字节位置（如图 4.5 中 n 为 8）。

如果串行帧的字节数大于 8，那么处理器从串行帧首个字符开始，第一次取 8 个字符依次填充到 CAN 报文的数据域。将数据发至 CAN 总线后，再转换余下的串行帧数据填充到 CAN 报文的数据域，直到其数据被转换完。



图 4.5 串行帧转换成 CAN 报文（透传带信息）

转换示例：

假设置置的转换成 CAN 报文的帧信息为“标准帧”，帧 ID（ID1，ID2）设置为 0060，那么转换格式如图 4.6 所示。

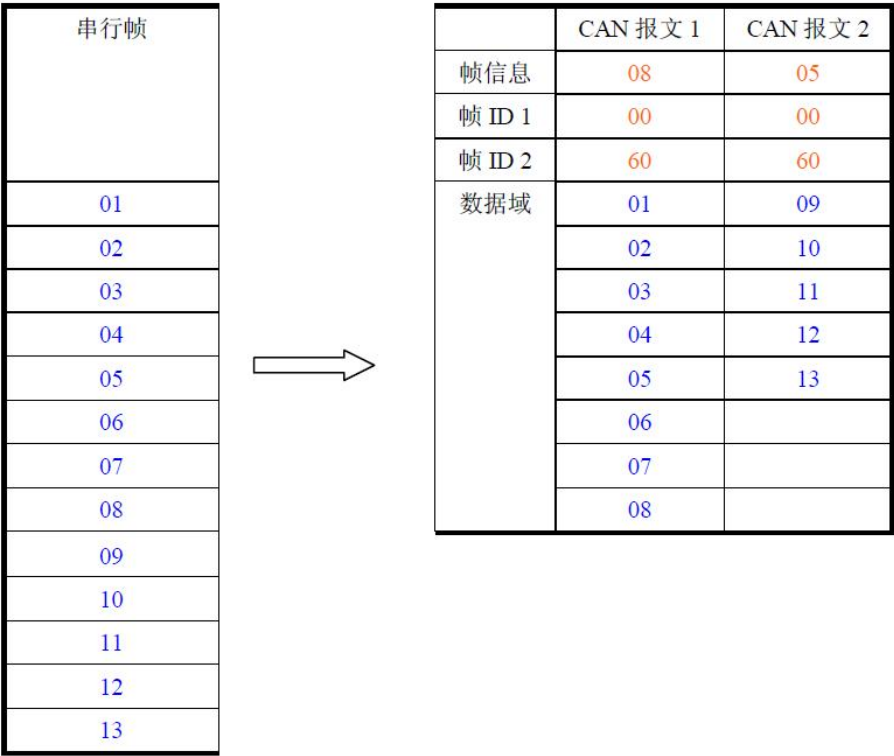


图 4.6 串行帧转 CAN 报文示例（透传带信息转换）

2. CAN 报文转串行帧

转换器一检测到 CAN 总线上有数据后就立即接收并转换，收到一帧 CAN 报文就立即转换一帧，每次转换的时候把 CAN 帧信息与帧 ID 添加在串行帧中（此转换与下面协议模式的 CAN 报文转串行帧一致，详情请看协议模式）。如下图所示

请注意：无论是串行帧还是 CAN 报文在应用的时候其帧格式（标准帧还是扩展帧）应该符合事先配置的帧格式要求，否则可能导致通讯不正常。



图 4.7 CAN 报文转串行帧（透明带信息转换）

CAN 发送：
 帧格式：扩展帧
 帧类型：数据帧
 ID：0x12345678
 数据：AAh BBh CCh DDh EEh
 串行帧接收：85 12 34 56 78 AA BB CC DD EE 00 00 00
 0x85 表示帧格式为扩展帧，帧类型为数据帧，数据长度为 5
 后四位表示 CAN ID 为 12345678
 最后 8 位为数据区，有效长度为 5，其余位补齐 0

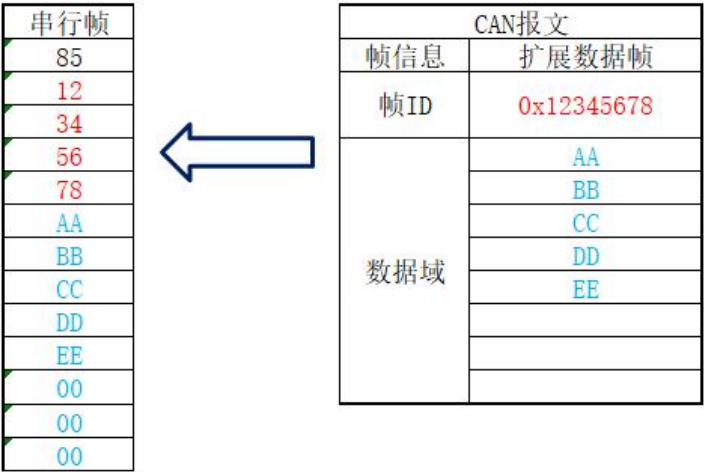
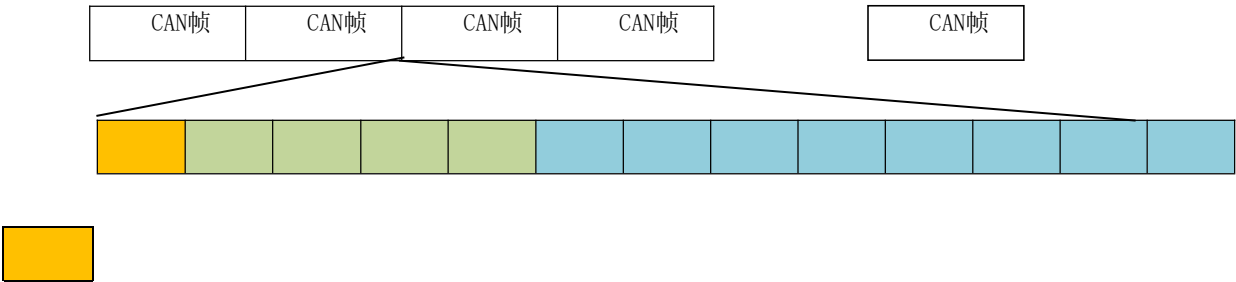


图 4.8 CAN 报文转串行帧示例（透明带信息转换）

4.2.3 协议模式

E810-DTU(CAN-RS485) 设备数据转换格式如下所示，每一个 CAN 帧包含 13 个字节，13 个字节的内容包括 CAN 帧信息+帧 ID+帧数据。



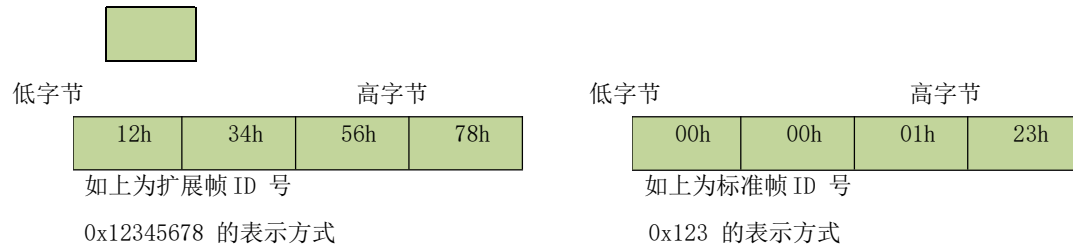
帧信息：长度 1 个字节，用于标识 CAN 帧的一些信息，如类型、长度等。



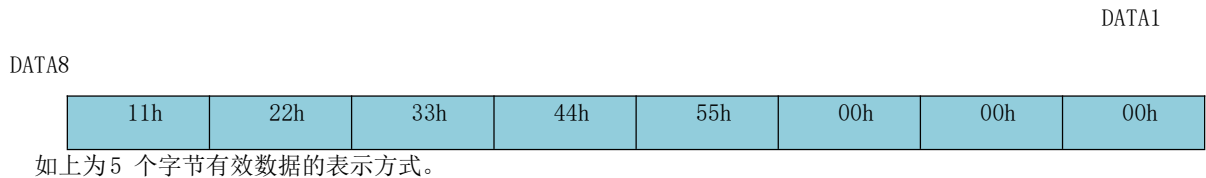
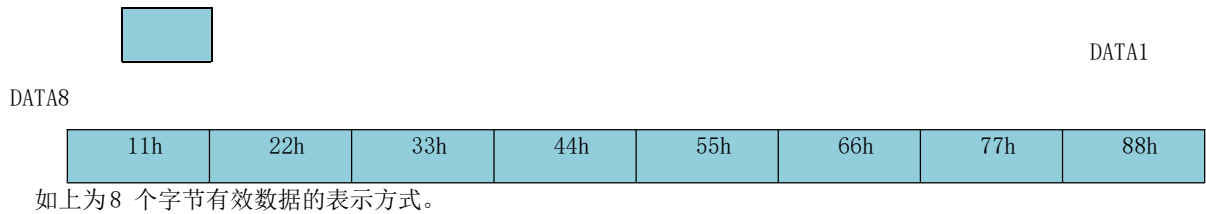
FF：标准帧和扩展帧的标识位，1 为扩展帧，0 为标准帧。
RTR：远程帧和数据帧的标识位，1 为远程帧，0 为数据帧。
保留：保留值为 0，不可写入 1。

D3~D0: 数据长度位，标识该 CAN 帧的数据长度。

帧 ID: 长度 4 个字节，标准帧有效位 11 位，扩展帧有效位 29 位。



帧数据: 长度 8 个字节，有效长度由帧信息的 D3~D0 的值决定。

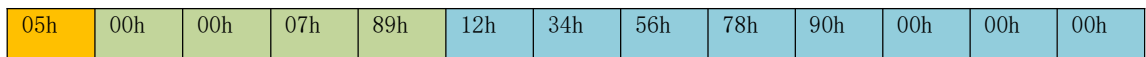


举例说明:

以下例子是一个扩展数据帧，帧 ID 为 0x11223344，包含 8 个字节有效数据（11h, 22h, 33h, 44h, 55h, 66h, 77h, 88h）的表示方式。



以下例子是一个标准数据帧，帧 ID 为 0x789，包含 5 个字节有效数据（12h, 34h, 56h, 78h, 90h）的表示方式。



请注意: 每一帧固定是 13 个字节，不足的必须补 0，否则将导致通信错误。

例:

1. CAN 到串行帧:

CAN 发送:

帧格式: 扩展帧

帧类型: 数据帧

ID : 0x12345678

数据 : AAh BBh CCh DDh EEh

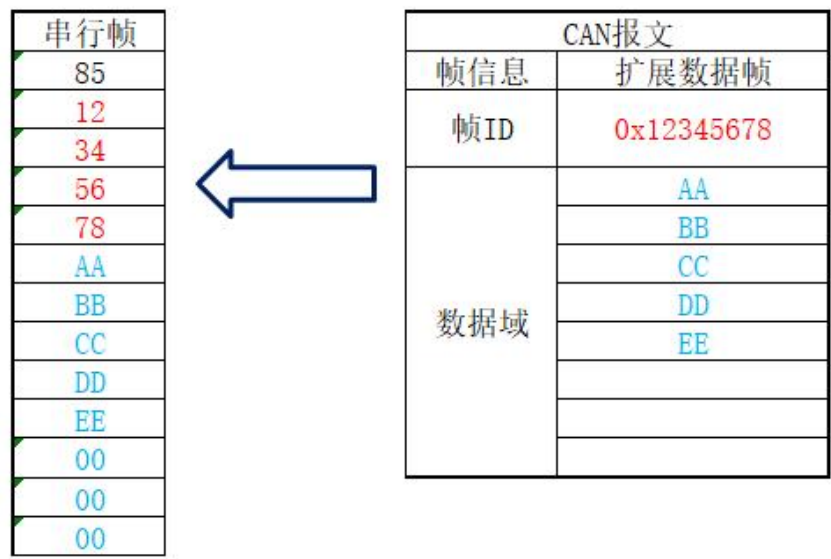
串行帧接收: 85 12 34 56 78 AA BB CC DD EE 00 00 00

0x85 表示帧格式为扩展帧，帧类型为数据帧，数据长度为 5

后四位表示 CAN ID 为 12345678

最后 8 位为数据区，有效长度为 5，其余位补齐 0

如下图所示:



2. 串行帧到 CAN:

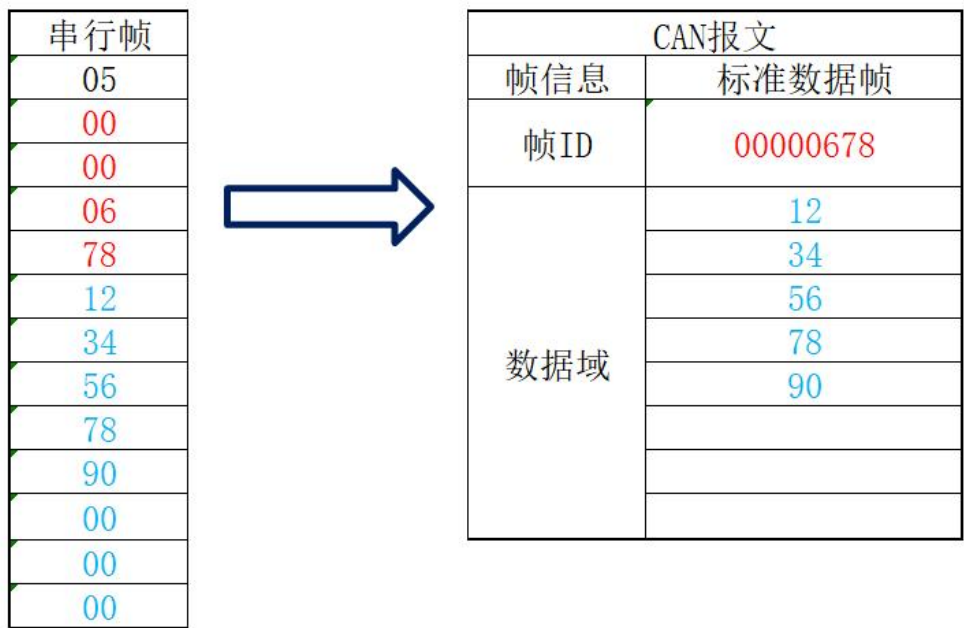
串行帧发送: 05 00 00 06 78 12 34 56 78 90 00 00 00

0x05 表示帧格式为标准帧, 帧类型为数据帧, 数据长度为 5

00 00 06 78 表示 ID 为 0678

12 34 56 78 90 00 00 00 为数据区, 有效长度为 5

如下图所示:



4.2.4 Modbus模式

Modbus 转换模式, 支持 RTU 转换模式。E810-DTU (CAN-RS485) 设备接收并响应 (通过 UART) 发送过来的命令。

E810-DTU (CAN-RS485) 设备支持两种 Modbus 命令:

读取寄存器（功能码 03）、写多个寄存器（功能码 16）。

转换设备内部有一缓存器用于缓存接收到的 CAN 帧数据，缓存器按照地址 0~63 总共有 64 级缓存。缓存地址从 0 开始，到地址 63，可连续缓存 8 帧 CAN 数据（每帧数据 8 字节，共 64 字节）。当接收到第一帧 CAN 数据时，该 CAN 帧数据存储在地址 0 中，以后接收到的 CAN 帧数据按照地址逐级递增顺序存放。如果 64 级缓存都存储满了，则新接收到的 CAN 帧数据将存放到地址 0 中并覆盖原来的数，遵循先进先出。

读取寄存器（功能码 03）：

发送命令：

[设备地址] [命令号 03(0x03)] [起始寄存器地址高 8 位] [低 8 位] [读取的寄存器数高 8 位] [低 8 位] [CRC 校验的高 8 位] [CRC 校验的低 8 位]

读取格式只允许是从地址 00 00 开始 读取 00 08 （一次性读取 8 个字节即一帧数据，即读取了 00 00 - 00 07 地址的数据），读取成功后，该 8 个字节数据会被清空，它地址之后的数据会往前移 8 个数据。

比如：

设备在 MODBUS 模式下，CAN 总线收到了 4 帧数据：

第一帧： 0x01 0x02 0x03 0x04 共 4 字节数据

第二帧： 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 共 6 字节数据

第三帧： 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 共 8 字节数据

第四帧： 0xAA 0xBB 0xCC 共 3 字节数据

它们在 MODBUS 缓存地址中是存放的情况：

（没有数据的地址为 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	0xEE	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

用指令： 01 03 00 00 00 08 （在地址为 01 的从设备中，读取从地址 0000 开始的 8 条数据）将此指令写到专用工具（Modbus CRC 16 计算器），计算出 CRC 校验值并添加到指令后面。通过 UART 发出该指令： 01 03 00 00 00 08 44 0C ，从机接收到该指令后，会返回从机转换设备内部的缓存值。（当没有接收到新的 CAN 帧数据时，主机读到的缓存值则全为 0）

返回指令：

[设备地址] [命令号 03] [返回的字节个数] [数据 1] [数据 2]...[数据 n] [CRC 校验的高 8 位] [CRC 校验的低 8 位]

如：

从机收到指令 01 03 00 00 00 08 44 0C, , 则返回：

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

读取完成后，该 8 个字节数据被清空，它地址之后的数据往前移 8 个数据。如下图所示：

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

由此可进行下一次读取寄存器命令

写多个寄存器（功能码 16）：成功写入时，会将写入的数据发到 CAN 总线。

发送命令：

[设备地址] [命令号 16 (0x10)] [需下置的寄存器地址高 8 位] [低 8 位] [数据数量高 8 位] [数据数量低 8 位] [下置的数据高 8 位] [低 8 位] [.....] [.....] [CRC 校验的高 8 位] [CRC 校验的低 8 位]

如：

通过写命令写 5 字节数据发送出去： 01 10 00 00 00 05 0A 00 11 00 22 00 33 00 44 00 55, 将此指令写到专用工具 (Modbus CRC 16 计算器)，计算出 CRC 校验值并添加到指令后面：01 10 00 00 00 05 0A 00 11 00 22 00 33 00 44 00 55 47 84, 以 HEX 格式发送，即可写入 5 字节数据，成功写入时，会将写入的数据发到 CAN 总线，即此时 CAN 总线会发送 5 个字节的数据：11, 22, 33, 44, 55。CAN 报文中，帧信息，帧 ID 为用户事先配置好的。

注意：此命令的寄存器地址只允许是 00 00, 不允许发如：01 10 00 01 00 05 0A 00 11 00 22 00 33 00 44 00 55 这样的指令格式。

且最多一次性发 8 字节数据：

通过写命令写 8 字节数据发送出去：01 10 00 00 00 08 10 00 11 00 22 00 33 00 44 00 55 00 66 00 77 00 88，将此指令写到专用工具（Modbus CRC 16 计算器），计算出 CRC 校验值并添加到指令后面：01 10 00 00 00 08 10 00 11 00 22 00 33 00 44 00 55 00 66 00 77 00 88 FE D5，以 HEX 格式发送，即可写入 8 字节数据，成功写入时，会将写入的数据发到 CAN 总线，即此时 CAN 总线会发送 8 个字节的数据：0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88。CAN 报文中，帧信息，帧 ID 为用户事先配置好的。

不允许发如：01 10 00 00 00 09 12 00 11 00 22 00 33 00 44 00 55 00 66 00 77 00 88 00 99 这样的指令格式。

第五章 操作指令

5.1 进入指令配置说明

注：一般工作模式下模式指示灯以 1Hz 频率闪烁；当进入指令配置模式时，指示灯以 5Hz 频率闪烁，无论是软件进入还是硬件进入，配置的参数皆是在复位后生效。

从其他模式切换至指令模式的时序（软件进入）：

通过 UART 连续发送“+++”，设备收到“+++”后，3 秒计时超时开始启动，如果在超时时间内接收到任意的 AT 指令则成功切换到配置模式（进入后，设备指示灯闪烁频率变为 5Hz）。此时可以进行参数的配置了，参数列表及参数说明请看下面介绍。

参数配置成功后，通过 UART 发送指令“AT+ EXAT”，设备在接收到指令后，返回“+OK”同时退出配置模式回到配置前的模式。然后短接复位 RESET 引脚，参数生效。也可以在用 AT 指令参数配置成功后，用指令“AT+REBT”复位设备

5.2 指令概述

AT 指令是指，在命令模式下用户通过 UART 与设备进行命令传递的指令集，后面将详细讲解 AT 指令的使用格式。上电成功后切换到配置模式，可以通过 UART 对设备进行设置。

从透传模式切换至指令模式的时序：

串口设备给设备连续发送“+++”，设备收到“+++”后，3 秒计时超时开始启动，如果在超时时间内接收到任意的 AT 指令则成功切换到配置模式

最后，通过 UART 发送指令“AT+ EXAT”，设备在接收到指令后，返回“+OK”同时退出配置。然后短接复位 RESET 引脚，参数生效。也可以在用 AT 指令参数配置成功后，用指令“AT+REBT”复位设备。

注：配置参数后，一定是复位后参数才生效的。

说明：<CR>：ASCII 码 0x0D

<LF>：ASCII 码 0x0A

5.3 指令错误码

错误码	说明
-1	无效的指令格式
-2	无效的指令
-3	无效的操作符
-4	无效的参数
-5	操作不允许

5.4 指令列表

AT	测试指令
CANFLT	查询/设置CAN过滤器信息
CAN	查询/设置CAN发送参数信息
EXAT	退出AT指令
E	查询/设置AT指令回显
MODBUSID	查询/设置设备MODBUS地址
MODE	查询/设置设备工作模式
MID	查询设备设备信息
RESTORE	设备参数恢复出厂设置指令
REBT	设备复位指令
UARTPKT	查询/设置设备串口分包参数
UART	查询/设置设备串口参数
VER	查询设备版本信息

5.5 指令详情

5.5.1 AT 测试指令

功能：测试指令

格式：设置

发送：AT<CR>

返回：<CR><LF>+OK<CR><LF>

示例：AT<CR>

5.5.2 AT+CANFLT 查询/设置CAN过滤器信息

功能：查询/设置 CAN 过滤器信息

格式：查询

发送：AT+CANFLT<CR>

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

设置

发送：AT+ CANFLT =<id,mask id,mode><CR>

返回：<CR><LF>+OK<CR><LF>

参数：

id： 过滤 id,与过滤 mask id 配合使用，该参数只在用户自定义过滤模式生效

mask id： 过滤 mask id, 与 id 配合使用，该参数只在用户自定义模式过滤生效

mode： 过滤模式 共 8 种模式：

OFF 接收所有类型的

EDTF 只接收扩展数据帧

ERTF 只接收扩展远程帧
NRTF 只接收标准远程帧
NDTF 只接收标准数据帧
ETF 只接收扩展帧
NTF 只接收标准帧

UARF 用户自定义过滤模式（即标准 CAN 标识符屏蔽位模式，请参考 CAN 相关资料查阅使用，需自行设置标识符匹配值（即参数中的 id），与屏蔽码（即参数中的 mask id））

示例：AT+CANFLT=0,0,OFF <CR>

5.5.3 AT+CAN 查询/设置CAN发送参数信息

功能：查询/设置 CAN 发送参数信息。

格式：查询

发送：AT+CAN<CR>
返回：<CR><LF>+OK=<baud,id,mode><CR><LF>

设置

发送：AT+ CANFLT =<baud,id,mode><CR>
返回：<CR><LF>+OK<CR><LF>

参数：

baud: CAN 波特率 单位: kbps 范围: 6, 10, 20, 50, 100, 120, 125, 150, 200, 250, 400, 500, 600, 750, 1000

id: 发送帧 ID（标识符） NDTF 模式: 0-7FF EDTF 模式: 0-1FFFFFFF

mode: 发送模式 共 2 种模式: NDTF 发送标准数据帧
EDTF 发送扩展数据帧

示例：AT+CAN=100,70,NDTF <CR>

5.5.4 AT+EXAT 退出AT指令

功能：退出 AT 指令

格式：设置

发送：AT+EXAT<CR>
返回：<CR><LF>+OK<CR><LF>

示例：AT+EXAT<CR>

5.5.5 AT+E 查询/设置指令回显模式

功能：查询/设置指令回显模式。

格式：查询

发送：AT+E<CR>
返回：<CR><LF>+OK=<sw><CR><LF>

设置

发送：AT+E=<sw><CR>
返回：<CR><LF>+OK<CR><LF>

参数:

sw AT 指令回显开关 共两种状态:

ON 打开回显, 回显 AT 指令下输入的指令

OFF 关闭回显, AT 指令模式下, 输入指令不回显

示例: AT+E=ON<CR>

5.5.6 AT+MODBUSID 查询/设置设备MODBUS地址

功能: 查询/设置设备 MODBUS 地址。

格式: 查询

发送: AT+ AT+MODBUSID <CR>

返回: <CR><LF>+OK=<id><CR><LF>

设置

发送: AT+ AT+MODBUSID =<id><CR>

返回: <CR><LF>+OK<CR><LF>

参数:

id: MODBUS ID 范围: 0-255

示例: AT+MODBUSID=9 <CR>

5.5.7 AT+MODE 查询/设置设备工作模式

功能: 查询/设置设备工作模式。

格式: 查询

发送: AT+MODE <CR>

返回: <CR><LF>+OK=<mode><CR><LF>

设置

发送: AT+MODE =<mode><CR>

返回: <CR><LF>+OK<CR><LF>

参数:

mode: 设备工作模式 共 4 种模式:

TRANS 透明传输模式

TPRTL 透明带信息模式

MODBUS MODBUS 模式

PROTOL 协议模式

示例: AT+MODE=TRANS <CR>

5.5.8 AT+MID 查询设备名称

功能: 查询设备名称。

格式: 查询

发送: AT+MID<CR>

返回: <CR><LF>+OK=<name><CR><LF>

参数:

name 设备名称

示例: AT+MID=E810-TTL<CR>

5.5.9 AT+RESTORE 恢复出厂设置

功能: 恢复出厂设置

格式: 设置

发送: AT+RESTORE<CR>

返回: <CR><LF>+OK<CR><LF>

参数: 无

示例: AT+ RESTORE <CR>

<注意>: 该命令正确执行后, 请使用 AT+REBT 指令重启设备。

5.5.10 AT+REBT 重启设备

功能: 重启设备

格式: 设置

发送: AT+REBT<CR>

返回: <CR><LF>+OK<CR><LF>

参数: 无

示例: AT+ REBT <CR>

<注意>: 该命令正确执行后, 设备重新启动, 将退出 AT 模式。

5.5.11 AT+UARTPKT 查询/设置串口分包信息

功能: 查询/设置串口分包信息。

格式: 查询

发送: AT+UARTPKT<CR>

返回: <CR><LF>+OK=<size,time><CR><LF>

设置

发送: AT+ UARTPKT=<size,time><CR>

返回: <CR><LF>+OK<CR><LF>

参数:

size 打包长度, 0, 4-1000 字节, 0 为关闭, 两个都为 0 则是默认打包参数

time 打包时间, 0, 10~1000 毫秒, 0 为关闭, 两个都为 0 则是默认打包参数

示例: AT+UARTPKT=1000,10<CR>

5.5.12 AT+UART 查询/设置串口参数

功能: 查询/设置串口参数。

格式: 查询

发送: AT+UART<CR>

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

设置

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

返回：<CR><LF>+OK<CR><LF>

参数：

baud 波特率，支持 300-115200 任意波特率

data 数据位 8 位

stop 停止位 1、2 位

parity 校验位 ODD（奇校验）、EVEN（偶校验）、NONE（无校验）

flowctrl 流控位 NFC（无硬件流控）

5.5.13 AT+VER 查询设备版本信息

功能：查询设备版本信息

格式：查询

发送：AT+VER<CR>

返回：<CR><LF>+OK=<ver><CR><LF>

参数：

ver 设备版本号

示例：AT+VER<CR>

第六章 常见问题

6.1 设备易损坏

- 请检查供电电源，确保在推荐供电电压之间，如超过最大值会造成设备永久性损坏。
- 请检查电源稳定性，电压不能大幅频繁波动。
- 请确保安装使用过程防静电操作。
- 请确保安装使用过程湿度不宜过高，部分元件为湿度敏感器件。
- 如果没有特殊需求不建议在过高、过低温度下使用。

6.2 无法成功指令设置

- 请检查指令格式是否正确。
- 请检查指示灯闪烁频率，指令配置状态下，指示灯闪烁频率为5Hz。
- 请检查指令所使用的模式。

6.3 参数更改后使用失败

- 请检查串口波特率和CAN波特率是否设置正确。
- 请检查数据格式是否符合规定。

- 请检查电脑串口是否死机。
- 请仔细阅读指令使用说明。

修订历史

版本	修订日期	修订说明	维护人
1. 2	2025-05-06	版本修改	Blue

关于我们



销售热线：4000-330-990
技术支持：support@cdebyte.com 官方网站：www.ebyte.com
公司地址：四川省成都市高新西区西区大道199号B5栋

