



EBYTE

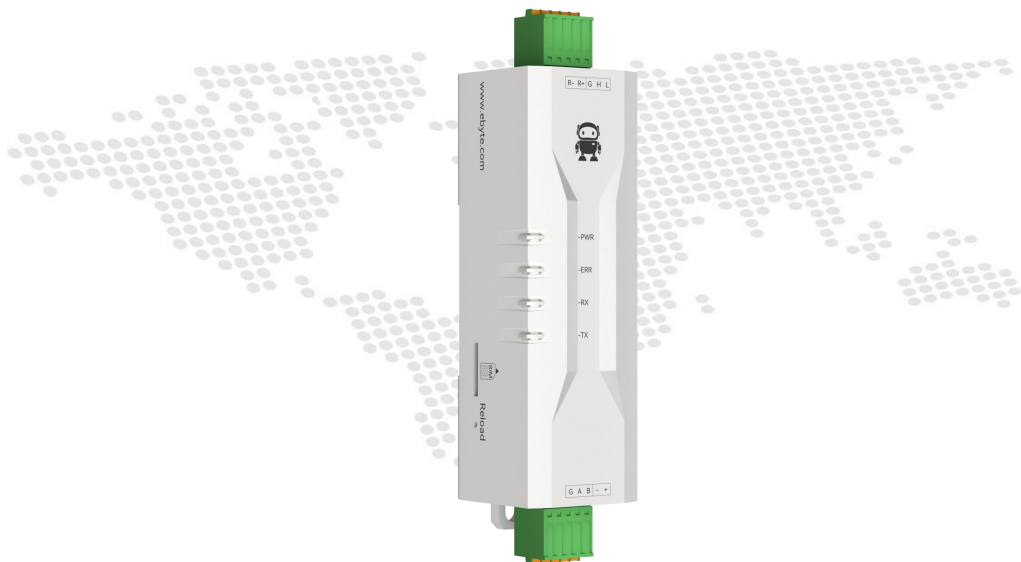
成都亿佰特电子科技有限公司

Chengdu Ebyte Electronic Technology Co.,Ltd.

Wireless Modem

用户使用手册

ECAN-101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

| | |
|-------------------------------|-----------|
| ECAN-101 智能协议转换器 | 1 |
| 第一章 概述 | 2 |
| 1.1 简介 | 3 |
| 1.2 特点功能 | 3 |
| 1.3 应用场景 | 3 |
| 第二章 产品规格参数及特性 | 3 |
| 2.1 基本参数 | 4 |
| 2.2 出厂默认参数 | 4 |
| 第三章 快速入门 | 5 |
| 3.1 连接设备 | 5 |
| 3.1.1 RS485 连接方法 | 7 |
| 3.1.2 CAN 连接方法 | 7 |
| 3.2 设备透传调试 | 8 |
| 3.3 上位机配置 | 10 |
| 第四章 硬件参数设计介绍 | 13 |
| 4.1 设计介绍 | 14 |
| 4.2 尺寸图 | 15 |
| 第五章 模式说明 | 15 |
| 5.1 数据转换方式 | 15 |
| 5.1.1 透明转换模式 | 15 |
| 5.1.2 透传带标识模式 | 19 |
| 5.1.3 协议模式 | 23 |
| 5.1.4 Modbus 模式 | 25 |
| 5.1.5 自定义协议模式 | 27 |
| 第六章 AT 指令 | 30 |
| 6.1 进入 AT 指令 | 30 |
| 6.2 退出 AT 指令 | 31 |
| 6.3 查询版本 | 31 |
| 6.4 恢复默认参数 | 31 |
| 6.5 回显设置 | 32 |
| 6.6 串口参数 | 32 |
| 6.7 设置/查询 CAN 信息 | 33 |
| 6.8 设置/查询模块转换方式 | 33 |
| 6.9 设置/查询 CAN 总线的滤波方式 | 34 |
| 6.10 设置/查询帧头帧尾数据 | 34 |
| 6.11 设置/查询标识参数 | 35 |
| 6.12 设置/查询标识参数 | 35 |
| 6.13 设置/查询传输方向 | 36 |
| 6.14 设置/查询过滤参数 | 36 |
| 6.15 删除已经设置过滤参数 | 37 |
| 修订历史 | 38 |
| 关于我们 | 38 |

第一章 概述

1.1 简介

ECAN-101 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、1 路 RS485 接口，可实现 CAN 与 RS485 不同协议数据之间的双向转换。该产品支持串口 AT 指令配置以及上位机配置设备参数和工作模式，支持透明转换、透明带标识转换、协议转换、Modbus RTU 转换、自定义（user）等五种数据转换模式，在 MODBUS 模式下，能够起到单路 CAN 接口替代多路 485 线的作用。同时，ECAN-101 智能协议转换器具有体积小巧、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

- CAN与RS485不同协议数据之间的双向转换；
- 支持透明转换、透明带标识转换、协议转换、Modbus RTU转换、自定义协议转换；
- 支持RS485接口参数配置；
- 支持AT指令参数配置；
- 支持上位机参数配置；
- 支持AT指令、上位机和按键恢复出厂设置；
- 拥有电源指示灯，状态指示灯等多种状态指示；
- 多主多从功能，用单路CAN接口替代多路485；
- 自带120欧姆终端电阻；

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 | 直流8V~28V，推荐12V或者24V供电 |
| 工作电流 | 9.5mA@12V（待机） |
| 接口类型 | 串行接口：RS485：2.5弹簧端子，压线方式 CAN接口：2.5弹簧端子，压线方式 |
| 接口特点 | 串行接口：标准RS485电平接口，波特率范围：600bps~460800bps CAN接口：遵循ISO 11898标准，支持CAN2.0A/B，CAN波特率范围：6Kbps~1000Kbps |
| CAN终端电阻 | 集成，2.5弹簧端子，压线方式，短接R120+与R120-使用 |
| 通讯指示灯 | 板载电源(PWR)、报错(ERR)、接收(RX)、发送(TX)指示灯 |
| 工作温度 | -40℃~85℃，工业级 |
| 工作湿度 | 15%~90%RH，无凝露 |

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 | 0x00000000 |
| 默认工作模式 | 透明转换模式 | 接收所有数据类型 |

第三章 快速入门

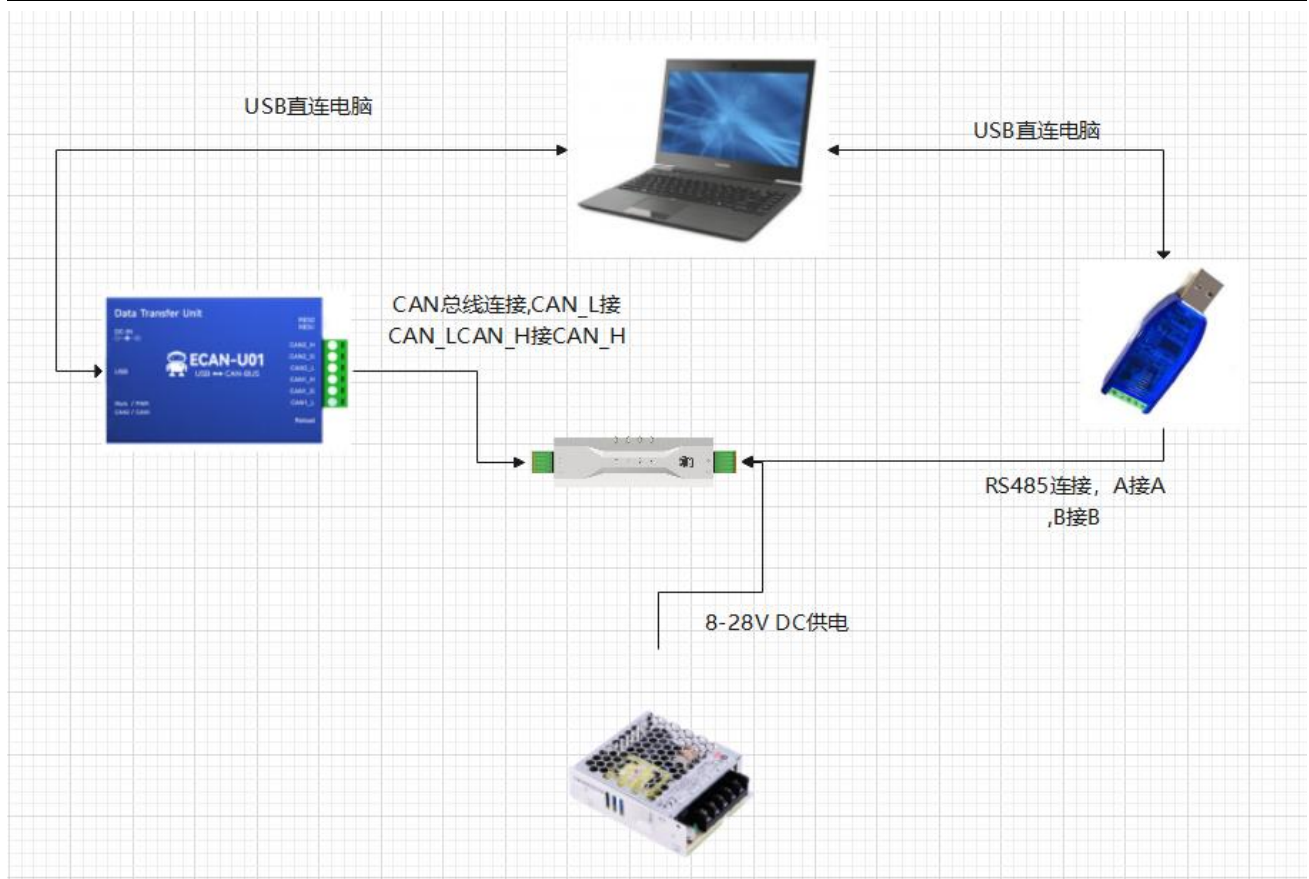
硬件准备：USB转RS485转换器*1、电源*1（推荐DC 12V供电）、电脑*1、USB转CAN设备*1（推荐[ECAN-U01](#)）

| | | |
|--|--|--|
|  |  |  |
| ECAN-101 | 电脑 | ECAN-U01 |
|  |  |  |
| DC 12V开关电源 | USB转RS485 | 线缆若干 |

软件准备：可在官网产品“相关下载”中下载“串口助手（X-COM）”、配置上位机

3.1 连接设备

这里以本设备搭配[ECAN-U01](#)为例，演示快速入门使用。

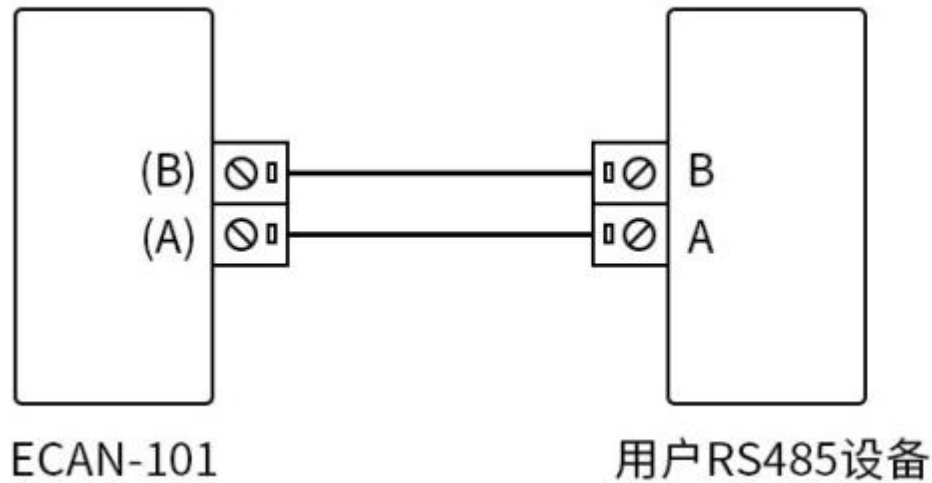


首先使用USB转RS485转换器连接本设备与电脑，转换器的RS485_A、RS485_B对应连接设备的RS485_A、RS485_B；转换器USB端与电脑直连；

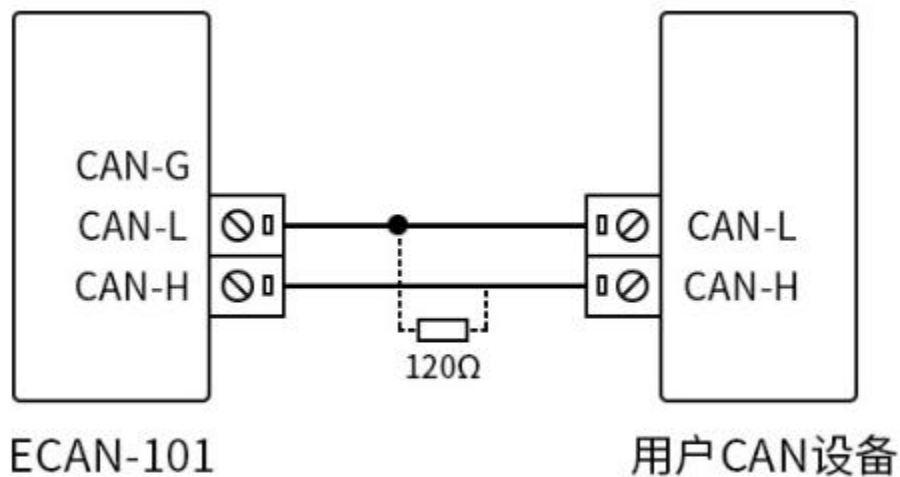
其次使用ECAN-U01连接本设备与电脑；ECAN-U01的CAN端接设备CAN端，CAN-L接CAN-L，CAN-H接CAN-H。ECAN-U01的USB端与电脑直连。

注意：第一次使用USB转RS485转换器，需要安装软件和驱动，详细使用方法请咨询相关厂家。

3.1.1 RS485连接方法



3.1.2 CAN连接方法



直线型拓扑结构是 CAN 总线布线规范中最为常用的。即主干的两条线上分支出支线到各个节点。主干的两端配置合适的终端电阻实现阻抗匹配（2km 内通常为120 欧）。

（按照ISO 11898规范，为了增强CAN-bus 通讯的可靠性，CAN-bus 总线网络的两个端点 通常要加入终端匹配电阻（120Ω）。终端匹配电阻的大小由传输电缆的特性 阻抗所决定，例如，双绞线的特性阻抗为120Ω，则总线上的两个端点也应集成120Ω 终端电阻。）

3.2 设备透传调试

按照上述方式，接好线后，给所有设备上电。

电脑打开串口调试助手，选择设备串口所用的串口号（进入软件后，选择连接到 ECAN-101 的串口号，点击“连接设备”即可建立连接。如用户不知道串口号，可通过进入 PC 机的设备管理器中查看。），并将串口参数设置为设备默认参数（波特率115200，8数据位，无校验位，1停止位）；



打开ECAN-U01调试上位机，设置CAN波特率为100K（ECAN-101默认出厂CAN波特率）；



经过以上步骤，可开始透传转换调试：

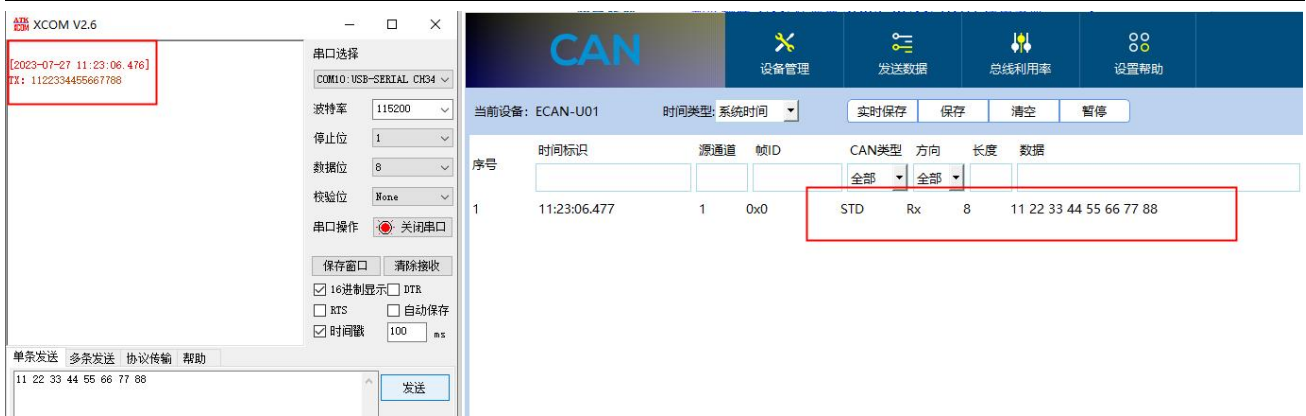
首先进行CAN帧转换为串行帧：

CAN调试器（ECAN-U01）发送数据：帧ID:0x0123，数据域：99 88 77 66 55 44 33 22，设备CAN端接收下方CAN调试器（ECAN-U01）发送的数据，可看到串口助手上显示RS485接口转发的CAN数据（16进制显示）：99 88 77 66 55 44 33 22。



串行帧转化为CAN帧：

从串口调试助手以16进制发送数据：11 22 33 44 55 66 77 88，设备CAN端转换出的数据，可在下方CAN调试器（ECAN-U01）显示，数据为：11 22 33 44 55 66 77 88，帧ID:0x0000，标准帧。



从上述调试操作中，可看到设备实现了串行帧与CAN帧的透明转换。

3.3 上位机配置

设备可通过AT指令或者上位机配置设备参数，这里简单介绍一下上位机配置的使用方法。

1. 更改设备参数可打开ECAN-101配置上位机，选择默认串口参数（波特率115200，8数据位，无校验位，1停止位），点击打开串口。



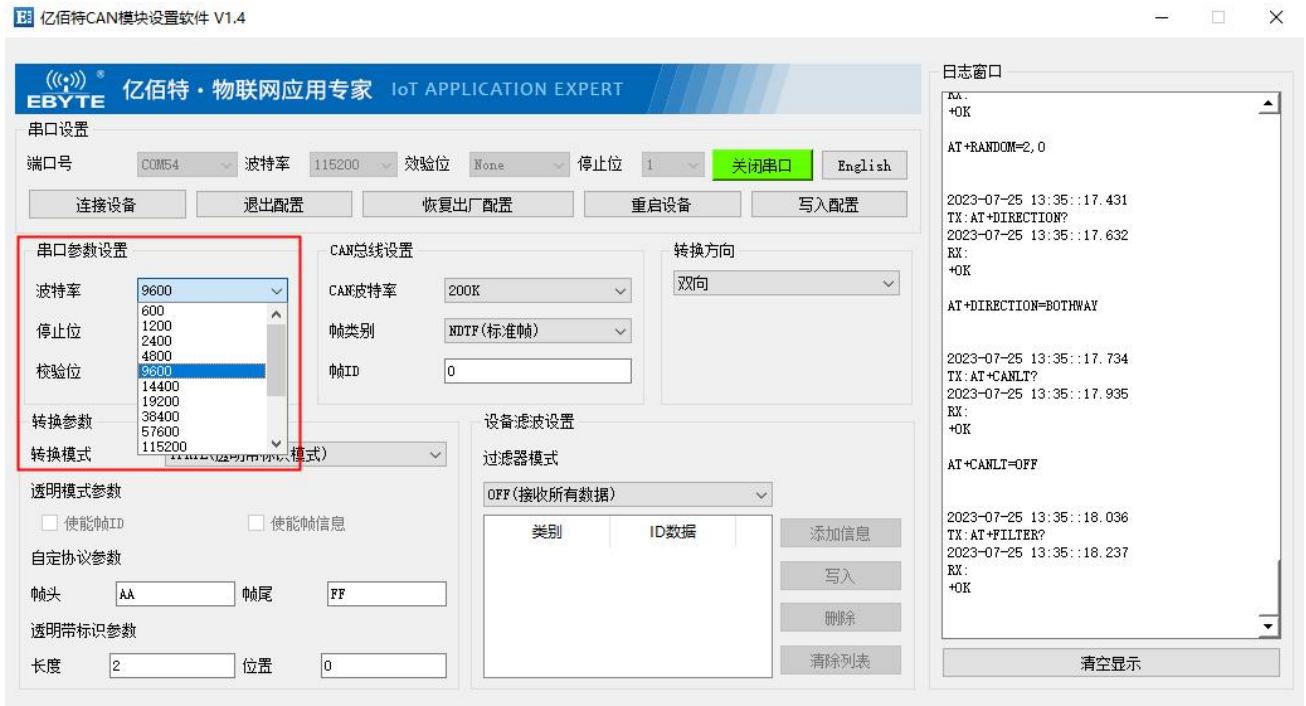
2. 点击“连接设备”，设备进入配置模式并读取参数



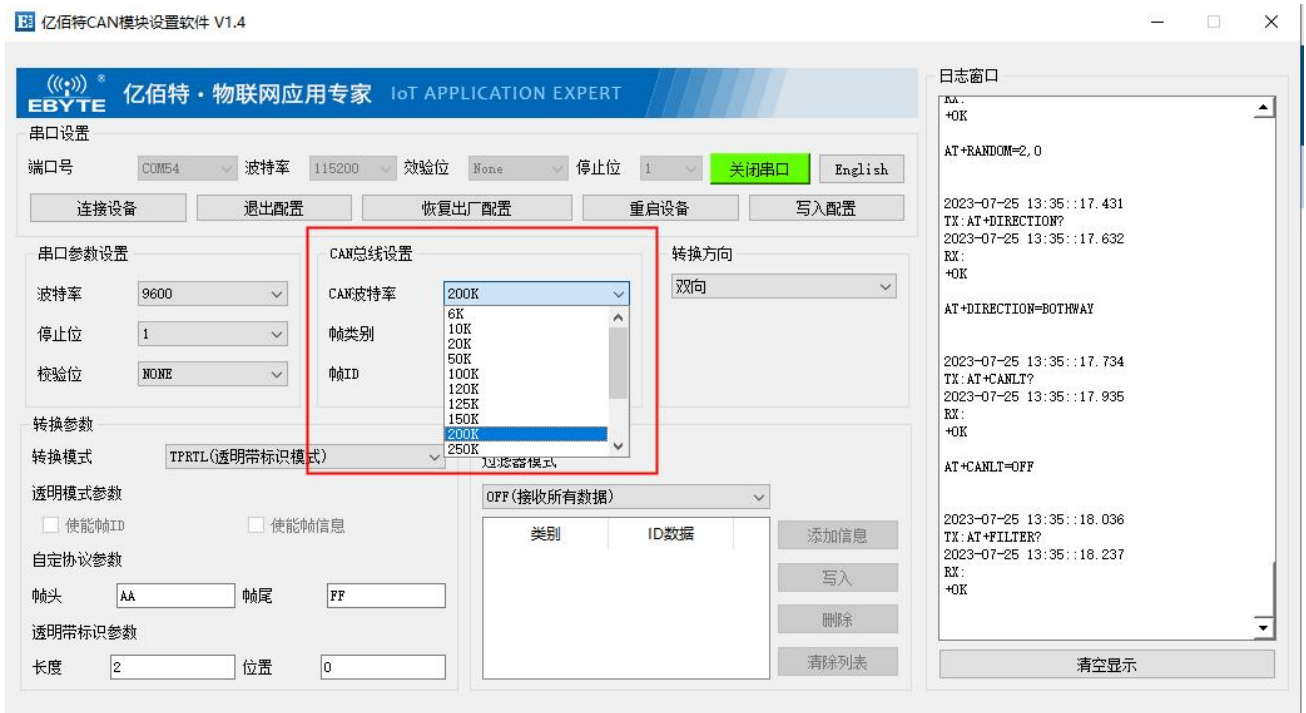
3. 可根据不同应用场景来更改设备“串口参数设置”、“CAN总线设置”、“转换方向”、“转换模式”、“设备过滤设置”等参数，修改参数后点击“写入配置”并“重启设备”后参数生效。



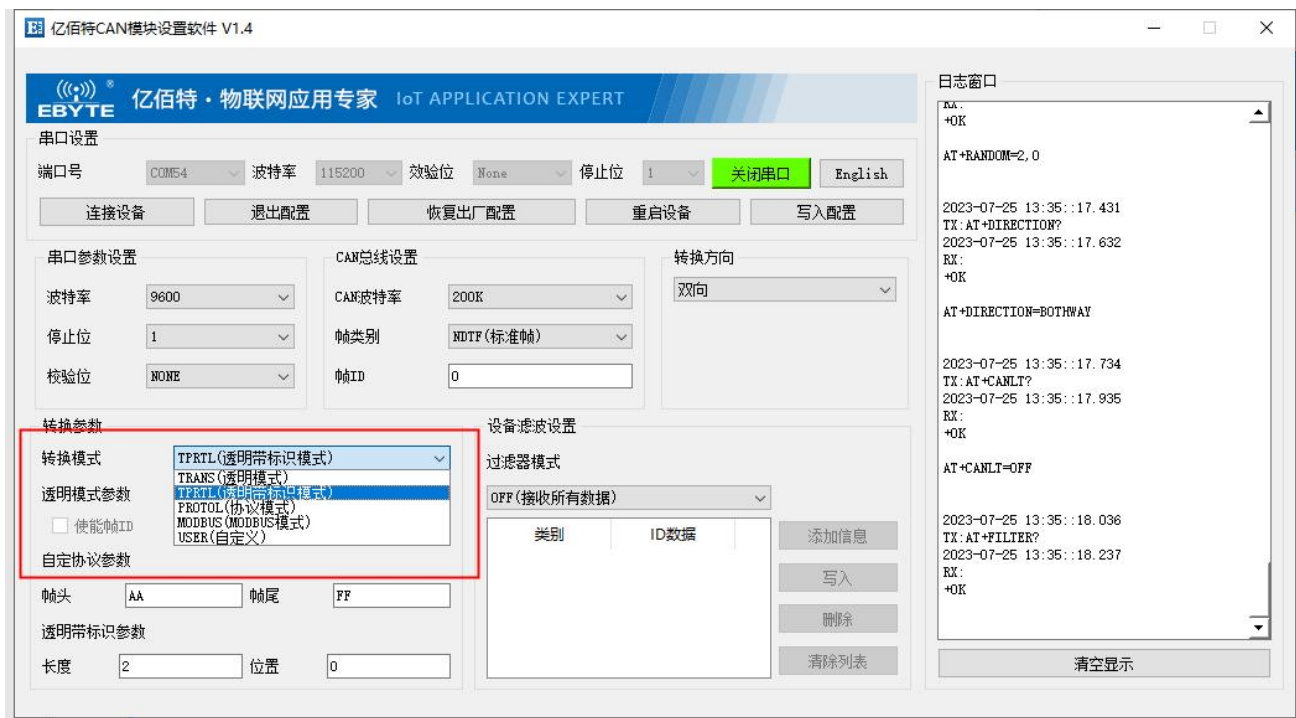
串口参数设置：此部分设置是串口参数设置，分为“波特率”、“校验位”、“停止位”，波特率范围有：600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 43000, 57600, 76800, 115200, 128000, 230400, 256000, 460800 单位：bps；校验位：奇校验（ODD）、偶校验（EVEN）、无校验（NONE）；停止位：1、2。



CAN总线设置：此部分设置是CAN部分设置，分为“CAN波特率”、“帧类别”、“帧ID”，CAN波特率范围：6K, 10K, 20K, 50K, 100K, 120K, 125K, 150K, 200K, 250K, 400K, 500K, 600K, 750K, 1000K 单位：bps，如设置的波特率与接入模块的总线波特率不一致时，将无法正常通信转换。帧类别：NDTF (标准帧)，EDTF (扩展帧)，帧ID： 0~7FF (标准帧)，0~1FFFFFF (扩展帧)



转换模式：分为五种，TRANS (透明)，TPRTL (透明带标识)，PROTOL (协议模式)，USER (自定义协议)，MODBUS (MODBUS)，每种模式协议规则不一样，详情见第五章描述。



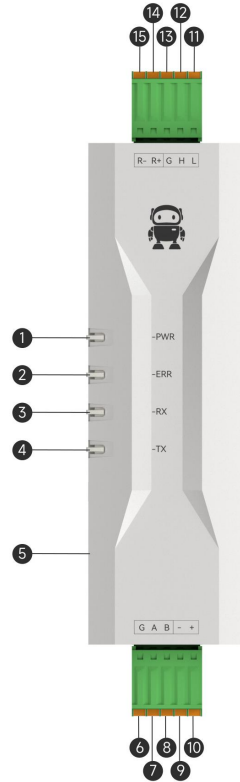
设备过滤设置：分为四种过滤方式，1：接收所有数据 2. 只接收扩展帧 3. 只接收标准帧 4. 用户自定义。

前三种只用选择配置即可，最后一种用户自定义过滤可以添加14组自定义ID过滤。点击“添加信息”，在左侧信息框内添加想要的滤波ID，然后点击“写入”即可。



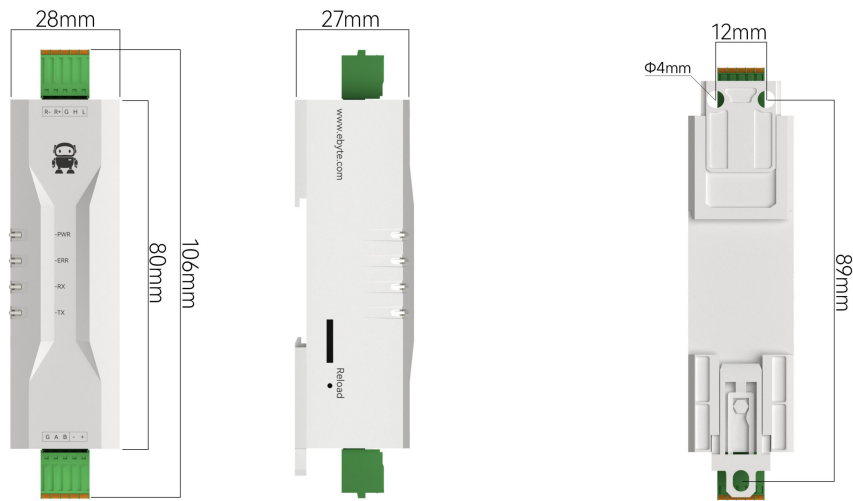
第四章 硬件参数设计介绍

4.1 设计介绍



| 引脚序号 | 引脚名称 | 引脚用途 |
|------|---------|----------------------------|
| 1 | PWR | 电源指示灯，上电常亮（红） |
| 2 | ERR | 报错指示灯，总线报错常亮（红），总线恢复熄灭 |
| 3 | RX | 数据指示灯，串口接收数据时，闪烁（绿） |
| 4 | TX | 数据指示灯，串口发送数据是，闪烁（绿） |
| 5 | Restore | Restore按键，长按5s以上恢复出厂设置 |
| 6 | GND | 信号地 |
| 7 | 485-A | RS485总线数据A |
| 8 | 485-B | RS485总线数据B |
| 9 | V- | 电源负极，直流8V~28V，推荐12V或者24V供电 |
| 10 | V+ | 电源正极，直流8V~28V，推荐12V或者24V供电 |
| 11 | CAN_L | CAN通信接口 |
| 12 | CAN_H | CAN通信接口 |
| 13 | CAN_G | CAN接地 |
| 14 | R+ | 120欧姆匹配电阻正，短接R120-使用 |
| 15 | R- | 120欧姆匹配电阻负，短接R120+使用 |

4.2 尺寸图



第五章 模式说明

在“透明转换”和“格式转换”时，使用一个字节的帧信息来标识该CAN 帧的一些信息，如类型、格式、长度等。帧信息格式如下。

帧信息说明

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|------|------|------|------|------|------|------|------|
| FF | RTR | NO | NO | DLC3 | DLC2 | DLC1 | DLC0 |

表 1.1 帧信息

FF: 标准帧和扩展帧的标识，0 为标准帧，1 为扩展帧；

RTR: 远程帧和数据帧的标识，0 为数据帧，1 为远程帧；

NO: 不使用；

NO: 不使用；

DLC3~DLC0: 标识该 CAN 报文数据长度；

5.1 数据转换方式

ECAN-101 设备支持五种数据转换方式：透明转换、透明带标识转换、协议转换、MODBUS 转换和自定义协议转换。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 |
| 透明带标识转换 | 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 |
| 自定义协议转换 | CAN 与 RS485 双向转换 |

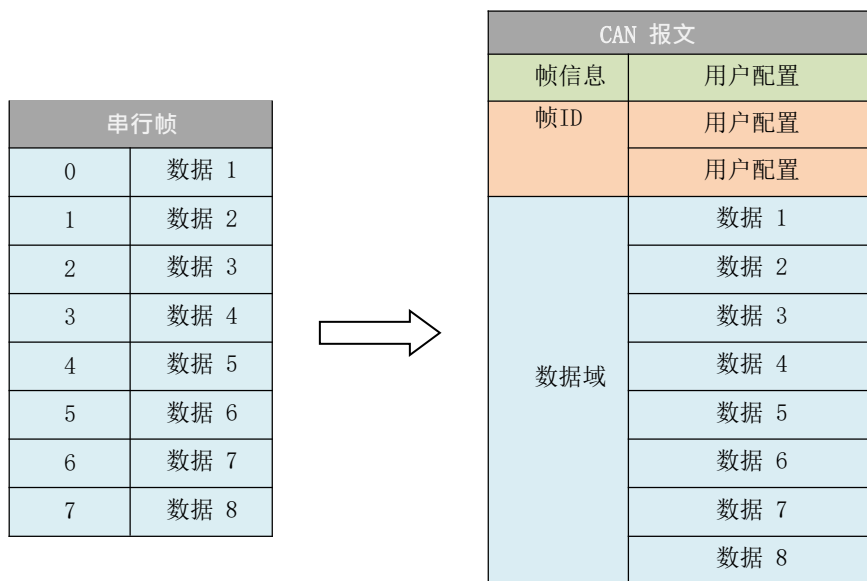
5.1.1 透明转换模式

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。

ECAN-101 设备可以将 CAN 总线收到的有效数据原封不动的转换到 RS485 串行总线输出。同样的，设备也可以将 RS485 串行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报文帧的数据域里。模块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报文帧信息（帧类型部分）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。

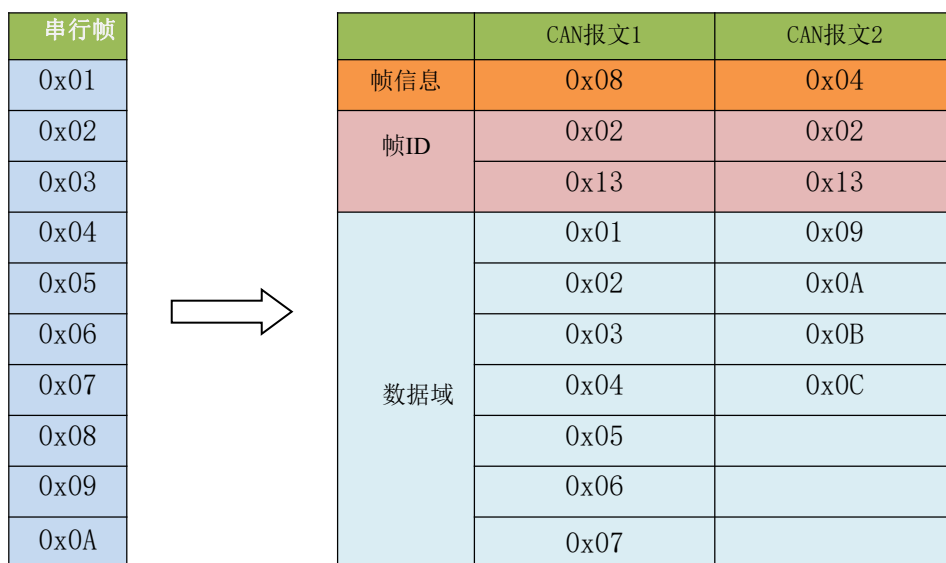


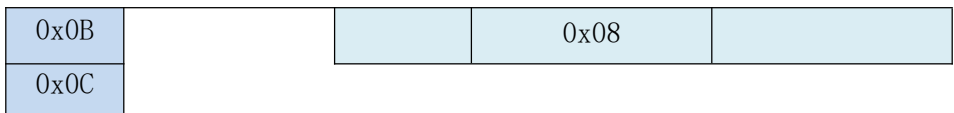
串行帧转换成 CAN 报文（透明方式）

转换示例：

串行帧转换成 CAN 报文（透明方式）。

假设置CAN帧信息为“标准帧”，帧 ID：“0x0213，串行帧数据为 0x01 ~ 0x0C，那么转换格式如下所示。CAN报文的帧 ID为 0x0213（用户配置），帧类别：标准帧（用户配置），串行帧中的数据部分将不作任何修改地转换到CAN报文中。



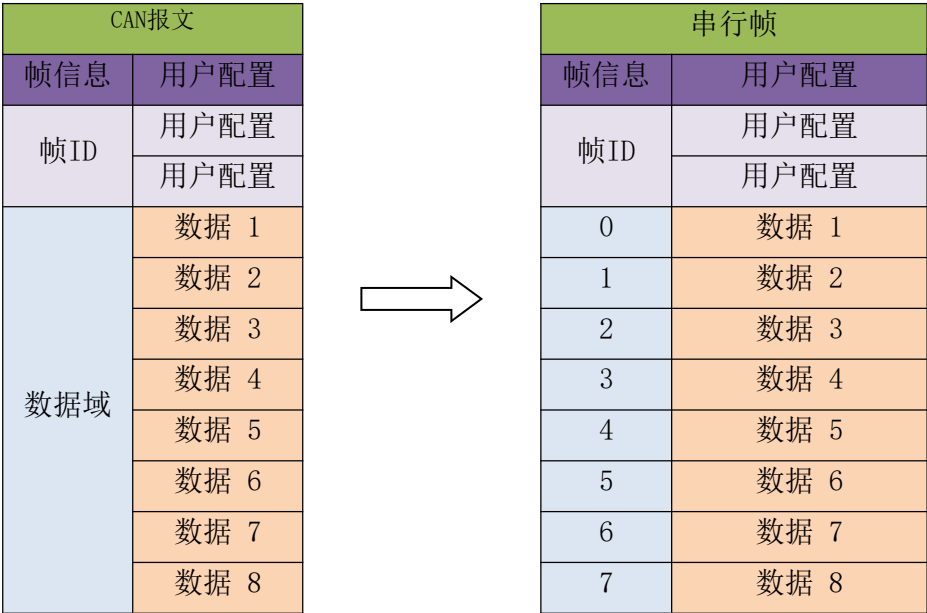


串行帧转换成 CAN 报文（透明方式）

2. CAN 报文转串行帧

转换时将CAN报文数据域中的数据依序全部转换到串行帧中。如果在配置的时候,勾选“使能帧信息”，那么模块会将CAN报文的“帧信息”字节直接填充至串行帧。如果勾选“使能帧ID ”，那么也将CAN报文的“帧ID”字节全部填充至串行帧。

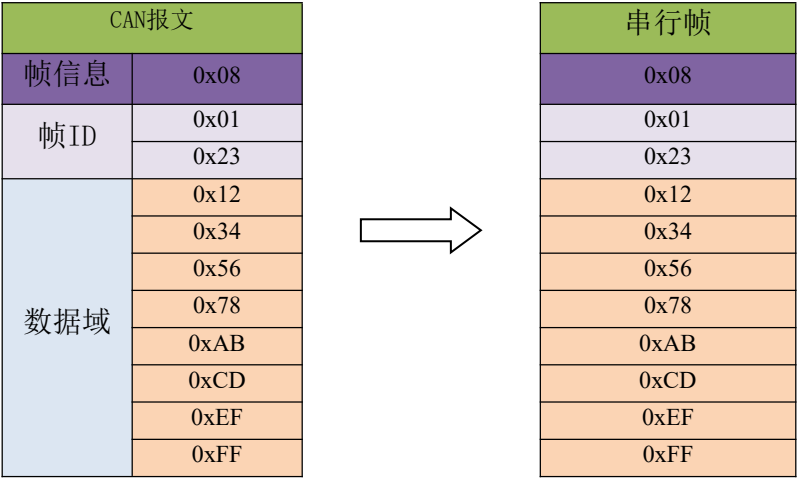
注意：如果要在串行接口收到CAN的帧信息或者帧ID 需要使能对应的功能。才可收到对应的信息。



CAN报文转换成串行帧 （透明方式）

转换示例：

该示例配置的CAN报文“帧信息”使能，“帧 ID”使能。帧 ID1：0x123，帧类别：标准帧，帧类型：数据帧。转换方向：双向。数据为 0x12， 0x34， 0x56， 0x78， 0xAB， 0xCD， 0xEF， 0xFF。则转换前后数据如下：



CAN报文转换成串行帧 （透明方式）

5.1.2 透传带标识模式

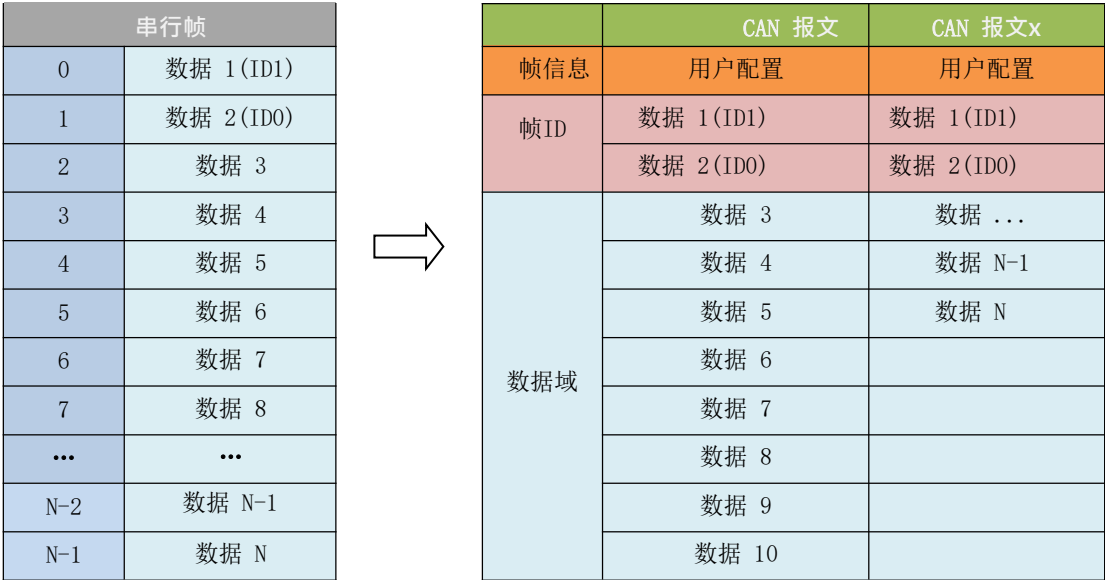
透明带标识转换是透明转换的特殊用法，在串行帧中带 CAN报文的ID信息，可以根据需要发送不同ID的CAN报文。有利于用户通过模块更方便的组建自己的网络，使用自定的应用协议。该方式把串行帧中的 ID 信息自动转换成CAN总线的帧ID。只要在配置中告诉模块该ID信息在串行帧的起始位置和长度，模块在转换时提取出这个帧ID填充在CAN报文的帧ID域里，作为该串行帧的转发时的CAN报文的ID。在CAN报文转换成串行帧的时候也把CAN报文的ID转换在串行帧的相应位置。

转换方式：

1. 串行帧转 CAN 报文

串行帧中所带有的CAN报文的“帧 ID”在串行帧中的起始地址和长度可由配置设定。起始地址的范围是0~7，长度范围分别是1~2（标准帧）或1~4（扩展帧）。转换时根据事先的配置将串行帧中的 CAN报文“帧 ID”对应全部转换到CAN报文的帧ID域中（如果所带帧ID个数少于CAN报文的帧 ID 数，那么在CAN报文中帧ID的高字节补0。），其它的数据依序转换，如果一帧CAN报文未将串行帧数据转换完，则仍然用相同的ID作为CAN报文的帧ID继续转换直到将串行帧转换完成。

注意：如果将ID长度大于2，设备发送的帧类别会设置为扩展帧。此时用户配置的帧ID和帧类别无效，由串行帧里边的数据决定。标准帧的帧ID范围为：0x000-0x7FF，分别表示为帧 ID1、帧 ID0，其中帧 ID1 为高字节，扩展帧的帧 ID 范围为：0x00000000-0x1FFFFFFF，分别表示为帧 ID3、帧 ID2、帧 ID1、帧 ID0，其中帧 ID3 为高字节。



串行帧转换成 CAN 报文（透传带标识）

转换示例：

串行帧转CAN报文（透明带标识）。

该示例配置的CAN配置参数。转换模式：透明带标识转换，起始地址2，长度3。帧类型：扩展帧，帧ID：无需配置，转换方向：双向。上位机配置界面如下：



则转换前后数据示意图如下：

| 串行帧 | | CAN报文2 | | CAN报文2 | |
|-----|--|--------|----|--------|----|
| 01 | | 帧信息 | 88 | 帧信息 | 85 |
| 02 | | 帧ID | 00 | 帧ID | 00 |
| 03 | | | 03 | | 03 |
| 04 | | | 04 | | 04 |
| 05 | | | 05 | | 05 |
| 06 | | 数据域 | 01 | 数据域 | 0C |
| 07 | | | 02 | | 0D |
| 08 | | | 06 | | 0E |
| 09 | | | 07 | | 0F |
| 0A | | | 08 | | 10 |
| 0B | | | 09 | | |
| 0C | | | 0A | | |
| 0D | | | 0B | | |
| 0E | | | | | |
| 0F | | | | | |
| 10 | | | | | |

| 串行帧 | | CAN报文2 | | CAN报文2 | |
|-----|--|--------|----|--------|----|
| 01 | | 帧信息 | 88 | 帧信息 | 85 |
| 02 | | 帧ID | 00 | 帧ID | 00 |
| 03 | | | 03 | | 03 |
| 04 | | | 04 | | 04 |
| 05 | | | 05 | | 05 |
| 06 | | 数据域 | 01 | 数据域 | 0C |
| 07 | | | 02 | | 0D |
| 08 | | | 06 | | 0E |
| 09 | | | 07 | | 0F |
| 0A | | | 08 | | 10 |
| 0B | | | 09 | | |
| 0C | | | 0A | | |
| 0D | | | 0B | | |
| 0E | | | | | |
| 0F | | | | | |
| 10 | | | | | |

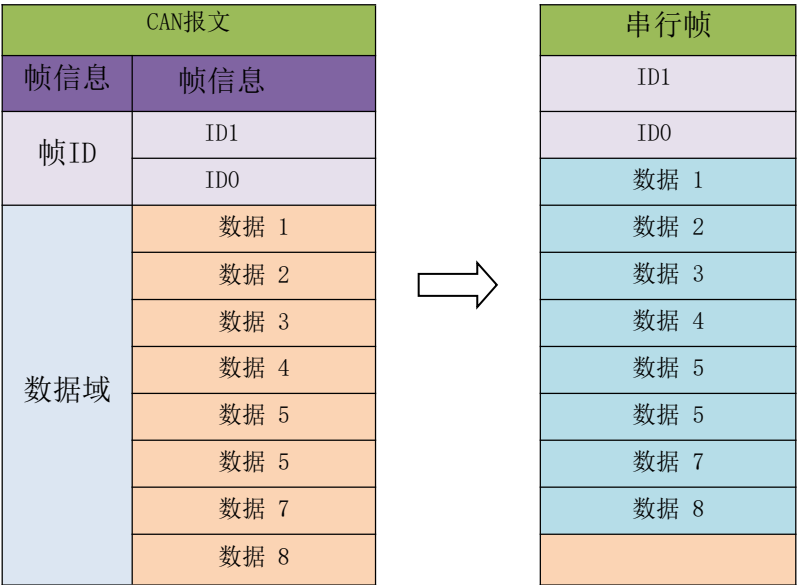
串行帧转 CAN 报文示例（透传带标识转换）

串口调试助手发送串行帧：01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10，则在CAN调试器上接收到转换出来的CAN帧：帧ID为0x00030405，第一帧数据域为：01 02 06 07 08 09 0A 0B，第二帧数据域为：0C 0D 0E 0F 10，从这里可以看出，设备已经将串行帧中第二位开始，后面三位03 04 05作为帧ID提取出来，剩下数据作为CAN帧数据域重新组包。



2. CAN 报文转串行帧

对于CAN报文，收到一帧就立即转发一帧，每次转发的时候也是根据事先配置的 CAN帧ID在串行帧中的位置和长度把接收到的CAN报文中的ID作相应的转换。其它数据依序转发。值得注意的是，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，否则可能致使通讯不成功。



CAN报文转换成串行帧

转换示例：

该示例配置的CAN配置参数。转换模式：透明带标识转换，起始地址2，长度3。帧类型：扩展帧，帧类型：数据帧。转换方向：双向。上位机配置界面如下：



发送标识符：0x00000123，则转换前后数据如下。

| CAN报文 | | | | 串行帧 | |
|-------|----|-----|----|-----|--|
| 帧信息 | 88 | 帧ID | 00 | 99 | |
| | 00 | | 00 | 88 | |
| | 01 | | 01 | 00 | |
| | 23 | | 23 | 01 | |
| | 99 | | 99 | 23 | |
| 数据域 | 88 | | 77 | 77 | |
| | 77 | | 66 | 66 | |
| | 66 | | 55 | 55 | |
| | 55 | | 44 | 44 | |
| | 44 | | 33 | 33 | |
| | 33 | | 22 | 22 | |

CAN 报文转串行帧示例（透明带信息转换）

从ECAN-U01（CAN调试器）发送数据：帧ID:0x00000123，数据域：99 88 77 66 55 44 33 22，串口调试助手收到串行帧：99 88 00 01 23 77 66 55 44 33 22，从这里可以看出，CAN帧在转换成串行帧时，按照我们的设定的规则来填充数据，将CAN帧ID填进数据域第二位开始的后三位。



5.1.3 协议模式

CAN格式转换固定13字节表示一个CAN帧数据，13个字节内容包括CAN 帧信息+帧ID +帧数据。在该转换模式下，设置的CANID无效，因为此时发送的标识符（帧 ID）由上述的格式串行帧中的帧ID数据填充。配置的帧类型也无效，由格式串行帧中的帧信息来决定帧类型。格式如下：

| CAN 固定格式串行帧（13 字节） | | |
|--------------------|-------|-------|
| 帧信息 | 帧 ID | 帧数据 |
| 1Byte | 4Byte | 8Byte |

帧信息如下表：

帧 ID 的长度为 4 字节，标准帧有效位是 11 位，扩展帧有效位是 29 位。

| 扩展帧 ID 号 0x12345678 | | | |
|---------------------|------|------|------|
| 0x12 | 0x34 | 0x56 | 0x78 |

| 标准帧 ID 号 0x3FF | | | |
|----------------|------|------|------|
| 0x00 | 0x00 | 0x03 | 0xFF |

1. 串行帧转 CAN 报文

在串行帧转CAN报文的过程中，以固定字节(13字节)对齐的串行数据帧中，某段固定字节的数据格式不标准，将会不对固定字节长度进行转换，接着转换后面的数据。如果转换后发现少了某些 CAN 报文，请检查对应报文的固定字节长度串行数据格式是否符合标准格式。

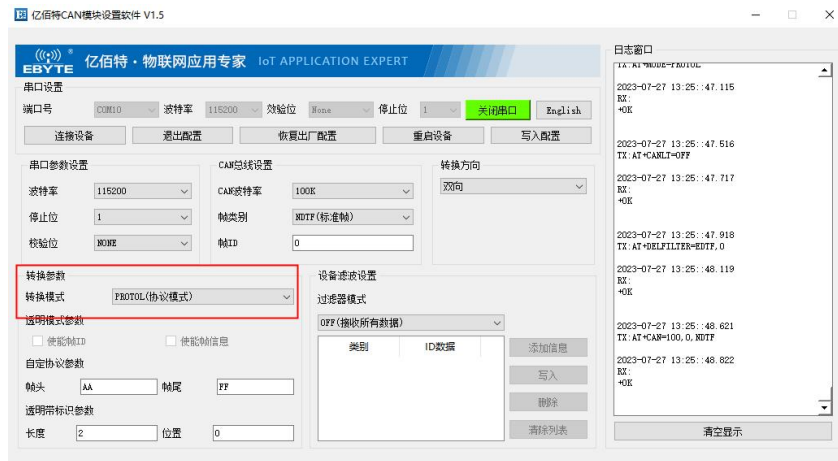
帧数据在CAN格式转换时，长度固定为8字节。有效长度由DLC3~DLC0的值决定，有效数据不足固定长度时，需要补0到固定长度。

此模式下，要注意严格按照固定字节的格式串行数据格式才能转换成功，CAN 模式转换可参考示例（CAN 格式转换标准帧示例）所示，转换时首先要确保帧信息无误，数据长度表示无误，否则不会进行转换。

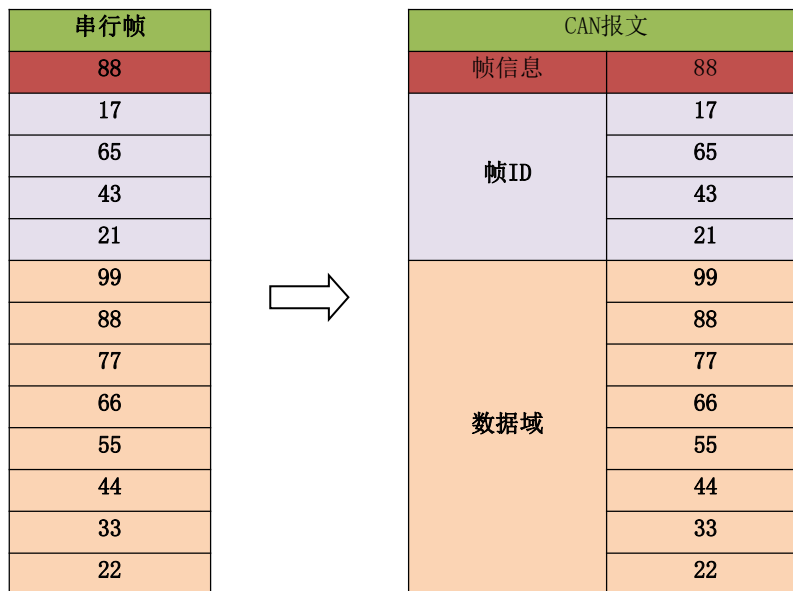
转换示例：

串行帧转CAN报文（协议模式）。

该示例配置的CAN配置参数。转换模式：协议模式；帧类别：扩展帧；转换方向：双向。帧ID：无需配置；上位机配置如下：



转换前后数据示意图如下：



串行帧转CAN报文（协议模式）

串口调试助手发送数据：88 17 65 43 21 99 88 77 66 55 44 33 22，则在CAN调试器上接收到转换出来的CAN帧：帧ID为0x17654321，数据域为：99 88 77 66 55 44 33 22的扩展帧。从这里可以看出串行帧中的第一位数据“88”，已经表明帧类型以及帧数据域长度：扩展帧、数据域长度8。



5.1.4 Modbus模式

Modbus协议是一种标准的应用层协议，广泛应用于各种工控场合。该协议开放，实时性强，通讯验证机制好，非常适用于通信可靠性要求较高的场合，模块在串口侧使用的是标准Modbus RTU协议格式，所以模块不仅支持用户使用Modbus RTU协议，模块也可以直接和其它支持Modbus RTU协议的设备相接口。在CAN侧，制定了一个简单易用的分段通讯格式来实现Modbus通讯，一个长度大于一个CAN报文最大数据长度的信息进行分段以及重组的方法，“数据 1”用于分段标识数据，传输的Modbus协议内容即可从“数据 2”字节开始，如果协议内容大于7个字节，那么将剩下的协议内容照这种分段格式继续转换，直到转换完成。当CAN总线上不存在其他数据，可以不设置帧过滤器。即可完成通讯。当总线上面有其他数据则需要设置过滤器。区分设备接收数据来源。根据此方式。可以实现一条总线上面存在多个主机的通讯。在CAN总线上传输的数据不需要CRC校验方式。CAN总线上的数据校验已经具有更加完善的校验方式。此模式下，设备支持的是Modbus验证和转发，而不是Modbus的主机或者从机，用户按照Modbus协议通讯即可。

分段传输协议：

| | | | | | | | | |
|-------|---------------|------|-----|---------------|-----------|---|---|---|
| 位编号 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 帧信息 | FF | RTR | EDL | BRS | DLC（数据长度） | | | |
| 帧 ID3 | X | X | X | ID. 28-ID. 24 | | | | |
| 帧 ID2 | ID. 23-ID. 16 | | | | | | | |
| 帧 ID1 | ID. 15-ID. 8 | | | | | | | |
| 帧 ID0 | ID. 7-ID. 0 | | | | | | | |
| 数据 1 | 分段标记 | 分段类型 | | 分段计数器 | | | | |
| 数据 2 | DATE | | | | | | | |
| 数据 3 | DATE | | | | | | | |
| 数据 4 | DATE | | | | | | | |
| 数据 5 | DATE | | | | | | | |
| 数据 6 | DATE | | | | | | | |
| 数据 7 | DATE | | | | | | | |
| 数据 8 | DATE | | | | | | | |

一个长度大于一个CAN报文最大数据长度的信息进行分段以及重组的方法，CAN报文时，“数据 1”用于分段标识数据，分段报文格式如下，传输的Modbus协议内容即可从“数据 2”字节开始，如果协议内容大于7个字节，那么将剩下的协议内容照这种分段格式继续转换，直到转换完成。

| 分段标记 | 分段类型 | | 分段计数器 | | | | |
|------|------|------|-------|------|------|------|------|
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |

- 分段标记：表明该报文是否是分段报文。该位为 0 表示单独报文，为 1 表示属于被分段报文中的一帧。
- 分段类型：表明是第一段、中间段的还是最后段。

| 位值 | 含义 | 说明 |
|----|-------|----------------------------|
| 0 | 第一个分段 | 如果分段计数器包含值 0，那么这是分段系列中的第一段 |
| 1 | 中间分段 | 表明这是一个中间分段 |
| 2 | 最后分段 | 标志最后一个分段 |

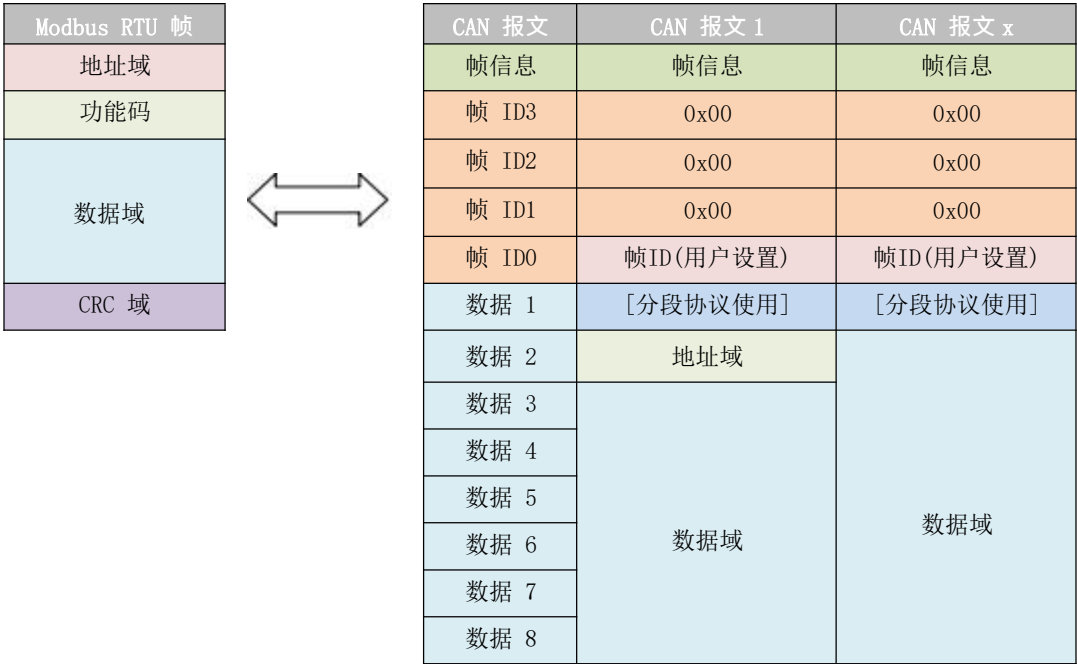
- 分段计数器：每一个段的标志，表示该段在整个报文中的序号，如果是第几个段，计数器的值就是几。这样在接收时就能够验证是否有分段被遗失。共使用5Bit，范围为 0~31。

1. 串行帧转 CAN报文

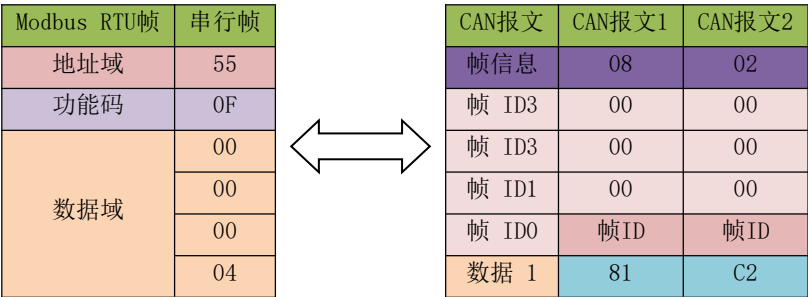
串行接口采用的是标准的 Modbus RTU 协议，所以用户帧符合此协议即可。如果传输的帧不符合 Modbus RTU 格式，那么模块会将接收到的帧丢弃，而不予转换。

2. CAN报文 转串行帧

对于CAN总线的Modbus协议数据，无需做循环冗余校验（CRC16），模块按照分段协议接收，接收完一帧解析后自动加上循环冗余校验（CRC16），转换成Modbus RTU 帧发送至串行总线。如果接收到的数据不符合分段协议，则将该组数据丢弃不予转换。



转换示例1:



| | | | | |
|------|----|------|----|----|
| | 02 | 数据 2 | 55 | 00 |
| | 00 | 数据 3 | 0F | 00 |
| | 00 | 数据 4 | 00 | |
| CRC域 | 29 | 数据 5 | 00 | |
| | 13 | 数据 6 | 00 | |
| | | 数据 7 | 04 | |
| | | 数据 8 | 02 | |

转换示例2:

串口侧 Modbus RTU 数据（十六进制）：

01 03 14 00 0A 00 00 00 00 00 14 00 00 00 00 00 17 00 2C 00 37 00 C8 4E 35

第一个字节 01 是 Modbus RTU 的地址码，转换成 CAN 的 ID. 7-ID. 0: 最后 2 个字节（4E 35）为 Modbus RTU 的 CRC 校验，丢掉不转换。

最终转成 CAN 数据报文如下：

第 1 帧 CAN 报文：81 01 03 14 00 0A 00 00

第 2 帧 CAN 报文：a2 00 00 00 14 00 00 00

第 3 帧 CAN 报文：a3 00 00 17 00 2C 00 37

第 4 帧 CAN 报文：c4 00 c8 CAN 报文的帧类型（标准帧或扩展帧）通过配置软件设置；

每个 CAN 报文的第一个数据都有分段信息来填充（81、a2、a3 和 c4），该信息不转换 到 Modbus RTU 帧中，仅做为报文的确认控制信息。



5.1.5 自定义协议模式

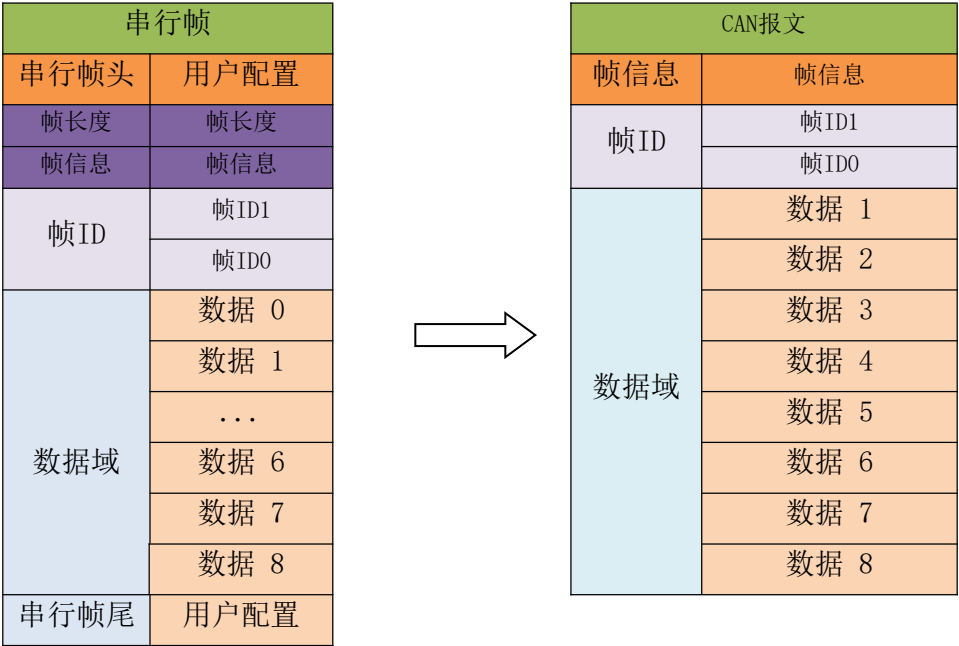
必须是完整的符合自定义协议规定的串行帧格式，要包含用户所配置模式下的串行帧所有内容，除数据域外若其他字节内容若有误此帧将不能成功发送。串行帧包含的内容：帧头、帧长度、帧信息、帧ID、数据域、帧尾。

注意：此模式下用户配置的帧ID和帧类别无效，将根据串行帧里边的格式进行数据转发。

1. 串行帧转CAN报文

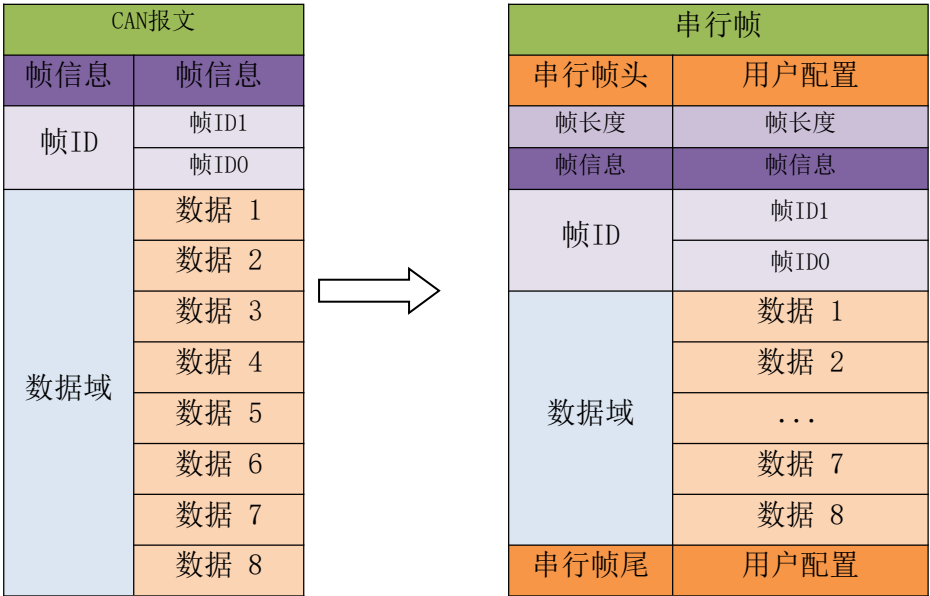
串行帧格式必须符合规定的帧格式，由于CAN帧格式是基于报文的，串行帧格式是基于字节传输的。因此为了让用户方便使用CAN-bus，将串行帧格式向CAN帧格式靠拢，在串行帧中规定了一帧的起始及结束，即AT命令中的“帧头”和“帧尾”，用户可自行配置。帧长度指的是从帧信息开始到最后一个数据结束的长度，不包括串行帧尾。帧信息分为扩展帧和标准帧，标准帧固定表示为0x00，扩展帧固定表示为0x80，与透明转换和透明带标识转换不同，自定义协议转换中，无论每帧数据域包含的数据长度为多少，其帧信息内容都固定不变。当帧类型为标准帧（0x00）时，帧类型后两个字节表示帧 ID，其中高位在前；当帧信息为扩展帧（0x80）时，帧类型后4个字节表示帧ID，其中高位在前。

注意：自定义协议转换中，无论每帧数据域包含的数据长度为多少，其帧信息内容都固定不变。固定为标准帧（0x00）或者扩展帧（0x80）。帧ID需要符合ID范围，否则ID可能出错。



2.CAN报文转串行帧

CAN总线报文收到一帧即转发一帧，模块会将CAN报文数据域中的数据依次转换，同时会向串行帧添加帧头、帧长度、帧信息等数据，实际为串行帧转CAN报文的逆向形式。



CAN报文转换成串行帧

转换示例：

串行帧转CAN报文（自定义协议）。

该示例配置的CAN配置参数。转换模式：自定义协议，帧头AA，帧尾：FF, 转换方向：双向。帧ID：无需配置，帧类别：无需配置，上位机配置如下：

亿佰特CAN模块设置软件 V1.5

EBYTE 亿佰特·物联网应用专家 IoT APPLICATION EXPERT

串口设置
端口号 COM10 波特率 115200 校验位 None 停止位 1 打开串口 English

连接设备 退出配置 恢复出厂配置 重启设备 写入配置

串口参数设置
波特率 115200 停止位 1 校验位 NONE

CAN总线设置
CAN波特率 100K 帧类别 NDTF (标准帧) 帧ID 0

转换方向
双向

转换参数
转换模式 USER(自定义)

透明模式参数
☐ 使能帧ID ☐ 使能帧信息

自定义协议参数
帧头 AA 帧尾 FF

透明带标识参数
长度 2 位置 0

设备滤波设置
过滤器模式 OFF (接收所有数据)

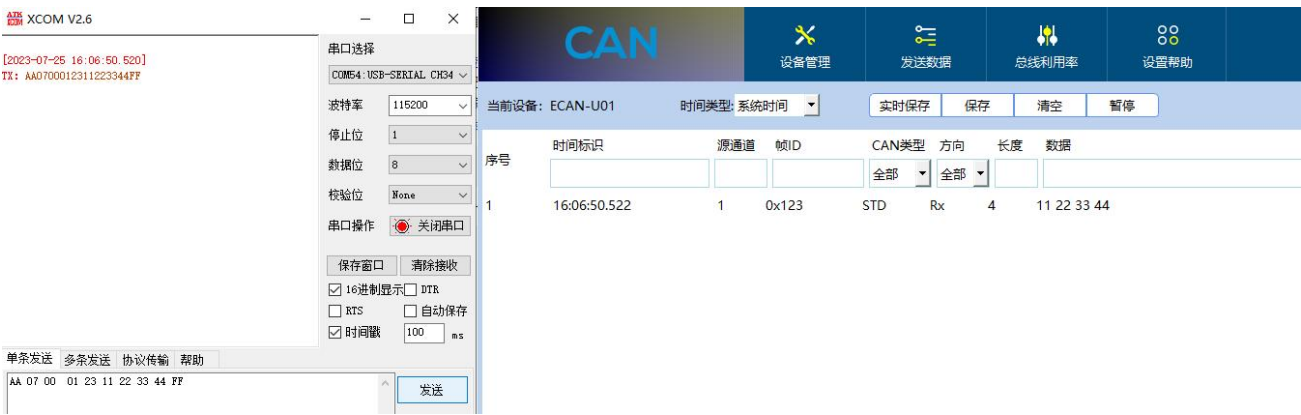
| 类别 | ID数据 |
|----|------|
| | |

添加信息 写入 删除 清除列表

则转换前后数据如下：

| 串行帧 | | CAN报文 | |
|------|----|-------|----|
| 串行帧头 | AA | 帧信息 | 00 |
| 帧长度 | 07 | 帧ID | 01 |
| 帧信息 | 00 | | 23 |
| 帧ID | 01 | 数据域 | 11 |
| | 23 | | 22 |
| | | | 33 |
| | | | 44 |
| 数据域 | | | |
| 串行帧尾 | FF | | |

因为已经设置了帧头为AA，帧尾为FF，要发送数据域为11 22 33 44，帧ID为0x123的标准帧数据，串口调试助手这边就按照协议发送：AA 07 00 01 23 11 22 33 44 FF，设备CAN端转换过来的数据在下方CANT调试器（ECAN-U01）上显示的就是数据域11 22 33 44，帧ID为0x123的标准帧。同理CAN帧转串行帧同样按照这个逻辑封装。



第六章 AT 指令

- 1、进入AT指令模式：串口发送 +++ ，3秒内再次发送 AT，设备回显AT MODE，则进入AT指令模式。
- 2、若无特殊说明，后续所有的AT指令操作都需增加 “\r\n”。
- 3、所有示例都是关闭命令回显功能下进行的。
- 4、设置参数后需要重启设备即可是设置的参数生效。

错误码表：

| 错误码 | 说明 |
|-----|---------|
| -1 | 无效的命令格式 |
| -2 | 无效的命令 |
| -3 | 暂未定义 |
| -4 | 无效的参数 |
| -5 | 暂未定义 |

默认参数

| 参数类别 | 参数名称 | 参数值 | 相关指令 |
|------|------|--------|---------|
| 串口 | 波特率 | 115200 | AT+UART |
| | 数位 | 8 | |
| | 停止位 | 1 | |
| | 奇偶校验 | 无 | |

6.1 进入AT指令

| | |
|----|---------------------|
| 指令 | AT |
| 功能 | 进入AT指令模式 |
| 发送 | AT |
| 返回 | <CR><LF>+OK<CR><LF> |

【举例】

发送：+++ // 没有换行符

发送：AT // 没有换行符

响应：<CR><LF>AT MODE<CR><LF>

6.2 退出AT指令

| 指令 | EXAT |
|----|---------------------|
| 功能 | 退出AT指令模式 |
| 设置 | AT+EXAT<CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |

【举例】

发送：AT+EXAT\r\n

响应：<CR><LF>+OK<CR><LF>

6.3 查询版本

| 指令 | VER? |
|----|---------------------------|
| 功能 | 查询固件版本 |
| 查询 | AT+VER?<CR><LF> |
| 返回 | <CR><LF> VER=x. x<CR><LF> |
| 备注 | x. x 版本号 |

【举例】

发送：AT+VER? \r\n

响应：<CR><LF> VER=x. x <CR><LF>

6.4 恢复默认参数

| 指令 | RESTORE |
|----|---------------------|
| 功能 | 恢复设备的默认参数（出厂参数） |
| 设置 | AT+RESTORE<CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 备注 | 需要重启设备是参数生效 |

【举例】

发送：AT+RESTORE \r\n

响应：<CR><LF>+OK<CR><LF>

6.5 回显设置

| 指令 | E |
|----|-------------------------|
| 功能 | 用户命令回显设置/查询 |
| 设置 | AT+E=ON<CR><LF><CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 备注 | ON(开启) OFF(关闭) |

【举例】

设置:

发送: AT+E=OFF\r\n

响应: <CR><LF>+OK<CR><LF>

查询:

发送: AT+E?\r\n

响应: <CR><LF>+OK<CR><LF>

6.6 串口参数

| 指令 | UART |
|----|---|
| 功能 | 设置模块串口通讯的参数 |
| 设置 | AT+UART=baud, date, stop, parity, flowcontrol |
| 返回 | <CR><LF>+OK=<snString><CR><LF> |
| 查询 | AT+UART? |
| 参数 | <p>Baud (串口波特率):</p> <p>600,1200,2400,4800,9600,14400,19200,38400,43000,57600,76800, 115200, 128000, 230400, 256000, 460800 单位: bps</p> <p>date: 8</p> <p>stop: 1,2</p> <p>parity: NONE,EVEN,ODD.</p> <p>flowcontrol: NFC(无流控), FC(流控),</p> |

【举例】

设置:

发送: AT+UART=115200, 8, 1, EVEN, NFC\r\n

响应: <CR><LF>+OK<CR><LF>

查询:

发送: AT+UART?\r\n

响应: <CR><LF>+OK<CR><LF> AT+UART=115200, 8, 1, EVEN, NFC <CR><LF>

6.7 设置/查询CAN信息

| 指令 | CAN |
|----|--|
| 功能 | 设置CAN接口通讯参数 |
| 设置 | AT+CAN =baud, id, mode<CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 查询 | AT+CAN? |
| 参数 | Baud (CAN波特率): 6K, 10K, 20K, 50K, 100K, 120K, 125K, 150K, 200K, 250K, 400K, 500K, 600K, 750K, 1000K 单位: bps id (帧ID): 0~7FF (标准帧), 0~1FFFFFF (扩展帧) mode: (帧类别): NDTF (标准帧), EDTF (扩展帧) |

【举例】

设置:

发送: AT+CAN=100, 70, NDTF\r\n

响应: <CR><LF>+OK<CR><LF>

查询:

发送: AT+ CAN?\r\n

响应: <CR><LF>+OK<CR><LF> AT+CAN=100, 70, NDTF <CR><LF>

6.8 设置/查询模块转换方式

| 指令 | MODE |
|----|--|
| 功能 | 设置/查询模块转换方式 |
| 设置 | AT+ MODE=mode<CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 查询 | AT+MODE? |
| 参数 | mode (模块工作模式): TRANS (透明), TPRTL (透明带标识), PROTOL (协议模式), USER (自定义协议), MODBUS (MODBUS), |

【举例】

设置:

发送: AT+CANLT=ETF\r\n

响应: <CR><LF>+OK<CR><LF>

查询:

发送: AT+ CANLT?\r\n

响应: <CR><LF>+OK<CR><LF> AT+CANLT=ETF<CR><LF>

6.9 设置/查询CAN总线的滤波方式

| 指令 | CANLT |
|----|---|
| 功能 | 设置/查询CAN总线的滤波方式 |
| 设置 | AT+CANLT =mode<CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 查询 | AT+CANLT? |
| 参数 | mode (滤波模式):OFF(接收所有功能), ETF(只接收扩展帧), NTF(只接收标准帧), USER (自定义) |

【举例】

设置:

发送: AT+MODE=MODBUS\r\n

响应: <CR><LF>+OK<CR><LF>

查询:

发送: AT+ MODE?\r\n

响应: <CR><LF>+OK<CR><LF>AT+MODE=MODBUS <CR><LF>

6.10 设置/查询帧头帧尾数据

| 指令 | UDMHT |
|----|--------------------------------------|
| 功能 | 设置/查询自定义模式下帧头帧尾数据 |
| 设置 | AT+UDMHT=head, tail<CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 查询 | AT+ UDMHT? |
| 参数 | head (帧头数据), tail (帧尾数据)。数据范围 0~0xFF |

【举例】

设置: 设置帧头数据为FF, 帧尾数据为55

发送: AT+UDMHT=FF, 55 \r\n

响应: <CR><LF>+OK<CR><LF>

查询:

发送: AT+UDMHT?\r\n

响应: <CR><LF>+OK<CR><LF> AT+UDMHT=FF, 55<CR><LF>

6.11 设置/查询标识参数

| 指令 | RANDOM |
|----|---|
| 功能 | 设置/查询标识参数 |
| 设置 | AT+RANDOM = idLength, idLocation <CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 查询 | AT+RANDOM? |
| 参数 | idLength (帧头ID长度), idLocation (帧ID位置)。 数据范围: 长度范围0-4 位置0-7 |

【举例】

设置: 设置帧ID长度4, 位置2

发送: AT+RANDOM=4,2 \r\n

响应: <CR><LF>+OK<CR><LF>

查询:

发送: AT+RANDOM?\r\n

响应: <CR><LF>+OK<CR><LF> AT+RANDOM=4,2 <CR><LF>

6.12 设置/查询标识参数

| 指令 | MSG |
|----|---|
| 功能 | 设置/查询帧ID帧信息使能 |
| 设置 | AT+MSG =flag_id, flag_type<CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 查询 | AT+MSG? |
| 参数 | flag_id (帧头数据), tail (帧尾数据)。数据范围 0~0xFF |

【举例】

设置: 使能帧ID, 帧信息

发送: AT+MSG=1,1 \r\n

响应: <CR><LF>+OK<CR><LF>

查询:

发送: AT+MSG?\r\n

响应: <CR><LF>+OK<CR><LF> AT+MSG=1,1<CR><LF>

6.13 设置/查询传输方向

| 指令 | DIRECTION |
|----|---|
| 功能 | 设置/查询帧ID帧信息使能 |
| 设置 | AT+DIRECTION= parameter<CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 查询 | AT+ DIRECTION? |
| 参数 | parameter (方向参数)，UART-CAN (串口到can)。CAN-UART (CAN到串口) BOTHWAY (双向) |

【举例】

设置：只转换串口数据到can总线

发送：AT+DIRECTION=UART-CAN\r\n

响应：<CR><LF>+OK<CR><LF>

查询：

发送：AT+ DIRECTION?\r\n

响应：<CR><LF>+OK<CR><LF> AT+DIRECTION=UART-CAN <CR><LF>

6.14 设置/查询过滤参数

| 指令 | FILTER |
|----|--|
| 功能 | 设置/查询过滤帧信息 |
| 设置 | AT+FILTER=id_type, date<CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 查询 | AT+FILTER? |
| 参数 | type (帧类别)，date (帧数据) type: NDTF 代表此条命令为标准ID，EDTF代表此条命令为扩展帧ID date: ID 数据。 |

【举例】

设置：设置帧过滤参数：标准帧ID，719

发送：AT+LFILTER=NDTF,719 \r\n

响应：<CR><LF>+OK<CR><LF>

查询：将会返回已经设置的所有ID

发送：AT+ FILTER?\r\n

响应：<CR><LF>+OK<CR><LF> AT+LFILTER=NDTF,719 <CR><LF>

6.15 删除已经设置过滤参数

| | |
|----|---|
| 指令 | DELFILTER |
| 功能 | 设置/查询过滤帧信息 |
| 设置 | AT+DELFILTER=id_type,date<CR><LF> |
| 返回 | <CR><LF>+OK<CR><LF> |
| 参数 | type（帧类别），date（帧数据） NDTF：代表此条命令为标准ID，EDTF 代表此条命令为扩展帧ID。 date：ID 数据。 |

【举例】

设置：删除过滤参数：标准帧 719

发送：AT+DELFILTER=NDTF,719 \r\n

响应：<CR><LF>+OK<CR><LF>

修订历史

| 版本 | 修订日期 | 修订说明 | 维护人 |
|-----|------------|------|-----|
| 1.0 | 2023-09-28 | 初始版本 | LM |

关于我们



销售热线：4000-330-990

技术支持：support@cdebyte.com

公司地址：四川省成都市高新西区西区大道199号B5栋

公司电话：028-61543675

官方网站：www.ebyte.com

