

1. 说在前面：

DrEmpower 系列关节电机 CAN 接口为标准 CAN 2.0A，理论上可适配市面上所有常见 CAN 控制器（包括 USB 转 CAN 模块或微控制器 CAN 接口等）。但是市面上的 USB 转 CAN 模块有很多种，USB 到 CAN 转换的通信协议也各不相同，为此我们针对关节电机专门研发了这款 **USB 转 CAN 模块（如图 1-1）**。

该 USB 转 CAN 模块适配了关节电机配套上位机软件及 CAN 版本的 python 库函数及 ROS 库函数，其他 USB 转 CAN 模块一般情况下无法直接使用配套上位机软件及库函数。



隔离型USB转CAN模块(带TTL接口)

支持USB转CAN/TTL/CAN-FD、TTL转CAN/CAN-FD等多种模式

图 1-1 USB 转 CAN 模块实物图

该 CAN 转 USB 模块购买链接如下：

<https://item.taobao.com/item.htm?spm=a21n57.1.0.0.6fbf523c3v7FQg&id=705379289168&ns=1&abbucket=0#detail>

2. 硬件接口及设备驱动

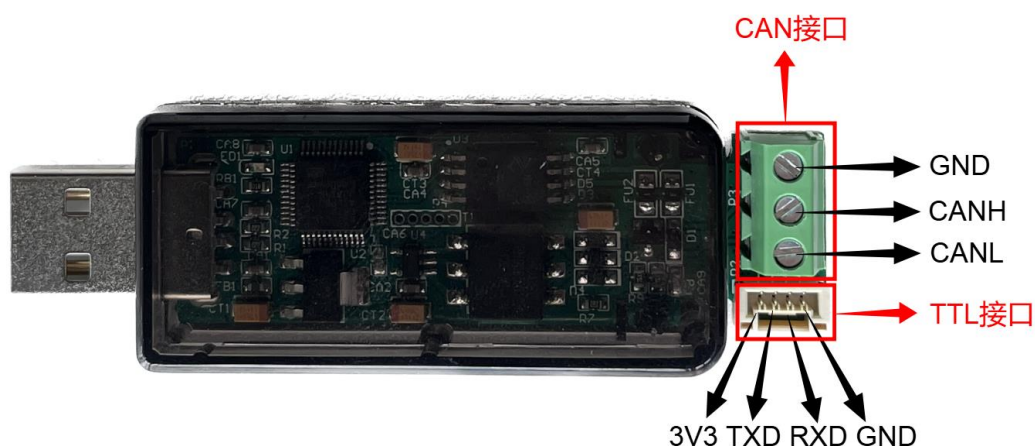


图 2-1 USB 转 CAN 模块硬件接口示意图

该 USB 转 CAN 模块有往外预留了 2 个接口，分别为 CAN 接口（KF128L-3.81-3P）、TTL 接口（GH1.25-4P）。正常情况下，该 USB 转 CAN 模块插入电脑后，在设备管理器中会显示为一个串口设备并分配一个 COM 号（如图 2-2 所示的 COM325）。

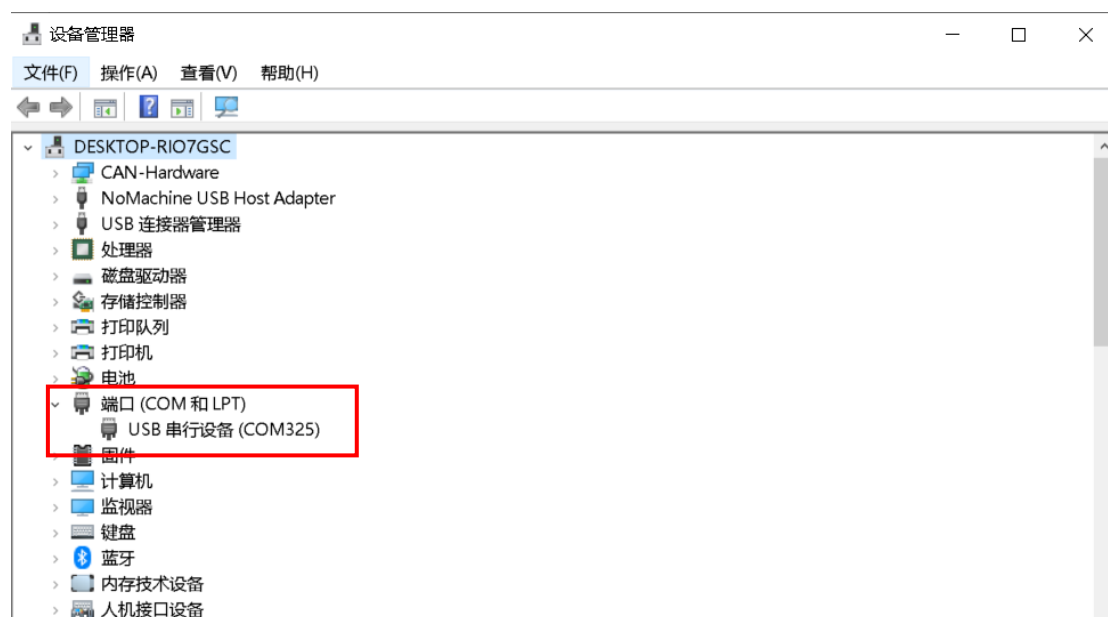


图 2-2 USB 转 CAN 模块在设备管理器中显示

正常情况下，该 USB 转 CAN 模块无需安装驱动即可正常使用（win10 免驱，win7 需要安装驱动）。如果插入电脑后未正常显示成一个串口设备，则需要安装驱动。安装方法也很简单，只需解压驱动文件，双击进行安装即可。

3. 参数配置

3.1 默认配置

正常情况下，该 USB 转 CAN 模块参数无需任何修改，采用出厂默认参数即可，默认参数如下所示。

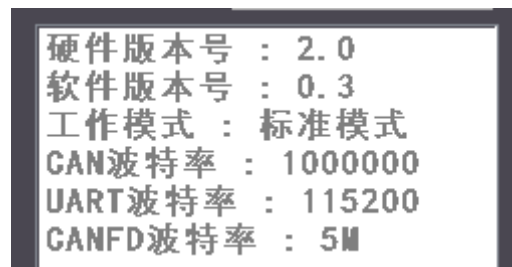


图 2-3 USB 转 CAN 模块出厂默认参数配置

默认参数下，USB 转 CAN 模块的工作模式为**标准模式**，CAN 波特率为 **1M**（关节电机出厂默认 CAN 波特率也为 1M），UART 波特率为 **115200**（关节电机 UART 波特率固定为 115200）。在标准模式下，不仅可以通过 CAN 接口控制电机，正常使用上位机软件及库函数；同时还支持使用 TTL 接口连接关节电机的 UART 接口，也能使用上位机软件和 python 库函数（DrEmpower_uart.py）控制电机。两者的区别是 CAN 接口可以同时控制多个电机，TTL 接口只能控制一个电机（即使多个电机通过 CAN 线相互连接）。

3.2 内部参数

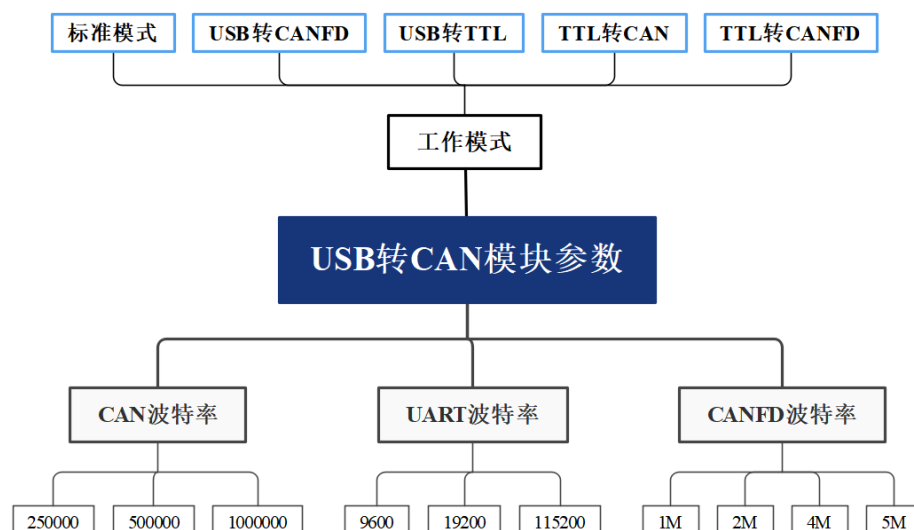


图 2-4 USB 转 CAN 模块内部参数示意图

USB 转 CAN 模块共有 4 个参数可供修改，分别为**工作模式**、**CAN 波特率**、**UART 波特率**、**CANFD 波特率**。该 USB 转 CAN 模块共支持 5 种工作模式，分别为标准模式、USB 转 CANFD、USB 转 TTL、TTL 转 CAN、TTL 转 CANFD。

标准模式为默认模式，专用于 DrEmpower 系列关节电机控制。该模式下主要有两个功能，一个是 USB 转 CAN，另一个是 USB 转 TTL，前者专用于关节电机 CAN 接口控制，后者专用于关节电机 UART 接口控制；两种接口均可使用上位机软件及对应的库函数。此外该模式下还适配了 socketcan 协议，因此不仅支持标准 CAN 库函数，还可使用 linux 系统下专用 CAN 函数（DrEmpower_socketcan.py）和 C++版本 ROS 库函数及 linux 版本上位机软件。另外需要注意的是，该模式下 UART 波特率不能修改，固定为 115200。

USB 转 CANFD 模式下，CAN 协议标准由原来的 CAN 2.0 升级为 CAN FD，两种协议标准的主要区别在于最大传输速率和最大数据长度不同。CAN 2.0：最大传输速率 1Mbps，最大数据长度为 8 个字节。CAN FD：最大传输速率为 5Mbps(数据段)，最大数据长度为 64 字节。另外 CAN FD 协议还向下兼容 CAN 2.0，但是取消了远程帧；目前关节电机暂不支持 CAN FD，后续升级版本会推出支持 CAN FD 协议的关节电机；

USB 转 TTL 模式下，模块的 CAN 接口将失效，此时模块变成一个我们常见的 USB 转 TTL（串口）模块，类似于市面上常见的 CH340 模块。该模式下 UART 波特率可以修改，目前支持 115200、19200、9600 三种常见的波特率。另外需要注意的是，该模式下载传输的串口波特率由模块内部参数 UART 波特率的值确定，与上位机软件（例如串口调试助手）上选择的波特率无关（上位机软件上的波特率可以随便选，不影响传输速率）。

TTL 转 CAN 模式主要用在一些无 CAN 接口但有 UART 接口的控制器上，例如 Arduino 等。这时如果想要控制关节电机，一种方法是直接使用关节电机的 uart 接口进行控制，但是只能控制一个电机。如果想要同时控制多个电机，则需要使用 CAN 接口进行控制。这时只需将该模块的工作模式设置为 TTL 转 CAN 模式，然后将控制器的 uart 接口与该模块的 uart 接口连接起来，就可以使用使用 CAN 接口控制多个关节电机；该模块使用的 TTL 转 CAN 通信协议为关节电机的 UART 通信协议（每条串口指令由 11 个字节组成），CAN 指令固定为数据长度为 8 字节的标准数据帧，详见下文第 4 小节的协议转换部分。该模式下供电方式有两种：一种是通过 USB 供电；另一种是通过 UART 接口的 3.3V 给该模块供电（供电电流需 $\geq 50\text{mA}$ ）。

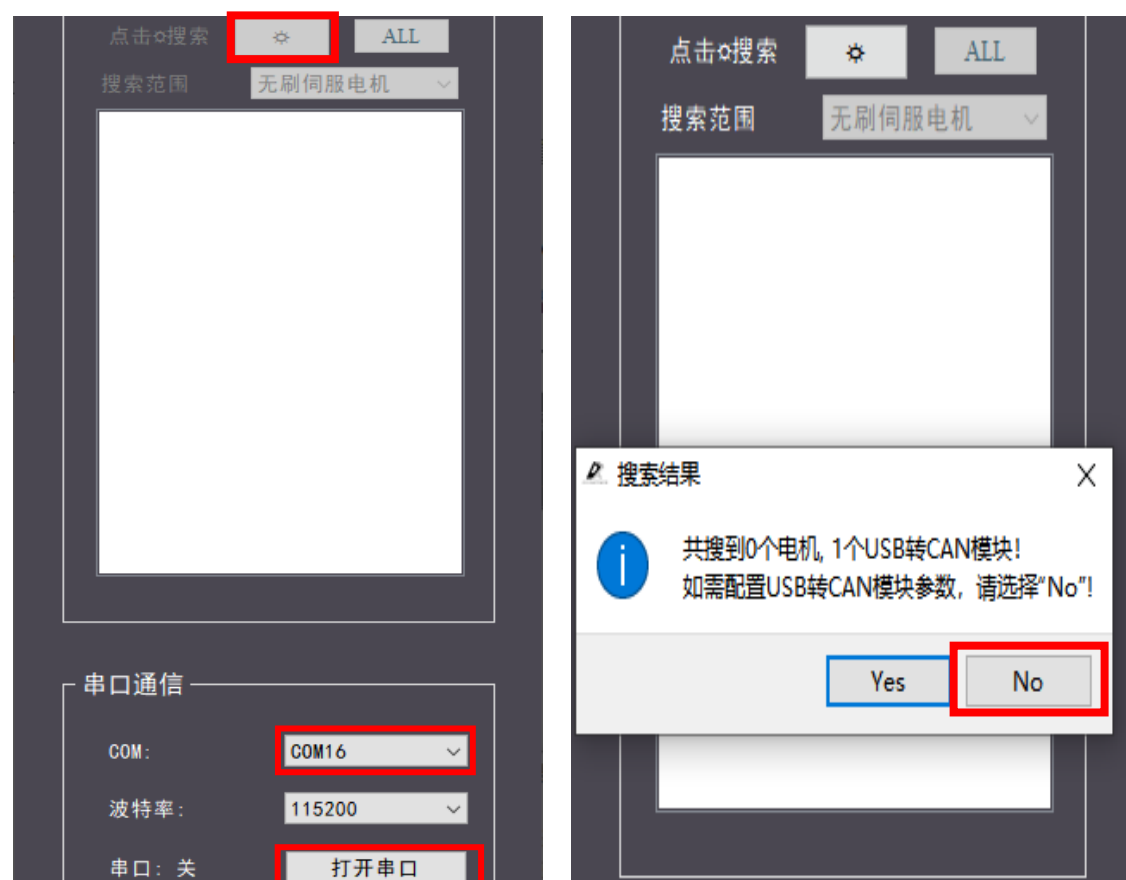
TTL 转 CANFD 模式跟 TTL 转 CAN 类似，只是 CAN 协议由 CAN2.0 升级为 CAN FD，同时通信协议与 TTL 转 CAN 也不同。该模式下支持多种报文帧，例如扩展帧，同时数据长度也可变，支持所有 CAN FD 支持的数据长度。

3.3 参数修改

修改 USB 转 CAN 模块参数功能目前已经集成到最新版本的关节电机上位机软件中。使用方法如下：

(1) 首先将 USB 转 CAN 模块插入电脑，**同时将模块与关节电机之间连线全部断开**（如果能搜索到电机，则不会出现图 3-1(b) 所示参数修改界面入口）。

(2) 然后打开上位机软件进入关节电机搜索界面（如图 3-1(a)所示），选择 USB 转 CAN 模块对应的 COM 口，点击【打开串口】按钮，串口打开后，点击上方【🔍】按钮进行搜索，当出现图 3-1(b)所示弹出的提示框，选择【No】就可以进入 USB 转 CAN 模块参数设置界面。



(a)上位机搜索界面

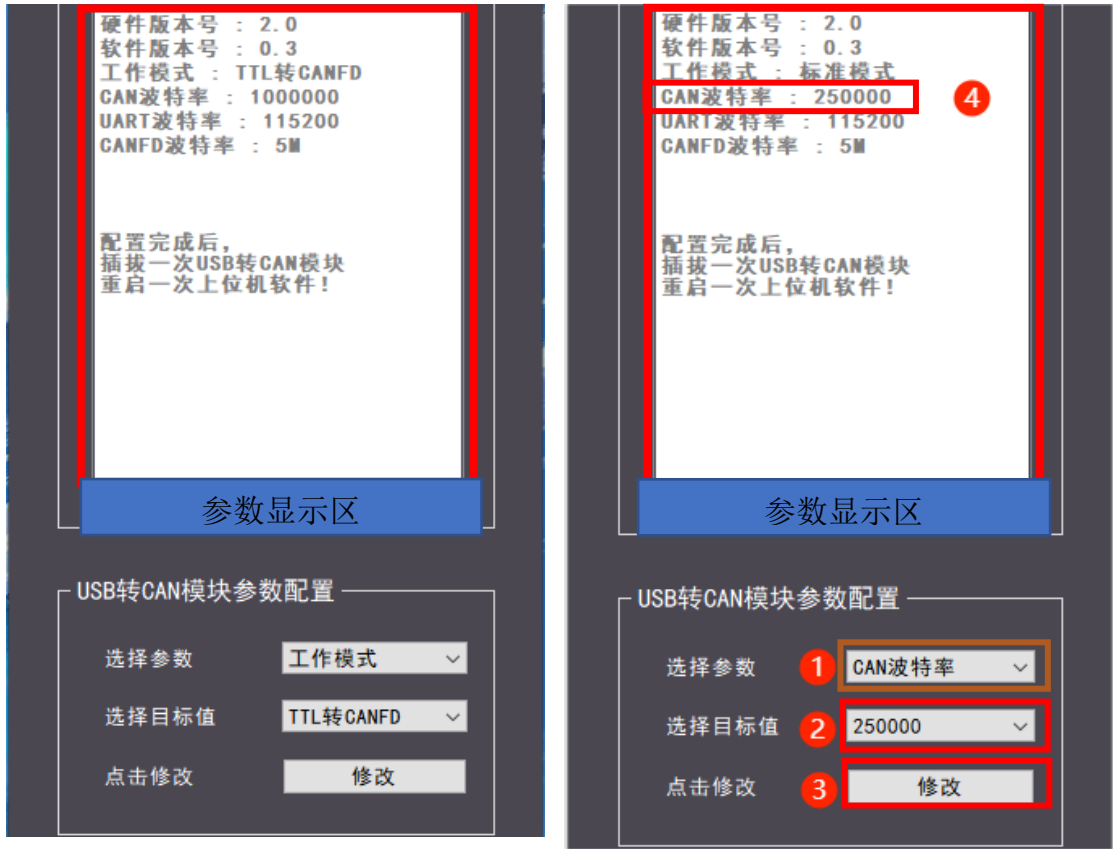
(b) 弹出提示框后点击“NO”

图 3-1 USB 转 CAN 模块搜索并进入参数配置界面

USB 转 CAN 模块参数设置界面如图 3-2(a)所示，界面中间空白区域显示是模块参数当前值及软硬件版本号。下面以修改 CAN 波特率为例（将 CAN 波特率设置为 250000）进行讲解 USB 转 CAN 模块参数修改流程：

- ①在选择参数下拉菜单中选中【CAN 波特率】；
- ②在选择目标值下拉菜单中选中【250000】；
- ③点击下方的【修改】按钮，即可完成修改；

- ④如果修改成功,参数显示区对应参数右侧的值会自动更新(如图 3-2(b)所示),否则可能未修改成功,此时可点击上方【✖】按钮再次进行搜索并查看参数是否更新。如果参数值未变,可按照上述步骤再次修改一遍;
- ⑤参数修改成功后,重启上位机软件,重新插拔一次 USB 转 CAN 模块即可正常使用;



(a) 参数设置界面显示

(b) CAN 波特率配置

图 3-2 USB 转 CAN 模块参数设置界面

4. 协议转换

4.1 TTL 转 CAN 协议

TTL 转 CAN 协议专用于模块工作在 TTL 转 CAN 模式下，UART 接口与 CAN 接口之间数据转换的协议。该协议下 CAN 只能发送和接收 CAN 标准数据帧报文（关节电机控制所用的 CAN 指令均为标准数据帧），而且数据长度固定为 8。对于 UART 接口而言，串行帧的长度为固定的 11 字节，第 1 位为包头 0x08；第 2~3 位为 CAN 报文的帧 ID，其中帧 ID1 为帧 ID 高位，帧 ID2 为帧 ID 低位；第 4~11 位为 8 字节数据段。

串行帧中的帧 ID 和数据信息根据电机控制指令确定，具体参考《DrEmpower 电机 CAN 通讯协议说明 v2.0》。在使用控制器的 UART 接口进行电机控制时，有两种方法。

第一种是将控制器(或 USB 转 TTL 模块)的 UART 接口与关节电机的 UART 接口直接相连，通过 UART 通信协议（G）控制电机，即根据《DrEmpower 电机 CAN 通讯协议说明 v2.0》，确定下图中串行帧中的帧 ID 和数据段，然后通过 UART 直接发送给电机即可。电机收到串口指令后，会执行相应的控制指令。不过这种方法只能控制一个电机，且只适用于 G 系列的驱动板的关节（M 系列串口控制协议为 ascii 协议）。

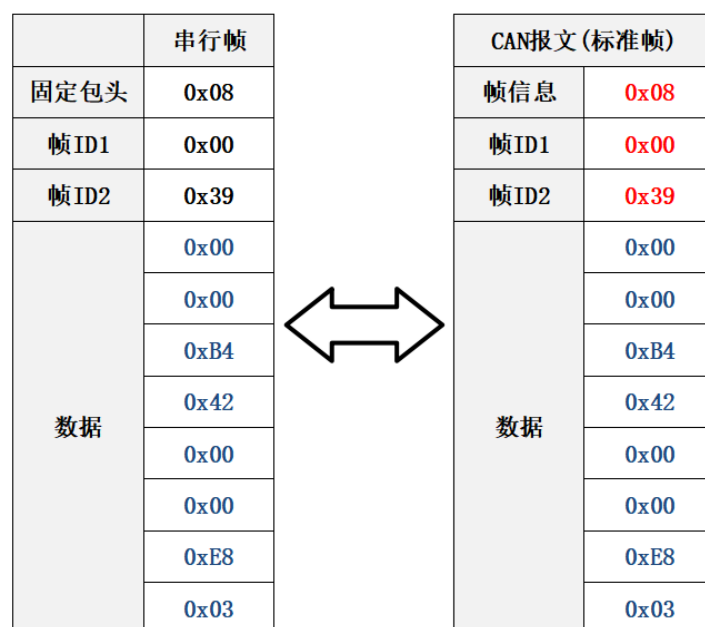


图 4-1 TTL 转 CAN 协议转换示例

第二种是利用 USB 转 CAN 模块(TTL 转 CAN 模式下)，将控制器的 UART 接口与 USB 转 CAN 模块的 UART 接口相连，然后将 USB 转 CAN 模块的 CAN

接口与电机相连。控制器端依旧是按照《DrEmpower 电机 CAN 通讯协议说明 v2.0》，确定下图中串行帧中的帧 ID 和数据段，然后往 UART 接口发出即可。USB 转 CAN 模块接收到 UART 接口数据后，会往 CAN 接口发送对应数据的 CAN 报文，关节电机收到 CAN 指令后，会执行相应的控制指令。这种方法的好处是可以控制多个电机，只需将多个电机通过 CAN 线连接起来即可。

上述两种方法虽然实现原理不同，但是对于控制端来说，其往 UART 接口发送的数据是一样的，所以其库函数也是一样的。可使用的库函数包括 python 版本库函数 `DrEmpower_uart.py`，及 STM32 和 arduino 平台下 UART 版本对应的 `uart` 协议库函数。

4.2 USB/TTL 转 CANFD 协议

USB/TTL 转 CANFD 协议为一种全功能通用转换协议，可支持多种 CAN/CANFD 报文。因为 CAN FD 向下兼容 CAN 2.0A(除了不支持远程帧)，该协议不仅支持 CANFD 协议标准，还支持 CAN 2.0 协议标准。与相比上面的 TTL 转 CAN 协议相比，其数据段长度 0xN 不是固定的，支持 0~8 及仅 CANFD 报文支持的 12、16、20、24、36、48、64。因此整个串行帧的长度也不是固定的，总长度为数据段长度 0xN+8。前面固定长度的 8 字节依次为固定包头 0xAA，扩展帧标识、报文标识、数据长度及 4 个字节的帧 ID。

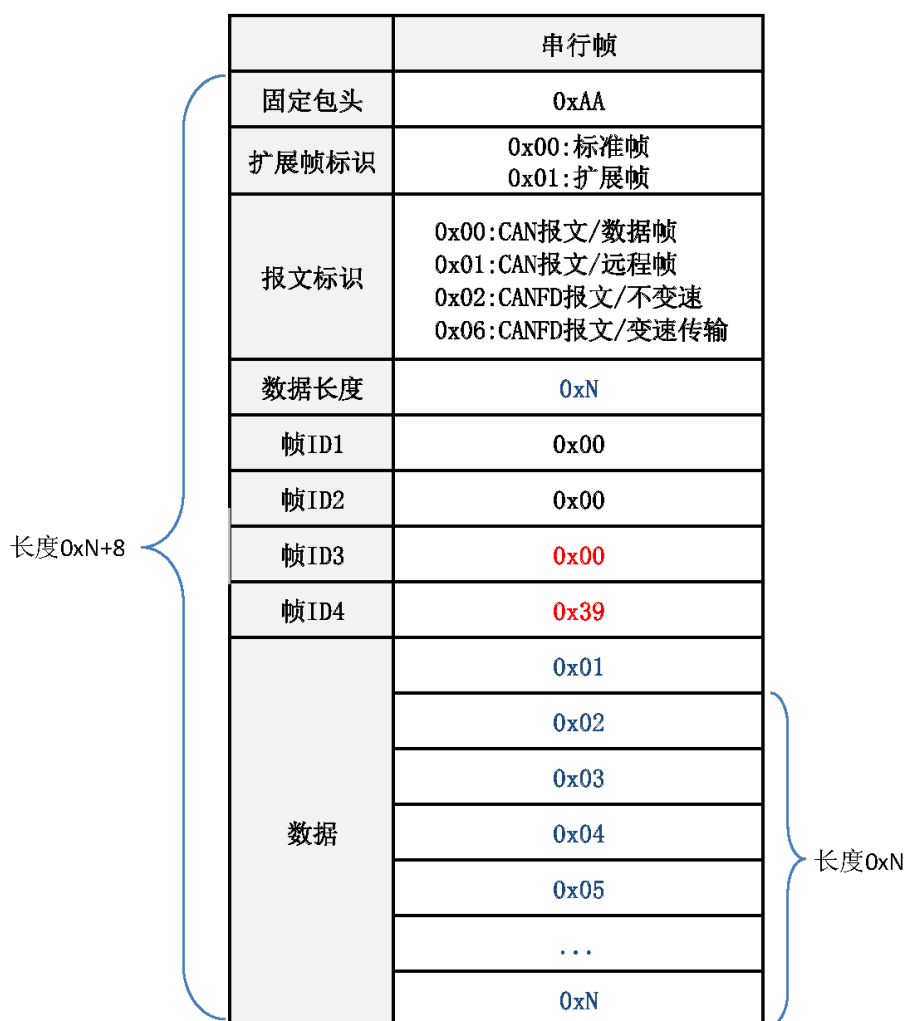


图 4-2 USB/TTL 转 CANFD 协议示例

其中扩展帧标识用来指定传输的 CAN/CANFD 报文是标准帧还是扩展帧。报文标识包含多种信息，常用的主要有 4 种，如图 4-2 所示。前两种为 CAN 报文，后两种为 CANFD 报文。其中 CANFD 报文/不变速指的是发送的为 CAN FD 数据帧，同时 BRS 位为显性（不转换速率），此时 CAN FD 报文的仲裁段和数

据段采用相同的 CAN 波特率进行传输；CANFD 报文/变速传输指的是发送的为 CAN FD 数据帧，同时 BRS 位为隐性（可变转换速率使能），此时 CAN FD 报文的仲裁段和数据段采用不同波特率进行传输，仲裁段采用 CAN 波特率进行传输，数据段采用 CANFD 波特率进行传输。

4 个字节的帧 ID 与第 2 位扩展帧标识配合使用。如果为扩展帧，其有效的 CANID 位数为 29，因此需要 4 个字节进行存储，高位在前，低位在后。如果为标准帧，CANID 有效位为 11，只需使用 4 个字节中的后两个，同样也是高位在前，低位在后，此时帧 ID 前两个字节为固定值 0x00。

USB/TTL 转 CANFD 协议在多种工作模式下使用，包括标准模式、USB 转 CANFD 模式及 TTL 转 CANFD 模式。下面两张图片分别为 CAN 报文-标准数据帧转换示例和 CANFD 报文-扩展数据帧/变速传输转换示例，以供参考。

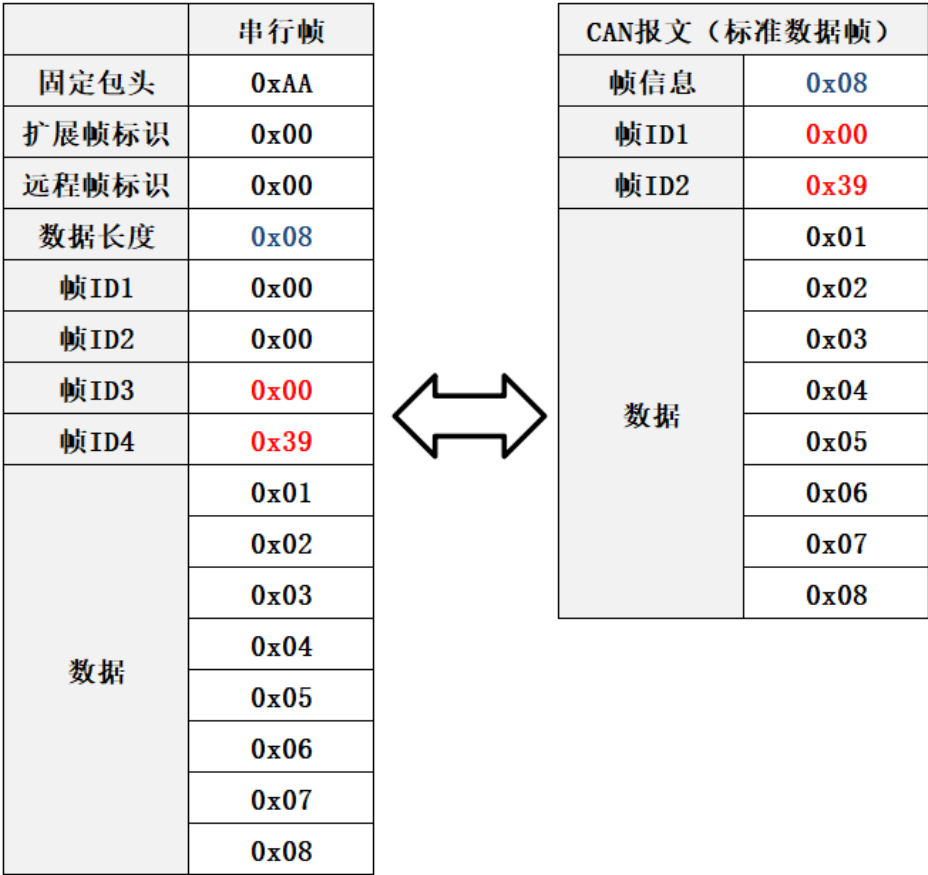


图 4-3 USB/TTL 转 CAN 报文示例

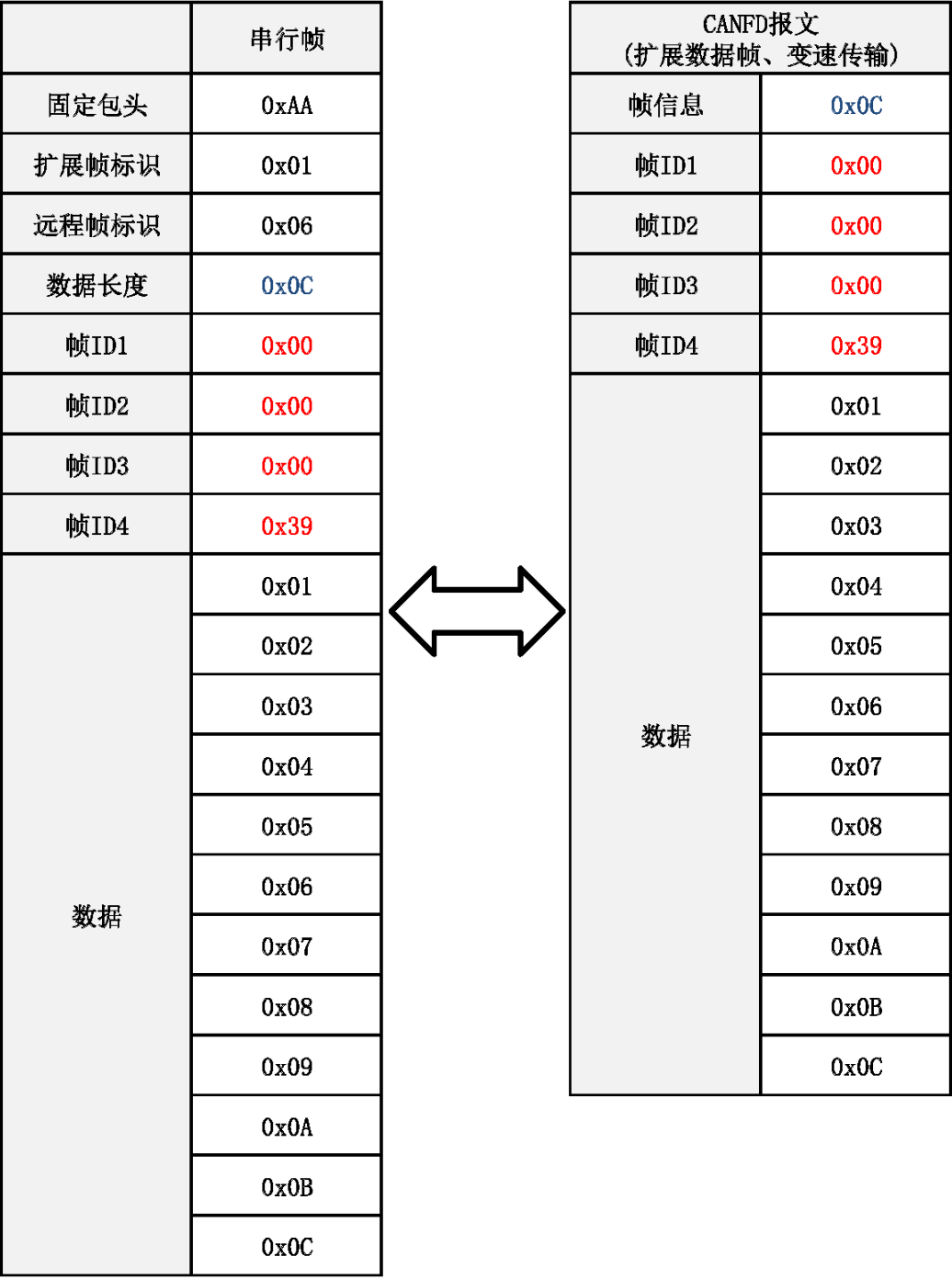


图 4-4 CANFD 报文转换示例

4.3 slcan 协议

SLCAN (Serial Line CAN) 协议是一种用于通过串行线路传输 CAN (Controller Area Network) 数据的协议。它允许将 CAN 总线连接到计算机或其他设备的串行端口, 以便进行 CAN 数据的收发和监控。

该协议主要用在 linux 平台上, 通过如下命令将其转换成 linux 下常用的 socketcan 接口, 用于支持 python 库函数 (DrEmpower_socketcan.py) 及 C++ 版本 ROS 库函数。slcan 协议在标准模式及 USB 转 CANFD 模式下可以使用。

```
sudo slcand -o -c -s0 /dev/ttyACM0 can0
sudo ifconfig can0 up
sudo ifconfig can0 txqueuelen 1000
```