

详细介绍下 USBcan 上位机和下位机程序结构+协议

USB 的厂家及 id: idProduct = 0x5750, idVendor=0x0483

一、协议

所有帧的数据长度都为 64 字节。

1.1 计算机发送给 STM32

表 1:

字节	说明
Byte0	固定为 0x01
Byte1	0 : Byte2-Byte63 为 CAN 参数相关 1-4: Byte2-Byte63 为 CAN 数据帧数, 标记为 N

表 2: Byte1=0 的情况下

字节	说明		
Byte2	0: 读取参数	1: 设置参数	Byte2=0 和 1 , STM32 都会向电脑 返回数据, 返回格 式见表 5
Byte3	忽略数据	波特率的高 8 位	
Byte4		波特率的低 8 位	
Byte5: 63	参数详见 见程序函数 CanInterfceSet		

表 3: Byte1=1-4 的情况下,

字节	说明	
Byte2	Bit7-0: 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	
Byte3	第一帧 CAN 数据-ID 的 32Bits [31: 24]	
Byte4	第一帧 CAN 数据-ID 的 32Bits [23: 16]	
Byte5	第一帧 CAN 数据-ID 的 32Bits [15: 8]	
Byte6	第一帧 CAN 数据-ID 的 32Bits [7: 0]	
Byte7: 14	第一帧 CAN 数据的 8 个 Data 数据	
Byte15	第二帧 CAN 信息	当 N<=1, 忽略
Byte16: 19	第二帧 CAN 数据-ID	
Byte20: 27	第二帧 CAN 数据的 8 个 Data 数据	
Byte28	第三帧 CAN 信息	当 N<=2, 忽略
Byte29: 32	第三帧 CAN 数据-ID	
Byte33: 40	第三帧 CAN 数据的 8 个 Data 数据	
Byte41	第四帧 CAN 信息	当 N<=3, 忽略
Byte42: 45	第四帧 CAN 数据-ID	
Byte46: 53	第四帧 CAN 数据的 8 个 Data 数据	
Byte54: 63	忽略	

1.2 STM32 发送给计算机

表 5 为响应命令，只有当计算机发送了相关帧命令，表位对应帧命令响应。表 6 为 CAN 接口接收到数据主动发送给计算机

表 4:

字节	说明
Byte0	固定为 0x02
Byte1	0 : Byte2-Byte63 为 CAN 参数相关 1-5: Byte2-Byte63 为 CAN 数据帧数，标记为 N

表 5: Byte1=0 的情况下，此表为响应命令

字节	说明
Byte2	0: 读取参数 1: 设置参数
Byte3	波特率的高 8 位
Byte4	波特率的低 8 位
Byte5: 63	参数详见 见程序函数 CanInterfaceSet

表 6: Byte1=1 当 CAN 接口有接收到数据，将数据通过 USB 发送给计算机。

字节	说明	
Byte2	Bit7-0: 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	
Byte3	第一帧 CAN 数据-ID 的 32Bits [31: 24]	
Byte4	第一帧 CAN 数据-ID 的 32Bits [23: 16]	
Byte5	第一帧 CAN 数据-ID 的 32Bits [15: 8]	
Byte6	第一帧 CAN 数据-ID 的 32Bits [7: 0]	
Byte7: 14	第一帧 CAN 数据的 8 个 Data 数据	
Byte15	第二帧 CAN 信息	当 N≤1, 忽略
Byte16: 19	第二帧 CAN 数据-ID	
Byte20: 27	第二帧 CAN 数据的 8 个 Data 数据	
Byte28	第三帧 CAN 信息	当 N≤2, 忽略
Byte29: 32	第三帧 CAN 数据-ID	
Byte33: 40	第三帧 CAN 数据的 8 个 Data 数据	
Byte41	第四帧 CAN 信息	当 N≤3, 忽略
Byte42: 45	第四帧 CAN 数据-ID	
Byte46: 53	第四帧 CAN 数据的 8 个 Data 数据	
Byte54: 63	忽略	

二、USB 转 CAN 的下位机程序结构

