

CANFDBridge-Pro

多路信号复用使用说明

	内容
关键词	CANFDBridge-Pro, 拆分, 多路信号复用
摘要	CANFDBridge-Pro地址复用使用说明

修订历史

版本	日期	原因
V1.0.0	2022/12/7	创建文档

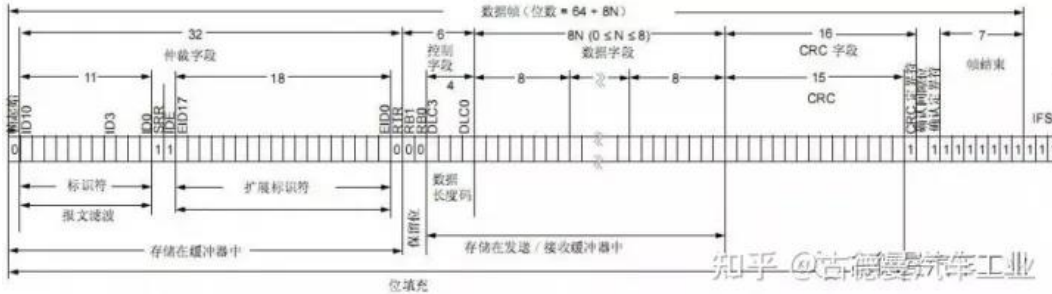
目录

1 功能简介	4
11 拆分功能与地址复用	4
2 操作演示	6
21 配置模块	6
22 数据测试	11
2.2.1 示例1	12
2.2.2 示例2	13
2.2.1 示例3	14

1 功能简介

1.1 拆分功能与地址复用

我们都知道一帧标准的CAN报文拥有一个长度是64bit共8个byte的数据域，



常规的用法是总线工程师根据实际需要，在64个位中定义CAN通讯矩阵。但还有一种非主流的用法，允许message定义一个信号，作为标志位。根据这个信号的值判断切换不同的矩阵方案。这么说有点抽象，还是先举个例子：

	bit1	bit2	bit3	bit4	bit5	bit6	bit7	bit8
byte1	FlagSig							
byte2	Signal_1			Signal_2				
byte2	Signal_3						Signal_4	

假设Byte1中bit1~bit4为信号标记位FlagSig，而Byte2的定义了两种矩阵。一种为bit1~bit3为Signal_1、bit4~bit8为Signal_2；另一种为bit1~bit6为Signal_3、bit7~bit8为Signal4。

正常情况这样定义byte2是会起冲突，这里神奇的地方就是可以根据FagSig的值进行切换。

	bit1	bit2	bit3	bit4	bit5	bit6	bit7	bit8
byte1	FlagSig=0x00							
byte2	Signal_1			Signal_2				

FlagSig为0x00的时候解析Signal1与Signal2

	bit1	bit2	bit3	bit4	bit5	bit6	bit7	bit8
byte1	FlagSig=0x01							
byte2	Signal_3						Signal_4	

FlagSig为0x01的时候解析Signal3和Signal4; 同理还可以继续定义下去

FlagSig=0x02~0x0F所对应的信号, 这里就不进行重复!

如果看完上面的叙述还是不明白, 可以看看下面的代码描。

```
Switch(FlagSig)
{
    Case0xb0: 当前解析信号=Signal1与Signal2;
    Case0x01: 当前解析信号=Signal3与Signal4;
    ...
}
```

而在CANFDBridge-Pro中, 为了解决使用DBC文件发送时候, 相同ID的不同定义的现象, 启用了多路信号复用的拆分功能。拆分功能指的是, 将CANFD帧拆分成多个CAN帧; 多路信号复用指的是, 允许CANFD帧message定义一个信号, 作为标志位, 根据这个信号值的不同, 将相同ID的CANFD帧拆分成不同ID的CAN帧。

多路复用信号

复用使能 复用值

 复用起始位 复用长度

在多路复用模式中, 各个参数定义如下

名称	定义
复用使能	打开\关闭多路信号复用功能
复用值	代表上文中的标记位FlagSig
复用起始位	表示复用值从哪一位开始复用
复用长度	表示复用值位数的长度

比如: 复用值是1, 复用起始位为0, 复用长度为8, 对应的位表示为“0000 0001”

复用值为10, 复用起始位为8, 复用长度为8, 对应的位表示为“0000 0000 0000 1010”

2 操作演示

2.1 配置模块

连接模块将CAN0设置为CANFD模式，如图2.1:

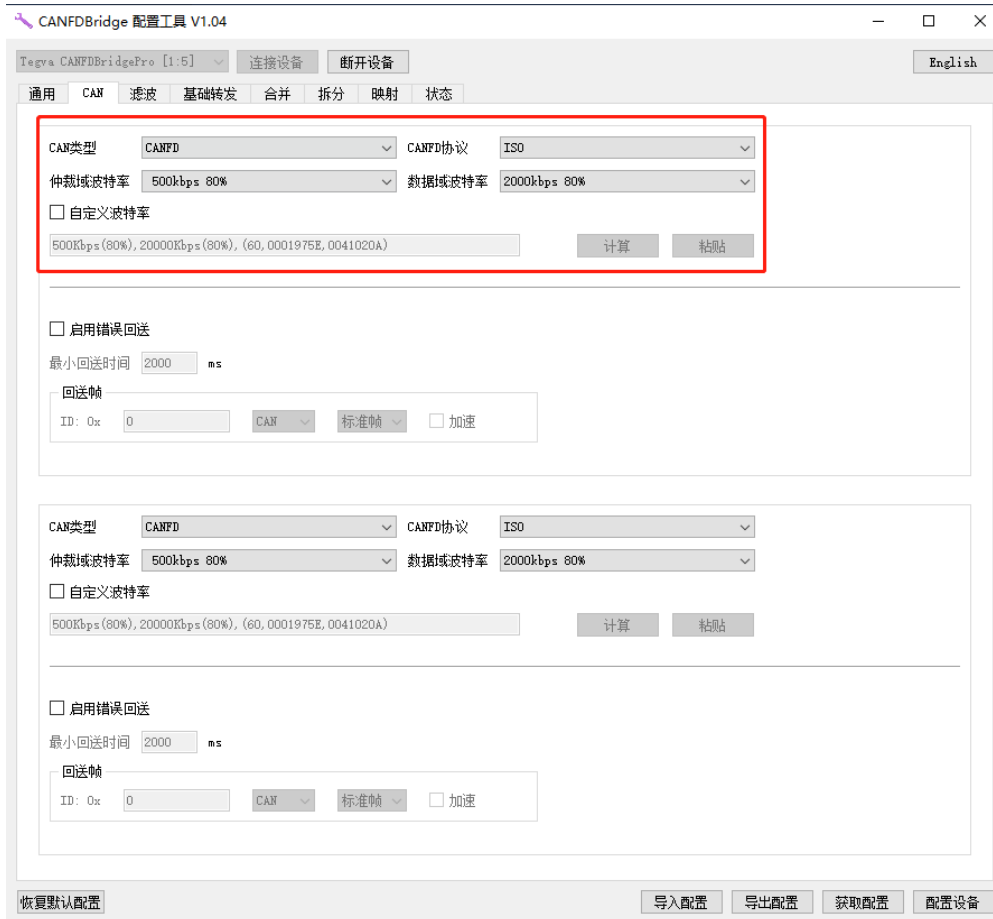


图 2.1 CANFD模式

打开CAN0的拆分功能，点击“增加”按钮，增加一条ID为0x100拆分信息，如图2.2:

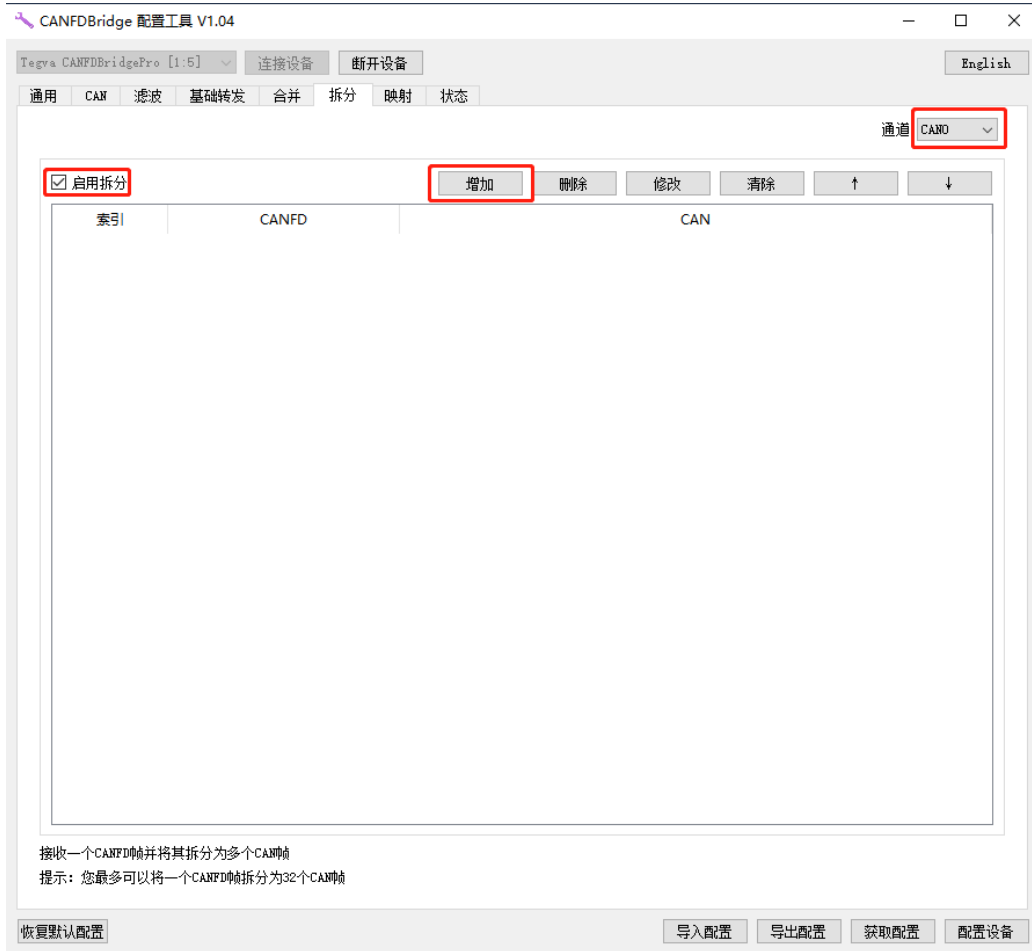


图 2.2 启动拆分

将此CANFD帧拆分成ID为0x11, 0x22, 0x33的CAN帧, 如图2.3, 点击“OK”, 完成添加。



图 2.3 填写拆分信息

点击“增加”按钮，继续添加一条ID为0x100拆分信息，并启用复用功能，如图2.4：

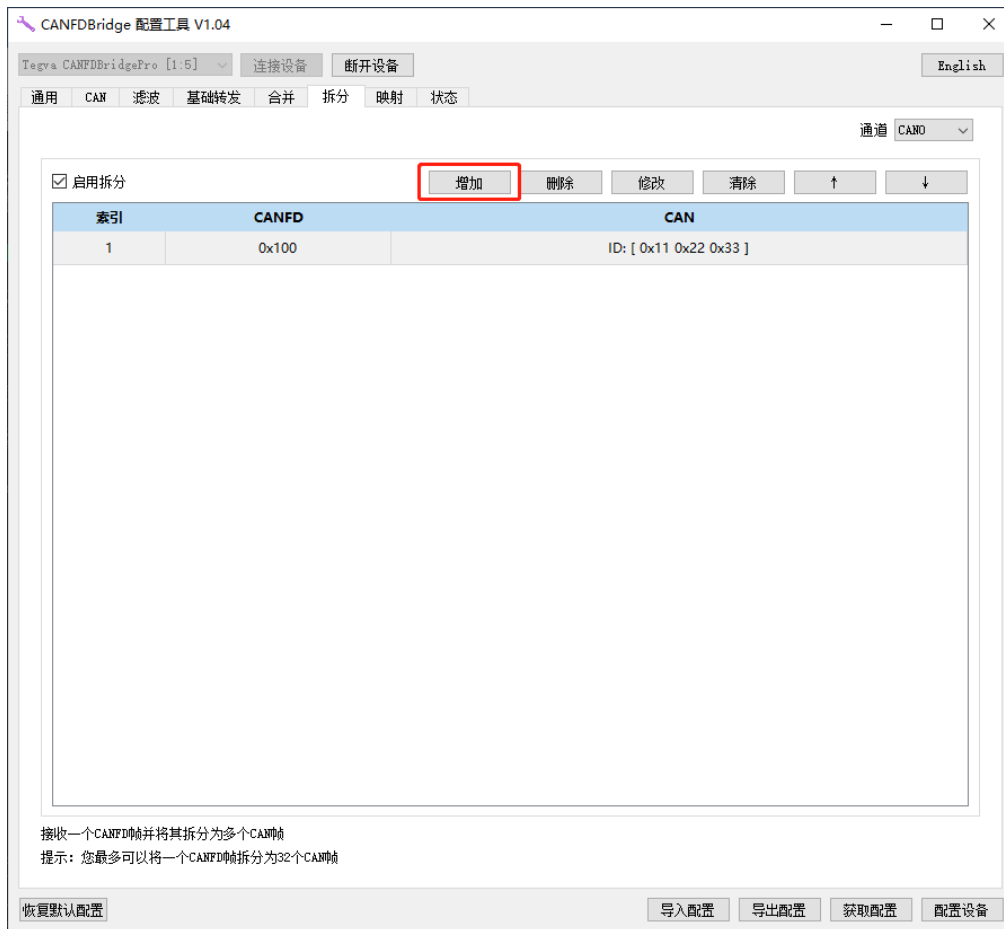


图 2.4 增加拆分信息

将此CANFD帧拆分成ID为0x44, 0x55, 0x66的CAN帧, 设置复用值为5, 复用起始位为0, 复用长度为8, 如图2.5, 点击“OK”, 完成添加。



图 2.5 填写拆分信息

点击“增加”按钮, 继续添加一条ID为0x100拆分信息, 并启用复用功能, 如图2.6:

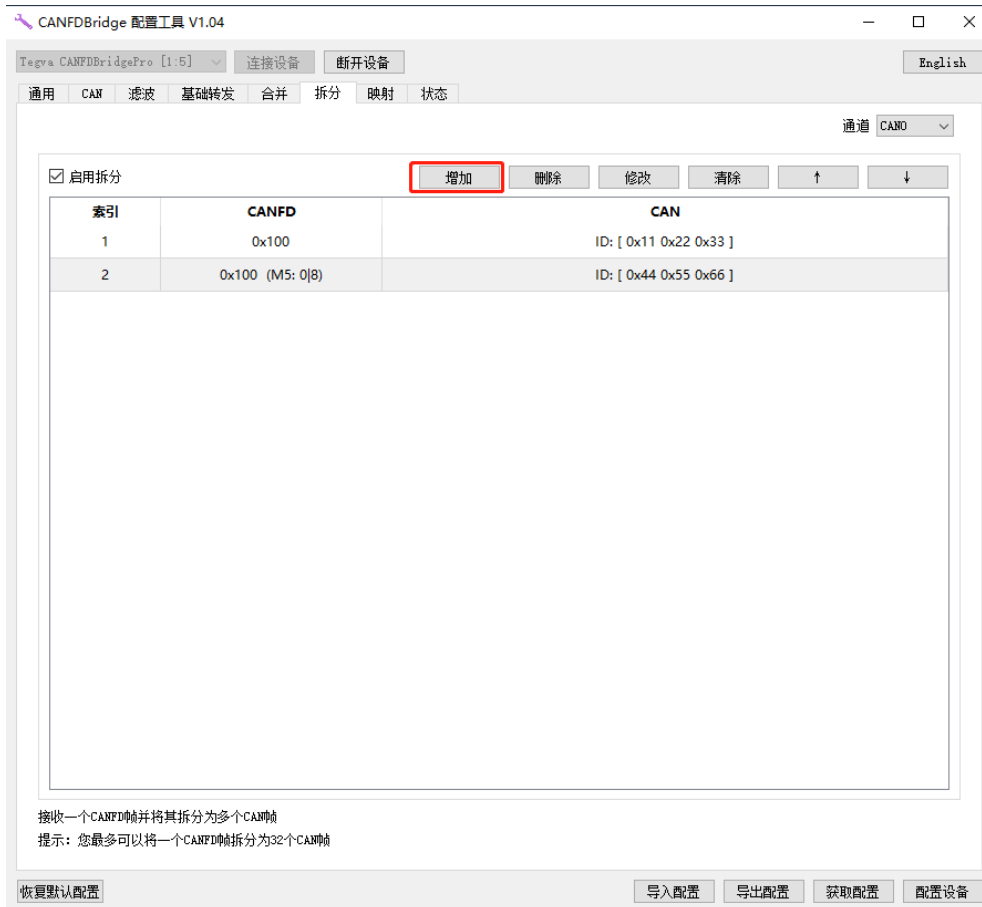


图 2.6 增加拆分信息

将此CANFD帧拆分成ID为0x77, 0x88, 0x99的CAN帧, 设置复用值为10, 复用起始位为8, 复用长度为8, 如图2.7, 点击“OK”, 完成添加。



图 2.7 增加拆分信息

点击“配置设备”按钮, 将此配置保存至模块。如图2.8:

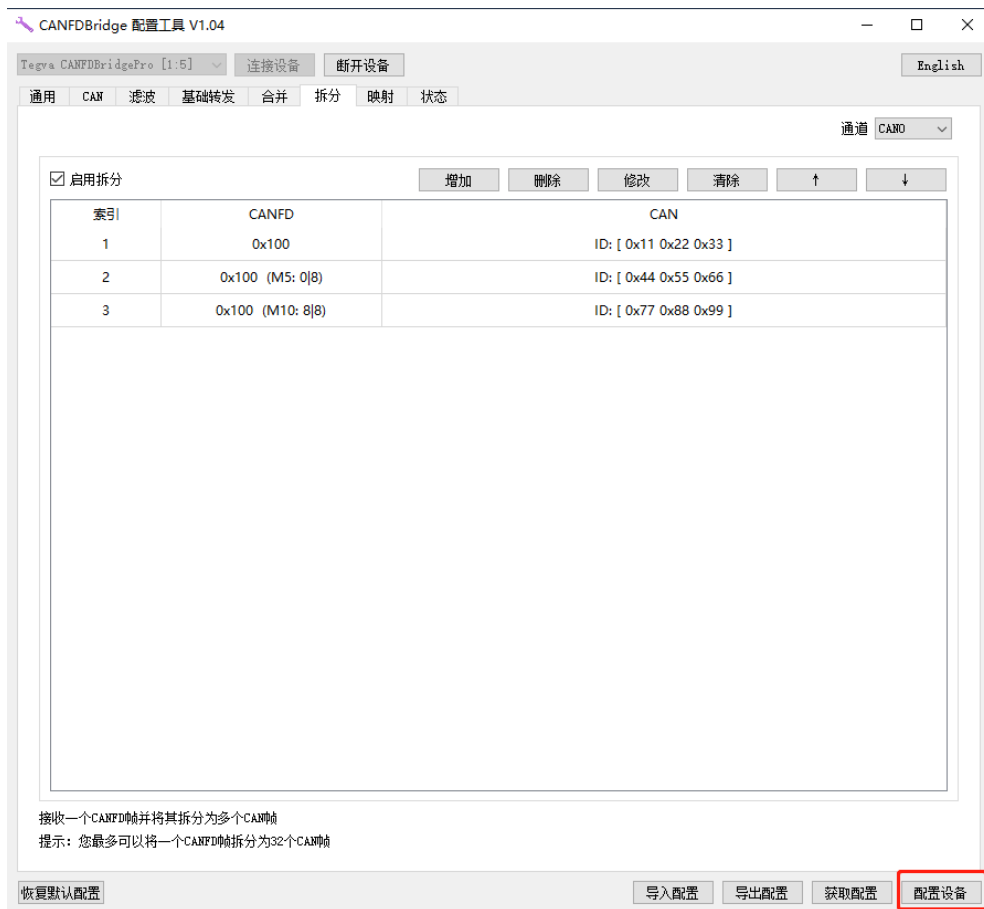


图 2.8 配置设备

2.2 数据测试

如图2.9, 将USBCANFD-II的CAN0、CAN1分别于CANFDBridge-Pro的CAN0、CAN1连


接, 打开  TegcanView, 设备选择USBCANFD-II, 具体配置如图2.10:



图 2.9 连接设备

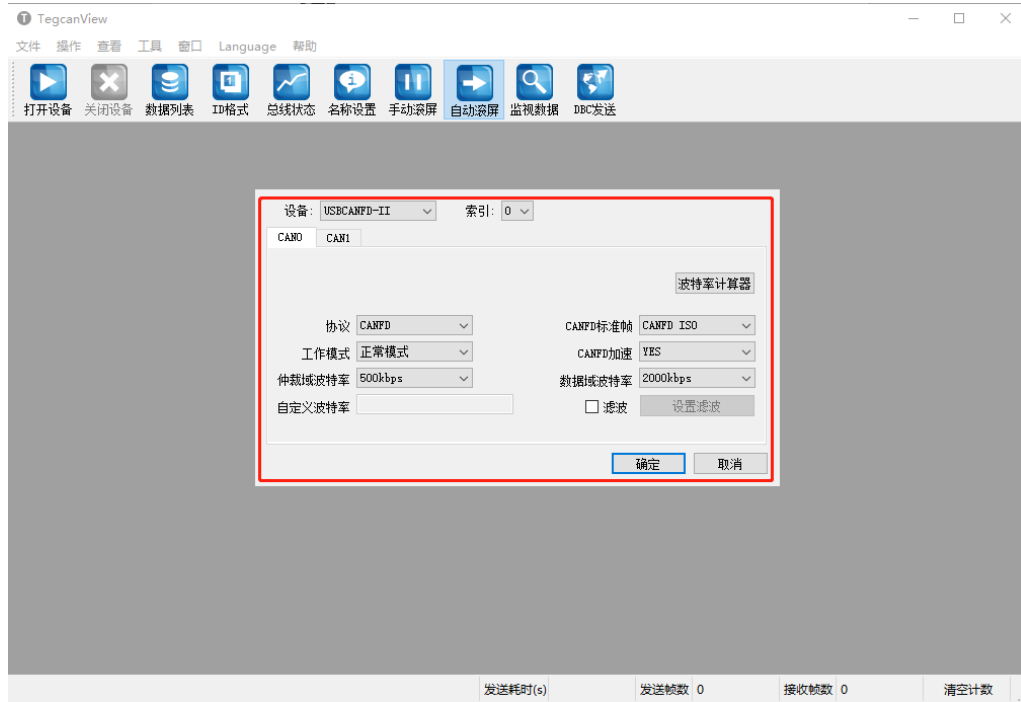


图 2.10 打开设备

2.2.1 示例1

按照第一条拆分信息，CAN0发送ID为0x100，数据为“11 22 33 44 55 66 77 88”的CANFD帧，如图2.11:

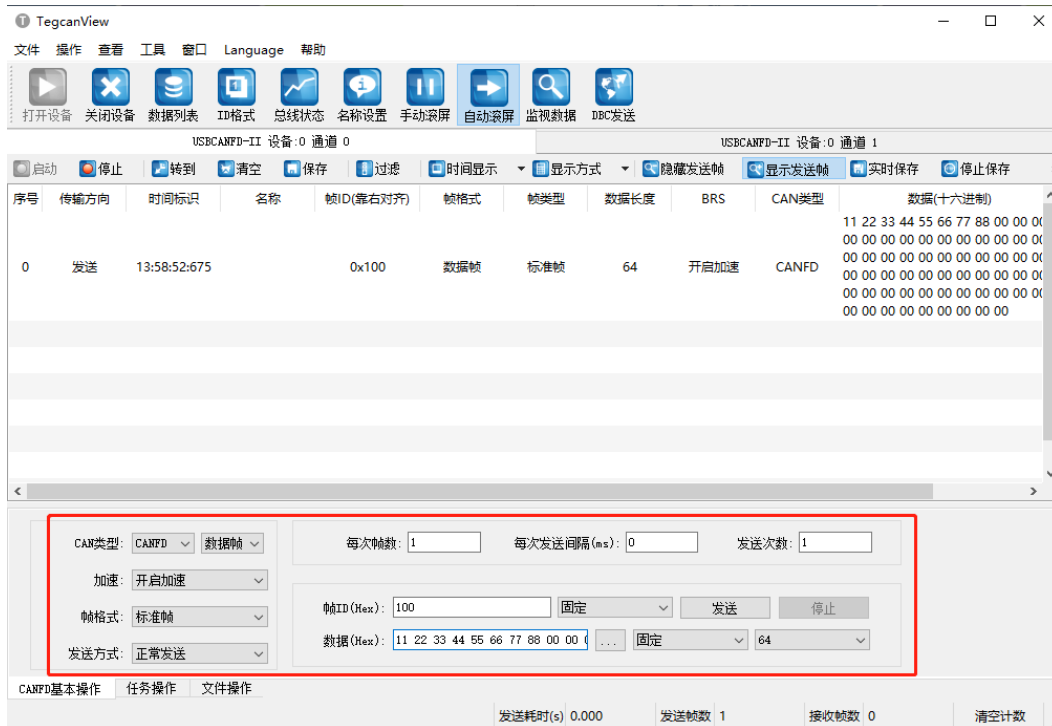


图 2.11 发送消息

此时，CAN1口 接收到ID为0x11, 0x22, 0x33的CAN帧，如图2.12:

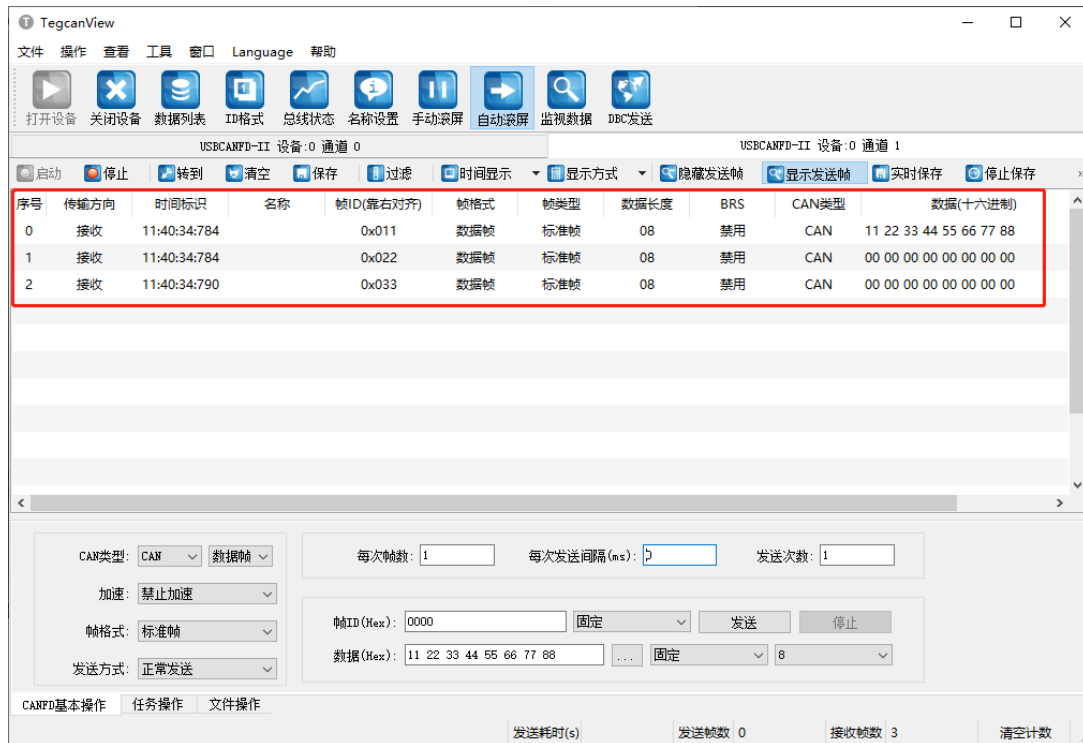


图 2.12 接收消息

2.2.2 示例2

按照第二条拆分信息，CAN0发送ID为0x100，数据为“05 22 33 44 55 66 77 88”的CANFD帧，因为此条信息打开了复用功能，按照配置，复用值为5，复用起始位置为0，长度为8，所以数据的第一个字节为05（注：如果复用无效，尝试将拆分配置表中打开复用功能的信息放在首行），如图2.13：

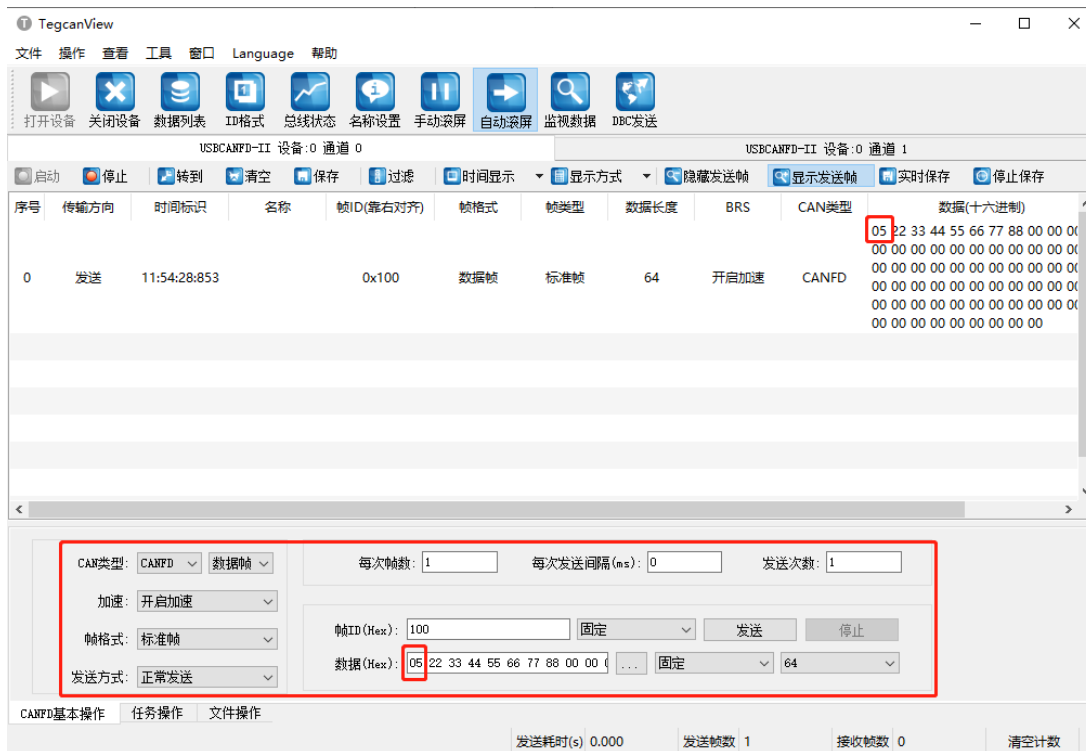


图 2.13 发送消息

此时，CAN1口 接收到ID为0x44, 0x55, 0x66的CAN帧，如图2.14:

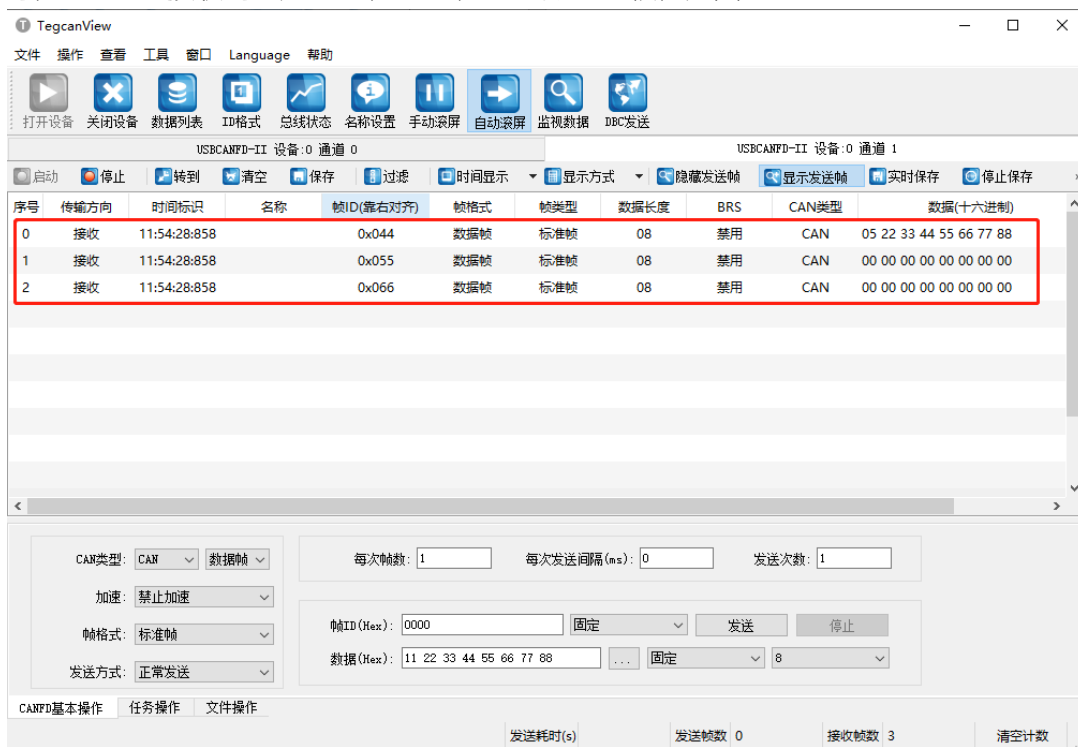


图 2.14 接收消息

2.2.1 示例3

按照第三条拆分信息，CAN0发送ID为0x100，数据为“11 0A 33 44 55 66 77 88”的CANFD帧，因为此条信息打开了复用功能，按照配置，复用值为10，复用起始位置为8，长度为8，所以数据的第二个字节为0A，如图2.15:

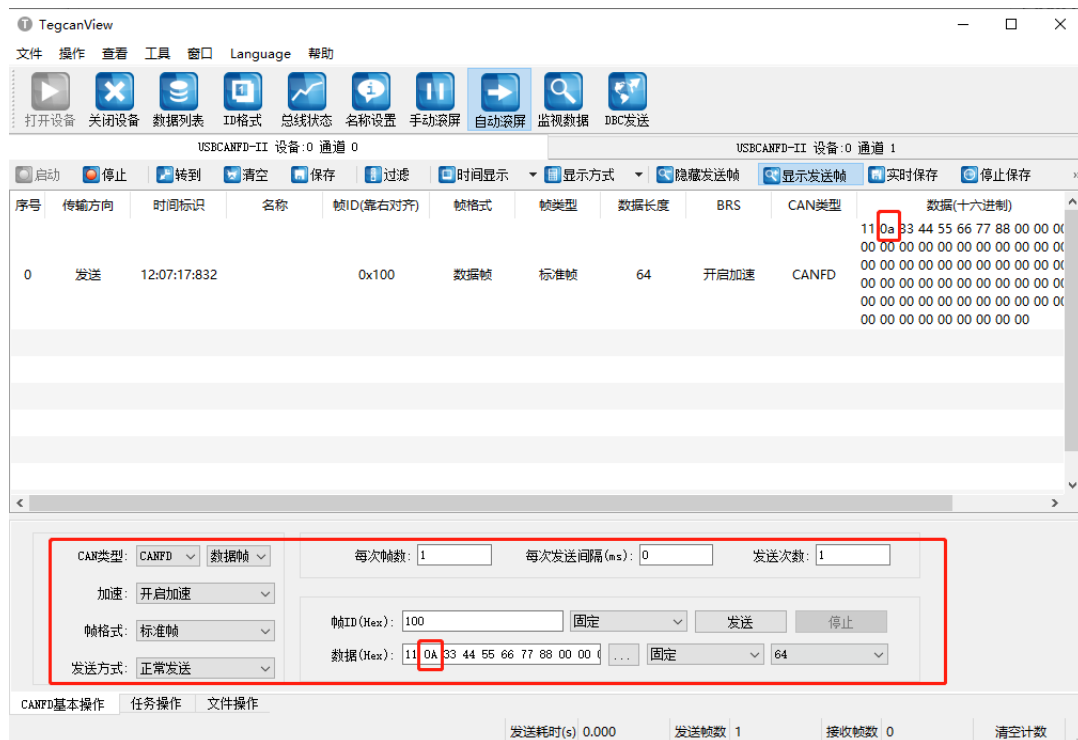


图 2.15 发送消息

此时，CAN1口 接收到ID为0x77, 0x88, 0x99的CAN帧，如图2.16：

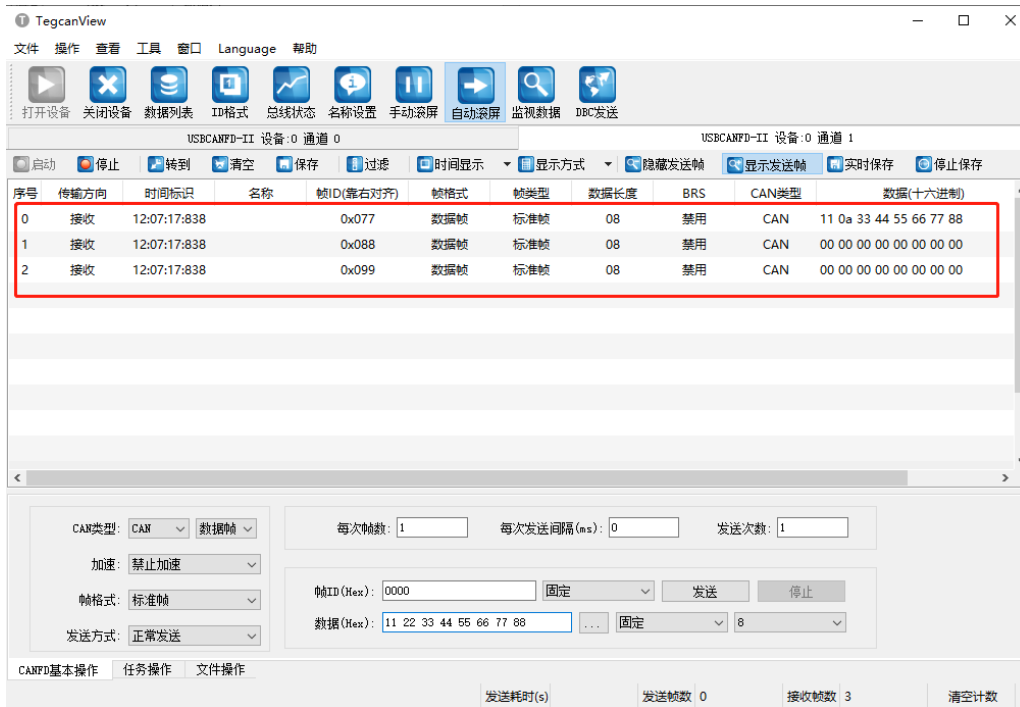


图 2.16 接收消息

备注：如果复用无效，尝试将拆分配置表中打开复用功能的信息放在首行。