

## CAN 总线不能正常通信的原因总结：

### 1. 终端电阻问题。

必须保证 can 总线布线拓扑是手拉手的形式，并且在首尾各有一个  $120\ \Omega$  的电阻。在断电情况下，用万用表测量 CANH 和 CANL 的电阻，正常时候，大约为  $60\ \Omega$ 。

### 2. CAN 线断了，或者接触不良。

这种情况，也出现了不少案例，传统的成品杜邦线，可靠性不好。可靠性要求高的应用上，请勿使用杜邦线。粗略统计客户反馈，已经出现过 20 多例杜邦线接头处没有压好铜芯的情况导致无法正常通信。

### 3. 编译器不能生成正确的程序。

出现了几次，用户的开发环境，编译下载的程序根本就不能运行，导致 can 通信不能正常工作。比如优化过度、keil 工程上芯片没有选择正确。

### 4. 假芯片问题。

市场上有大把的假的 can 芯片，这类芯片完全不能工作。在跑着 can 程序的板上，直接用示波器测 CANH 引脚的波形，如果波形完全是平的，则可以断定芯片为假货，或者芯片已经坏了。

### 5. 芯片供电错误问题。

一些开发板，完全不测试就出货了。TJA1050 此类的芯片，必须要 5V 供电才能正常工作，有些开发板给它供电了 3.3V，然后就拿出来卖了。然后出售开发板的也不做测试是否能用，简直是坑死用户。

### 6. CAN 编组错误。

给的例程，都是针对 0 组的驱动器广播，如果编组不在 0 组，直接下载例程是无法通信的，需要修改程序再使用。最省事的方法，直接对驱动器进行恢复出厂设置，即可使用。

### 7. 串口线数据异常导致编组编号异常。

有一些通信不正常的串口线，把意外的不确定的 id 号写入了驱动器，也会造成无法通信。发生这种情况，直接将驱动器恢复出厂设置即可。

### 8. 波特率不匹配问题。

驱动器默认的波特率是 1M，如果主控程序上的波特率不是 1M，就必然通信异常。

### 9. 晶振问题。

如使用 STM32，有些人设计的 PCB，给的晶振是 8MHz，也有 12MHz, 16MHz, 25MHz。如果硬件上的晶振与程序上的晶振不同，则一定要先修改程序再使用。否则，波特率一定是错的，一定无法通信。另外注意，从淘宝购买的 STM32 开发板，有的厂家原理图上标的晶振与实际焊接的晶振不同，一定要看清楚，此类问题困扰过不少客户，浪费了大量时间去排查问题。

10. 提供的例程，仅针对于主频 72MHz 和 168MHz 的 MCU，如果 MCU 主频设置为其他，则需要修改相应的 CAN 初始化参数。例如：STM32F429，主频 180MHz 的情况下，要修改 CAN 的 CAN\_BSx 寄存器。

11. 给 can 收发器供电的电源纹波过大。出现过一些客户硬件经验不足，自行设计板卡，开关电源布线不考究，导致纹波巨大，造成 can 收发器供电不稳定。此类情况，建议按 datasheet 提供的开关电源布线参考图来进行布线。如信心不足，建议 DC-DC 降压到 5.5V，再用 LDO 降压至 5.0V 给 5V 的 can 收发器供电，LDO 降压至 3.3V 给 3.3V 的 can 收发器供电。
12. 另外建议，为避免 can 通信的收发器之间压差过大导致 can 收发器损坏。CAN 线布线，请多走一根 GND 线，即 can 线为三根线：CANH CANL GND。
13. STM32 接出的 CAN-TX/CAN-RX 与 CAN 收发器的 TX 和 RX，使用直连方式连接是正确的，请勿交叉连接。例如：STM32 的 PB9/PA12 需要接的是 CAN 收发器的 TX，STM32 的 PB8/PA11 需要接的是 CAN 收发器的 RX。
14. 如果主控板是正点原子的开发板，注意 PA11/PA12，也就是 CANTX 和 CANRX 有跳线帽，与 USB 共用，需要修改跳线帽到 CAN 端。
15. 复位指令和模式选择指令之间的延时过小。此会导致模式选择发送出去的时候，驱动器还没初始化 CAN 总线，以至于收不到 CAN 报文。
16. 没有给驱动器正确的供电电压，比如最近两年的 RMDS-10x 供电电压必须在 7-33V 之间，不可以只给 5V 供电。
17. 主控板整板只供电了 3.3V，而 CAN 收发器的额定电压是 5V，CAN 收发器没有得到足够的供电。需要给主控板足够的供电电压，比如额定 12V 的主控板，给 12V 供电。
18. 有一类的开发板，CAN 收发器额定电压为 5V，5V 电压的获取直接是 DC9V 口，属于设计缺陷。当用户直接对 5V 供电口插入了 9V 或者 12V 的电源适配器，会直接烧毁板上 5V 的所有器件。

## 运行官方例程驱动器报错的原因和解决方案：

如果使用 2 个及以上驱动器，连接在同一条 CAN 线上，必须设置为不同的编号，比如 0 组 1 号和 0 组 2 号，如果不设置为不同的编号，一旦让驱动器回传报文，报文 ID 则会相同，就会造成总线致命冲突，此时驱动器会进入硬件错误状态，蜂鸣器响并且闪红灯。

同一条 CAN 总线的不同主机，不能同时发送相同 ID 的报文！

另外运行官方例程，还需要注意，CAN 的 io 口（PA11 PA12 PB8 PB9）和外部晶振是否匹配。

## 如何配合售后调试 CAN 总线？

如果 CAN 总线不通，首先要执行如下：

1. 将驱动器恢复出厂设置，然后退出调试软件，以保证驱动器处于默认编号和默认波特率。
2. 使用从 [www.robomodule.net](http://www.robomodule.net) 下载的例程，任何东西什么都不要改，如果改了什么的话，请告诉售后。

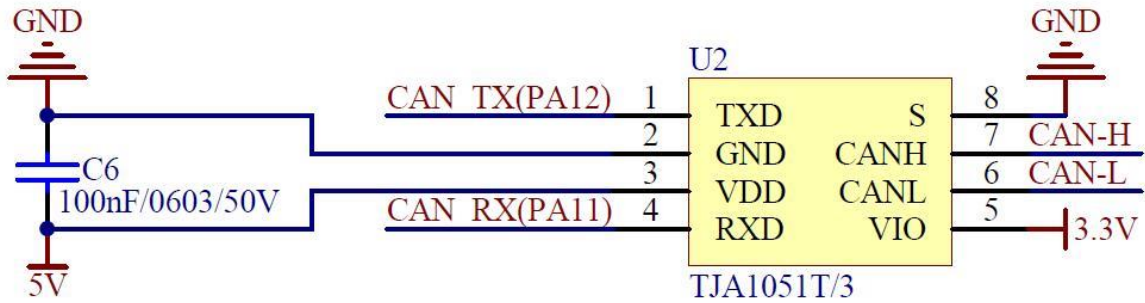
然后应该告诉售后如下信息：

1. 在网站发布的例程里面，你使用的是哪个例程？
2. CAN 收发器用的什么型号？比如 TJA1050 还是 VP230
3. CAN 收发器的供电电压是多少？5V 还是 3.3V？
4. STM32 主控用的什么型号？
5. STM32 主控用的晶振，板上焊接的是多少 MHz 的？ 8MHz 12MHz 25MHz ？
6. STM32 连接 CAN 收发器的 IO 是哪个？ PA11/PA12, 还是 PB8/PB9? 还是其它？
7. 断电后，CANH 与 CANL 使用万用表测量的电阻值是多少？
8. 连接线是否能保证是通的？
9. 联系电话是多少？
10. 驱动器的供电电压是多少，万用表实测值是多少？

几个有效的调试手段：

1. 如何保证程序是正确下载的？在程序里面插入一个 LED 闪烁的程序，看程序是否正确运行？比如，点亮 LED，延时 500ms，熄灭 LED，延时 500ms，重复 2 次。看看 LED 是不是一秒闪一次。
2. 购买 USBCAN 来辅助调试。将 USBCAN 的 CANH CANL 也接入电路中，然后在电脑上的 USBCAN 调试软件上看，主控是否有发送出 CAN 消息，发送出来的 CAN 消息是否符合规则。
3. 如果实在不愿意花钱买 USBCAN，那就用示波器。将主控的 CAN 线拔掉，然后主控上电，用示波器去测量 CANH 对地的波形。如果没有波形，那么就说明主控根本就没发出来任何东西。

## 推荐主控使用的 CAN 硬件原理图



说明：以上原理图，一般来说 STM32 的开发板会自带。如果没有，直接淘宝搜 CAN 收发器，便可以买到类似的，不过最常用的 CAN 收发器还是 TJA1050。