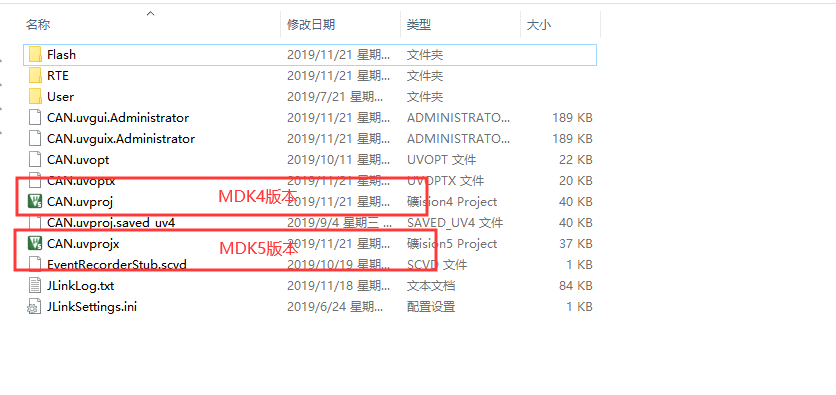
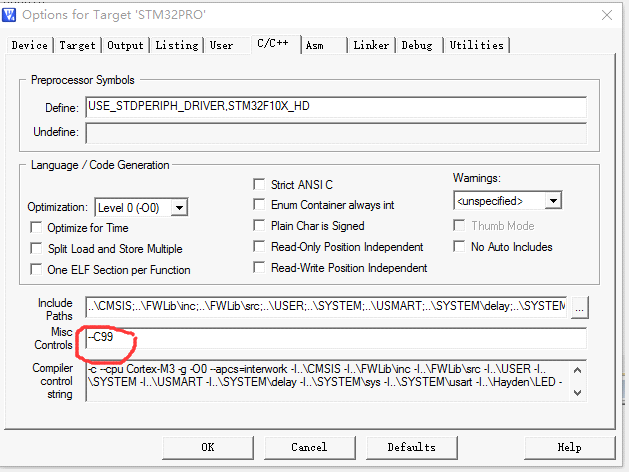
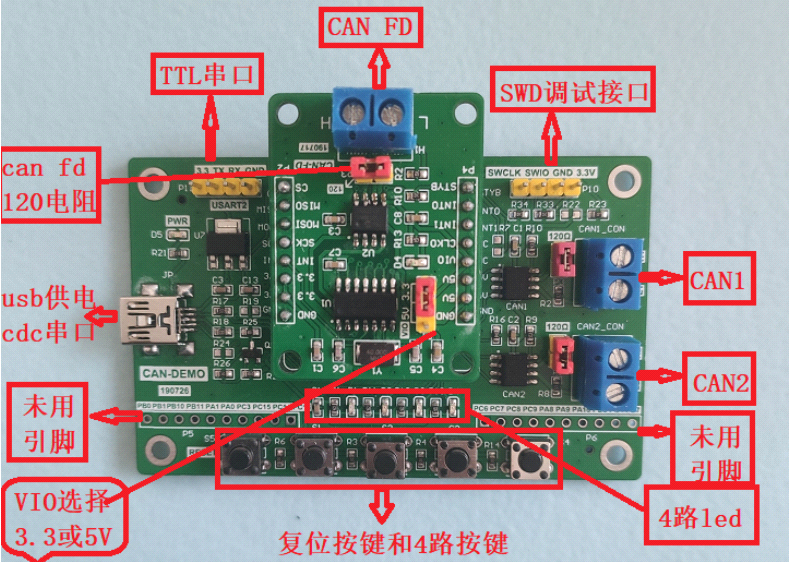
工程结构：

MDK4和MDK5选择不同工程文件打开，使用MDK4打开工程可能会出现编译错误，请按如下配置即可



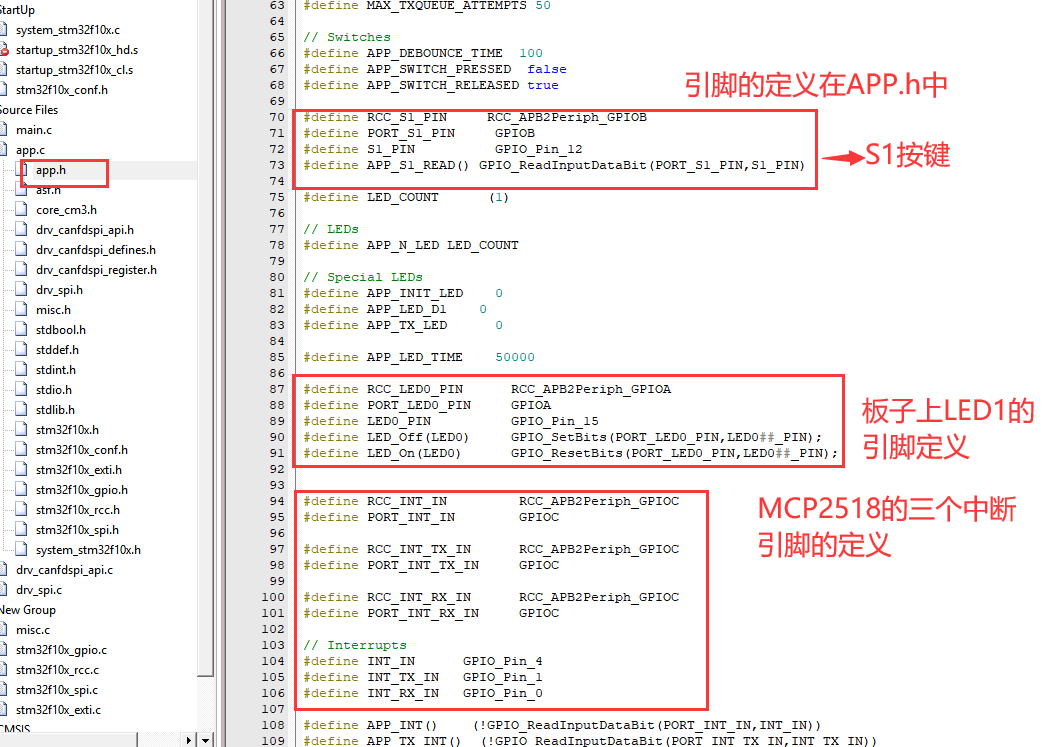


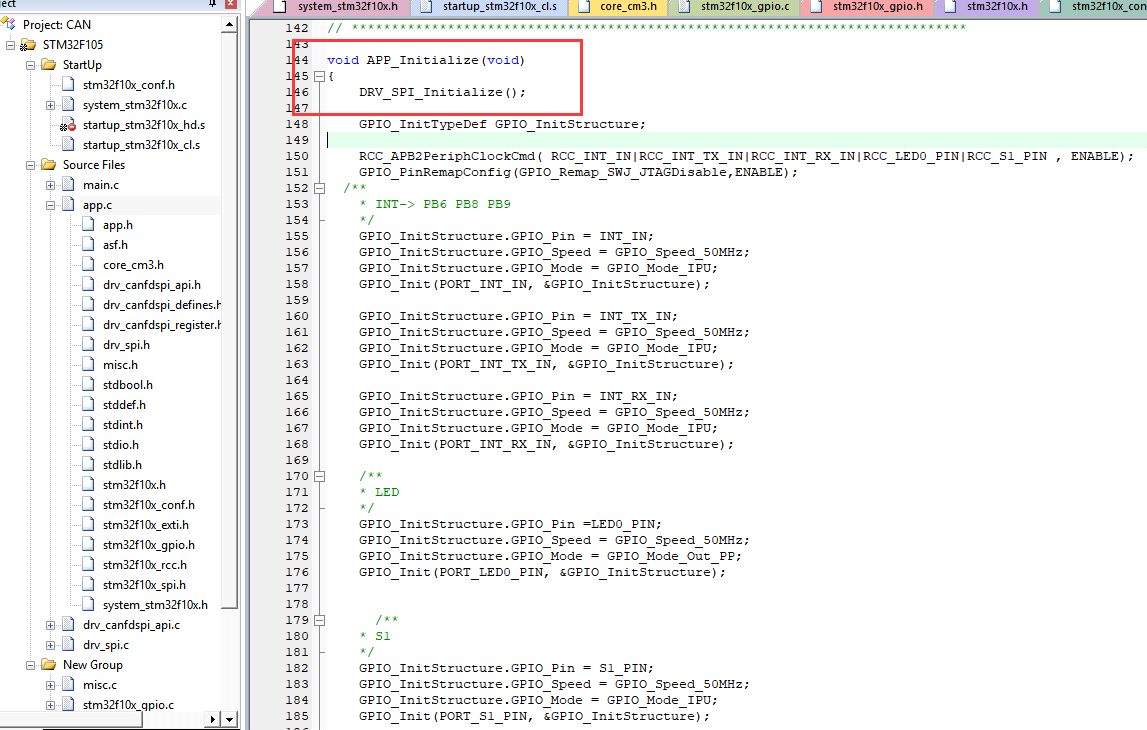
板子硬件接口



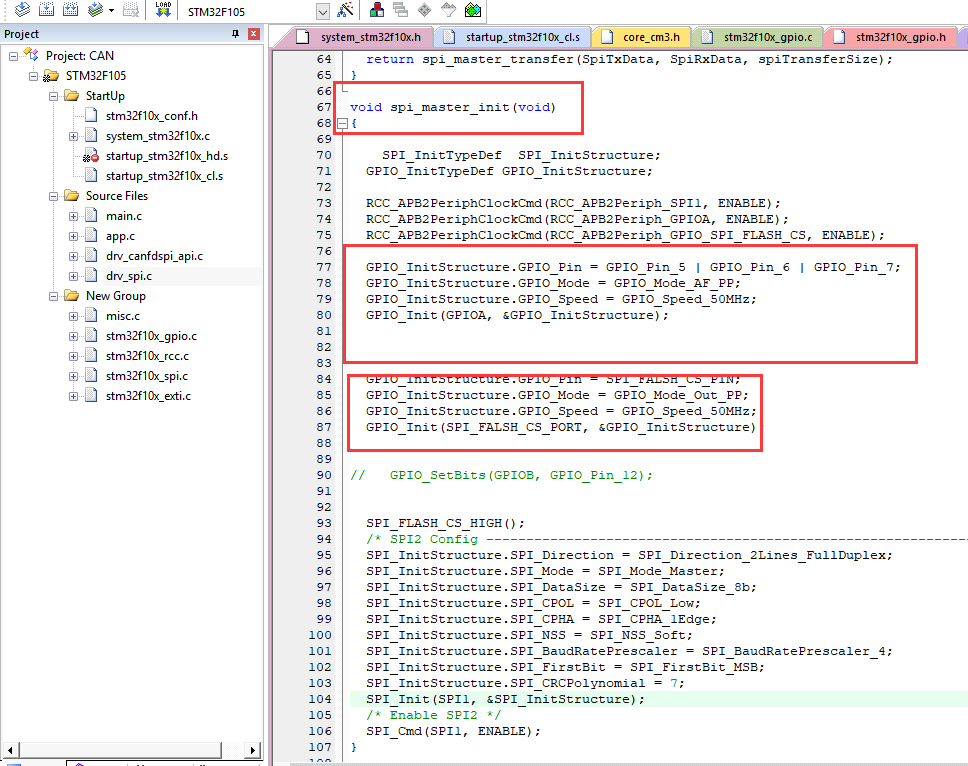
务必使用usb供电，不能使用调试器供电。

1.APP.h文件中端口的宏定义，移植时修改这些定义，初始化在app.c中





2.SPI初始化在drv\_spi.c中



3.APP.c中函数介绍

例程测试，LED闪烁几次后初始化完成，状态机在app\_Task中循环切换。

状态机简介

APP\_STATE\_INIT：CAN FD初始化。

APP\_STATE\_INIT\_TXOBJ：发送数据的初始化。

APP\_STATE\_FLASH\_LEDS：闪LED

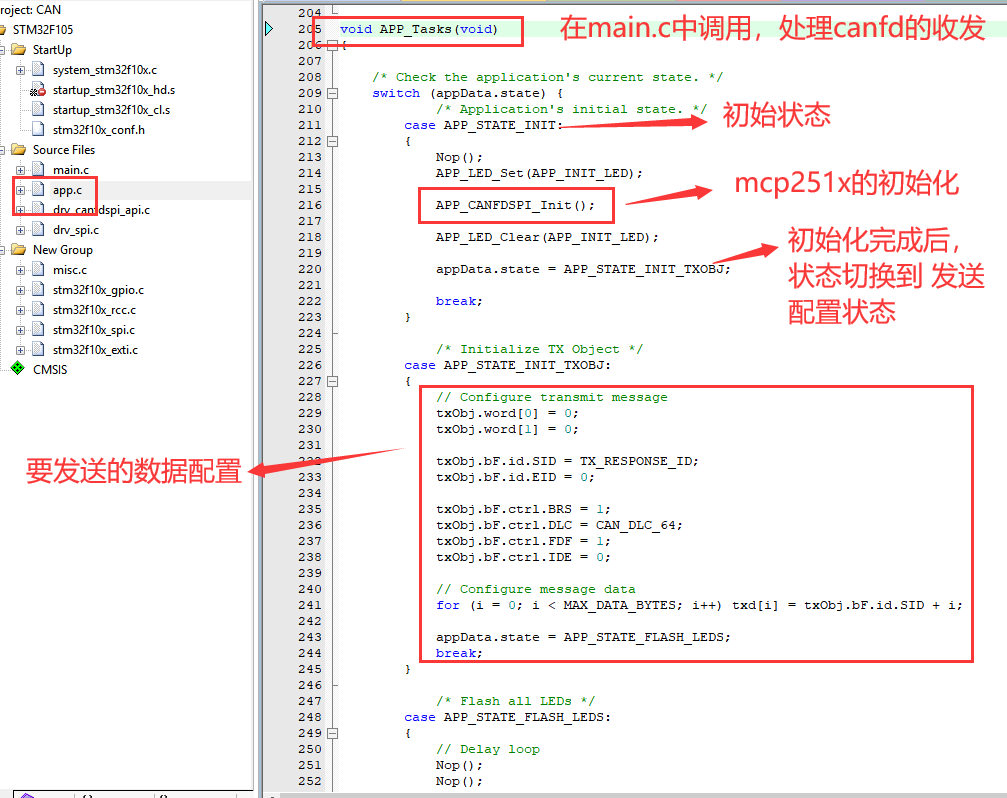
APP\_STATE\_RECEIVE：查询接收到的数据

APP\_STATE\_PAYLOAD：总线负载数据能力的测试

APP\_STATE\_SWITCH\_CHANGED：按键状态改变发送数据

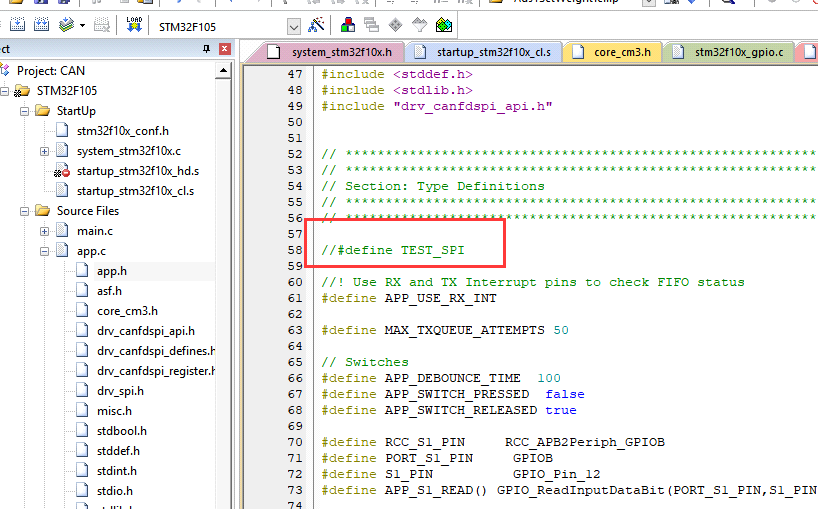
APP\_STATE\_TEST\_RAM\_ACCESS：2517 RAM 读写测试

APP\_STATE\_TEST\_REGISTER\_ACCESS：2517 寄存器读写测试

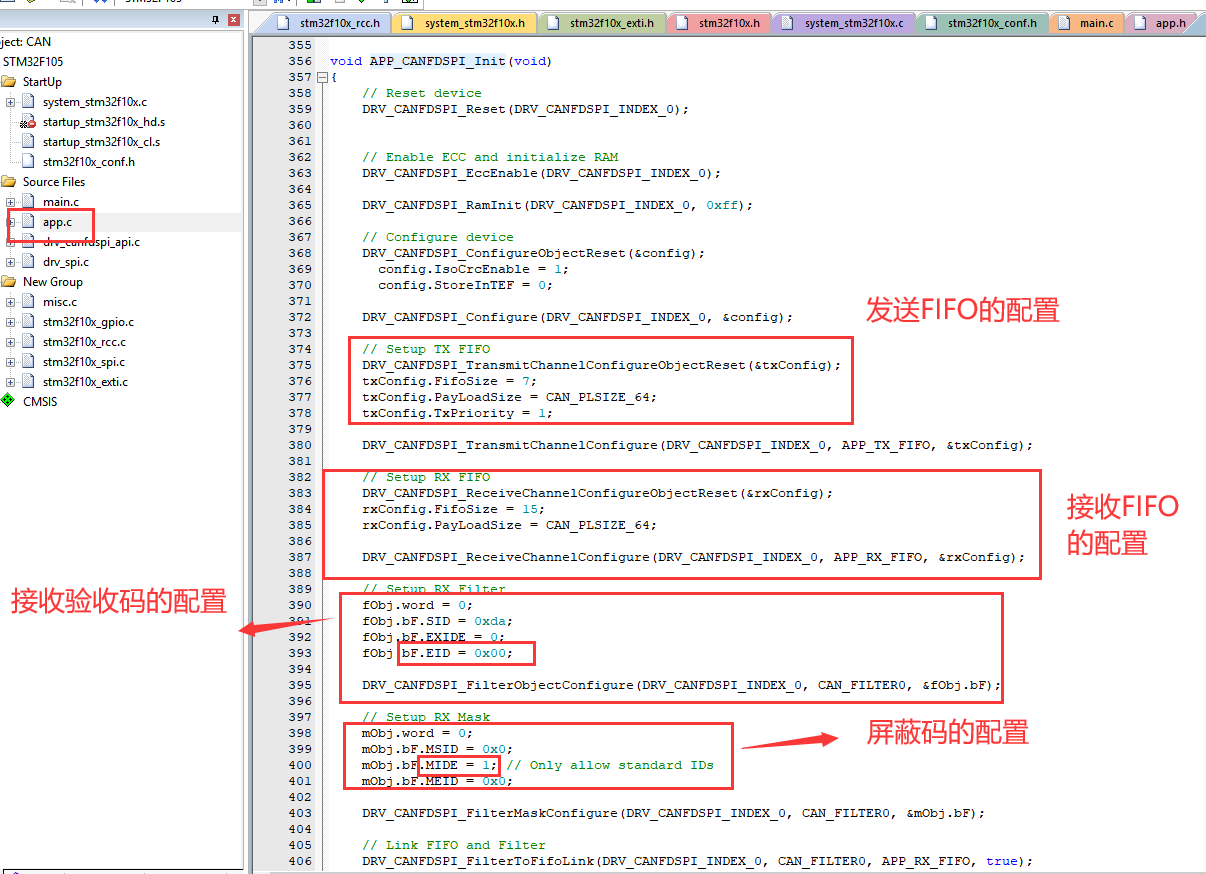




测试SPI操作mcp251x是否成功时，在APP.h中打开TEST\_SPI宏定义，程序测试SPI读写MCP2517ram和寄存器是否成功。RAM测试和寄存器读写测试，要在251x复位状态下进行



4.MCP251x的初始化配置

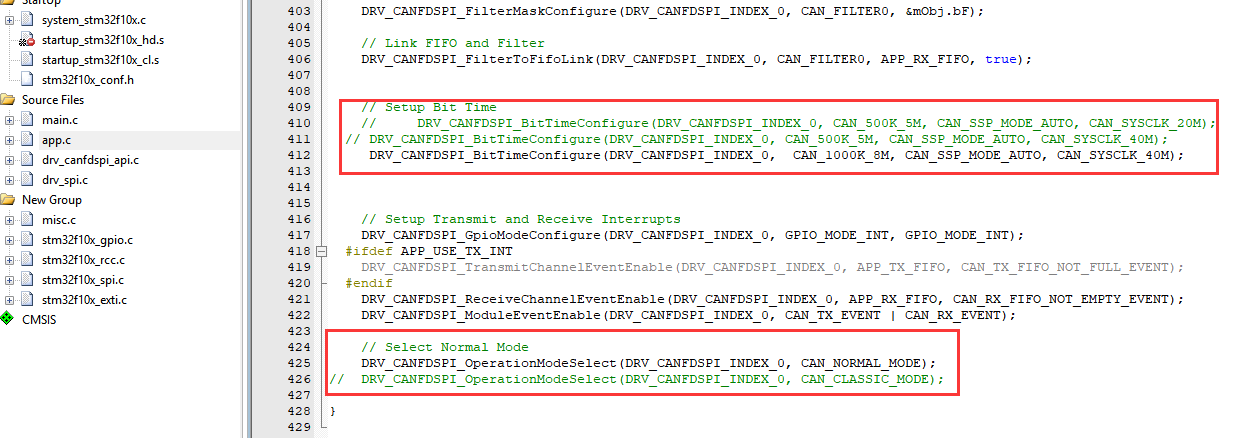


接收标准帧和扩展帧的滤波设置

主要是EXIDE和MIDE的配置，当MIDE=1时，接收扩展帧还是标准帧，是根据EXIDE的设置来决定的，EXIDE=1时，接收扩展帧，EXIDE=0时，接收标准帧。

当MIDE=0时，扩展和标准都有效



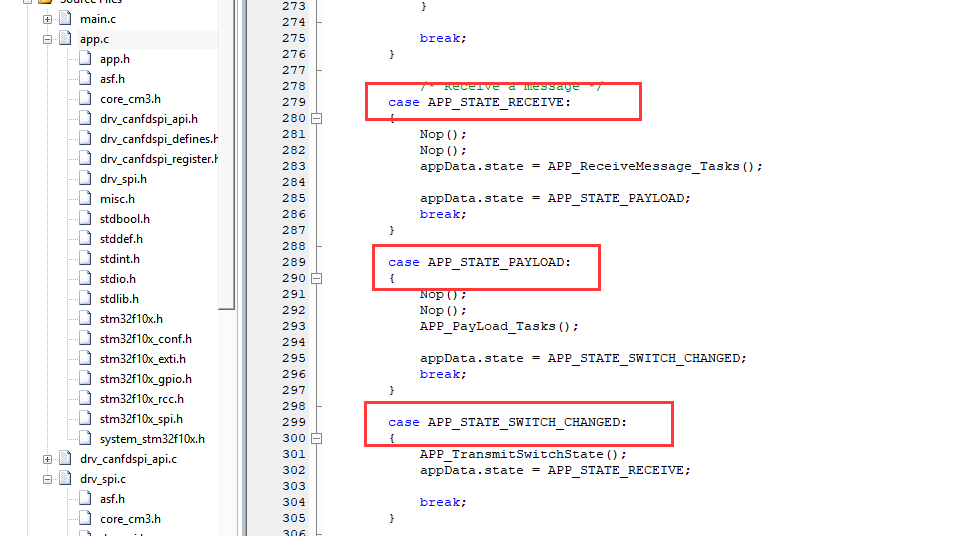


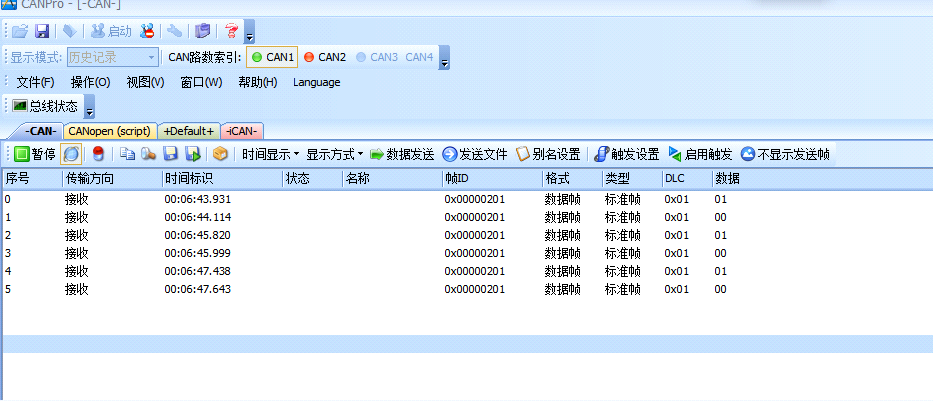
第一个红框中mcp251x配置选择波特率（开发板默认程序是1M\_8M ），SSP自动模式，焊接的晶振（开发板mcp251x焊接的是40M），

第二个红框中配置工作模式，

使用CAN FD时选择CAN\_NORMAL\_MODE,普通CAN 2.0时选择CAN\_CLASSIC\_MODE

5.关掉SPI\_TEST宏定义，正确初始化后，在图中接收、发送、按键检测状态机中循环切换，按下按键S1后，CAN会发按键状态数据，发送一次，D1闪烁一次，发送的数据如usb canfd截图所示。将收到的数据发回给开发板，会看到D1的状态变化。





6.接收函数说明

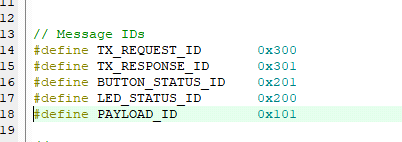
接收到特定ID的数据处理

特定ID

收到0x300是主机请求数据，将会用id 0x301将数据发回给主机

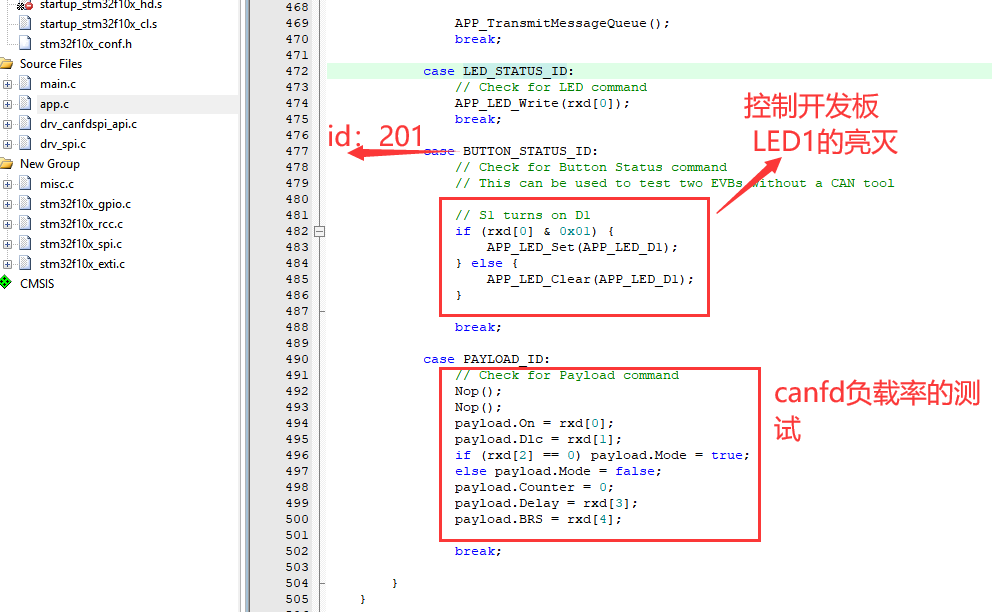
0x200和0x201,操作LED1的亮灭。

0x101 是负载测试





收到101 id后，会按照配置向主机发送数据，配置字节数据第一个字节是使能，数据第二个字节是请求发送的长度，第三个字节是模式（0是发随机数，1是发计数次数），第四个是上传的间隔时间 ，第五个是是否可变速率配置



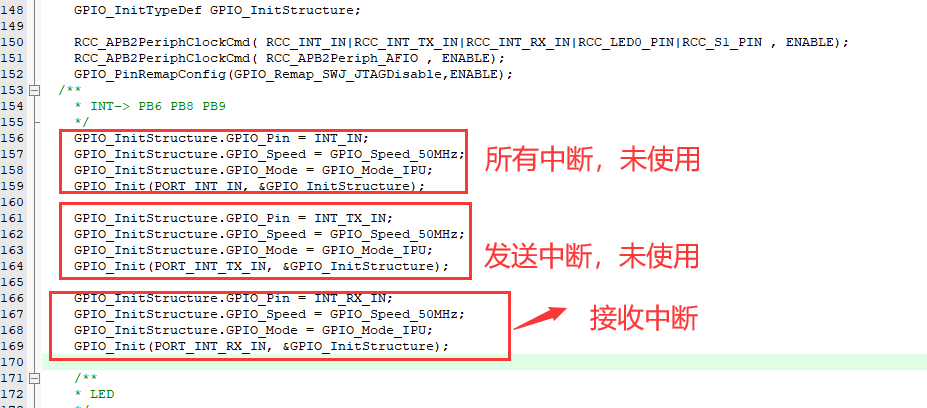
7.S1按键按下后 ，发送数据说明





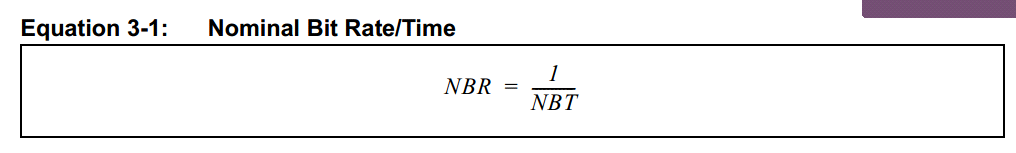
关于中断的说明：

目前例子使用的中断引脚情况，INT是所有中断，包括接收，发送，错误等引起的中断，INT0是发送中断，INT1是接收中断。

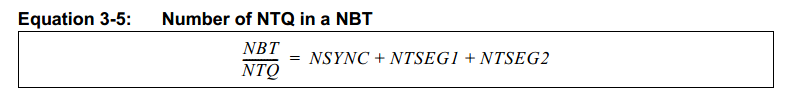


波特率配置说明(仲裁波特率配置，数据域和这个类似)

波特率NBR



NBT/NTQ的公式，NSYNC固定=1，SEG1和SEG2可配。1/NTQ=频率/分频值

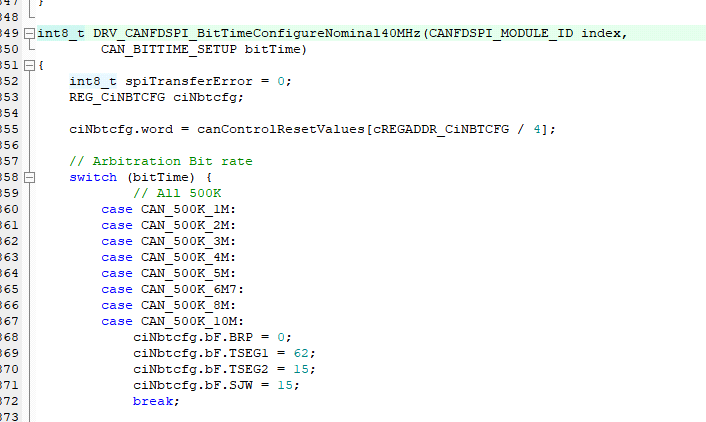


这是500K的配置例子，BRP=0，就是1/NTQ = 40M/1=40M 没有分频；

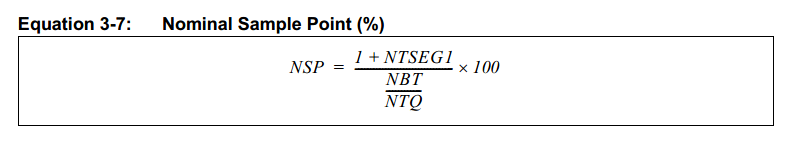
TSEG1=62，TSEG2=15，此处NTSEG=TSEG+1，所以NTSET1=63，NTSEG2=16，

NSYN固定=1，

所以NBT =(1+63+16=80)\*NTQ, NBR=1/80\*NTQ=40M/80=500K

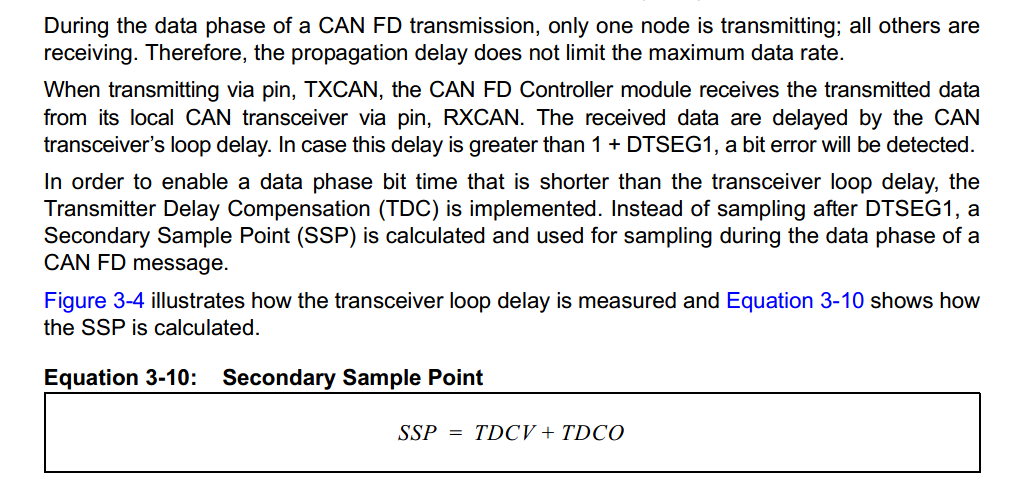


采样点计算方法



关于SSP的介绍

收发器延迟补偿（网上拷贝的）  
　　目前应用的CAN收发器存在回路延迟，延迟时间最高可达255ns。CAN FD协议中，帧的数据段位时间有可能小于255ns，如果当采样点到来时发送节点还没有收到其发出的位，发送节点就会报错，因此 CAN FD通过可选择性的收发器延迟补偿机制避免这种情况发生。发送节点将这种补偿机制应用到帧的高速数据段，接收节点不需要收发器延迟补偿。  
　　在收发器延迟补偿机制中，定义一个次级采样点SSP(Second Sample Point)并在接收位的SSP处比发送的位值进行比较检查位错误，同时忽略原来采样点的位值。CAN FD协议控制器在位速率转换之前的EDL位到r0位的下降沿上测量收发器的回路延迟Trv\_Delay，延迟Trv\_Delay通过计数器测量，起始于CAN\_tx的r0下降沿终止于CAN\_rx的r0下降沿。  
　　Trv\_Delay加上一个可变的偏移量(例如1/2高速位时间)即为SSP的位置。如果SSP发生在发送位发送结束后，那么将发送位缓存等待SSP到达再进行位错误检查。如果在SSP处检测到位错误，那么将这个位错误的信息缓存，直到下一个采样点到达时才对位错误做处理，此时位速率转换回仲裁段的低速率进行数据传输。如果直到CRC界定符的采样点到达都没有检测到位错误，CAN FD协议控制器转换位速率并返回到标准位错误检测模式。



**大意就是在数据段增加了ssp，使采样点后移了，避免出现错误，ssp包括测量的传输延迟和补偿时间，tdco是补偿时间相当于原来的dtseg1**