**请打开“导航窗格”以更清晰地查看文件结构**

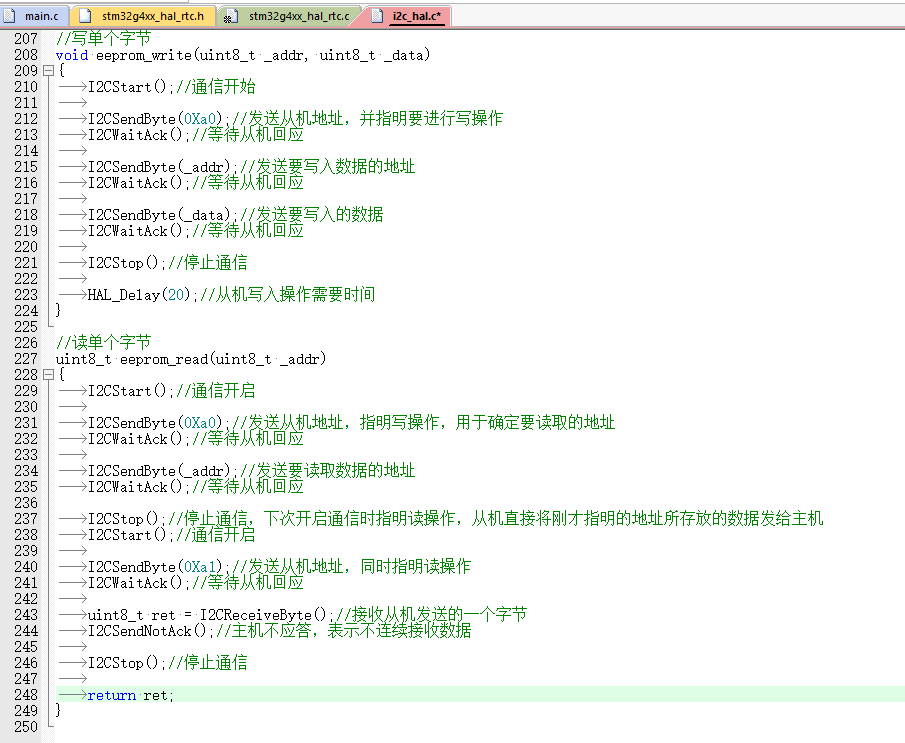
**07EEPROM**

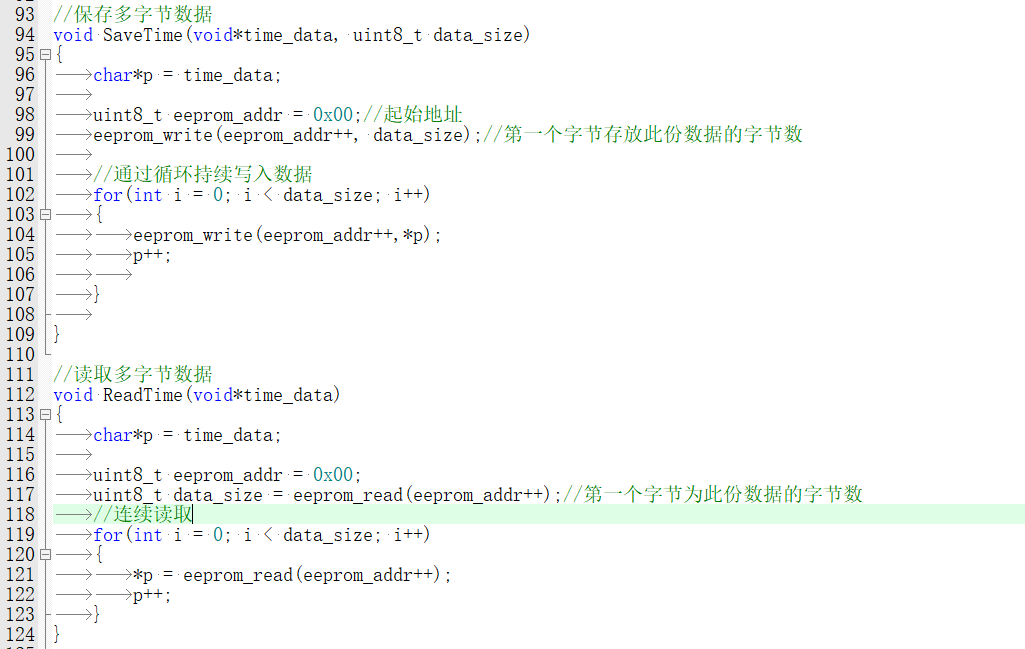
**1.配置**

无需cubemx配置

**2.函数**

**第一种连续写入方式**

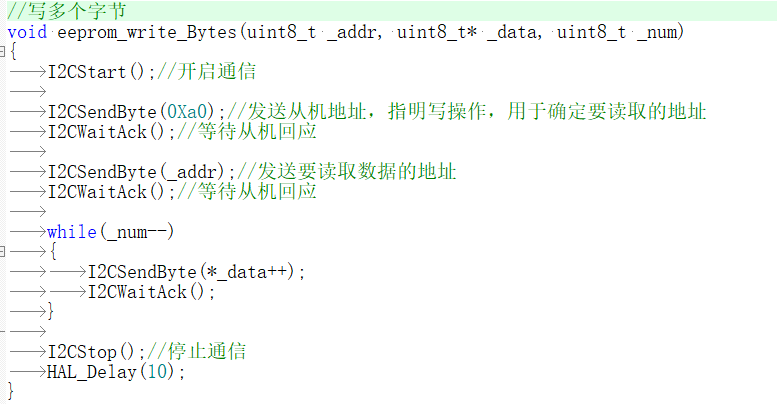




**第二种连续读写方式（兼容单个字节和多个字节）**

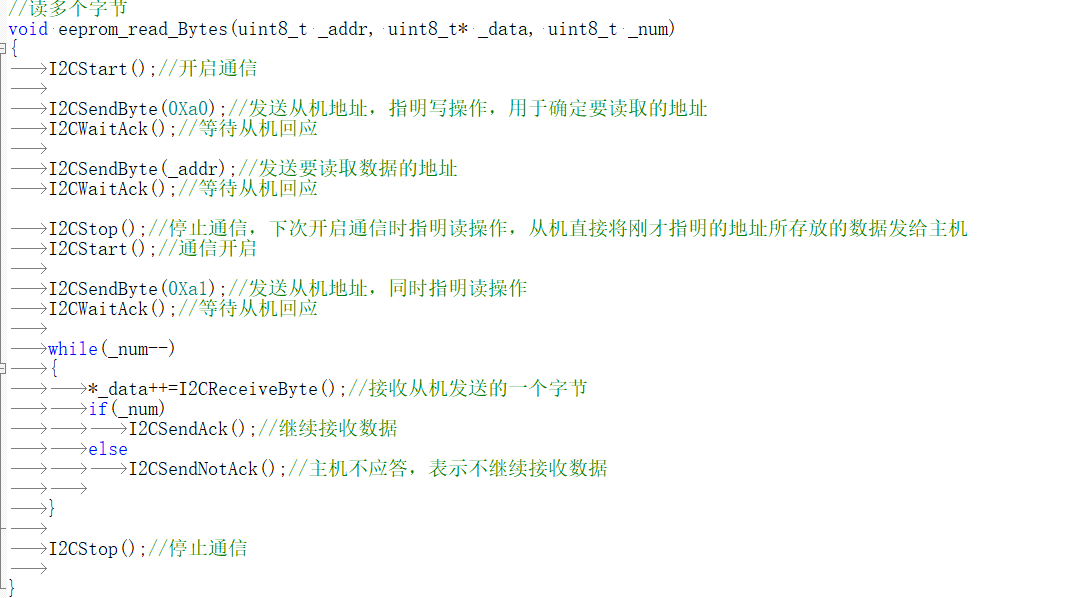
1. 写入

* 开启通信
* 发送从机地址，最后一位为写入位（置0）
* 等待从机回应
* 发送数据（8位）
* 等待从机回应
* 重复以上两个步骤，直到写入的块数达到需求
* 停止通信
* 进行10ms延时（根据实际情况而定，不能阻塞到其他功能，比如按键读取）



1. 读出

* 开启通信
* 发送从机地址，最后一位为写入位（置0）（此步骤为确定EEPROM的读取地址，如果不需要指定也可以直接跳转到读取）
* 等待从机回应
* 发送要读取数据的地址
* 等待从机回应
* 停止通信
* 开启通信
* 发送从机地址，最后一位为读取位（置1）
* 等待从机回应
* 接收数据（注意要用变量接收）
* 如果还要接收数据，则发送应答，否则发送非应答
* 接收完毕，停止通信



**3.相关文件**

i2c\_hal.h

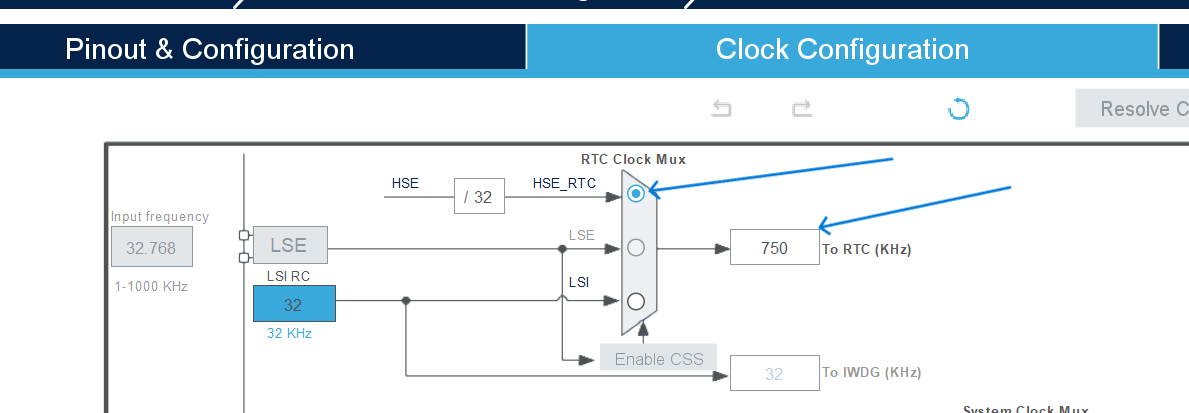
**4.注意事项**

* 使用前，需要将I2C初始化
* 写延时需要根据实际情况而定，不能干扰到其他模块的运行（比如按键）
* 读取和写入注意指针的移动，data++可独立写为一行
* 兼容版的参数num为字节数

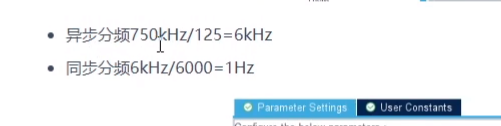
**0801RTC**

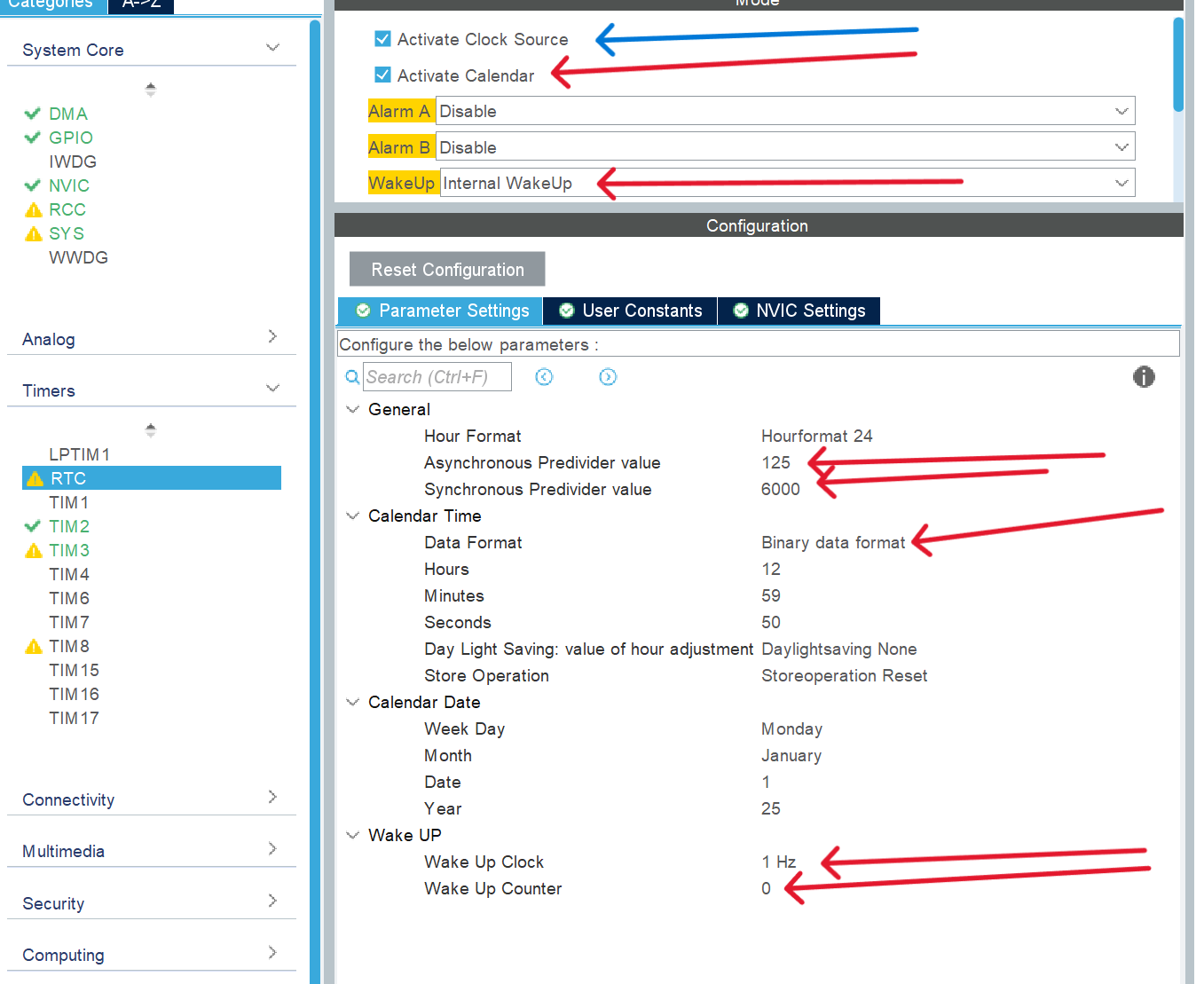
**1.配置**

* RTC的时钟改为HSE，24M/32=750KHz

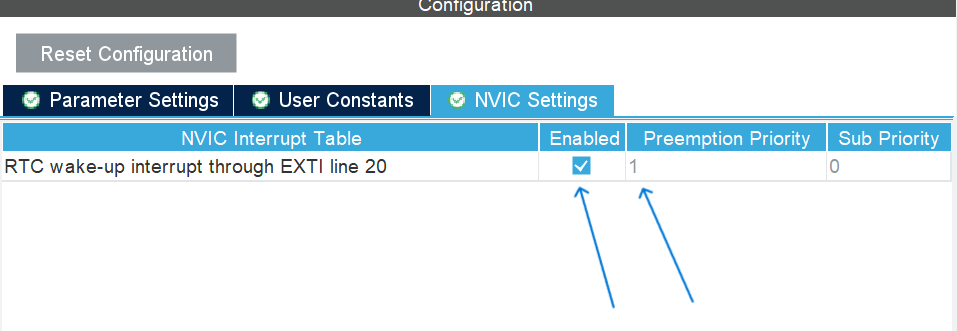


* 通过同步分频和异步分频将频率设置为1Hz



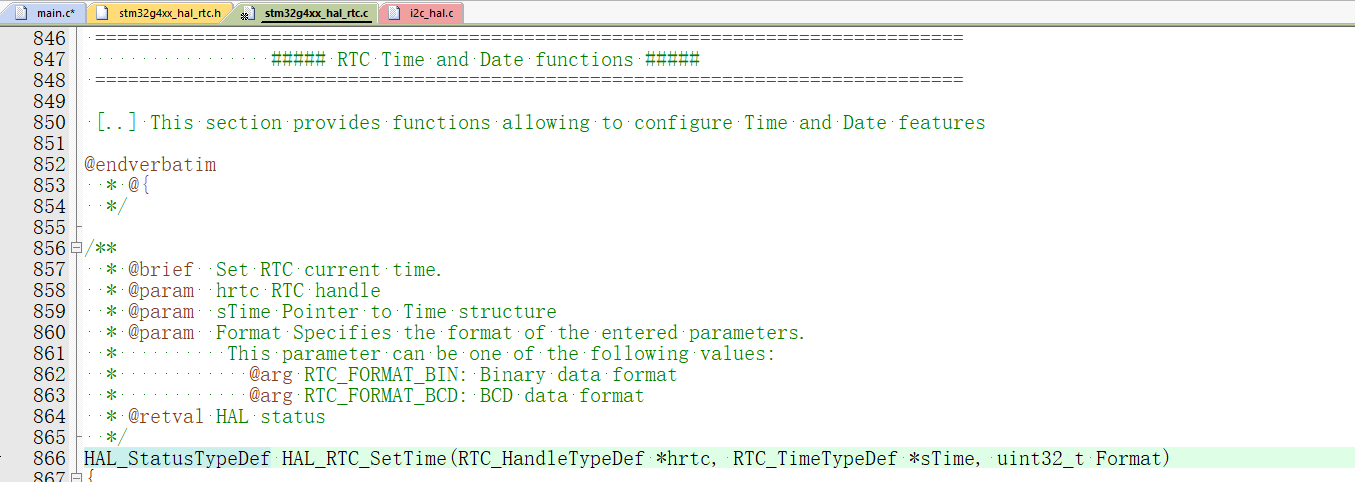


* 开启中断

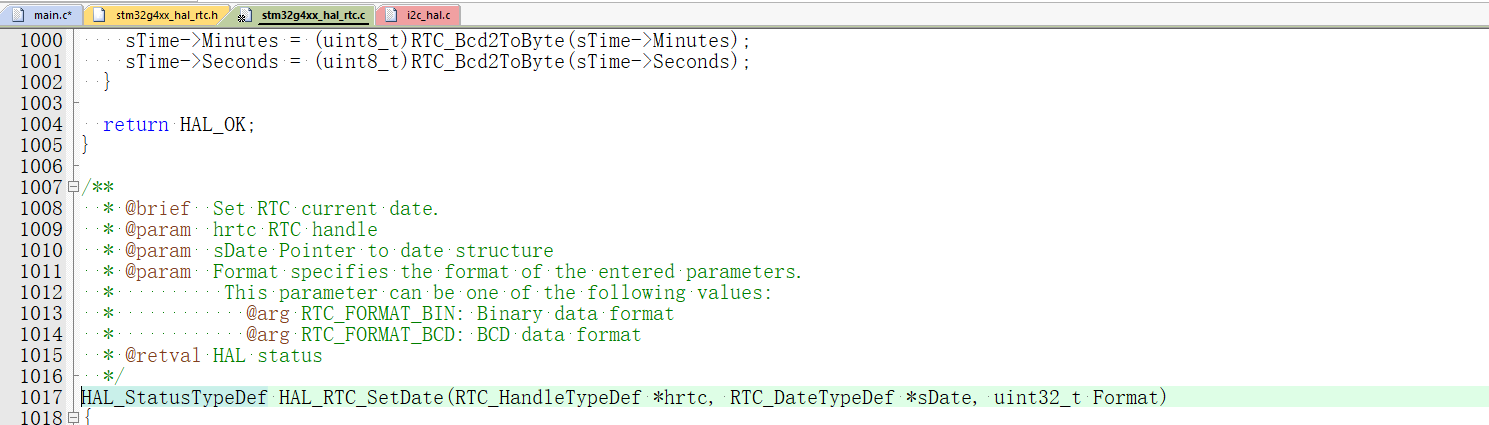


**2.函数**

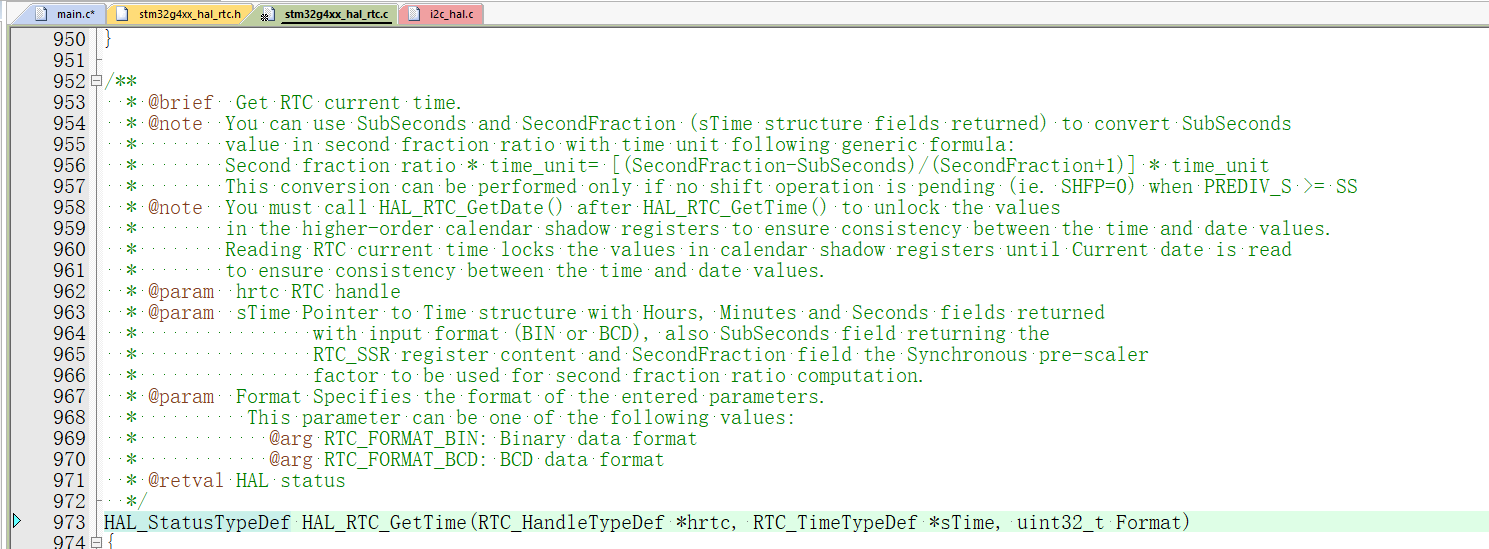
**HAL\_RTC\_SetTime(&hrtc,&rtc\_time,RTC\_FORMAT\_BIN);**



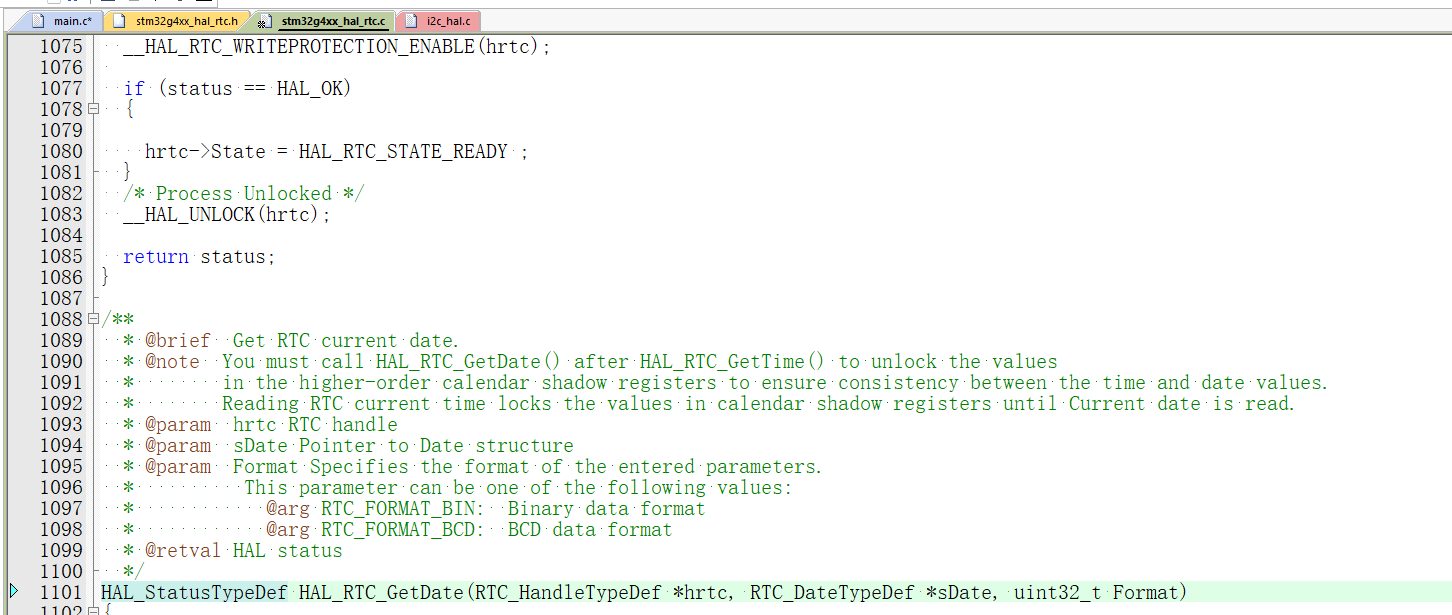
**HAL\_RTC\_SetDate(&hrtc,&rtc\_date,RTC\_FORMAT\_BIN);**



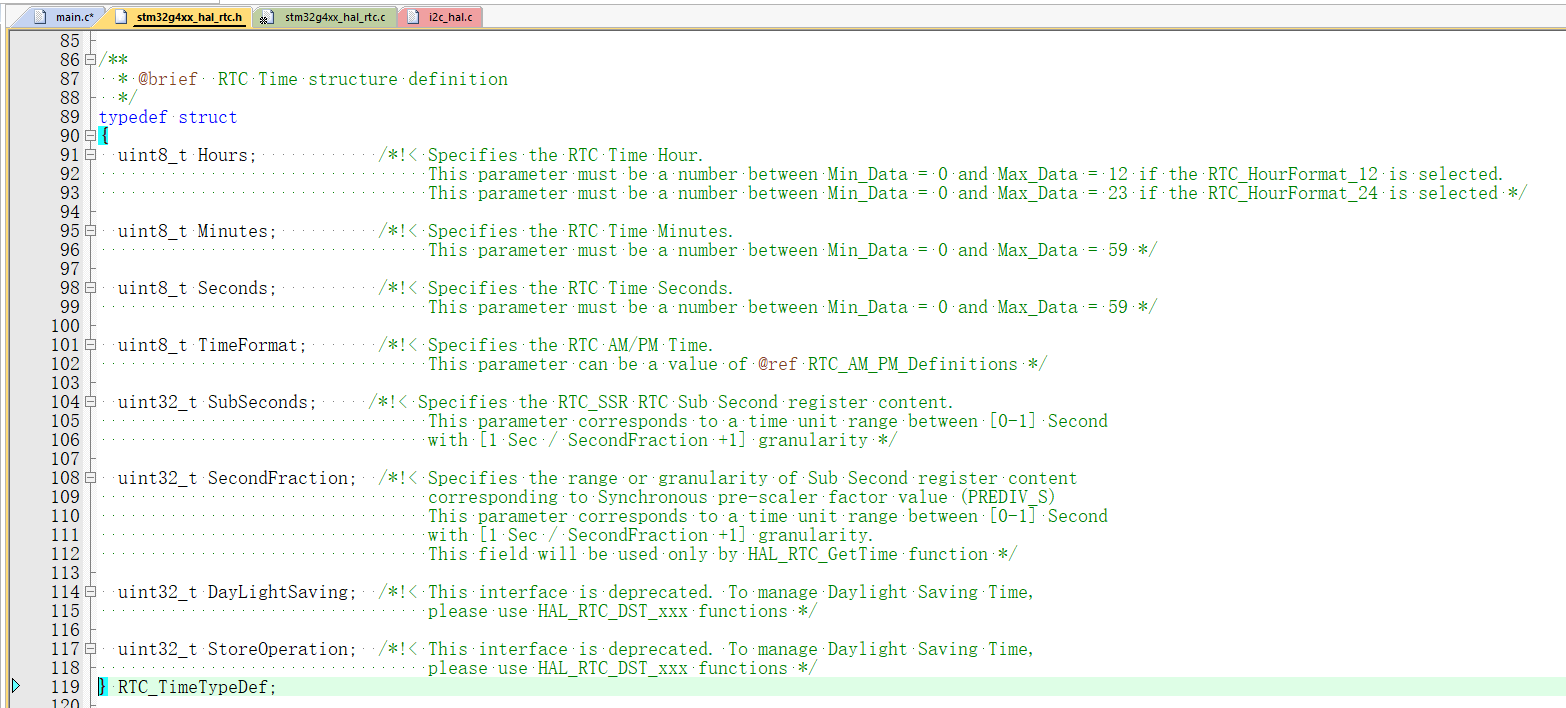
**HAL\_RTC\_GetTime(&hrtc,&rtc\_time,RTC\_FORMAT\_BIN);**



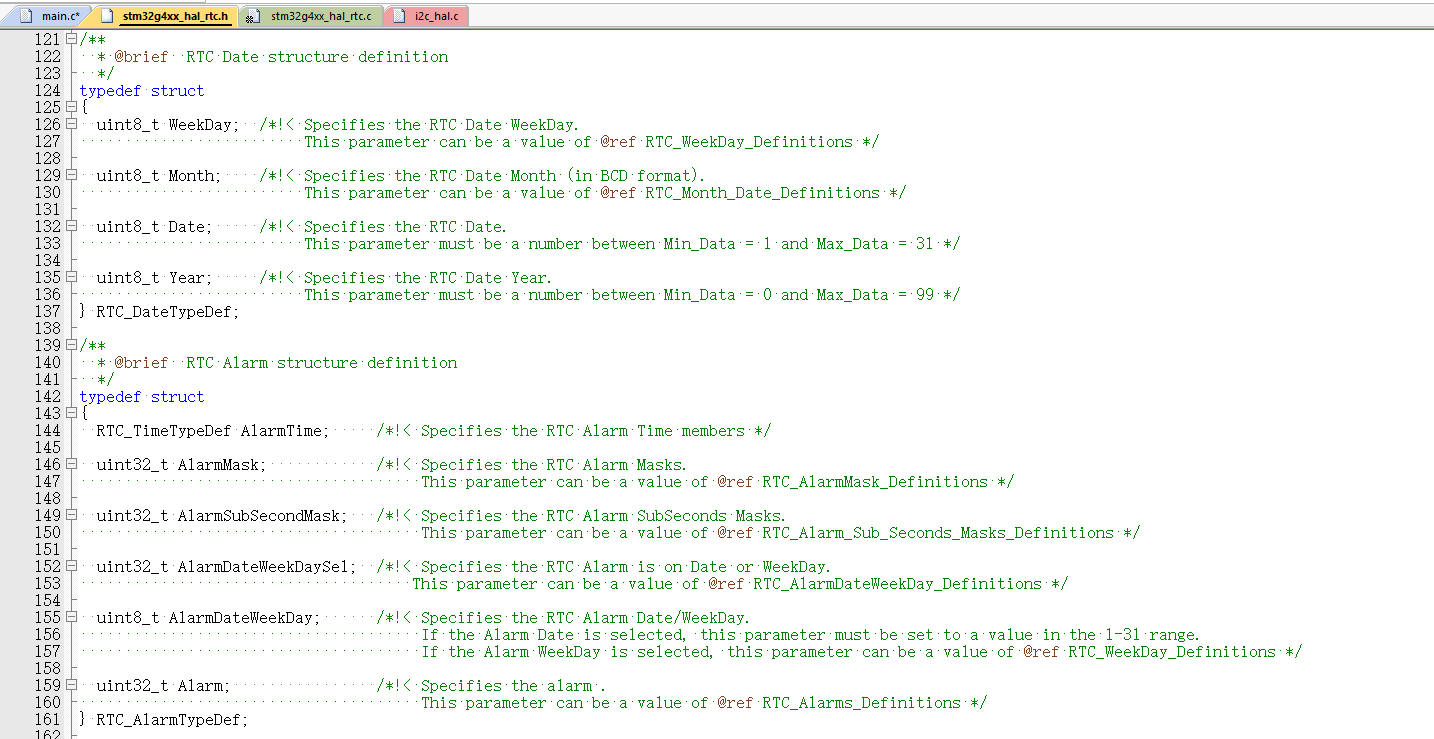
**HAL\_RTC\_GetDate(&hrtc,&rtc\_date,RTC\_FORMAT\_BIN);**



**RTC\_TimeTypeDef**

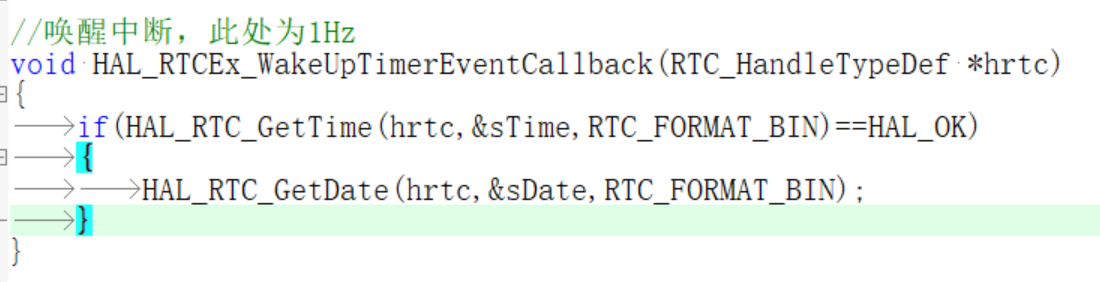


**RTC\_DateTypeDef**



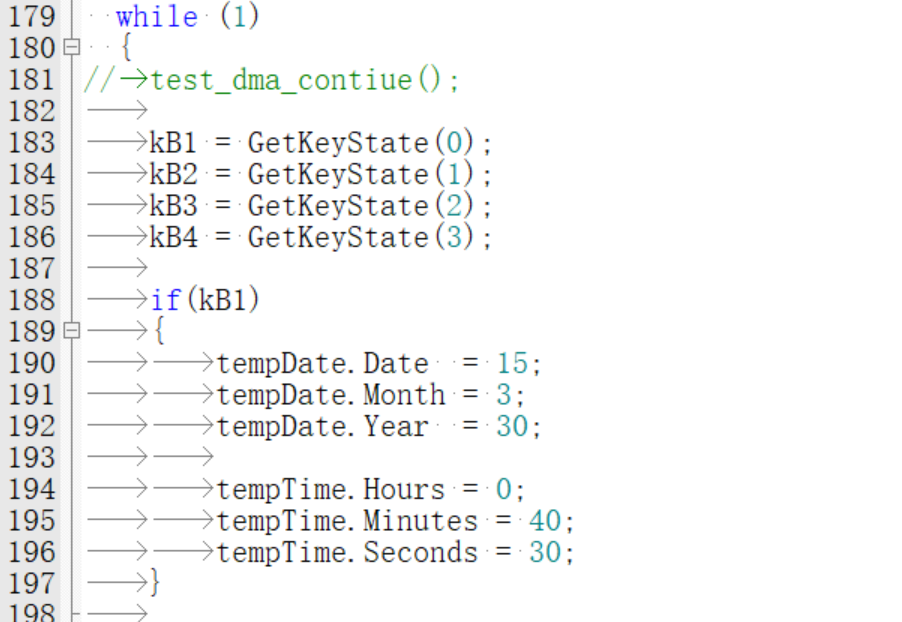
**唤醒回调函数**

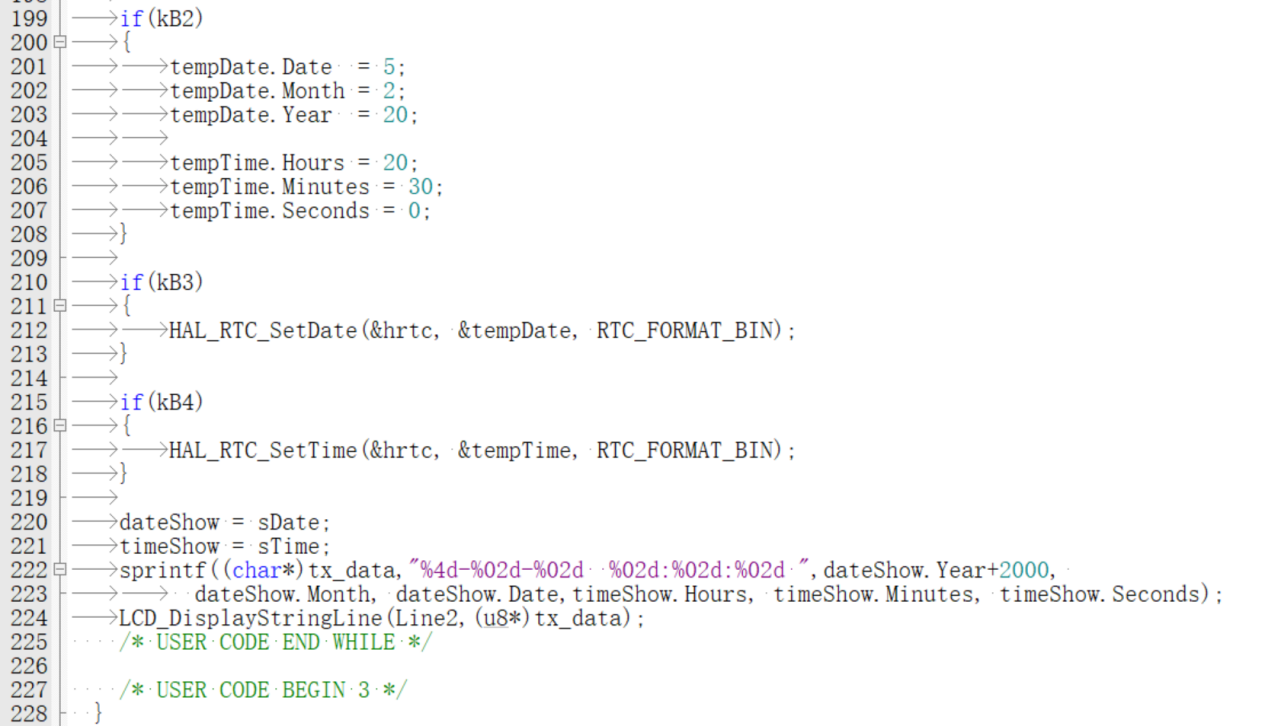
* 可在stm32g4xx\_it.c中找到RTC唤醒中断函数，进而找到回调函数



**设置与显示时间**

* 用其他日期和时间变量进行设置，不能使用 用于接收的两个变量，显示时也使用其他两个变量进行显示
* 在唤醒中断中置标志位，显示函数中检测到该标志位后再进行显示





**3.相关文件**

stm32g4xx\_hal\_rtc.h

stm32g4xx\_hal\_rtc.c

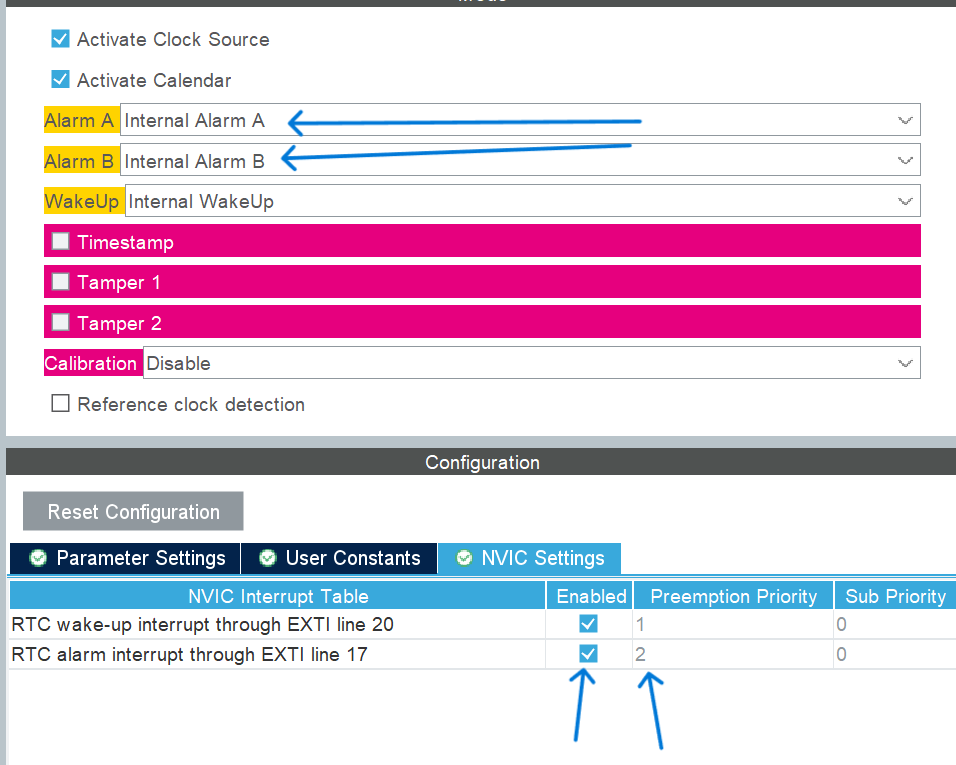
**4.注意事项**

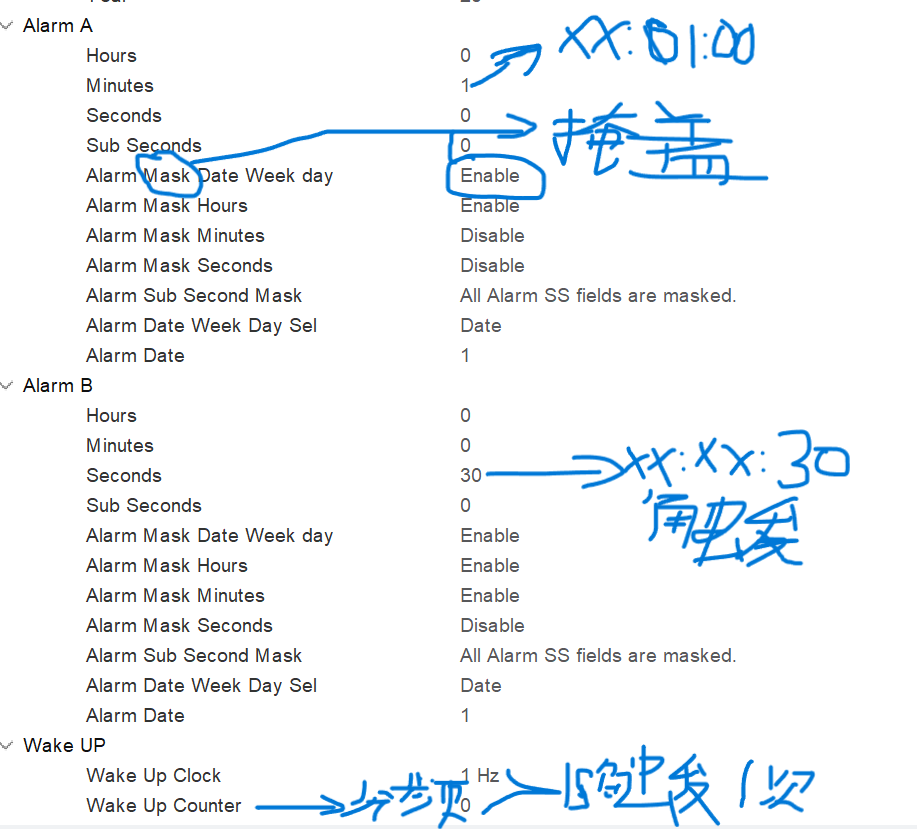
* 记录RTC时间数据（即日期、小时等）的操作需要放在LCD显示之前，否则数据会出错
* 在唤醒回调函数中设置标志位，外部检测到标志位后再进行显示，然后将标志位清空，等待下一次中断置位
* Wake Up Counter设置为**0**

**0802RTC+Alarm**

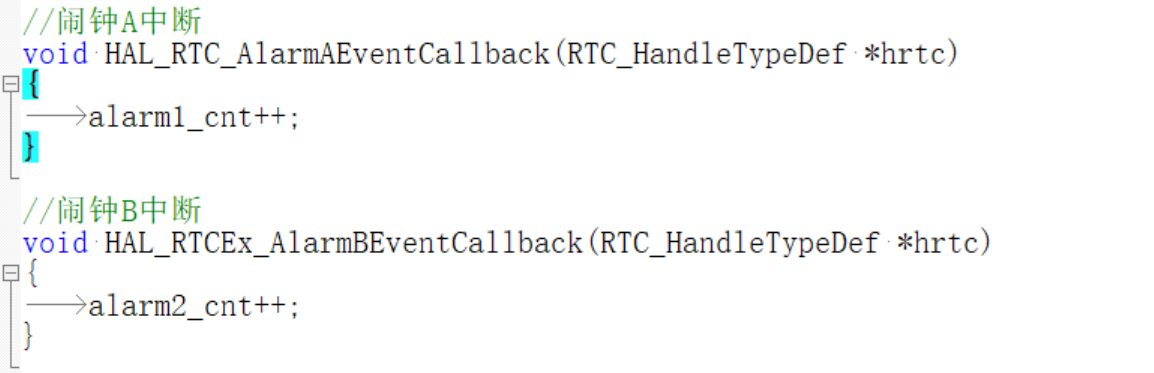
**1.配置**

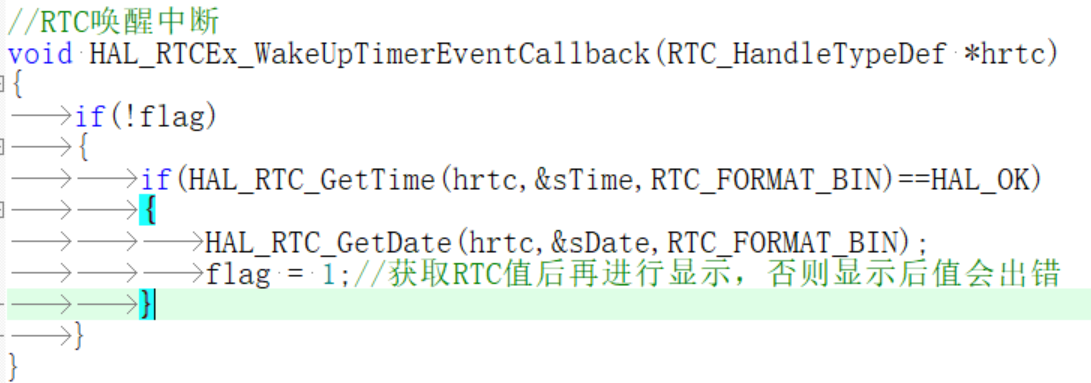
在RTC的基础配置上添加如下配置：





**2.函数**



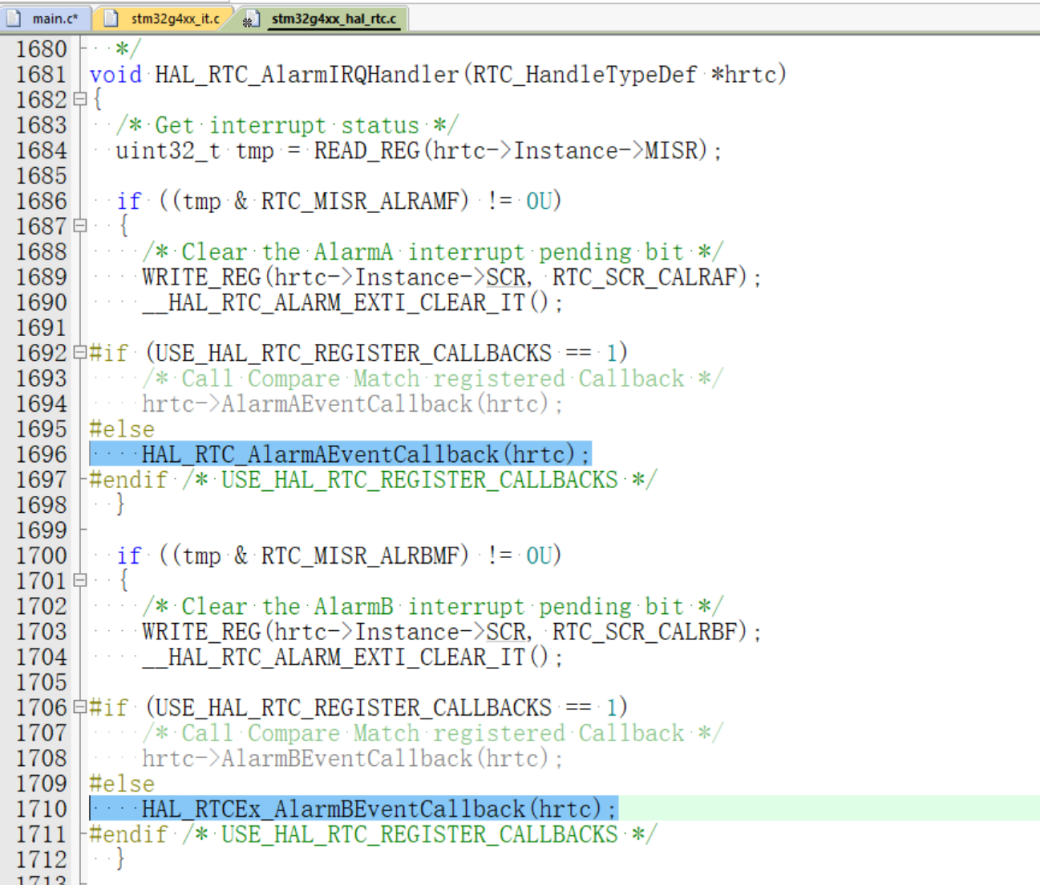




**3.相关文件**

**4.注意事项**

* 不同闹钟的回调函数名称的差别可能不止一个字母，需要根据闹钟中断所调用的回调函数为准



* 以xx:aa:00为闹钟时，需要将秒的Mask也设置为Disable（即不能掩盖），否则当时钟走到xx:aa:xx时会重复触发（每秒触发一次，直到时间走过aa）
* **如果需要在中断函数中调用HAL\_Delay，则需要将Time Base:System tick timer的优先级设为0，否则可能会卡死主程序**
* 需要cubemx中开启对应的中断