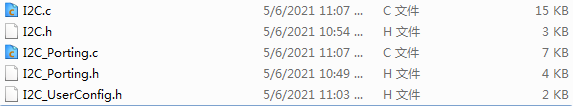
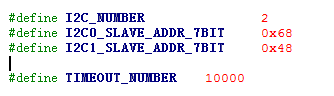
通用GPIO模拟I2C程序

MCU在使用过程中，有时不得不使用GPIO来模拟I2C的操作，比如MCU内部的I2C个数不够，或者原理图设计时将SCL和SDA接反，又或者MCU内部的I2C模块有BUG（工程师在MCU选型前切记看一遍Errata）

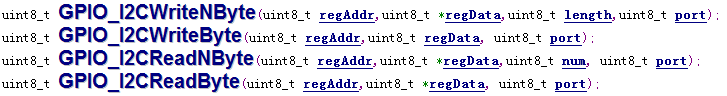
I2C的时序本身并不复杂，这里就不展开介绍。今天主要安利一个小程序，它有5个文件组成：



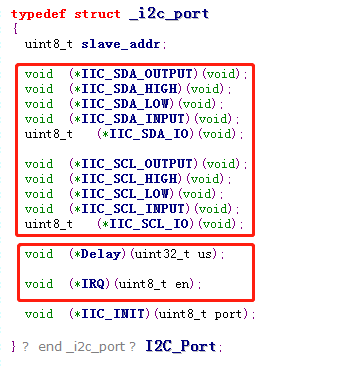
其中I2C\_UserConfig.h定义了用户需要模拟几个I2C主设备，每个主设备所要访问的7bit从机地址以及等待超时时间：



而I2C.c/h文件用来实现I2C的时序操作，用户无需做任何修改，其提供4用户API：



I2C\_Porting.c/h需要用户根据不同的MCU平台以及不同的Pin来配置，每模拟一个I2C外设，需要移植的函数有12个



根据函数名大致可以看出函数的作用

OUTPUT：设置管脚为输出

INPUT：设置管脚为输入

HIGH：设置管脚输出高

LOW：设置管脚输出低

IO：获取IO的输入状态

Delay：延时函数，单位us

IRQ：中断使能函数，输入0表示禁止，其他表示开启中断

IIC\_INIT函数主要是用于释放总线，并设置默认电平

用户使用该程序的流程：

1. 设置I2C\_UserConfig.h的I2C\_NUMBER，I2C0\_SLAVE\_ADDR\_7BIT(注意这里要写7bit地址，程序会自动左移1bit发送)，TIMEOUT\_NUMBER(等待ACK的超时时间)
2. 完成I2C\_Porting.c中所需的12个函数
3. 在函数GPIO\_I2C\_InitPort(I2C\_Porting.c)中挂载步骤2所完成的函数
4. 主程序中添加#include "I2C\_Porting.h"，#include "I2C.h"
5. 主程序中添加初始化函数GPIO\_I2C\_InitPort(x);x表示需要访问的外设号
6. 主程序中通过I2C.h提供的API来读写外设即可