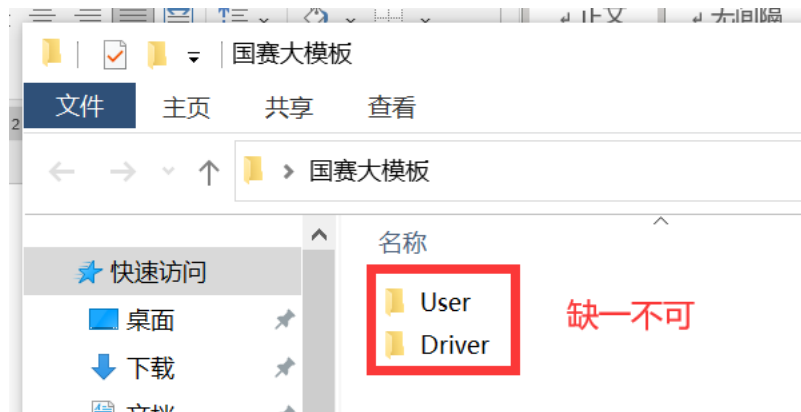


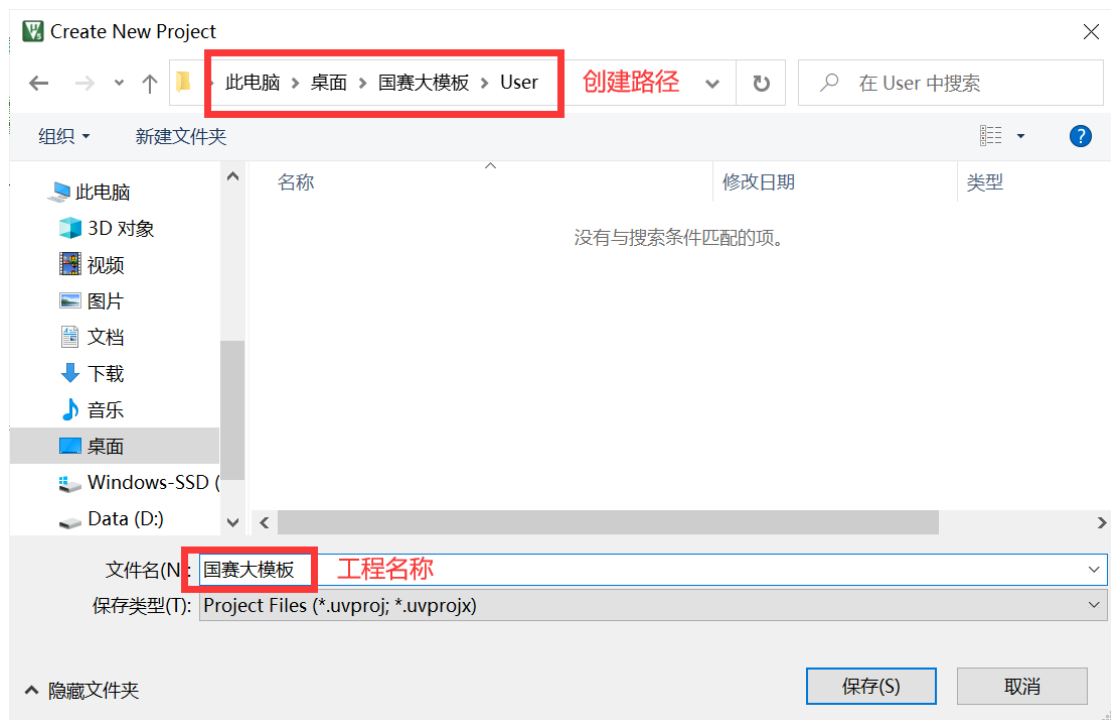
# 协会第四次培训——国赛大模板

前言：本次培训适用于相对有编程基础想冲击蓝桥杯省赛国赛的同学，模块化的封装工程有利于后期嵌入式开发的学习，不作硬性要求掌握，学会三大模板就可以拿下本次校赛。

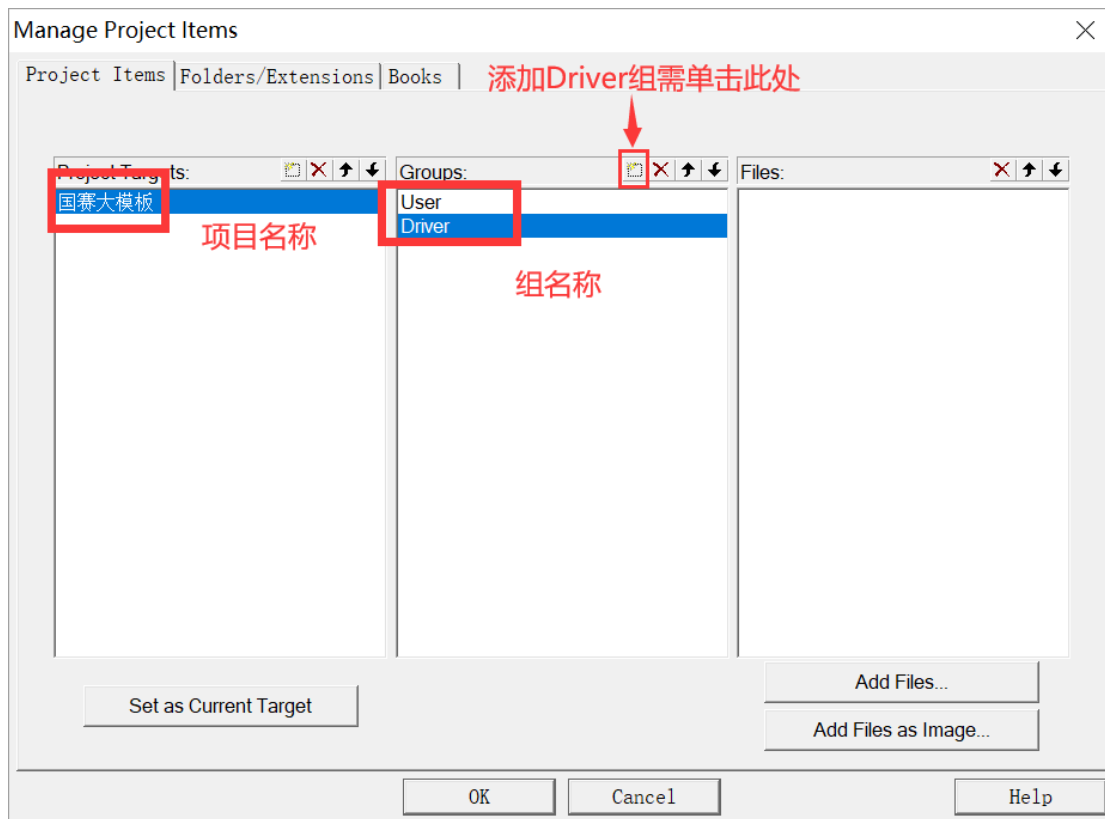
## 1、如图所示创建好工程文件夹（新增 Driver 文件夹，用于存放底层驱动）



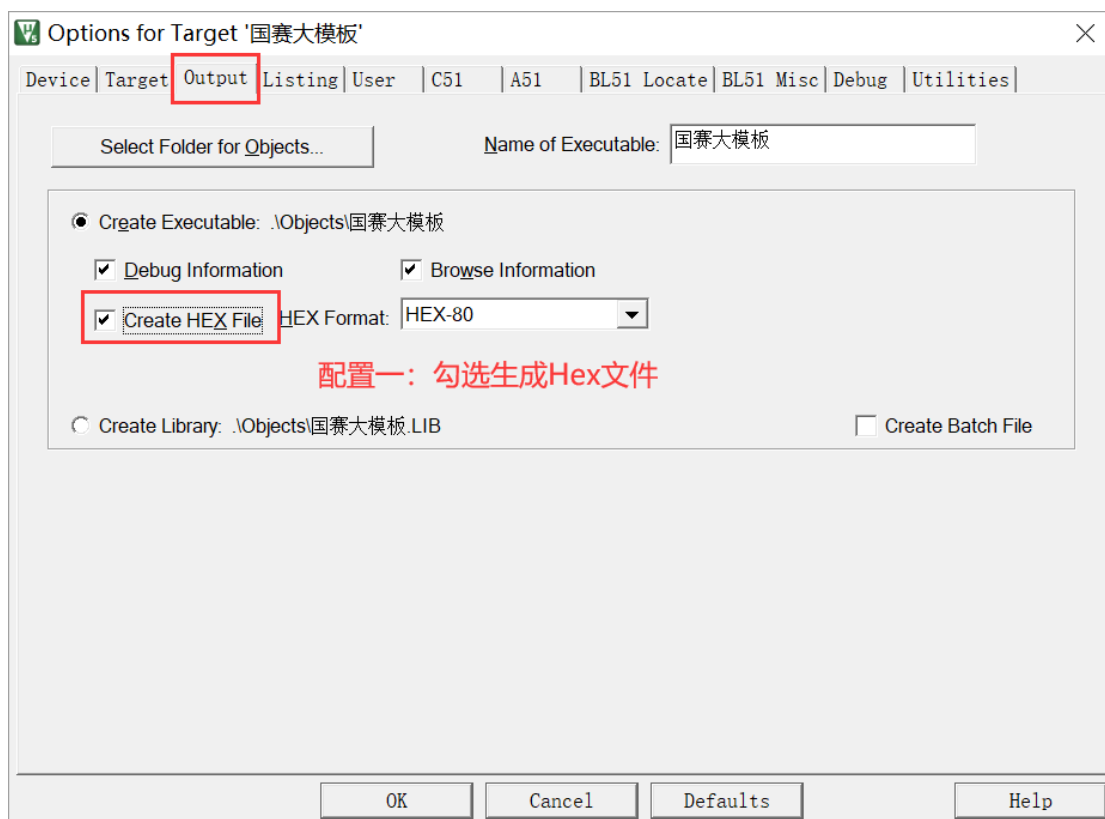
## 2、使用 Keil5 在工程的 User 文件夹内新建工程

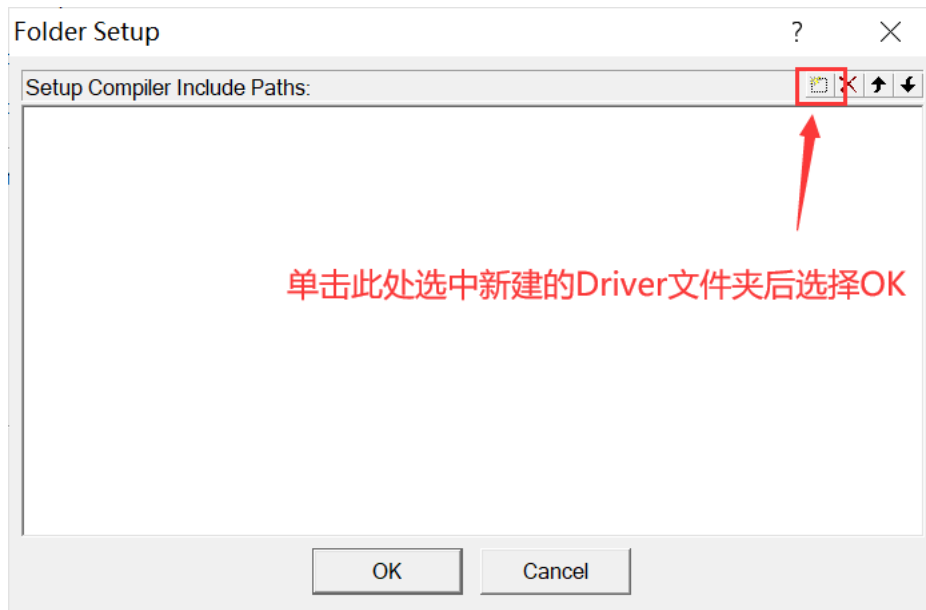
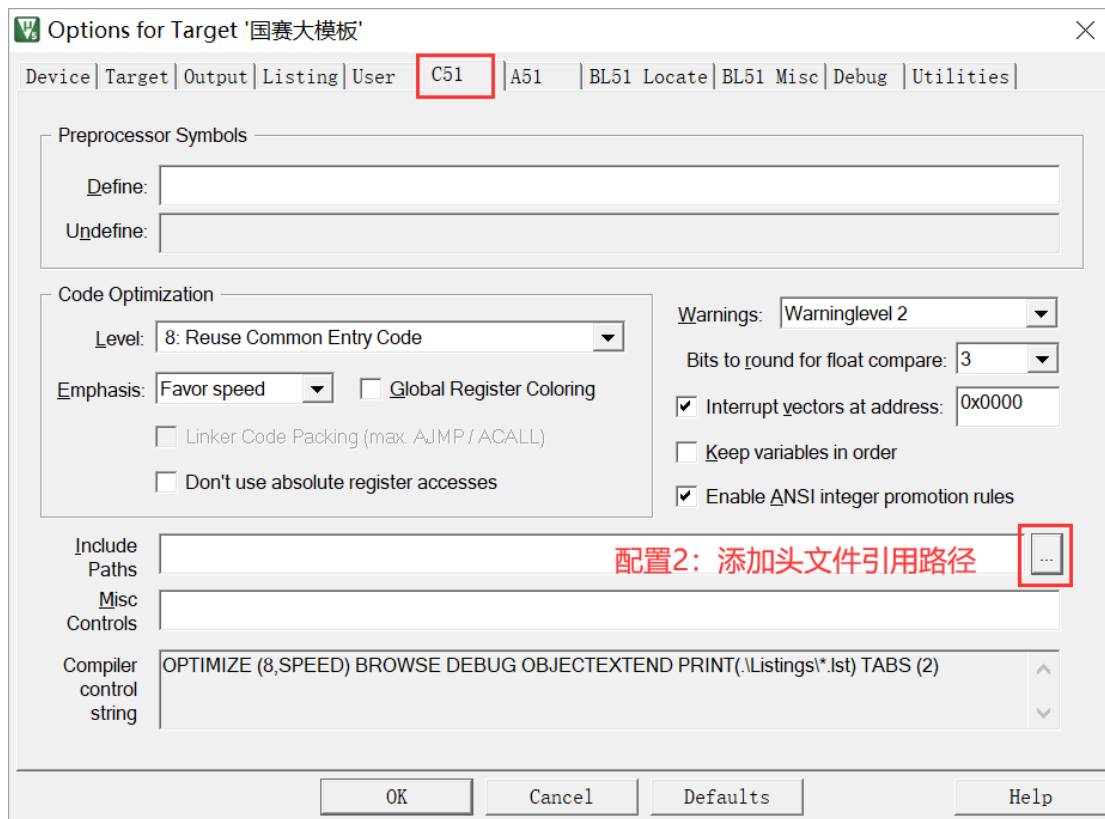


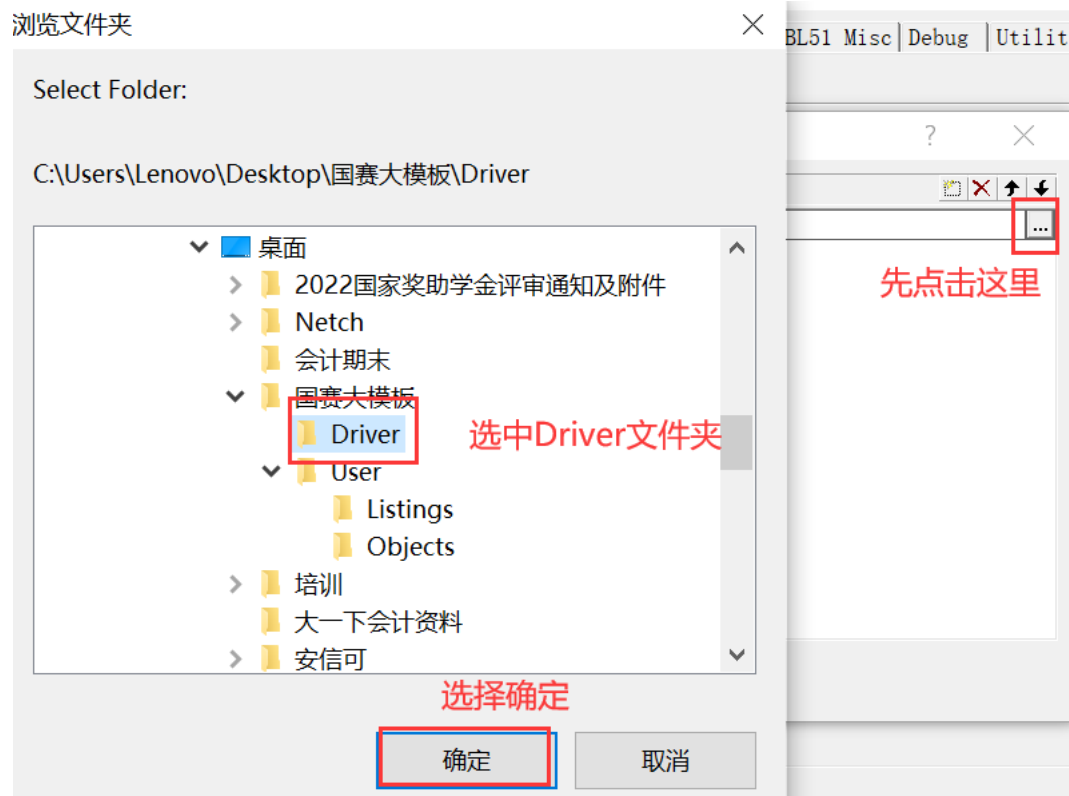
## 3、配置工程“品”字工具栏（第二栏内新增 Driver 组，用于工程存放底层）



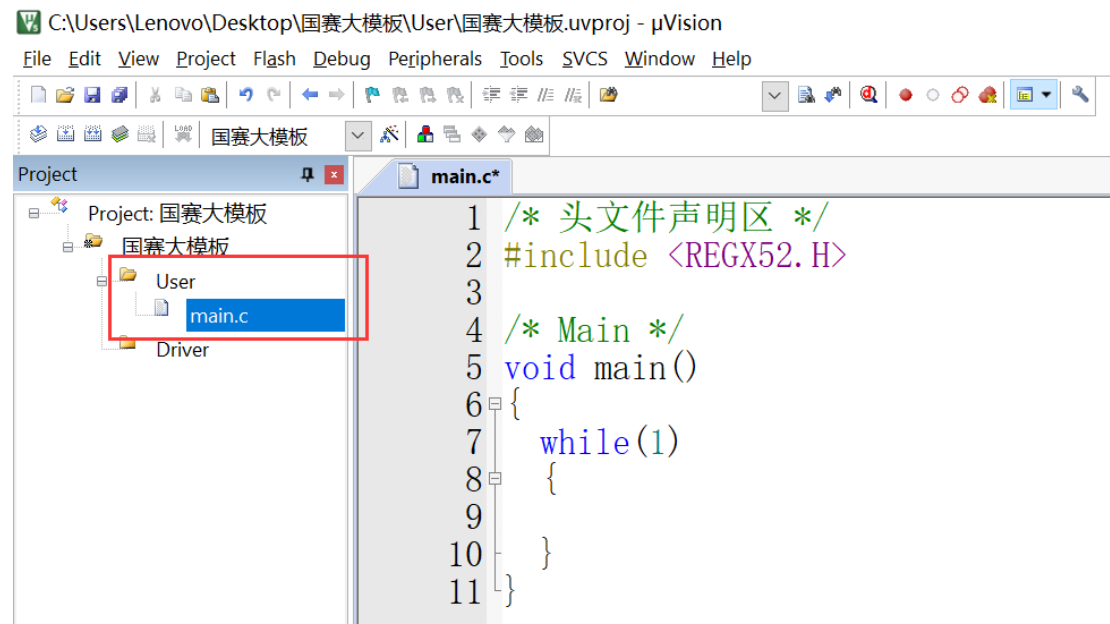
#### 4、配置魔术棒（新增对头文件引用路径添加）



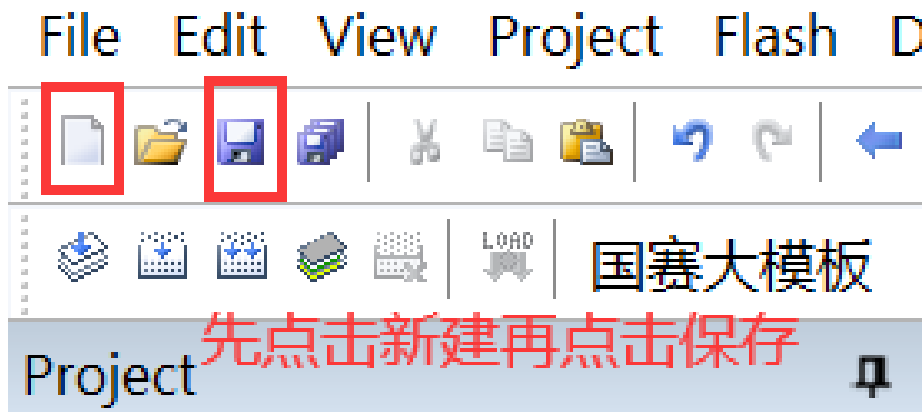




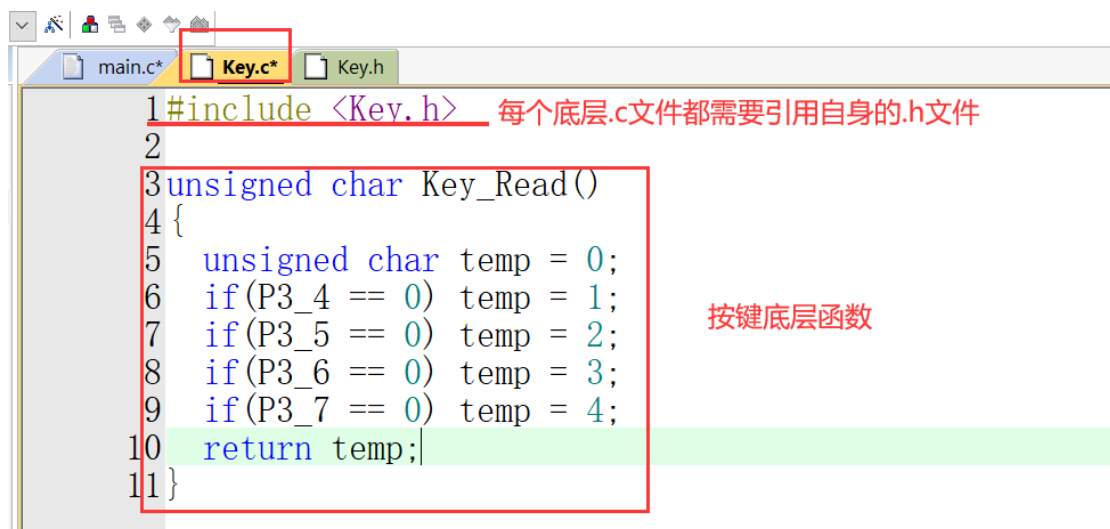
#### 4、在 **User** 组内添加 main.c，并编写程序主体框架



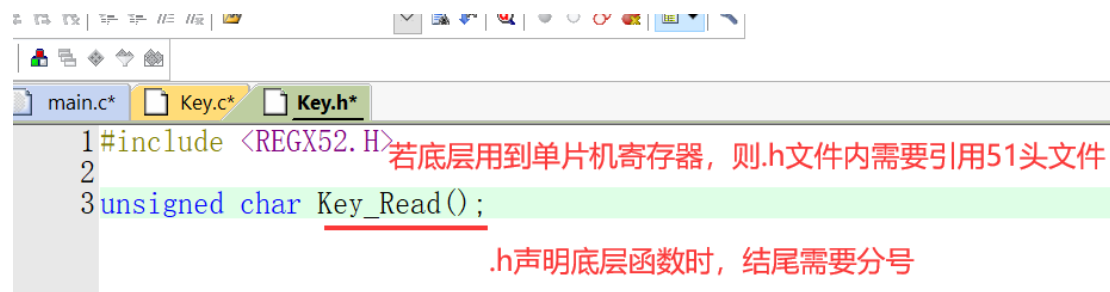
#### 5、新建 Key 底层文件 (Key.c、Key.h)，并保存在 Driver 文件夹中



## 6、在 Key.c 内编写按键底层



## 7、在 Key.h 内声明按键底层



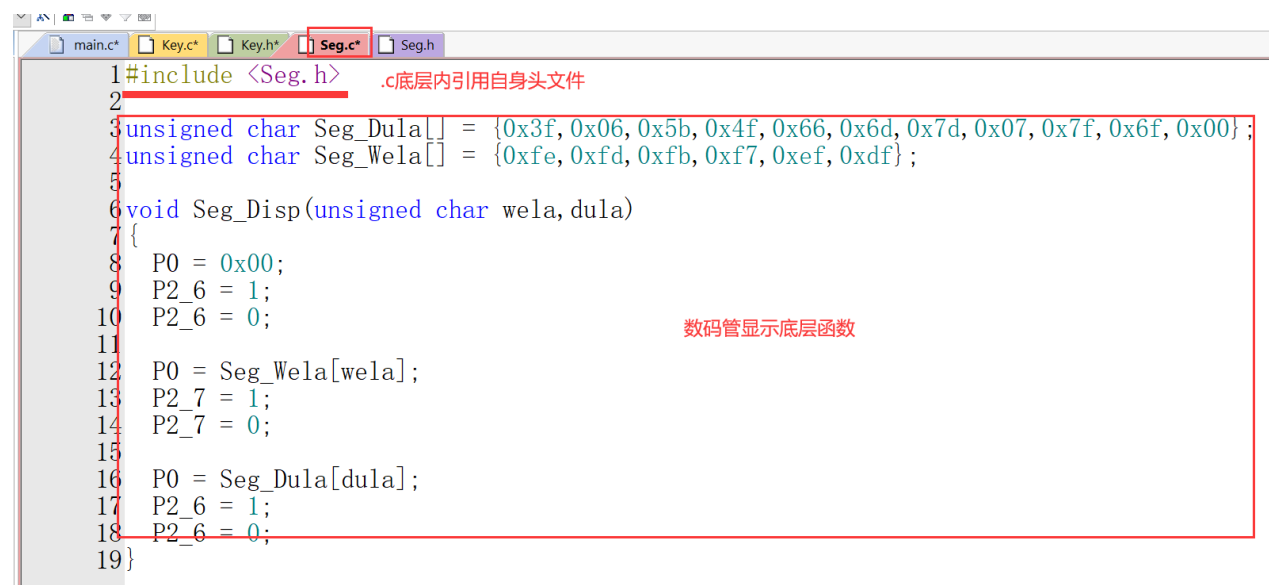
```
1#include <REGX52.H>
2
3unsigned char Key_Read();
```

若底层用到单片机寄存器，则.h文件内需要引用51头文件

.h声明底层函数时，结尾需要分号

## 8、新建 Seg 底层文件 (Seg.c、Seg.h)，并保存在 Driver 文件夹中 (操作步骤参考第五步)

## 9、在 Seg.c 内编写数码管底层

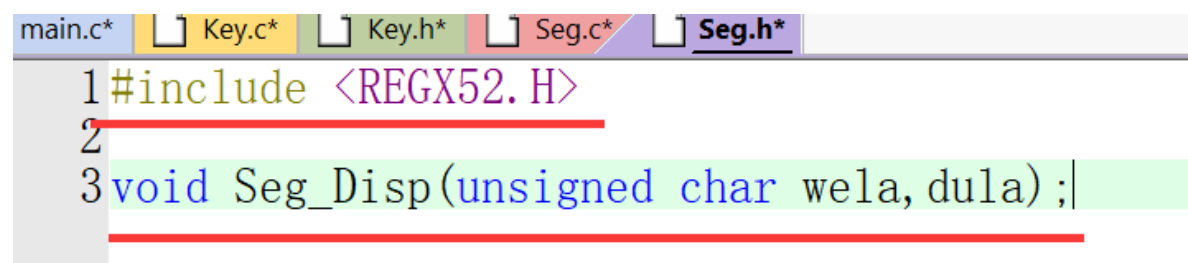


```
1#include <Seg.h>
2
3unsigned char Seg_Dula[] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x00};
4unsigned char Seg_Wela[] = {0xfe, 0xfd, 0xfb, 0xf7, 0xef, 0xdf};
5
6void Seg_Dis(unsigned char wela, dula)
7{
8    P0 = 0x00;
9    P2_6 = 1;
10   P2_6 = 0;
11
12   P0 = Seg_Wela[wela];
13   P2_7 = 1;
14   P2_7 = 0;
15
16   P0 = Seg_Dula[dula];
17   P2_6 = 1;
18   P2_6 = 0;
19}
```

.c底层内引用自身头文件

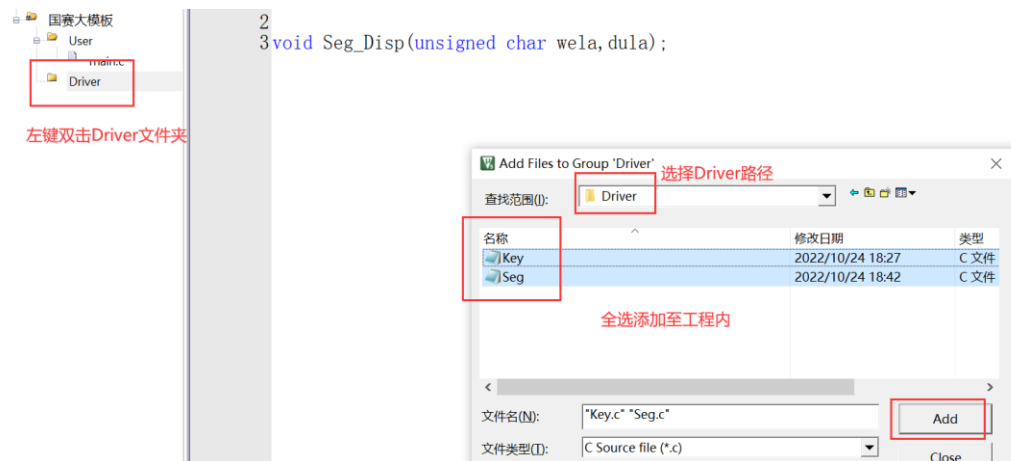
数码管显示底层函数

## 10、在 Seg.h 内声明数码管底层

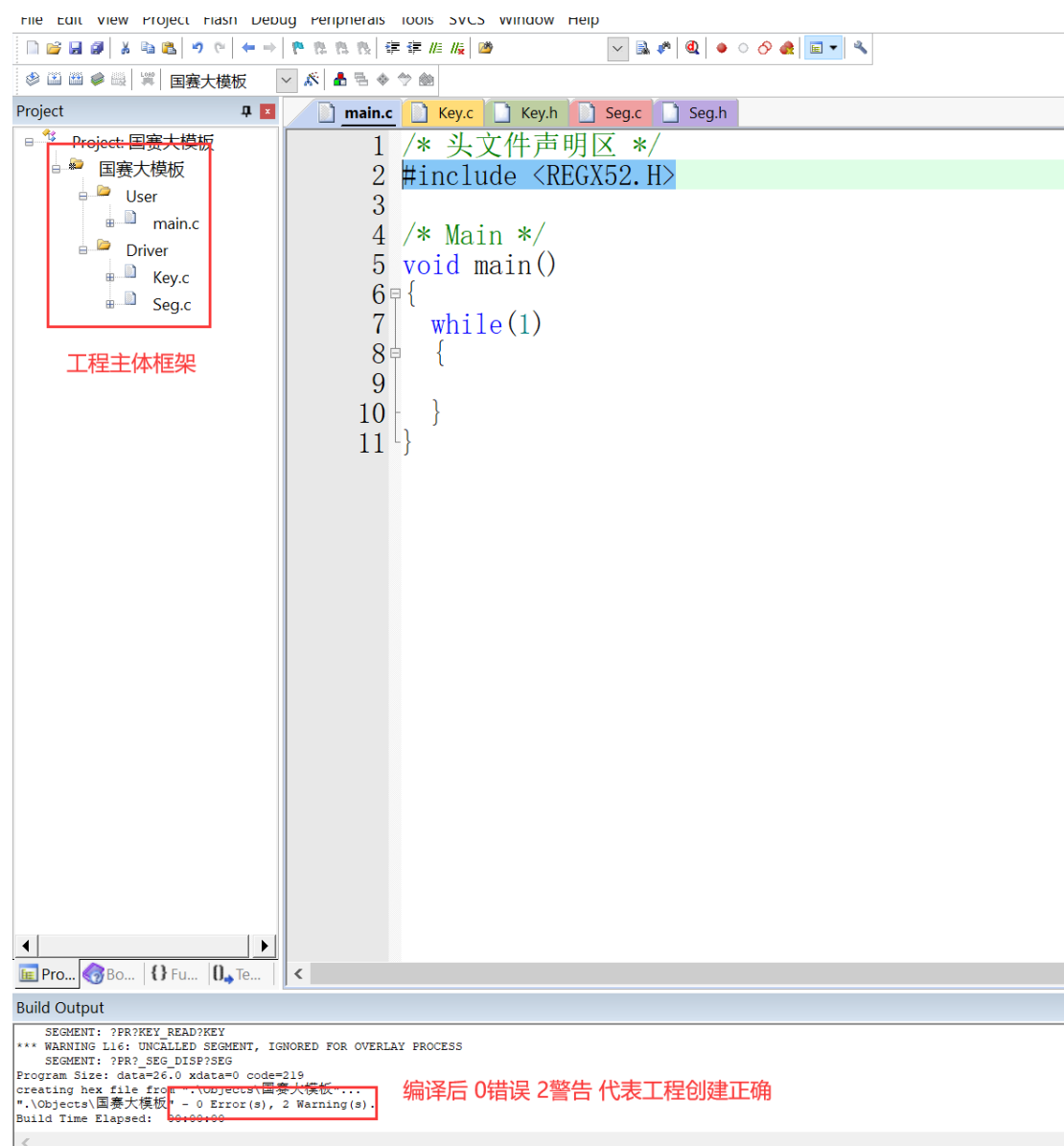


```
1#include <REGX52.H>
2
3void Seg_Dis(unsigned char wela, dula);
```

## 11、添加底层到工程中



## 12、测试工程框架是否正确



## 13、在 main.c 内编写国赛大模板（多敲就可记住）

```

7 unsigned char Key_Val, Key_Down, Key_Old; //按键专用变量
3 unsigned char Key_Slow_Down; //按键减速专用变量
9 unsigned char Seg_Buf[6] = {10, 10, 10, 10, 10, 10}; //数码管显示数据存放数组
0 unsigned char Seg_Pos; //数码管扫描专用变量
1 unsigned int Seg_Slow_Down; //数码管减速专用变量
2
3
4 /* 键盘处理函数 */
5 void Key_Proc()
6 {
7     if(Key_Slow_Down) return;
8     Key_Slow_Down = 1; //键盘减速程序
9
10    Key_Val = Key_Read(); //实时读取键码值
11    Key_Down = Key_Val & (Key_Old ^ Key_Val); //捕捉按键下降沿
12    Key_Old = Key_Val; //辅助扫描变量
13
14 }
15
16 /* 信息处理函数 */
17 void Seg_Proc()
18 {
19     if(Seg_Slow_Down) return;
20     Seg_Slow_Down = 1; //数码管减速程序
21
22 }
23
24 /* 其他显示函数 */
25 void Led_Proc()
26 {
27
28 }
29
30 /* 定时器0中断初始化函数 */
31 void Timer0Init(void) //1毫秒@12.000MHz
32 {
33     TMOD &= 0xF0; //设置定时器模式
34     TMOD |= 0x01; //设置定时器模式
35     TLO = 0x18; //设置定时初始值
36     TH0 = 0xFC; //设置定时初始值
37     TF0 = 0; //清除TF0标志
38     TR0 = 1; //定时器0开始计时
39     ET0 = 1; //定时器0中断打开
40     EA = 1; //总中断打开
41 }
42

```



```

3  /* 定时器0中断服务函数 */
4  void Timer0Server() interrupt 1
5  {
6      TL0 = 0x18;    //设置定时初始值
7      TH0 = 0xFC;    //设置定时初始值
8      if(++Key_Slow_Down == 10) Key_Slow_Down = 0; //键盘减速专用
9      if(++Seg_Slow_Down == 500) Seg_Slow_Down = 0; //数码管减速专用
0      if(++Seg_Pos == 6) Seg_Pos = 0; //数码管显示专用
1      Seg_Displ(Seg_Pos, Seg_Buf[Seg_Pos]);
2  }
3
4  /* Main */
5  void main()
6  {
7      Timer0Init();
8      while (1)
9      {
0          Key_Proc();
1          Seg_Proc();
2          Led_Proc();
3      }
4  }

```