



STM32 MCU Development

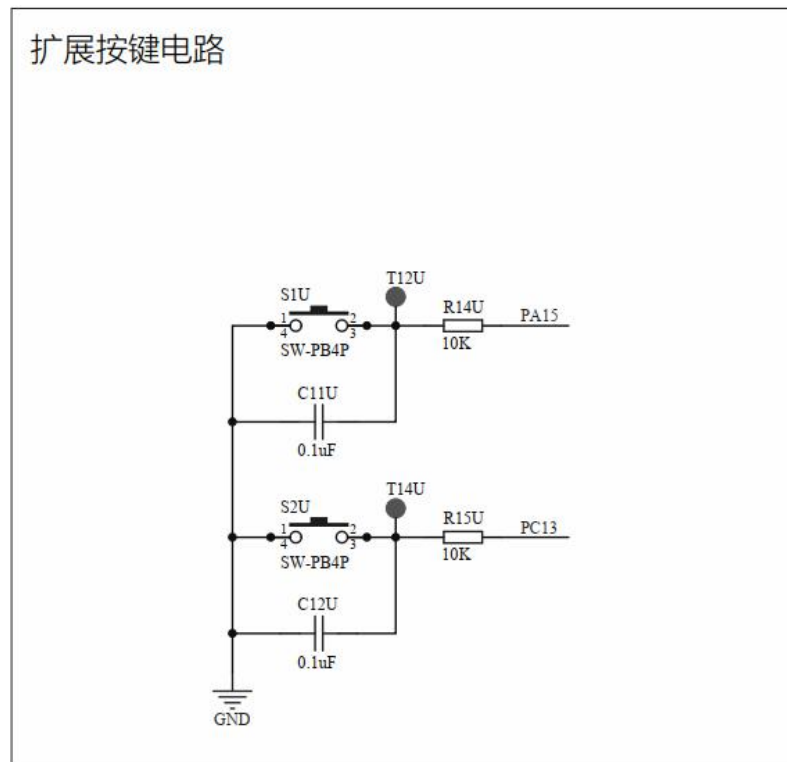
STM32 单片机开发

-- STM32 按键中断开发

目录

1. 按键工作原理分析.....	3
2. 轮询模式.....	4
2.1. 配置按键管脚为输入模式.....	4
2.2. 自动生产代码.....	4
2.3. 修改 main.c 源代码.....	5
2.4. 运行测试.....	5
3. 中断模式.....	6
3.1. 配置按键管脚为中断模式.....	6
3.2. 配置按键中断优先级.....	6
3.3. 自动生产代码.....	7
3.4. 修改 main.c 源代码.....	7
3.5. 修改 gpio.c 源代码.....	8
3.6. 运行测试：	8
4. 验证与思考.....	8

1. 按键工作原理分析



工作原理分析：

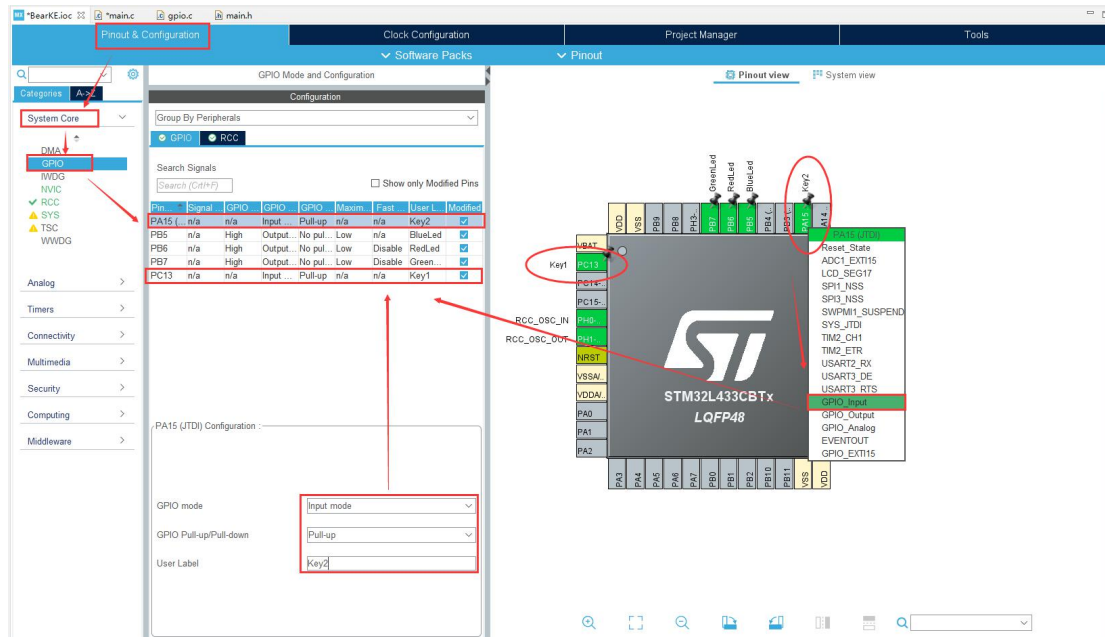
- ✓ 内部上拉、下拉；
- ✓ 轮询模式、中断模式

中断触发模式：

- 水平触发： 高电平/低电平
- 沿触发： 上升沿/下降沿

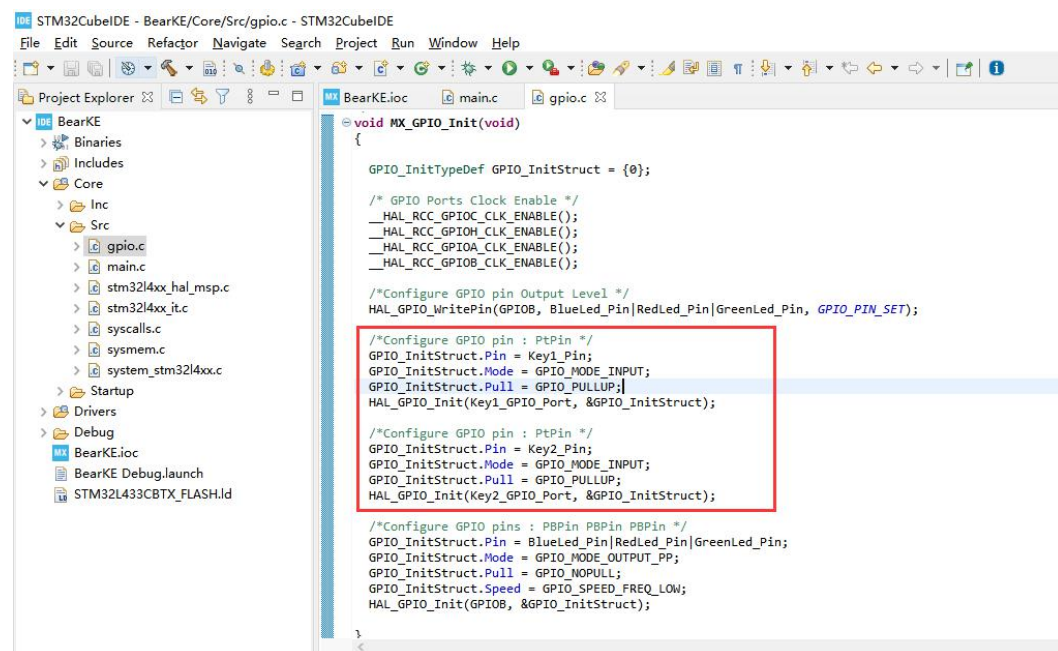
2. 轮询模式

2.1. 配置按键管脚为输入模式



2.2. 自动生产代码

任意位置按 Ctrl+S 将开始自动生成代码。



2.3. 修改 main.c 源代码

修改 main.c 中 main()函数代码，在 while()循环中添加按键探测和三色 Led 灯控制的代码。

```
... ..  
/* Infinite loop */  
/* USER CODE BEGIN WHILE */  
sysled_heartbeat();  
while (1)  
{  
    if( GPIO_PIN_RESET == HAL_GPIO_ReadPin(Key1_GPIO_Port, Key1_Pin) )  
    {  
        blink_led(BlueLed, 500);  
    }  
  
    if( GPIO_PIN_RESET == HAL_GPIO_ReadPin(Key2_GPIO_Port, Key2_Pin) )  
    {  
        blink_led(RedLed, 500);  
    }  
  
/* USER CODE END WHILE */  
  
/* USER CODE BEGIN 3 */  
}  
/* USER CODE END 3 */  
... ..
```

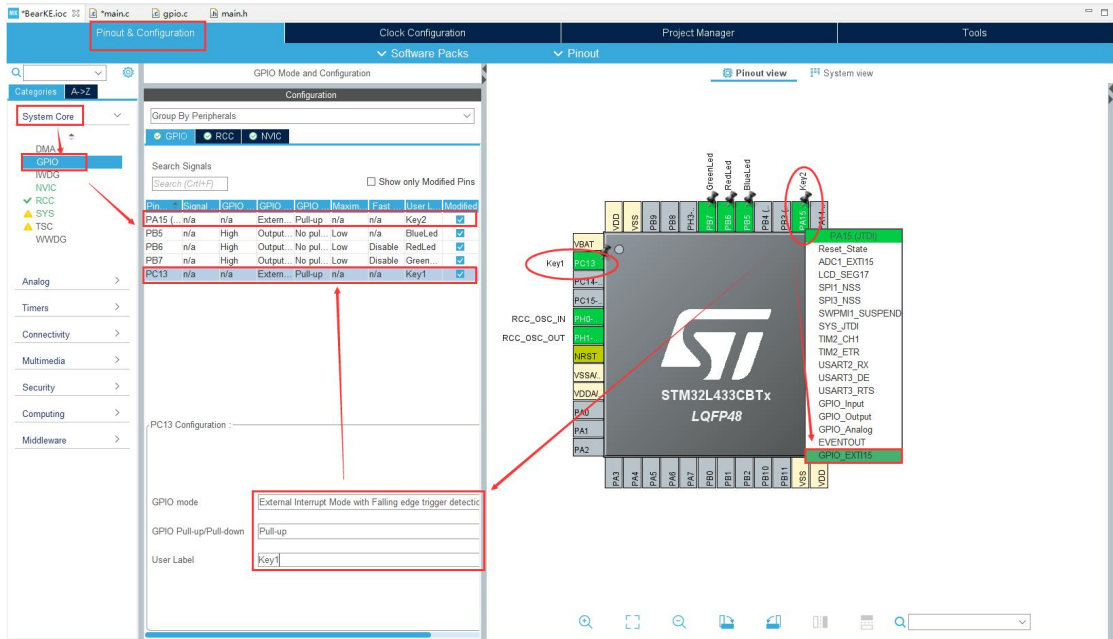
2.4. 运行测试

重新编译、烧录并运行程序，按下 Key1 时蓝色灯亮；按下 Key 时红色灯亮。在轮询模式下，while()循环里不能做其它事情，否则如果 CPU 正在执行其它代码，此时按下按键 CPU 来不及执行查询按键状态的代码，而导致错过事件。

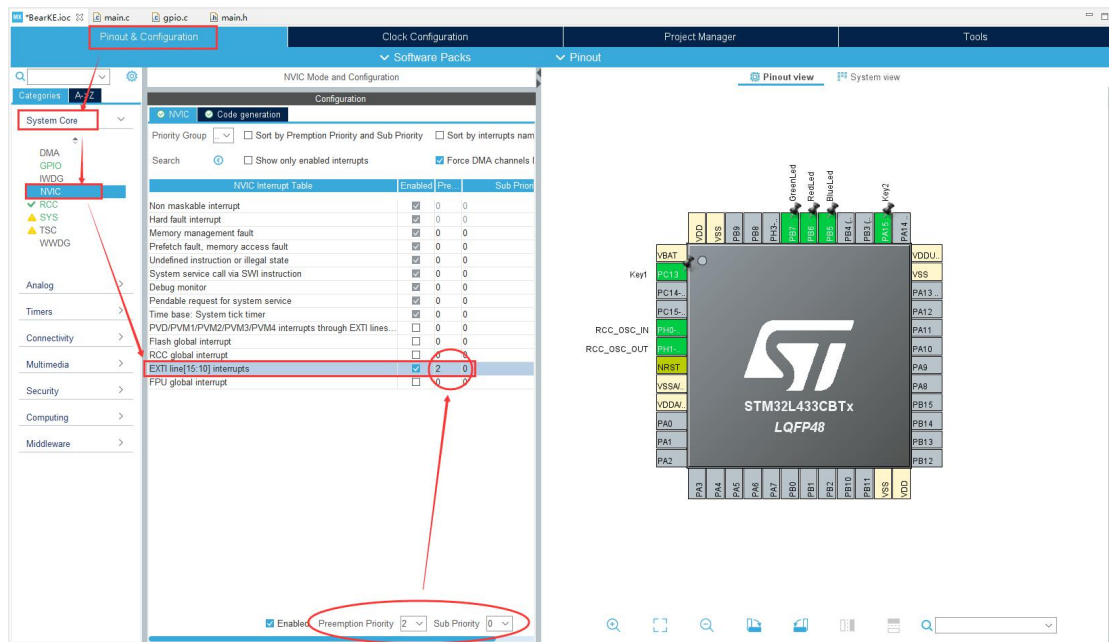
对按键或类似事件的处理，我们一般不会使用轮询模式，而是使用中断模式。

3. 中断模式

3.1. 配置按键管脚为中断模式

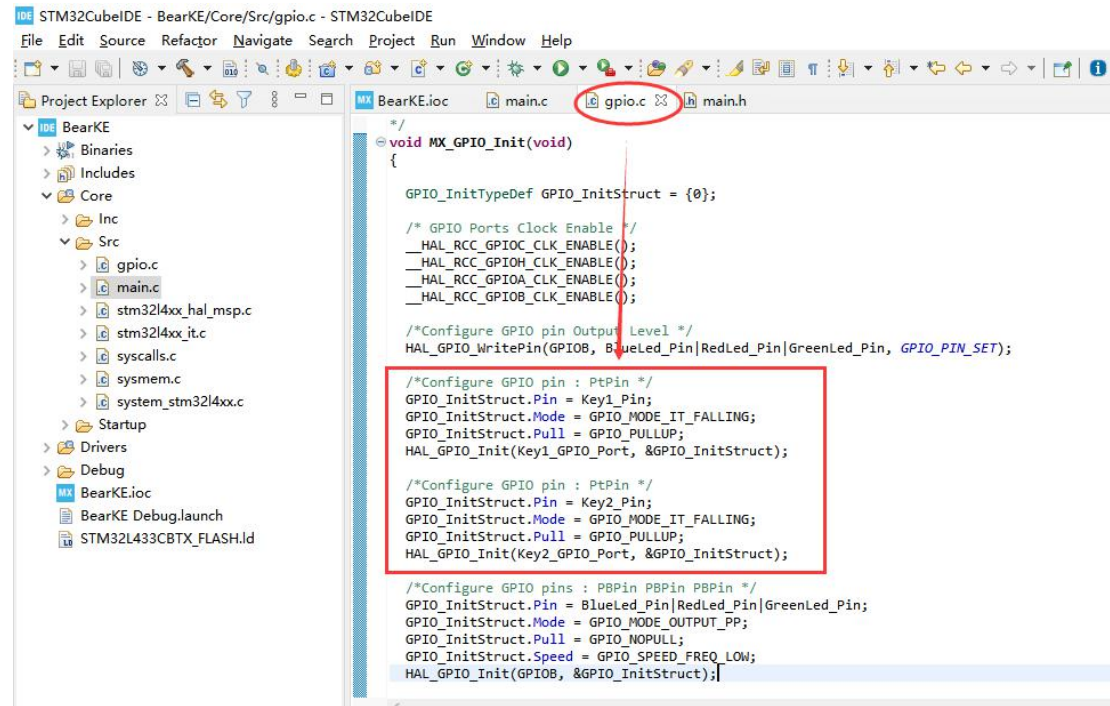


3.2. 配置按键中断优先级



3.3. 自动生产代码

任意位置按 Ctrl+S 将开始自动生成代码。



3.4. 修改 main.c 源代码

删除 main.c 文件中的 while 循环里控制三色 Led 灯的代码，在中断服务程序中控制。

```
/* USER CODE BEGIN WHILE */
sysled_hearbeat();
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}
```

3.5. 修改 gpio.c 源代码

在 gpio.c 文件的合适位置，添加按键中断服务处理程序，当按键按下时让灯闪烁一下。

```
/* USER CODE BEGIN 2 */
...
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if( Key1_Pin == GPIO_Pin )
    {
        blink_led(BlueLed, 500);
    }
    else if( Key2_Pin == GPIO_Pin )
    {
        blink_led(RedLed, 500);
    }
}
/* USER CODE END 2 */
```

3.6. 运行测试：

重新编译、烧录并运行程序，按下 Key1 时蓝色灯亮；按下 Key2 时红色灯亮。

4. 验证与思考

将按键中断的优先级调整为 0， 然后重新烧录程序运行，会看到什么现象，为什么？