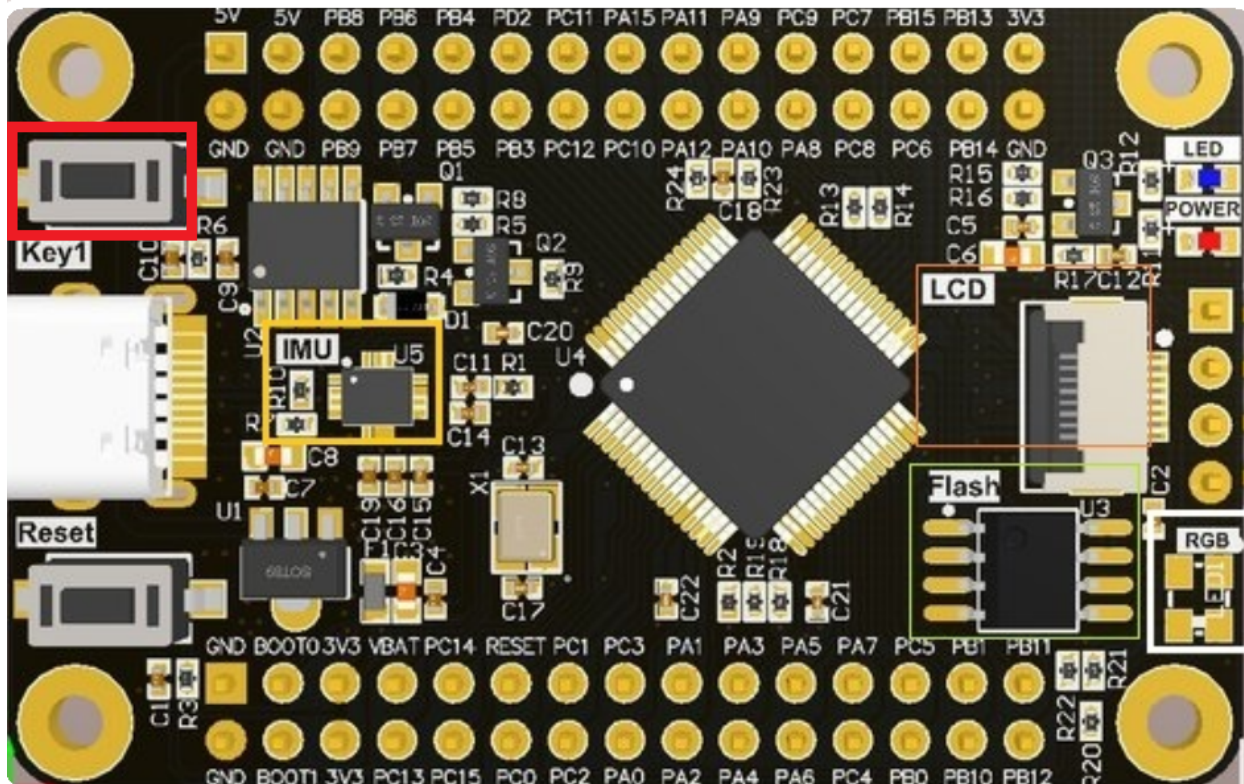
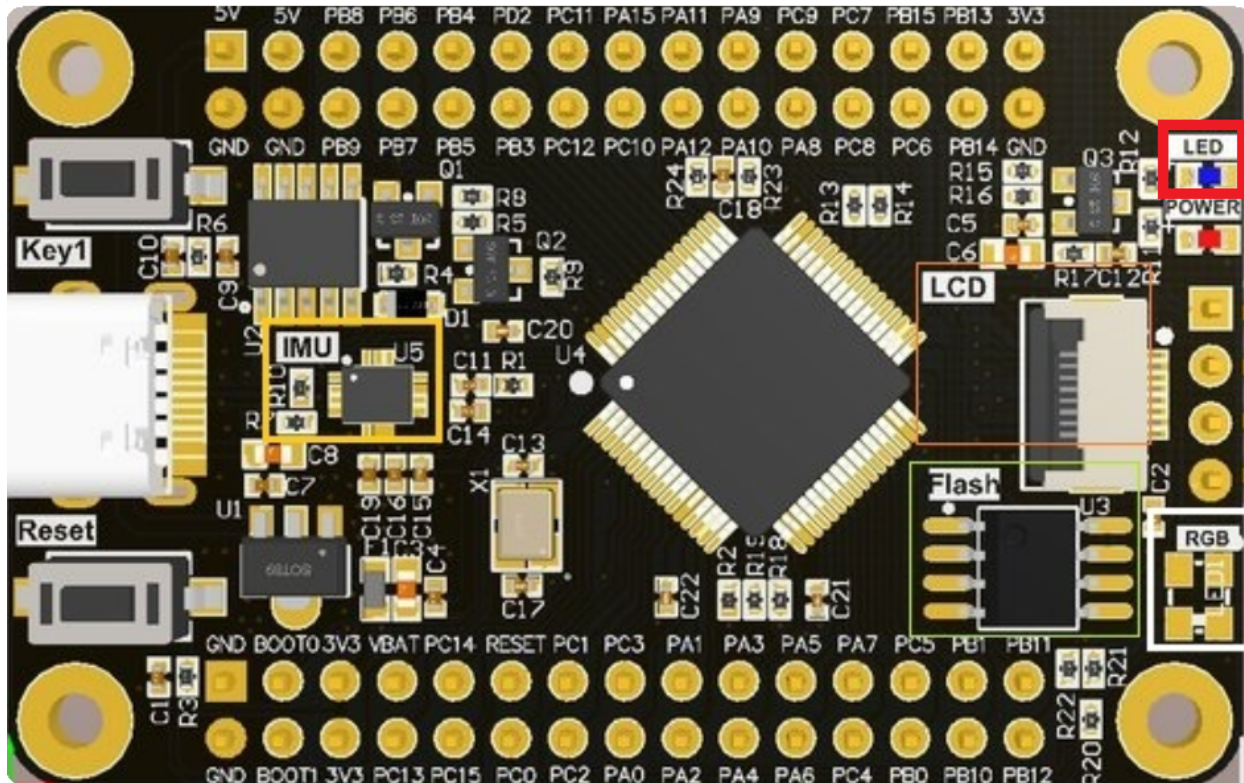


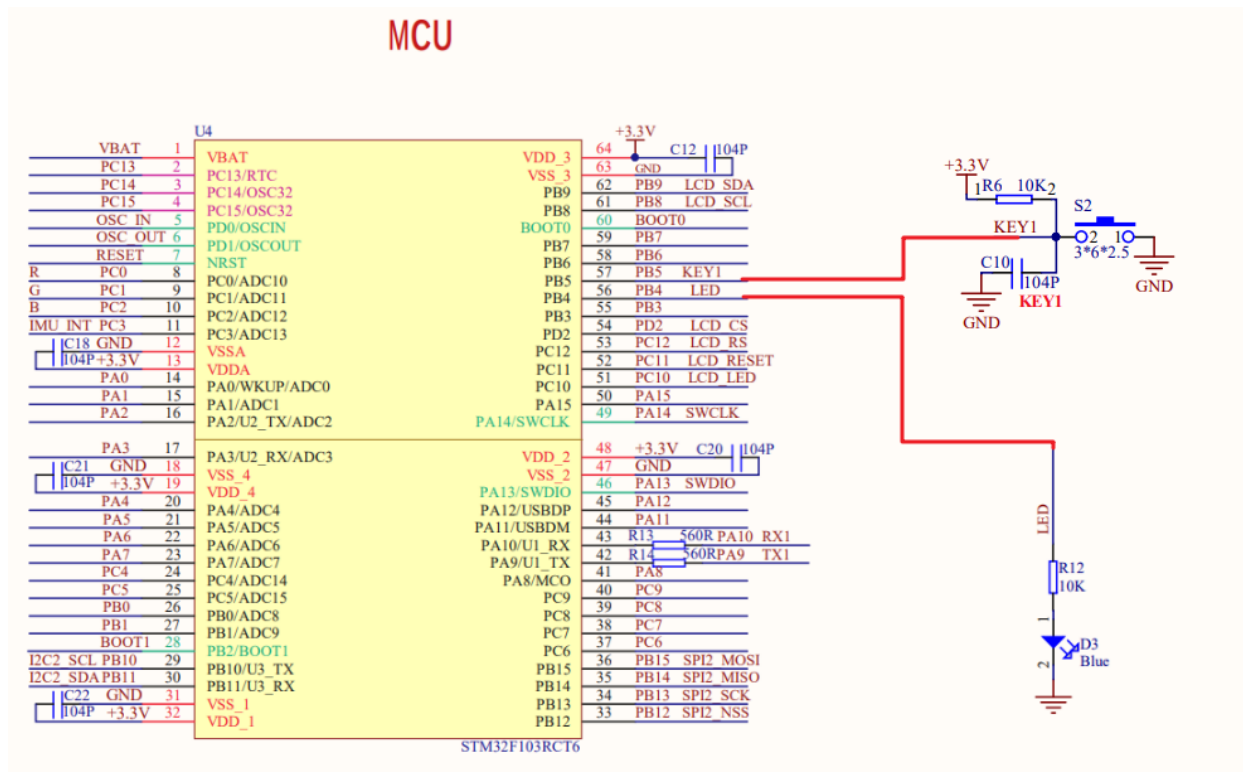
Key control LED: External interrupt

Hardware: On-board buttons + LED



Brief principle

Circuit schematic



The LED is connected to the PB4 pin, you need to pay attention to the pin configuration of PB4 (see the code for the specific configuration):

PB4 output high level, LED on;

The PB4 output is low and the LED is off.

KEY1 is connected to the PB5 pin, and the key is pressed by detecting the high and low level of the PB5 pin; By configuring PB5 to trigger an external interrupt on the falling edge, it is possible to respond accurately and quickly to the action of pressing a key.

KEY1 is not pressed, and the PB5 pin level is high;

KEY1 is pressed and the PB5 pin level is low.

The external interrupt function reverses the state of the LED, and the LED state flips once each time an interrupt is entered.

Main code

main.c

```
#include "stm32f10x.h"
#include "LED.h"
#include "EXTI.h"

int main(void)
{
    LED_Init();//LED初始化(PB4)
```

```
EXTI5_Init(); //外部中断5初始化(PB5 EXTI5)

while(1)
{

}
}
```

LED.c

```
#include "LED.h"

void LED_Init(void) //LED初始化(PB4)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    /* Enable GPIOB and AFIO clocks */
    /* 使能GPIOB和功能复用IO时钟 */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB | RCC_APB2Periph_AFIO, ENABLE);

    /* JTAG-DP Disabled and SW-DP Enabled */
    /* 禁用JTAG 启用SWD */
    GPIO_PinRemapConfig(GPIO_Remap_SWJ_JTAGDisable, ENABLE);

    /* Configure PB4 in output pushpull mode */
    /* 配置PB4 推挽输出模式 */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOB, &GPIO_InitStructure);

    /* Set the GPIOB port pin 4 */
    /* 设置PB4端口数据位 */
    GPIO_WriteBit(GPIOB, GPIO_Pin_4, Bit_RESET);
}
```

LED.h

```
#ifndef __LED_H__
#define __LED_H__

#include "stm32f10x.h"

void LED_Init(void); //LED初始化(PB4)

#endif
```

EXTI.c

```
#include "EXTI.h"

void EXTI5_Init(void)//外部中断5初始化(PB5 EXTI5)
{
    EXTI_InitTypeDef    EXTI_InitStructure;
    GPIO_InitTypeDef    GPIO_InitStructure;
    NVIC_InitTypeDef    NVIC_InitStructure;

    /* Enable GPIOB and AFIO clock */
    /* 使能GPIOB AFIO时钟 */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB | RCC_APB2Periph_AFIO, ENABLE);

    /* Configure PB5 pin as input floating */
    /* 配置PB5上拉输入模式 */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;
    GPIO_Init(GPIOB, &GPIO_InitStructure);

    /* Connect EXTI5 Line to PB5 pin */
    /* 连接EXTI5到PB5引脚 */
    GPIO_EXTILineConfig(GPIO_PortSourceGPIOB, GPIO_PinSource5);

    /* Configure EXTI5 line */
    /* 配置EXTI5 */
    EXTI_InitStructure.EXTI_Line = EXTI_Line5;
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_Init(&EXTI_InitStructure);

    /* Enable and set EXTI9_5 Interrupt to the lowest priority */
    /* 使能和配置EXTI5优先级及中断向量入口地址 */
    NVIC_InitStructure.NVIC_IRQChannel = EXTI9_5_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0x00;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x00;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);
}

void EXTI9_5_IRQHandler(void)
{
    if(EXTI_GetITStatus(EXTI_Line5) != RESET)
    {
        /* Toggle LED(PB4) */
        /* 反转LED(PB4)状态 */
        GPIO_WriteBit(GPIOB, GPIO_Pin_4, (BitAction)(1 - GPIO_ReadOutputDataBit(GPIOB,
        GPIO_Pin_4)));
        /* Clear the EXTI line 5 pending bit */
        /* 清除EXTI5中断标志位 */
    }
}
```

```
    EXTI_ClearITPendingBit(EXTI_Line5);  
}  
}
```

EXTI.h

```
#ifndef __EXTI_H__  
#define __EXTI_H__  
  
#include "stm32f10x.h"  
  
void EXTI5_Init(void);  
  
#endif
```

Phenomenon

After the download program goes in, the LED will light up, press the Reset button once, and the downloaded program will run.

Each time the key key is pressed, the LED on and off status is reversed.