计算机科学与技术学院

嵌入式系统实验报告

（四）

姓　　名：　Banban

专　　业：　计 算 机 科 学 与 技 术

班　　级：

学　　号：

指 导 教 师：

2023 年 3 月 24 日

一、任务要求

1、 任务一：跑通 demo（key1 按下触发中断，改变 LED1 状态或 beep 状态），理解外部中断操纵原理。

2、 任务二：改变 PE5 中断源为 PD12 中断源，功能不变。

3、 任务三：设 2 个按键，key1 按下导致每次加 1 并在数码管上显示，key2 按下导致每次减 1 并在数码管上显示，计数范围 0—F，数码管初始显示 0，超范围则 beep 响。

二、实验报告要求

1、 任务 2-任务 3 中自编程序的源代码（加上注释）

2、 能说明软件仿真结果的截图、反映硬件电路连接和硬件验证结果的图片或视频

三、实验过程

一. 任务一：跑通 demo，理解外部中断操纵原理

1. 代码

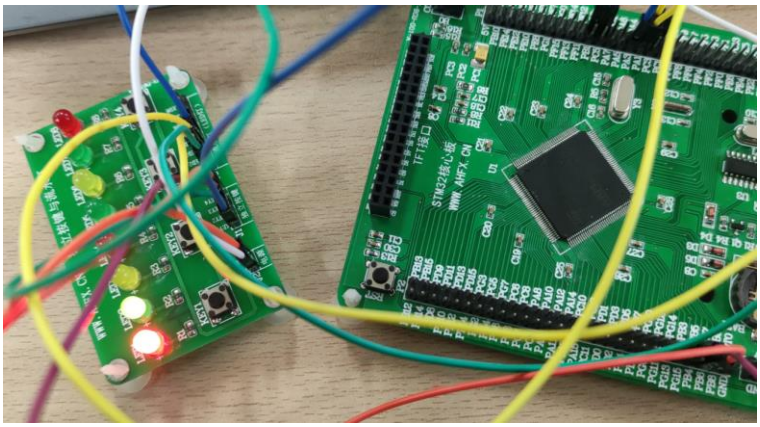
```
// stm32f10x_conf.c 所用到的固件库函数头文件
// -----------------------------------------------------------
#include "stm32f10x_exti.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"
#include "misc.h"


// exti.c
// -----------------------------------------------------------
#include "exti.h"
// 配置嵌套向量中断控制器 NVIC
static void NVIC_Configuration(void) {
  NVIC_InitTypeDef NVIC_InitStructure;
```

```
  /* 配置一个比特的抢占优先权 */
  NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);
  /* 配置 P[A|B|C|D|E]5 为中断源 */
  NVIC_InitStructure.NVIC_IRQChannel = EXTI9_5_IRQn;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
  NVIC_Init(&NVIC_InitStructure);
}

// 配置 PE5 为线中断口，并设置中断优先级
void EXTI_PE5_Config(void) {
    GPIO_InitTypeDef GPIO_InitStructure;
    EXTI_InitTypeDef EXTI_InitStructure;

    /* 配置 extiline(PE5)时钟和 AFIO 时钟 */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOE |
RCC_APB2Periph_AFIO,ENABLE);
    NVIC_Configuration();   /* config the NVIC(PE5) */

    /* EXTI line gpio config(PE5) */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;  // 上拉输入
    GPIO_Init(GPIOE, &GPIO_InitStructure);

    /* EXTI line(PE5) mode config */
    GPIO_EXTILineConfig(GPIO_PortSourceGPIOE, GPIO_PinSource5);
    EXTI_InitStructure.EXTI_Line = EXTI_Line5;
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling; //下降沿中断
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_Init(&EXTI_InitStructure);
}
```

```
// stm32f10x_it.c
// ------------------------------------------------------------
/* I/O 线中断，中断线为 PE5 */
void EXTI9_5_IRQHandler(void){
    if(EXTI_GetITStatus(EXTI_Line5) != RESET) {//确保是否产生了 EXTI Line 中断
        // LED1 取反
        GPIO_WriteBit(GPIOC, GPIO_Pin_3, (BitAction)((1-
GPIO_ReadOutputDataBit(GPIOC, GPIO_Pin_3))));
        EXTI_ClearITPendingBit(EXTI_Line5);     // 清除中断标志位
    }
}

// main.c
// ------------------------------------------------------------
#include "stm32f10x.h"
#include "exti.h"
#define ON  0
#define OFF 1

#define LED1(a) if (a) GPIO_SetBits(GPIOC,GPIO_Pin_3);  \
                   else GPIO_ResetBits(GPIOC,GPIO_Pin_3)
#define LED2(a) if (a) GPIO_SetBits(GPIOC,GPIO_Pin_4);  \
                   else  GPIO_ResetBits(GPIOC,GPIO_Pin_4)
#define LED3(a) if (a) GPIO_SetBits(GPIOC,GPIO_Pin_5);  \
                   else GPIO_ResetBits(GPIOC,GPIO_Pin_5)

void LED_GPIO_Config(void){
    GPIO_InitTypeDef GPIO_InitStructure;
    RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOC, ENABLE);
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3 ;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOC, &GPIO_InitStructure);
    GPIO_SetBits(GPIOC, GPIO_Pin_3 );    // turn off all led
}
```

```c
int main(void) {
    LED_GPIO_Config();  /* 配置 LED  */
    LED1( ON );
    EXTI_PE5_Config(); /* 外线配置 */
    while(1){ }   /* 等待中断 */
}
```

2. 图片效果



二.　任务二：改变 PE5 中断源为 PD12 中断源，功能不变

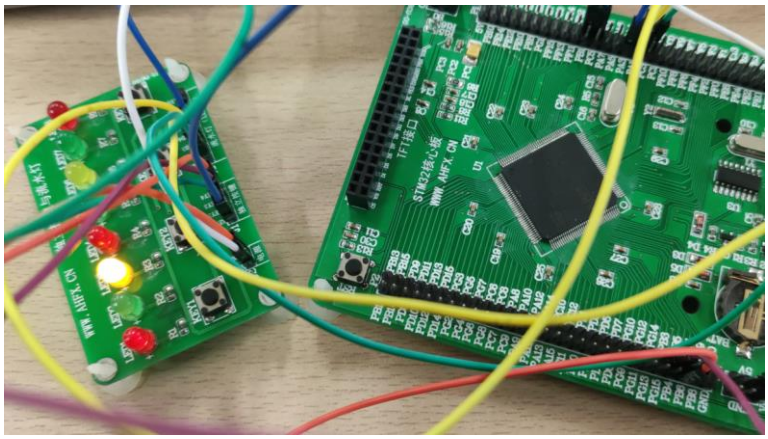
1. 代码

```c
// stm32f10x_conf.c 不变 同上
// -----------------------------------------------------------


// exti.c 改变
// -----------------------------------------------------------
#include "exti.h"
static void NVIC_Configuration(void){
  NVIC_InitTypeDef NVIC_InitStructure;
  /* Configure one bit for preemption priority */
  NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);
  /* 配置 P[A|B|C|D|E]5 为中断源 */
  NVIC_InitStructure.NVIC_IRQChannel = EXTI15_10_IRQn;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
```

```
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
  NVIC_Init(&NVIC_InitStructure);
}

void EXTI_PE5_Config(void){
    GPIO_InitTypeDef GPIO_InitStructure;
    EXTI_InitTypeDef EXTI_InitStructure;
    /* config the extiline(PD12) clock and AFIO clock */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD | RCC_APB2Periph_AFIO,
ENABLE);
    NVIC_Configuration();
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_12;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;  // 上拉输入
    GPIO_Init(GPIOD, &GPIO_InitStructure);

    GPIO_EXTILineConfig(GPIO_PortSourceGPIOD, GPIO_PinSource12);
    EXTI_InitStructure.EXTI_Line = EXTI_Line12;
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling; //下降沿中断
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_Init(&EXTI_InitStructure);
}

// stm32f10x_it.c 改变
// ------------------------------------------------------------
/* I/O 线中断，中断线为 PD12 */
void EXTI15_10_IRQHandler (void){
    // 确保是否产生了 EXTI Line 中断
    if(EXTI_GetITStatus(EXTI_Line12) != RESET) {
        // LED1 取反
        GPIO_WriteBit(GPIOC, GPIO_Pin_3,
            (BitAction)((1-GPIO_ReadOutputDataBit(GPIOC, GPIO_Pin_3))));
        EXTI_ClearITPendingBit(EXTI_Line12);     // 清除中断标志位
    }
```

```
}

// main.c 不变 同上
// ------------------------------------------------------------
```

2. 图片效果



四、总结与分析

在老师的帮助指导下，我顺利完成了本次 STM32 外部中断实验（按键中断）实验，本次实验中可以通过外部中断来检测按键事件，并触发相应的中断服务程序，从而实现按键的检测和相应的处理。

在进行 STM32 外部中断实验（按键中断）时，我发现需要注意以下内容：需要确认使用哪些 GPIO 引脚作为外部中断输入，同时需要确认中断输入信号的电平极性（上升沿触发还是下降沿触发）；需要编写相应的中断服务程序，以响应中断事件。在编写中断服务程序时，需要注意保护现场和恢复现场的问题，避免中断服务程序对系统的其他部分造成影响。

最后在进行实验时，需要对实验进行调试，检查是否能够正确地检测到按键事件并触发相应的中断服务程序。同时，还需要注意避免可能存在的噪声干扰等问题。

通过 STM32 外部中断实验（按键中断），可以更深入地了解 STM32 的外部中断功能，并掌握如何使用 STM32 的 GPIO 模块和中断寄存器来实现按键检测和相应的处理，这对于深入学习嵌入式系统编程和开发具有重要意义。