**作业三（NVIC与外部中断）**

班级 ，姓名 ，学号 ，成绩

要求：1、A4纸正反打印，手工、独立、认真完成，严禁抄袭。

2、按学号顺序收交

1、Cortex-M4 NVIC支持多少中断？STM32F407支持多少中断？

Cortex-M4 NVIC支持256级可编程中断，其中内核中断16个，核外中断240个，中断优先级分组为9组。

STM32F407支持16级92个可编程中断，其中内核中断10个，核外中断82个，中断优先级分组有5种。

2、Cortex-M4 NVIC对中断优先级是如何管理的？其中断嵌套规则是什么？

略

3、STM32F4 中，若采用中断分组2，则抢占优先级和响应优先级各占几位？若中断a的抢占优先级为2，响应优先级为1；中断b的抢占优先级为3，响应优先级为0；中断c的抢占优先级为2，响应优先级为0；中断d的抢占优先级为1，响应优先级为2；则4个中断的优先级顺序如何，嵌套规则如何？

各占2位。

优先顺序为：**中断d >中断c > 中断a > 中断b**。

中断d可以嵌套中断a、b、c；

中断a、c可以嵌套中断b

4、写出把PB2配置到外部中断线EXTI2的固件库函数调用语句。

SYSCFG\_EXTILineConfig( EXTI\_PortSourceGPIOB , EXTI\_PinSource2);

5、简述stm32中中断和事件的区别，相比中断使用事件方式有什么好处？

略

6、写出设置中断分组2，使能EXTI0，并将其初始化为抢占优先级为2，响应优先级为1的代码。

NVIC\_InitTypeDef NVIC\_InitStructure;

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2);//设置中断分组2

NVIC\_InitStructure.NVIC\_IRQChannel = EXTI0\_IRQn;//EXTI0中断

NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority=0x02 ;// 抢占优先级为2

NVIC\_InitStructure.NVIC\_IRQChannelSubPriority = 0x01;// 响应优先级为1

NVIC\_InitStructure.NVIC\_IRQChannelCmd = ENABLE;//IRQ通道使能

NVIC\_Init(&NVIC\_InitStructure); //根据上面指定的参数初始化NVIC寄存器

7、将EXTI2设置为上升沿和下降沿均可触发的中断，并将其使能，写出相应的代码。

EXTI\_InitTypeDef EXTI\_InitStructure ;

EXTI\_InitStructure.EXTI\_Line=EXTI.Line2; //指定需要配置的中断线EXTI2

EXTI\_InitStructure.EXTI\_Mode=EXTI\_Mode\_Interrupt;//设置触发模式为中断

EXTI\_InitStructure.EXTI\_Trigger= EXTI\_Trigger\_Rising |EXTI\_Trigger\_Falling;//上下降沿触//发或EXTI\_InitStructure.EXTI\_Trigger= EXTI\_Trigger\_Rising\_Falling;//上下降沿触发

EXTI\_InitStructure.EXTI\_LineCmd=ENABLE;//使能

EXTI\_Init(&EXTI\_InitStructure);//EXTI初始化

8、若PA5连接一LED指示灯（PA5为0时灯亮，为1时灯灭），PB5连接一按钮开关（开关闭合时为低电平）。现欲实现开关每按一次，LED的状态翻转一次。请用中断方式编写实现该功能的代码。

//LED、KEY 端口定义

#define LED PAout(5) // LED接PA5

#define KEY GPIO\_ReadInputDataBit(GPIOB,GPIO\_Pin\_5) //按钮开关接PB5

//LED IO 初始化

void LED\_Init(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_AHB1PeriphClockCmd(RCC\_AHB1Periph\_GPIOA, ENABLE);//使能 GPIOA 时钟

//GPIOA5初始化设置

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_5;//LED0 对应 IO 口

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_OUT;//普通输出模式

GPIO\_InitStructure.GPIO\_OType = GPIO\_OType\_PP;//推挽输出

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_100MHz;//100MHz

GPIO\_InitStructure.GPIO\_PuPd = GPIO\_PuPd\_UP;//上拉

GPIO\_Init(GPIOA, &GPIO\_InitStructure);//初始化 GPIO

GPIO\_SetBits(GPIOA,GPIO\_Pin\_5);//GPIOA5 设置高，灯灭

}

//按键初始化函数

void KEY\_Init(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_AHB1PeriphClockCmd(RCC\_AHB1Periph\_GPIOB, ENABLB);//使能 GPIOB时钟

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_5; //KEY 对应引脚

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_IN;//普通输入模式

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_100MHz;//100M

GPIO\_InitStructure.GPIO\_PuPd = GPIO\_PuPd\_UP;//上拉

GPIO\_Init(GPIOB, &GPIO\_InitStructure);//初始化 GPIOB

}

//外部中断初始化程序

void EXTIX\_Init(void)

{

NVIC\_InitTypeDef NVIC\_InitStructure;

EXTI\_InitTypeDef EXTI\_InitStructure;

KEY\_Init(); //按键对应的 IO 口初始化

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_SYSCFG, ENABLE);//使能 SYSCFG 时钟

SYSCFG\_EXTILineConfig(EXTI\_PortSourceGPIOB, EXTI\_PinSource5);//PB5 连接线5

/\* 配置 EXTI\_Line5 \*/

EXTI\_InitStructure.EXTI\_Line = EXTI\_Line5;//LINE5

EXTI\_InitStructure.EXTI\_Mode = EXTI\_Mode\_Interrupt;//中断事件

EXTI\_InitStructure.EXTI\_Trigger = EXTI\_Trigger\_Rising; //上升沿触发

EXTI\_InitStructure.EXTI\_LineCmd = ENABLE;//使能 LINE5

EXTI\_Init(&EXTI\_InitStructure);

/\* 配置 NVIC \*/

NVIC\_InitStructure.NVIC\_IRQChannel = EXTI9\_5\_IRQn; //EXTI9\_5通道

NVIC\_InitStructure.NVIC\_IRQChannelPreemptionPriority = 0x00;//抢占优先级0

NVIC\_InitStructure.NVIC\_IRQChannelSubPriority = 0x02; //响应优先级2

NVIC\_InitStructure.NVIC\_IRQChannelCmd = ENABLE; //使能

NVIC\_Init(&NVIC\_InitStructure); //NVIC初始化

}

//外部中断服务程序

void EXTI9\_5\_IRQHandler(void)

{

delay\_ms(10); //消抖

if(KEY==0) LED=!LED;

EXTI\_ClearITPendingBit(EXTI\_Line5); //清除 LINE5 上的中断标志位

}

int main(void)

{

NVIC\_PriorityGroupConfig(NVIC\_PriorityGroup\_2);//设置系统中断优先级分组 2

delay\_init(168); //初始化延时函数

uart\_init(115200); //串口初始化

LED\_Init(); //初始化 LED 端口

EXTIX\_Init(); //初始化外部中断输入

LED = 0; //先点亮红灯

while (1) ; //无限循环，等待中断

}