Homework for lecture 4

Write a summary report:

(1) LED turn on and off alternately (Based on FWLib and Modularization, use a delay function to generate a time delay)

(2) Press the key to control the change between LED on and off (in the report, the programming flow chart is necessary too)

LED turn on and off alternately (Based on FWLib and Modularization, use a delay function to generate a time delay):

FWLib:

```
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
void delay(void)
{
    uint32_t i=0x200000;
    while(i--);
}
int main(void)
{
    GPIO_InitTypeDef GPIO_InitStruct;
    //打开 GPIOB 端口的时钟
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB,ENABLE);
    //配置 IO 口为输出
    GPIO_InitStruct.GPIO_Mode=GPIO_Mode_Out_PP;
    GPIO_InitStruct.GPIO_Pin=GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1;
    GPIO_InitStruct.GPIO_Speed=GPIO_Speed_10MHz;
    GPIO_Init(GPIOB,&GPIO_InitStruct);

    GPIO_InitTypeDef GPIO_InitStruct1;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA,ENABLE);
    GPIO_InitStruct1.GPIO_Mode=GPIO_Mode_Out_PP;
    GPIO_InitStruct1.GPIO_Pin=GPIO_Pin_8;
    GPIO_InitStruct1.GPIO_Speed=GPIO_Speed_10MHz;
    GPIO_Init(GPIOA,&GPIO_InitStruct1);

    //点灯
    while(1)
    {
        GPIO_ResetBits(GPIOB,GPIO_Pin_5); //LED on
      delay();
      GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
      delay();
        GPIO_ResetBits(GPIOB,GPIO_Pin_0); //LED on
      delay();
      GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);   //LED off
      delay();
        GPIO_ResetBits(GPIOB,GPIO_Pin_1); //LED on
      delay();
      GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);   //LED off
      delay();
        GPIO_ResetBits(GPIOB,GPIO_Pin_5);
        GPIO_ResetBits(GPIOB,GPIO_Pin_0);
        delay();
      GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);   //LED off
      delay();
        GPIO_ResetBits(GPIOB,GPIO_Pin_0);
        GPIO_ResetBits(GPIOB,GPIO_Pin_1);
        delay();
      GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);   //LED off
      delay();
        GPIO_ResetBits(GPIOB,GPIO_Pin_5);
        GPIO_ResetBits(GPIOB,GPIO_Pin_1);
        delay();
      GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);   //LED off
      delay();
        GPIO_ResetBits(GPIOA,GPIO_Pin_8);   //BEEP off
        delay();
        GPIO_SetBits(GPIOA,GPIO_Pin_8);     //BEEP on
        delay();
        GPIO_ResetBits(GPIOA,GPIO_Pin_8);   //BEEP off
    }
}
```

Modularization:

```
#include "stm32f10x.h"
#include "led.h"
void delay(void)
{
    uint32_t i=0x200000;
    while(i--);
}

int main(void)
{
    LED_GPIO_Config();
    while(1)
    {
      LED_ON();
        delay();
      LED_OFF();
        delay();
    }
}
```

Press the key to control the change between LED on and off (in the report, the programming flow chart is necessary too):

1. include stm32f10.h、led.h
2. 准备 delay 函数
3. 准备 key_scan 函数
4. 打开对应端口
5. 设置初始值
6. 进入循环 while(1)
7. 检测到按键，则 count++，同时设置安全保障使其不大于 6
8. 根据 count 分配灯的输出值

代码：

```
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
void delay(void)
{
    uint32_t i=0x200000;
    while(i--);
}
uint16_t Key_Scan(void)
{
    if( GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0) == 1 )
    {
        delay();
        if(GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0) == 1)
        {
            // 松手检测
            while(GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0) == 1);
            delay();
        }
        return 1;
    }
    else
    {
        return 0;
    }
}
```

```
int main(void)
{
    GPIO_InitTypeDef GPIO_InitStruct;
    //打开 GPIOB 端口的时钟
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
    //配置 IO 口 GPIO_Pin_5,GPIO_Pin_0,GPIO_Pin_1 为输出
    GPIO_InitStruct.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_InitStruct.GPIO_Pin = GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1;
    GPIO_InitStruct.GPIO_Speed = GPIO_Speed_10MHz;
    GPIO_Init(GPIOB, &GPIO_InitStruct);

    GPIO_InitTypeDef GPIO_InitStruct1;
    //打开 GPIOA 端口的时钟
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
    //配置 IO 口 GPIO_Pin_0 为输出
    GPIO_InitStruct1.GPIO_Mode = GPIO_Mode_IPU;
    GPIO_InitStruct1.GPIO_Pin = GPIO_Pin_0;
    GPIO_InitStruct1.GPIO_Speed = GPIO_Speed_10MHz;
    GPIO_Init(GPIOA, &GPIO_InitStruct1);


    GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
    uint16_t count = 0;

    while(1)
    {

        if(Key_Scan() == 1)
        {
            count = count + 1;
        }
        if(count>=7)
        {
            count = count - 7;
        }

        if(count == 1)
        {
            GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
            GPIO_ResetBits(GPIOB,GPIO_Pin_5); //LED red on
        }
        if(count == 2)
        {
            GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
            GPIO_ResetBits(GPIOB,GPIO_Pin_0); //LED green on
        }
        if(count == 3)
        {
            GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
            GPIO_ResetBits(GPIOB,GPIO_Pin_1); //LED blue on
        }
        if(count == 4)
        {
            GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
            GPIO_ResetBits(GPIOB,GPIO_Pin_5); //LED red on
            GPIO_ResetBits(GPIOB,GPIO_Pin_0); //LED green on
        }
        if(count == 5)
        {
            GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
            GPIO_ResetBits(GPIOB,GPIO_Pin_0); //LED green on
            GPIO_ResetBits(GPIOB,GPIO_Pin_1); //LED blue on
        }
        if(count == 6)
        {
            GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
            GPIO_ResetBits(GPIOB,GPIO_Pin_5); //LED red on
            GPIO_ResetBits(GPIOB,GPIO_Pin_1); //LED blue on
        }
        if(count == 0)
        {
            GPIO_SetBits(GPIOB,GPIO_Pin_5|GPIO_Pin_0|GPIO_Pin_1);    //LED off
        }

    }
}
```

Homework for lecture 5

(1) Sort out the process of NVIC configuration. Try to delete the relevant code in the example and write it yourself in Project “LED_BasicTImer”.
(2) Modify the project “定时器中断（需修改）”, run it in your board. Record what you have modified.

Sort out the process of NVIC configuration. Try to delete the relevant code in the example and write it yourself in Project “LED_BasicTImer”.

```
static void BASIC_TIM_NVIC_Config(void)
{
	NVIC_InitTypeDef NVIC_InitStructure; //定义结构体
	RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM6,ENABLE); //打开时钟
	NVIC_InitStructure.NVIC_IRQChannel = TIM6_IRQn ; // 设置中断来源为TIM6
	NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0; // 设置主优先级为 0
	NVIC_InitStructure.NVIC_IRQChannelSubPriority = 3;  // 设置抢占优先级为3
	NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE; //使能中断
	NVIC_Init(&NVIC_InitStructure); //输入结构体
}
```

Modify the project “定时器中断（需修改）”, run it in your board. Record what you have modified.

in led.h:

A8是蜂鸣器的管脚，想要实现闪灯，应该使用B5或B0或B1

```
#ifndef __LED_H__
#define __LED_H__

#include "sys.h"

#define LED_GPIO_PIN      GPIO_Pin_5//原来是GPIO_Pin_8
#define LED_GPIO_PORT    GPIOB//原来是GPIOA
#define LED_GPIO_CLK    RCC_APB2Periph_GPIOB//原来是GPIOA

void LED_GPIO_Config(void);
void LED_Change(void);

#endif
```