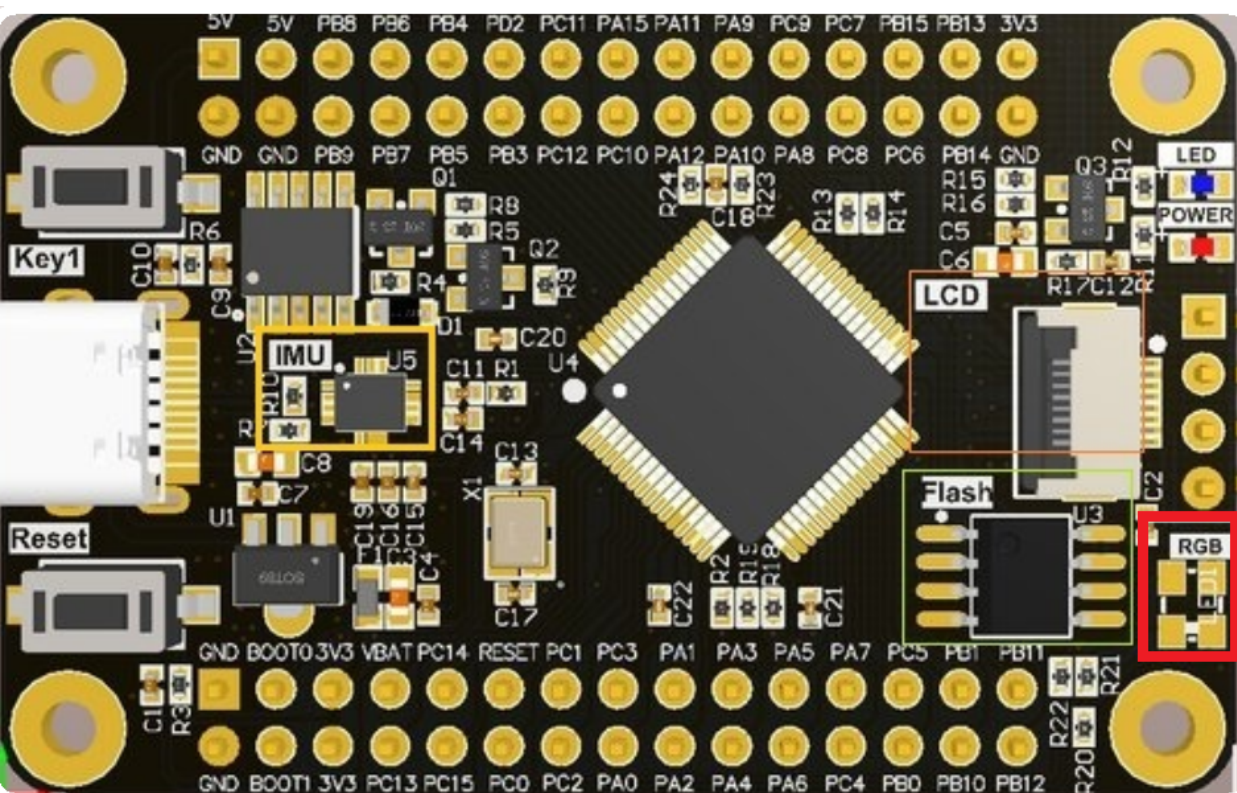


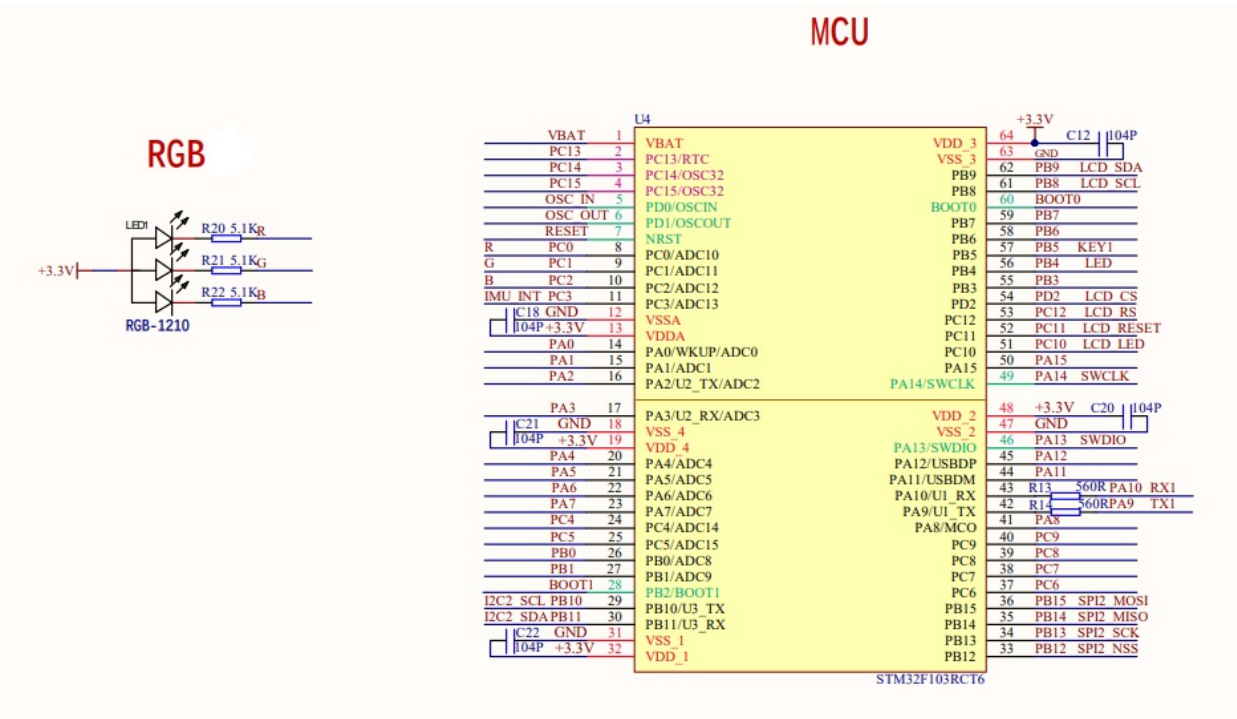
RGB lamp module

Hardware: On-board RGB



Brief principle

Circuit schematic



The RGB lamp module corresponds to 3 LED lights, which can be lit separately to achieve three colors of red, green and blue, and other colors can be realized with combination (it can be seen in the code that different LED combinations light up and display different colors).

3 LEDs in RGB lighting mode:

Low level lighting

High level off

Main code

main.c

```
#include "stm32f10x.h"
#include "SysTick.h"
#include "LED.h"
#include "RGB.h"

int main(void)
{
    SysTick_Init(); //滴答定时器初始化
    LED_Init(); //LED初始化(PB4)
    RGB_Init(); //RGB(R: PC0 G: PC1 B: PC2)

    while(1)
    {
        RGB_Control('R'); //RGB: 红色
        Delay_us(1000000); //延时1秒
        RGB_Control('G'); //RGB: 绿色
        Delay_us(1000000); //延时1秒
        RGB_Control('B'); //RGB: 蓝色
        Delay_us(1000000); //延时1秒
        RGB_Control('Y'); //RGB: 黄色
        Delay_us(1000000); //延时1秒
        RGB_Control('C'); //RGB: 青色
        Delay_us(1000000); //延时1秒
        RGB_Control('W'); //RGB: 白色
        Delay_us(1000000); //延时1秒
    }
}
```

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

```

SysTick.c

```

#include "SysTick.h"

unsigned int Delay_Num;

void SysTick_Init(void) //滴答定时器初始化
{
    while(SysTick_Config(72)); //设置重装载值 72 对应延时函数为微秒级
    //若将重装载值设置为72000 对应延时函数为毫秒级
    SysTick->CTRL &= ~(1 << 0); //定时器初始化后关闭，使用再开启
}

void Delay_us(unsigned int NCount) //微秒级延时函数
{
    Delay_Num = NCount;
    SysTick->CTRL |= (1 << 0); //开启定时器
    while(Delay_Num);
    SysTick->CTRL &= ~(1 << 0); //定时器初始化后关闭，使用再开启
}

```

```

void SysTick_Handler(void)
{
    if(Delay_Num != 0)
    {
        Delay_Num--;
    }
}

```

SysTick.h

```

#ifndef __SYSTICK_H__
#define __SYSTICK_H__

#include "stm32f10x.h"

void SysTick_Init(void); //滴答定时器初始化
void Delay_us(unsigned int NCount); //微秒级延时函数

#endif

```

RGB.c

```

#include "RGB.h"

void RGB_Init(void) //RGB(R: PC0 G: PC1 B: PC2)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    /* GPIOC Periph clock enable */
    /* 使能GPIOC时钟 */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOC, ENABLE);

    /* Configure PC0, PC1 and PC2 in output pushpull mode */
    /* 配置PC0 PC1 PC2 通用推挽输出模式 */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_Init(GPIOC, &GPIO_InitStructure);

    /* Set the GPIOC port pins */
    /* 设置PC0 1 2端口数据位 */
    GPIO_WriteBit(GPIOC, GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2, Bit_RESET);
}

void RGB_Control(char Color) //控制RGB显示三种颜色
{
    switch(Color)
    {

```

```

    case 'R':// Red          RGB: 1 0 0(数字是表示对应灯需要亮 并不是对应引脚电平 RGB灯低
电平点亮 刚好相反)
        GPIO_WriteBit(GPIOC, GPIO_Pin_0 , Bit_RESET);
        GPIO_WriteBit(GPIOC, GPIO_Pin_1 | GPIO_Pin_2, Bit_SET);
        break;
    case 'G':// Green      RGB: 0 1 0
        GPIO_WriteBit(GPIOC, GPIO_Pin_1 , Bit_RESET);
        GPIO_WriteBit(GPIOC, GPIO_Pin_0 | GPIO_Pin_2, Bit_SET);
        break;
    case 'B':// Blue       RGB: 0 0 1
        GPIO_WriteBit(GPIOC, GPIO_Pin_2 , Bit_RESET);
        GPIO_WriteBit(GPIOC, GPIO_Pin_0 | GPIO_Pin_1, Bit_SET);
        break;
    case 'Y':// Yellow    RGB: 1 1 0
        GPIO_WriteBit(GPIOC, GPIO_Pin_2 , Bit_SET);
        GPIO_WriteBit(GPIOC, GPIO_Pin_0 | GPIO_Pin_1, Bit_RESET);
        break;
    case 'C':// Cyan       RGB: 0 1 1
        GPIO_WriteBit(GPIOC, GPIO_Pin_0 , Bit_SET);
        GPIO_WriteBit(GPIOC, GPIO_Pin_1 | GPIO_Pin_2, Bit_RESET);
        break;
    case 'W':// white     RGB: 1 1 1
        GPIO_WriteBit(GPIOC, GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2, Bit_RESET);
        break;
}
}

```

RGB.h

```

#ifndef __RGB_H__
#define __RGB_H__

#include "stm32f10x.h"

void RGB_Init(void);//RGB(R: PC0 G: PC1 B: PC2)
void RGB_Control(char Color);//控制RGB显示三种颜色

#endif

```

Phenomenon

After downloading the program, the LED will be lit, press the Reset button once, the downloaded program will run, and RGB will switch six colors of red, green, blue, yellow, cyan and white.

Note: Some of the color switches will be mixed with other colors, because the brightness of the lamp is not the same, so the corresponding color will be highlighted.