



南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

计算机科学与工程学院

“嵌入式系统”实验报告书

题目：嵌入式系统作业 1——GPIO

学号：922106840127

姓名：刘宇翔

成绩

日期：2025 年 3 月 3 日

1 题目要求

1. 题目设计要求

熟悉开发板功能，完成通过按键控制 LED 灯的亮灭：按下“RIGHT”键，两个 LED 灯同时亮，否则两个 LED 灭！

2. 拟实现的具体功能

当开发板通电的时候，两个 LED 灯默认不亮，按下 right 键时两个 LED 灯同时亮，松开的时候两个 LED 灯同时灭，完成本次 gpio 实验功能。

我采用了 STM32CubeMX 和 RT-Tread Studio 两个不同的 IDE 分别完成了上述的相关内容。

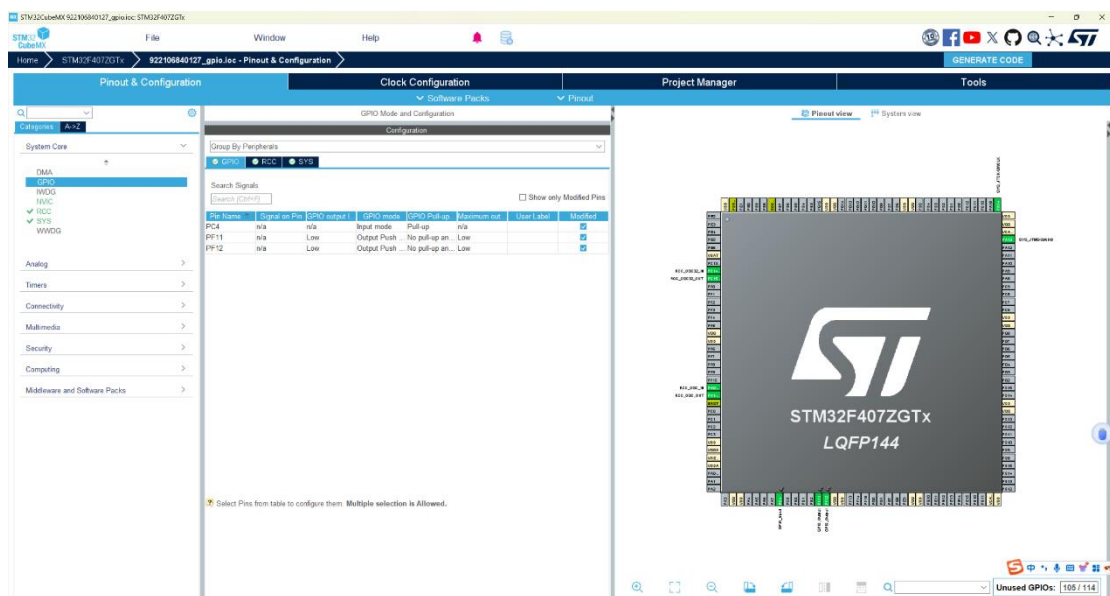
2 总体设计

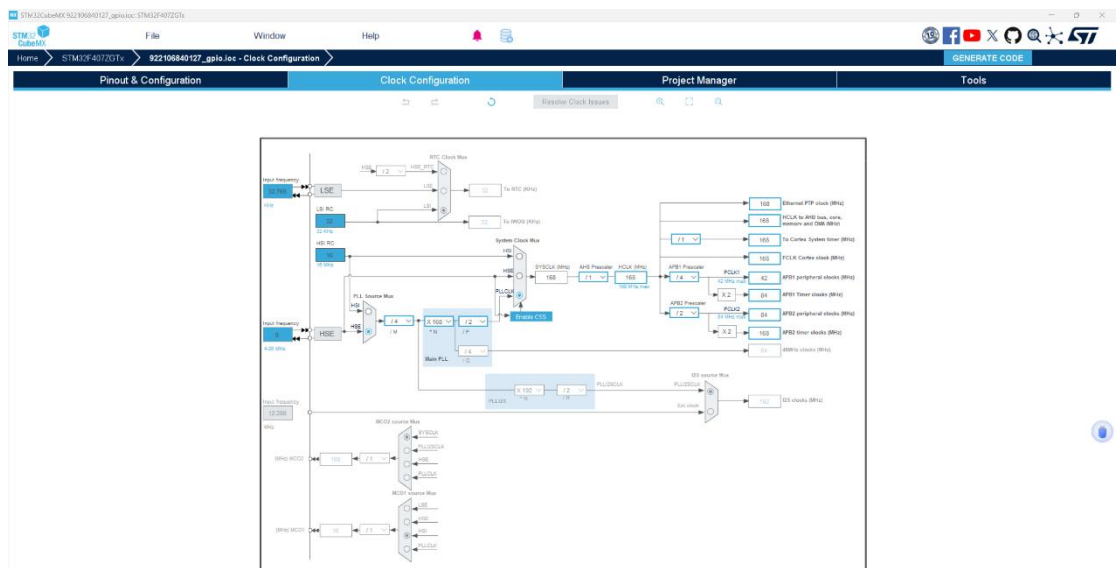
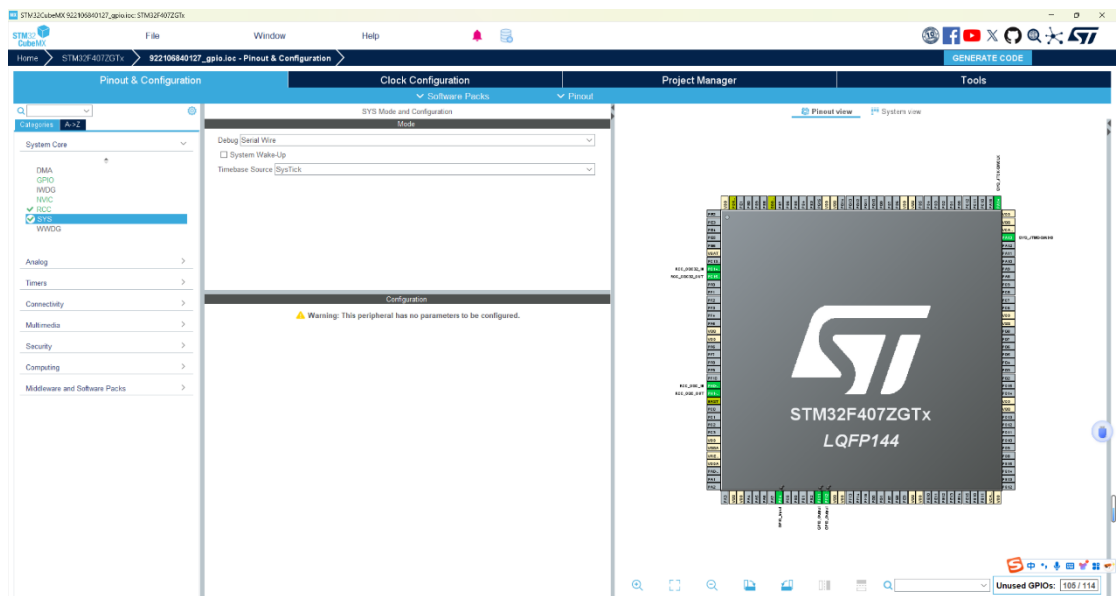
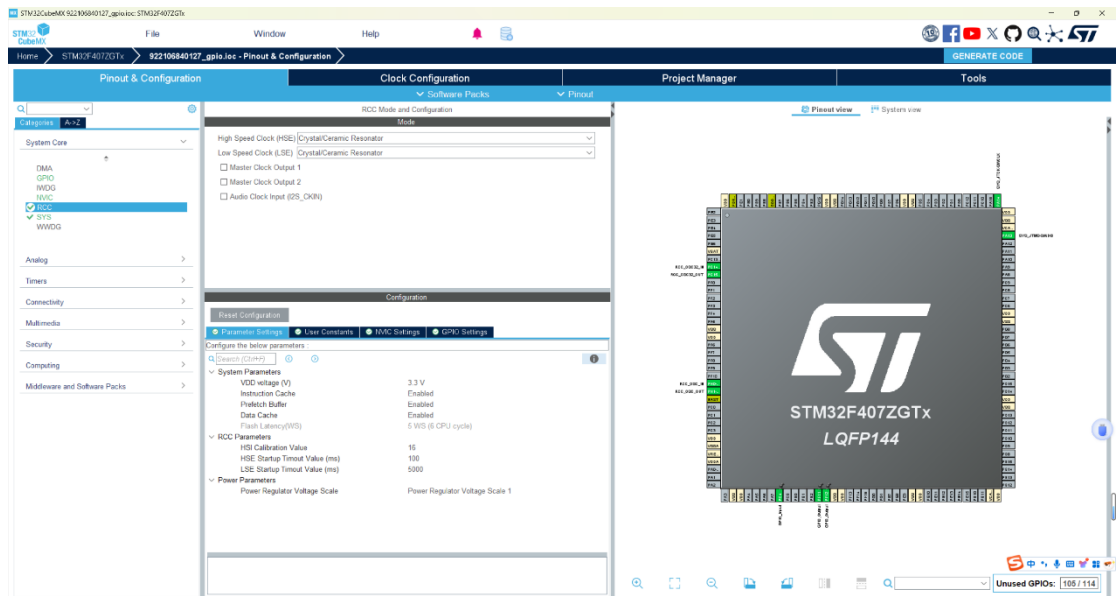
2.1 硬件设计

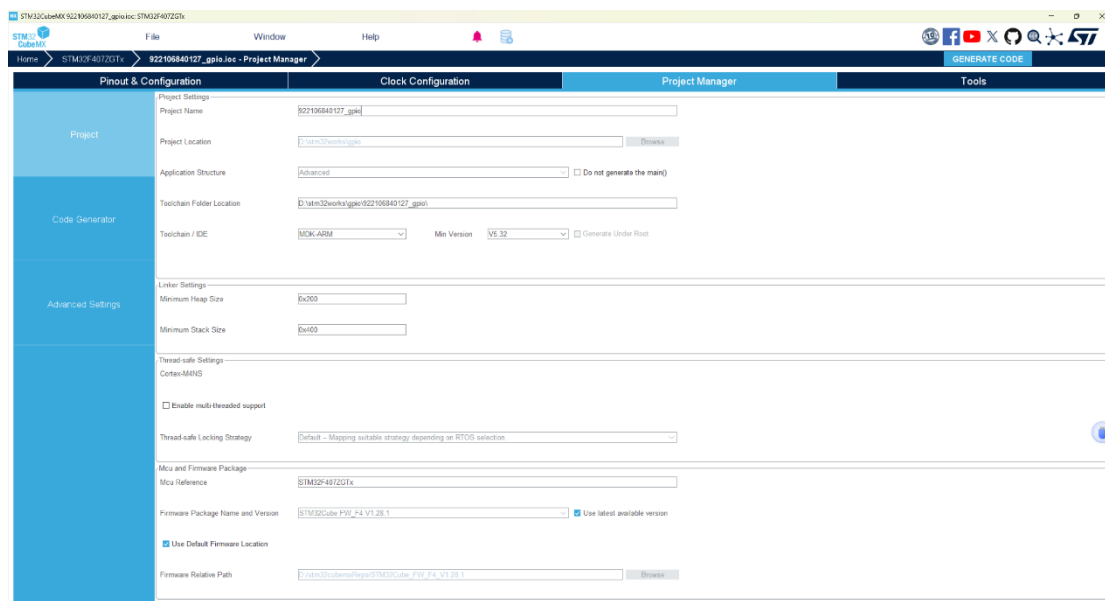
1. 硬件设计思路

我通过查阅相关开发板原理图确定了两个 LED 灯和一个“right”按键的引脚值，采用的是 PF11/PF12 以及 PC4。

2. CubeMX 图形化设计流程&关键参数设置图







以上配置图是我作为 STM32CUBEMX 进行的配置设置，设置后点击“Generate Code”初始化代码。

3. RT-Tread Studio 的对应配置过程

Project name: 922106840127_gpio

☒ 使用缺省位置(D)

位置(L): D:\RT-ThreadStudio\workspace\922106840127_gpio 浏览(B)...

☐ 基于芯片 ☒ 基于开发板

开发板: STM32F407-RT-SPARK

BSP: 1.2.0

类型: 示例工程

示例: 02_basic_key

RT-Thread: 4.1.1

调试器: ST-LINK 接口: SWD

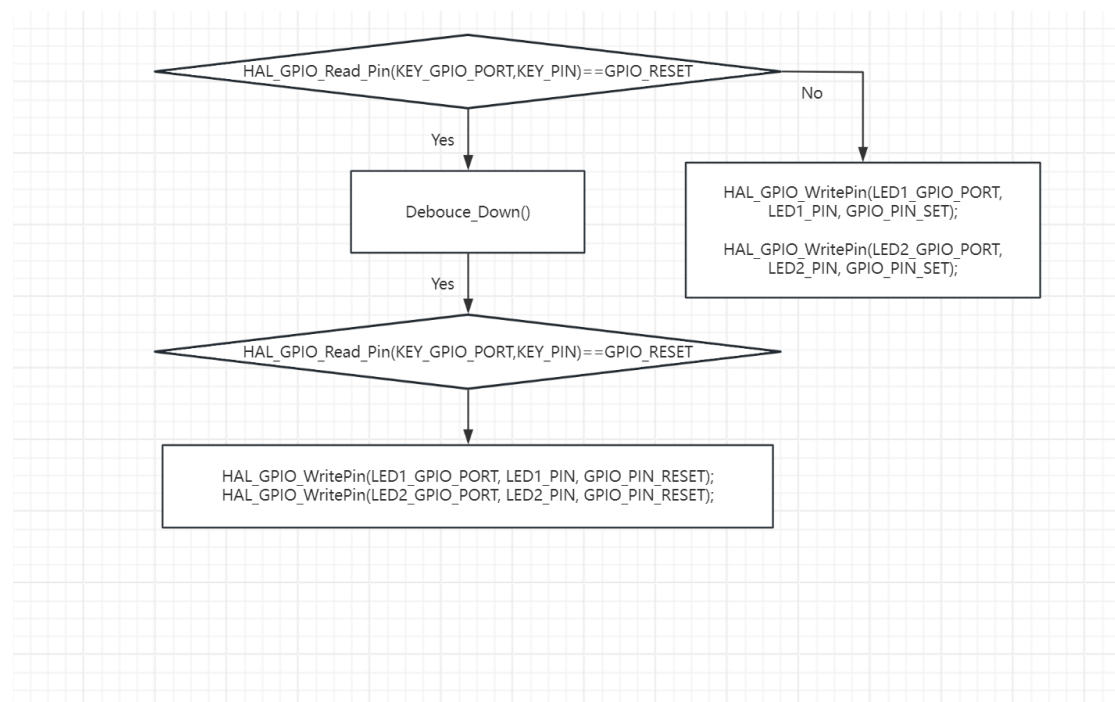
2.2 软件设计

1. 软件设计概述

软件设计主要采用的是 STM32CubeMX 进行代码初始化，再通过 μ vision 或者 RT-Tread Studio 进行代码文件的开发，完成了实验需求功能并添加了防抖功能，防抖功能为若触发“right”方向键小于 50ms 则不亮灯，反之亮灯。

软件设计概述

2. 软件流程图



3. μ vision 详细代码

由于代码过长，在此我只展示对于初始化代码的文件有变动的代码内容，均在 `main.c` 文件中进行修改。

(1) 宏定义相关端口号

```
#define LED1_PIN      GPIO_PIN_11
#define LED1_GPIO_PORT  GPIOF
#define LED2_PIN      GPIO_PIN_12
#define LED2_GPIO_PORT  GPIOF
#define KEY_PIN        GPIO_PIN_4
#define KEY_GPIO_PORT  GPIOC
```

(2) 添加防抖函数

```
void Debounce_Down(void)
{
    HAL_Delay(50); // 50ms debounce delay
}
```

(3) 修改主函数内容

```
while (1)
{
    // Check if button is pressed (active low)
    if (HAL_GPIO_ReadPin(KEY_GPIO_PORT, KEY_PIN) ==
GPIO_PIN_RESET)
    {
        Debounce_Down();
        if (HAL_GPIO_ReadPin(KEY_GPIO_PORT, KEY_PIN) ==
GPIO_PIN_RESET)
        {
            // Turn on both LEDs
            HAL_GPIO_WritePin(LED1_GPIO_PORT, LED1_PIN,
GPIO_PIN_RESET);
            HAL_GPIO_WritePin(LED2_GPIO_PORT, LED2_PIN,
GPIO_PIN_RESET);
        }
    }
    else
    {
        // Turn off LEDs
        HAL_GPIO_WritePin(LED1_GPIO_PORT, LED1_PIN,
GPIO_PIN_SET);
        HAL_GPIO_WritePin(LED2_GPIO_PORT, LED2_PIN,
GPIO_PIN_SET);
    }
}
```

```

        HAL_Delay(10); // Add small delay to prevent CPU lockup
    }

```

4. RT-Tread Studio 详细代码

```

/*
 * Copyright (c) 2006-2021, RT-Thread Development Team
 *
 * SPDX-License-Identifier: Apache-2.0
 *
 * Change Logs:
 *
 * Date           Author       Notes
 * 2023-5-10      ShiHao       first version
 */

#include <rtthread.h>
#include <rtdevice.h>
#include <board.h>

#define DBG_TAG "main"
#define DBG_LVL DBG_LOG
#include <rtdbg.h>

/* 配置 LED 灯引脚 */
#define PIN_LED_B          GET_PIN(F, 11)      // PF11 :  LED_B
--> LED

#define PIN_LED_R          GET_PIN(F, 12)      // PF12 :  LED_R
--> LED

/* 配置 KEY 输入引脚 */
#define PIN_KEY0           GET_PIN(C, 0)       // PC0:  KEY0
-->

```

```

KEY
#define PIN_KEY1          GET_PIN(C, 1)          // PC1 :  KEY1          -->
KEY
#define PIN_KEY2          GET_PIN(C, 4)          // PC4 :  KEY2          -->
KEY
#define PIN_WK_UP         GET_PIN(C, 5)          // PC5:  WK_UP         -->
KEY

```

```

int main(void)
{
    unsigned int count = 1;

    /* 设置 RGB 红灯引脚的模式为输出模式 */
    rt_pin_mode(PIN_LED_R, PIN_MODE_OUTPUT);
    rt_pin_mode(PIN_LED_B, PIN_MODE_OUTPUT);
    /* 设置 KEY0 引脚的模式为输入上拉模式 */
    rt_pin_mode(PIN_KEY2, PIN_MODE_INPUT_PULLUP);

    while (count > 0)
    {
        /* 读取按键 KEY0 的引脚状态 */
        if (rt_pin_read(PIN_KEY2) == PIN_LOW)
        {
            rt_thread_mdelay(50);
            if (rt_pin_read(PIN_KEY2) == PIN_LOW)
            {
                /* 按键已被按下，输出 log，点亮 LED 灯 */
                LOG_D("KEY0 pressed!");
                LOG_D("LED R&B HAVE BEEN ACTIVATED!");//灯被激活
                rt_pin_write(PIN_LED_R, PIN_LOW);

```



```

        rt_pin_write(PIN_LED_B, PIN_LOW);
    }
}
else
{
    /* 按键没被按下，熄灭 LED 灯 */
    rt_pin_write(PIN_LED_R, PIN_HIGH);
    rt_pin_write(PIN_LED_B, PIN_HIGH);
}
rt_thread_mdelay(10);
count++;
}
return 0;
}

```

3 实验结果分析与总结

如图所示，我成功完成了实验所需的要求内容，两个 LED 灯在按下按钮的时候一起亮起。

