

**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！

1. 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的内容。

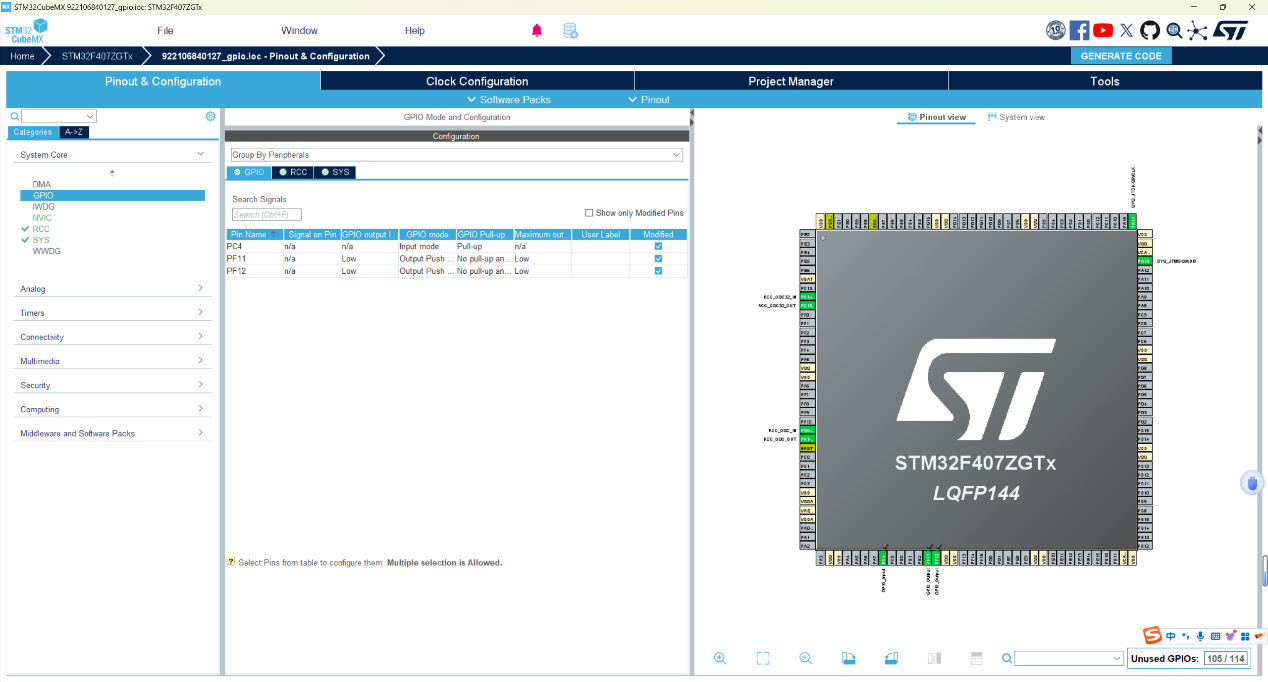
# 2 总体设计

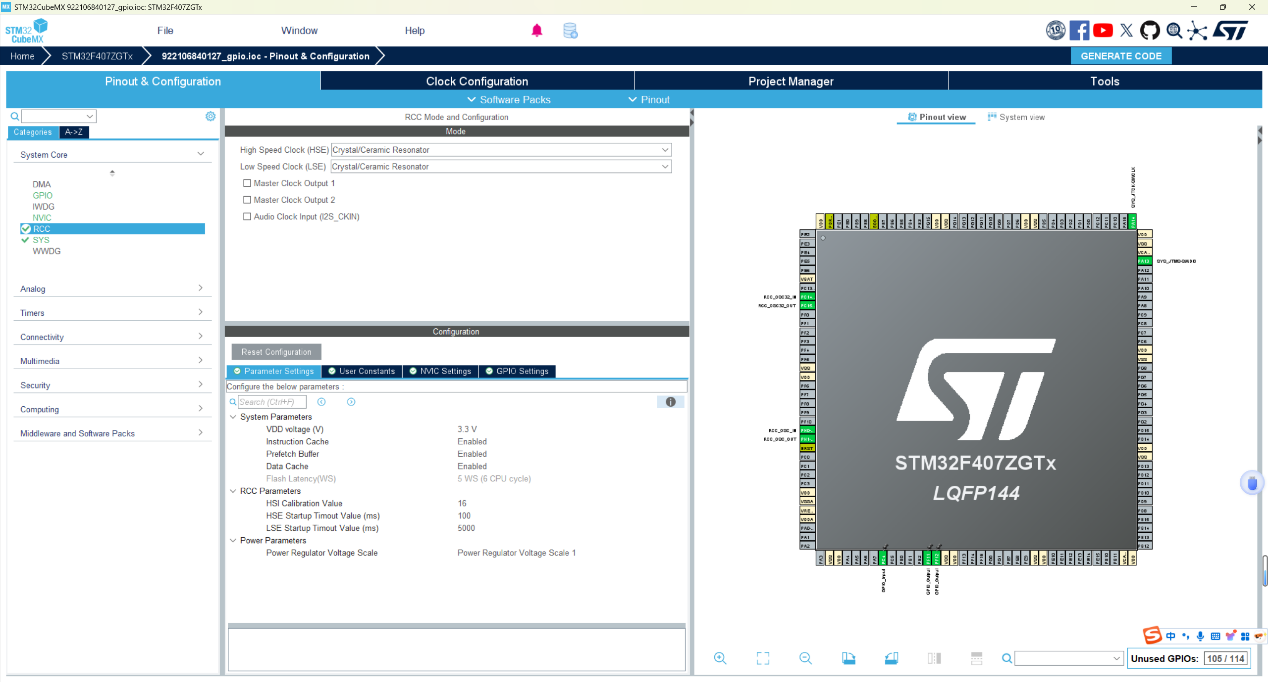
## 2.1硬件设计

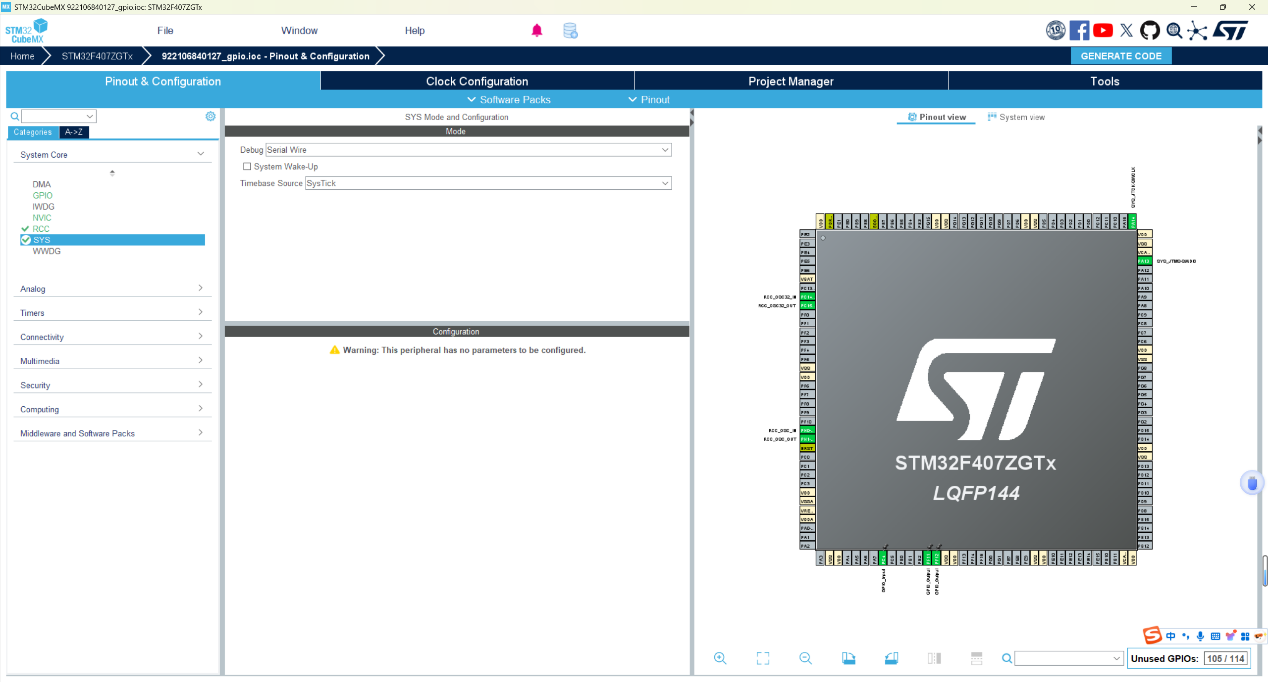
1. 硬件设计思路

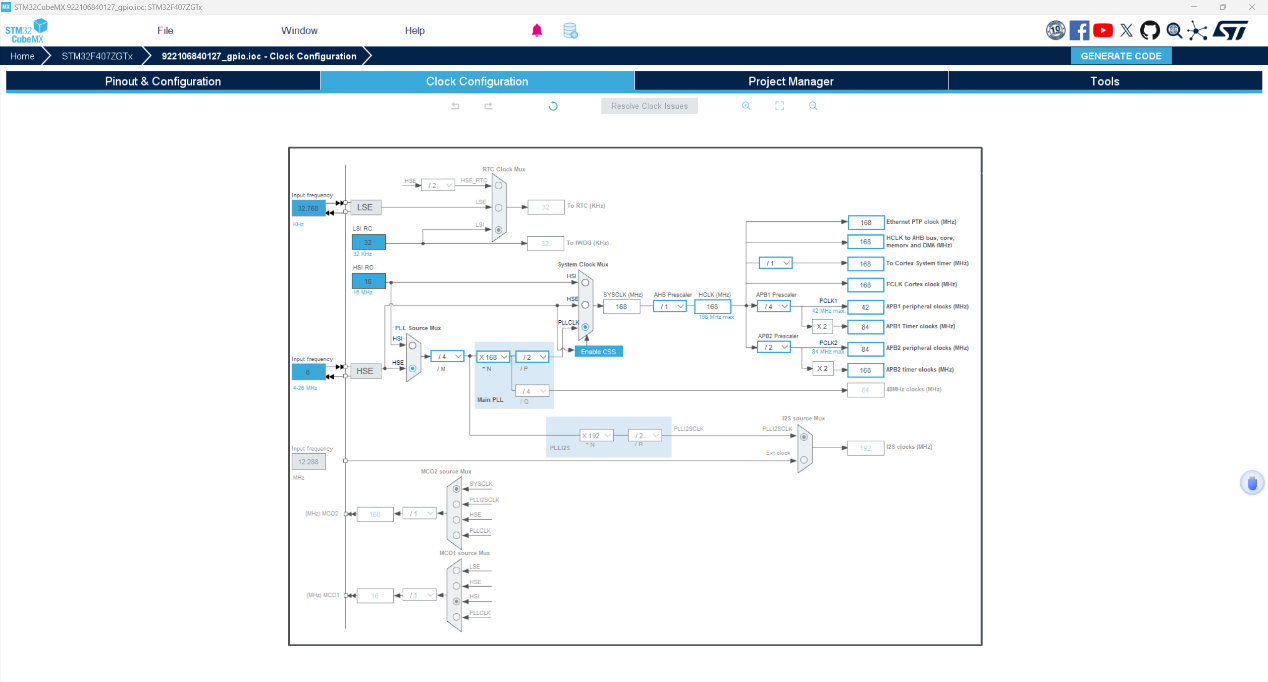
我通过查阅相关开发板原理图确定了两个LED灯和一个“right”按键的引脚值，采用的是PF11/PF12以及PC4。

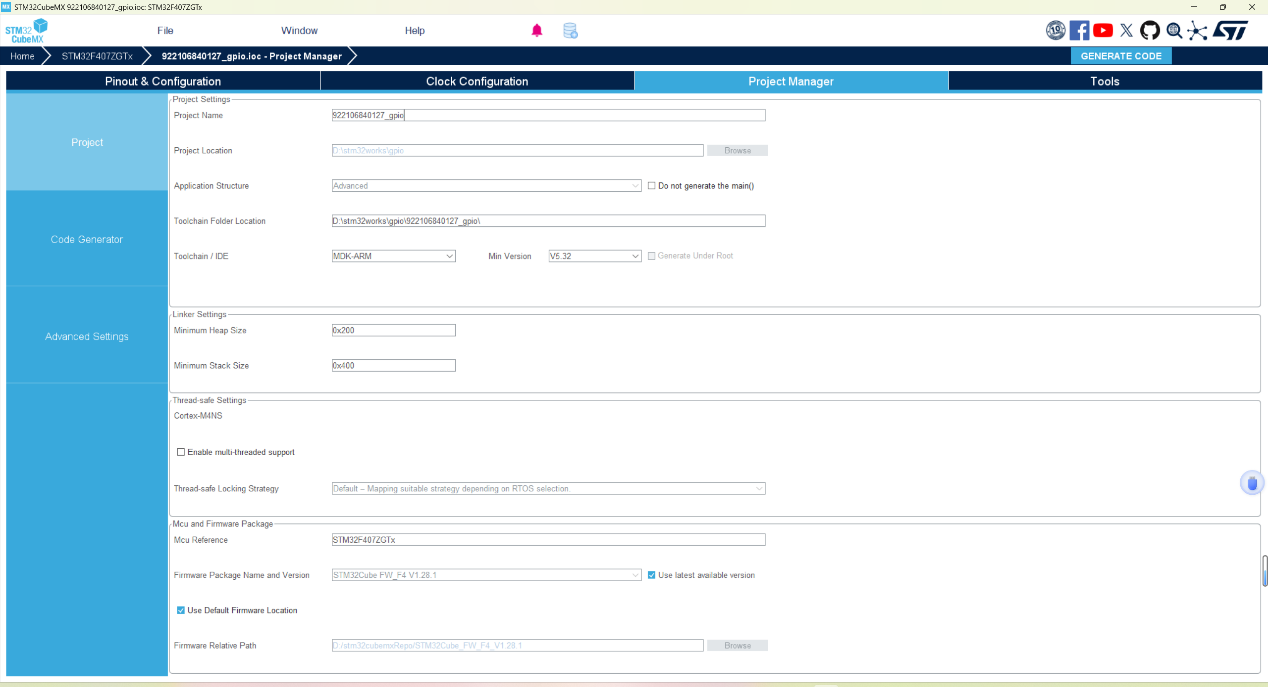
2. CubeMX图形化设计流程&关键参数设置图











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。

1. RT-Tread Studio的对应配置过程



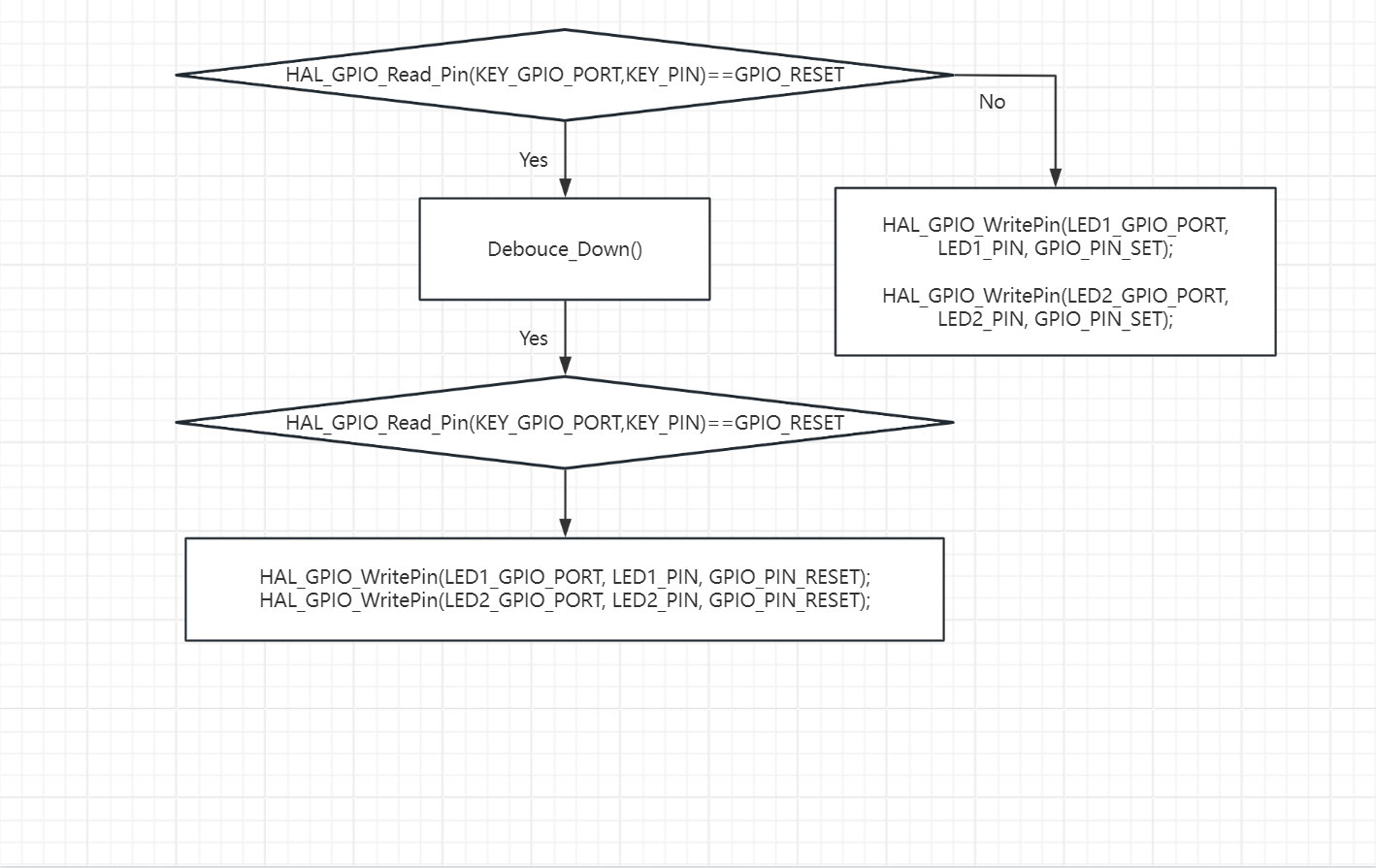
## 2.2 软件设计

**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μvision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. μvision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

1. 添加防抖函数

void Debounce\_Down(void)

{

HAL\_Delay(50); // 50ms debounce delay

}

1. 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。

