# Lab2 按键控制的LED

一 实验环境

二 软件实现

## 一 实验环境

下载 STM32F10x_StdPeriph_Lib_V3.6.0

手动搭建 STM32 工程项目

WIN10 19045

Keil MDK v5.3.9.0.0

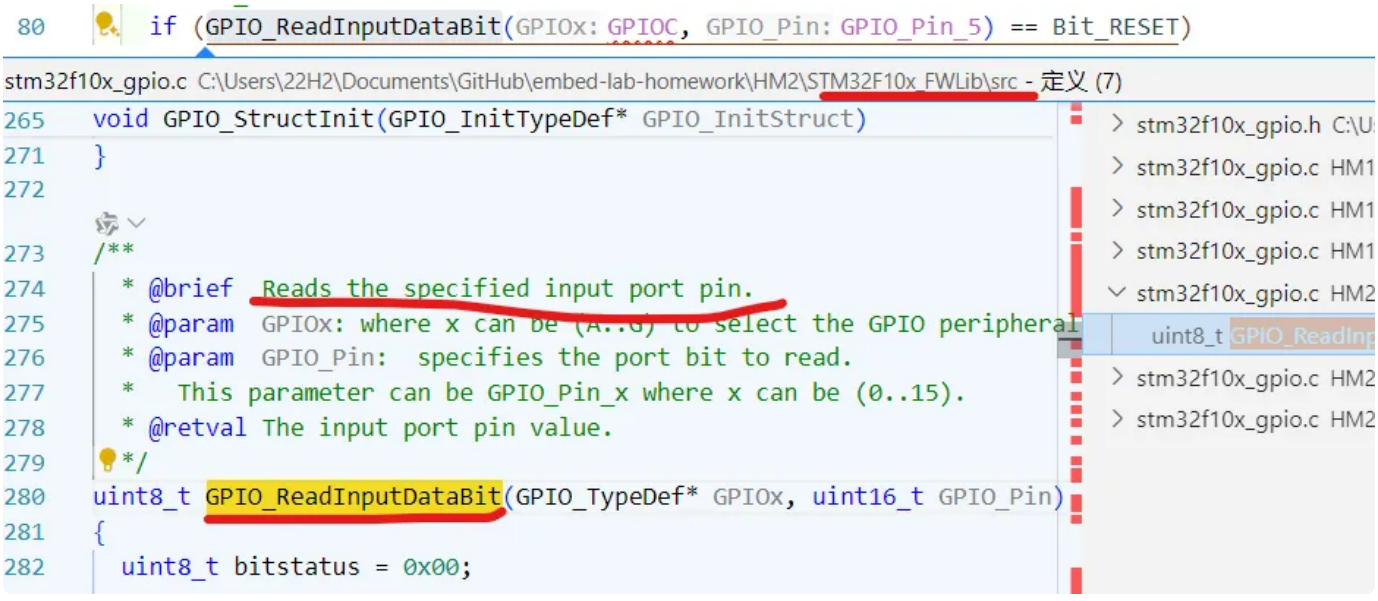
ARM Compiler v6.21

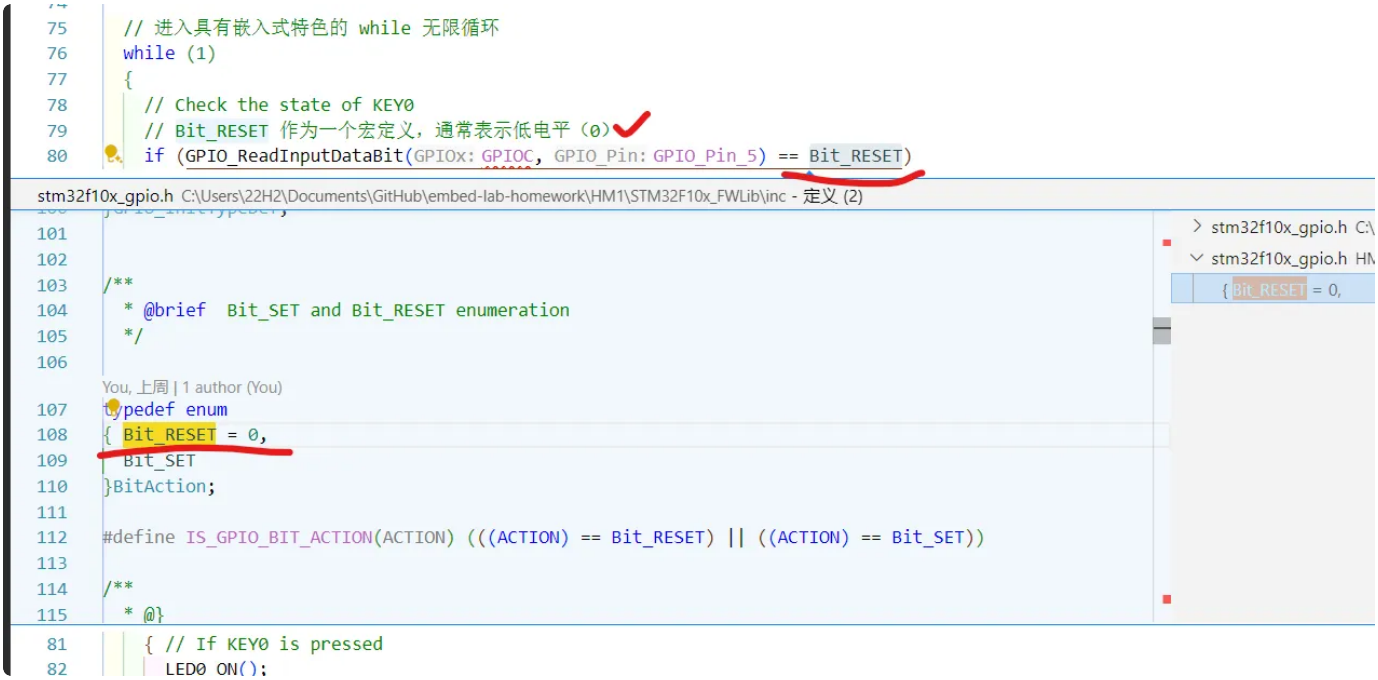
VSCode

## 二 软件实现

在作业 HM1 的基础上继续完善，实验按键 KEY0 来进行 LED 的闪烁控制。

会用到 `Bit_REST` 和之前用到的一个函数



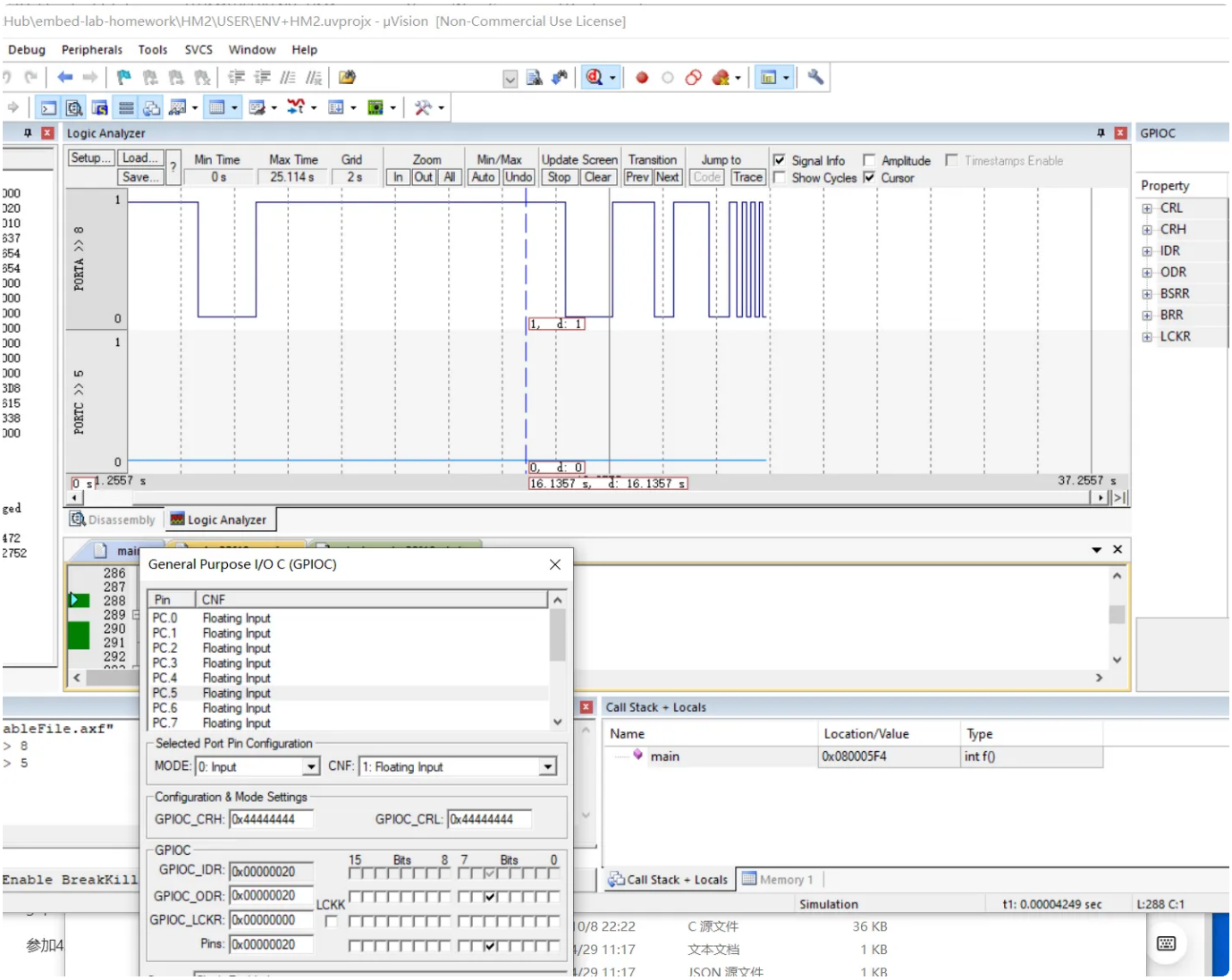


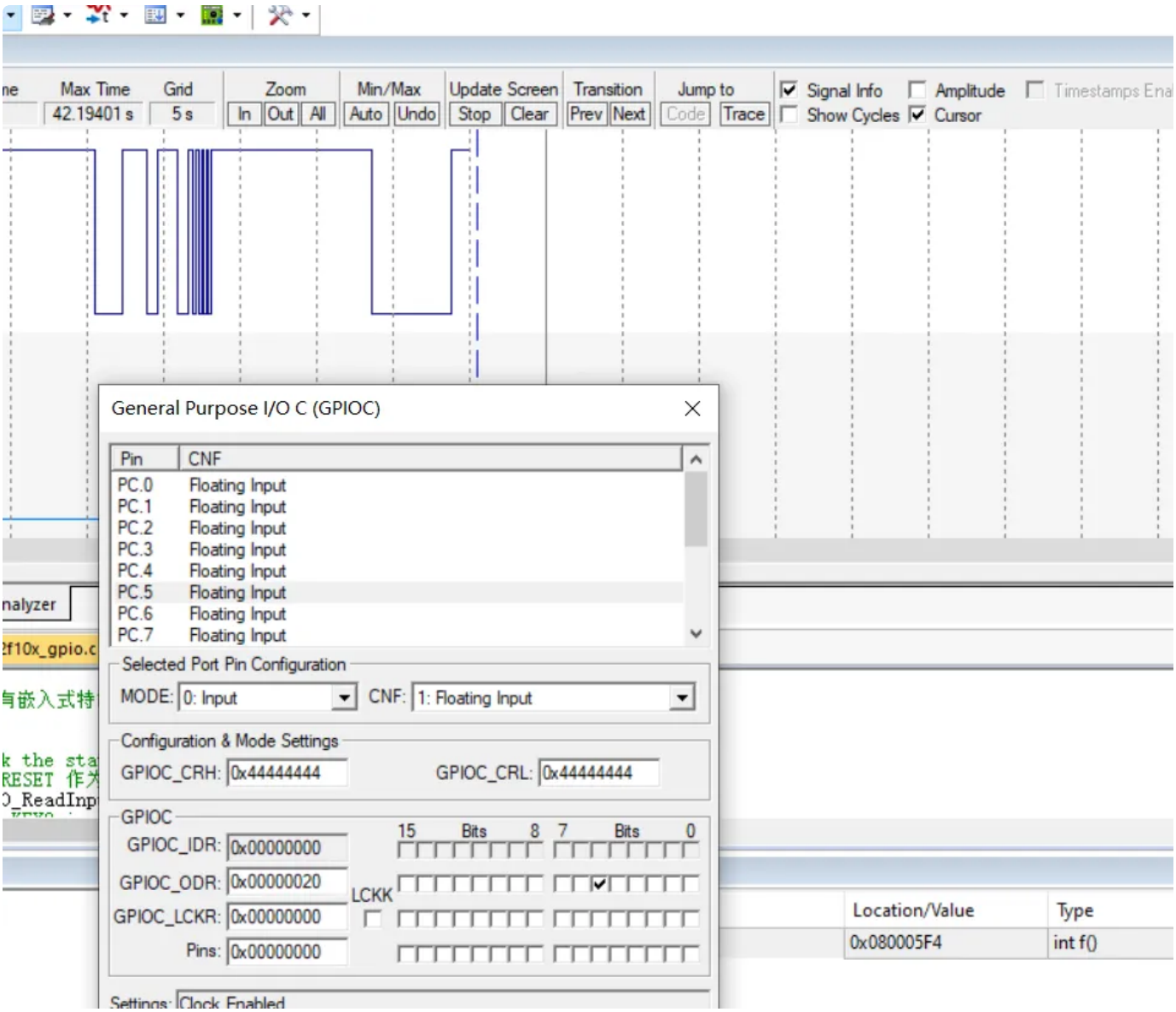

KEY0 的按键函数逻辑主要用到 `GPIO_Mode` 的上拉输入

KEY0 Function Imp

```c
// 根据要求 KEY0 读取PC5
void KEY0_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStructure;
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOC, ENABLE);
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;         // KEY0对应的引脚PC5
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU; // 输入上拉（上拉电阻）
  GPIO_Init(GPIOC, &GPIO_InitStructure);
}
```

模拟结果如下：


Hub\embed-lab-homework\HM2\USER\ENV+HM2.uvprojx - µVision [Non-Commercial Use License]
Debug Peripherals Tools SVCS Window Help
Logic Analyzer
GPIOC
Property
CRL
CRH
IDR
ODR
BSRR
BRR
LCKR
General Purpose I/O C (GPIOC)
Pin CNF
PC.0 Floating Input
PC.1 Floating Input
PC.2 Floating Input
PC.3 Floating Input
PC.4 Floating Input
PC.5 Floating Input
PC.6 Floating Input
PC.7 Floating Input
Selected Port Pin Configuration
MODE: 0: Input
CNF: 1: Floating Input
Configuration & Mode Settings
GPIOC_CRH: 0x44444444
GPIOC_CRL: 0x44444444
GPIOC
GPIOC_IDR: 0x00000020
GPIOC_ODR: 0x00000020
GPIOC_LCKR: 0x00000000
Pins: 0x00000020
LCKK
Call Stack + Locals
Name Location/Value Type
main 0x080005F4 int f()
Memory 1
Simulation
t1: 0.00004249 sec
L:288 C:1




勾选 PC5 模拟按键按下/释放

源代码如下：

main.c C

```c
/**
  ******************************************************************************
  * @file    Project/STM32F10x_StdPeriph_Template/main.c
  * @author  MCD Application Team
  * @version V3.6.0
  * @date    20-September-2021
  * @brief   Main program body
  ******************************************************************************
  * @attention
  *
  * Copyright (c) 2011 STMicroelectronics.
  * All rights reserved.
  *
  * This software is licensed under terms that can be found in the LICENSE file
  * in the root directory of this software component.
  * If no LICENSE file comes with this software, it is provided AS-IS.
  *
  ******************************************************************************
  */

/* Includes ------------------------------------------------------------------*/
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"

/* 使用STM32F10x的 按键KEY0 来控制LED的闪烁，在HM1的基础上继续完成功能-----------------------------------------------*/
void LED0_Init(void)
{
  // 定义一个GPIO_InitTypeDef类型的结构体 C:\Users\22H2\Documents\Boards\Keil5\HM1\STM32F10x_FWLib\inc\stm32f10x_gpio.h
  GPIO_InitTypeDef GPIO_InitStructure;
  // 使能GPIOA的时钟，打开APB2总线时钟 C:\Users\22H2\Documents\Boards\Keil5\HM1\STM32F10x_FWLib\inc\stm32f10x_rcc.h
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
  // 配置GPIOB的Pin8为推挽输出
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8;        // Specifies the GPIO pins to be configured.
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; // 推挽输出

```

```
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_2MHz; //  Specifies the speed for the selected pins.
  GPIO_Init(GPIOA, &GPIO_InitStructure);
}

void LED0_ON(void)
{
  GPIO_ResetBits(GPIOA, GPIO_Pin_8);
}

void LED0_OFF(void)
{
  GPIO_SetBits(GPIOA, GPIO_Pin_8); // GPIOx: A,B,C,D,E,F,G; GPIO_Pin: Specifies the port bit to be written.
}

// __IO 作为一个宏（修饰），来告诉编译器后面这个变量是一个输入/输出变量
void Delay(__IO u32 nCount)
{
  unsigned long i;
  for (i = 0; i < nCount; i++)
    ; // 空循环（消耗时间）
}

// 根据要求 KEY0 读取PC5
void KEY0_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStructure;
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOC, ENABLE);
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;        // KEY0对应的引脚PC5
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU; // 输入上拉（上拉电阻）
  GPIO_Init(GPIOC, &GPIO_InitStructure);
}

int main(void)
{
  // 初始化LED0
  // 初始化按键KEY0
  LED0_Init();
  KEY0_Init();

  // 进入具有嵌入式特色的 while 无限循环
  while (1)
  {
    // Check the state of KEY0
    // Bit_RESET 作为一个宏定义，通常表示低电平（0）
    if (GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_5) == Bit_RESET)
    { // If KEY0 is pressed
```

```
83          LED0_ON();
84        }
85        else
86        { // If KEY0 is released
87          LED0_OFF();
88        }
89      }
90    }
```