

# lab8 FreeRTOS的闪烁

---

- lab8 FreeRTOS的闪烁
  - 实验目的
  - 实验器材
    - 硬件
    - 软件
  - 实验原理
    - FreeRTOS
    - LED原理图连接方式
    - 连线
  - 实验步骤
    - 连线
    - Cube IDE上配置STM32F103C8
    - 编写使得PC13上的LED闪烁的程序
  - 实验心得

## 实验目的

- 在103上跑一个FreeRTOS的程序，实现板上LED的闪烁

## 实验器材

### 硬件

- STM32F103核心板1块;
- ST-Link1个;
- 杜邦线若干。

### 软件

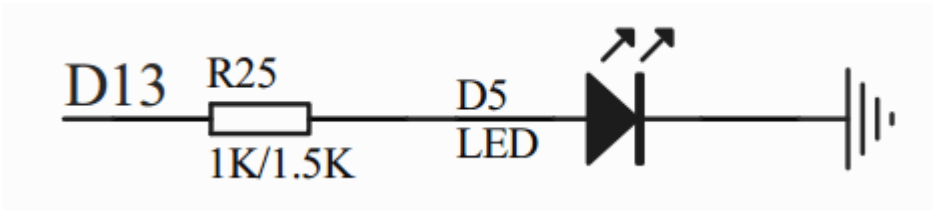
- STM32CubeIDE

## 实验原理

### FreeRTOS

- FreeRTOS是一个迷你的实时操作系统内核。作为一个轻量级的操作系统，功能包括：任务管理、时间管理、信号量、消息队列、内存管理、记录功能、软件定时器、协程等，可基本满足较小系统的需要。
- 由于RTOS需占用一定的系统资源(尤其是RAM资源)，只有 $\mu$ C/OS-II、embOS、salvo、FreeRTOS等少数实时操作系统能在小RAM单片机上运行。相对 $\mu$ C/OS-II、embOS等商业操作系统，FreeRTOS操作系统是完全免费的操作系统，具有源码公开、可移植、可裁减、调度策略灵活的特点，可以方便地移植到各种单片机上运行。

### LED原理图连接方式



- 只要给D13输出一个高电平就可以将LED点亮，低电平熄灭。

连线

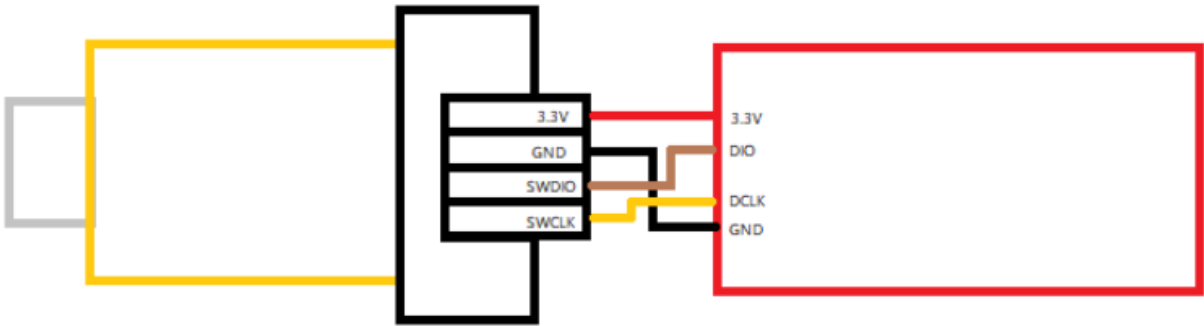
- 找出4条杜邦线，其中有一条是红色，一条是黑色，其余两条不同颜色。
- 连线的对应关系如下表

ST-Link	STM32F103	线的颜色	含义
(2)SWCLK	DCLK	橙色	时钟
(4)SWDIO	DIO	棕色	数据
(6)GND	GND	黑色	地
(8)3.3V	3.3V	红色	电源

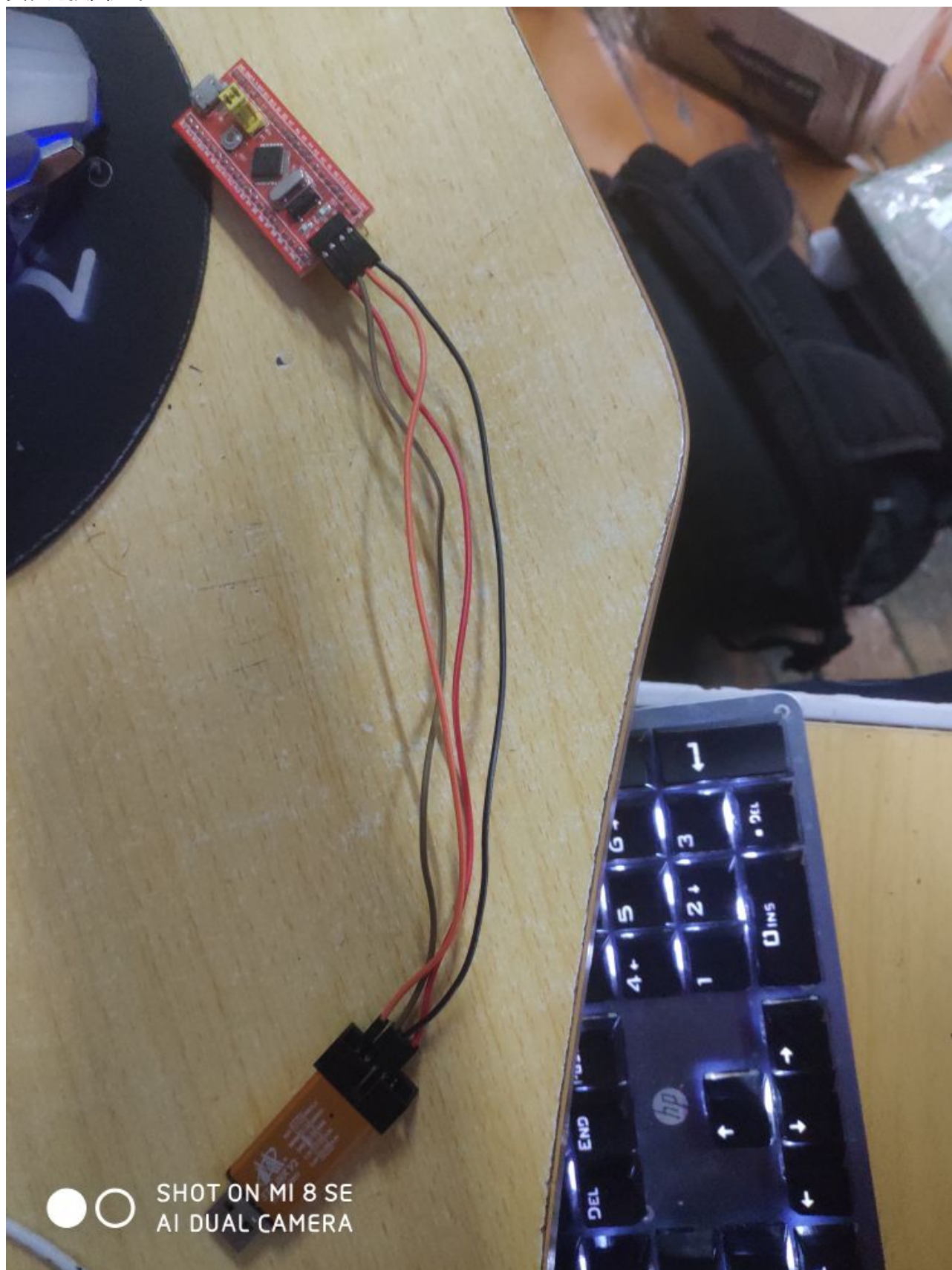
实验步骤

连线

- 根据实验原理中列出的连线对应关系表，将选取的杜邦线的一头连接103板子侧面的标着GND、DCLK、DIO、3.3V的四个插针，另一头接ST-Link的(2)SWCLK、(4)SWDIO、(6)GND、(8)3.3V。在ST-Link的插针处，有缺口的一侧是奇数，另一侧代表偶数。
- 连接示意图如下：



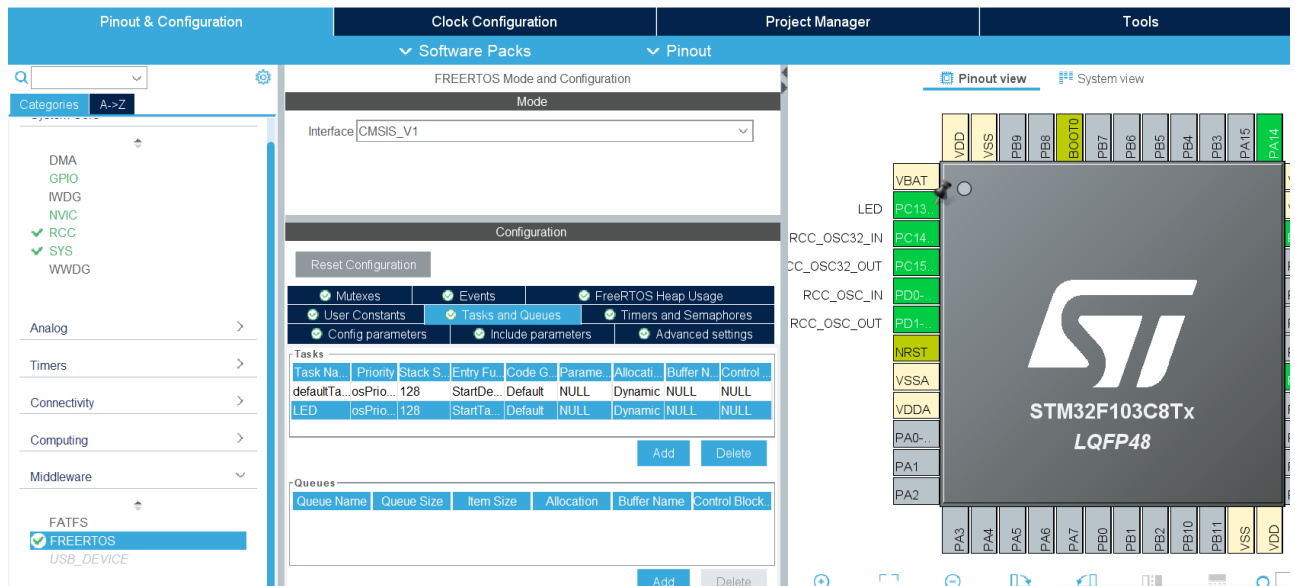
- 实物连接图如下：



Cube IDE上配置STM32F103C8

- 根据lab1的实验指南创建并配置工程。

- 配置FreeRTOS，如图



- 开启vTaskDelayUntil选项便可以使用更加精确的延时。

## 编写使得PC13上的LED闪烁的程序

```
/* freertos.c中寻找StartTask02()函数 */
void StartTask02(void const * argument)
{
    /* USER CODE BEGIN StartTask02 */
    /* Infinite loop */
    uint32_t PreviousWakeTime = osKernelSysTick();    // 更加精确的延时
    for(;;)
    {
        HAL_GPIO_TogglePin(LED_GPIO_Port, LED_Pin);
        osDelayUntil(&PreviousWakeTime, 200);    // 更加精确的延时
    }
    /* USER CODE END StartTask02 */
}
```

## 实验心得

- 本次实验通过最简单的LED灯闪烁，对FreeRTOS有一个较为初步的了解。由于大部分精力都集中在完成lab7的引导装载程序上，因此对FreeRTOS中的参数还没有进行较为详细的了解。不过作为一个轻量级的操作系统，还是很期待其能够为开发者带来哪些相较于裸机更加强大的功能。