

实验二 LED 闪烁

潘盛琪 3170105737

实验目的：

1. 了解嵌入式板卡一般情况；
2. 熟悉嵌入式板卡的供电、下载和调试的接线方式；
3. 安装交叉编译环境，理解交叉编译；
4. 熟练掌握编译、下载、运行的过程。

实验器材：

硬件：

1. STM32F103 核心板 1 块；
2. ST-Link 线 1 根；
3. 杜邦线（孔-孔）4 根

软件：

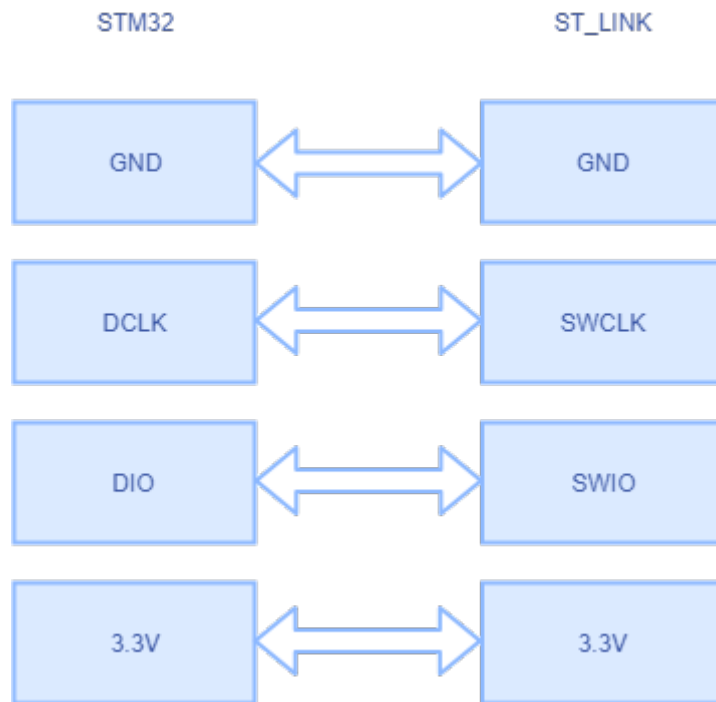
在 st.com 下载安装 Cube IDE（包含了 Eclipse/CDT、GCC 交叉编译程序和 ST-Link 驱动）。

实验步骤：

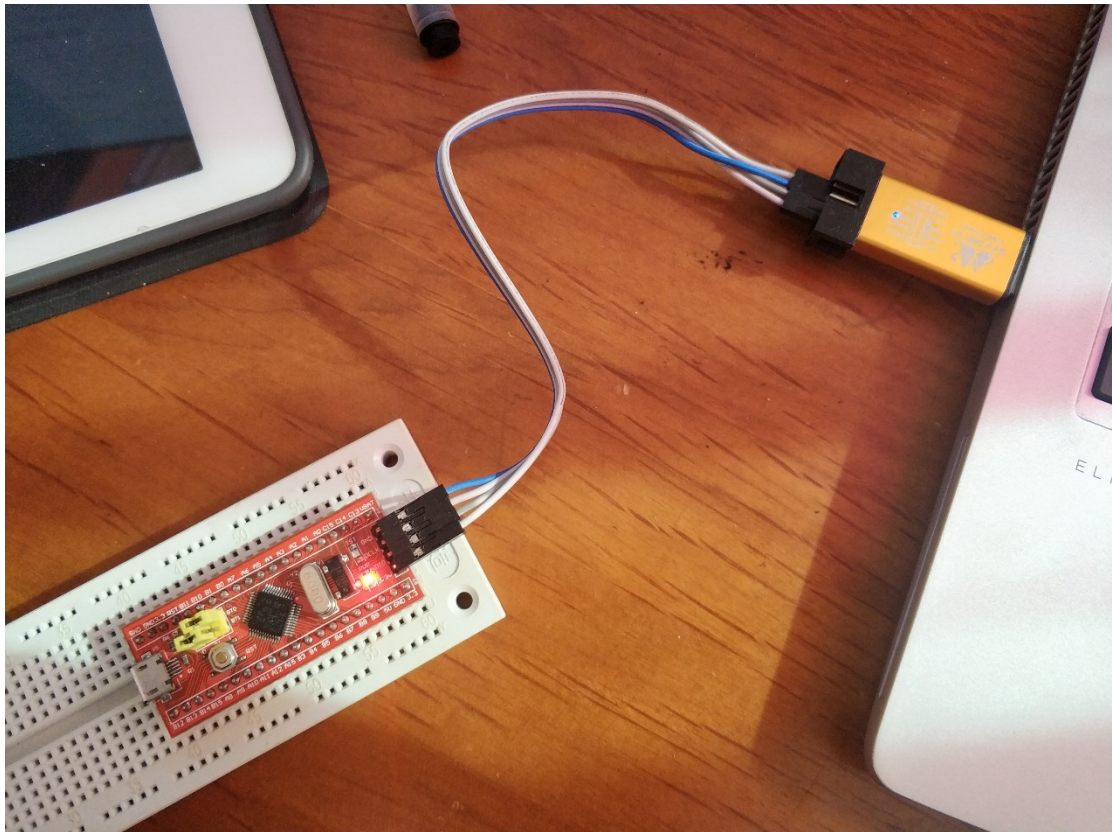
1. 在 PC 上安装好 Cube IDE；
2. Cube IDE 上配置 STM32F103C8；
3. 编写使得 PC13 上的 LED 闪烁的程序；
4. 编译产生可下载/运行的二进制映像文件；
5. 将 103 核心板安装在面包板上，连接板卡和 PC，即连接 ST-Link 和核心板（GND、DCLK、DIO 和 3.3V 四线）；
6. 给板卡上电，下载程序到板卡，观察 LED 是否闪烁。

实验结果：

1. 实际实施的连接示意图



2. 实际摆设的板卡连接照片



3. 程序配置编译和下载的过程

新建项目，选择 stm32f103c8

新建完成后将 pc13 设为输出并命名为 LED

PC13-TAMPER-RTC Configuration :

GPIO output level	Low
GPIO mode	Output Push Pull
GPIO Pull-up/Pull-down	No pull-up and no pull-down
Maximum output speed	Low
User Label	LED

随后会自动生成 STM32 基本配置的代码

要使 LED 闪烁，也就是隔一段时间让 LED 驱动电平翻转一次；其中电平翻转的功能可以由 HAL_GPIO_TogglePin() 函数来实现，而定时功能则有如下两种实现方法

(1) 利用 HAL_Delay()

第一种就是利用 hal 中的函数 HAL_Delay() 来实现，这种方法的缺点在于：需要占用 CPU 时间，也就是说在延迟过程中不能完成其他任务；此外利这种方法进行延时，时间也不精确。代码如下：

```

95  while (1)
96  {
97      /* USER CODE END WHILE */
98      /* USER CODE BEGIN 3 */
99      HAL_GPIO_TogglePin(LED_GPIO_Port,LED_Pin); //Toggle LED
100     HAL_Delay(1000); //Delay 1 Seconds
101 }

```

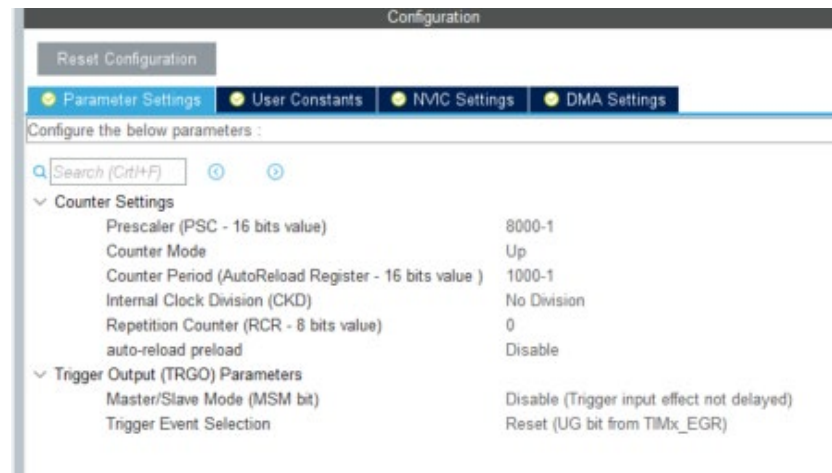
(2) 利用定时器

第二种方法就是利用定时器，配置界面如下图：在 Timers 中选择 Timer 中选择 TIM1，Clock Source 选择内部时钟在下方 NVIC Settings 中选择 update interrupt 也就是溢出中断。

The screenshot shows the STM32CubeMX configuration interface. On the left, the 'Timers' category is expanded, and 'TIM1' is selected. The main panel displays the 'TIM1 Mode and Configuration' settings. Under the 'Mode' section, 'Slave Mode' is set to 'Disable', 'Trigger Source' is 'Disable', and 'Clock Source' is 'Internal Clock'. The 'Configuration' section at the bottom shows the 'NVIC Interrupt Table' with the 'TIM1 update interrupt' checked under the 'Enabled' column.

Configuration			
Reset Configuration			
Parameter Settings ● User Constants ● NVIC Settings ● DMA Settings			
NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
TIM1 break interrupt	<input type="checkbox"/>	0	0
TIM1 update interrupt	<input checked="" type="checkbox"/>	0	0
TIM1 trigger and commutation interrupts	<input type="checkbox"/>	0	0
TIM1 capture compare interrupt	<input type="checkbox"/>	0	0

定时时间配置如下图，由于 TIM1 挂在 APB2 总线上，查看时钟数可以看到当前频率 8MHz，若希望每隔 1 秒进入一次中断，需要将预分频设为 8000-1，自动重载值设为 1000-1。



配置好始终后生成初始代码，随后中断响应程序的编写

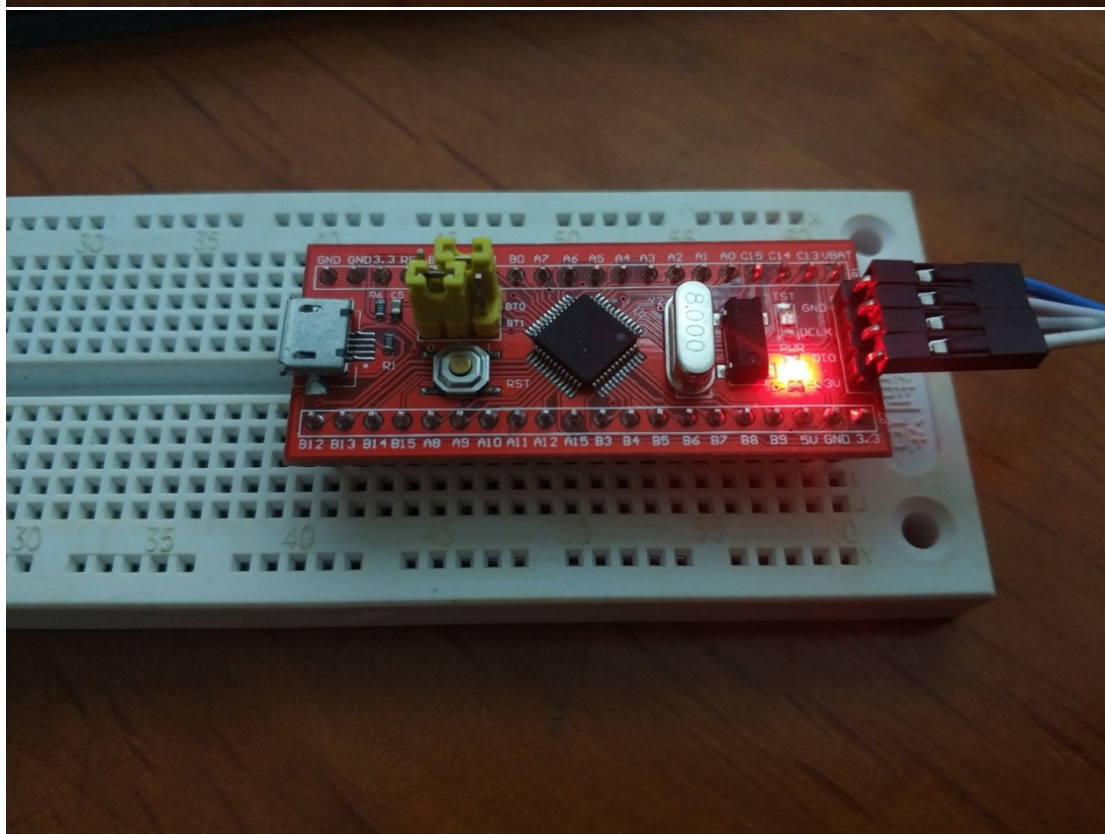
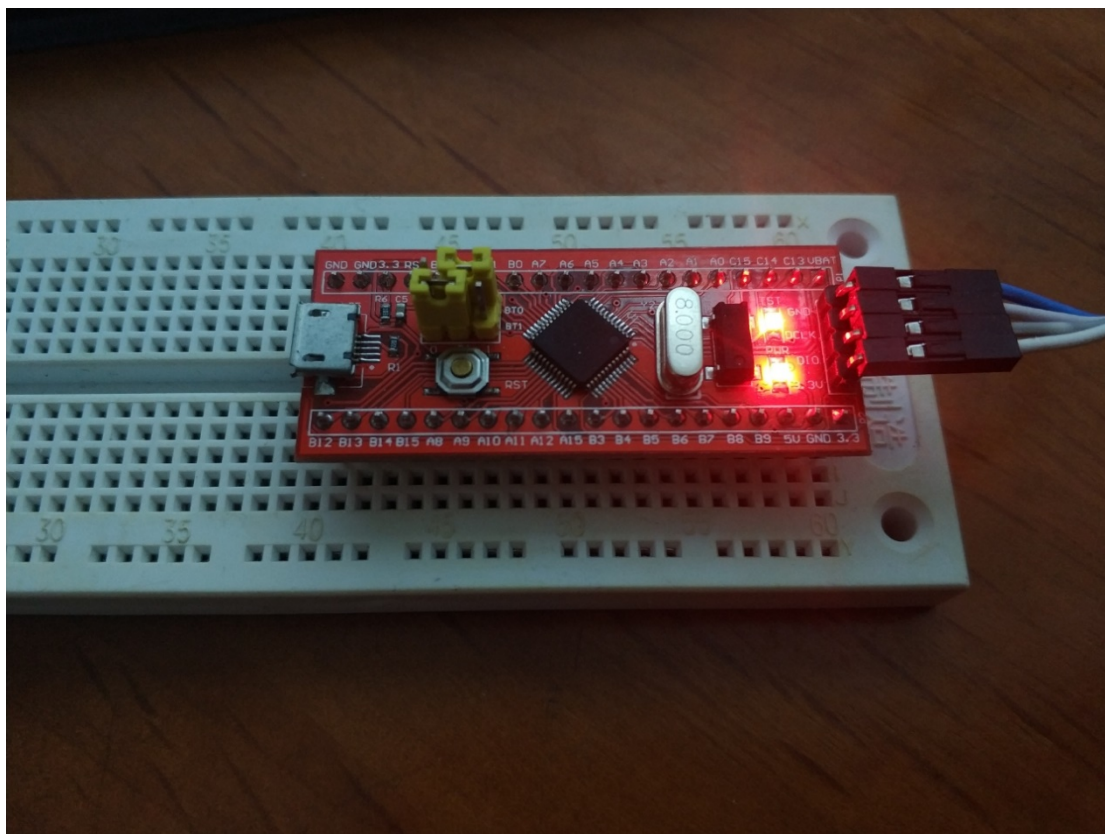
```
203 void TIM1_UP_IRQHandler(void)
204 {
205     /* USER CODE BEGIN TIM1_UP_IRQn 0 */
206
207     /* USER CODE END TIM1_UP_IRQn 0 */
208     HAL_TIM_IRQHandler(&htim1);
209     /* USER CODE BEGIN TIM1_UP_IRQn 1 */
210     HAL_GPIO_TogglePin(LED_GPIO_Port,LED_Pin); //Toggle LED
211     /* USER CODE END TIM1_UP_IRQn 1 */
212 }
---
```

如下图，在程序开始前启动定时器即可

```
90  MX_GPIO_Init();
91  MX_TIM1_Init();
92  /* USER CODE BEGIN 2 */
93  HAL_TIM_Base_Start_IT(&htim1);
94  /* USER CODE END 2 */
95
96  /* Infinite loop */
97  /* USER CODE BEGIN WHILE */
98  while (1)
99  {
100     /* USER CODE END WHILE */
101
102     /* USER CODE BEGIN 3 */
103 }
104 /* USER CODE END 3 */
105 }
---
```

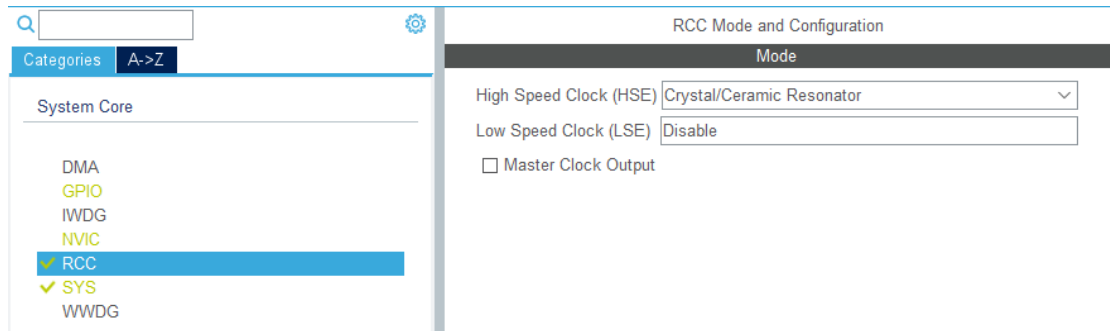
可以看到在主循环 while(1)中是没有任何代码的

4. LED 闪烁、亮灭的照片

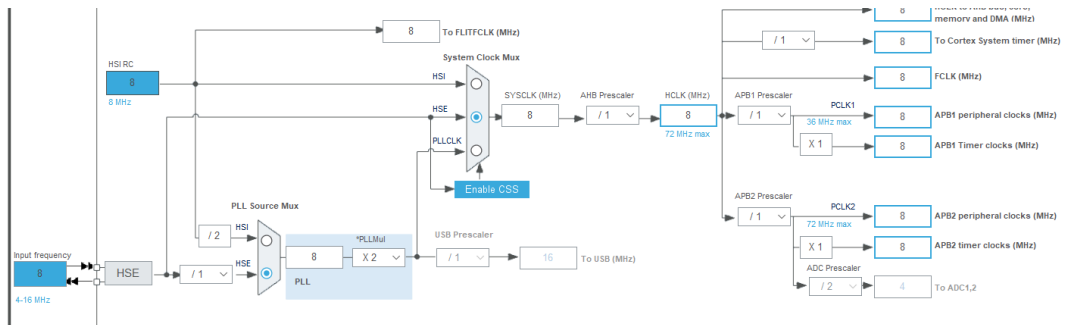


5. 拓展内容

如图，将高速时钟设定为外部晶振



随后在时钟树的界面，将原本用到的内部 HSI 时钟源都切换为 HSE，即外部高速时钟



定时器的配置以及代码与之前相同，可以发现 LED 正常闪烁。