

实验三 LED 闪烁

潘盛琪 3170105737

实验目的：

1. 熟练掌握串口在 PC 和 MCU 上的使用；熟悉嵌入式板卡的供电、下载和调试的接线方式；
2. 掌握 MCU 的中断响应程序编写方法；
3. 熟练掌握 MCU 的时钟中断配置和时钟中断程序响应程序的编写方法；
4. 掌握 MCU 程序的定时和获得时间的方法（采用 SysTick 和定时器中断两种方法）。

实验器材：

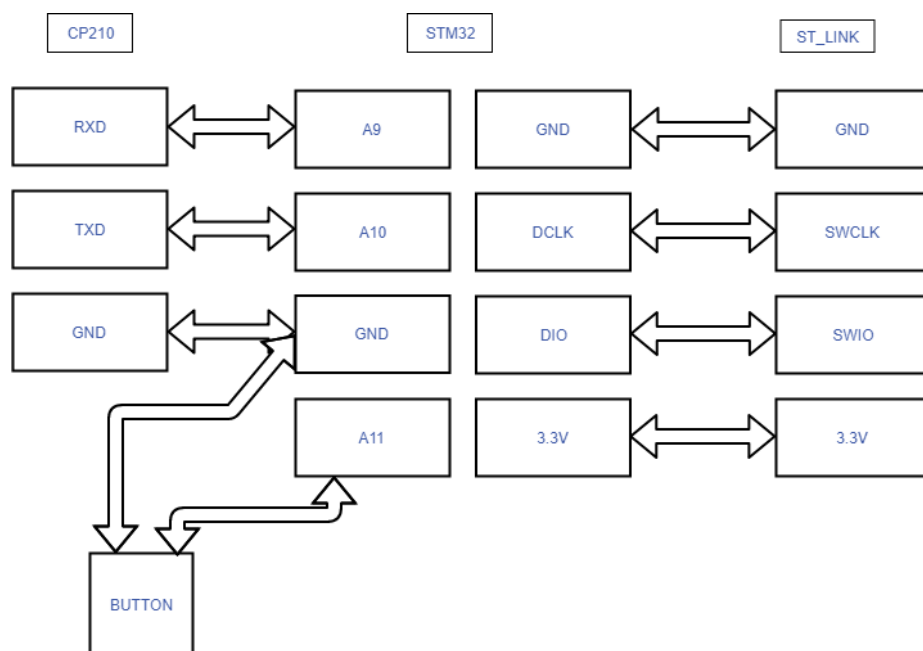
硬件：

1. STM32F103 核心板 1 块；
2. ST-Link 线 1 根；
3. USB-TTL 串口线 1 根（CP2102 芯片）；
4. 杜邦线若干；
5. 按钮；
6. 面包板。

软件：

1. PC 上的 CubeIDE；
2. PC 上的 USB-TTL 串口线配套的驱动程序；
3. PC 上的串口调试助手。

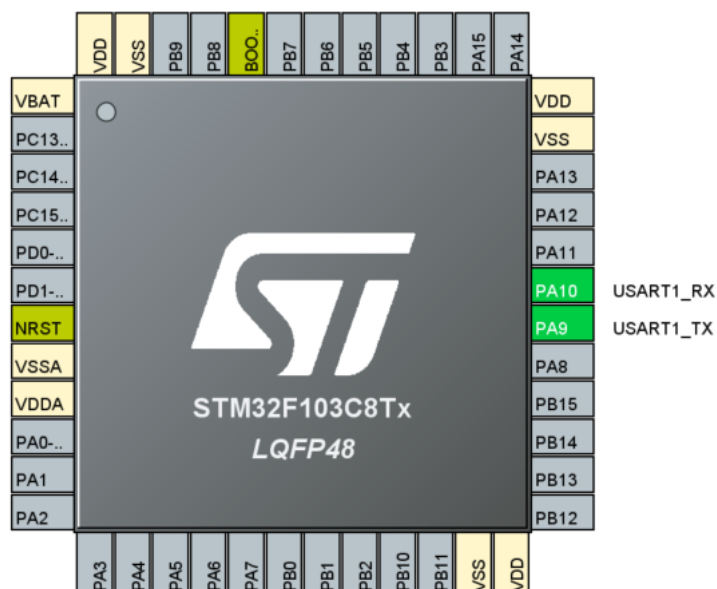
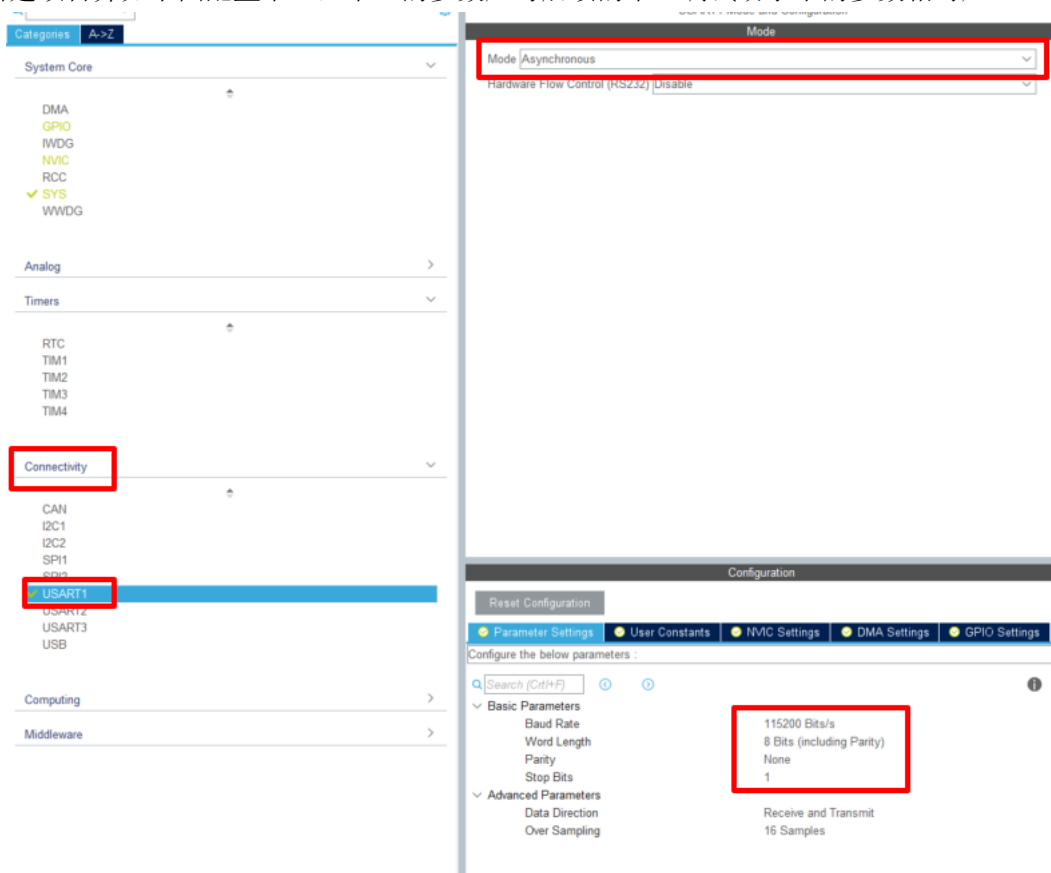
实验接线示意图如下：



实验过程与结果：

1. 在串口上输出 **Hello World**

新建项目并如下图配置串口，串口的参数应与后续的串口调试助手中的参数相对应：



在 `stm32f1xx_hal_uart.c` 中可以找到串口发送函数的原型：

```
HAL_StatusTypeDef HAL_UART_Transmit(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size, uint32_t Timeout)
```

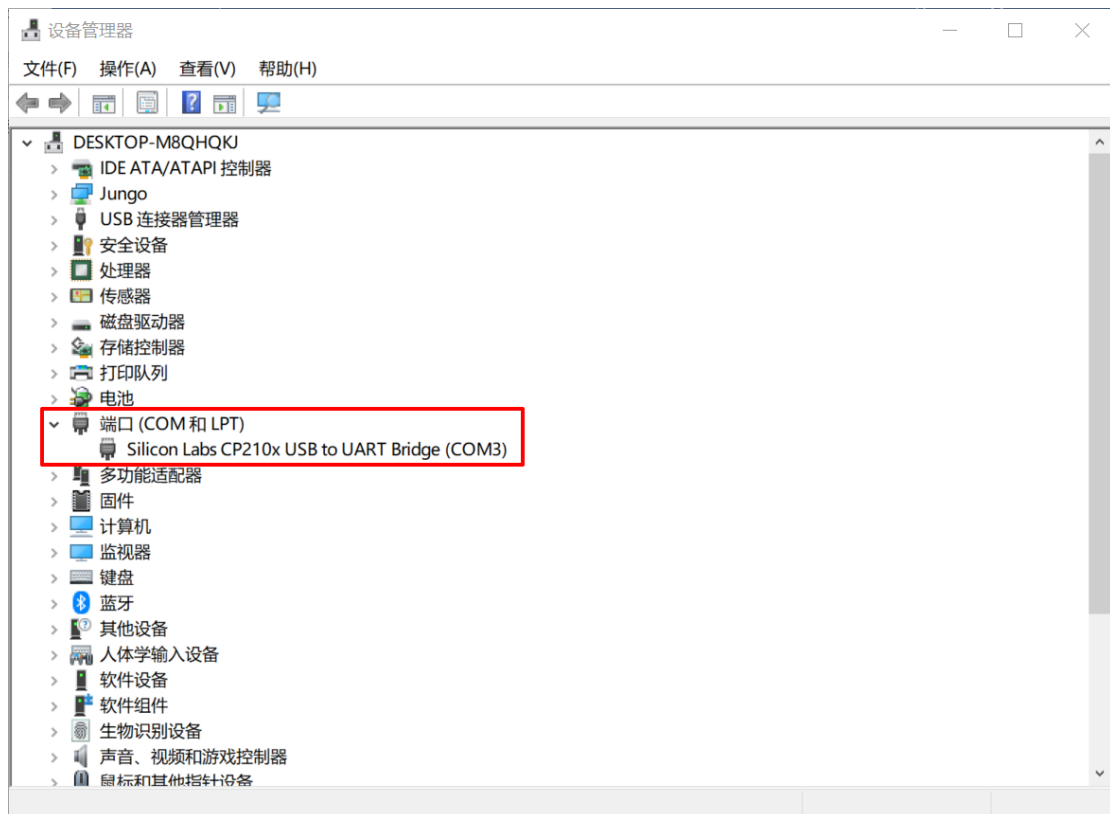
在 `main.c` 中添加如下代码：

```

92  /* USER CODE BEGIN 2 */
93  uint8_t msg[] = "Hello World!\n";
94  /* USER CODE END 2 */
95
96  /* Infinite loop */
97  /* USER CODE BEGIN WHILE */
98  while (1)
99  {
100     /* USER CODE END WHILE */
101     HAL_UART_Transmit(&huart1, (uint8_t *)&msg, sizeof(msg),0xFFFF);
102     HAL_Delay(1000);
103     /* USER CODE BEGIN 3 */
104 }
105 /* USER CODE END 3 */

```

将 CP210 连接至 PC，在设备管理器中可以找到其端口号：

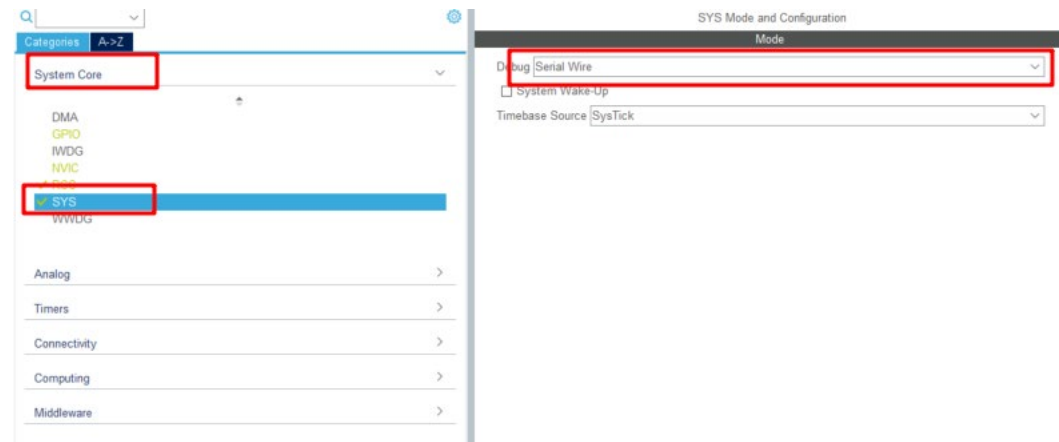


随后打开串口调试助手，调节好端口波特率等参数，再编译运行程序，即可看到在串口调试助手的界面输出 Hello World!

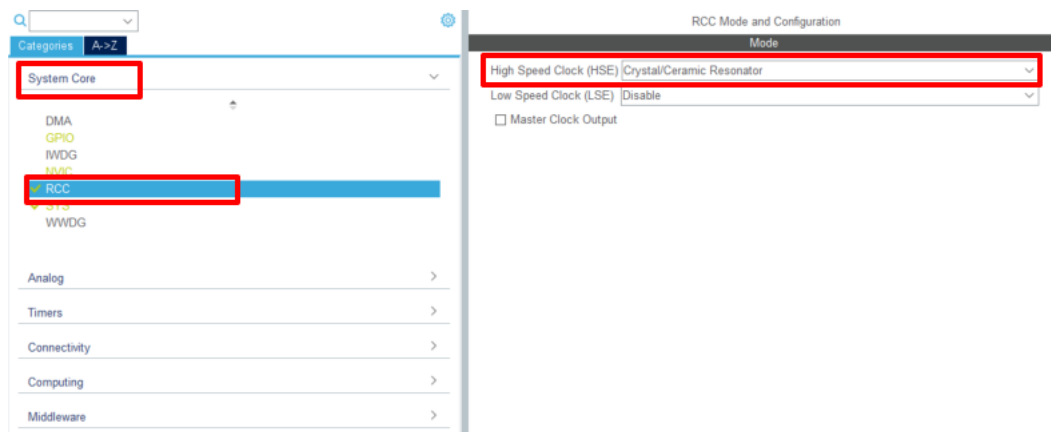


2. 连接按钮进行控制

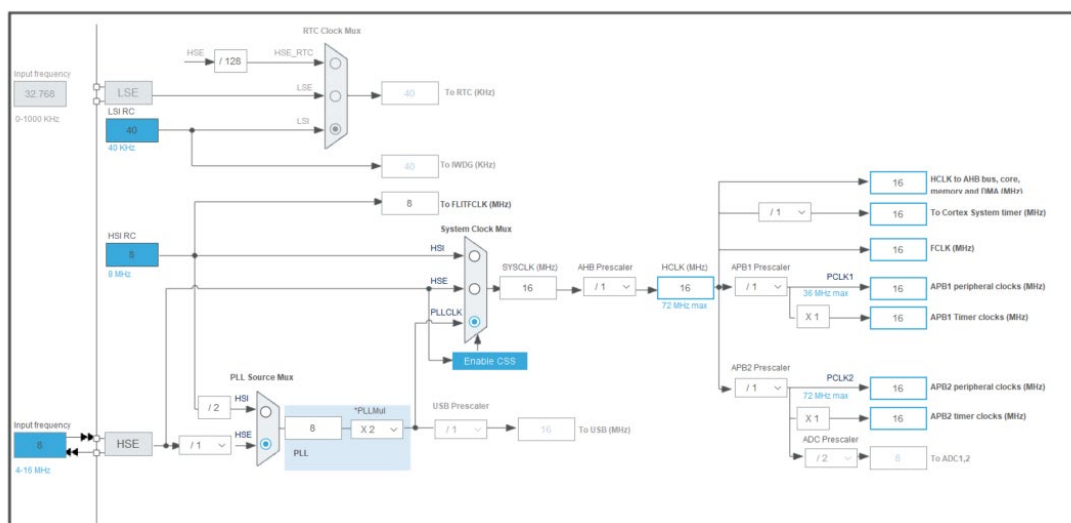
因为这个程序相比前面的稍微有些复杂了，需要打开 debug 功能，如下图操作即可：



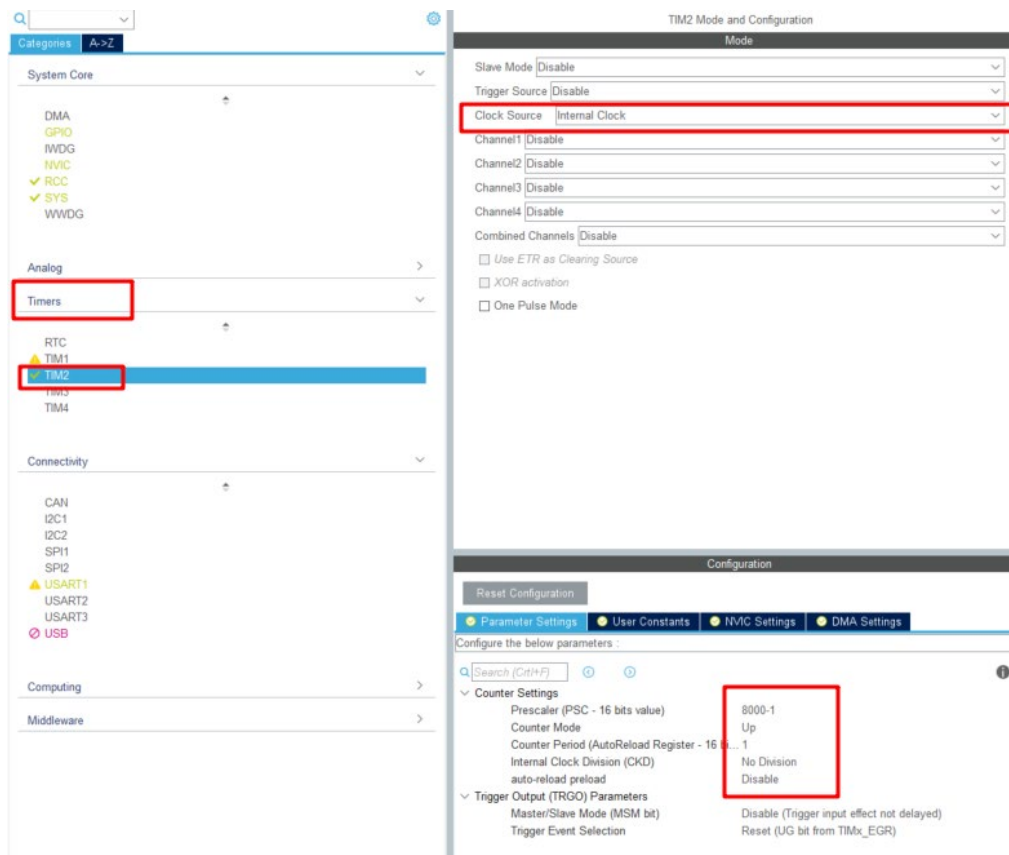
首先配置 LED 的 GPIO 和按钮的 GPIO，需要注意的是按钮的 GPIO 需要设置上拉电阻。RCC 设置如下：



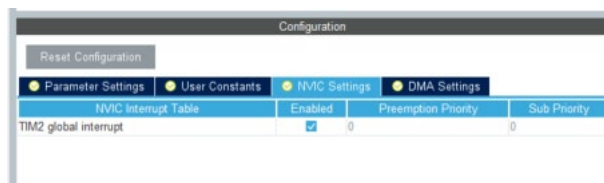
如下图配置时钟，最终 APB2 总线的频率为 16MHz：



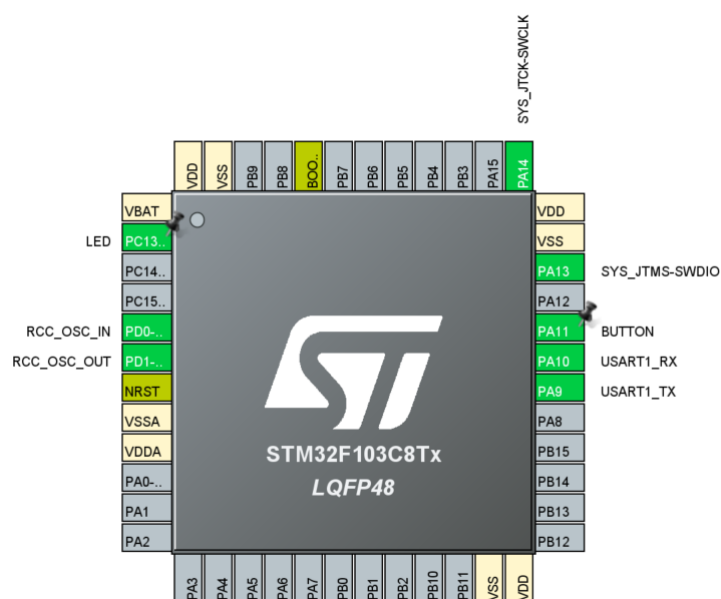
配置计时器参数，16MHz 经过 8000-1 分频后周期为 0.5ms，设置 Period 为 1 则每 1ms 溢出一次：



还需要再 NVIC 中开启定时器中断如下图：



最终配置完成的引脚情况如下图：



串口的配置与前述 HelloWorld 相同，完成后生成代码进行程序编写。

➤ 按键去抖逻辑：

在正常按下按键时，仅会产生一个上升沿或下降沿，相应的按键响应事件也只执行一次。但是由于机械按键的结构原因，通常会产生抖动导致在一次动作中产生多个上升沿或下降沿，从而导致按键响应事件执行多次。

要消除按键抖动带来的影响，我的做法是加一个用于计数的变量 cnt，每当检测到按键按下，cnt 累加，cnt 累加到一定值说明按键真正被按下了，这时候再执行按键响应事件。

主循环代码如下：

```
while (1)
{
    /* USER CODE END WHILE */

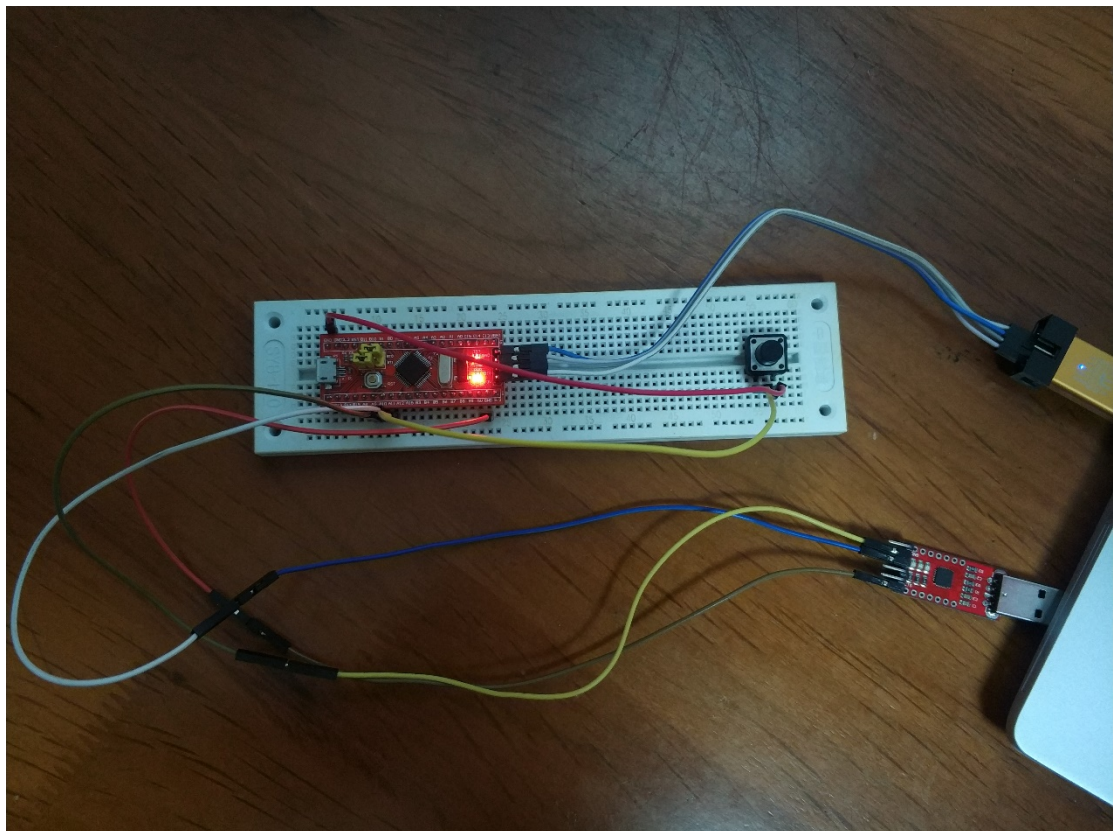
    /* USER CODE BEGIN 3 */
    uint8_t outBuffer[15] = {};
    int i = 0;
    int tmpTime = currentTime;
    if (!BUTTONState)//按键松开
    {
        if(HAL_GPIO_ReadPin(BUTTON_GPIO_Port,BUTTON_Pin) == GPIO_PIN_SET)//去抖
        {
            cnt++;
        }
        if(cnt > 200)
        {
            HAL_TIM_Base_Stop_IT(&htim2);
            cnt = 0;
            BUTTONState = BUTTONState ^ 1;
            while(tmpTime)
            {
                tmpTime /= 10;
                i++;
            }
            while(currentTime)
            {
                outBuffer[i] = currentTime % 10 + 48;
                currentTime /= 10;
                i--;
            }
            HAL_UART_Transmit(&huart1, (uint8_t *)&outBuffer, sizeof(outBuffer),0xFFFF);
            HAL_UART_Transmit(&huart1, (uint8_t *)&outBufferPost, sizeof(outBufferPost),0xFFFF);
            currentTime = 0;
        }
    }
    else//按键按下
    {
        if(HAL_GPIO_ReadPin(BUTTON_GPIO_Port,BUTTON_Pin) == GPIO_PIN_RESET)//去抖
        {
            cnt++;
        }
        if(cnt > 200)
        {
            HAL_TIM_Base_Start_IT(&htim2);
            cnt = 0;
            BUTTONState = BUTTONState ^ 1;
            LEDState = LEDState ^ 1;
            HAL_GPIO_TogglePin(LED_GPIO_Port,LED_Pin);
            if(LEDState)
                HAL_UART_Transmit(&huart1, (uint8_t *)&msgOn, sizeof(msgOn),0xFFFF);
            else
                HAL_UART_Transmit(&huart1, (uint8_t *)&msgOff, sizeof(msgOff),0xFFFF);
        }
    }
}
```

同时添加 timer2 回调函数，用于计时。

编译运行程序并在串口调试助手查看输出结果：

串口调试功能		显示串口接收到的数据
端口	<input type="text" value="COM1"/>	LED off
波特率	<input type="text" value="115200"/>	171ms
校验位	<input type="text" value="None"/>	LED on
数据位	<input type="text" value="8"/>	137ms
停止位	<input type="text" value="1"/>	LED off
<input type="radio"/>	<input type="button" value="打开串口"/>	955ms
<input type="button" value="清空接收区"/>	<input type="button" value="停止显示"/>	LED on
<input type="checkbox"/> 自动清空	<input type="button" value="接收转向文件"/>	1694ms
<input type="checkbox"/> 十六进制显示		LED off
		100ms
		LED on
		140ms
		LED off
		125ms
		LED on
		132ms
		LED off
		111ms
		LED on
		107ms

实验实际接线图如下：



拓展内容：