



实验报告

课程名称: 机电一体化

课程编号: ME333

实验题目: 定时器中断实验

学号: 12313215

姓名: 梁家源

专业: 机器人工程

指导教师: 曾千里 柯文德

实验成绩:

实验日期: 2025 年 3 月 26 日

定时器中断实验

一、实验目的

- 实现定时器溢出时通过串口输出计数信息。通过定时器中断控制 LED 的周期性亮灭，验证中断与硬件协同工作的逻辑。

二、实验仪器和用具

主要仪器设备：

- 计算机
- Keil uVision 软件
- XCOM/SSCOM 串口助手

三、实验方法与步骤

1、定时器原理

分频系数（**PSC**）：10800-1，将 108MHz 主频分频为 10kHz。

重装载值（**ARR**）：5000-1，定时周期为：

$$T = (5000) \times (10800) / 108 \times 10^6 = 0.5 \text{ 秒} (500 \text{ ms})$$

功能：每 500ms 触发一次定时器中断。

2、程序初始化

声明全局变量，count 记录计时溢出次数，led_blinking 记录 led 状态，led_blink_start 记录时间以保证 led 亮起时长。

```
17 TIM_HandleTypeDef TIM3_Handler;          //定时器句柄
18 int count;
19 int led_blinking;
20 uint32_t led_blink_start;
```

3、每次计数溢出时，在串口调试助手屏幕显示次数。每累计 5 次溢出，LED0、LED1 同时亮起 1 秒后再恢复。

在计时器溢出时，增加计数器并输出溢出次数。每五次溢出使得 LED 亮起一秒。此处使用 HAL_GetTick() 函数而非 delay 使得计数和亮灯可以同时进行。

```

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if(htim==&TIM3_Handler)
    {
        count++;
        printf("本次是第 %d 次计数溢出\r\n",count);
        if(count%5==0){
            LED0(0);
            LED1(0);
            led_blink_start = HAL_GetTick();
            led_blinking = 1;
        }
        else if(led_blinking){
            if (HAL_GetTick() - led_blink_start >= 1000) {
                LED0(1);
                LED1(1);
                led_blinking = 0;
            }
        }
    }
}

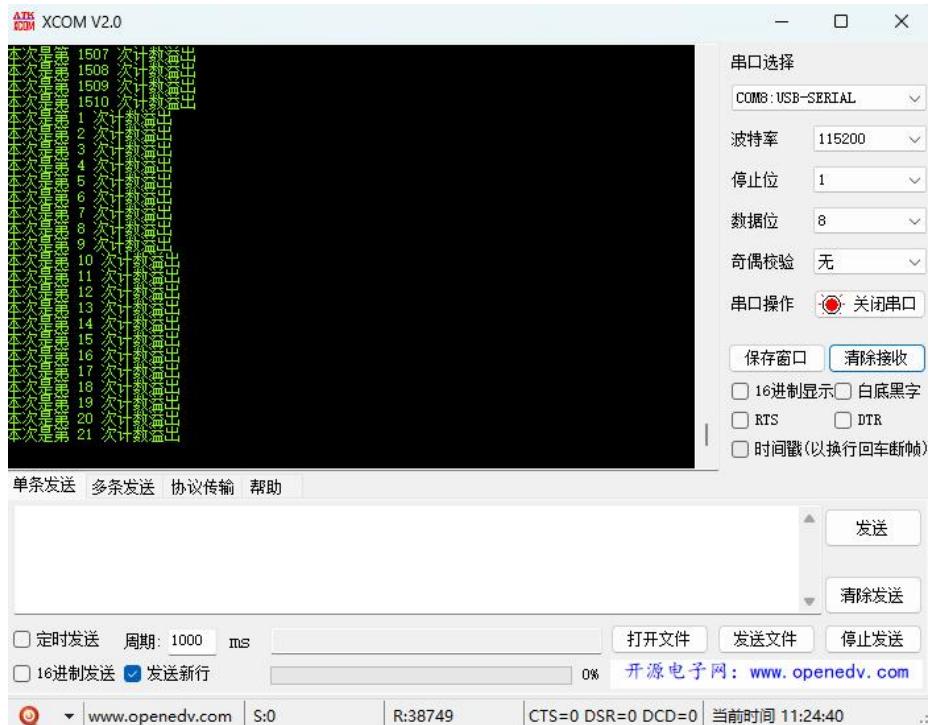
```

四、实验结果

一、实验结果分析

1、串口输出每 500ms 输出“本次是第 X 次计数溢出”

2、每 5 次中断 LED 同时亮起，LED0 和 LED1 亮 1 秒后关闭



二、实验结论

通过本实验，成功实现了以下功能：

- 1、定时器 TIM3 周期性中断控制 LED1 翻转及串口通信。
- 2、每 5 次中断触发 LED0 和 LED1 同步亮灭，验证了中断与硬件协同工作的逻辑。
- 3、串口调试信息的实时输出，增强了系统状态的可观测性。