



机械与能源工程系

SUSTech Department of
Mechanical and Energy
Engineering

实 验 报 告

课程名称： 机电一体化

课程编号： ME3333

实验题目： PWM 输出实验

学 号： 12313215

姓 名： 梁家源

专 业： 机器人工程

指导教师： 柯文德 曾千里

实验成绩：

实验日期： 2025 年 4 月 9 日

输入捕获实验

一、实验目的

本次实验的目的是通过使用 STM32 的定时器 TIM3 的通道 4 来生成 PWM 信号，以控制 LED 亮度的变化。此外，通过按键控制 PWM 输出次数的统计，验证定时器 PWM 的控制效果。

二、实验仪器和用具

主要仪器设备：

1. 计算机
2. Keil uVision 软件

三、实验方法与步骤

1. 定时器 PWM 输出配置

使能 TIM3 定时器和相应的 GPIO 时钟，配置 PB1 为复用功能（TIM3 通道 4 输出）。设置 TIM3 的 ARR（自动重装载寄存器）和 PSC（预分频寄存器），确定 PWM 的频率。配置 TIM3 通道 4 的 PWM 输出模式，并设置占空比。

2. 初始化部分

在代码的主循环部分，程序不断地运行并执行 PWM 控制的任务。首先，程序通过 `delay_ms(1)` 函数执行了一个 1 毫秒的延时，确保程序在每次执行时有一定的时间间隔。接着，`ledOpwmval` 的值根据 `dir` 的状态进行增减。若 `dir` 为 1，`ledOpwmval` 会递增，从而增加 PWM 信号的占空比；若 `dir` 为 0，`ledOpwmval` 递减，减少 PWM 信号的占空比。通过这种方式，PWM 信号的占空比会在一定范围内不断变化。随后，通过 `TIM_SetTIM3Compare4(ledOpwmval)` 和 `TIM_SetTIM3Compare3(ledOpwmval)` 函数调用，分别修改了 TIM3 的通道 4 和通道 3 的比较寄存器值，更新了 PWM 信号的占空比。

```
if(dir) ledOpwmval++;  
else ledOpwmval--;  
if(ledOpwmval>100) dir=0;  
if(ledOpwmval==0) dir=1;  
TIM_SetTIM3Compare4(ledOpwmval);  
TIM_SetTIM3Compare3(ledOpwmval);
```

3. 按键扫描

使用 KEY_Scan(0) 函数检查按键的状态。如果检测到按键被按下，程序会根据按键的不同执行不同的操作。例如，按下 WKUP_PRES 键时，程序会调用 Get_PWMCycleCount1() 函数获取 PWM 输出的周期计数，并通过串口打印输出 PWM 信号的输出次数。之后，LED 会保持亮 2 秒，进入下一步操作。

其他按键操作（KEY1_PRES, KEY2_PRES 等）执行相应的 PWM 输出操作或者控制 LED 亮度的改变，具体行为包括启动计数、停止计数以及修改 PWM 信号的占空比等。

```
key = KEY_Scan(0);

if(key)
{
    switch(key)
    {
        case WKUP_PRES:
            pwm_count = Get_PWMCycleCount1();
            printf("1的第%lu个PWM输出\r\n", (long)pwm_count);

            TIM_SetTIM3Compare4(300);
            HAL_Delay(2000);
            TIM_SetTIM3Compare4(0);
            break;
        case KEY2_PRES:

            pwm_count = Get_PWMCycleCount2();
            printf("0的第%lu个PWM输出\r\n", (long)pwm_count);

            TIM_SetTIM3Compare3(300);
            HAL_Delay(2000);
            TIM_SetTIM3Compare3(0);
            break;
        case KEY1_PRES:

            Count_Begin2();
            printf("开始计数0\r\n");

            HAL_Delay(100);
            TIM_SetTIM3Compare3(0);
            break;
        case KEY0_PRES:
            Count_Begin1();
            printf("开始计数1\r\n");

            HAL_Delay(100);
            TIM_SetTIM3Compare4(0);
            break;
    }
    else delay_ms(10);
}
```

四、实验分析及结论

1. 实验结果

实验中，通过按键控制了两个不同 PWM 输出的计数，并能够正确调节 LED 的

亮度。按键 0 开始计数并显示 PWM 周期，按键 1 停止计数，按键 2 改变另一个 PWM 输出的占空比。

2. 实验总结

本实验成功实现了通过定时器 TIM3 产生 PWM 信号，并通过修改 CCR 寄存器控制 PWM 的占空比。实验验证了定时器的 PWM 输出功能，以及按键控制 PWM 输出计数的功能。同时，通过串口输出结果，达到了预期的实验目的。