**《计算系统设计与实现》**

实验报告

**姓名： 常添**

**班级： 2203102**

**学号： 2022111699**

**哈尔滨工业大学计算学部**

**2025年3月**

### 实验5 三色呼吸灯实验

**一、实验目的（2分）**

** 熟悉 PWM 模块的初始化流程，理解装载值、比较值与占空比的关系。**

** 利用 正弦查表实现 RGB LED 的柔和“呼吸”效果，并学习占空比的实时更新。**

** 掌握 外部中断（PF4/SW1） 的使用方法，实现按键触发的模式切换（彩色呼吸 → 单色灯循环）。**

**二、实验环境（2分）**

| **项目** | **说明** |
| --- | --- |
| **开发板** | **TI EK-TM4C123GXL（TM4C123GH6PM）** |
| **IDE** | **Code Composer Studio 12.8.1** |
| **库** | **TivaWare C Series 2.2.0.295** |
| **接口** | **板载 USB 调试口 (ICDI)、UART0 115 200 bps** |
| **供电** | **USB 5 V** |

**三、实验内容**

1. 实验中系统时钟频率、PWM时钟频率和PWM信号频率分别为多少？写出系统时钟使能和PWM时钟设定的代码。（20分）

**① 系统时钟**：16 MHz（内部主晶振）

**代码：**

**SysCtlClockSet( SYSCTL\_SYSDIV\_1 | SYSCTL\_USE\_OSC |**

**SYSCTL\_OSC\_MAIN | SYSCTL\_XTAL\_16MHZ );**

**② PWM 外设时钟**：16 MHz ÷ 64 = **250 kHz**

**代码：**

**SysCtlPWMClockSet(SYSCTL\_PWMDIV\_64);**

**pwm\_clk = SYSCLK\_HZ / 64; /\* 250 kHz \*/**

**③ PWM 信号频率**：500 Hz

**代码：**

l**oad\_val = (pwm\_clk / PWM\_FREQ\_HZ) - 1; /\* 250 000/500 – 1 = 499 \*/**

**PWM\_FREQ\_HZ = 500; /\* → 500 Hz \*/**

| **频率** | **数值** |
| --- | --- |
| 系统时钟 **f<sub>sys</sub>** | 16 MHz |
| PWM 时钟 **f<sub>pwmclk</sub>** | 250 kHz |
| PWM 输出 **f<sub>PWM</sub>** | 500 Hz |

2. 实现3色led灯控制的PWM通道分别是什么？写出使能相应PWM时钟和配置相应GPIO端口的代码。（20分）

| **颜色** | **PWM 通道** | **引脚** | **生成器** |
| --- | --- | --- | --- |
| Red | **M1PWM5** | PF1 | Generator 2 |
| Green | **M1PWM7** | PF3 | Generator 3 |
| Blue | **M1PWM6** | PF2 | Generator 3 |

**对应代码片段：**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_PWM1); /\* ① 使能 PWM1 模块时钟 \*/**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF); /\* ② 使能 GPIOF 时钟 \*/**

**/\* ③ GPIO 复用到 PWM 功能 \*/**

**GPIOPinConfigure(GPIO\_PF1\_M1PWM5); /\* 红 \*/**

**GPIOPinConfigure(GPIO\_PF2\_M1PWM6); /\* 蓝 \*/**

**GPIOPinConfigure(GPIO\_PF3\_M1PWM7); /\* 绿 \*/**

**GPIOPinTypePWM(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3);**

3. 写出动态更新PWM占空比的主循环程序。（10分）

**while (1)**

**{**

**uint32\_t duty\_r = 1, duty\_g = 1, duty\_b = 1;**

**if (mode == 0) /\* 彩色呼吸 \*/**

**{**

**float angle = idx \* (2.0f \* APP\_PI / STEPS);**

**duty\_r = (uint32\_t)(half\_load \* (1.0f + sinf(angle)));**

**duty\_g = (uint32\_t)(half\_load \* (1.0f + sinf(angle + 2.0f\*APP\_PI/3)));**

**duty\_b = (uint32\_t)(half\_load \* (1.0f + sinf(angle + 4.0f\*APP\_PI/3)));**

**idx = (idx + 1) & (STEPS - 1);**

**}**

**else /\* 单色常亮 (mode = 1/2/3) \*/**

**{**

**switch (mode) {**

**case 1: duty\_r = load\_val; break;**

**case 2: duty\_g = load\_val; break;**

**case 3: duty\_b = load\_val; break;**

**}**

**}**

**/\* 写比较寄存器 —— 最小值保持 1，避免完全关断 \*/**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_5, duty\_r ? duty\_r : 1);**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_6, duty\_g ? duty\_g : 1);**

**PWMPulseWidthSet(PWM1\_BASE, PWM\_OUT\_7, duty\_b ? duty\_b : 1);**

**SysCtlDelay(SYSCLK\_HZ / 250); /\* 4 ms 一步，呼吸周期约 1 s \*/**

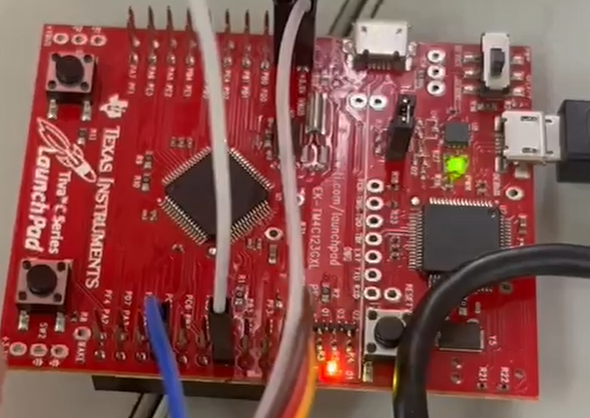
**}**

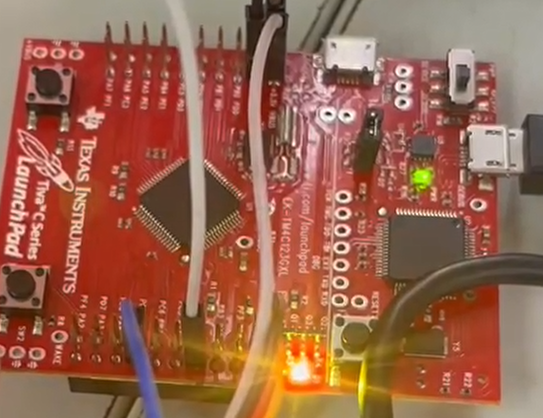
**四、实验结果（4分）**

1. **观察到的呼吸灯结果**：

* 上电后 RGB LED 以约 1 s 周期进行亮度渐变，呈平滑的彩色呼吸效果。
* 按下 **板载 SW1 (PF4)** 触发外部中断：
  1. 第一次按键 → LED 立即转为 **红光常亮**
  2. 第二次 → **绿光常亮**
  3. 第三次 → **蓝光常亮**
  4. 后续按键在红/绿/蓝三色间循环。

**2. 效果图（图片）**：

**• 彩色呼吸渐变过程 • 单色红/绿/蓝常亮状态（中断）**

****

**五、实验总结（2分）**

* **PWM + LUT：通过查表法配合 sinf() 生成 0–100 % 占空比，避免在中断中做浮点计算，主循环刷新可达 250 Hz 以上。**
* **外部中断：采用 PF4 下降沿触发，并加入 20 ms 软件去抖，按键响应稳定可靠。**
* **功耗：单色常亮模式占空比 100 %，电流最大；呼吸模式平均亮度约 50 %，有一定节能效果。**
* **改进方向**
  + **使用 DMA + PWM 死区 可以完全脱离 CPU 参与，实现更精确、更省电的波形输出。**
  + **将模式切换加入长按/双击判断，可扩展更多灯效。**

**实验顺利完成，达到预期目标。**