



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： ME333

实验题目： 按键输入实验

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

# 按键输入实验

## 一、实验目的

1. 利用板载的 4 个按键，来控制板载的两个 LED 的亮灭。

## 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、代码执行流程

主循环持续调用按键检测函数，获取按键值 key。根据 key 值进入 switch 分支，执行对应 LED 状态操作。每次按键操作后更新 LED0 和 LED1 的输出状态，未检测到按键时延时 10ms 降低 CPU 占用。

### 2、在 void KEY\_Init(void)中修改按键优先级别 KEY0<KEY1<KEY2<WK\_UP

通过调整判断 KEY0、KEY1、KEY2、WK\_UP 是否按下的 case 语句位置，优先级高的在前面，来更改按键优先级别。

```
if(key)
{
    switch(key)
    {
        case WKUP_PRES:
            break;
        case KEY2_PRES:
            break;
        case KEY1_PRES:
            break;
        case KEY0_PRES:
            break;
    }
    LED0(led0sta);    //控制LED0状态
    LED1(led1sta);    //控制LED1状态
}else delay_ms(10);
```

3、按下 WK\_UP 时，按下时，led0sta 和 led1sta 恢复到初始状态。通过设置 led0sta 和 led1sta 的值为 1 即可实现。

```
case WKUP_PRES: //控制LED0,LED1恢复至初始状态
    led1sta=1;
    led0sta=1;
    break;
```

4、按下 KEY2 时，LED0 和 LED1 同时闪烁 5 次，每次闪烁间隔 200ms。

在 main 函数中先声明 int 类型变量 i 作为 while 循环的控制条件。在 KEY2 按下时，将 i 值重置为 0。进入 while 循环后先将 led1sta、led0sta 设为 0 并通过 LED0(led0sta)和 LED1(led1sta)语句来控制亮灯，然后通过 delay\_ms(100)来保持亮灯 100ms 以确保能被看到；再将 led1sta、led0sta 设为 1 并通过 LED0(led0sta)和 LED1(led1sta)语句来控制灭灯，并通过 delay\_ms(200)来实现要求的 200ms；最后通过 i++来实现以上闪烁过程循环 5 次。

```
case KEY2_PRES: //LED0,LED1同时闪烁5次，每次闪烁间隔200ms
    i=0;
    led0sta=1;
    led1sta=1;
    LED0(led0sta);
    LED1(led1sta);
    delay_ms(200);
    while(i<5){
        led1sta=0;
        led0sta=0;
        LED0(led0sta);
        LED1(led1sta);
        delay_ms(100);
        led0sta=1;
        led1sta=1;
        LED0(led0sta);
        LED1(led1sta);
        delay_ms(200);
        i++;
    }
    break;
```

5、按下 KEY1 时，led0sta 翻转，led1sta 取反，实现 LED0 和 LED1 交替闪烁。

通过 led0sta 对自身取反来实现 LED0 的翻转效果，通过 led1sta 对 led0sta 取反以实现两灯交替闪烁的效果。

```
case KEY1_PRES: //LED0翻转,LED1取反
    led0sta=!led0sta;
    led1sta=!led0sta;
    break;
```

6、按下 KEY0 时，led0sta 和 led1sta 同时翻转，实现 LED0 和 LED1 同时亮或灭。

通过 led0sta 和 led1sta 同时对自身取反即可实现同时翻转。

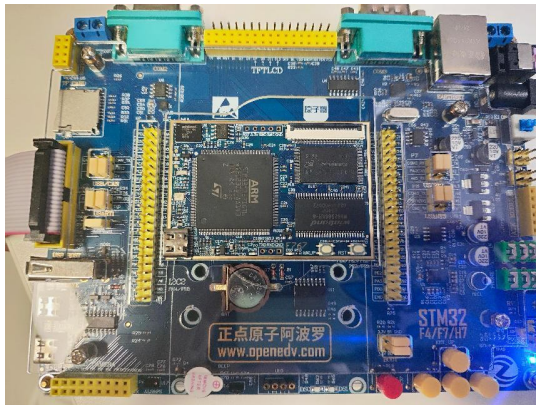
```
case KEY0_PRES: //LED0,LED1同时翻转
    led0sta=!led0sta;
    led1sta=!led1sta;
    break;
```

## 四、实验分析及结论

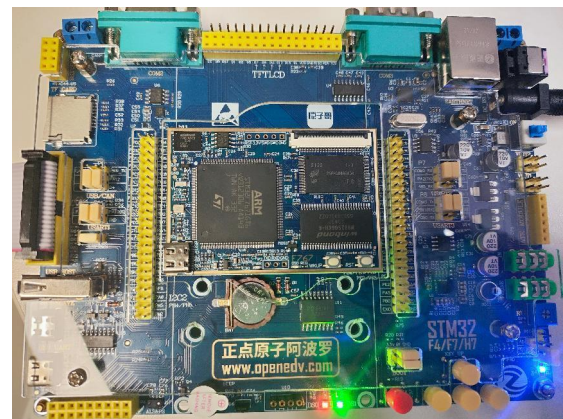
### 一、实验结果分析

#### 1. 功能实现验证

同步亮灭 (KEY0) : led0sta 与 led1sta 同步取反, 逻辑正确, LED 同步亮灭效果稳定。



全灭



全亮



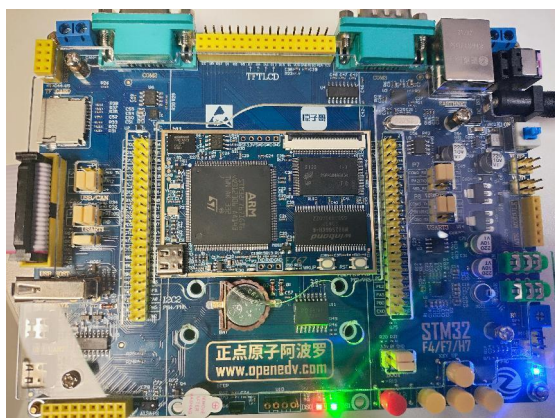
红亮绿灭



绿亮红灭



交替闪烁 (KEY1)：led1sta 依赖 led0sta 的新值取反，逻辑无误。例如，若原 led0sta=0 (亮)，按下后 led0sta=1 (灭)，led1sta=0 (亮)，实现交替。

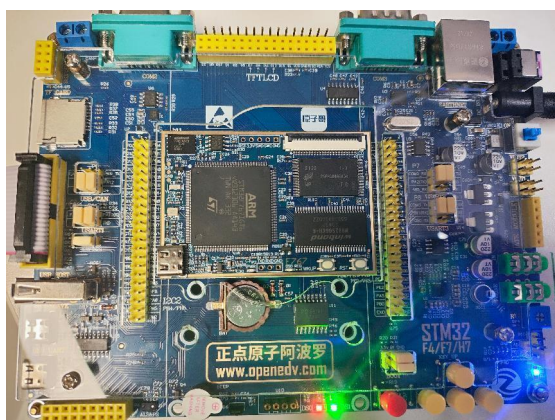


全亮

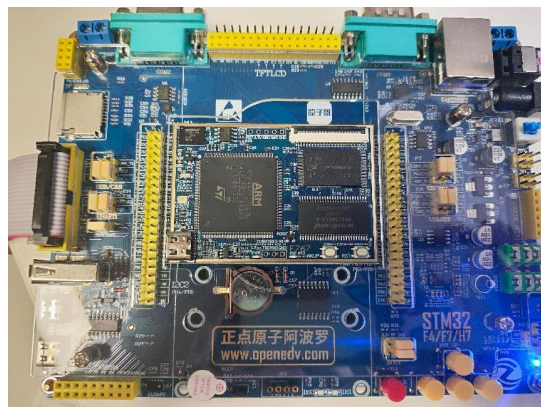


绿亮红灭

定时闪烁 (KEY2)：通过 while 循环与 delay\_ms 实现 5 次闪烁。

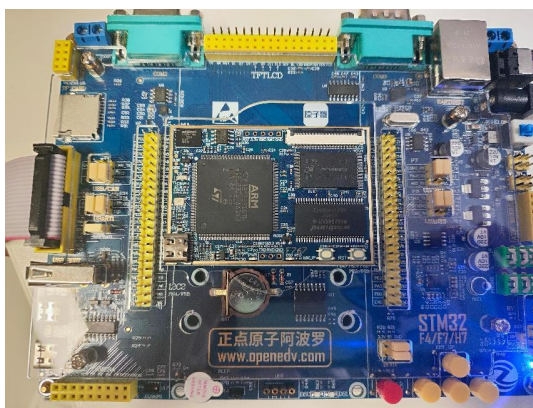


全亮



全灭

状态复位 (WK\_UP)：强制置 1 使 LED 灭，符合初始状态定义。



全灭

## 2. 潜在问题

**阻塞延时影响响应：**KEY2 的 delay\_ms 阻塞主循环，导致闪烁期间无法检测其他按键。

## 二、实验结论

本实验通过修改按键优先级与 LED 控制逻辑，实现了四类按键对 LED 的多样化控制功能，核心需求均得到满足。然而，阻塞延时存在优化空间，后续可通过引入定时器中断或非阻塞定时器提升系统响应效率与实时性。实验结果表明，基础功能实现正确，但需进一步优化以适应复杂应用场景。