



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： 串口通信实验

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 12 日

外部中断实验

一、实验目的

1. 利用单片机上的四个按键实现外部中断并向上位机发送通知。

二、实验仪器和用具

主要仪器设备：

1. 计算机
2. Keil uVision 软件
3. XCOM/SSCOM 串口助手

三、实验方法与步骤

1、代码执行流程

主循环持续监测 GPIO_PIN，获取按键值 key。按下按键后，对应 IO 口输入中断，并向上位机发送通知。

2、串口助手初始化

调整波特率为 115200 并连接上串口。



3、按下 Key_UP 后，中断线 GPIO_PIN_0 触发。

如程序所示，按下 Key_UP 时，GPIO_PIN 被触发为 GPIO_PIN_0，控制 LED0，LED1 互斥点亮，并向上位机发送“WK_UP 被按下，中断线 GPIO_PIN_0 触发，现在 LED0 处于 XXX 状态，LED1 处于 XXX 状态。”

```
case GPIO_PIN_0:
    if(WK_UP==1) //控制LED0,LED1互斥点亮
    {
        led1sta=!led1sta;
        led0sta=!led1sta;
        LED1(led1sta);
        LED0(led0sta);
        printf("WK_UP被按下，中断线GPIO_PIN_0触发，现在LED0处于%s状态，LED1处于%s状态。\\r\\n",led0sta ? "灭" : "亮",led1sta ? "灭" : "亮");
    }
    break;
```

4、按下 Key1 后，中断线 GPIO_PIN_2 触发。

如程序所示，按下 Key1 时，GPIO_PIN 被触发为 GPIO_PIN_2，控制 LED1 翻转，并向上位机发送“Key1 被按下，中断线 GPIO_PIN_2 触发，现在 LED0 处于 XXX 状态，LED1 处于 XXX 状态。”

```
case GPIO_PIN_2:
    if(KEY1==0) //控制LED1翻转
    {
        led1sta=!led1sta;
        LED1(led1sta);
        printf("KEY1被按下，中断线GPIO_PIN_2触发，现在LED0处于%s状态，LED1处于%s状态。\\r\\n",led0sta ? "灭" : "亮",led1sta ? "灭" : "亮");
    };
    break;
```

5、按下 Key0 后，中断线 GPIO_PIN_3 触发。

如程序所示，按下 Key0 时，GPIO_PIN 被触发为 GPIO_PIN_3，控制 LED1 翻转，并向上位机发送“Key0 被按下，中断线 GPIO_PIN_3 触发，现在 LED0 处于 XXX 状态，LED1 处于 XXX 状态。”

```
case GPIO_PIN_3:
    if(KEY0==0) //同时控制LED0,LED1翻转
    {
        led1sta=!led1sta;
        led0sta=!led0sta;
        LED1(led1sta);
        LED0(led0sta);
        printf("KEY0被按下，中断线GPIO_PIN_3触发，现在LED0处于%s状态，LED1处于%s状态。\\r\\n",led0sta ? "灭" : "亮",led1sta ? "灭" : "亮");
    }
    break;
```

5、按下 Key2 后，中断线 GPIO_PIN_13 触发。

如程序所示，按下 Key2 时，GPIO_PIN 被触发为 GPIO_PIN_13，控制 LED1 翻转，并向上位机发送“Key2 被按下，中断线 GPIO_PIN_13 触发，现在 LED0 处于 XXX 状态，LED1 处于 XXX 状态。”

```
case GPIO_PIN_13:
    if(KEY2==0) //控制LED0翻转
    {
        led0sta=!led0sta;
        LED0(led0sta);
        printf("KEY2被按下，中断线GPIO_PIN_13触发，现在LED0处于%s状态，LED1处于%s状态。\\r\\n",led0sta ? "灭" : "亮",led1sta ? "灭" : "亮");
    }
    break;
```

四、实验分析及结论

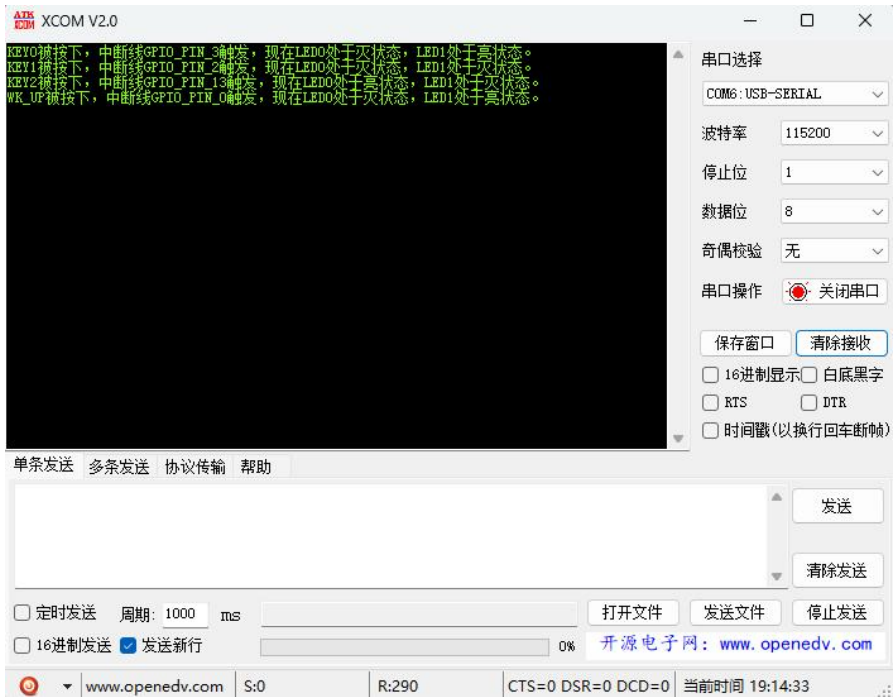
一、实验结果

1. 中断触发与 LED 控制

成功配置外部中断，将按键（如 KEY0、KEY1）连接到对应的 GPIO 引脚（例如 PA0、PA1），并设置为下降沿触发。按下按键时，LED 状态实时切换，中断响应迅速，LED 状态切换稳定，无延迟或误触发现象。

2. 串口通信与信息显示

使用 XCOM V2.0 串口助手，波特率设置为 115200，每次按键触发中断时，串口均能正确接收并显示如下格式信息：KEY0 被按下，中断线 GPIO_PIN_3 触发，现在 LED0 处于熄灭状态，LED1 处于点亮状态。信息内容与 LED 实际状态完全一致，无数据丢失或格式错误。



二、实验结论

本实验通过配置 STM32 的外部中断，实现了按键触发实时控制 LED 状态，并通过串口反馈操作信息。实验结果表明：外部中断能够高效响应外部事件，满足实时性要求。中断服务函数与回调函数的层级设计，提高了代码的可维护性。串口通信作为调试手段，能有效验证程序逻辑的正确性。