



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： 红外遥控实验

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 4 月 3 日

红外遥控实验

一、实验目的

- 1.实现红外遥控器按键次数统计，并通过串口在上位机实时显示。
- 2.在 LCD 屏幕上实现数字输入、加减运算及结果显示功能，支持负数显示与清零操作

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显示按键次数

1. 初始化变量，用 str 指示当前输入按键，count[21]统计 21 个按键各自的按下次数

```
u8 key;  
u8 t=0;  
u8 *str=0;  
static u8 last_key = 0  
int count[21]={0};  
int chr = 0;
```

```

switch(key)
{
    case 0: str="ERROR"; chr=1; break;
    case 162: str="POWER"; chr=2; break;
    case 98: str="UP"; chr=3; break;
    case 2: str="PLAY"; chr=4; break;
    case 226: str="ALIENTEK"; chr=5; break;
    case 194: str="RIGHT"; chr=6; break;
    case 34: str="LEFT"; chr=7; break;
    case 224: str="VOL-"; chr=8; operation=2; input_stage=2; break;
    case 168: str="DOWN"; chr=9; break;
    case 144: str="VOL+"; chr=10; operation=1; input_stage=2; break;
    case 104: str="1"; chr=11; digit_val=1; is_digit=1; break;
    case 152: str="2"; chr=12; digit_val=2; is_digit=1; break;
    case 176: str="3"; chr=13; digit_val=3; is_digit=1; break;
    case 48: str="4"; chr=14; digit_val=4; is_digit=1; break;
    case 24: str="5"; chr=15; digit_val=5; is_digit=1; break;
    case 122: str="6"; chr=16; digit_val=6; is_digit=1; break;
    case 16: str="7"; chr=17; digit_val=7; is_digit=1; break;
    case 56: str="8"; chr=18; digit_val=8; is_digit=1; break;
    case 90: str="9"; chr=19; digit_val=9; is_digit=1; break;
    case 66: str="0"; chr=20; digit_val=0; is_digit=1; break;
    case 82:
        str="DELETE";
        num1=0;
        num2=0;
        operation=0;
        input_stage=0;
        chr=21;
        LCD_Fill(30, 130, 240, 220, WHITE); // 清除计算区域
        LCD_ShowString(30, 130, 40, 16, 16, "Num1:"); // 重绘标签
        LCD_ShowString(30, 150, 40, 16, 16, "Op: ");
        LCD_ShowString(30, 170, 40, 16, 16, "Num2:");
        LCD_ShowString(30, 200, 80, 16, 16, "Result:");

        LCD_ShowNum(80, 130, 0, 5, 16); // Num1
        LCD_ShowNum(80, 170, 0, 5, 16); // Num2
        break;
    default: str="UNKNOWN"; chr=0; break;
}
if(chr>=1 && chr<=21){
    count[chr-1]++;
    printf("本次是第 %d 次按下 %s\r\n", count[chr-1], str);
}

```

2. 修改原有代码，通过检测 chr 值来判断输入按键并用 count[]来统计按下次数并通过上位机显示次数。

②加减运算功能

```

if(is_digit || key==144 || key==224) {
    // 显示第一个数
    LCD_ShowString(30, 130, 40, 16, 16, "Num1:");
    LCD_ShowNum(80, 130, num1, 16, 16);

    // 显示操作符
    LCD_ShowString(30, 150, 40, 16, 16, "Op: ");
    if(operation == 1) LCD_ShowString(200, 150, 16, 16, 16, "+");
    else if(operation == 2) LCD_ShowString(200, 150, 16, 16, 16, "-");
    else LCD_Fill(80, 150, 96, 166, WHITE); // 清除操作符

    // 显示第二个数
    LCD_ShowString(30, 170, 40, 16, 16, "Num2:");
    LCD_ShowNum(80, 170, num2, 16, 16);
}

```

1. 初始化变量, num1 和 num2 为待运算的数字, operation 指示运算符, input_stage 指示输入状态, 0 时为输入第一个数字, 1 时为输入操作符, 2 时为输入第二个数字。

```

static int num1 = 0;
static int num2 = 0;
static char operation = 0;
static u8 input_stage = 0;
int is_digit = 0;
int digit_val = -1;

```

2. 数字输入处理

```

if(is_digit) {
    if(input_stage == 0 || input_stage == 1) { // 输入第一个数
        num1 = num1*10 + digit_val;
        input_stage = 1;
    } else if(input_stage == 2) { // 输入第二个数
        num2 = num2*10 + digit_val;
    }
}

```

通过 input_stage 状态来控制两个数分别的连续输入

```

if(input_stage == 2 && num2 !=0) {
    int result = operation==1 ? (num1+num2) : (num1-num2);

    LCD_ShowString(30,200,80,16,16,"Result:");
    LCD_Fill(100,200,240,216,WHITE);

    // 添加符号显示
    if(result <0) {
        LCD_ShowString(100,200,16,16,16,"-");
        LCD_ShowNum(116,200,-result,12,16); // 显示绝对值
    } else {
        LCD_Fill(100,200,116,216,WHITE); // 清除负号
        LCD_ShowNum(110,200,result,12,16);
    }
}

```

当第二个数输入后显示结果，分正负数处理（若不分开，显示可能有问题）

3. 数字显示

```

if(is_digit || key==144 || key==224) {
    // 显示第一个数
    LCD_ShowString(30,130,40,16,16,"Num1:");
    LCD_ShowNum(80,130,num1,16,16);

    // 显示操作符
    LCD_ShowString(30,150,40,16,16,"Op: ");
    if(operation == 1) LCD_ShowString(200,150,16,16,16,"+");
    else if(operation == 2) LCD_ShowString(200,150,16,16,16,"-");
    else LCD_Fill(80,150,96,166,WHITE); // 清除操作符

    // 显示第二个数
    LCD_ShowString(30,170,40,16,16,"Num2:");
    LCD_ShowNum(80,170,num2,16,16);
}

```

通过 LCD_ShowString 和 LCD_ShowNum 来显示输入提示和计算结果。

4. 内容清空

```

// DELETE键清除显示
if(key == 82) {
    LCD_Fill(30,130,240,220,WHITE); // 清除从130行到220行的区域
    LCD_ShowString(30,130,40,16,16,"Num1:");
    LCD_ShowNum(80,130,num1,16,16);
    LCD_ShowString(30,150,40,16,16,"Op: ");
    LCD_ShowString(30,170,40,16,16,"Num2:");
    LCD_ShowString(30,200,80,16,16,"Result:");
    LCD_ShowNum(80,170,num2,16,16);
}

```

当输入按键为 DELETE 键时，将数字和符号显示区域清空。

5. 按键间断输入

```
64 while(1)
65 {
66     key=Remote_Scan();
67     if(key)
68     {
69         if(last_key ==0){
70             last_key = key;
71             chr=0;
72             is_digit = 0;
73             digit_val = -1;
74             switch(key)
75             {
114                 if(chr>=1 && chr<=21){
118                     if(is_digit) {
126                         if(is_digit || key==144 || key==224) {
142                             // 结果显示部分 (位置: 30,200)
143                             if(input_stage == 2 && num2 !=0){
159                                 // DELETE键清除显示
160                                 if(key == 82){
170                                     }
171                                 }
172                             else {
173                                 last_key=0;
174                                 delay_ms(10);
175                             }
176                             t++;
177                             if(t==20)
178                             {
179                                 t=0;
180                                 LED0_Toggle;
181                             }
182     }
```

原代码中按键按下时会一直发送信号导致多次判定，现增加 last_key 判断按键是否松开，即在松开后才能进行下一次输入，以实现按一次只发送一次数据

四、实验分析及结论

一、实验结果分析

1. 按键统计

操作：依次按下不同按键（如数字键 1、POWER 键）

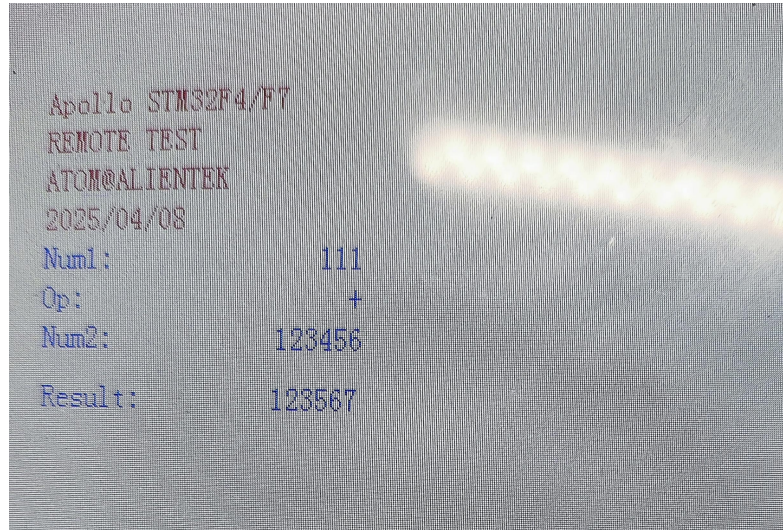
结果：上位机串口输出按键名称及按下次数，LCD 显示当前按键名称（SYMBOL 区域）。



2. 加法测试

操作：输入 123 → VOL+ → 456

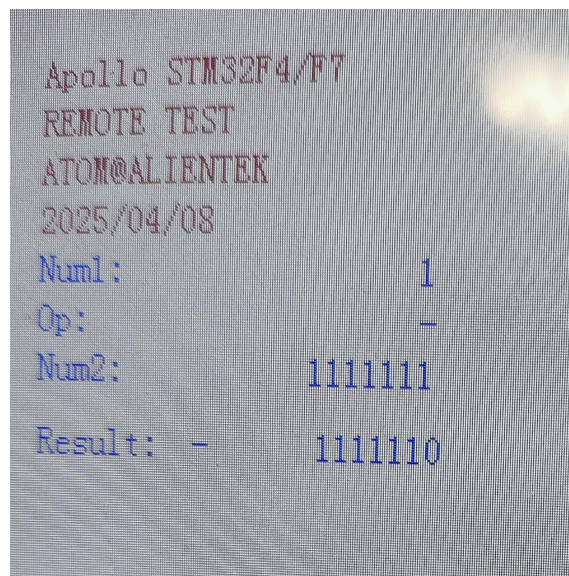
结果：LCD 显示：Num1: 123 Op:+ Num2:456 Result:579



3. 减法测试

操作：输入 200 → VOL- → 300

结果：LCD 显示：Num1: 200 Op:- Num2:300 Result:-100



4. 清零功能测试

操作：在任意阶段按下 DELETE 键

结果：所有显示归零：Num1: 0 Op: Num2:0 Result:

二、实验结论

本实验成功实现了红外按键统计与简易计算器功能，系统运行稳定，显示效果符合设计要求。通过问题分析与优化，进一步提升了代码健壮性与用户体验，为复杂嵌入式应用开发奠定了基础。