

东南大学

嵌入式系统设计 实验报告

姓名：叶宏庭 学号：71118415

姓名：李钰翰 学号：71118411

时间： 2021 年 6 月

基于键盘实验的简易运算器

一、实验目的：

- 1、学习 TM4C123G 的 GPIO 使用及其相关的 API 函数；
- 2、掌握 GPIO 的模拟 TM1638 的控制时序。
- 3、使用 GPIO 控制键盘，实现简单的二元运算：加减乘除、指数、取模
- 4、实现计算器中的 PRE_ANS 键，记录上一次结果。

二、实验内容：

- 1) 用 TM4C123G 的 GPIO 控制 TM1638 驱动 8 段数码管、LED、键盘。
- 2) 使用键盘，实现简单的二元运算：加减乘除、指数、取模
- 3) 实现计算器中的 PRE_ANS 键，记录上一次结果，方便后续使用。

三、实验设备：

- 1、TM4C123G 教学实验箱，PentiumII 以上的 PC 机，J-link 仿真器；
- 2、PC 操作系统 WIN98 或 WIN2000 或 WINXP，Keil MDK4.01 集成开发环境，J-link 仿真调试驱动程序。

四、实验原理：

- 1) 使用 TM4C123G 的 GPIO 端口模拟 TM1638 的时序驱动 8 段数码管、LED、键盘。
- 2) 修改键盘的底层函数，实现运算逻辑。

五、实验要求

- 1) 实现二元加法。
- 2) 实现二元减法。
- 3) 实现二元乘法。
- 4) 实现二元除法。
- 5) 实现指数运算
- 5) 实现模运算。
- 6) 实现存储历史结果功能（PRE_ANS 键）。

六、实验步骤：

- 1、连线：将实验箱 5V/2A 电源线连接好，并将 J-link 仿真器与试验箱 CPU 板和电脑连接好；
- 2、新建一个文件夹，在里面建立一个 keil 工程，然后完成相应代码；
- 3、点击 Build 按钮，编译程序；
- 4、编译成功后，实验箱上电，点击 Load 按钮下载程序；
- 5、下载成功，进入测试

七、实验程序

```

void TM1638Test(void)
{
    int index = 0; // 标志位（数1
or 数2）
    int num1=0; // 数字一
    int num2=0; // 数字二
    int pre_ans = 0; // 先前answer
    static char result1[10]; // 结果一字符串
    static char result2[10]; // 结果二字符串
    char expression[100] = {0}; // 完整表达式
    char opt[1] = "w"; // 运算位
    char equals_symbol[1] = "="; // 计算结果操作符
    int cal_result=0; // 计算结果
    static char cal_str_result[10]; // 结果字符串
    u8 i, j;
    u8 keyDat;
    // u8 tempcount=0;
    u8 temp_key1=0x00;
    u8 temp_key2=0x00;
    u8 temp_key3=0x00;

    Init_TM1638();

    for(i=0;i<16;i++)
    {
        for(j=0;j<8;j++)
        {
            Write_DATA(j<<1, tab[i]);
            delay_tim(1000);
        }
    }
    delay_tim(100);

    for(j=0;j<8;j++)
        Write_DATA(j<<1, tab[0]);

    for(i=0;i<16;i++)
    {
        i++;
        for(j=0;j<3;j++)
        {
            Write_oneLED(i, j);
            delay_tim(1000);
        }
    }
}

```

```

}

//while(tempcount<24)
//while(1)
temp_key1=0x00;
temp_key2=0x00;
temp_key3=0x00;

while((temp_key1!=0xff) || (temp_key2!=0xff) || (temp_key3!=0xff))
//while(0)
{
    keyDat=Read_key();

#if 1
    keyDat=keytab[keyDat-1];
    if(keyDat<0x1f)
    {
        //printf("%c\n",keyDat);
        //for(j=0;j<8;j++)
        Write_DATA(2, tab[keyDat&0x0f]);
        Write_DATA(0, tab[(keyDat&0xf0)>>4]);
        if(keyDat!=key_tmp)
        {
            //tempcount++;
            key_tmp=keyDat;
        }
        switch (keyDat) {
            case 0x01:
                if(index==0) {
                    // 标志数1操作
                    num1 = num1 * 10 + 1;
                    // 数1变化
                    LCD_Clear(Black);
                    // 清屏
                    sprintf(result1, "%d", num1);
                    // 数1打印
                    LCD_ShowString(120, 120, result1);
                    // 打印数字1
                } else {
                    // 标志数2操作
                    num2 = num2 * 10 + 1;
                    //数2变化
                    LCD_Clear(Black);
                    // 清屏
                    sprintf(result2, "%d", 1);
                    // 打印结果2

```

表达式

```
        strcat(expression, result2);                                // 拼接

        LCD_ShowString(120, 120, expression);                      //展示结果表达式
    }

    break;
case 0x02:
    if(index==0) {
        num1 = num1 * 10 + 2;
        LCD_Clear(Black);
        sprintf(result1, "%d", num1);
        LCD_ShowString(120, 120, result1);
    } else {
        num2 = num2 * 10 + 2;
        LCD_Clear(Black);
        sprintf(result2, "%d", 2);
        strcat(expression, result2);
        LCD_ShowString(120, 120, expression);
    }
    break;
case 0x03:
    if(index==0) {
        num1 = num1 * 10 + 3;
        LCD_Clear(Black);
        sprintf(result1, "%d", num1);
        LCD_ShowString(120, 120, result1);
    } else {
        num2 = num2 * 10 + 3;
        LCD_Clear(Black);
        sprintf(result2, "%d", 3);
        strcat(expression, result2);
        LCD_ShowString(120, 120, expression);
    }
    break;
case 0x04:
    if(index==0) {
        num1 = num1 * 10 + 4;
        LCD_Clear(Black);
        sprintf(result1, "%d", num1);
        LCD_ShowString(120, 120, result1);
    } else {
        num2 = num2 * 10 + 4;
        LCD_Clear(Black);
        sprintf(result2, "%d", 4);
```

```

        strcat(expression, result2);
        LCD_ShowString(120, 120, expression);
    }
break;
case 0x05:
    if(index==0) {
        num1 = num1 * 10 + 5;
        LCD_Clear(Black);
        sprintf(result1, "%d", num1);
        LCD_ShowString(120, 120, result1);
    } else {
        num2 = num2 * 10 + 5;
        LCD_Clear(Black);
        sprintf(result2, "%d", 5);
        strcat(expression, result2);
        LCD_ShowString(120, 120, expression);
    }
break;
case 0x06:
    if(index==0) {
        num1 = num1 * 10 + 6;
        LCD_Clear(Black);
        sprintf(result1, "%d", num1);
        LCD_ShowString(120, 120, result1);
    } else {
        num2 = num2 * 10 + 6;
        LCD_Clear(Black);
        sprintf(result2, "%d", 6);
        strcat(expression, result2);
        LCD_ShowString(120, 120, expression);
    }
break;
case 0x07:
    if(index==0) {
        num1 = num1 * 10 + 7;
        LCD_Clear(Black);
        sprintf(result1, "%d", num1);
        LCD_ShowString(120, 120, result1);
    } else {
        num2 = num2 * 10 + 7;
        LCD_Clear(Black);
        sprintf(result2, "%d", 7);
        strcat(expression, result2);
        LCD_ShowString(120, 120, expression);
    }

```

```

    }
break;
case 0x08:
if(index==0) {
    num1 = num1 * 10 + 8;
    LCD_Clear(Black);
    sprintf(result1, "%d", num1);
    LCD_ShowString(120, 120, result1);
} else {
    num2 = num2 * 10 + 8;
    LCD_Clear(Black);
    sprintf(result2, "%d", 8);
    strcat(expression, result2);
    LCD_ShowString(120, 120, expression);
}
break;
//
case 0x09:
if(index==0) {
    num1 = num1 * 10 + 9;
    LCD_Clear(Black);
    sprintf(result1, "%d", num1);
    LCD_ShowString(120, 120, result1);
} else {
    num2 = num2 * 10 + 9;
    LCD_Clear(Black);
    sprintf(result2, "%d", 9);
    strcat(expression, result2);
    LCD_ShowString(120, 120, expression);
}
break;
case 0x0a:
if(index==0) {
    opt[0] = '+';
    strcat(expression, result1);
    strcat(expression, opt);
    index = index + 1;
    LCD_Clear(Black);
    LCD_ShowString(120, 120, expression);
} else {
    LCD_Clear(Black);
    LCD_ShowString(120, 120, "Error input!!");
}
break;

```

```

case 0x0b:
    if(index==0) {
        opt[0] = '-' ;
        strcat(expression, result1);
        strcat(expression, opt);
        index = index + 1;
        LCD_Clear(Black);
        LCD_ShowString(120, 120, expression);
    } else {
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Error input!!");
    }
break;
case 0x0c:
    if(index==0) {
        opt[0] = '*' ;
        strcat(expression, result1);
        strcat(expression, opt);
        index = index + 1;
        LCD_Clear(Black);
        LCD_ShowString(120, 120, expression);
    } else {
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Error input!!");
    }
break;
case 0x0d:
    if(index==0) {
        opt[0] = '/' ;
        strcat(expression, result1);
        strcat(expression, opt);
        index = index + 1;
        LCD_Clear(Black);
        LCD_ShowString(120, 120, expression);
    } else {
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Error input!!");
    }
break;
case 0x0e:
    if(index==1) {
        strcat(expression, equals_symbol);
        switch(opt[0]) {
            case '+':

```



```

        cal_result = num1 + num2;
        break;
    case '-':
        cal_result = num1 - num2;
        break;
    case '*':
        cal_result = num1 * num2;
        break;
    case '/':
        cal_result = num1 / num2;
        break;
    }
    pre_ans = cal_result;
    sprintf(cal_str_result, "%d", cal_result);
    strcat(expression, cal_str_result);
    LCD_Clear(Black);
    LCD_ShowString(120, 120, expression);
    index = 0;
    num1=0;
    num2=0;
} else{
    LCD_Clear(Black);
    LCD_ShowString(120, 120, expression);
}
break;
case 0x0f:
    index = 0;
    num1=0;
    num2=0;
    memset(expression, '\0', sizeof(expression));
    LCD_Clear(Black);
    LCD_ShowString(120, 120, expression);
break;
case 16:
    if(index==0){
        num1 = pre_ans;
        LCD_Clear(Black);
        sprintf(result1, "%d", num1);
        LCD_ShowString(120, 120, result1);
    } else{
        num2 = pre_ans;
        LCD_Clear(Black);
        sprintf(result2, "%d", pre_ans);
        strcat(expression, result2);
    }
}

```

```

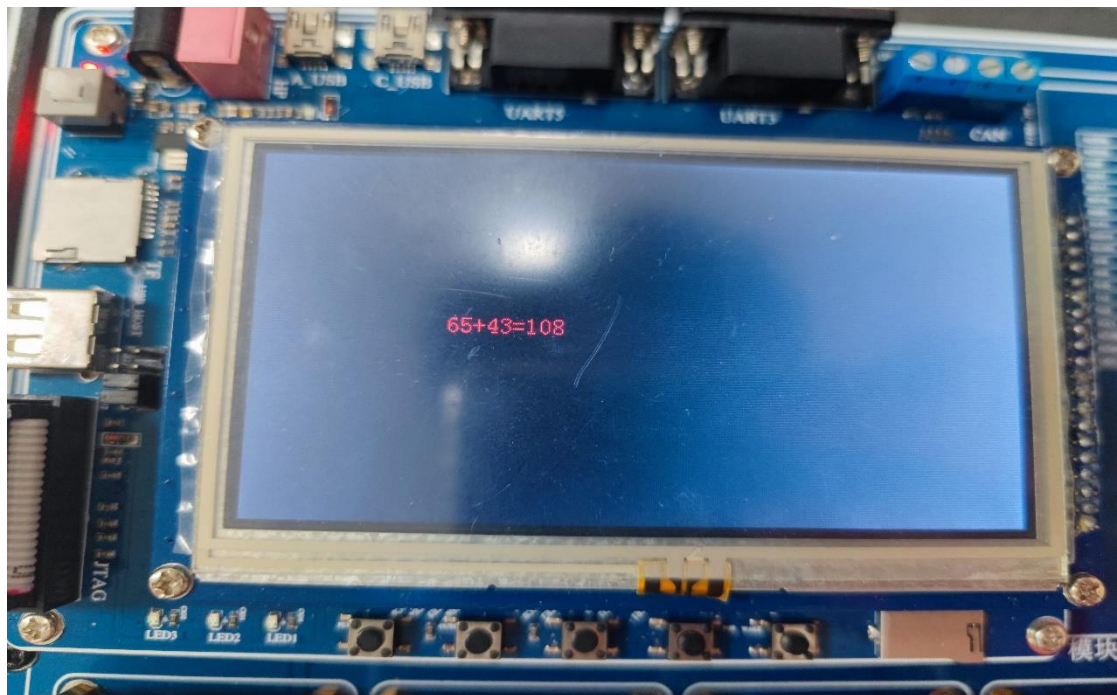
        LCD_ShowString(120, 120, expression);
    }
    break;
//
    case 0x11:
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Not available key!!");
        temp_key3=temp_key3|0x01;
        break;
    case 0x12:
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Not available key!!");
        temp_key3=temp_key3|0x02;
        break;
    case 0x13:
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Not available key!!");
        temp_key3=temp_key3|0x04;
        break;
    case 0x14:
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Not available key!!");
        temp_key3=temp_key3|0x08;
        break;
    case 0x15:
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Not available key!!");
        temp_key3=temp_key3|0x10;
        break;
    case 0x16:
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Not available key!!");
        temp_key3=temp_key3|0x20;
        break;
    case 0x17:
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Not available key!!");
        temp_key3=temp_key3|0x40;
        break;
    case 0x18:
        LCD_Clear(Black);
        LCD_ShowString(120, 120, "Not available key!!");
        temp_key3=temp_key3|0x80;
        break;

```

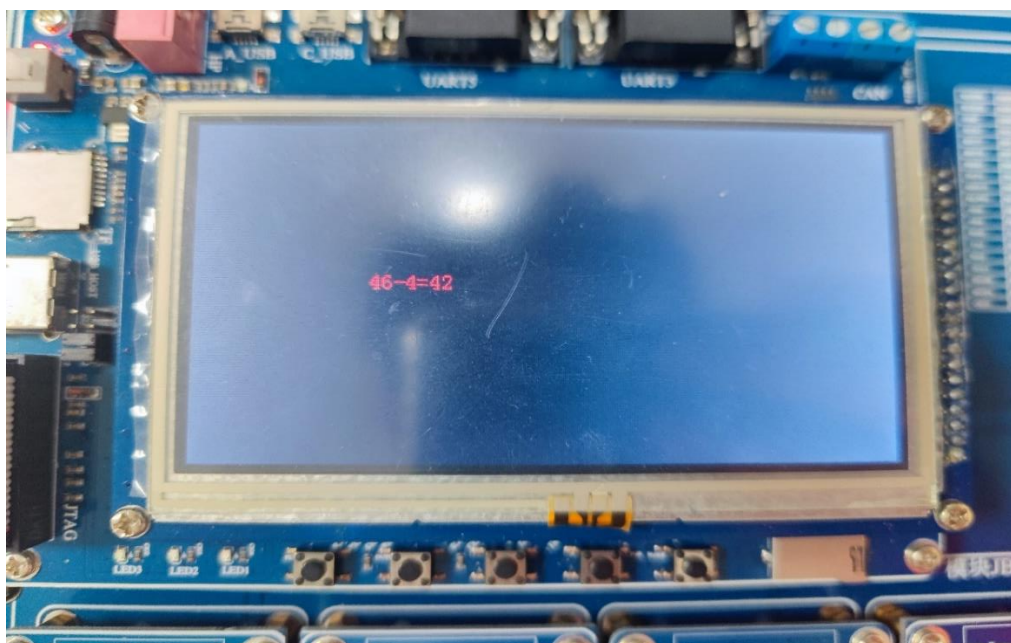
```
        default:  
            break;  
        }  
    }  
#endif  
}
```

八、实验结果

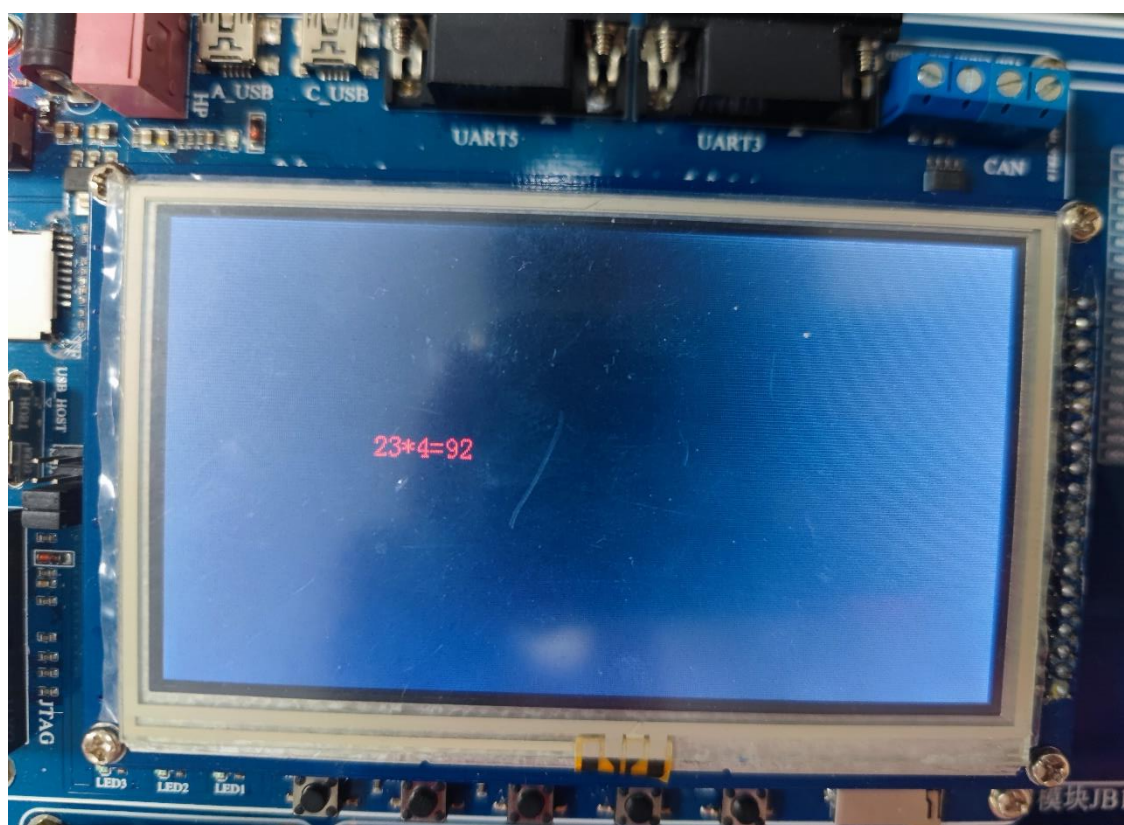
加法运算结果



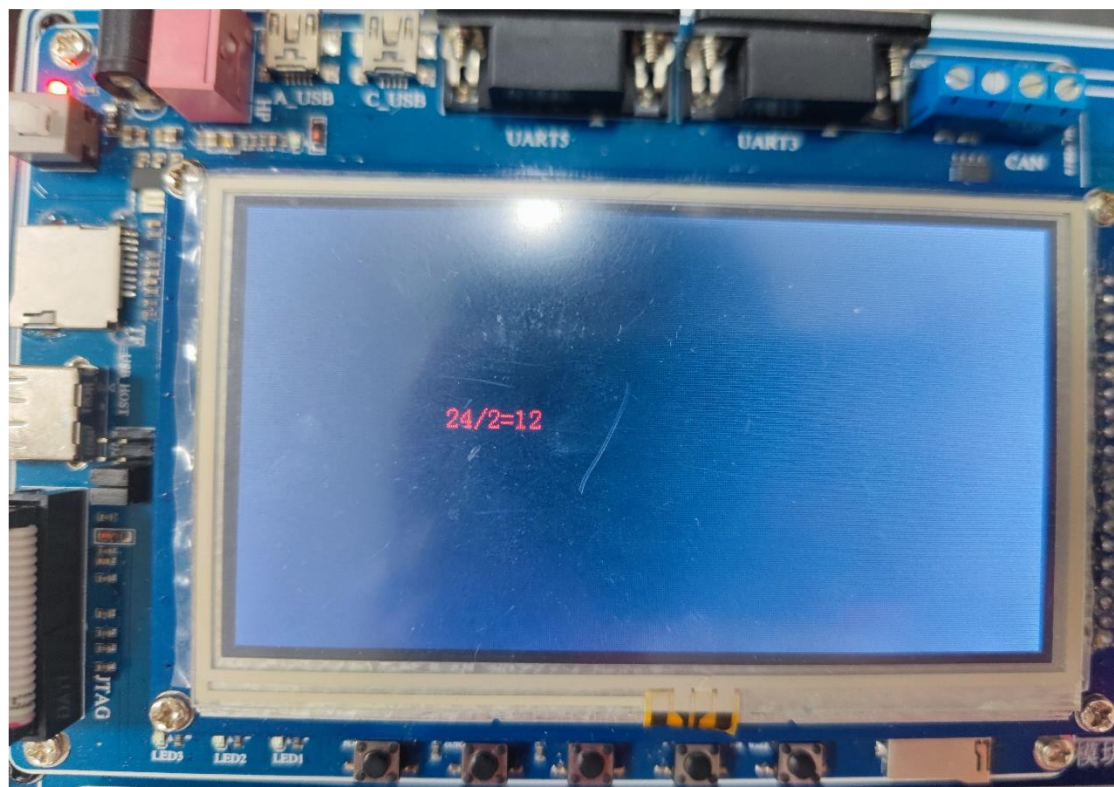
减法运算结果



乘法运算结果:



除法运算结果:



九、实验心得

由于是第一次进行嵌入式开发,在作业初期遇到了一系列的问题,包括工具使用不熟练、代码缺少头文件等,导致前期的进度十分不理想。在解决了示例项目无法跑通的种种问题后,项目进度得到了快速推进,在学习了键盘实验相关内容后,我们想利用键盘来实现一个计算器,因此进一步去了解键盘的控制 API,最终完成二元计算器的设计和编码。

在这次的作业中,我们认识到开发工作需要积极的心态和锲而不舍的精神,在遇到未知的错误时要积极去解决而不是放着,这样才能更好的完成工作。