

深圳大学实验报告

课程名称： 微型计算机技术

实验项目名称： 汇编语言实验——按键以及散转语句

学院： 医学院

专业： 生物医学工程

指导教师： 徐海华、刘昕宇

报告人： 陈焕鑫 学号： 2016222042 班级： 生工2班

实验时间： 2018-10-16

实验报告提交时间： 2018-10-30

教务处制

一、实验目的

1. 初步使用汇编语言进行编程，了解单片机内程序运行情况；
2. 了解汇编语言的 JB,JNB,ADD,RL,散转语句的用法；
3. 使用汇编语言实现按键判断功能，并且使用散转语句，实现 LED 状态的变化

二、实验仪器

微机原理实验板

三、实验内容

- 1、使用汇编语言编程，实现如下功能：
 - 1) 首先让 LED1~4 全灭；
 - 2) 判断 KEY1 按键是否按下，并且每一次按下，让 LED 转换一种状态，至少 5 种状态；（比如，开始的状态是灭灭灭灭，按下 KEY1 一次之后，变为亮灭灭灭，在按一次变为亮亮灭灭...最后一个状态之后，再次重复此循环）。
- 2、使用 C 语言编写 1 中的程序，体会 C 语言和汇编语言编程的相同和不同之处。

四、实验原理

从实验平台的原理图中我们可以看到 LED 的负极连接着芯片的引脚为 P2.4、P2.5、P2.6、P2.7，LED 正极连接着高电平，如图 1 所示。所以，当我们使用引脚 P2.4、P2.5、P2.6、P2.7 输出高电平时，由于 LED 的正极也是连接着高电平的，LED 的两端没有电势差，没有电流流过 LED 灯，LED 灯不会被点亮；当我们使用引脚 P2.4、P2.5、P2.6、P2.7 输出低电平时，LED 的负极电势低于 LED 正极的电势，会有电流流过 LED 灯，所以 LED 灯会被点亮。因此，我们就可以通过控制引脚输出的电平值来控制 LED 灯的亮灭。

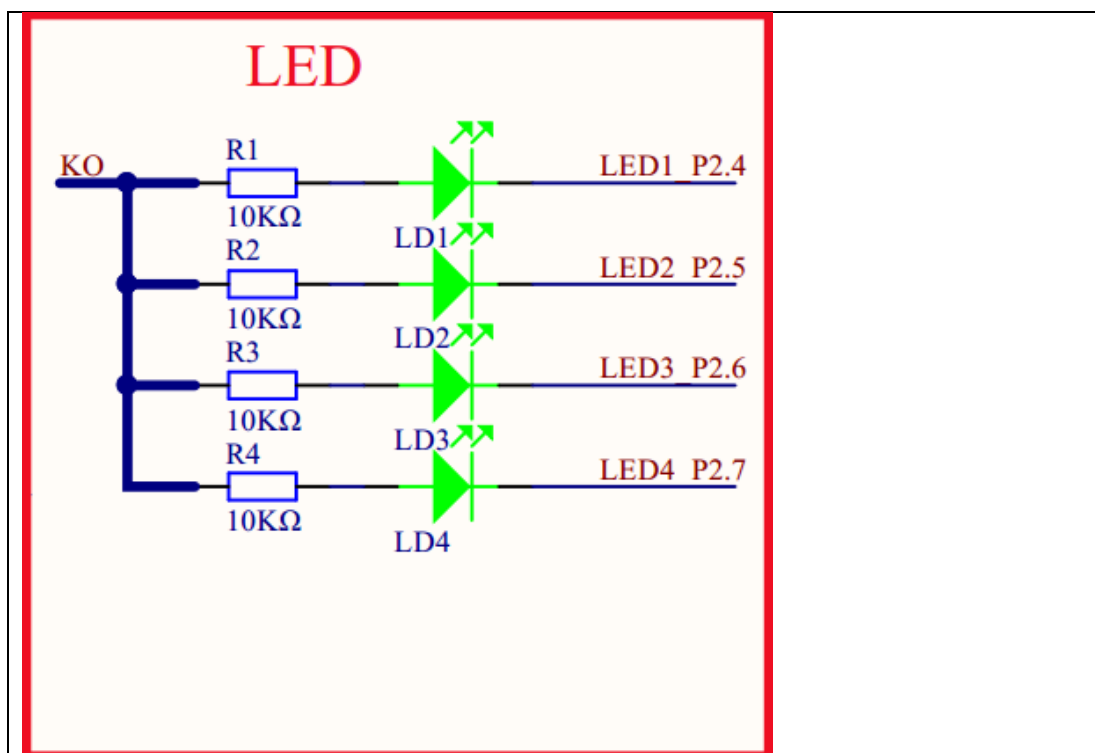


图 1

五、实验方法及步骤

首先，打开 Keil 软件，新建工程名为 Lab1Prj，在工程中添加空的 main.asm 汇编文件和默认的 STC12C5A60S2.h 头文件。

然后，对 main.asm 文件进行修改。结合原理图，我们对按键和四个 LED 灯的引脚进行定义。

```
LED1 BIT P2.4    ;将 P2.4 用伪指令命名为 LED1
LED2 BIT P2.5    ;将 P2.5 用伪指令命名为 LED2
LED3 BIT P2.6    ;将 P2.6 用伪指令命名为 LED3
LED4 BIT P2.7    ;将 P2.7 用伪指令命名为 LED4
```

```
KEY1 BIT P3.2    ;将 P3.2 用伪指令命名为 KEY1
```

然后就是对一些重要的地址进行命名操作，比如 DelayTimes_2MS、KEY1_State。有了这些定义之后我们就可以方便的对数据进行操作了。进入主程序的编写，首先，因为堆栈指针 SP 默认是指在地方的，所以在初始化时要把 SP 指向 70H 的地址，将四个 LED 设置为熄灭。调用按键扫描函数，在按键扫描函数中，每次按键按下，先延时 2ms，实现按键去抖，然后就会判断按键是否再次弹起，如果按键未弹起，则循环等待按键弹起，弹起后改变 KEY1_State 状态，调用 LEDRUN 函数，LEDRUN 函数中使用散转语句实现了 LED 灯的状态变化选择。如此可以实现全灭->A->AB->ABC->ABCD 的跑马灯效果。

具体的代码如下所示:

```
LED1 BIT P2.4    ;将 P2.4 用伪指令命名为 LED1
LED2 BIT P2.5    ;将 P2.5 用伪指令命名为 LED2
LED3 BIT P2.6    ;将 P2.6 用伪指令命名为 LED3
LED4 BIT P2.7    ;将 P2.7 用伪指令命名为 LED4

KEY1 BIT P3.2    ;将 P3.2 用伪指令命名为 KEY1

KEY1_State EQU 69H    ;将地址 69H 用伪指令命名为 KEY1_State

DelayTimes_2MS EQU 6FH    ;将地址 6FH 用伪指令命名为 DelayTimes_100MS

;*****
;--主程序开始--
ORG 0000H    ;定义起始地址, 程序将从此处开始运行
LJMP START    ;长跳转, 跳转到标号"START"处

ORG 0100H    ;定义 START 程序地址
START:
    MOV SP, #70H    ;设置堆栈, 堆栈空间从 70H 到 7FH, 共 16 个字节
    SETB LED1    ;熄灭 LED1
    SETB LED2    ;熄灭 LED2
    SETB LED3    ;熄灭 LED3
    SETB LED4    ;熄灭 LED4

    LJMP START1    ;长跳转, 跳转到标号"START1"处

START1:
    LCALL SCANKEY    ;调用 SCANKEY 函数

    LJMP START1    ;长跳转, 跳转到标号"START1"处

;*****
;以下为延时函数
;
Delay2Ms:
    MOV R7, DelayTimes_2MS

    Delay2Ms_1: MOV R6, #30
    Delay2Ms_2: MOV R5, #200
    Delay2Ms_3: DJNZ R5, Delay2Ms_3
                 DJNZ R6, Delay2Ms_2
                 DJNZ R7, Delay2Ms_1
```

```

RET ;子函数返回

;-----
;扫描按键函数
;-----
SCANKEY:
    JB KEY1, UP
    DOWN:
        MOV DelayTimes_2MS, #1
        LCALL Delay2Ms ;延时 2ms, 按键去抖
        JB KEY1, PRESS
        JMP DOWN
    PRESS:
        MOV A, KEY1_State
        ADD A, #1
        MOV KEY1_State, A
        CLR C
        SUBB A, #4
        JB CY, NORESET
        MOV KEY1_State, #0
    NORESET:
        MOV A, #0
        LCALL LEDRUN ;调用 LEDRUN 函数
    UP:
        RET

LEDRUN:
    MOV A, KEY1_State ;散转语句
    RL A
    ADD A, KEY1_State
    MOV DPTR, #KEY1_Flash
    JMP @A+DPTR
KEY1_Flash:
    LJMP KEY1_Flash0 ;长跳转, 跳转到标号"KEY1_Flash0"处
    LJMP KEY1_Flash1 ;长跳转, 跳转到标号"KEY1_Flash1"处
    LJMP KEY1_Flash2 ;长跳转, 跳转到标号"KEY1_Flash2"处
    LJMP KEY1_Flash3 ;长跳转, 跳转到标号"KEY1_Flash3"处
    LJMP KEY1_Flash4 ;长跳转, 跳转到标号"KEY1_Flash4"处
    RET

KEY1_Flash0:
    CLR LED1 ;LED1 灯全亮
    CLR LED2
    CLR LED3

```

```

        CLR LED4
        RET
KEY1_Flash1:
        SETB LED1    ;LED1 灭
        CLR LED2     ;LED2 亮
        CLR LED3     ;LED3 亮
        CLR LED4     ;LED4 亮
        RET
KEY1_Flash2:
        SETB LED1    ;LED1 灭
        SETB LED2     ;LED2 灭
        CLR LED3     ;LED3 亮
        CLR LED4     ;LED4 亮
        RET
KEY1_Flash3:
        SETB LED1    ;LED1 灭
        SETB LED2     ;LED2 灭
        SETB LED3     ;LED3 灭
        CLR LED4     ;LED4 亮
        RET
KEY1_Flash4:
        SETB LED1    ;LED1 灯全灭
        SETB LED2
        SETB LED3
        SETB LED4
        RET
END

```

使用 C 语言编写的程序如下所示:

```

#include <STC12C5A60S2.h>
sbit LED1 = P2^4;
sbit LED2 = P2^5;
sbit LED3 = P2^6;
sbit LED4 = P2^7;

void main()
{
    unsigned long int i; //用于延时
    unsigned int j;      //用于计数状态

    while(1)             //死循环
    {
        for(i = 0; i < 50000; i++); //延时
        switch(j)         //switch 语句改变状态

```

```

{
    case 0:
        LED1 = 1; //LED 灯全灭
        LED2 = 1;
        LED3 = 1;
        LED4 = 1;
        break;
    case 1:
        LED1 = 0;
        LED2 = 1;
        LED3 = 1;
        LED4 = 1;
        break;
    case 2:
        LED1 = 0;
        LED2 = 0;
        LED3 = 1;
        LED4 = 1;
        break;
    case 3:
        LED1 = 0;
        LED2 = 0;
        LED3 = 0;
        LED4 = 1;
        break;
    case 4:
        LED1 = 0; //LED 灯全亮
        LED2 = 0;
        LED3 = 0;
        LED4 = 0;
        break;
}
if(j >= 4)
    j = 0;
else
    j++;
}
}

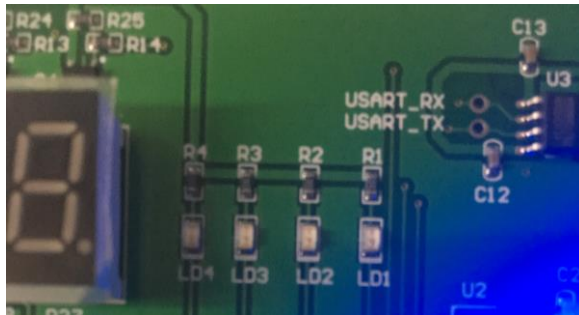
```

两种语言代码实现的功能相同，但是 C 语言的代码相对更易于理解编写简单，而汇编语言比较难懂，编写比较复杂

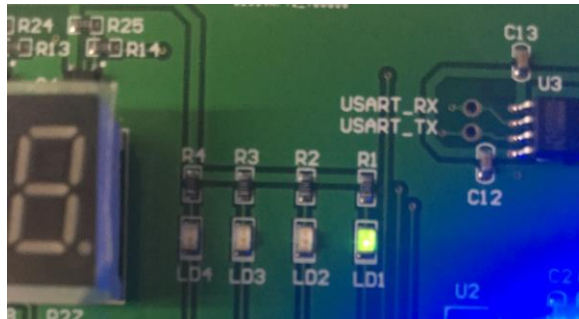
检查代码无误之后，编译、链接、生成 Hex 文件，将两种代码的 Hex 文件分别通过串口烧进实验平台中，观察实验现象，可以观察到实验得到的现象相同。

六、实验现象

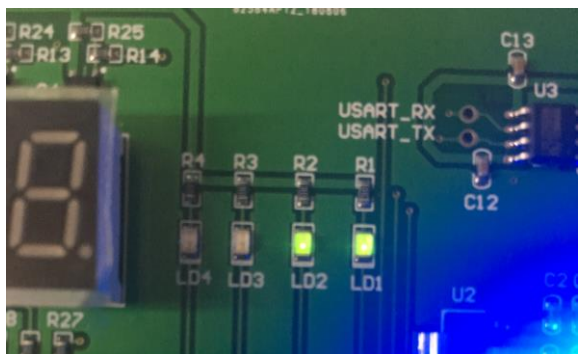
状态 1：全灭



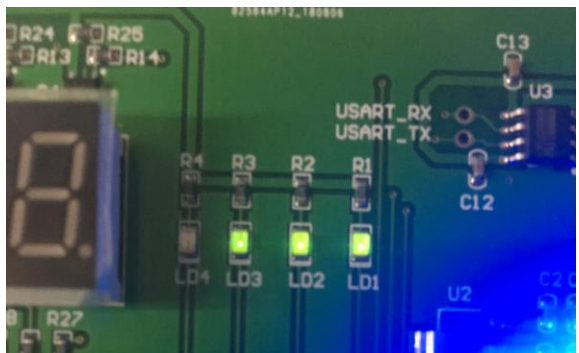
状态 2



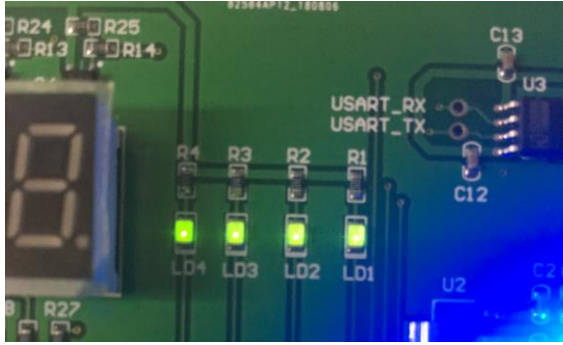
状态 3:



状态 4:



状态 5：全亮



将程序烧进实验平台后，可以看到每按下一次按键 LED 灯的状态就变化一次，呈全灭->A->AB->ABC->ABCD 的流水状态，表明程序正确。

七、实验结论

通过这次实验，我学会了使用汇编语言来对单片机进行编程，汇编语言使我更能理解单片机的工作原理，程序运行情况。同时，我学会了使用 JB、JNB、ADD、RL 和散转语句实现需要的功能，并用汇编语言实现了按键的判断功能，使用散转语句实现了 LED 状态的变化。

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整 and 补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。