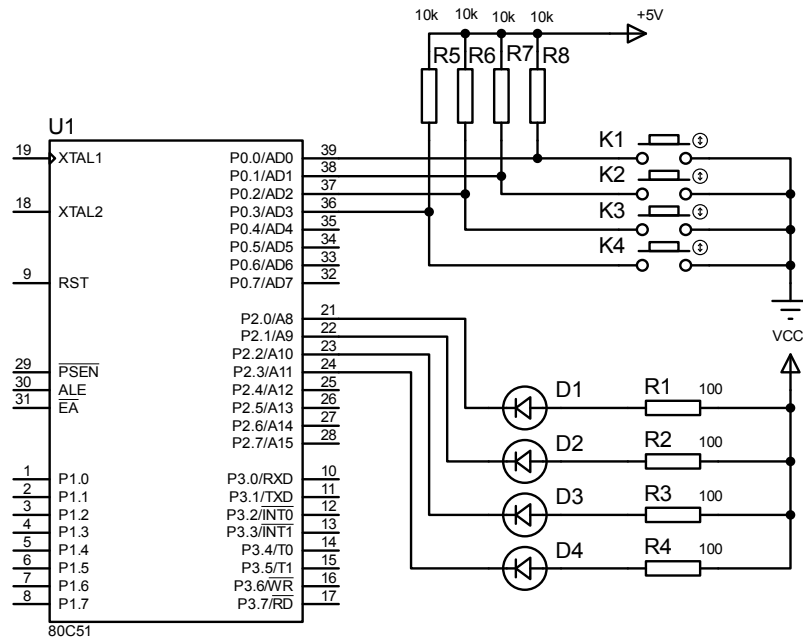


实例 1 独立按键识别

【要求】采用独立按键方式实现下述功能：开机时 LED 全熄，然后根据按键动作使相应灯亮，并将亮灯状态保持到按压其它键时为止。

独立按键——每个按键都彼此独立地各占有一位 I/O 口线。特点是电路简单，但占用 I/O 口线较多。

【分析】为使按键抬起后 LED 能保持先前的点亮状态，需要在按键未压下期间禁止向 P2 输出 P0 状态值。



```
void main( ) {  
    char key = 0;           //定义按键变量  
    while(1){  
        key = P0 & 0x0f;    //读取按键状态,高 4 位清零  
        if (key != 0x0f) P2 = key; //有按键动作时，P0 状态值送 P2  
    }  
}
```

【要求】

K2 为“停止键”，按压 K2 可终止流水灯的运动；

```

void main(){
    bit dir=0,run=0;           //标志位/初始化
    char i;
    while(1){
        switch(P0 & 0x0f){      //根据键值修改标志位
            case 0x0e:run=1;break;
            case 0x0d:run=0,dir=0;break;
            case 0x0b:dir=1;break;
            case 0x07:dir=0;break;
        }
        if(run)                  //若 run=dir=1, 自上而下流动
            if(dir)
                for(i=0;i<=3;i++){
                    P2=led[i];
                    delay(200);
                }
            else                  //若 run=1, dir=0, 自下而上流动
                for(i=3;i>=0;i--){
                    P2=led[i];//
                    delay(200);
                }
            else P2=0;           //若 run=0, 灯全灭
    }
}

```

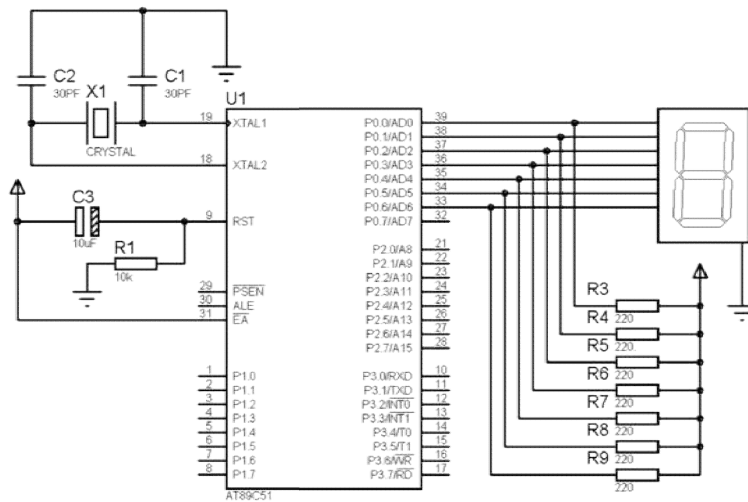
实例 3 混合编程

将实例 2 中 C51 的 delay 函数改用汇编语言实现，并完成系统的混合编程。

```
#include "reg51.h"
unsigned char led[]={0xfe,0xfd,0xfb,0xf7}; //LED 灯的花样数据
void delay(unsigned char time);
void main(){
    bit dir=0,run=0;           //标志位定义及初始化
    char i;
    while(1){
        switch (P0 & 0x0f){    //读取键值
            case 0x0e:run=1;break; //K1 动作，设 run=1
            case 0x0d:run=0,dir=0;break; //K2 动作，设 run=dir=0
            case 0x0b:dir=1;break; //K3 动作，设 dir=1
            case 0x07:dir=0;break; //K4 动作，设 dir=0
        }
        if (run)                //若 run=dir=1, 自上而下流动
            if(dir)
                for(i=0;i<=3;i++){
                    P2=led[i];
                    delay(255);
                }
            else                //若 run=1,dir=0, 自下而上流动
                for(i=3;i>=0;i--){
                    P2=led[i];
                    delay(255);
                }
        else P2=0xff;           //若 run=0, 灯全灭
    }
}
```

实例 4 LED 数码管显示

在 P0 口连接一个共阴极数码管，使之循环显示 0~9 数字，间隔为 500 循环步。



分析：

将显示码循环输出到 P0 口即可实现循环显示。但由于数字 0~9 的显示段码没有规律可循，需要采取查表方式进行操作：

- ①将显示码按序存放在一个数组中，顺序号与代表的显示字符相对应（如，char led_mod[] = {x1, x2, ..., xn}）；
- ②通过查表语句（如，P0=led_mod[i]）输出显示码。

参考程序

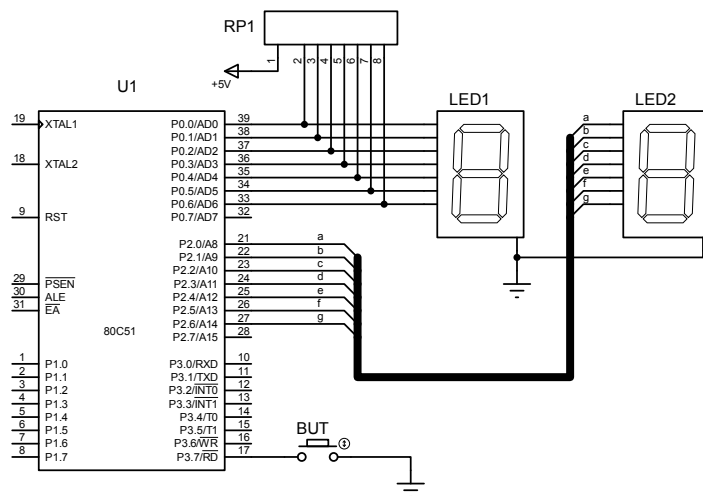
```
#include <reg51.h> //包括一个 51 标准内核的头文件
char led_mod[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f}; //LED 显示字模

void delay(unsigned int time){
    unsigned int j = 0;
    for(;time>0;time--){
        for(j=0;j<125;j++);
    }
}

void main(void) {
    char i = 0;
    while(1){
        for(i=0;i<=9;i++) {
            P0=led_mod[i];
            delay(500);
        }
    }
}
```

实例5 （实验1）

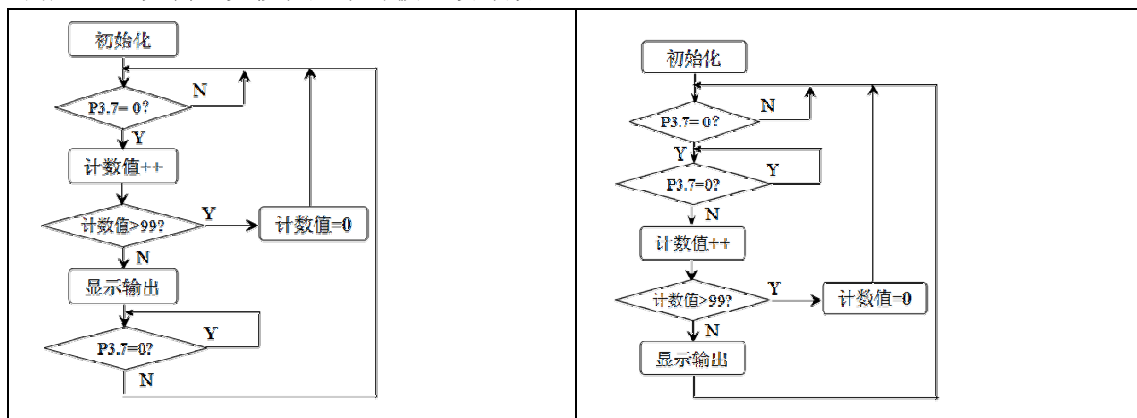
根据如下共阴极型数码管电路，编程实现计数显示功能，即以十进制形式显示击键次数，次数大于99后重新由0开始。



两个新问题：

问题1：如何将计数值拆解成个位和十位两个数？取余数、整除

问题2：如何避免按键压下时被连续计数？



```

01 #include <reg51.h>
02 sbit P3_7=P3^7;
03 unsigned char count =0;
04 unsigned char code table[]=
05     {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f, 0x6f};
06
07 void main(void) {
08     P0=P2=table[0];
09     while(1) {
10         if(P3_7==0) {
11             count++;
12             if(count==100) count=0;
13             P0=table[count/10];
14             P2=table[count%10];
15             while(P3_7==0);
16         }
17     }
18 }

```

//定义计数器端口
 //定义计数器
 //显示初值00
 //检测按键是否压下
 //计数器增1
 //判断循环是否超限
 //十位输出显示
 //个位输出显示
 //等待按键抬起，防止连续计数

实例5.c