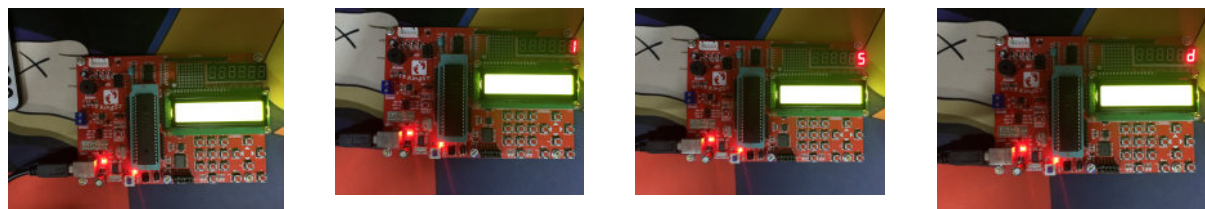
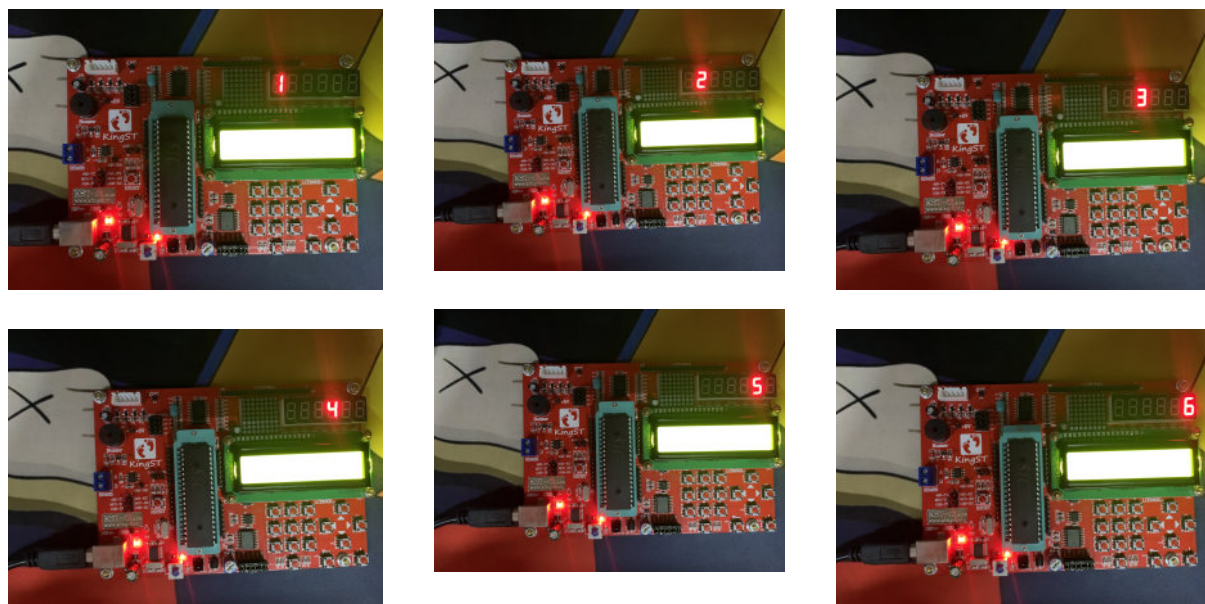


汇编第五次实验报告

(1) 课前完成：运行3.1中的程序，回答3.1中的问题；在上文的程序中，对程序中的每一句进行备注；拍摄一到两张程序运行中的图片作为报告附图。（相关答案请直接手写填写在上文中）



(2) 课后完成：根据3.2中的要求进行编程，简单描述程序的整体结构和各子函数的功能，并对程序的每一句进行简单备注；拍摄一到两张程序运行中的图片作为报告附图。



设计思路：此次实验比较容易，只需在原有程序的基础上加上一个循环数码管的程序即可

各子程序功能：

MAIN为程序循环的主程序，每循环一次显示一次数码管，检测两次按键是否弹起

JUDGE为ROW按键检测函数，即检测ROW按键是否有按下

KEY为获取键值函数，即通过P2口读取按下的按键的键值

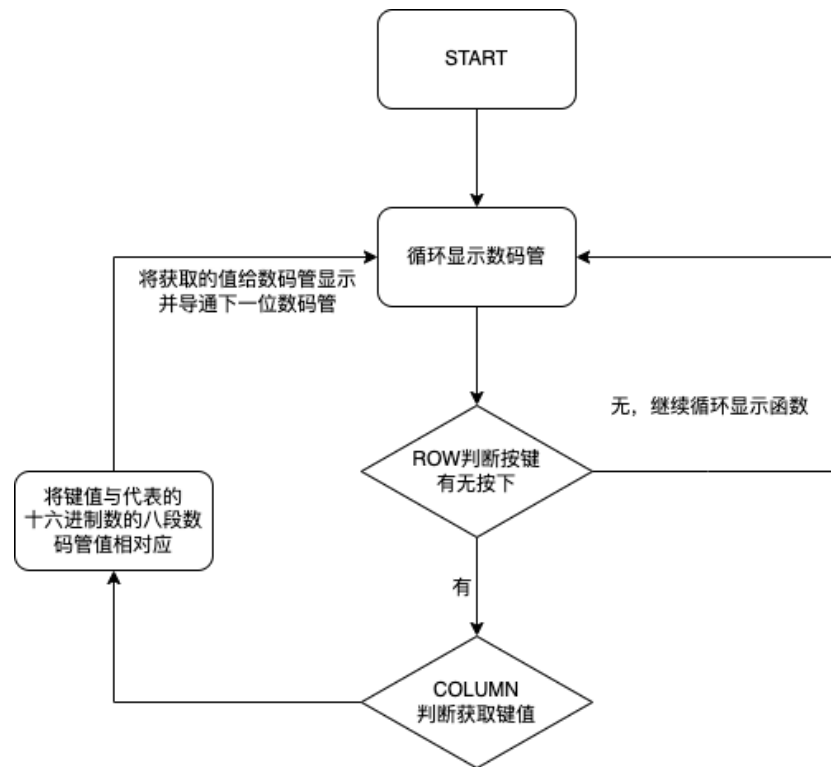
NUM为获取键值对应的16进制数的八段数码管值函数

DELAY为10ms延时函数

TAB1存放的为16个按键存储的键值

TAB2存放的为各个键值对应的16进制数的八段数码管值

流程图：



具体程序代码：

```

ORG 0000H
    LJMP START
ORG 0050H
START:  MOV SP, #60H    设置堆栈指针
        MOV R5, #16    将16作为R5的初始值，灭灯
        MOV R3, #0EEH  该值减一为数码管LED9的导通值
        MOV 40H, #00H  40H用于存储按钮的行列值。第0~3位代表行
                        KeyOut4~1, , 第4~7位代表列KeyIn1~4
MAIN:   MOV A, R5       将R5的值给A, 用于查表
        MOV DPTR, #TAB2 将TAB2的起始地址给DPTR
        MOVC A, @A+DPTR 查表
        MOV P0, A       数码管显示A的值
        MOV P1, R3       导通数码管
        LCALL DELAY      延时10ms
        ACALL JUDGE      进入JUDGE, 判断ROW有无按键按下, 返回一个A值
        JZ MAIN          如果A为0就循环MAIN自己
        LCALL DELAY      延时10ms
        ACALL JUDGE      再进入JUDGE, 检查没有误判, 判断ROW有无按键按下, 返回一个A值
        JZ MAIN          如果A为0就循环MAIN自己
        ACALL KEY        跳至KEY, 获取按下的按键对应的键值
        ACALL NUM        跳至NUM, 将获取的键值与对应的16进制数的八段数码值相对应
        LJMP MAIN        循环MAIN
JUDGE:  MOV P2, #0FH     将00001111给P2, 允许P2低四位读取外部输入
        MOV A, P2        将读取到的p2给A
        CLR C            清零进位位
        SUBB A, #0FH     A与00001111相减并赋给A, 判断ROW是否有按键按下
        RET
KEY:    PUSH ACC         存储ACC
K_LINE: MOV P2, #0FH     将00001111给P2, 允许P2低四位读取外部输入
        MOV A, P2        将低四位存入R1中
        MOV R1, A
K_ROW:  MOV P2, #0F0H    将11110000给P2, 允许P2高四位读取外部输入
        MOV A, P2        将高四位存入R0中
        MOV R0, A
K_VALUE: MOV A, R1       将R1与00001111做与运算, 得到Row的键值
        ANL A, #0FH
  
```

	MOV R1, A	将键值还给R1
	MOV A, R0	将R0与11110000做与运算，得到Column的键值
	ANL A, #0F0H	
	ORL A, R1	将Row的键值与Column的键值做或运算，得到完整的键值
	MOV 40H, A	将键值存入地址40H中
	DEC R3	将数码管选通值减一，即数码管右移
	CJNE R3, #0E7H, RET1	判断数码管显示是否到达最右端LED2， 如到达将其初始化为LED9
RET1:	MOV R3, #0EDH	初始化为LED9
	POP ACC	取出ACC
	RET	返回
NUM:	PUSH DPL	存储数据指针低八位
	PUSH DPH	存储数据指针高八位
	PUSH ACC	存储ACC
	MOV DPTR, #TAB1	将表1的地址初始值给数据指针
	MOV R5, #0FFH	0FFH+1为00H，用于查表
NUM0:	INC R5	查下一位
	MOV A, R5	将R5的值给A，查表
	MOVC A, @A+DPTR	将查到的值给A
	CLR C	清零进位位
	SUBB A, 40H	将键值与A做比较
	JNZ NUM0	如果不相等就循环，直到找到与表中对应的值为止
NUM1:	LCALL DELAY	10ms延时
	MOV P2, #0FH	检查按键是否弹起
	MOV A, P2	将00001111给p2，允许读取外部信号
	CJNE A, #0FH, NUM1	与00001111比较，确保按键弹起
	POP ACC	取出ACC
	POP DPH	取出数据指针高八位
	POP DPL	取出数据指针低八位
	RET	返回
DELAY:	MOV R7, #20	10ms延时函数
DEL1:	MOV R6, #229	
DEL2:	DJNZ R6, DEL2	
	DJNZ R7, DEL1	
	RET	
TAB1:	DB 0EEH, 0E7H, 0D7H, 0B7H, 0EBH, 0DBH, 0BBH, 0EDH, 0DDH, 0BDH, 77H, 7BH, 7EH, 7DH, 0DEH, 0BEH	
TAB2:	DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H, 80H, 90H, 88H, 83H, 0C6H, 0A1H, 86H, 8EH, 0FFH	
	END	