

计算机工程学院

**汇编语言与接口技术**

**实验报告**

学年学期：  **2022 - 2023 学年 第1学期**

实验名称：  **实验六 电子计数器**

班级： 计算2114

学号： 202121331104

姓名： 庄佳强

日期： 11.20

成绩：

**一、 实验要求**

1. 掌握数码管显示方法
2. 掌握软件延时方法
3. 掌握键盘扫描及去抖动方法

**二、 实验设计**

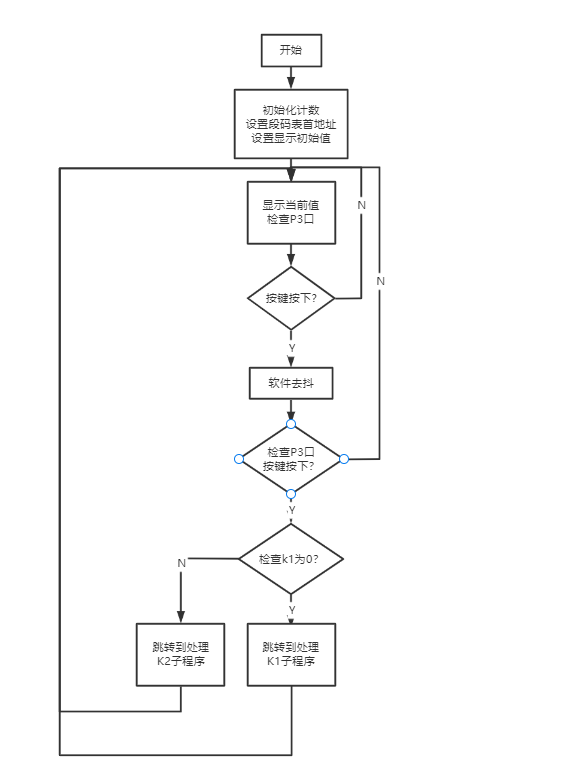
**1.整体思路**

先参考开发板原理图，得知控制按钮的I/O口位P3,控制LED数码管为P2,P0。

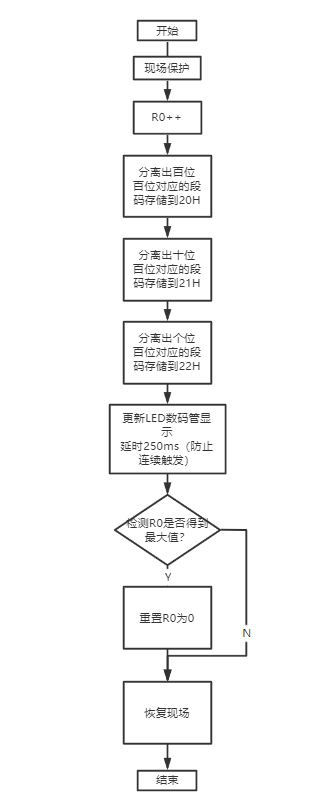
再设计一个循环电路用于检查P3口的电位变化，当P3.1和P3.0口点位变化时，跳转到各自的处理子程序中。在子程序中实现加1和减1的操作后把个位十位百位分离后返回到存储中，在延时后返回到开始重新检查，这是如果还按着按钮就会被再次检测重复操作，实现连加或连减的操作。

**2.流程图**

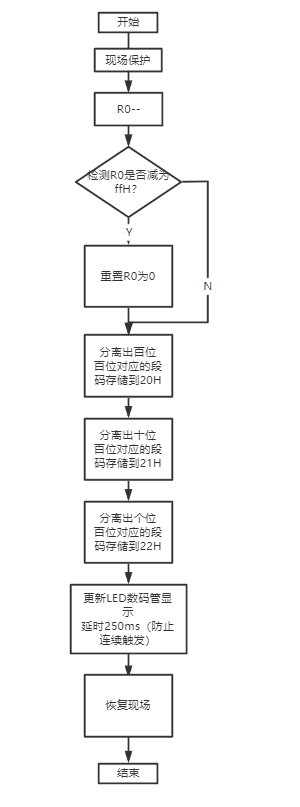
主程序：



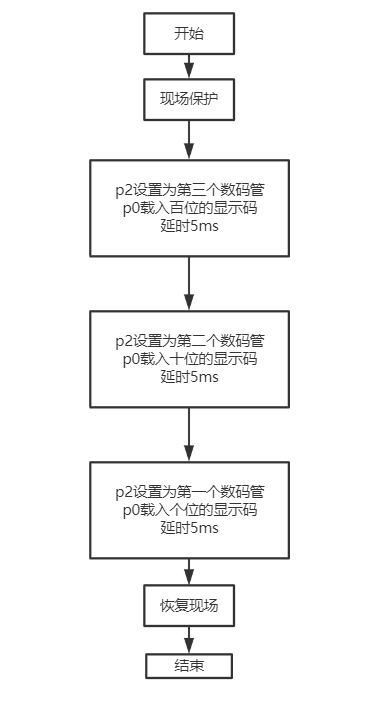
加一子程序：



减一子程序：



显示子程序：



**3.主要模块设计思路及分析**

1. 主模块

一开始对R0(计数值),DPTR,22H,21H,20H进行初始化。接着先进行数码管的动态显示，接着反复检查P3口的电位值，当没有按钮时返回开始循环进行，当有电位变化后，先软件去抖，再检查一次电位变化，确保真实的按钮按下。接着就是检测按个电位变化，跳到对应的子程序就行就下，最后返回开始重新检测，如果这时还按着，就实现了连续的效果。

1. 加/减子程序

先进行现场保护。再对R0进行加一或减一操作。完成后，就把R0除100获得当前的百位，再对照@DPTR+A 获得对应数字的段码，存储到20H中。同理把除剩下的数再除10获得十位和个位，同样的方法后存储到对应的21H和22H中，通过延迟250ms,实现冷却的效果，防止按一下连续加减多次。最后恢复现场，结束。

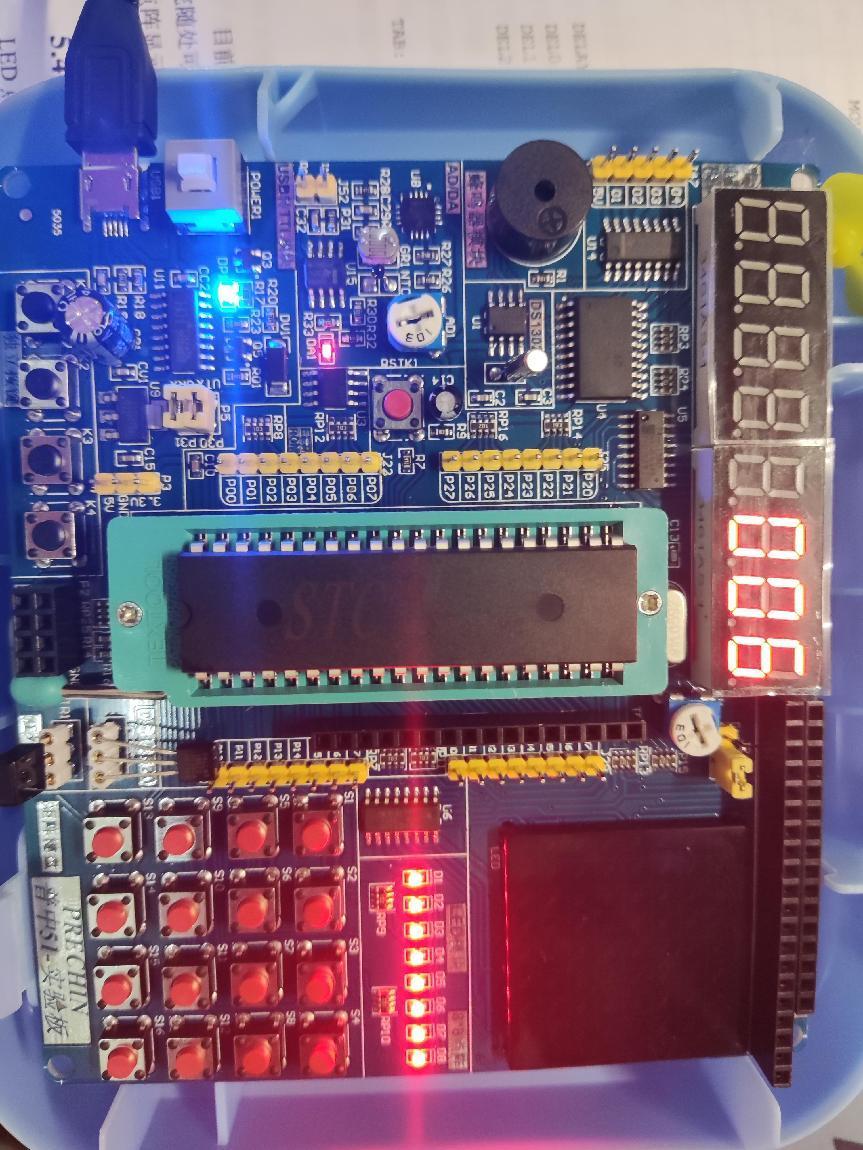
PS:在延迟中，一个加上动态显示LED的子程序，避免在250ms延迟中，数码管中的数字消失。

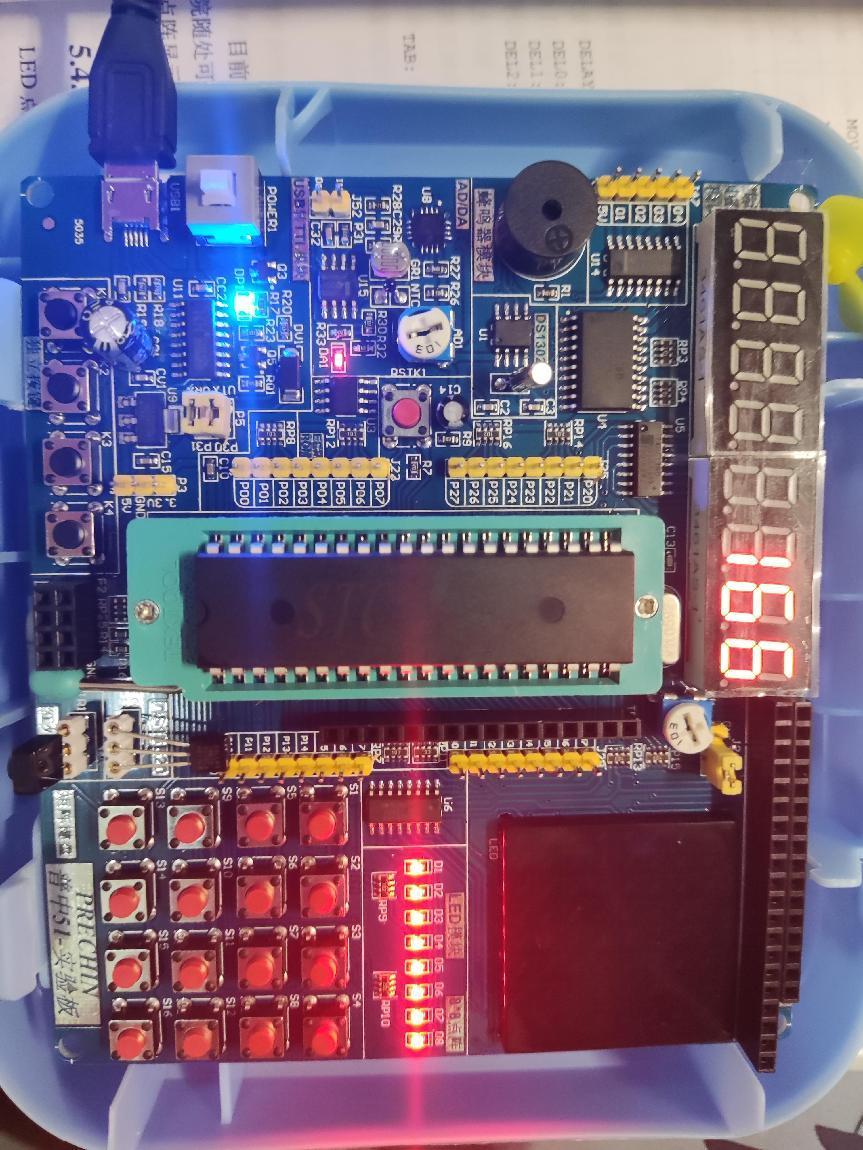
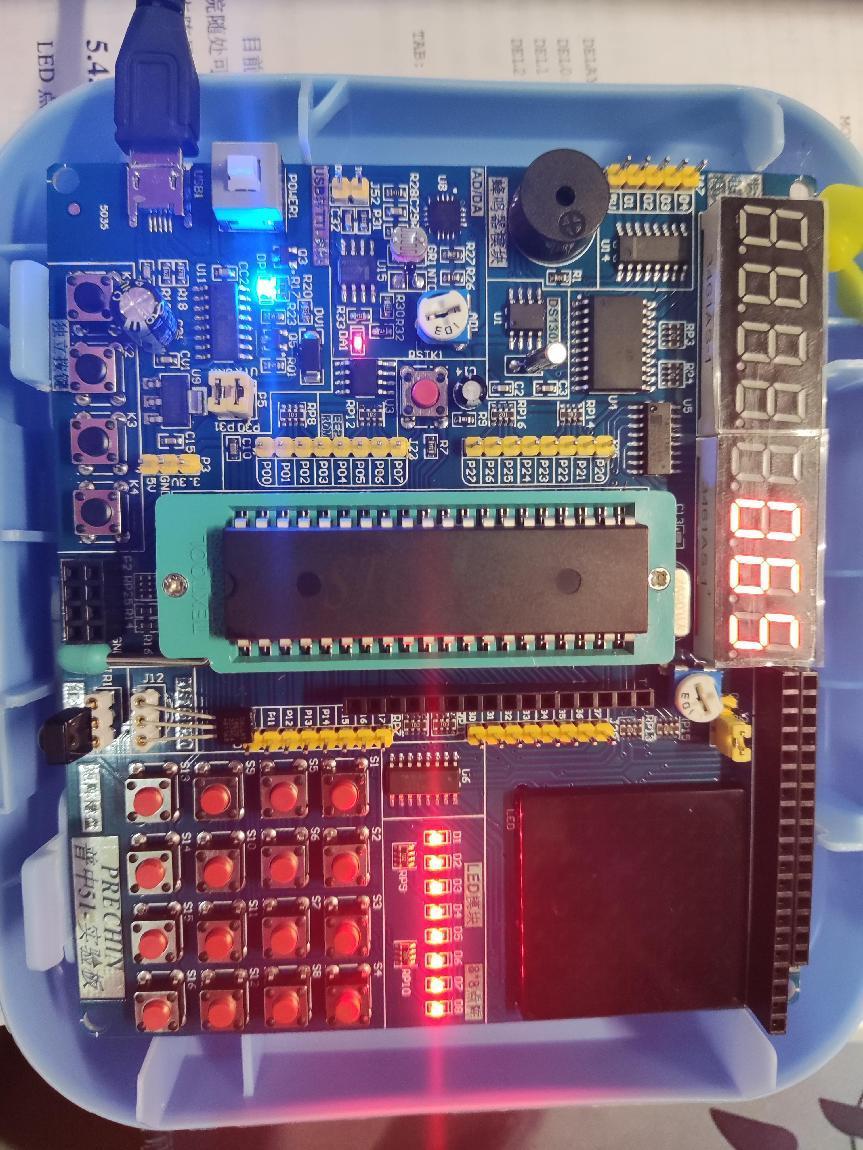
1. 显示子程序：

先进行现场保护。再设置P2值为00001000B对应第三个LED管，再设置P0为20H对应百位，这样第三个LED管就可以显示百位的值，再延迟5ms，达到存在的效果。同理设置P2值为00000100B对应第二个LED管，再设置P0为21H对应十位，这样第二个LED管就可以显示十位的值，再延迟5ms，达到存在的效果。最后个位同理。

PS:延时5ms是反复测试后的结果，这个延时是没有频闪和较稳定的时间。00000001B和00000010B点亮都是第一个LED数码管。

**三、 实现效果**





1. **总结**

1.一开始没有看原理图，使用的是静态的方法，在测试的时候发现无法运行，才知道只能使用动态的方法。

2.开始按按钮没有反应，通过分开测试发现不是按钮读取相应的问题，而是读取后无法显示。单独实验显示数码管后发现问题是显示重复。最后解决。

**附录：**

ORG 0000H

AJMP KEY1

ORG 0100H

KEY1:

MOV R0,#00H //初始化 计数

MOV DPTR,#TABLE //设置段码表首地址

MOV R1,#10

MOV R2,#100

MOV 22H,#3FH //初始化显示值

MOV 21H,#3FH

MOV 20H,#3FH

KEY2:

LCALL Change //显示当前值

MOV P3,#0FFH //p3口写入全1 设置p3口为输入状态

MOV A,P3 //按下的值为0

CPL A //取反，按下值为1

JZ KEY2 //无按键按下，跳转重新开始

LCAll D10ms //非0 ，延时10ms,软件去抖

MOV A,P3 //重读一次按键状态

CPL A

JZ KEY2

JB ACC.1,PK1 //k1闭合，跳转到k1键的处理符号pk1

JB ACC.0,PK2 //k2闭合，跳转到k2键的处理符号pk2

LJMP KEY2

PK1:

LCALL pkey1 //跳转k1键的处理子程序

LJMP KEY2 //键处理完成后，返回主程序

PK2:

LCALL pkey2

LJMP KEY2

pkey1: //加一

PUSH PSW //现场保护

PUSH Acc

INC R0 //加一

MOV A,R0

MOV B,R2

DIV AB //分离百位

MOVC A,@A+DPTR

MOV 20H,A //显示百位

MOV A,B

MOV B,R1

DIV AB //分离十位和个位

MOVC A,@A+DPTR

MOV 21H,A //显示十位

MOV A,B

MOVC A,@A+DPTR

MOV 22H,A //显示个位

LCALL Change //立刻刷新一次显示状态，减少延迟

LCALL D250ms //延迟250ms,防止过快

CJNE R0,#255,pkey11 //未到达最高值255时跳转 2位十六进制数的最高

MOV R0,#00H //重置计数值

pkey11:

POP Acc

POP PSW

RET

pkey2: //减1

PUSH PSW

PUSH Acc

DEC R0

CJNE R0,#255,pkey22 //当减到负值时，重置成0

MOV R0,#00H

pkey22:

MOV A,R0

MOV B,R2

DIV AB

MOVC A,@A+DPTR

MOV 20H,A //显示百位

MOV A,B

MOV B,R1

DIV AB

MOVC A,@A+DPTR

MOV 21H,A //显示十位

MOV A,B

MOVC A,@A+DPTR

MOV 22H,A //显示个位

LCALL Change

LCALL D250ms

POP Acc

POP PSW

RET

Change:

PUSH PSW

PUSH Acc

MOV p2,#00001000B //第三个数码管

MOV p0,20H //输入百位到显示位中

LCALL D5ms //延迟5ms （反复尝试后，5ms效果最好）

MOV p2,#00000100B //第二个数码管

MOV p0,21H //输入十位到显示位中

LCALL D5ms

MOV p2,#00000010B //第一个数码管

MOV p0,22H //输入个位到显示位中

LCALL D5ms

POP Acc

POP PSW

RET

D10ms: //延时10ms

PUSH PSW

PUSH Acc

MOV R7,#25

D1:

MOV R6,#200 //200\*2\*25=10000纳秒=10ms

DJNZ R6,$

DJNZ R7,D1

POP Acc

POP PSW

RET

D250ms: //延时250ms 使用3次循环

PUSH PSW

PUSH Acc

MOV R5,#12

D2:

MOV R6,#10

D3:

MOV R7,#248

D4:

DJNZ R7,D4

DJNZ R6,D3 //((248\*2+4)\*10+15000)\*12约250000ms=0.25s

LCAll change //在延迟中反复刷新显示，防止数码管的显示消失

DJNZ R5,D2

POP Acc

POP PSW

RET

D5ms:

PUSH PSW

PUSH Acc

MOV R7,#25

D1x:

MOV R6,#100 //100\*2\*25=5000纳秒=5ms

DJNZ R6,$

DJNZ R7,D1x

POP Acc

POP PSW

RET

TABLE: //共阳级数码管0~9段码表

DB 3FH,06H,5BH,4FH,66H

DB 6DH,7DH,07H,7FH,6FH

END