## 任务简介

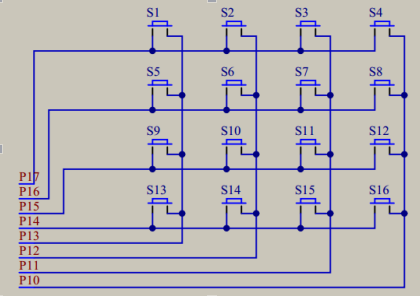
电话键盘及拨号的模拟  
设计要求：设计一个模拟电话拨号的显示装置，即把电话键盘中拨出的某一电话号码，显示在LCD显示屏或LED数码管上。除了0~9的10个数字键外，还有“\*”和“#”，“\*”实现删除功能，即删除最后输入的一位号码，“#”用于清除显示屏上所有的数字显示。要求每按下一个键都发出声响。

## 具体功能

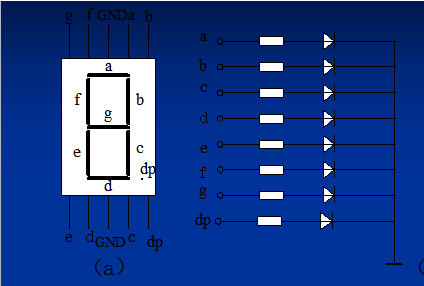
1. 用自己单片机上面的矩阵按键实现电话号码输入（0到9）。
2. 10号按键功能设置为一个一个删除电话号码。
3. 11号按键设置为全部清除电话号码。
4. 每要求按下一个键都发出声音，即在判断键有没有按下的keydown函数里面调用一个定时200ms的延时函数，在这个函数里面对P1.5口不断取反，蜂鸣器就可以发出声音。

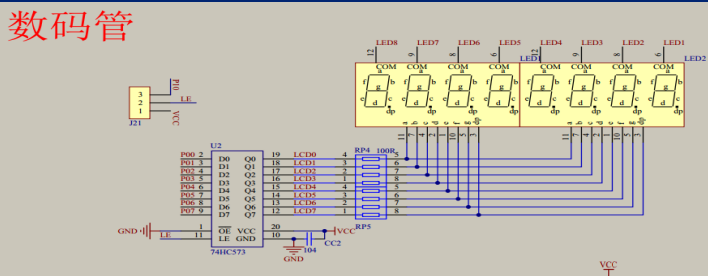
## 实验原理图

## 矩阵按键：

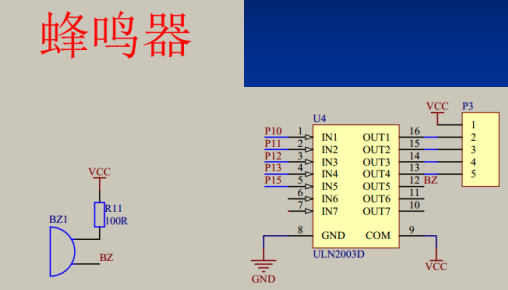


## 数码管：





## 蜂鸣器：



## 控制程序设计

1. 按键0到9任务：

编写一个keydown函数，判断按键有没有按下。

keydown: //判断按键

mov P1,#0FH

mov A,P1

CJNE A,#0FH,LOOP0 //检查有无按键按下，

RET

LOOP0: LCALL DELAY10MS //延时10ms消抖

mov A,P1

CJNE A,#0FH,LOOP //再次检查有无按键按下

LOOP: mov R2,#01H

mov P1,#0FH //有按键按下

mov A,P1

上面是检测有没有按键按下，并且做10ms延时消抖处理

在keydown函数里面用行列检测法判断哪一个按下

CJNE A,#07H,LOOP1

mov R3,#0 ;R3为按键值

SJMP LOOP4

LOOP1: CJNE A,#0BH,LOOP2 ;获得列值

mov R3,#1

SJMP LOOP4

LOOP2: CJNE A,#0DH,LOOP3

mov R3,#2

SJMP LOOP4

LOOP3: CJNE A,#0EH,LOOP4

mov R3,#3

SJMP LOOP4

具体过程见后面贴出的源代码keydown函数

1. 按键一个一个删除部分：

CJNE R3,#10,LOOP29 //将10号按键设为一个一个删除

LCALL deleteone //如果按了10就调用deleteone函数

SJMP LOOP39

1. 全部删除按键设置：

LOOP40: CJNE R3,#11,LOOP39

LCALL deleteall //11号按键设为删除全部

SJMP LOOP39

deleteall: //删除全部就是不让数码管亮

mov 40H,#00H //全部覆盖掉

mov 41H,#00H

mov 42H,#00H

mov 43H,#00H

mov 44H,#00H

mov 45H,#00H

mov 46H,#00H

mov 47H,#00H

RET

1. 蜂鸣器部分：

DELAY200MS: ;误差 0us

MOV R7,#04H

DL11: MOV R6,#74H

DL01: MOV R5,#0D6H

DJNZ R5,$

CPL P1.5 //延时过程蜂鸣器P1.5口不断取反

DJNZ R6,DL01

DJNZ R7,DL11

NOP

RET

按键按下就调用这个函数

## 调试分析

1. 开始数码管并不会移位显示每一次都覆盖了，只是在第一个数码管显示后来就用了数码管移位这种方法。

mov 47H,46H //进行移位

mov 46H,45H

mov 45H,44H

mov 44H,43H

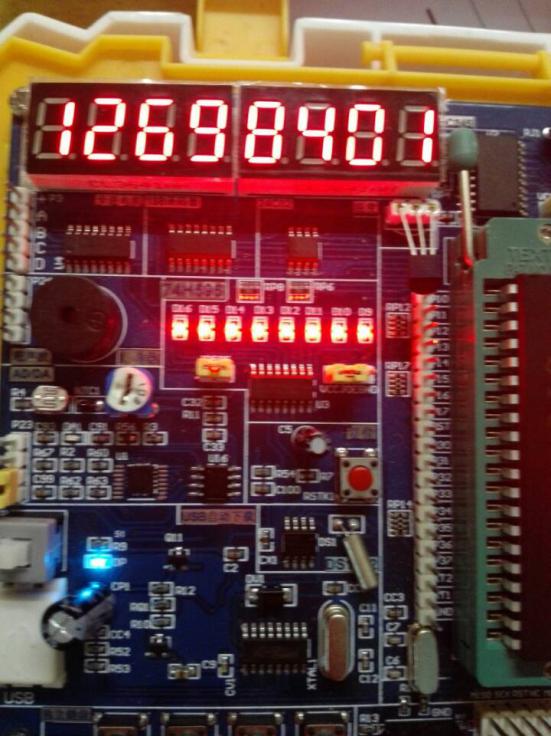
mov 43H,42H

mov 42H,41H

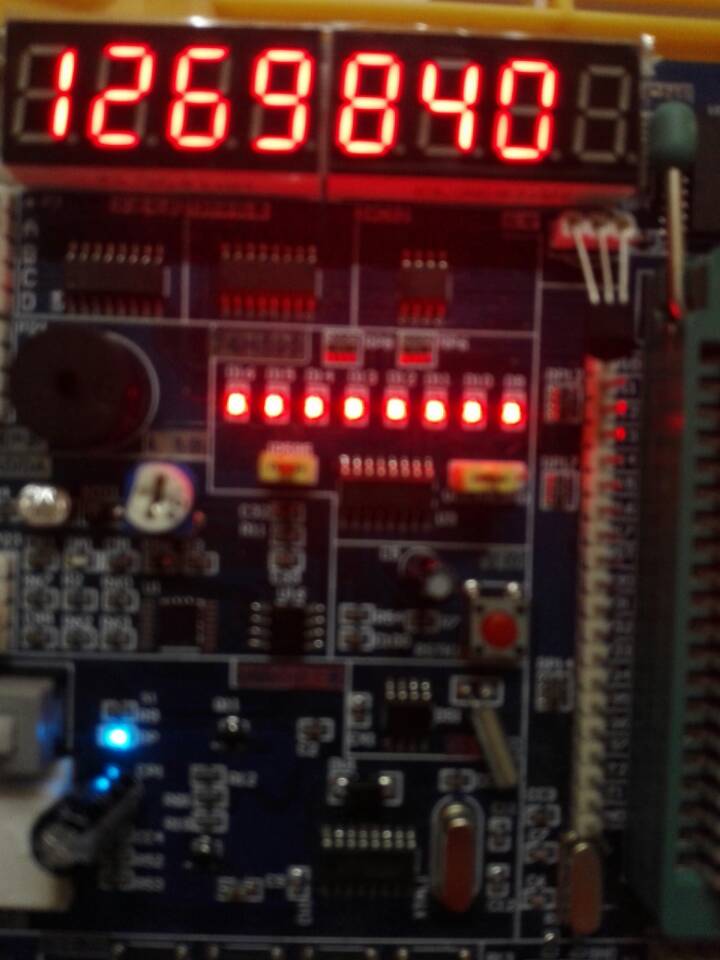
mov 41H,40H

1. 设置蜂鸣器过程中也碰到了没有声音这种情况，后来才知道要蜂鸣器发出声音要不断对它取反，不能只是取反一次，否则听不到，然后就想到了把它放到一个延时函数里面对它不断取反。延时时间不能太长也不能太短。
2. 然后经过不断将程序调试，优化，这个课设就完美解决了。

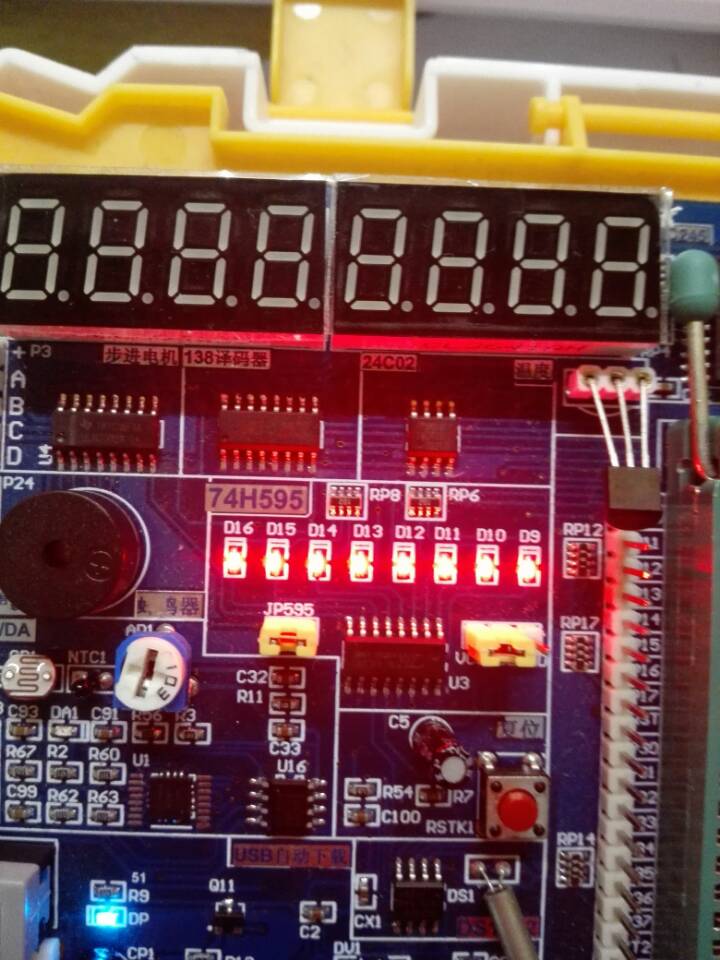
## 输入电话号码：



## 按下删除一个键：



## 按下删除全部键：



## 实验总结

第一次拿到课设题目，感觉相比大部分人的题目我这个题目有点难度，课设第

一天完全不会，甚至一点思路都没有，完全不知道怎么下手，然后第一天就去看我买的单片机视频资料，关于矩阵按键与数码管部分，学了2天之后就大概知道了思路，然后第3天就开始写，然后写了2天半，接下来的周末就一直在调试解决错误，这个课设我一共花了5天半左右。然后感想就是很多东西只要你去花时间，去学，去做就一定可以完成。

## 用户使用说明

矩阵按键的前10个按键分别为0到9，这十个数字，用户可以自由输入自己想要输入的电话号码。第11个按键为一个一个删除最后一个，第12个按键是删除全部输入的。

## 八．实验源程序

//R4为第几位数码管显示

//R3为按键值

//R2为按键状态，1表示按下，0没有

//P0口控制数码管上显示数字几

//矩阵按键连在P1口

//P2口控制第几个数码管显示

org 0000H

mov R2,#0 //设按键初始状态为0

mov R0,#0 //开始按删除最后一个数字的次数为0

MAIN:

LCALL keydown

start:

iskey: CJNE R2,#01,t //判断按键是否按下

CJNE R3,#10,LOOP29 //将10号按键设为一个一个删除

LCALL deleteone

SJMP LOOP39

LOOP29: mov 47H,46H //进行移位

mov 46H,45H

mov 45H,44H

mov 44H,43H

mov 43H,42H

mov 42H,41H

mov 41H,40H

CJNE R3,#0,LOOP30

mov 40H,#3FH

LOOP30: CJNE R3,#1,LOOP31

mov 40H,#06H

LOOP31: CJNE R3,#2,LOOP32

mov 40H,#5BH

LOOP32: CJNE R3,#3,LOOP33

mov 40H,#4FH

LOOP33: CJNE R3,#4,LOOP34

mov 40H,#66H

LOOP34: CJNE R3,#5,LOOP35

mov 40H,#6DH

LOOP35: CJNE R3,#6,LOOP36

mov 40H,#7DH

LOOP36: CJNE R3,#7,LOOP37

mov 40H,#07H

LOOP37: CJNE R3,#8,LOOP38

mov 40H,#7FH

LOOP38: CJNE R3,#9,LOOP40

mov 40H,#6FH

LOOP40: CJNE R3,#11,LOOP39

LCALL deleteall //11号按键设为删除全部

SJMP LOOP39

LOOP39: mov R2,#0 //将按键状态重新置为0

t: LCALL display

recycle: SJMP MAIN

//上面为主函数部分

//下面为子函数部分

DELAY10MS: //延时10ms

MOV R7,#01H

DL1:

MOV R6,#26H

DL0:

MOV R5,#82H

DJNZ R5,$

DJNZ R6,DL0

DJNZ R7,DL1

RET

keydown: //判断按键

mov P1,#0FH

mov A,P1

CJNE A,#0FH,LOOP0 //检查有无按键按下，

RET

LOOP0: LCALL DELAY10MS //延时10ms消抖

mov A,P1

CJNE A,#0FH,LOOP //再次检查有无按键按下

LOOP: mov R2,#01H

mov P1,#0FH //有按键按下

mov A,P1

CJNE A,#07H,LOOP1

mov R3,#0 //R3为按键值

SJMP LOOP4

LOOP1: CJNE A,#0BH,LOOP2 //获得列值

mov R3,#1

SJMP LOOP4 //行列检测法

LOOP2: CJNE A,#0DH,LOOP3

mov R3,#2

SJMP LOOP4

LOOP3: CJNE A,#0EH,LOOP4

mov R3,#3

SJMP LOOP4

LOOP4: mov P1,#0F0H //获得行值

mov A,P1

CJNE A,#70H,LOOP5

INC R3 //R3=R3

DEC R3

SJMP LOOP8

LOOP5: CJNE A,#0B0H,LOOP6

INC R3

INC R3 //R3=R3+4

INC R3

INC R3

SJMP LOOP8

LOOP6: CJNE A,#0D0H,LOOP7 //R3=R3+8

INC R3

INC R3

INC R3

INC R3

INC R3

INC R3

INC R3

INC R3

SJMP LOOP8

LOOP7: CJNE A,#0E0H, LOOP8 //R3=R3+12

INC R3

INC R3

INC R3

INC R3

INC R3

INC R3

INC R3

INC R3

INC R3

INC R3

INC R3

INC R3

//检测按键松手检测

LOOP8:

mov R1,P1

CJNE R1,#0F0H,LOOP9

LCALL sound

RET

LOOP9:

LCALL DELAY10MS

SJMP LOOP8

display: //显示在数码管

mov R4,#0 //数码管显示位数

one:

CJNE R4,#0,two //显示第0位

mov P2,#00000000B

LJMP send

two: CJNE R4,#1,three //显示第1位

mov P2,#00000100B

LJMP send

//显示第2位

three: CJNE R4,#2,four

mov P2,#00001000B

LJMP send

four: CJNE R4,#3,five //显示第3位

mov P2,#00001100B

LJMP send

five: CJNE R4,#4,six //显示第4位

mov P2,#00010000B

LJMP send

six: CJNE R4,#5,seven //显示第5位

mov P2,#00010100B

LJMP send

seven: CJNE R4,#6,height //显示第6位

mov P2,#00011000B

LJMP send

height: CJNE R4,#7,send //显示第7位

mov P2,#00011100B

LJMP send

send:

CJNE R4,#0H,LOOP20 //将准备显示的8个值依次放缓冲区

mov P0,40H

LOOP20: CJNE R4,#1H,LOOP21

mov P0,41H

LOOP21: CJNE R4,#2H,LOOP22

mov P0,42H

LOOP22: CJNE R4,#3H,LOOP23

mov P0,43H

LOOP23: CJNE R4,#4H,LOOP24

mov P0,44H

LOOP24: CJNE R4,#5H, LOOP25

mov P0,45H

LOOP25: CJNE R4,#6H,LOOP26

mov P0,46H

LOOP26: CJNE R4,#7H, LOOP13

mov P0,47H

LOOP13: INC R4

mov R5,#50 //扫描时间100微秒

DJNZ R5,$

mov P0,#00H

CJNE R4,#8,LOOP12

RET

LOOP12:

LJMP one

DELAY200MS: ;误差 0us

MOV R7,#04H

DL11:

MOV R6,#74H

DL01:

MOV R5,#0D6H

DJNZ R5,$

CPL P1.5 //延时过程蜂鸣器P1.5口不断取反

DJNZ R6,DL01

DJNZ R7,DL11

NOP

RET

sound: //蜂鸣器发出声音

LCALL DELAY200MS

LOOP50: RET

deleteall: //删除全部就是不让数码管亮

mov 40H,#00H //全部覆盖掉

mov 41H,#00H

mov 42H,#00H

mov 43H,#00H

mov 44H,#00H

mov 45H,#00H

mov 46H,#00H

mov 47H,#00H

RET

deleteone:

INC R0 //判断是第几次按删除键

CJNE R0,#1,LOOP41

mov 40H,#00H //第1次按删除键

LJMP LOOP48

LOOP41: CJNE R0,#2,LOOP42 //第2次按删除键

mov 40H,#00H

mov 41H,#00H

LJMP LOOP48

LOOP42: CJNE R0,#3 ,LOOP43 //第3次按删除键

mov 40H,#00H

mov 41H,#00H

mov 42H,#00H

LJMP LOOP48

LOOP43: CJNE R0,#4 ,LOOP44 //第4次按删除键

mov 40H,#00H

mov 41H,#00H

mov 42H,#00H

mov 43H,#00H

LJMP LOOP48

LOOP44: CJNE R0,#5 ,LOOP45 //第5次按删除键

mov 40H,#00H

mov 41H,#00H

mov 42H,#00H

mov 43H,#00H

mov 44H,#00H

LJMP LOOP48

LOOP45: CJNE R0,#6 ,LOOP46 //第6次按删除键

mov 40H,#00H

mov 41H,#00H

mov 42H,#00H

mov 43H,#00H

mov 44H,#00H

mov 45H,#00H

LJMP LOOP48

LOOP46: CJNE R0,#7,LOOP47 //第7次按删除键

mov 40H,#00H

mov 41H,#00H

mov 42H,#00H

mov 43H,#00H

mov 44H,#00H

mov 45H,#00H

mov 46H,#00H

SJMP LOOP48

LOOP47: CJNE R0,#8,LOOP48 //第8次按删除键

mov 40H,#00H

mov 41H,#00H

mov 42H,#00H

mov 43H,#00H

mov 44H,#00H

mov 45H,#00H

mov 46H,#00H

mov 47H,#00H

LOOP48:

RET

END