****

课程设计报告书

**基于单片机的密码锁系统**

**学 院 电子与信息学院**

**专 业 信息工程**

**学生姓名 黄铤**

**学生学号 201530251409**

**指导教师 梁亚玲**

**课程编号 135146**

**课程学分 1**

**起始日期 2017年4月29日-6月16日**

|  |  |
| --- | --- |
| 教  师  评  语 | 教师签名：  日期： |
| 成  绩  评  定 |  |
| 备  注 |  |

**目录**

[基于单片机的密码锁系统 3](#_Toc485335667)

[一、选题背景 3](#_Toc485335668)

[二、方案论证(设计理念) 3](#_Toc485335669)

[1.键盘扫描 3](#_Toc485335670)

[2.中断处理 4](#_Toc485335671)

[3.显示指示器 4](#_Toc485335672)

[4.蜂鸣器 4](#_Toc485335673)

[三、过程论述 5](#_Toc485335674)

[（一）分立模块的设计 5](#_Toc485335675)

[（二）综合模块的设计 20](#_Toc485335676)

[四、结果分析 36](#_Toc485335677)

[五、课程设计总结 39](#_Toc485335678)

[1、心得体会 39](#_Toc485335679)

[2、关于《微机系统与接口》课程建议 40](#_Toc485335680)

[3、自我评价 40](#_Toc485335681)

[参考文献 41](#_Toc485335682)

# 基于单片机的密码锁系统

## 一、选题背景

随着人们生活水平的提高和安全意识的加强，对安全的要求也就越来越高。随着电子技术的发展，各类电子产品应运而生，电子密码锁就是其中之一。电子锁是通过键盘输入一组密码完成开锁过程。研究这种锁的初衷，就是为提高锁的安全性。由于电子锁的密钥量极大，可以与机械锁配合使用，并且可以避免因钥匙被仿制而留下安全隐患。电子锁只需记住一组密码，无需携带金属钥匙，免除了人们携带金属钥匙的烦恼，而被越来越多的人所欣赏。电子锁的种类繁多，例如数码锁，指纹锁，磁卡锁IC卡锁，生物锁等。但较实用的还是按键式电子密码锁。

基于这种情况，本课题在于设计一个程序，利用单片机实现通过按下按键，数码管会显示出相应按键所代表的键值，只有当你输入的键值符合要求时，才能进行下一道程序，否则，将提示错误信息。

通过设计任务，把所学的理论与实际相联系，加深、巩固“单片机”课程所学的知识并学会综合、灵活地运用，培养、训练单片机开发、应用的基本技能。

## 二、方案论证(设计理念)

### 1.键盘扫描

（1）矩阵键盘的介绍

单片机中的矩阵键盘是由51芯片中的P2端口所拉出来的8根引线组成的，其中P2.0~P2.3作为行线，P2.4~P2.7作为列线。在行线和列线的每个交叉点上设置一个按键，这样键盘上按键的个数就为4×4个。

（2）识别矩形键盘按键的方法

对矩阵键盘按键识别常用的有扫描法，反极法和位操作法：

①扫描法

首先判断是否有键按下，使全部行线输出低电平，检测列线的电平状态；确认有键按下后，每条行线分别单独输出低电平，检测列线为零的交点，由此得到的一个八位二进制可利用穷举法进行键值赋值。

②反极法

行线全送出低电平，检测列线状态，出现低电平的列线即被按下；此后，相反的，将全部列线送出低电平，检测行线状态，出现低电平的行线即被按下。将得到的两个表示状态的八位二进制进行或运算，就在进行查找键值。

③位操作法

把行输出线逐位置为低电平，对列输入线逐位扫描，有键按下时列线变为低电平。再把行线逐条置为低电平，其余行线置为高电平，用位比较指令逐列扫描，确定按下的键所在的列。赋予键值。把行输出线逐位置为低电平，对列输入线用位比较指令逐位扫描，若列线全部为高电平，表示按键已释放。

本实验采用位操作法得到的一个八位二进制，利用穷举法进行键值赋值。

### 2.中断处理

在AG嵌入式开发板V1.2中含有两个外部中断源，利用这两个中断源可以切换不同的功能，实验中通过按下中断0、中断1，可以分别执行不同的子功能。

单片机四位数码管会闪烁显示“ERR-”三次，同时蜂鸣器会响，当数码管显示完第三次“ERR-”后，蜂鸣器将会停止发声。接着单片机

### 3.显示指示器

AG嵌入式开发板V1.2配备了8个LED灯以及4个八段数码管，LED和数码管共用

一个I/O接口即接口P0。单片机中的LED灯为共阳极型，阴极接低电平时点亮，因此在触发条件后，只需向LED灯控制端输出低电平LED等就会被点亮。即如果P0.X为低电平时，与该引脚相连接的LED灯会被点亮，这样就可以通过控制P0的各个引脚的高低电平来点亮相应的LED。

### 4.蜂鸣器

通过一个I/O接口即P1.3与蜂鸣器相连，当传输为高电平时，三极管不导通，蜂鸣器接收到低电平；而当其为低电平时，三极管导通，蜂鸣器接收到高电平。对于蜂鸣器，I/O接口为低电平是发声，为高电平时消声（即低电平有效）。

## 三、过程论述

### （一）分立模块的设计

#### 1.输入按键与显示

在这一部分中，需要对矩阵键盘进行识别并将识别结果发送到LED显示器中去，使

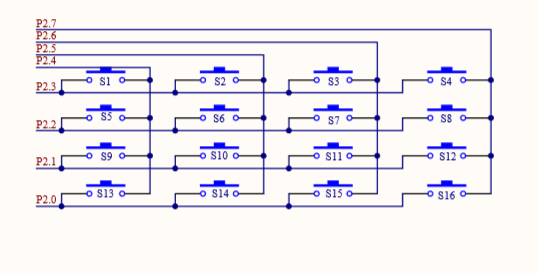
数码管显示相应的数值。下图3.1.1为矩阵键盘的结构图：

图3.1.1

从原理图中可以看出，矩阵键盘由P2口控制，利用扫描法对矩阵键盘进行识别。识别过程要加入延时防抖，防止误判。不同按键将有16个不同的识别结果，对应于0～F这16个数值可在段码表寻找显示相应数值的段码。如下表3.1.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表3.1.1 数码管的段码表 | | | | | | | | |
| 键值 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 段码 | 0FCH | 60H | 0DAH | 0F2H | 66H | 0B6H | 0BEH | 0E0H |
| 键值 | 8 | 9 | A | b | C | d | E | F |
| 段码 | 0FEH | 0F6H | 0EEH | 3EH | 9CH | 7AH | 9EH | 8EH |

注：为了与8,0区分开来，十进制数中的11,13分别用小写字母b，d表示

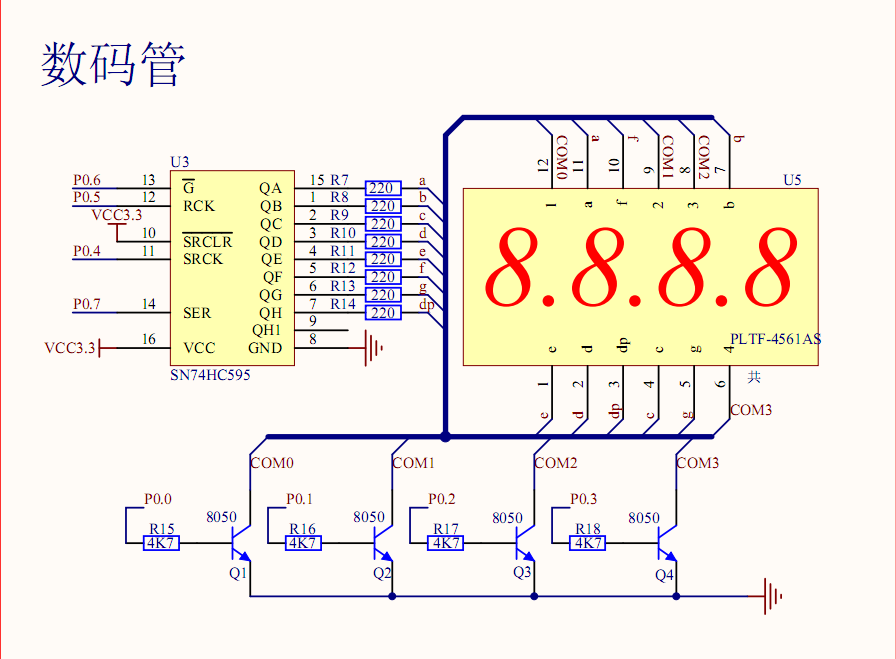


图3.1.2

将段码表中的段码作为控制信号输入到数码管的控制端就可实现数码管的显示功能，数码管的原理图如图3.1.2所示。

根据以上的设计要求，可以得到程序的运行框图如图3.1.3所示。

N

延时防抖

识别行线

识别列线

按键？

按键？

对应键值

对应段码

控制数码管显示

Y

Y

N

图3.1.3

为实现10～20ms的延时，设计的延时子程序参数设置为25和248，此时得到的延时效果为  
（（248\*2）+3）\*25+312.5ms

代码：

HC595\_SCK EQU P0.4

HC595\_RCK EQU P0.5

HC595\_RST EQU P0.6

HC595\_DAT EQU P0.7

ORG 0000H

JMP MAIN

ORG 0030H

LCALL SCAN ;扫描

MOV A,R2

CJNE A,#10H,L11

AJMP L12

L11: MOV R1,B ;有按键按下时R1的值变为10H

L12:;

LCALL SMG\_ShowNum1 ;将第四个数码管显示

ANL P0,#0F0H

ORL P0,#08H

LCALL delay2ms

AJMP L10

SCAN:

NOP ;键盘位操作法子程序

NOP

MOV B,#10H

CLR P2.0 ;将行线初始状态设置为0

CLR P2.1

CLR P2.2

CLR P2.3 ;将列线初始状态设置为1

SETB P2.4

SETB P2.5

SETB P2.6

SETB p2.7

K00: ;查列线判断是否有键按下

JNB P2.4, K01

JNB P2.5, K01

JNB P2.6, K01

JNB P2.7, K01

AJMP K10 ;无键按下再等待

K01: MOV R2 ,#0FH

ACALL DLY ;调用延时子程序去抖动

;查列线判断是否有键按下

JNB P2.4, K02

JNB P2.5, K02

JNB P2.6, K02

JNB P2.7, K02

AJMP K00 ;无键按下再等待

K02: CLR P2.0 ;首行置0判断是哪个键按下

SETB P2.1

SETB P2.2

SETB P2.3

JNB P2.4, K03

JNB P2.5, K03

JNB P2.6, K03

JNB P2.7, K03

SETB P2.0

CLR P2.1 ;第二行置0判断是哪个键按下

SETB P2.2

SETB P2.3

JNB P2.4, K03

JNB P2.5, K03

JNB P2.6, K03

JNB P2.7, K03

SETB P2.0

SETB P2.1

CLR P2.2 ;第三行置0判断是哪个键按下

SETB P2.3

JNB P2.4, K03

JNB P2.5, K03

JNB P2.6, K03

JNB P2.7, K03

SETB P2.0

SETB P2.1

SETB P2.2 ;第三行置0判断是哪个键按下

CLR P2.3

JNB P2.4, K03

JNB P2.5, K03

JNB P2.6, K03

JNB P2.7, K03

AJMP K02

K03: MOV A,P2 ;隔离最高位取键值

MOV B,A ;暂存A中的键值与B中

K04: JNB P2.4,K04

K05: JNB P2.5,K05

K06: JNB P2.6,K06

K07: JNB P2.7,K07

MOV R2,#0FH

ACALL DLY ;调用延时子程序去抖动

K077: JNB P2.4,K04

JNB P2.5,K04

JNB P2.6,K04

JNB P2.7,K04

SETB P2.0 ;行线恢复1状态

SETB P2.1

SETB P2.2

SETB P2.3

MOV R1,#16 ;总共16个键，依次查表，也表示循环次数

MOV A,#0

MOV DPTR, #KEYT ;表头

K08: PUSH ACC

MOVC A,@A+DPTR ;按下的键值与键值表比较

CJNE A,B,K09 ;通过查表找到与B相同的键值

POP ACC

MOV B,A ;把按键对应的数字赋给B

AJMP K10

K09: POP ACC

INC A

DJNZ R1,K08 ;16键未完继续比较

KEYT: DW 777bH, 7d7eH ;对应数值0,1,2,3

DW 0b7bbH, 0bdbeH ;对应数值4,5,6,7

DW 0d7dbH, 0dddeH ;对应数值8,9,A,B

DW 0e7ebH, 0edeeH ;对应数值C,D,E,F

K10: MOV R2,B ;将数值存入R1

RET

;--------------向数码管发送一个字节---------------------

HC595\_send\_byte:

K11:MOV R4,#08H

MOV R3,#01H

K12:MOV A,R3

ANL A,R7

MOV R6,A

CJNE R6,#00,Y10

CLR HC595\_DAT

JMP WRIT

Y10:

SETB HC595\_DAT

WRIT:

SETB HC595\_SCK

NOP

CLR HC595\_SCK

NOP

XCH A,R3

RL A

XCH A,R3

DJNZ R4,K12

CLR HC595\_RCK

NOP

SETB HC595\_RCK

RET

;----------向第四个数码管发送R1储存的键值--------------

SMG\_ShowNum1:

MOV A,R1

MOV DPTR,#KEYT1 ;表头

MOVC A,@A+DPTR ;选择相应的段选

MOV R7,A

LCALL HC595\_send\_byte

RET

KEYT1: DW 0fc60H, 0daf2H ;对数码管0,1,2,3

DW 66b6H, 0bee0H ;对数码管4,5,6,7

DW 0fef6H, 0ee3eH ;对数码管8,9,A,B

DW 9c7aH, 9e8eH ;对数码管C,D,E,F

;-----------------延时12ms--------------------

DLY: MOV R6,#25

DLY0: MOV R7,#248

DLY1: DJNZ R7,DLY1

DJNZ R6,DLY0

RET

;-----------延时2ms------------------

delay2ms:

MOV R6,#50

DELAY\_2:

DJNZ R6,DELAY\_2

RET

END

#### 2、外部中断0

按下KEY0，数码管闪烁“ERR-”三次并使驱动蜂鸣器发音；接着任意按下按键S1-

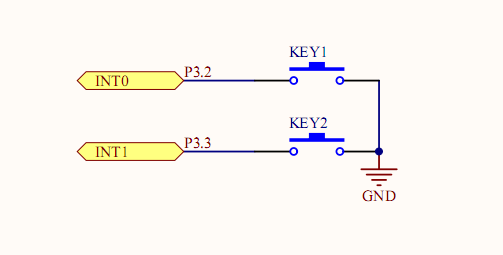
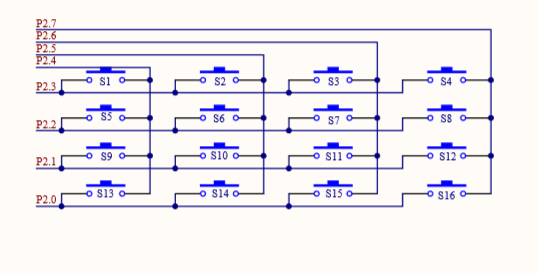
S7，可以将相应的LED灯单独点亮，例如按下S0键，LED0会被点亮。AG嵌入式开发板的外部中断按键如图3.2.1，矩形键盘结构如图3.2.2，LED流水灯的原理电路如图3.2.3，蜂鸣器的原理图如图3.2.4

图3.2.1

 图3.2.2

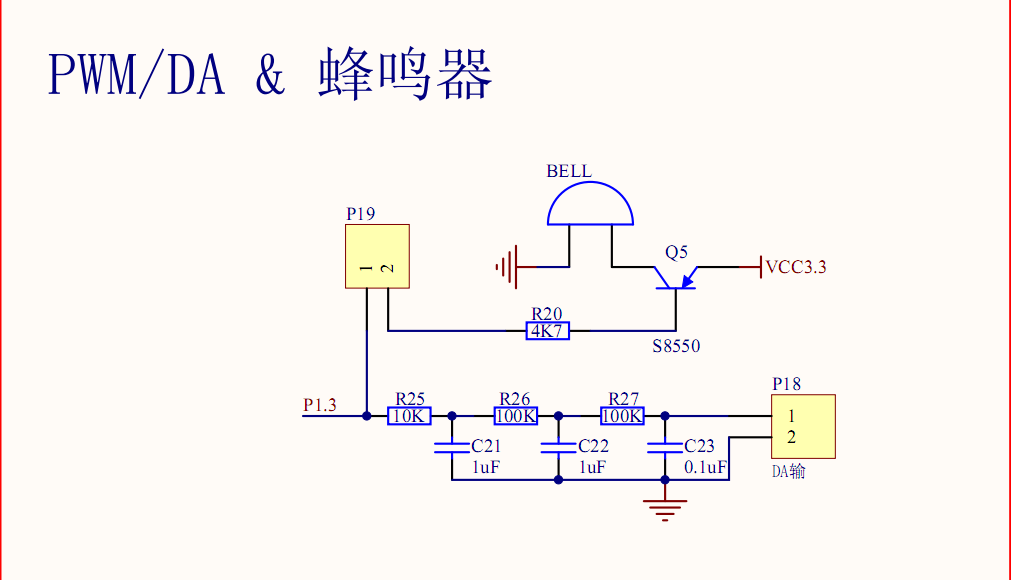


图3.2.4

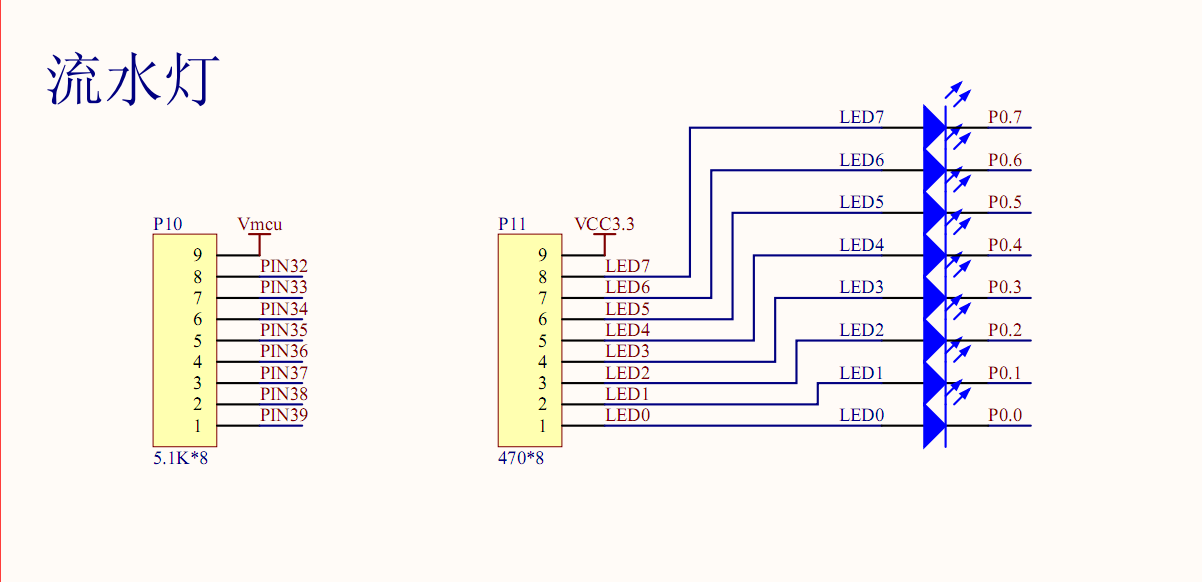


图3.2.3

利用中断0使程序转至相应的中断子程序执行相应的功能，这样可以实现单片机功能的切换。当接收到中断信号时，将控制LED的P1.3口清0，并通过P0口控制数码管显示“ERR-”，接着便进入用键盘控制LED灯亮的功能，便可以实现将这两个子功能整合为一个功能。

系统运行的流程图如图3.2.5所示

设计的程序如下：

HC595\_SCK EQU P0.4

图3.2.5

等待中断0

使数码管闪烁“ERR-”三次，并驱动蜂鸣器发音

中断？

判断按下何键

扫描

Y

N

Y

HC595\_RCK EQU P0.5

HC595\_RST EQU P0.6

HC595\_DAT EQU P0.7

beep EQU P1.3

ExtInt0 EQU P3.2

ORG 0000H

AJMP MAIN

ORG 0003H

AJMP INT00

ORG 0030H

Y

判断是否按下中断0

MAIN:

N

MOV SP,#60H

SETB ExtInt0

LCALL Exti0\_Init ;初始化中断0

N

按键按下？健？

AJMP MAIN

;-------------数码管显示ERR- -----------------

SMG\_ShowNum2:

MOV DPTR, #KEYT3 ;表头

点亮对应的LED灯

MOVC A,@A+DPTR ;选择相应的段选

MOV R7,A

返回主函数

AJMP OK

KEYT3: DW 9E8CH, 8C02H ;数码管显示EER-

OK: LCALL HC595\_send\_byte

RET

;---------------------------------------- ;这里的HC595\_send\_byte函数在第一个分立模块已经定义，这里就直接调用，不再重复定义

SMG\_Display2:

MOV A,#00H

LCALL SMG\_ShowNum2

ANL P0,#0F0H

ORL P0,#01H

LCALL delay2ms

MOV A,#01H

LCALL SMG\_ShowNum2

ANL P0,#0F0H

ORL P0,#02H

LCALL delay2ms

MOV A,#02H

LCALL SMG\_ShowNum2

ANL P0,#0F0H

ORL P0,#04H

LCALL delay2ms

MOV A,#03H

LCALL SMG\_ShowNum2

ANL P0,#0F0H

ORL P0,#08H

LCALL delay2ms

RET

;-----------按下中断0时点亮LED灯 ---

Light\_Off2:

MOV R7,#00H

LCALL HC595\_send\_byte

ANL P0,#0F0H

ORL P0,#01H

LCALL delay2ms

MOV R7,#00H

LCALL HC595\_send\_byte

ANL P0,#0F0H

ORL P0,#02H

LCALL delay2ms

LCALL HC595\_send\_byte

ANL P0,#0F0H

ORL P0,#04H

LCALL delay2ms

LCALL HC595\_send\_byte

ANL P0,#0F0H

ORL P0,#08H

LCALL delay2ms

RET;-------------外部中断0初始化函数----------------

Exti0\_Init:

MOV TCON,#01H

SETB EX0

SETB EA

RET

;--------------寻找相应的按键Sx并点亮对应的LEDx灯-------------------------

research:

MOV A,R1

CLR C

SUBB A,#08H

JNC unfind

MOV A,R1

MOV DPTR, #KEYT2 ;表头

MOVC A,@A+DPTR ;按下的键值与键值表比较

AJMP find

KEYT2: DB 0FEH, 0FDH ;对应键字0,1,

DB 0FBH, 0F7H ;对应键字2,3

DB 0EFH, 0DFH ;对应键字4,5

DB 0BFH, 7FH ;对应键字6.7

find: MOV P0,A

RET

unfind: MOV P0,#0FFH ;若按下S8-S15，LED灯全部熄灭

RET

;-------------中断0服务函数--------------------

INT00:

MOV A,P0

ANL A,#0BFH

MOV P0,A

MOV R1,#03H

L23:

MOV R2,#10H

L20:MOV R0,#50

LM20:

LCALL SMG\_Display2

CLR beep

DJNZ R0,LM20

DJNZ R2,L20

MOV R2,#10H

LM22:

MOV R0,#50

LM21:

LCALL Light\_Off2

SETB beep

DJNZ R0,LM21

DJNZ R2,LM22

DJNZ R1,L23

;--------控制led灯亮-------------------------------

MOV R1,#0FFH

L30:

LCALL SCAN ;扫描（这里的SCAN函数在第一个分立模块已经定义，这里直接调用，不再重复定义）

MOV A,R2

CJNE A,#10H,L31

AJMP L32

L31:

MOV A,R2

MOV R1,A

L32: LCALL research

CJNE R1,#0EH,L33

CLR EX0

MOV P0,#00

AJMP L34

L33:

LCALL delay50ms

AJMP L30

L34:

RETI

delay2ms:

MOV R6,#50

DELAY\_2:

DJNZ R6,DELAY\_2

RET

END

#### 3、外部中断1

与外部中断0的分立模块相似，都是利用中断独立按键实现对程序的控制。当按下KEY1后，单片机进入中断1服务子程序，接着通过定时器1的溢出中断，将程序跳转到TIME1中断子程序，在TIME1中断子程序中点亮一个LED灯。即让程序循环中断1跳到定时器1的溢出中断，每次跳转均按顺序点亮一个LED灯。由于中断1的优先级比定时器1的溢出中断高，因此，应给予定时器1更高的优先级。整体的流程设计如图3.3.1

定时器1设置高优先级

等待中断1

外部中断？

延时

N

跳到定时器1溢出中断子程序

走马灯循环次数i=10

依次点亮8个LED灯

N

打开定时器1

Y

i-10?

定时器计数

N

N

定时器溢出中断

图3.3.1

Y

设计的程序如下：

ORG 0000H

JMP MAIN

ORG 0013H

AJMP INT011

ORG 001BH

AJMP TIME1

ORG 0030H

MAIN:

MOV SP,#60H

MOV P0,#8FH

SETB ET1

SETB ExtInt1 ;将p3.3置1

L10:

LCALL Exti0\_Init

LCALL Exti1\_Init1 ;设置外部中断

LCALL TIME\_INIT1

AJMP L10

;------------定时器1初始化------------

TIME\_INIT1:

MOV TMOD,#90H ;采用方式1

MOV TH1,#00H ;定时器1计数的初始值从0开始

MOV TL1,#00H

SETB ET1

SETB PT1

RET

;-------------外部 中断1初始化函数------------------------

Exti1\_Init1:

SETB IT1

SETB EX1

SETB EA ;定义全局中断允许

CLR TF1

CLR TR1

RET

;-------------定时器1服务函数--------------------

TIME1: ;定时器

DJNZ R2,RETURN

MOV A,R1

MOV DPTR, #KEYT4 ;表头

MOVC A,@A+DPTR ;按下的键值与键值表比较

MOV P0,A

INC R1

MOV R2,#02 ;每循环两次溢出中断后才点亮后面的另一个LED灯，即每个LED灯的点亮时间为60ms\*2=1.2S

CJNE R1,#15,RETURN

MOV R1,#00

DEC R3

CJNE R3,#00,RETURN

CLR ET1

RETURN: RETI

KEYT4: DB 0FEH, 0FDH ;对应键字0,1,

DB 0FBH, 0F7H ;对应键字2,3

DB 0EFH, 0DFH ;对应键字4,5

DB 0BFH, 7FH ;对应键字6.7

DB 0BFH, 0DFH

DB 0EFH, 0F7H

DB 0FBH, 0FDH

DB 0FEH

;-------------中断1服务函数-----------------------------

INT011:

MOV R1,#00

MOV R2,#02

MOV R3,#05 ;走马灯来回次数，一次来回算一次

START: SETB TR1 ;打开定时器1，开始计数

NOT\_FINISH:

JB TF1,NOT\_FINISH

JNB ET1,K40

AJMP START

K40:

CLR EX1

MOV P0,#00

RETI

### （二）综合模块的设计

通过把各个分立模块整合到一个程序中，即可得到最基本的综合模块——密码锁。在此模块中，可以实现功能如下：

本课题设计的密码锁跟一般的密码锁有所不同，主要区别是我们这个密码锁有三层密码：①首先要通过键盘任意输入四个数（0、1、2……F），输完之后，系统会按照之前键盘输入的键值顺序依次点亮相应的LED灯。其中键值0对应LED0、键值1对应LED1……键值7对应LED7，键值8到键值F没有LED灯与之对应。若出现从右往左依次点亮4个连续的LED灯的现象（即输入的四个键值均在键值0~7之间，且连续递增），则说明第一层密码输入正确。否则，数码管会显示“ERR-”三次同时蜂鸣器会发出警告声，提示第一层密码输入有误，需要重新输入。②只有在第一层密码输入正确后，按下KEY1时，系统才会演示走马灯10次，说明第二层密码输入有效。反之，若第一层密码输入错误，按下KEY1键后，系统不会演示走马灯，此时说明第二层密码输入无效。③在第一层密码输入正确且第二层密码有效时，按下KEY0键后，数码管会显示“YES-”三次，同时蜂鸣器会发音，说明成功打开了密码锁的三层密码。

综合模块的流程图如图3.4.1所示

按下KEY1键，输入第二层密码

判断输入的密码是否连续且在0~7范围内

蜂鸣器发出警告声

数码管显示ERR-三次

输入四个密码?

演示走马灯10次

图3.4.1

按下KEY0键，输入第三层密码

蜂鸣器发出提示声，打开密码锁

数码管闪烁“YES-”三次

N

Y

Y

N

整合得到的综合模块程序如下：

HC595\_SCK EQU P0.4

HC595\_RCK EQU P0.5

HC595\_RST EQU P0.6

HC595\_DAT EQU P0.7

beep EQU P1.3

ExtInt0 EQU P3.2

ExtInt1 EQU P3.3

;---------------------------------

ORG 0000H

JMP MAIN

ORG 0003H

AJMP INT00

ORG 0013H

AJMP INT011

ORG 001BH

AJMP TIME1

ORG 0030H

MAIN:

MOV SP,#60H

MOV P0,#8FH

SETB ExtInt0

SETB ExtInt1 ;将p3.3置1

L10:

LCALL Exti0\_Init0 ;关闭中断

LCALL Exti1\_Init11

L12:

MOV P0,#0FH

LCALL SCAN

LCALL SMG\_ShowNum1 ;将第一个数码管显示

ANL P0,#0F0H

ORL P0,#01H

MOV A,R1

PUSH ACC

LCALL SCAN

LCALL SMG\_ShowNum1 ;将第二个数码管显示

ANL P0,#0F0H

ORL P0,#02H

MOV A,R1

PUSH ACC

LCALL SCAN

LCALL SMG\_ShowNum1 ;将第三个数码管显示

ANL P0,#0F0H

ORL P0,#04H

MOV A,R1

PUSH ACC

LCALL SCAN

LCALL SMG\_ShowNum1 ;将第四个数码管显示

ANL P0,#0F0H

ORL P0,#08H

MOV A,R1

PUSH ACC

LCALL delay2s

POP ACC ;将四个键值存进寄存器

MOV R5,A

POP ACC

MOV R4,A

POP ACC

MOV R3,A

POP ACC

MOV R2,A

MOV A,R5 ;再将四个键值重新压入栈，即用两个栈实现一个队列

PUSH ACC

MOV A,R4

PUSH ACC

MOV A,R3

PUSH ACC

MOV A,R2

PUSH ACC

MOV B,#04

POP ACC ;分别弹出四个键值，点亮键值对应的Led

MOV R1,A

LCALL research

LCALL delay2s

POP ACC

MOV R1,A

LCALL research

LCALL delay2s

POP ACC

MOV R1,A

LCALL research

LCALL delay2s

POP ACC

MOV R1,A

LCALL research

LCALL delay2s

MOV A,B

CJNE A,#00,VIOCE ;若输入的键值至少有一个大于7，则不允许中断

MOV A,#01

ADD A,R2

SUBB A,R3

CJNE A,#00,VIOCE

MOV A,#01

ADD A,R3

SUBB A,R4

CJNE A,#00,VIOCE

MOV A,#01

ADD A,R4

SUBB A,R5

CJNE A,#00,VIOCE ;输入四个连续的数才允许中断1

SETB ET1

LCALL Exti1\_Init1 ;开启外部中断1

LCALL TIME\_INIT1

F1:AJMP L12

F2:AJMP L12

F3:AJMP L12

VIOCE:LCALL ERROR

AJMP L10

;--------------扫描是否按下按键-------------------

SCAN:

NOP ;键盘位操作法子程序

NOP

MOV B,#10H

CLR P2.0 ;将行线初始状态设置为0

CLR P2.1

CLR P2.2

CLR P2.3 ;将列线初始状态设置为1

SETB P2.4

SETB P2.5

SETB P2.6

SETB p2.7

K00: ;查列线判断是否有键按下

JNB P2.4, K01

JNB P2.5, K01

JNB P2.6, K01

JNB P2.7, K01

AJMP K00 ;无键按下再等待

K01: MOV R2 ,#0FH

ACALL DLY ;调用延时子程序去抖动

;查列线判断是否有键按下

JNB P2.4, K02

JNB P2.5, K02

JNB P2.6, K02

JNB P2.7, K02

AJMP K00 ;无键按下再等待

K02: CLR P2.0 ;首行置0判断是哪个键按下

SETB P2.1

SETB P2.2

SETB P2.3

JNB P2.4, K03

JNB P2.5, K03

JNB P2.6, K03

JNB P2.7, K03

SETB P2.0

CLR P2.1 ;第二行置0判断是哪个键按下

SETB P2.2

SETB P2.3

JNB P2.4, K03

JNB P2.5, K03

JNB P2.6, K03

JNB P2.7, K03

SETB P2.0

SETB P2.1

CLR P2.2 ;第三行置0判断是哪个键按下

SETB P2.3

JNB P2.4, K03

JNB P2.5, K03

JNB P2.6, K03

JNB P2.7, K03

SETB P2.0

SETB P2.1

SETB P2.2 ;第三行置0判断是哪个键按下

CLR P2.3

JNB P2.4, K03

JNB P2.5, K03

JNB P2.6, K03

JNB P2.7, K03

AJMP K02

K03: MOV A,P2 ;隔离最高位取键值

MOV B,A ;暂存键值 ,,,,,,,,

K04: JNB P2.4,K04

K05: JNB P2.5,K05

K06: JNB P2.6,K06

K07: JNB P2.7,K07

MOV R2,#0FH

ACALL DLY ;调用延时子程序去抖动

K077: JNB P2.4,K04

JNB P2.5,K04

JNB P2.6,K04

JNB P2.7,K04

SETB P2.0 ;行线恢复1状态

SETB P2.1

SETB P2.2

SETB P2.3

MOV R1,#16 ;总共16个键，依次查表，也表示循环次数

MOV A,#0

MOV DPTR, #KEYT ;表头

K08: PUSH ACC

MOVC A,@A+DPTR ;按下的键值与键值表比较

CJNE A,B,K09

POP ACC

MOV B,A ;把键值赋给B .............

AJMP K10

K09: POP ACC

INC A

DJNZ R1,K08 ;16键未完继续比较

KEYT: DW 777bH, 7d7eH ;对应数值0,1,2,3

DW 0b7bbH, 0bdbeH ;对应数值4,5,6,7

DW 0d7dbH, 0dddeH ;对应数值8,9,A,B

DW 0e7ebH, 0edeeH ;对应数值C,D,E,F

K10: MOV R1,B ;将键值存入R1

RET

;--------------发送一个字节---------------------

HC595\_send\_byte:

K11:MOV R4,#08H

MOV R3,#01H

K12:MOV A,R3

ANL A,R7

MOV R6,A

CJNE R6,#00,Y10

CLR HC595\_DAT

JMP WRIT

Y10:

SETB HC595\_DAT

WRIT:

SETB HC595\_SCK

NOP

CLR HC595\_SCK

NOP

XCH A,R3

RL A

XCH A,R3

DJNZ R4,K12

CLR HC595\_RCK

NOP

SETB HC595\_RCK

RET

;----------向数码管发送R1储存的键值--------------

SMG\_ShowNum1:

MOV A,R1

MOV DPTR,#KEYT1 ;表头

MOVC A,@A+DPTR ;选择相应的段选

MOV R7,A

LCALL HC595\_send\_byte

RET

KEYT1: DW 0fc60H, 0daf2H ;对数码管0,1,2,3

DW 66b6H, 0bee0H ;对数码管4,5,6,7

DW 0fef6H, 0ee3eH ;对数码管8,9,A,B

DW 9c7aH, 9e8eH ;对数码管C,D,E,F

;--------------- 建立ERR-对应键值表-----------------------------------

SMG\_ShowNum2:

MOV DPTR, #KEYT3 ;表头

MOVC A,@A+DPTR ;选择相应的段选

MOV R7,A

AJMP OK

KEYT3: DW 9EEEH, 0EE02H ;对数码管0,1,2,3

OK: LCALL HC595\_send\_byte

RET

;---------------建立YES-对应键值表---------------------

SMG\_ShowNum3:

MOV DPTR, #KEYT4 ;表头

MOVC A,@A+DPTR ;选择相应的段选

MOV R7,A

AJMP OK1

KEYT4: DW 769EH, 0B602H ;对数码管y,E,S,-

OK1: LCALL HC595\_send\_byte

RET

;--------------寻找相应的按键并使相应的led灯亮-------------------------

research:

DEC B

MOV A,R1

CLR C

SUBB A,#08H

JNC unfind

MOV A,R1

MOV DPTR, #KEYT2 ;表头

MOVC A,@A+DPTR ;按下的键值与键值表比较

AJMP find

KEYT2: DB 0FEH, 0FDH ;对应键字0,1,

DB 0FBH, 0F7H ;对应键字2,3

DB 0EFH, 0DFH ;对应键字4,5

DB 0BFH, 7FH ;对应键字6.7

find: MOV P0,A

RET

unfind:

INC B

MOV P0,#0FFH

ENDS40:RET

;-------------显示ERR- ---------------------------

SMG\_Display2:

MOV A,#00H

LCALL SMG\_ShowNum2

ANL P0,#0F0H

ORL P0,#01H

LCALL delay2ms

MOV A,#01H

LCALL SMG\_ShowNum2

ANL P0,#0F0H

ORL P0,#02H

LCALL delay2ms

MOV A,#02H

LCALL SMG\_ShowNum2

ANL P0,#0F0H

ORL P0,#04H

LCALL delay2ms

MOV A,#03H

LCALL SMG\_ShowNum2

ANL P0,#0F0H

ORL P0,#08H

LCALL delay2ms

RET

;-----------显示YES----------------

SMG\_Display3:

MOV A,#00H

LCALL SMG\_ShowNum3

ANL P0,#0F0H

ORL P0,#01H

LCALL delay2ms

MOV A,#01H

LCALL SMG\_ShowNum3

ANL P0,#0F0H

ORL P0,#02H

LCALL delay2ms

MOV A,#02H

LCALL SMG\_ShowNum3

ANL P0,#0F0H

ORL P0,#04H

LCALL delay2ms

MOV A,#03H

LCALL SMG\_ShowNum3

ANL P0,#0F0H

ORL P0,#08H

LCALL delay2ms

RET

;-----------------熄灭数码管--------------------

Light\_Off2:

MOV R7,#00H

LCALL HC595\_send\_byte

ANL P0,#0F0H

ORL P0,#01H

LCALL delay2ms

MOV R7,#00H

LCALL HC595\_send\_byte

ANL P0,#0F0H

ORL P0,#02H

LCALL delay2ms

LCALL HC595\_send\_byte

ANL P0,#0F0H

ORL P0,#04H

LCALL delay2ms

LCALL HC595\_send\_byte

ANL P0,#0F0H

ORL P0,#08H

LCALL delay2ms

RET

;----------------提示错误---------------------

ERROR:

MOV A,P0

ANL A,#0BFH

MOV P0,A

MOV R1,#03H

L23:

MOV R2,#20H

L20:MOV R0,#50

LM20:

LCALL SMG\_Display2

CLR beep

DJNZ R0,LM20

DJNZ R2,L20

MOV R2,#20H

LM22:

MOV R0,#50

LM21:

LCALL Light\_Off2

SETB beep

DJNZ R0,LM21

DJNZ R2,LM22

DJNZ R1,L23

RET

;------------定时器1初始化------------

TIME\_INIT1:

MOV TMOD,#90H

MOV TH1,#00H

MOV TL1,#00H

SETB ET1

SETB PT1

RET

;-------------外部 中断1初始化函数------------------------

Exti1\_Init1:

SETB IT1

SETB EX1

SETB EA

CLR TF1

CLR TR1

RET

;-------------阻止外部 中断1函数------------------------

Exti1\_Init11:

CLR IT1

CLR EX1

CLR EA

CLR TF1

CLR TR1

RET

;-------------外部 中断0初始化函数------------------------

Exti0\_Init:

MOV TCON,#01H

SETB EX0

SETB EA

RET

;-----------------阻止外部 中断0函数----------------

Exti0\_Init0:

MOV TCON,#01H

CLR EX0

CLR EA

RET

;-------------定时器1服务函数--------------------

TIME1: ;定时器

DJNZ R2,RETURN

MOV A,R1

MOV DPTR, #KEYT5 ;表头

MOVC A,@A+DPTR ;按下的键值与键值表比较

MOV P0,A

INC R1

MOV R2,#02

CJNE R1,#15,RETURN

MOV R1,#00

DEC R3

CJNE R3,#00,RETURN

CLR ET1

RETURN: RETI

KEYT5: DB 0FEH, 0FDH ;对应键字0,1,

DB 0FBH, 0F7H ;对应键字2,3

DB 0EFH, 0DFH ;对应键字4,5

DB 0BFH, 7FH ;对应键字6.7

DB 0BFH, 0DFH

DB 0EFH, 0F7H

DB 0FBH, 0FDH

DB 0FEH

;-------------中断0服务函数--------------------

INT00:

MOV A,P0

ANL A,#0BFH

MOV P0,A

MOV R1,#03H

L203:

MOV R2,#20H

L200:MOV R0,#50

LM200:

LCALL SMG\_Display3

CLR beep

DJNZ R0,LM200

DJNZ R2,L200

MOV R2,#20H

LM202:

MOV R0,#50

LM201:

LCALL Light\_Off2

SETB beep

DJNZ R0,LM201

DJNZ R2,LM202

DJNZ R1,L203

CLR EX0

RETI

;-------------中断1服务函数---------------------

INT011:

MOV R1,#00

MOV R2,#02

MOV R3,#05

START:

SETB TR1

NOT\_FINISH:

JB TF1,NOT\_FINISH

JNB ET1,K40

AJMP START

K40:

CLR EX1

MOV P0,#00

LCALL Exti0\_Init ;打开中断0

RETI

;-----------延时2ms------------------

delay2ms:

MOV R6,#50

DELAY\_2:

DJNZ R6,DELAY\_2

RET

;-----------------延时20ms--------------------

DLY: MOV R6,#25

DLY0: MOV R7,#248

DLY1: DJNZ R7,DLY1

DJNZ R6,DLY0

RET

;------------延时50ms-------------------------

delay50ms:

MOV R6,#60

DLY101: MOV R7,#248

DLY111: DJNZ R7,DLY111

DJNZ R6,DLY101

;------------延时2s-------------------------

delay2s:

MOV R1,#20

DLY12: MOV R6,#240

DLY10: MOV R7,#248

DLY11: DJNZ R7,DLY11

DJNZ R6,DLY10

DJNZ R1,DLY12

RET

END

## 四、结果分析

由于已经整合出综合模块，故直接使用综合模块进行调试。

测试过程如下：

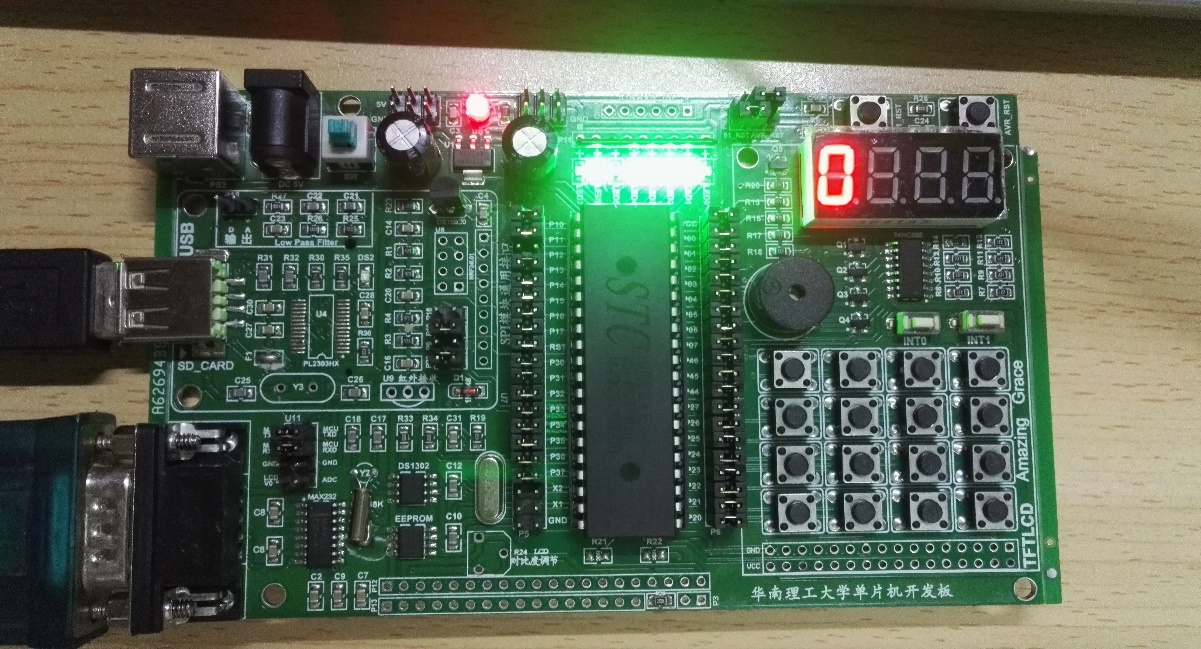
①启动单片机后，依次按下S0、S1、S2、S3，得到如图4.1.1、图4.1.2、图4.1.3、图4.1.4的结果：

图4.1.1

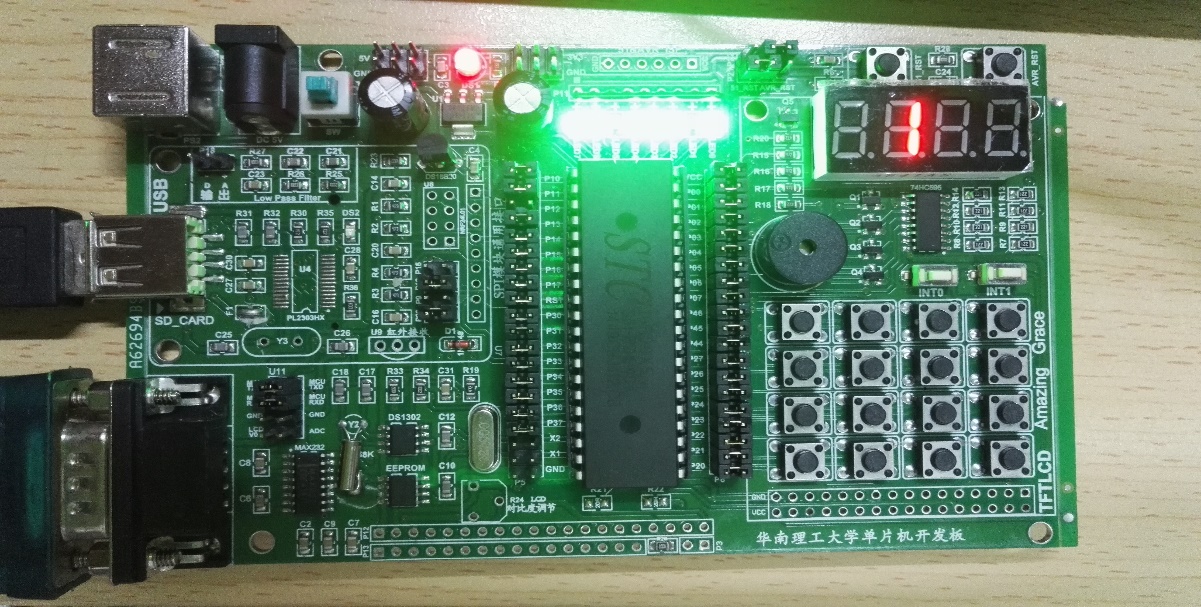


图4.1.2

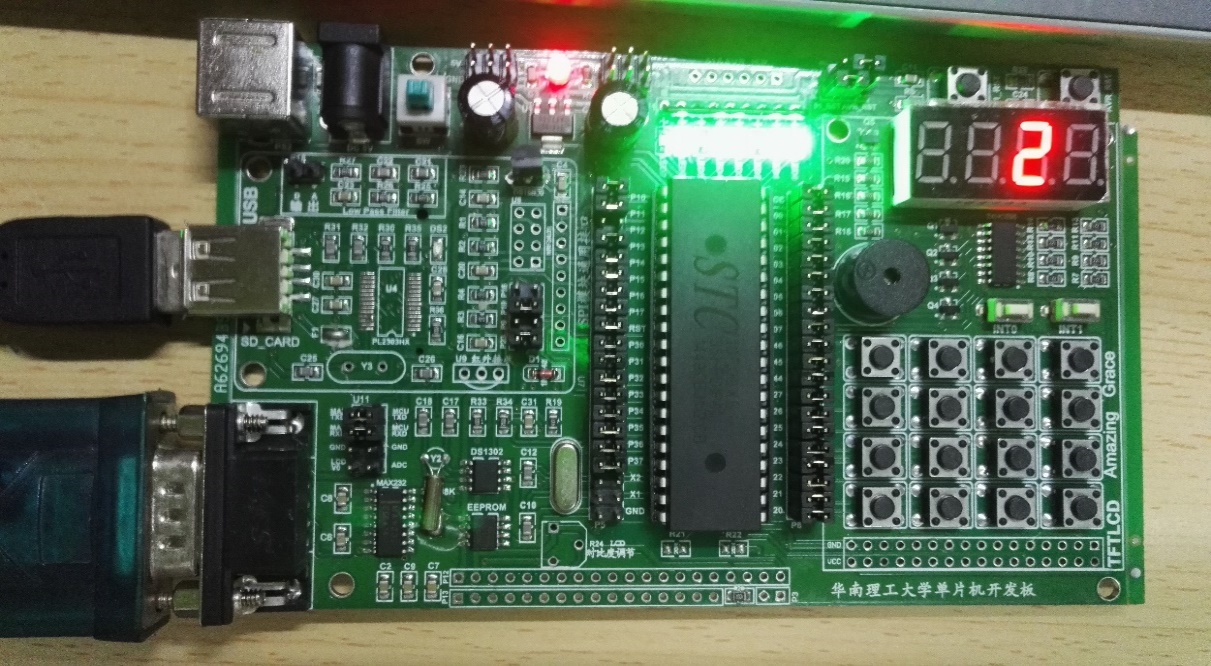
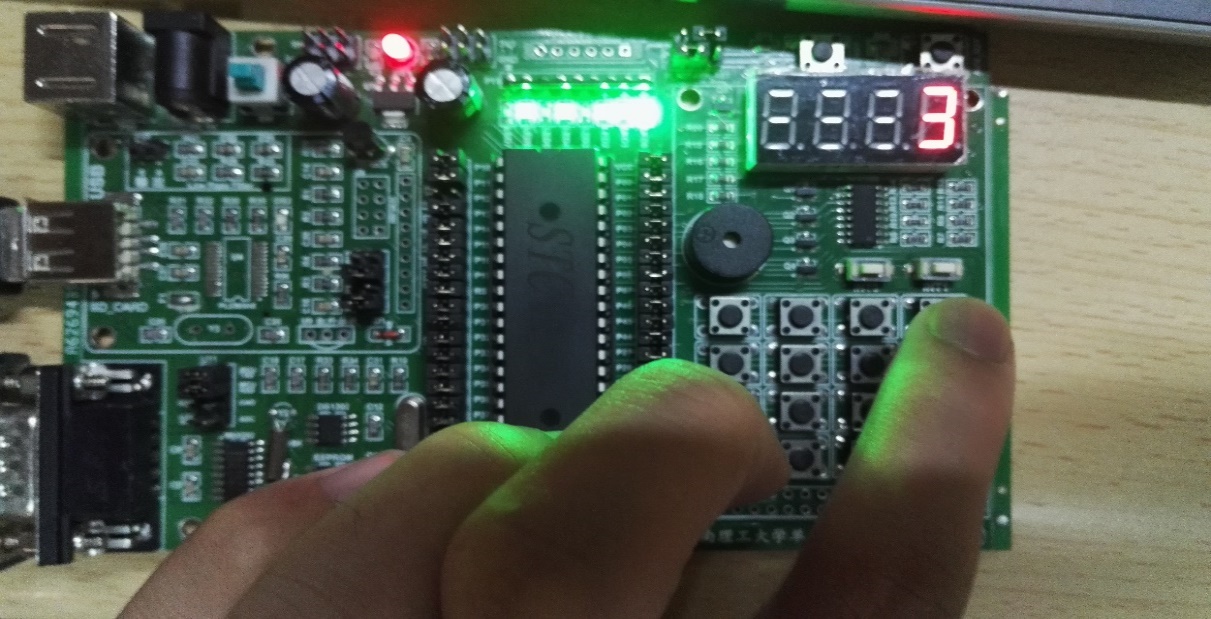
 图4.1.3

图4.1.4

输入这四位密码之后，LED0、LED1、LED2、LED3会被依次点亮，由于是从右往左依次点亮4个连续的LED灯，说明第一层密码输入正确。

若依次按下键S7、S8、S9、S10，数码管会依次显示7、8、9、A，由于8、9、A已经超出了0~7的范围，只点亮了LED7，因此数码管会闪烁“ERR-”三次同时蜂鸣器会发出警告声，如图4.1.5

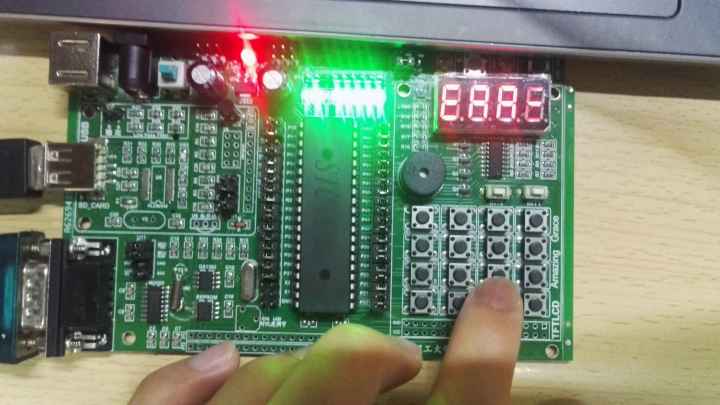
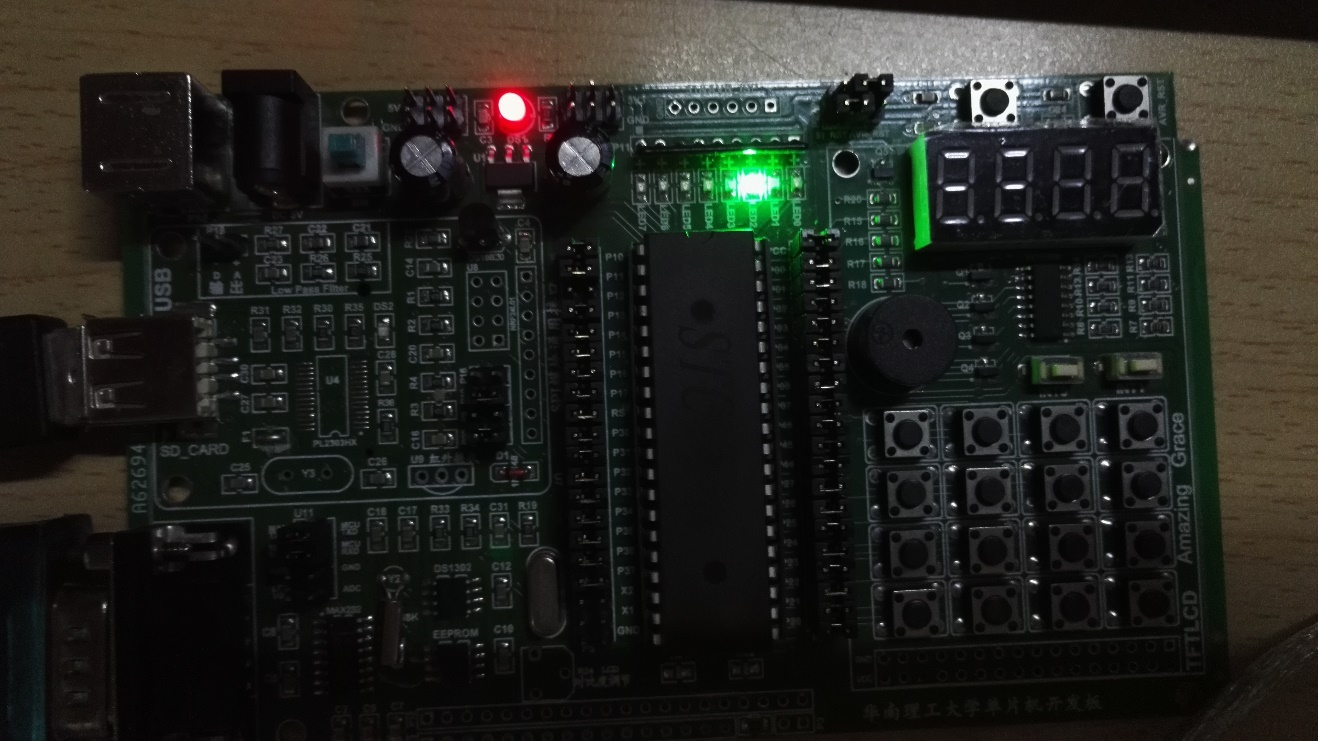


图4.1.5

由于此时没有出现按从右到左的方向依次点亮4个连续的LED灯的现象，故第一层密码输入错误，即使按下KEY1、KEY0，单片机都不会有任何反应。

②当第一层密码输入正确后，观察到有4个连续的LED灯从右到左依次被点亮。此时再按下KEY1键，会观察到LED灯执行走马灯功能，得到如图4.2.1所示的结果：

 图4.2.1

此时LED灯会演示走马灯10次，说明第二层密码生效，否则，若按下KEY1键后没有走马灯，说明第二层密码无效，原因可能是第一层密码输入错误，导致无法输入第二层密码。

③若按下KEY1键后，LED灯会演示走马灯10次，此时再按下KEY0键，数码管会闪烁“YES-”三次，同时蜂鸣器会发出声音，得到如图4.3所示的结果：

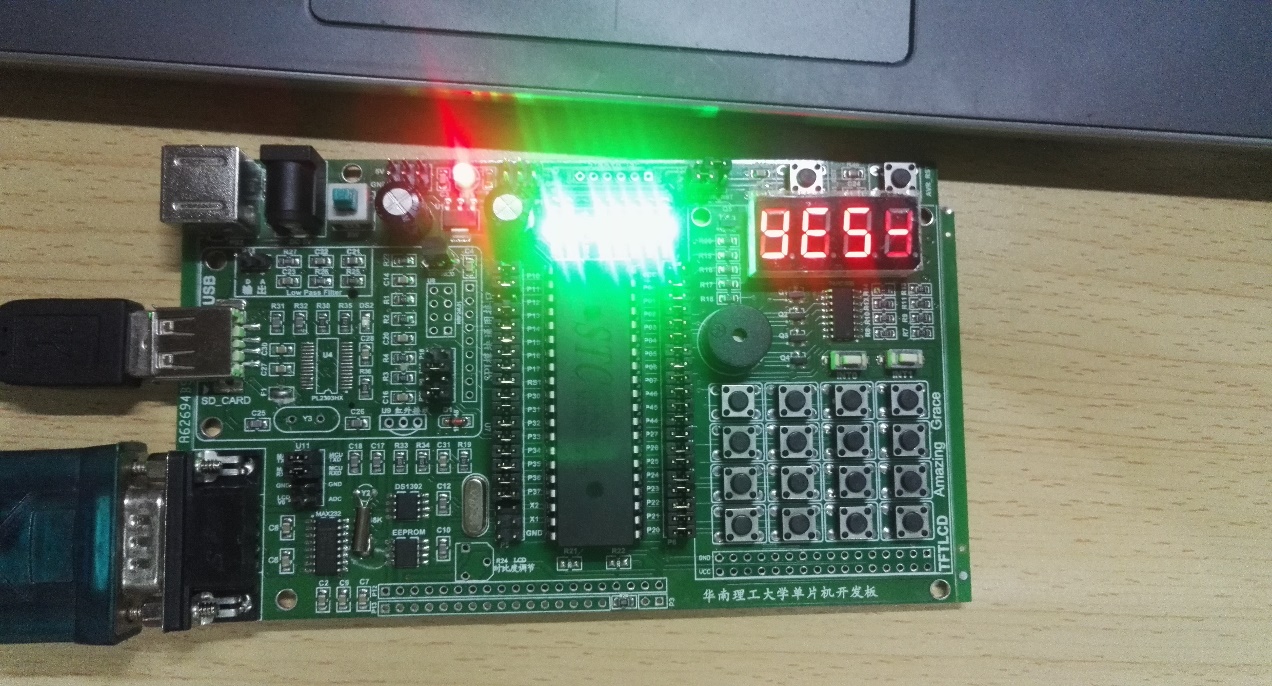


图4.3

此时，按下键KEY0后，单片机有如上相应，说明第三层密码生效，成功打开了密码锁。

因此，可得知密码锁的密码依次为n，n+1，n+2，n+3（n4），KEY1，KEY0时（前四个密码代表的是按键的键值），就可以成功解锁；否则必须重新输入。

## 五、课程设计总结

### 1、心得体会

在本次课程设计中，也算是自己第一次接触到单片机，了解到课本中的东西如何在实际当中去运用，在课上学的比较零散的知识在本次实践中都有了客观的认识，对单片机也有了一个整体的认识。从一开始不了解单片机的运作方式到这一次课程设计中完成并实现了大部分功能，自己还是取得了很大的进步。

在程序设计过程中，也遇到了相当多的问题。例如，在设计显示数码管的时候，没注意到要显示四位数码管的时候需要分别对每一位数码管的端口进行取值，这样就导致了最后只能显示最后一位数码管，其他三位数码管均只显示一会儿就熄灭。后来意识到这个问题，才多加了一个循环模块，让数码管能一直显示对应的数字。还有就是在设计走马灯程序的时候，需要用到定时器的中断以及外部中断功能，但由于未设计中断的优先级，导致在使用外部中断的时候定时器的中断无法触发，导致走马灯未能正确执行。还有就是之前组员用直接地址来存放数据，导致程序一直不能正常执行，经过了长时间的debug，还是未能发现问题，后来索性把程序分为一小部分一小部分单独执行，尝试缩小查找错误的范围，最终觉得是直接地址存放数据的问题，尝试把直接地址换成寄存器，终于解决了问题。在编写程序过程中还遇到过许多问题，从中感受到的是用汇编来编写代码需要注意的细节十分多，稍微不注意就会犯错，寄存器的使用也要十分留意，一不小心就在某一处改变了其中的值。而且，这次实验中也感受到分析问题能力的重要，程序编写后一般都不能达到预期效果，这时候要通过观察现象以及查看代码一步步检查，在这次实验中学会的是将一个程序分为几个小模块去调试执行，观察各部分功能是否正常，以此缩小错误的范围。这其中也感到单片机比较麻烦的一个就是它不能单步调试，查看各个寄存器的值，这也让调试变得比较艰难。

总之，这次课程设计还是让我对单片机有了个整体的认识，学会如何通过编写程序去实现你想要实现的功能，对以后的学习也是有较大的帮助。

### 2、关于《微机系统与接口》课程建议

在微机课程中，我觉得可多安排一些实践的内容，才能加深一些知识的理解，这比单纯的课本理论讲解的效果应该会好很多，课本的理论内容毕竟有些枯燥乏味，对于学生理解起来也比较困难，而实践可以充分调动起我们的兴趣，让我们在实践中去学习，在实践过程中才能注意到一些不容易被注意到的内容，。个人觉得，像此次的课程设计还是能让我们在实践中学到平时理论课没有学到的东西，在课程中多增加一些与实际相结合的东西可能会有更好的效果，这让我们能将学到的内容运用在实际当中，加深理解。

### 3、自我评价：良好

## 参考文献

[1]林土胜·单片机技术及工程实践·北京：机械工业出版社，2011.6