西安财经大学

XI'AN UNIVERSITY OF FINANCE AND ECONOMICS

嵌入式操作系统大作业

题目：基于 STC89C52 单片机的电梯控制系统设计

——软件部分

学生姓名： 陈伯硕

学　　号： 1831050010

专　　业： 计算机科学与技术

班　　级： 计本 1801

指导教师： 魏晋雁

完成日期：2021 年 6 月 25 日

《嵌入式操作系统大作业》成绩评定表

班级：计本 1801　　姓名：陈伯硕　　学号：1831050010　　阅卷教师签名：

设计任务	评分项	分值	子项评分标准及分值	得分
设计实现部分	总体设计	20	任务分析明确（5）	
			方案设计合理有新意（10）	
			软件和硬件功能划分合理（5）	
	硬件设计	20	片内器件分配正确、合理（5）	
			电路原理图设计正确（10）	
			电路图布线整齐、合理（5）	
	软件设计	20	算法和数据结构设计正确、合理（5）	
			流程图设计正确、简明（5）	
			程序设计正确、有新意（10）	
	设计实现	20	调试顺序正确（5）	
			能熟练排除错误，回答问题正确（10）	
			调试后运行正确（5）	
设计报告部分	设计报告	20	书写规范、整齐（5）	
			内容详实具体（5）	
			图形绘制正确、完整、全面（5）	
			能正确客观评价分析设计结果（5）	
总成绩				

目 录

一、设计要求及性能指标

1.1 设计要求

设计一款能设定密码的电子智能密码锁。

1、 能输入 6 位数字密码，显示输入数字个数，但不显示密码，能删除并重新输入，输入范围为数字 0-9(不接受英文字母密码)。

2、有二次输入再确认密码功能。

3、掉电以后密码不会丢失，3 次输入错误则会报警。

1.2 主要性能指标

1.2.1 采用 AT89C52 作为主控制芯片

AT89C52 片内 ROM 全部采用 Flash ROM，能在 3V 的超低压下工作，同时也与 MCS-51 系列单片机完全兼容，该芯片内部存储器为 8KB ROM 存储空间，同样具有 AT89C51 的功能，且具有在线编程可擦除技术[1]，当在对电路进行调试时，由于程序的错误需要修改或对程序进行增加功能需要烧入程序时，不需要对芯片多次插拔，所以不会对芯片造成损坏。

1.2.2 采用 LCD 液晶显示器

LCD 液晶显示器的显示功能强大，可显示大量文字、图形，显示多样、清晰可见。

二、系统总体方案设计

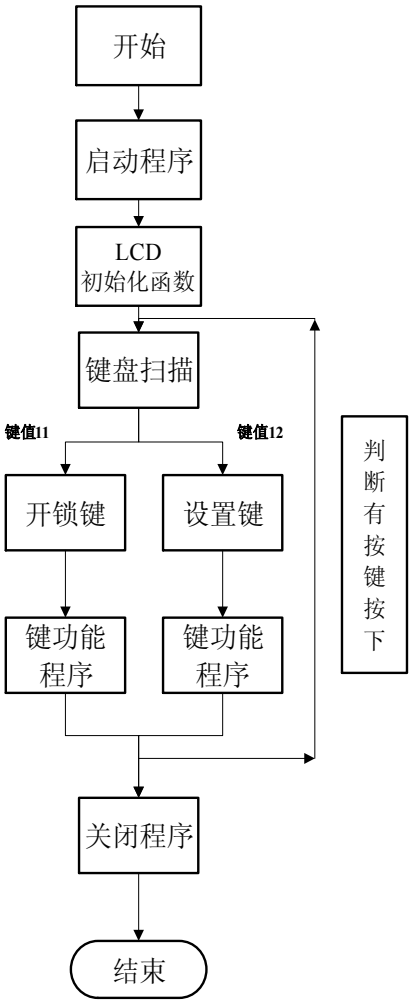
2.1 总体设计思想

本设计选用单片机 AT89C52 作为本设计的核心元件，利用单片机灵活的编程设计和丰富的 I/O 端口，及其控制的准确性，实现基本的密码锁功能[2]。在单片机的外围电路外接输入键盘用于密码的输入和一些功能的控制，外接 LM016L 显示器用于显示作用。当用户需要开锁时，先按键盘开锁键之后按键盘的数字键

0－9 输入密码。密码输完后按下确认键，如果密码输入正确则开锁，不正确显示密码错误重新输入密码，当三次密码错误则发出报警；当用户需要修改密码时，先按下键盘设置键后输入原来的密码，只有当输入的原密码正确后才能设置新密码。新密码输入无误后按确认键使新密码将得到存储，密码修改成功。

2.2 系统总体框图设计

本系统主要由单片机、矩阵键盘、液晶显示器和密码存储等部分组成。其中矩阵键盘用于输入数字密码和进行各种功能的实现。由用户通过连接单片机的矩阵键盘输入密码，后经过单片机对用户输入的密码与自己保存的密码进行对比，从而判断密码是否正确，然后控制引脚的高低电平传到开锁电路或者报警电路控制开锁还是报警。



2.3 系统总体方案的确定

2.3.1 方案一：采用数字电路控制

用以 74LS112 双 JK 触发器构成的数字逻辑电路作为密码锁的核心控制，共设了 9 个用户输入键，其中只有 4 个是有效的密码按键，其它的都是干扰按键，若按下干扰键，键盘输入电路自动清零，原先输入的密码无效，需要重新输入；如果用户输入密码的时间超过 10 秒（一般情况下，用户不会超过 10 秒，若用户觉得不便，还可以修改）电路将报警 20 秒，若电路连续报警三次，电路将锁定键盘 2 分钟，防止他人的非法操作。采用数字电路设计的方案好处就是设计简单但控制的准确性和灵活性差。故不采用。

2.3.2 方案二：采用以单片机为核心的控制方案

电子密码锁所用元件是单片机的硬件和软件相结合的方法，利用了 C 语言的强大功能，通过编写一个合适的正确的程序，依靠所接的按键开关输入相应的指令就可以进行一系列的程序操作，从而实现所需要的功能。电路节省了大量的硬件电路设计过程，使得硬件电路的焊接对设计的要求和结果的影响达到最低的限度，而 52 系列单片机所用元件简单，成本也较低；电路不是很复杂，易于焊接；如果在电路的测试过程中出现了一些问题，可以很容易的检查出来[3]。

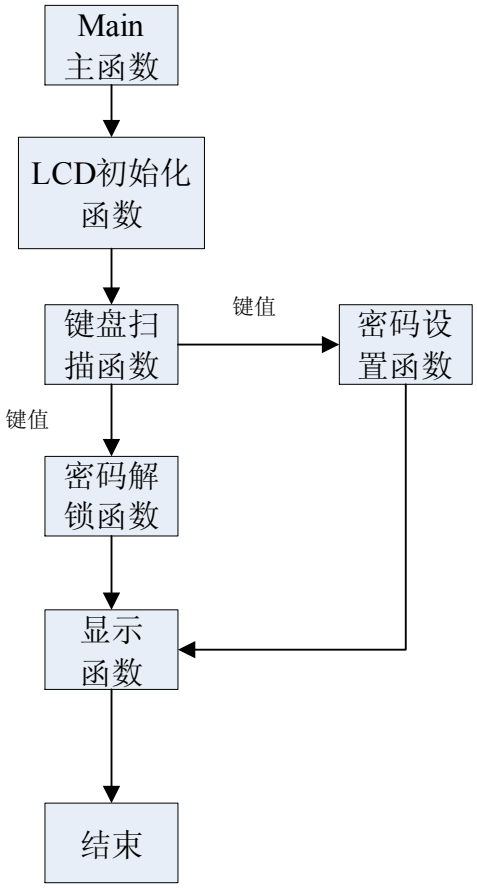
由于单片机种类繁多，各种型号都有其一定的应用环境，因此在选用时要多加比较，合理选择，以期获得最佳的性价比。一般来说在选取单片机时从下面几个方面考虑：性能、存储器、运行速度、I/O 口、定时/计数器、串行接口、模拟电路功能、工作电压、功耗、封装形式、抗干扰性、保密性，除了以上的一些的还有一些最基本的比如：中断源的数量和优先级、工作温度范围、有没有低电压检测功能、单片机内有无时钟振荡器、有无上电复位功能等。在开发过程中单片机还受到：开发工具、编程器、开发成本、开发人员的适应性、技术支持和服务等等因素。基于以上因素本设计选用单片机 AT89C52 作为本设计的核心元件，利用单片机灵活的编程设计和丰富的 I/O 端口，及其控制的准确性，实现基本的密码锁功能。在单片机的外围电路外接输入键盘用于密码的输入和一些功能的控制，外接 LCD1602 显示器用于显示作用。当用户需要开锁时，先按键盘开锁键之后按键盘的数字键 0－9 输入密码。密码输完后按下确认键，如果密码输入正确则开锁，不正确显示密码错误重新输入密码，当三次密码错误则发出报警；当用户需要修改密码时，先按下键盘设置键后可以设置新密码。新密码输入无误后按确认键使新密码将得到存储，密码修改成功。

可以看出方案二控制灵活准确性好且保密性强还具有扩展功能，根据现实生活的需要此次设计采用此方案。

三、系统软件设计

3.1 主程序设计思想及流程图

本系统软件设计由主程序、初始化程序、LCD显示程序、键盘扫描程序、键功能程序、密码设置程序、EEPROM读写程序和延时程序等组成[4]。如图为主程序流程图：



主程序流程图

3.2 初始化程序设计

对 LCD 液晶屏进行初始化，设置其显示模式，以及显示字符时光标的移动设

置和清屏、关闭。

```
void init()                    //LCD 初始化
{
    lcden=0;
    write_com(0x38);           //显示设置
    write_com(0x0c);           //显示开及光标设置
    write_com(0x06);           //显示光标移动设置，初始为 06H
    write_com(0x01);           //显示清屏
    write_com(0x80);           //显示关闭
}
```

3.3 LCD 显示子程序设计

switch 选择语句有两种情况，输入密码时，光标指针会从 LCD 第二行最右开始显示*，下次输入时，指针减一，向左移一位。密码输入完成后，*会全部清空(输出空格)。

```
    switch(b)
    {
        case 0:{                    //模式 0 时输入密码，输入密码时，LCD 显示*
                write_com(0x80+0x4f-symbol);
                write_data('*');
                symbol++;
               };break;
        case 1:{                    //模式 1 时输出空格
               for(i=0;i<16;i++)
                {//通过 write_com 对光标指针的移动来进行换行输出

                    write_com(0x80+0x40+i);
                    write_data(' '); //显示空格,清屏
                    Delay(2);
                }
                symbol=0;
               };break;
     }
```

3.4 键盘扫描子程序设计

按键按下时，与此键相连的行线与列线导通[6]，行线在无键按下时处在高电平。显然，如果让所有的列线也处在高电平，那么，按键按下与否不会引起行线电平的变化，因此，必须使所有列线处在低电平。只有这样，当有键按下时，该键所在的行电平才会由高电平变为低电平。[7]CPU 根据行电平的变化，便能判定相应的行有键按下，因而在函数中置 P0.0-P0.3 依次移位置低电平，等待按键按下时使得 P0.7-P0.4 口有低电平，由此可获得键值并返回。

```
while(1) //键盘扫描，一次扫描，有键按下才跳出
{
    P1=0xff;      //先向 P1 口写 1;端口读状态
    i=0xfe;
    if(U_Interrupt==1)   return   0x20;            //中断标志位
   for(j=0;j<4;j++)
   {
       P1=i;
       k=P1&0xf0;
       if(k!=0xf0)
       {
          Delay(5);
          if(k!=0xf0)//延时去抖动后再判断是否是键按下
          {
            t=k|(i&0x0f);
            for(m=0;m<16;m++)        //哪个键按下
            if(jian[m]==t)   break;
            while(k!=0xf0)
            {
                k=P1;
                k=k&0xf0;
            }
            Delay(5);
            while(k!=0xf0)
            {
                k=P1;
```

```
                    k=k&0xf0;
                }
                return m;       //返回键值
            }
        }
        i=(i<<1)|0x01;
    }
}
}
```

3.4.1 按键功能程序

进行键盘的行列扫描时，会返回一个键值，当键值为 0-9 时为密码输入；当键值为 15 时，会清空*，完成删除功能；[8]键值为 14 时，就会清屏退出，以便下次输入；键值为 11 时为设置键，用于用户设置新密码值。

```
if(Zhi==15)             //删除键按下
    {
        Xian_shi(1);    //清除*号
        Wei=0;              //重新存输入数值
        symbol=0;
        continue;
    }
if(Zhi==14)         //直接按下退出键退出
    {
        Txet=0;
        TR1=0;
        write_com(1);
        for(j=0;j<12;j++)
        write_data(table4[j]);    //显示"Is exiting"
        Delay(2000);   //延迟 2s 后清屏
        write_com(1);
        Wei=0;                  //下次重新存数值
        symbol=0;           //符号位归零，以便下一次输入显示
        return;
    }
```

3.5 密码设置程序

用户按下设置键时(键值为 11)，通过键盘扫描函数得到键值，屏幕显示”please input ”(定义数组存入每个字符)输入旧密码，并进行再次确认后，屏幕显示”please input new code:”提示输出新密码值并从键盘中获得键值存入新密码数组中，经两次密码验证一致(定义一个标志 new 用于标记新密码)后，屏幕显示”sucessfully”表示密码。通过行列扫描输入密码，并同时判断键值，若 0-9 时为正常输入密码，而其他特殊键功能对所设置密码进行存储或清空退出等控制[9]。

bb[6],cc[6] //设置 bb 数组为存储输入数据的数组,new 为新密码标志位 (若为 1 则为输入新密码),cc 数组为存放新密码数组。

```
while(1)
{
    Zhi=keyscan();    //返回键值为 zhi
```

获取键值来判断是正常的密码输入还是特殊键被按下：

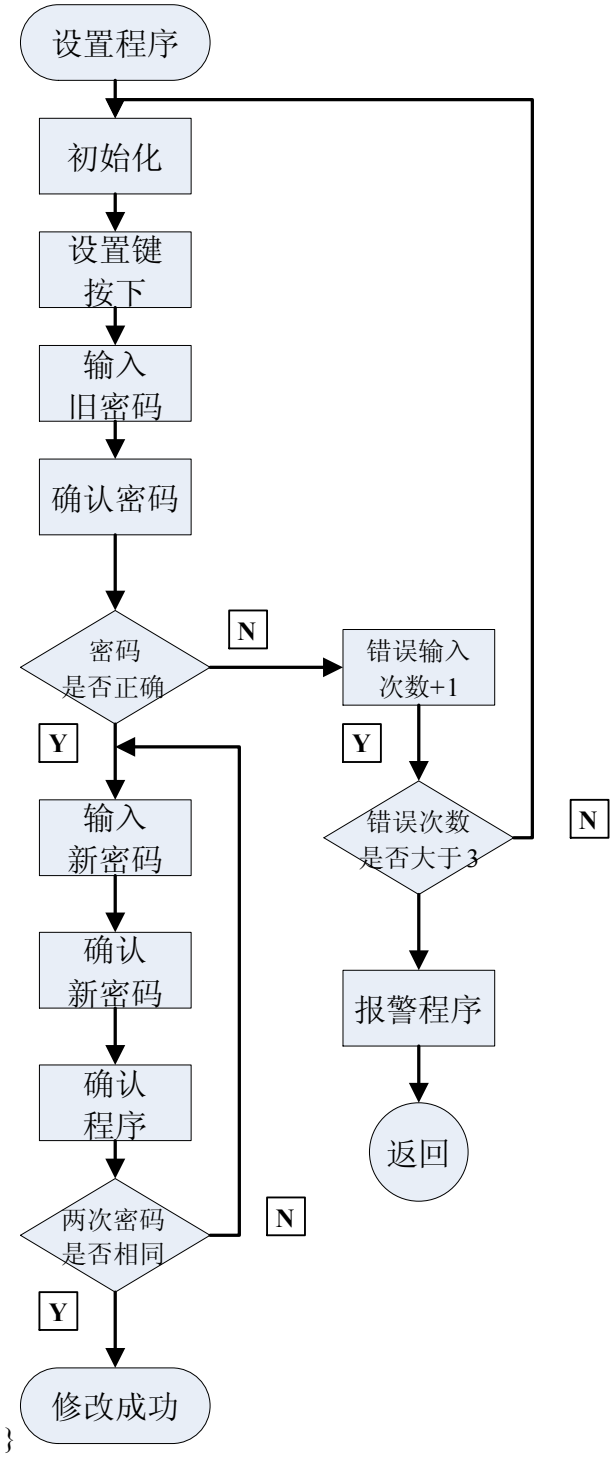
```
    if(Zhi>=0&&Zhi<=9)      //0~9 密码按键
    {
        Show(0);          //显示*号
        if(New==0)
        bb[Wei++]=Zhi;      //存输入数值
        if(New==1)
        cc[Wei++]=Zhi;
        continue;
    }
    if(Zhi==15)              //删除键按下
    {
        Show(1);    //清除*号
        Wei=0;                  //重新存输入数值
        symbol=0;               //置输出控制位为 0
        continue;
    }
    if(Zhi==10||Zhi==11||Zhi==12)
        continue; //除确定,退出,删除,0~9 按键外，其他按键则跳出循环
```

```
if(Zhi==14)          //直接按下退出键退出
{
    Txet=0;
    TR1=0;
    write_com(1);    //清空显示屏
    for(j=0;j<12;j++)
    write_data(table4[j]);    //显示"Is exiting"
    Delay(2000);    //延迟 2s 后清屏
    write_com(1);
    Wei=0;                    //下次重新存数值
    symbol=0;            //符号位归零，以便下一次输入显示
    return;
}
if(Zhi==13)          //确定键按下
{
    New++;
    if(New==1)                    //第一次输入完新密码后，再输入一次
    {
        write_com(1);
        for(j=0;j<15;j++)
        write_data(table8[j]);   //清屏显示 table8[]="input new again"
        Wei=0;
        symbol=0;
        continue;
    }
```

在两次密码不一致错误后后获取键值来判断是退出程序还是确定后重新输入密码：

```
        i=keyscan();    //获取键值
        if(i==14)            // 退出键按下，跳出循环
        {
            Txet=0;
            TR1=0;              //关中断
            write_com(1);
            for(j=0;j<12;j++)
```

```
        write_data(table4[j]); //显示"Is exiting"
        Delay(2000);
        write_com(1);
        Wei=0;
        symbol=0;
        return ;
    }
    if(i==13)        //确定键按下，继续输入密码
    {
        write_com(1);
        for(j=0;j<14;j++)
        write_data(table7[j]);        //清屏显示 table7[]="input new code"
        Wei=0;                        //重新存键值
        symbol=0;                //重新输出符号
        New=0;
        continue;
    }
```



设置密码流程图

3.6 密码解锁子程序设计

在程序一开始，只有开锁键和设置键按下开锁流程才能开始，当开锁键被按下时，程序获取键值并输出”please input :”提示输入密码，程序将用户输入密码与原始密码进行匹配，并要求用户再次输入确认密码后，两次密码匹配成功并与

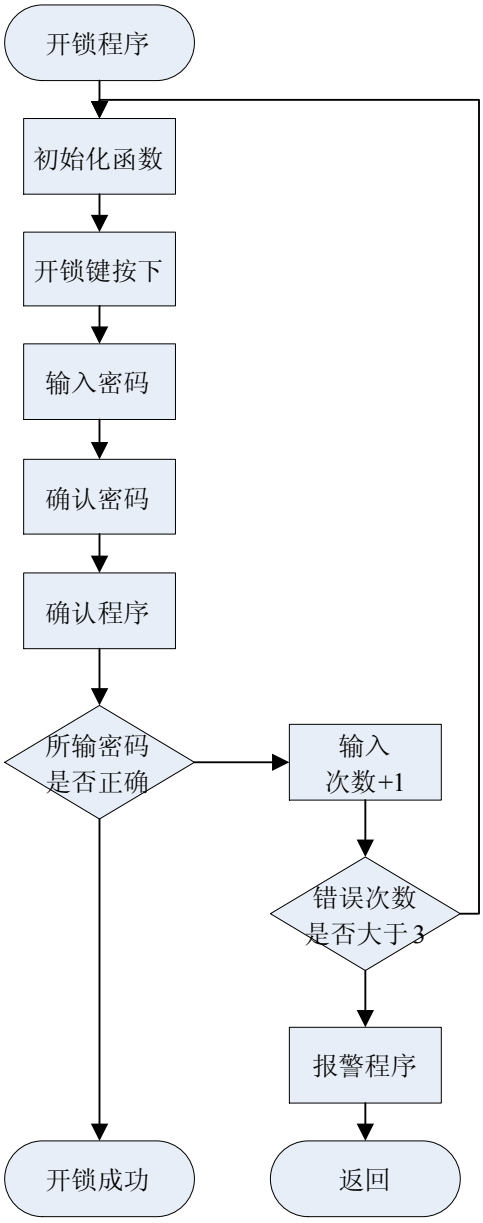
初始密码一致时，屏幕显示”code is right”，否则与原密码不一致时屏幕显示”code is wrong ”提示错误。

```
Times=0;        //输入密码错误次数
while(1)
{
    Zhi=keyscan();   //取键值
    if(Zhi==20)          //若产生了中断信号
    {
        U_Interrupt=0; //置中断标志位为 0
        right=0;
        return 0;
    }
    if(Zhi>=0&&Zhi<=9)       //0~9 密码按键
    {
        Show(0);          //显示*号
        if(right==0) aa[Wei++]=Zhi;       //第一次存输入数值
        if(right==1) bb[Wei++]=Zhi;       //第二次
        continue;
    }
    if(Zhi==15)              //删除键按下(全部删除)
    {
        Show(1);     //清除*号
        Wei=0;                  //重新存输入数值
        continue;
    }
    if(Zhi==10||Zhi==11||Zhi==12)    continue ;
    if(Zhi==14)          //直接按下退出键退出
    {
        Txet=0;          //暂停中断，下次重新计时
        TR1=0;
        write_com(1);
        for(i=0;i<12;i++)
        write_data(table4[i]);     //清屏显示"Is exiting!"
        Delay(2000);         ////延迟 2s 后清屏
        write_com(1);
        right=0;
        return 0;
    }
    if(Zhi==13)            //确定键按下
    {
        write_com(1);
        for(i=0;i<11;i++)
        write_data(table5[i]);     //清屏显示"Please Wait
```

```
        Delay(1000);
```
用 right 标志来判断输入密码与原始密码是否一致：
```
    if(right==0)
      {
        for(k=0;k<6;k++)
        if(aa[k]!=Users[k]) break;
      }
      if(right==1)
      {
        for(k=0;k<6;k++)
        if(bb[k]!=Users[k]) break;
      }
```
当密码错误后，times 标志来判断输入的错误次数，如若超过三次，则置 P3.1 口报警指示灯亮 ：
```
      if(k<6)                          //密码错误
      {
        Times++;
        if(Times==3)    //输入错误达到三次
        {
          Txet=0;
          TR1=0;
          write_com(1);
          write_com(0x80+4);
          for(i=0;i<8;i++)
          write_data(table13[i]);   //清屏显示 Warning!
          write_com(0x80+0x40);
          for(i=0;i<15;i++)
          write_data(table1[i]); //清屏显示 Code is wrong !"
          Warning=1;                  //开报警
          timer_1=1;
          Delay(2500);
          write_com(1);
          right=0;              //清除跳出前输入密码正确的次数
          return 0;
        }
        if(Times<3)           //错误次数小于三次时
        {
          write_com(1);
          for(i=0;i<15;i++)
          write_data(table1[i]); //清屏显示 Code is wrong"
          write_com(0x80+0x43);
          for(i=0;i<13;i++)
          write_data(table6[i]); //清屏显示 Input again ?"
          j=keyscan();
```

```
            if(j==20) //若键值为 20 时,检测到中断
            {
                U_Interrupt=0;
                right=0;
                return 0;
            }
            if(j==14)              // 退出键按下，跳出循环
            {
                Txet=0;
                TR1=0;
                write_com(1);
                for(i=0;i<12;i++)
                write_data(table4[i]);   //退出显示“Is Exiting!”
                Delay(5000);
                write_com(1);
                right=0;
                Wei=0;
                symbol=0; //指针归零，下次从头开锁显示*
                return 0;
            }
            if(j==13)        //确定键按下，继续输入密码
            {
                write_com(1);
                for(i=0;i<12;i++)
                write_data(table3[i]); //清屏显示"Please input"
                Wei=0;                          //重新存键值
                symbol=0;               //重新输出符号
                continue ;
            }
        }
    }
    if(k>=6)             //密码正确
    {
        right++;  //密码必须输入两次正确才行
        if(right==1)
        {
            write_com(1);
            for(i=0;i<11;i++)
            write_data(table6[i]);   //请屏显示 Input again(输入一次成功时)
            Wei=0;
            symbol=0;
            continue ;
        }
        if(right==2)
```

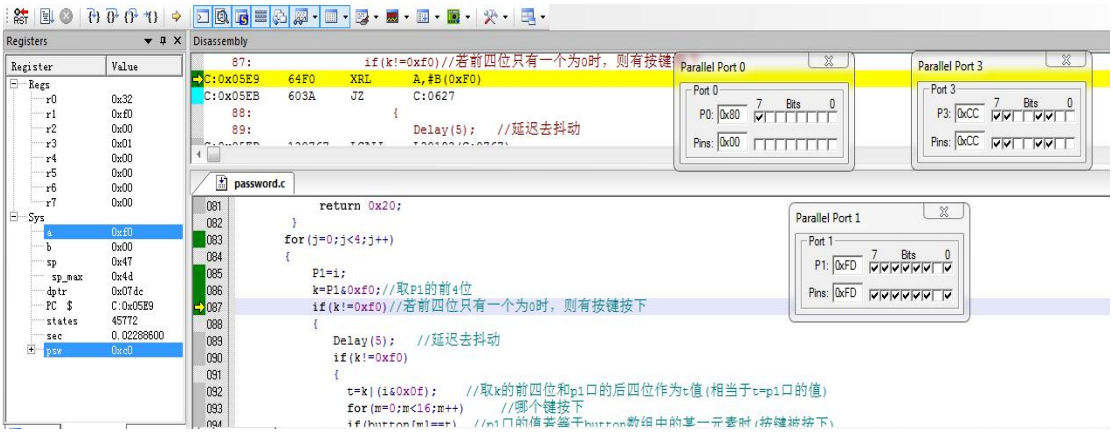
```
                {
                    Txet=0;      //密码正确，关暂停中断，重新计时
                    TR1=0;
                    right=0;
                   Wei=0;
                    symbol=0;
                    return 1;
                }
            }
        }
    }
```



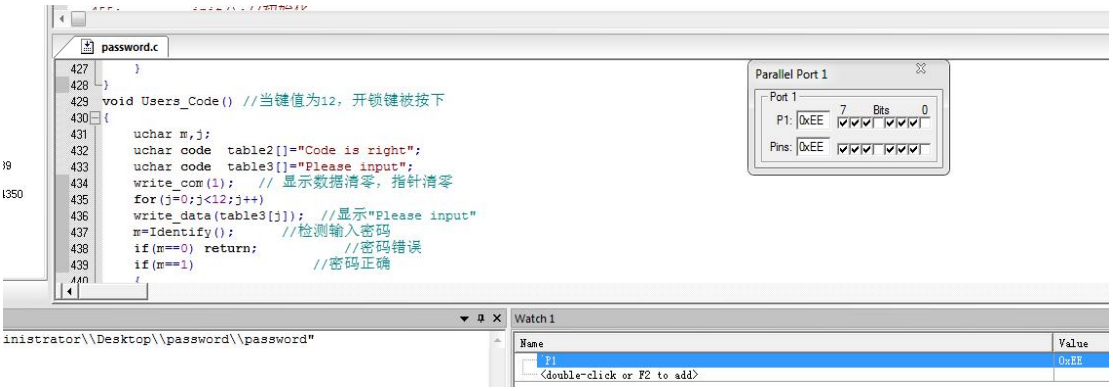
开锁流程图

四、系统调试及实现

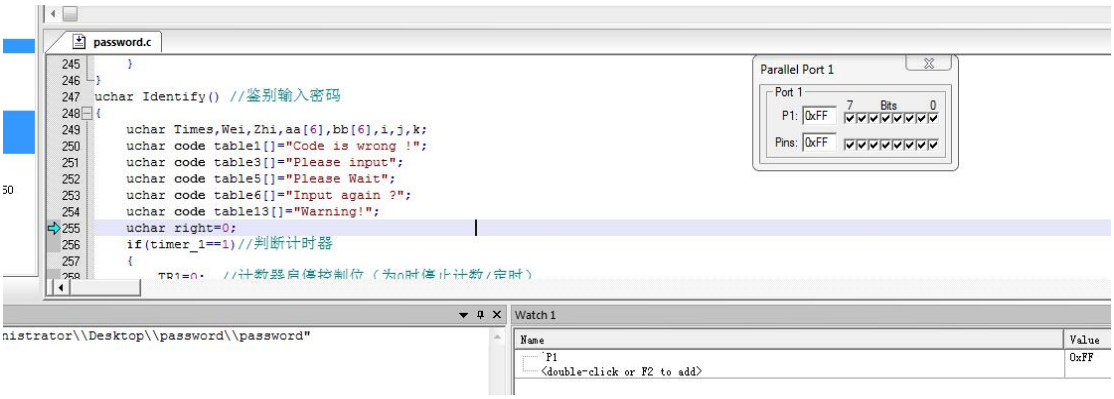
4.1 软件调试



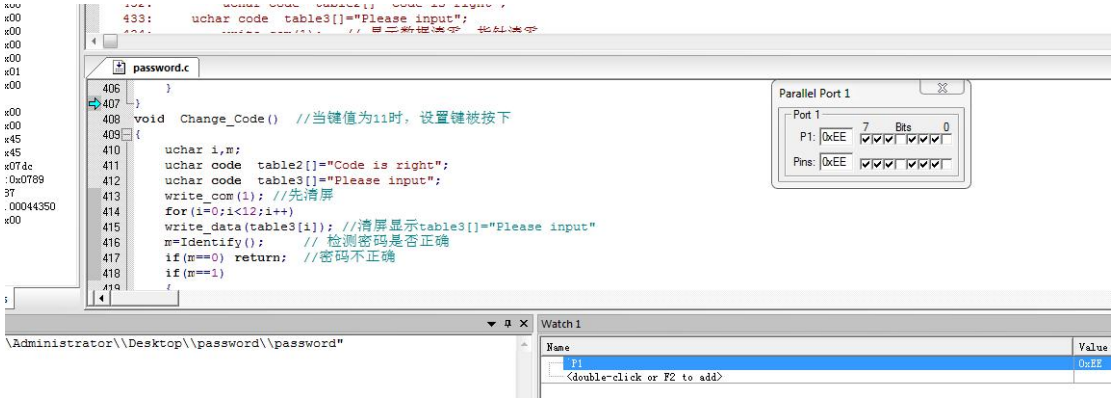
程序进入主函数的键盘扫描函数，置 P1.0-P1.3 口依次为 0



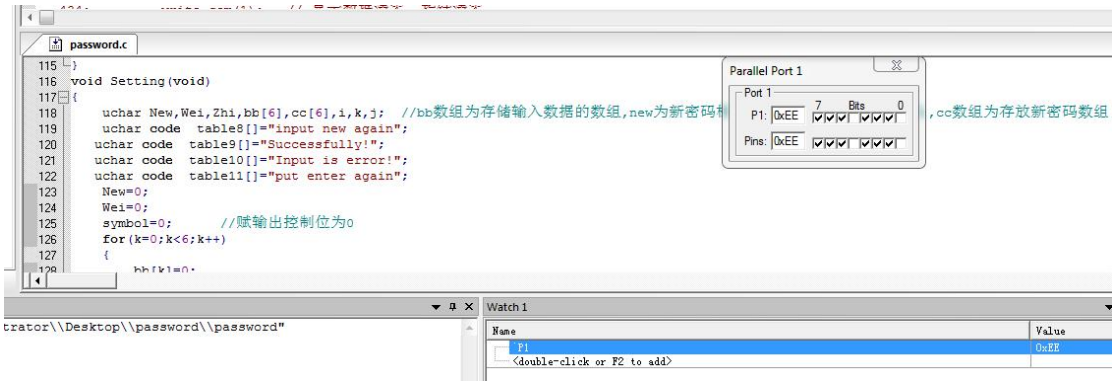
按键为开锁时跳转至 usercode 函数



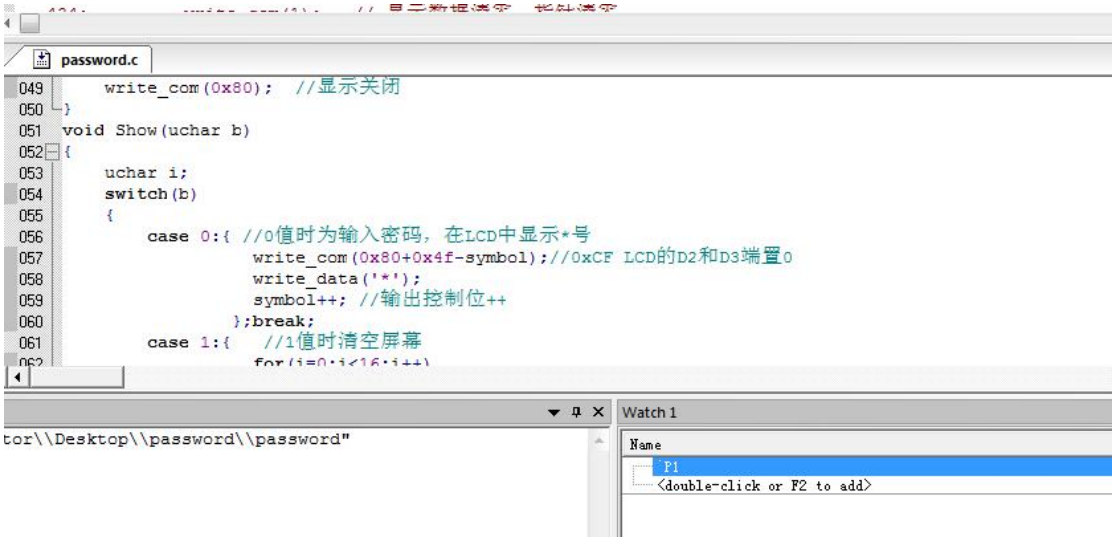
输入密码后跳转至判断密码函数



当设置键被按下时去修改密码



在 setting 函数中修改密码



显示函数

4.2 系统总体性能分析

本设计采用单片机AT89C52作为主控芯片和数据存储器单元，结合外部的键盘输入、显示、报警、开锁等电路并用C语言编写主控芯片的控制程序，研制了一款可以多次更改密码具有报警功能的电子密码锁[12]。虽然可能略显粗糙，不够完美，但是可以达到设计目的。使用单片机制作的电子密码锁软硬件设计简单，

易于开发，成本较低，安全可靠，操作方便。该设计还有按键有效提示，输入错误提示，控制开锁电路，控制报警电路和修改密码等多种功能。

特点：用52单片机来制作密码锁，相对于其他而言，成本较低，安全性高，而且易于操作，很适合没有足够经验的初学者。

实用价值：可以用于住宅、办公室的保险箱及档案柜等需要防盗的场所，可以在意外泄密的情况下随时修改密码。]保密性强，灵活性高，适用于家庭、办公室、学生宿舍等很多地方。

不足及改进方案：复位电路设计存在缺陷，要调整电阻的阻值，经查阅资料得知，复位电路中的电阻阻值应设置为100k即可。

五、设计总结

通过本密码锁的设计，使我对单片机有了更深的了解。特别是键盘扫描方法，以前只是理论上知识，而本次设计让我有机会去实践了这一知识。本设计只是一个简单的密码锁，还有很多功能可以添加上去，密码的明码显示等单片机是一门实用很强的技术，要完全掌握它很难，也需要花很多时间。而单片机课程设计不仅是对单片机掌握程度的一个测试，也对单片机的了解有很大作用。我对单片机有了一定了解。提高了自己的动手能力，也深刻明白了自己的不足。我意识到实践的重要性，同时也学到了很多书本上学不到的知识。

在这次的课程设计过程中，一开始遇到了不少困难，后来一一克服，最终基本成功地实现了密码锁的设计。虽然由于经验不足，设计比较粗糙，但无论是从软件还是硬件方面来说，我都学到了很多东西，积累了很多实践经验，也复习了理论知识，懂得了伙伴之间的合作交流是非常重要的，同时也获得了很大的满足感，对单片机的应用也有了一定的认识与了解，但是我们的设计还不够完善，以后会更加努力。同时也谢谢老师的耐心教导和搭档的配合，在以后的工作学习中，我会继续本着这种不断进取，不断拼搏的精神，努力做好每一件事。

但是由于设计经验不足，在设计上可能略显粗糙，没有做到每个细节都注意到观察到，有些功能键的设置没有足够完善。

这次课程设计的完成，是一个从无到有的过程，经历了兴奋、自信、失落、奋发、所悟、完成几个过程。刚开始做时，以为很难，但后面一步一步下来，查阅资料，老师教导，慢慢懂得一些。一分耕耘，一分收获，有付出才有回报，就在这样的痛苦与快乐的交换中，我学到了知识。

参考文献

[1] 祖龙起,刘仁杰•一种新型可编程密码锁[J].大连轻工业学院学报,2002.

[2] 李瀚荪•电路分析基础[M]，北京：高等教育出版社 1991.

[3] 童诗白,华成英•模拟电子技术基础[M]：高等教育出版社，2000.

[4] 王千•实用电子电路大全[M]，电子工业出版社，2001，p101.

[5] 何立民•单片机应用技术选编[M]，北京：北京航空大学出版社，1998.

[6] 彭为•单片机典型系统设计实例精讲[M]，北京：电子工业出版社，2006.

[7] 潘永雄•新编单片机原理与应用[M]，西安：西安电子科技大学出版社，2003.

[8] 董继成•一种新型安全的单片机密码锁[J].电子技术,2004.

[9] 祖龙起,刘仁杰,孙乃凌•一种新颖的电子密码锁[J].电子世界,2001.

[10] 李明喜•.新型电子密码锁的设计[J].机电产品开发与创新,2004.

[11] 杨茂涛•一种电子密码锁的实现[J].福建电脑,2004.

[12] 楼然苗•51 系列单片机设计实例.[M]，北京：北京航空航天大学出版社，2003

附录　程序清单

```
#include<reg51.h>
#define uchar unsigned char
#define uint unsigned int
uchar
button[16]={0xe7,0xee,0xde,0xbe,0xed,0xdd,0xbd,0xeb,0xdb,0xbb,0x7e,0x7d,0x7b,0x77,0xb7,0
xd7};
uchar Users[6]={0,1,2,3,4,5};        //初始用户密码
uchar symbol=0;                      //符号输出控制位
uchar U_Interrupt=0;                 //中断标志位
uchar Txet=0;
uchar timer_1=0;
uchar code    table4[]="Is exiting !";
uchar code    table7[]="input new code";
uchar code table12[]="Time is too long";
sbit lcden=P3^4;
sbit lcdrs=P3^5;
sbit Warning=P3^1;   //报警指示灯
sbit Lock=P3^0;       //开锁指示灯
void Delay(uint z)
{
      uint x,y;
      for(x=z;x>0;x--)
            for(y=110;y>0;y--);
}
void write_com(uchar com)       //LCD16002 控制指令
{
      lcdrs=0;
      P0=com;
      Delay(5);
      lcden=1;
      Delay(5);
      lcden=0;
}
void write_data(uchar date)     //LCD1602 输出指令
{
      lcdrs=1;
      P0=date;
      Delay(5);
      lcden=1;
      Delay(5);
      lcden=0;
```

```
}
void init()                          //LCD1602 初始化
{
    lcden=0;
    write_com(0x38);
    write_com(0x0c);
    write_com(0x06);
    write_com(0x01);
    write_com(0x80);
}
void Show(uchar b)
{
    uchar i;
    switch(b)
    {
        case 0:{
                    write_com(0x80+0x4f-symbol);
                    write_data('*');
                    symbol++;
                };break;
        case 1:{
                    for(i=0;i<16;i++)
                    {
                        write_com(0x80+0x40+i);
                        write_data(' '); //0x20
                        Delay(2);
                    }
                    symbol=0;
                };break;
     }
}
uchar keyscan(void)     //一次扫描，有键按下才跳出
{
    uchar k,i,j,m,t;
    while(1)
    {
        P1=0xff;
        i=0xfe;
        if(U_Interrupt==1)    return 20;
       for(j=0;j<4;j++)
       {
            P1=i;
            k=P1&0xf0;
            if(k!=0xf0)
```

```
                {
                    Delay(5);
                    if(k!=0xf0)
                    {
                      t=k|(i&0x0f);
                      for(m=0;m<16;m++)         //哪个键按下
                      if(button[m]==t)    break;
                      while(k!=0xf0)
                      {
                            k=P1;
                            k=k&0xf0;
                      }
                      Delay(5);
                      while(k!=0xf0)
                      {
                            k=P1;
                            k=k&0xf0;
                      }
                      return m;       //编译键值
                    }
                 }
                 i=(i<<1)|0x01;
            }
       }
}
void Setting(void)
{
      uchar New,Wei,Zhi,bb[6],cc[6],i,k,j;
      uchar code    table8[]="input new again";
    uchar code    table9[]="Successfully!";
    uchar code    table10[]="Input is error!";
    uchar code    table11[]="put enter again";
      New=0;
      Wei=0;
      symbol=0;
      for(k=0;k<6;k++)
      {
            bb[k]=0;
            cc[k]=0;
      }
      write_com(1);
      for(j=0;j<14;j++)
      write_data(table7[j]);         //清屏显示 table7[]="input new code"
      while(1)
```

```
{
    Zhi=keyscan();
    if(Zhi>=0&&Zhi<=9)      //0~9 密码按键
    {
        Show(0);          //显示*号
        if(New==0)
        bb[Wei++]=Zhi;      //存输入数值
        if(New==1)
        cc[Wei++]=Zhi;
        continue;
    }
    if(Zhi==15)              //删除键按下
    {
        Show(1);    //清除*号
        Wei=0;                  //重新存输入数值
        symbol=0;
        continue;
    }
    if(Zhi==10||Zhi==11||Zhi==12)   continue;
    if(Zhi==14)          //直接按下退出键退出
    {
        Txet=0;
        TR1=0;
        write_com(1);
        for(j=0;j<12;j++)
        write_data(table4[j]);    //显示"Is exiting"
        Delay(2000);   //延迟 2s 后清屏
        write_com(1);
        Wei=0;                    //下次重新存数值
        symbol=0;             //符号位归零，以便下一次输入显示
        return;
    }
    if(Zhi==13)            //确定键按下
    {
        New++;
        if(New==1)                        //第一次输入完新密码后，在输入一次
        {
            write_com(1);
            for(j=0;j<15;j++)
            write_data(table8[j]);   //清屏显示 table8[]="input new again"
            Wei=0;
            symbol=0;
            continue;
        }
```

```
if(New==2)          //第二次输入完成，比较两次是否一样
{
    for(k=0;k<6;k++)
    if(bb[k]!=cc[k]) break;
    if(k>=6)    //两次输入一样
    {
        write_com(1);
        for(j=0;j<13;j++)
        write_data(table9[j]);    //请屏显示 table9[]="Successfully!"
        for(j=0;j<6;j++)
        Users[j]=bb[j]; //更改用户密码
        Delay(2000);
        Txet=0;       //更改密码成功，暂停中断，重新计时
        TR1=0;
        write_com(1);
        for(j=0;j<12;j++)
        write_data(table4[j]);      //显示"Is exiting"
        Delay(2000);
        Wei=0;
        symbol=0;
        write_com(1);
        return;             //跳出循环
    }
    if(k<6)   //两次输入新密码不一样
    {
        write_com(1);
        for(j=0;j<15;j++)
        write_data(table10[j]);         //第一行显示 table10[]="Input is error!"
        write_com(0x80+0x40);
        for(j=0;j<15;j++)             //第二行显示 able11[]="put enter again"
        write_data(table11[j]);
        i=keyscan();
        if(i==14)               // 退出键按下，跳出循环
        {
            Txet=0;
            TR1=0;
            write_com(1);
            for(j=0;j<12;j++)
            write_data(table4[j]); //显示"Is exiting"
            Delay(2000);
            write_com(1);
            Wei=0;
            symbol=0;
            return ;
```

```
                        }
                        if(i==13)       //确定键按下，继续输入密码
                        {
                            write_com(1);
                            for(j=0;j<14;j++)
                            write_data(table7[j]);       //清屏显示 table7[]="input new code"
                            Wei=0;                       //重新存键值
                            symbol=0;              //重新输出符号
                            New=0;
                            continue;
                        }
                    }
                }
            }
        }
    }
uchar Identify()
{
    uchar Times,Wei,Zhi,aa[6],bb[6],i,j,k;
    uchar code table1[]="Code is wrong !";
    uchar code table3[]="Please input";
    uchar code table5[]="Please Wait";
    uchar code table6[]="Input again ?";
    uchar code table13[]="Warning!";
    uchar right=0;
    if(timer_1==1) TR1=0;
    else TR1=1;
    for(k=0;k<6;k++)
    {
        aa[k]=0;//初始化
        bb[6]=0;
    }
    Times=0;                    //输入密码错误次数
    Wei=0;                      //已存入了的密码的个数
    while(1)
    {
        Zhi=keyscan();
        if(Zhi==20)           //产生了中断信号
        {
            U_Interrupt=0;
            right=0;
            return 0;
        }
        if(Zhi>=0&&Zhi<=9)      //0~9 密码按键
```

```
{
    Show(0);          //显示*号
    if(right==0) aa[Wei++]=Zhi;      //第一次存输入数值
    if(right==1) bb[Wei++]=Zhi;      //第二次
    continue;
}
if(Zhi==15)              //删除键按下
{
    Show(1);    //清除*号
    Wei=0;                  //重新存输入数值
    continue;
}
if(Zhi==10||Zhi==11||Zhi==12)   continue ;
if(Zhi==14)          //直接按下退出键退出
{
    Txet=0;         //暂停中断，下次重新计时
    TR1=0;
    write_com(1);
    for(i=0;i<12;i++)
    write_data(table4[i]);    //清屏显示"Is exiting!"
    Delay(2000);         ////延迟 2s 后清屏
    write_com(1);
    right=0;
    return 0;
}
if(Zhi==13)            //确定键按下
{
    write_com(1);
    for(i=0;i<11;i++)
    write_data(table5[i]);    //清屏显示"Please Wait
    Delay(1000);
 if(right==0)
    {
        for(k=0;k<6;k++)
        if(aa[k]!=Users[k]) break;
    }
    if(right==1)
    {
        for(k=0;k<6;k++)
        if(bb[k]!=Users[k]) break;
    }
    if(k<6)                          //密码错误
    {
        Times++;
```

```
if(Times==3)     //输入错误达到三次
{
      Txet=0;
      TR1=0;
      write_com(1);
      write_com(0x80+4);
      for(i=0;i<8;i++)
      write_data(table13[i]);   //清屏显示 Warning!
      write_com(0x80+0x40);
      for(i=0;i<15;i++)
      write_data(table1[i]); //清屏显示 Code is wrong !"
      Warning=1;                  //开报警
      timer_1=1;
      Delay(2500);
      write_com(1);
      right=0;              //清除跳出前输入密码正确的次数
      return 0;
}
if(Times<3)
{
      write_com(1);
      for(i=0;i<15;i++)
      write_data(table1[i]); //清屏显示 Code is wrong"
      write_com(0x80+0x43);
      for(i=0;i<13;i++)
      write_data(table6[i]); //清屏显示 Input again ?"
      j=keyscan();
      if(j==20)
      {
            U_Interrupt=0;
            right=0;
            return 0;
      }
      if(j==14)              // 退出键按下，跳出循环
      {
            Txet=0;
            TR1=0;
            write_com(1);
            for(i=0;i<12;i++)
            write_data(table4[i]);
            Delay(5000);
            write_com(1);
            right=0;
            Wei=0;
```

```
                    symbol=0; //指针归零，下次从头开锁显示*
                    return 0;
                }
                if(j==13)       //确定键按下，继续输入密码
                {
                    write_com(1);
                    for(i=0;i<12;i++)
                    write_data(table3[i]); //清屏显示"Please input"
                    Wei=0;                  //重新存键值
                    symbol=0;              //重新输出符号
                    continue ;
                }
            }
        }
        if(k>=6)            //密码正确
        {
            right++;  //密码必须输入两次正确才行
            if(right==1)
            {
                write_com(1);
                for(i=0;i<11;i++)
                write_data(table6[i]);   //请屏显示 Input again
                Wei=0;
                symbol=0;
                continue ;
            }
            if(right==2)
            {
                Txet=0;     //密码正确，关暂停中断，重新计时
                TR1=0;
                right=0;
               Wei=0;
                symbol=0;
                return 1;
            }
        }
    }
  }
}
void   Change_Code()
{
    uchar i,m;
    uchar code   table2[]="Code is right";
    uchar code   table3[]="Please input";
```

```
        write_com(1);
        for(i=0;i<12;i++)
        write_data(table3[i]); //清屏显示 table3[]="Please input"
        m=Identify();     // 检测密码是否正确
        if(m==0) return;
        if(m==1)
        {
            write_com(1);
            for(i=0;i<13;i++)
            write_data(table2[i]);     //清屏显示 table2[]="Code is right"
            Warning=0;
            timer_1=0;
            Delay(1000);
            Setting();
        }
}
void Users_Code()
{
        uchar m,j;
        uchar code    table2[]="Code is right";
        uchar code    table3[]="Please input";
        write_com(1); // 显示数据清零，指针清零
        for(j=0;j<12;j++)
        write_data(table3[j]);     //显示"Please input"
        m=Identify();       //检测输入密码
        if(m==0) return;                //密码错误
        if(m==1)                        //密码正确
        {
            write_com(1);
            for(j=0;j<13;j++)
            write_data(table2[j]);     //显示"Code is right"
            Warning=0; //关报警
            timer_1=0;
            Lock=1; //开锁
            Delay(3000);     //5s 后自动关指示灯
            Lock=0;
            write_com(1);
        }
}
void main()
{
        uchar key=0;
        init();
        Warning=0;
```

```
    timer_1=0;
    Lock=0;
    TMOD=0x10;  // T1 方式 1（16 位
    TH1=(65535-50000)/256;//50ms
    TL1=(65535-50000)%256;
    EA=1;                       //开 CPU 中断
    ET1=1;          //开 T0,T1 中断
    TR1=0;              //暂停中断
    while(1)
    {
        key=keyscan();
        if(key==11) Change_Code();   //设置键按下
        if(key==12) Users_Code();      //开锁键按下，进入用户模式
    }
}
timer1() interrupt 3 using 2
{
  uchar j;
  Txet++;      //50ms 自加
  if(Txet==400)              //30s 时间到
  {
    write_com(1);
    for(j=0;j<16;j++)
    write_data(table12[j]);     //显示"Time is too long"
    Warning=1;        //超时报警
    timer_1=1;
    TR1=0;       //停止中断
    Txet=0;
    U_Interrupt=1;
    Delay(2000);
    write_com(1);
  }
   TH1=(65535-50000)/256;   //重装计数初值
    TL1=(65535-50000)%256;
}
```