实验报告

一、实验名称

模拟音乐盒-红外控制风扇

二、实验内容

1、进入状态时，数码管其中一位为0，P1^0接\_4IN，风扇的小电机端子插到 J2的 + 和D 上

2、密码锁，当按初始密码（000000）后进去红外风扇调控，密码锁可重输入和更改密码（但不能用数码管显示密码）

3、进入红外风扇调控，此时密码锁处于封锁状态，密码锁生效必须通过红外线按键退出，通过按键可调风速

4、通过按键可以以该风速旋转播放音乐盒

5、模拟音乐盒（用静电泡沫模拟音乐盒内的视觉效果）

三、相关模块

1、流水灯模块：其中一位流水灯用来表示密码锁的开与关。

2、数码管模块：数码管显示当前风扇的档位。

3、USB模块：外部连接，通电

4、红外线模块：通过红外线模块控制遥控器来控制风速的变化和音乐的播放暂停。

5、蜂鸣器：按键后播放音乐

6、矩阵键盘：模拟密码锁进行解密

7、电机模块：控制小电机模拟风扇，通过4\_IN1控制电机的开与关

8、一个定时器中断，一个外部中断：定时器控制音乐，外部中断控制红外线和风扇调控

四、实验代码

4．1 main函数：

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 【编写时间】： 2018年1月9日  \* 【作 者】： 魏楚楠  \* 【实验平台】： QX-MCS51 单片机开发板  \* 【外部晶振】： 11.0592mhz  \* 【主控芯片】： STC89C52  \* 【编译环境】： Keil uVision4  \* 【程序功能】：1、进入状态时，数码管其中一位为0，P1^0接\_4IN，风扇的小电机端子插到 J2的 + 和D 上。  2、密码锁，当按初始密码（000000）后进去红外风扇调控，密码锁可重输入和更改密码（但不能用数码管显示密码）。  3、进入红外风扇调控，此时密码锁处于封锁状态，密码锁生效必须通过红外线按键退出，通过按键可调风速。  4、通过按键可以以该风速旋转播放音乐盒。  5、模拟音乐盒（用静电泡沫模拟音乐盒内的视觉效果）。  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include "reg52.h" //此文件中定义了单片机的一些特殊功能寄存器  #include "irc.h"  #include "commen.h"  #include "fengshan.h"  #include "mimasuo.h"  #include "yinyue.h"  u8 old1,old2,old3,old4,old5,old6; //原始密码000000  u8 new1,new2,new3,new4,new5,new6; //每次MCU采集到的密码输入  u8 a=16,b=16,c=16,d=16,e=16,f=16;  u8 wei,key,temp;  u16 kk=0;  u8 aa=10,bb=100;  bit allow,genggai,ok,wanbi,retry,close; //各个状态位用于密码锁  sbit DU = P2^6; //数码管段选  sbit WE = P2^7; //数码管位选  sbit led2=P1^2;  sbit led33=P1^3;  unsigned char const discode[] ={ 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F,0x6F,0x40,0x00/\*-\*/};  void keyscan()  {  {  P3=0xfe;  temp=P3;  temp=temp&0xf0;  if(temp!=0xf0)  {  delay(10);  if(temp!=0xf0)  {  temp=P3;  switch(temp)  {  case 0xee:  key=0;  wei++;  break;  case 0xde:  key=1;  wei++;  break;  case 0xbe:  key=2;  wei++;  break;  case 0x7e:  key=3;  wei++;  break;  }  while(temp!=0xf0)  {  temp=P3;  temp=temp&0xf0;  beep=0;  }  beep=1;  }  }  P3=0xfd;  temp=P3;  temp=temp&0xf0;  if(temp!=0xf0)  {  delay(10);  if(temp!=0xf0)  {  temp=P3;  switch(temp)  {  case 0xed:  key=4;  wei++;  break;  case 0xdd:  key=5;  wei++;  break;  case 0xbd:  key=6;  wei++;  break;  case 0x7d:  key=7;  wei++;  break;  }  while(temp!=0xf0)  {  temp=P3;  temp=temp&0xf0;  beep=0;  }  beep=1;  }  }  P3=0xfb;  temp=P3;  temp=temp&0xf0;  if(temp!=0xf0)  {  delay(10);  if(temp!=0xf0)  {  temp=P3;  switch(temp)  {  case 0xeb:  key=8;  wei++;  break;  case 0xdb:  key=9;  wei++;  break;    case 0xbb:  genggai=1;  wei=0;  break;  case 0x7b:  if(allow)  ok=1;  break;  }  while(temp!=0xf0)  {  temp=P3;  temp=temp&0xf0;  beep=0;  }  beep=1;  }  }  P3=0xf7;  temp=P3;  temp=temp&0xf0;  if(temp!=0xf0)  {  delay(10);  if(temp!=0xf0)  {  temp=P3;  switch(temp)  {  case 0xe7:  retry=1;  break;  case 0xd7:  close=1;  break;  }  while(temp!=0xf0)  {  temp=P3;  temp=temp&0xf0;  beep=0;  }  beep=1;  }  }  }  }  void shumima() //对按键采集来的数据进行分配  {  if(!wanbi)  {  switch(wei)  {  case 1:new1=key;  if(!allow) a=17;  else a=key; break;  case 2:new2=key;  if(a==17) b=17;  else b=key; break;  case 3:new3=key;  if(a==17) c=17;  else c=key; break;  case 4:new4=key;  if(a==17) d=17;  else d=key; break;  case 5:new5=key;  if(a==17) e=17;  else e=key; break;  case 6:new6=key;  if(a==17) f=17;  else f=key;  wanbi=1; break;  }  }  }  void yanzheng() //验证密码是否正确  {  if(wanbi) //只有当六位密码均输入完毕后方进行验证  {  if((new1==old1)&(new2==old2)&(new3==old3)&(new4==old4)&(new5==old5)&(new6==old6))  allow=1; //当输入的密码正确，会得到allowe置一  }  }  void main()  {  u8 i;  unsigned char j = 0;  //inittimer1();  WE = 1;  P0 = 0XFE;  WE = 0;  DU = 1;  P0 = discode[j];  \_4IN=0;//使得当前风扇暂停  while(1)  {  keyscan();  shumima();  yanzheng();  if(allow==1) //验证完后，若allow为1，则开锁  {  led2=0;  kk=0;  while(!kk)  {  i=check\_key();//检查红外线按键  switch(i)  {  case(7): {aa=aa-20;j=j-1;P0 = discode[j];jian(aa,bb);}break;  //风扇的减速  case(15): {aa=aa+20;j=j+1;P0 = discode[j];jia(aa,bb);}break;  //风扇的加速  case(9): {beep=1;\_4IN=0;}break;  //用来暂停风扇的运转和音乐的播放  case(43):{kk=1;}break;  //用来退出红外线，进行密码锁操作  case(40):{play(j);}break;  //音乐播放  }  }  if(!genggai)  wanbi=1;  }  if(genggai) //当S16更改密码键被按下，genggai会被置一  {  if(allow) //若已经把锁打开，才有更改密码的权限  {  while(!wanbi) //当新的六位密码没有设定完，则一直在这里循环  {  keyscan();  shumima();  if(retry|close) //而当探测到重试键S18或者关闭密码锁键S19被按下时，则跳出  { wanbi=1;  break;  }  }  }  }  if(ok) //更改密码时，当所有六位新密码均被按下时，可以按下此键，结束密码更改  { //其他时间按下此键无效  ok=0; wei=0;  genggai=0;  old1=new1;old2=new2;old3=new3; //此时，旧的密码将被代替  old4=new4;old5=new5;old6=new6;  a=16;b=16;c=16;d=16;e=16;f=16;  }  if(retry) //当重试按键S18被按下，retry会被置位  {  retry=0; wei=0;wanbi=0;  a=16;b=16;c=16;d=16;e=16;f=16;  new1=0;new2=0;new3=0;new4=0;new5=0;new6=0;  }  if(close) //当关闭密码锁按键被按下，close会被置位  {  close=0;  genggai=0;//所有变量均被清零。  wei=0;  wanbi=0;  allow=0;  led2=1;  \_4IN=0;  a=16;b=16;c=16;d=16;e=16;f=16;  new1=0;new2=0;new3=0;new4=0;new5=0;new6=0;  }  }  } |

4.2 commom函数：延迟函数和数据类型声明（包括.H和.C文件）

|  |
| --- |
| #ifndef \_COMMEN\_H  #define \_COMMEN\_H  #include "reg52.h"  typedef unsigned int u16; //对数据类型进行声明定义  typedef unsigned char u8;  void delay(u16 i);  #endif  #include "commen.h"  void delay(u16 i)  {  while(i--);  } |

4.3 fengshan函数：用于风扇的加速减速（包括.H和.C文件）

|  |
| --- |
| #ifndef \_FENGSHAN\_H  #define \_FENGSHAN\_H  #include "reg52.h"  #include "commen.h"  sbit \_4IN = P1^0;//  void jian(unsigned char a,unsigned char b);  void jia(unsigned char a,unsigned char b);  //void inittimer1();  #endif  /\* 红外无线遥控风扇 + 加速 按- 减速 \*/  #include <fengshan.h>  #include <irc.h>  unsigned char value,count;  unsigned char timer1;  void jia(unsigned char a,unsigned char b)  {  while(check\_key()==15)  {  unsigned char i;  for(i=0;i<a;i++)  {  \_4IN=1;  }  delay(5);  for(i=0;i<b;i++)  {  \_4IN=0;  }  delay(5);  }  }  void jian(unsigned char a,unsigned char b)  {  while(check\_key()==7)  {  unsigned char i;  for(i=0;i<a;i++)  {  \_4IN=1;  }  delay(5);  for(i=0;i<b;i++)  {  \_4IN=0;  }  delay(5);  }  } |

4.4 irc函数：红外线遥控器（包括.H和.C文件）

|  |
| --- |
| #ifndef \_IRC\_H  #define \_IRC\_H  #include "reg52.h"  #include "commen.h"  sbit IRIN=P3^2;  u8 check\_key();  #endif  #include "irc.h"  u8 IrValue[6];  u16 Time;  void IrInit()//外部中断初始化  {  IT0=1;//下降沿触发  EX0=1;//打开中断0允许  EA=1; //打开总中断  IRIN=1;//初始化端口  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 函 数 名 : check\_key  \* 函数功能 : 判断按钮  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  u8 check\_key()//按键检测函数  {  u8 i;  IrInit();  if((IrValue[2]/16)==0 && (IrValue[2]%16)==7)  {  i= 7;  }  if((IrValue[2]/16)==4&&(IrValue[2]%16)==4)  {  i= 44;  }  if((IrValue[2]/16)==4&&(IrValue[2]%16)==0)  {  i= 40;  }  if((IrValue[2]/16)==4&&(IrValue[2]%16)==3)  {  i= 43;  }  if(((IrValue[2]/16 )==1&&(IrValue[2]%16)==9) )  {  i= 19;  }  if((IrValue[2]/16)==0&&(IrValue[2]%16)==9)  {  i= 9;  }  if((IrValue[2]/16)==1&&(IrValue[2]%16)==5)  {  i= 15;  }  return i;  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 函数名 : ReadIr()  \* 函数功能 : 读取红外数值的中断函数  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void ReadIr() interrupt 0  {  u8 j,k;  u16 err;  Time=0;  delay(700); //7ms  if(IRIN==0) //确认是否真的接收到正确的信号  {  err=1000; //1000\*10us=10ms,超过说明接收到错误的信号  /\*当两个条件都为真是循环，如果有一个条件为假的时候跳出循环，免得程序出错的时  侯，程序死在这里\*/  while((IRIN==0)&&(err>0)) //等待前面9ms的低电平过去  {  delay(1);  err--;  }  if(IRIN==1) //如果正确等到9ms低电平  {  err=500;  while((IRIN==1)&&(err>0)) //等待4.5ms的起始高电平过去  {  delay(1);  err--;  }  for(k=0;k<4;k++) //共有4组数据  {  for(j=0;j<8;j++) //接收一组数据  {  err=60;  while((IRIN==0)&&(err>0))//等待信号前面的560us低电平过去  {  delay(1);  err--;  }  err=500;  while((IRIN==1)&&(err>0)) //计算高电平的时间长度。  {  delay(10); //0.1ms  Time++;  err--;  if(Time>30)  {  return;  }  }  IrValue[k]>>=1; //k表示第几组数据  if(Time>=8) //如果高电平出现大于565us，那么是1  {  IrValue[k]|=0x80;    }  Time=0; //用完时间要重新赋值  }  }  }  if(IrValue[2]!=~IrValue[3])  {  return;  }  }  } |

4.5 yinyue函数：播放音乐（包括.H和.C文件）

|  |
| --- |
| #ifndef \_YINYUE\_H  #define \_YINYUE\_H  #include "reg52.h"  #include "commen.h"  #include "mimasuo.h"  void play(u16 aaa);  #endif  #include "yinyue.h"  #include "fengshan.h"  #include <irc.h>  bit flag; //标志音乐输出脚电平的高低  u8 ptr = 0x00; //取音符  u8 high; //计数器高位  u8 low; //计数器低位  u16 zzz;  // 本曲谱为 "新年好",前两个十六进制表示发声频率,后一个表示发声时间,0xFF,0xFF 表示休止符  // 0x00 表示结束  // 1 \_ 1\_ 1 .5  u8 code music[] = {  0xFC,0x44,0x7F, 0xFC,0x44,0x7F, 0xFC,0x44,0xFF, 0xFA,0x68,0xFF,  // 3 \_ 3\_ 3 1  0xFD,0x23,0x7F, 0xFD,0x23,0x7F, 0xFD,0x23,0xFF, 0xFC,0x44,0xFF,  // 1\_ 3\_ 5 5  0xFC,0x44,0x7F, 0xFD,0x23,0x7F, 0xFD,0x82,0xFF, 0xFD,0x82,0xFF,  // 4\_ 3\_ 2 -  0xFD,0x23,0x7F, 0xFD,0x23,0x7F, 0xFC,0xAC,0xFF, 0xFF,0xFF,0xFF,  // 2\_ 3\_ 4 4  0xFC,0xAC,0x7F, 0xFD,0x23,0x7F, 0xFD,0x34,0xFF, 0xFD,0x34,0xFF,  // 3\_ 2\_ 3 1  0xFD,0x23,0x7F, 0xFC,0xAC,0x7F, 0xFD,0x23,0xFF, 0xFC,0x44,0xFF,  // 1\_ 3\_ 2 .5  0xFC,0x44,0x7F, 0xFD,0x23,0x7F, 0xFC,0xAC,0xFF, 0xFA,0x68,0xFF,  // .7\_ 2\_ 1 -  0xFC,0x0C,0x7F, 0xFC,0xAC,0x7F, 0xFC,0x44,0xFF, 0xFF,0xFF,0xFF,  0x00//结束  };  void Init()  {  TMOD = 0x01; //定时器0 处于计时方式,16 位  EA = 1;  ET0 = 1; //定时器0 溢出中断  }  void DelayMs(unsigned int time)//重新设置延迟函数  {  unsigned int i;  unsigned int j;  for (j =0; j < time; j++) //每个循环 约 3ms  {  for (i =0; i < 363; i++)  {;}  }  }  void play(u16 aaa)  { u8 time;  zzz=aaa;  Init();  TH0 = high;  TL0 = low;  while (check\_key()==40)  {  if (music[ptr] != 0xFF && music[ptr] != 0x00)//判断是否是正常音符  {  TR0 = 0;  beep = 1;  DelayMs(10); //间歇  TR0 = 1;  high = music[ptr]; //取设置频率数值的高8 位  low = music[ptr + 1]; //取设置频率数值的低8 位  time = music[ptr + 2]; //取发声时间  DelayMs(time);  ptr += 3;  }  else if (music[ptr] == 0xFF) //判断是否是休止符  {  time = music[ptr + 2];  DelayMs(time);  ptr += 3;  }  else //结束符,停止2 秒后继续  {  TR0 = 0;  beep = 1;  ptr = 0;  }  if(check\_key()!=40)  {ET0=0;}  }  }  void panduan()//风扇调速（音乐内）  { u8 i;  if(zzz==1)  {  for(i=0;i<110;i++)  { }  \_4IN=1;  }  if(zzz==2)  { for(i=0;i<100;i++)  { }  \_4IN=1;  }  if(zzz==3)  { for(i=0;i<90;i++)  { }  \_4IN=1;  }  if(zzz==4)  { for(i=0;i<80;i++)  { }  \_4IN=1;  }  if(zzz==5)  {  \_4IN=1;  }  if(zzz==6)\_4IN=1;  }  void panduan1()  { u8 i;  if(zzz==1) \_4IN=0;  if(zzz==2) \_4IN=0;  if(zzz==3) \_4IN=0;  if(zzz==4) \_4IN=0;  if(zzz==5) \_4IN=0;  if(zzz==6)  { for(i=0;i<110;i++){}  \_4IN=1;  }  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 名称：Count1(void) interrupt 1  \* 功能：设置计时器0 溢出中断,每中断一次改变P2\_3 引脚电平  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void Count1(void) interrupt 1  {  TH0 = high;  TL0 = low;  if (flag == 0) //改变P2\_3 引脚电平  {  beep = 0;  panduan();  flag = 1;  }  else  {  beep = 1;  panduan1();  flag = 0;  }  } |

五、实验效果

视频显示：在附加文件中。

六、实验遇到的问题

1.总体设计时,观察配套材料中红外线的使用使用了一个定时器中断和外部中断，而其原理中的定时器中断对时间的要求及其严苛，后面的功能不好添加，**解决方法**：自己改动后将定时器改为用延迟函数实现的红外线遥控器。

2.在红外线中加入密码锁，由于密码锁与红外线不能不同时处于不断扫描的状态。

**解决方法：**通过加入参数，将两个扫描设置优先级，使得密码锁的优先级高于红外线的优先级，只有先进入密码锁才能进行红外扫描。

3.密码锁的进入后使得所有的流水灯亮，使得内部停止时风扇一直旋转。

**解决方法：**因为控制风扇的端口为P1^0，而流水灯的控制端口为P1，进入密码锁时，流水灯亮，使得P1全部处于高电平，风扇也处于高电平在运转，改动了密码锁进入后，使得流水灯只亮第二个。

4.密码锁进去红外线后退出红外线后第二次有时不能再次进入红外线

**解决方法（未解决）**：原因不明，反复检查后一直无结果，此处处于随机进入红外线状态

5.音乐与风扇冲突，不能同时进行

**解决方法：**由于音乐的播放是使得蜂鸣器在外部中断中不断交换高低电平，所以此处将风扇的控制位也放入了此外部中断，使得其同时传递，而由于参数问题，则设置了传值函数进行了从主函数到音乐函数，从音乐函数到外部中断函数的传值。可是此时由于风扇的加入使得音乐的旋律放慢，不过不影响此实验！

6.密码锁由于也要处于优先检测，但是不在主函数个人无法实现

**解决方法**：将密码锁的函数放入了主函数中实行，可是使得了代码的冗杂。

7.音乐暂停后，蜂鸣器确以同一旋律一直“尖叫”！

**解决方法**：在主函数中设置了beep=1，使得蜂鸣器在主函数中在停止一次来去除噪音。

8．一开始的加速减速本是通过定时器实现，可是冲突不断，一段时间后仍未发现问题，所以将风扇的加速减速通过延迟（加入FOR循环）来实现。