```c
                        //广西柳州市一职校 电气自动化专业部
                        //By: China o soft
#include <reg52.h>      //使用8052内核单片机
#include <stdlib.h>     //使用rand随机数函数
sbit SCL=P1^0;          //IIC时钟线(SCL)定义
sbit SDA=P1^1;          //IIC数据线(SDA)定义
sbit KEY1=P2^0;         //按钮1,用于将地址+1
sbit KEY2=P2^1;         //按钮2,用于将地址-1
sbit KEY3=P2^2;         //按钮3,用于将数据+1
sbit BEEP=P2^3;         //蜂鸣器,当读写测试失败时报警
sbit KEY5=P1^2;         //按钮5,用于读取当前地址数据
sbit KEY6=P1^3;         //按钮6,用于在现地址写入当前数据
sbit KEY7=P1^4;         //按钮7,用于进行256字节的读写测试
sbit KEY4=P1^5;         //按钮4,用于将数据-1
unsigned char code LED_SEG[16]={0x88,0xBE,0xC4,0x94,0xB2,0x91,0x81,0xBC,
                                0x80,0x90,0xA0,0x83,0xC9,0x86,0xC1,0xE1};
                        //数码管段码表0-f;a:D1,b:D0,c:D6,d:D5,e:D4,f:D2,g:D3,dp:D
7
unsigned char D1,D2,D3,D4;
                        //显示在数码管上的值
unsigned char ADDR,INDEX;
                        //当前地址,当前数据变量

void eeprom_write_byte(unsigned char addr,dat)
                        //eeprom写字节,传递参数1:要写的地址,参数2:要写的数据
  {                     //写字节程序开始
  unsigned char temp1,temp2;
                        //写入延时使用的临时变量
  SCL=1;                //时钟线(SCL)拉高
  SDA=1;                //数据线(SDA)拉高
  SDA=0;                //在时钟线(SCL)为高电平时,数据线(SDA)发生下跳变,总线启动

  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //AT24C02eeprom的器件地址(A0-A2接地),1010000x.
  SCL=1;                //读入一位数据(1)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //将数据1放到总线上
  SCL=1;                //读入一位数据(1)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上,此位特殊,为0代表写,为1代表读
  SCL=1;                //读入一位数据(0,写数据)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态
```

```
  SDA=1;                //放开数据线(SDA),以使得eeprom能够应答
  SCL=1;                //发送第九个脉冲,让eeprom应答,应答信号从数据线(SDA)输出
  SCL=0;                //应答结束,如果需要判断是否应答,应检查数据线(SDA)电平,本程
序不检查

  SDA=addr&0X80;        //将地址的最高位(MSB)放到总线上
  SCL=1;                //写入最高位(MSB D7)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X40;        //将地址的D6位放到总线上
  SCL=1;                //写入数据位(D6)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X20;        //将地址的D5位放到总线上
  SCL=1;                //写入数据位(D5)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X10;        //将地址的D4位放到总线上
  SCL=1;                //写入数据位(D4)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X08;        //将地址的D3位放到总线上
  SCL=1;                //写入数据位(D3)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X04;        //将地址的D2位放到总线上
  SCL=1;                //写入数据位(D2)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X02;        //将地址的D1位放到总线上
  SCL=1;                //写入数据位(D1)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X01;        //将地址的(LSB D0)位放到总线上
  SCL=1;                //写入数据位(D0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //放开数据线(SDA),以使得eeprom能够应答
  SCL=1;                //发送第九个脉冲,让eeprom应答,应答信号从数据线(SDA)输出
  SCL=0;                //应答结束,如果需要判断是否应答,应检查数据线(SDA)电平,本程
序不检查

  SDA=dat&0X80;         //将数据的最高位(MSB)放到总线上
  SCL=1;                //写入数据位(MSB D7)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=dat&0X40;         //将数据的D6位放到总线上
  SCL=1;                //写入数据位(D6)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=dat&0X20;         //将数据的D5位放到总线上
  SCL=1;                //写入数据位(D5)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=dat&0X10;         //将数据的D4位放到总线上
  SCL=1;                //写入数据位(D4)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=dat&0X08;         //将数据的D3位放到总线上
  SCL=1;                //写入数据位(D3)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=dat&0X04;         //将数据的D2位放到总线上
  SCL=1;                //写入数据位(D2)
```

```
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=dat&0X02;         //将数据的D1位放到总线上
  SCL=1;                //写入数据位(D1)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=dat&0X01;         //将数据的D0位(LSB)放到总线上
  SCL=1;                //写入数据位(D0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //放开数据线(SDA),以使得eeprom能够应答
  SCL=1;                //发送第九个脉冲,让eeprom应答,应答信号从数据线(SDA)输出
  SCL=0;                //应答结束,如果需要判断是否应答,应检查数据线(SDA)电平,本程
序不检查

  SDA=0;                //拉低数据线(SDA)
  SCL=1;                //拉高时钟线(SCL)
  SDA=1;                //在时钟线(SCL)为高电平时,数据线(SDA)发生上跳变,停止总线

                        //按ATMEL的手册,每次写完,都应该延时5ms以上
  temp1=temp2=20;       //准备延时
  do                    //外层do-while循环
  {                     //外层循环开始
  do                    //内层do-while循环
    {                   //内层循环开始
    ;                   //空操作,延时
    }while(--temp2);    //do-while结构
  }while(--temp1);      //do-while结构

  }                     //写字节程序结束


unsigned char eeprom_read_byte(unsigned char addr)
                        //eeprom读字节,传递参数为要读的地址,返回读到的数据
  {                     //读字节程序开始
  unsigned char temp;   //接收数据用的临时变量

  SCL=1;                //时钟线(SCL)拉高
  SDA=1;                //数据线(SDA)拉高
  SDA=0;                //在时钟线(SCL)为高电平时,数据线(SDA)发生下跳变,总线启动

  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //AT24C02eeprom的器件地址(A0-A2接地),1010000x.
  SCL=1;                //读入一位数据(1)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //将数据1放到总线上
  SCL=1;                //读入一位数据(1)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
```

```
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上,此位特殊,为0代表写,为1代表读
  SCL=1;                //读入一位数据(0,写数据)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //放开数据线(SDA),以使得eeprom能够应答
  SCL=1;                //发送第九个脉冲,让eeprom应答,应答信号从数据线(SDA)输出
  SCL=0;                //应答结束,如果需要判断是否应答,应检查数据线(SDA)电平,本程
序不检查

  SDA=addr&0X80;        //将地址的最高位(MSB)放到总线上
  SCL=1;                //写入最高位(MSB D7)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X40;        //将地址的D6位放到总线上
  SCL=1;                //写入数据位(D6)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X20;        //将地址的D5位放到总线上
  SCL=1;                //写入数据位(D5)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X10;        //将地址的D4位放到总线上
  SCL=1;                //写入数据位(D4)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X08;        //将地址的D3位放到总线上
  SCL=1;                //写入数据位(D3)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X04;        //将地址的D2位放到总线上
  SCL=1;                //写入数据位(D2)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X02;        //将地址的D1位放到总线上
  SCL=1;                //写入数据位(D1)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=addr&0X01;        //将地址的(LSB D0)位放到总线上
  SCL=1;                //写入数据位(D0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //放开数据线(SDA),以使得eeprom能够应答
  SCL=1;                //发送第九个脉冲,让eeprom应答,应答信号从数据线(SDA)输出
  SCL=0;                //应答结束,如果需要判断是否应答,应检查数据线(SDA)电平,本程
序不检查

  SCL=1;                //时钟线(SCL)拉高
  SDA=1;                //数据线(SDA)拉高
  SDA=0;                //在时钟线(SCL)为高电平时,数据线(SDA)发生下跳变,总线启动

  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //AT24C02eeprom的器件地址(A0-A2接地),1010000x.
  SCL=1;                //读入一位数据(1)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态
```

```
  SDA=1;                //将数据1放到总线上
  SCL=1;                //读入一位数据(1)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=0;                //将数据0放到总线上
  SCL=1;                //读入一位数据(0)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //将数据1放到总线上,此位特殊,为0代表写,为1代表读
  SCL=1;                //读入一位数据(1,读数据)
  SCL=0;                //只有时钟线(SCL)为低电平时才可以改变数据线(SDA)的状态

  SDA=1;                //放开数据线(SDA),以使得eeprom能够应答
  SCL=1;                //发送第九个脉冲,让eeprom应答,应答信号从数据线(SDA)输出
  SCL=0;                //应答结束,如果需要判断是否应答,应检查数据线(SDA)电平,本程
序不检查

  temp=temp|SDA;        //应答结束后,第1位数据(MSB D7)已经送出,直接读取

  SCL=1;                //释放时钟线(SCL),以准备产生下降沿
  SCL=0;                //拉低时钟线(SCL),以产生下降沿,使eeprom送出第2位数据(D6)
  temp=temp<<1;         //临时变量移位,以准备拼合新接收到的位
  temp=temp|SDA;        //拼合新的数据位

  SCL=1;                //释放时钟线(SCL),以准备产生下降沿
  SCL=0;                //拉低时钟线(SCL),以产生下降沿,使eeprom送出第3位数据(D5)
  temp=temp<<1;         //临时变量移位,以准备拼合新接收到的位
  temp=temp|SDA;        //拼合新的数据位

  SCL=1;                //释放时钟线(SCL),以准备产生下降沿
  SCL=0;                //拉低时钟线(SCL),以产生下降沿,使eeprom送出第4位数据(D4)
  temp=temp<<1;         //临时变量移位,以准备拼合新接收到的位
  temp=temp|SDA;        //拼合新的数据位

  SCL=1;                //释放时钟线(SCL),以准备产生下降沿
  SCL=0;                //拉低时钟线(SCL),以产生下降沿,使eeprom送出第5位数据(D3)
  temp=temp<<1;         //临时变量移位,以准备拼合新接收到的位
  temp=temp|SDA;        //拼合新的数据位

  SCL=1;                //释放时钟线(SCL),以准备产生下降沿
  SCL=0;                //拉低时钟线(SCL),以产生下降沿,使eeprom送出第6位数据(D2)
  temp=temp<<1;         //临时变量移位,以准备拼合新接收到的位
  temp=temp|SDA;        //拼合新的数据位

  SCL=1;                //释放时钟线(SCL),以准备产生下降沿
  SCL=0;                //拉低时钟线(SCL),以产生下降沿,使eeprom送出第7位数据(D1)
  temp=temp<<1;         //临时变量移位,以准备拼合新接收到的位
  temp=temp|SDA;        //拼合新的数据位

  SCL=1;                //释放时钟线(SCL),以准备产生下降沿
  SCL=0;                //拉低时钟线(SCL),以产生下降沿,使eeprom送出第8位数据(D0)
  temp=temp<<1;         //临时变量移位,以准备拼合新接收到的位
```

```
  temp=temp|SDA;        //拼合新的数据位

  SDA=1;                //放开数据线(SDA),以使得eeprom能够应答,此次eeprom将不应答
.
  SCL=1;                //发送第九个脉冲,让eeprom应答,应答信号从数据线(SDA)输出
  SCL=0;                //应答结束,如果需要判断是否应答,应检查数据线(SDA)电平,本程
序不检查

  SDA=0;                //拉低数据线(SDA)
  SCL=1;                //拉高时钟线(SCL)
  SDA=1;                //在时钟线(SCL)为高电平时,数据线(SDA)发生上跳变,停止总线

  return(temp);         //将读到的数据返回给调用程序
  }                     //读字节程序结束

void init(void)         //初始化程序,打开定时器0,用于扫描4位共阳LED
  {                     //初始化程序开始
  EA=1;                 //打开总中断
  ET0=1;                //打开定时器0(T0)中断
  TMOD=0X01;            //设置定时器0工作模式为方式1
  TR0=1;                //启动定时器0
  }                     //初始化程序结束

void LED_SCAN(void)     //LED扫描程序,用于扫描4位共阳LED,状态机机制
  {
  static unsigned char NUM;
                        //静态局部变量,记录状态机
  switch(NUM)           //切换状态机
    {
    case 0:             //0分支
    P2=P2|0XF0;         //关闭所有数码管
    P0=LED_SEG[D1];     //通过查表,将D1内容换成段码放置到P0口
    P2=P2&0X7F;         //打开对应的PNP三极管点亮LED
    NUM=1;              //状态转移
    break;              //分支结束
    case 1:             //1分支
    P2=P2|0XF0;         //关闭所有数码管
    P0=LED_SEG[D2];     //通过查表,将D2内容换成段码放置到P0口
    P2=P2&0XBF;         //打开对应的PNP三极管点亮LED
    NUM=2;              //状态转移
    break;              //分支结束
    case 2:             //2分支
    P2=P2|0XF0;         //关闭所有数码管
    P0=LED_SEG[D3];     //通过查表,将D3内容换成段码放置到P0口
    P2=P2&0XDF;         //打开对应的PNP三极管点亮LED
    NUM=3;              //状态转移
    break;              //分支结束
    case 3:             //3分支
    P2=P2|0XF0;         //关闭所有数码管
    P0=LED_SEG[D4];     //通过查表,将D4内容换成段码放置到P0口
    P2=P2&0XEF;         //打开对应的PNP三极管点亮LED
    NUM=0;              //状态转移
    break;              //分支结束
    default:            //默认、异常分支
    NUM=0;              //恢复状态机到正常值
    break;              //分支结束
    }                   //switch-case结束
  }                     //扫描程序结束

void timer0(void) interrupt 1 using 2
                        //定时器0中断程序
  {
  TH0=(65536-6000)/256; //每6ms扫描一次数码管,高八位
  TL0=(65536-6000)%256; //每6ms扫描一次数码管,低八位
  D1=ADDR>>4;           //获得高四位数据
```

```
  D2=ADDR&0X0F;         //获得低四位数据,左边两位数码管显示地址
  D3=INDEX>>4;          //获得高四位数据
  D4=INDEX&0X0F;        //获得低四位数据,右边两位数码管显示数据
  LED_SCAN();           //数据准备完毕,开始扫描
  }                     //定时器0中断程序结束

void main(void)         //主程序
  {                     //主程序开始
  init();               //初始化
  while(1)              //主循环
    {                   //主循环开始
    if(!KEY1){while(!KEY1);ADDR++;}
                        //如果按钮1被按下,则地址+1
    if(!KEY2){while(!KEY2);ADDR--;}
                        //如果按钮2被按下,则地址-1
    if(!KEY3){while(!KEY3);INDEX++;}
                        //如果按钮3被按下,则数据+1
    if(!KEY4){while(!KEY4);INDEX--;}
                        //如果按钮4被按下,则数据-1
    if(!KEY5){while(!KEY5);INDEX=eeprom_read_byte(ADDR);}
                        //如果按钮5被按下,则在当前地址读出数据
    if(!KEY6){while(!KEY6);eeprom_write_byte(ADDR,INDEX);}
                        //如果按钮6被按下,则在当前地址写入数据
    if(!KEY7)
                        //如果按钮7被按下,则进行256字节读写测试
      {
      unsigned char loop,readback;
                        //循环计数变量,回读临时变量
      srand(TL0);       //置随机数种子,不同的种子,将返回不同的随机数序列
      BEEP=1;           //关闭报警蜂鸣器
      do                //使用do-while循环
      {                 //循环开始
      ADDR=loop;        //地址跟随循环计数值变化
      INDEX=rand();     //取随机数
      eeprom_write_byte(ADDR,INDEX);
                        //在当前地址写入取到的随机数
      readback=eeprom_read_byte(ADDR);
                        //立即将刚才写入的数据读出来
      if(INDEX!=readback){BEEP=0;while(KEY7);BEEP=1;}
                        //判断写入的数据和读出的数据是否一致,如不一致则报警
                        //报警后,需要再次按动按钮7来消除报警
      }while(--loop);   //do-while循环结构
      }                 //按钮7处理程序结束
    }                   //主循环结束
  }                     //主程序结束
```