#include <STC89C5xRC.H>

#include <intrins.h>

#include "LCD12864\_No\_font\_library.h"

#include "zi\_ku.h"

#include "24C02.h"

uchar system\_time=0;

bit s0=0;

uchar beep\_time=0; //蜂鸣器端口定义

sbit beep=P0^5; //蜂鸣器端口定义

uchar key\_back=16; //按键返回值变量

uchar ms=0; //按键返回值变量

bit key\_flag=0;

uchar Box\_number=0;

uchar xdata password[6] ={0,0,0,0,0,0}; //存储密码的数组

uchar xdata password\_flsh[6]={0,0,0,0,0,0}; //存储密码的数组

uchar password\_count=0; //存储密码的数组计数变量

uchar xdata Password\_flsh2[6]; //这个存储的是，要修的密码密码的时候，再次输入的密码

bit Modify=0; //这个标志位是修改密码用的

uchar Box\_state[8];

uchar Surplus=0;

uchar admin[6];

uchar state=0;

uchar state\_clear=0;

bit clear\_flag=1;

sbit led0=P3^0;

sbit led1=P3^1;

sbit led2=P3^2;

sbit led3=P3^3;

sbit led4=P3^4;

sbit led5=P3^5;

sbit led6=P3^6;

sbit led7=P3^7;

uchar led0\_time=0;

uchar led1\_time=0;

uchar led2\_time=0;

uchar led3\_time=0;

uchar led4\_time=0;

uchar led5\_time=0;

uchar led6\_time=0;

uchar led7\_time=0;

void Uart1Data(uchar dat) //串口数据发送

{

SBUF=dat;

while(!TI);

TI=0;

}

void delay(unsigned int T)

{

while(T--);

}

void memory\_Pass(uchar dat1,dat2,dat3,dat4,dat5,dat6,Num) //保存密码

{

unsigned char tab[6];

tab[0]=dat1;

tab[1]=dat2;

tab[2]=dat3;

tab[3]=dat4;

tab[4]=dat5;

tab[5]=dat6;

Write\_dat\_24C02(tab,Num\*6,6);

}

void read\_Pass(Num) //读取密码

{

Read\_dat\_24C02(password\_flsh,Num\*6,6);

}

void memory\_Box() //保存箱子状态

{

Write\_dat\_24C02(Box\_state,100,8);

}

void read\_memory\_Box() //读箱子状态

{

Read\_dat\_24C02(Box\_state,100,8);

}

void memory\_admin() //保存箱子状态

{

Write\_dat\_24C02(admin,120,6);

}

void read\_memory\_admin() //读箱子状态

{

Read\_dat\_24C02(admin,120,6);

}

void Chinese\_characters\_8X16(unsigned char X\_dat,unsigned char Y\_dat,unsigned char Box\_dat[])

{

unsigned char i;

unsigned char dis\_Box[16];

for(i=0;i<16;i++)

{

dis\_Box[i]=Box\_dat[i];

}

LCD\_display\_byte(X\_dat ,Y\_dat,dis\_Box[0]);

LCD\_display\_byte(X\_dat+1,Y\_dat,dis\_Box[1]);

LCD\_display\_byte(X\_dat+2,Y\_dat,dis\_Box[2]);

LCD\_display\_byte(X\_dat+3,Y\_dat,dis\_Box[3]);

LCD\_display\_byte(X\_dat+4,Y\_dat,dis\_Box[4]);

LCD\_display\_byte(X\_dat+5,Y\_dat,dis\_Box[5]);

LCD\_display\_byte(X\_dat+6,Y\_dat,dis\_Box[6]);

LCD\_display\_byte(X\_dat+7,Y\_dat,dis\_Box[7]);

LCD\_display\_byte(X\_dat ,Y\_dat+1,dis\_Box[8]);

LCD\_display\_byte(X\_dat+1,Y\_dat+1,dis\_Box[9]);

LCD\_display\_byte(X\_dat+2,Y\_dat+1,dis\_Box[10]);

LCD\_display\_byte(X\_dat+3,Y\_dat+1,dis\_Box[11]);

LCD\_display\_byte(X\_dat+4,Y\_dat+1,dis\_Box[12]);

LCD\_display\_byte(X\_dat+5,Y\_dat+1,dis\_Box[13]);

LCD\_display\_byte(X\_dat+6,Y\_dat+1,dis\_Box[14]);

LCD\_display\_byte(X\_dat+7,Y\_dat+1,dis\_Box[15]);

}

void Chinese\_characters\_16X16(unsigned char characters\_X,unsigned char characters\_Y,unsigned char mu[])

{

unsigned char x=0,y=0;

unsigned char characters\_flsh[32];

for(x=0;x<32;x++)

{

characters\_flsh[x]= mu[x];

}

LCD\_display\_byte(characters\_X,characters\_Y,mu[0]);

LCD\_display\_byte(characters\_X+1,characters\_Y,mu[1]);

LCD\_display\_byte(characters\_X+2,characters\_Y,mu[2]);

LCD\_display\_byte(characters\_X+3,characters\_Y,mu[3]);

LCD\_display\_byte(characters\_X+4,characters\_Y,mu[4]);

LCD\_display\_byte(characters\_X+5,characters\_Y,mu[5]);

LCD\_display\_byte(characters\_X+6,characters\_Y,mu[6]);

LCD\_display\_byte(characters\_X+7,characters\_Y,mu[7]);

LCD\_display\_byte(characters\_X+8,characters\_Y,mu[8]);

LCD\_display\_byte(characters\_X+9,characters\_Y,mu[9]);

LCD\_display\_byte(characters\_X+10,characters\_Y,mu[10]);

LCD\_display\_byte(characters\_X+11,characters\_Y,mu[11]);

LCD\_display\_byte(characters\_X+12,characters\_Y,mu[12]);

LCD\_display\_byte(characters\_X+13,characters\_Y,mu[13]);

LCD\_display\_byte(characters\_X+14,characters\_Y,mu[14]);

LCD\_display\_byte(characters\_X+15,characters\_Y,mu[15]);

LCD\_display\_byte(characters\_X,characters\_Y+1,mu[16]);

LCD\_display\_byte(characters\_X+1,characters\_Y+1,mu[17]);

LCD\_display\_byte(characters\_X+2,characters\_Y+1,mu[18]);

LCD\_display\_byte(characters\_X+3,characters\_Y+1,mu[19]);

LCD\_display\_byte(characters\_X+4,characters\_Y+1,mu[20]);

LCD\_display\_byte(characters\_X+5,characters\_Y+1,mu[21]);

LCD\_display\_byte(characters\_X+6,characters\_Y+1,mu[22]);

LCD\_display\_byte(characters\_X+7,characters\_Y+1,mu[23]);

LCD\_display\_byte(characters\_X+8,characters\_Y+1,mu[24]);

LCD\_display\_byte(characters\_X+9,characters\_Y+1,mu[25]);

LCD\_display\_byte(characters\_X+10,characters\_Y+1,mu[26]);

LCD\_display\_byte(characters\_X+11,characters\_Y+1,mu[27]);

LCD\_display\_byte(characters\_X+12,characters\_Y+1,mu[28]);

LCD\_display\_byte(characters\_X+13,characters\_Y+1,mu[29]);

LCD\_display\_byte(characters\_X+14,characters\_Y+1,mu[30]);

LCD\_display\_byte(characters\_X+15,characters\_Y+1,mu[31]);

}

void dispaly()

{

if(state\_clear!=state)

{

state\_clear=state;

LCD\_clear(); password\_count=0;

}

if(state==0) //待机界面

{

password\_count=0;

//显示箱子状态

if(Box\_state[0]==0) Chinese\_characters\_16X16(0,0,num1);

else Chinese\_characters\_16X16(0,0,numEnd);

if(Box\_state[1]==0) Chinese\_characters\_16X16(0,2,num2);

else Chinese\_characters\_16X16(0,2,numEnd);

if(Box\_state[2]==0) Chinese\_characters\_16X16(0,4,num3);

else Chinese\_characters\_16X16(0,4,numEnd);

if(Box\_state[3]==0) Chinese\_characters\_16X16(0,6,num4);

else Chinese\_characters\_16X16(0,6,numEnd);

if(Box\_state[4]==0) Chinese\_characters\_16X16(112,0,num5);

else Chinese\_characters\_16X16(112,0,numEnd);

if(Box\_state[5]==0) Chinese\_characters\_16X16(112,2,num6);

else Chinese\_characters\_16X16(112,2,numEnd);

if(Box\_state[6]==0) Chinese\_characters\_16X16(112,4,num7);

else Chinese\_characters\_16X16(112,4,numEnd);

if(Box\_state[7]==0) Chinese\_characters\_16X16(112,6,num8);

else Chinese\_characters\_16X16(112,6,numEnd);

//显示欢迎界面

Chinese\_characters\_16X16(32,0,huan);

Chinese\_characters\_16X16(48,0,ying);

Chinese\_characters\_16X16(64,0,guang);

Chinese\_characters\_16X16(80,0,lin);

//显示剩余

Surplus=8-(Box\_state[0]+Box\_state[1]+Box\_state[2]+Box\_state[3]+Box\_state[4]+Box\_state[5]+Box\_state[6]+Box\_state[7]);

if(Surplus!=0)

{

Chinese\_characters\_16X16(32,2,sheng);

Chinese\_characters\_16X16(48,2,yu);

Chinese\_characters\_16X16(64,2,xiang);

Chinese\_characters\_8X16(80,2,Ascii\_mao\_hao);

switch(Surplus)

{

case 1 : Chinese\_characters\_8X16(88,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(88,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(88,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(88,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(88,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(88,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(88,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(88,2,Ascii\_8); break;

}

//显示提示语

Chinese\_characters\_16X16(24,4,cun);

Chinese\_characters\_16X16(40,4,wu);

Chinese\_characters\_16X16(56,4,qing3);

Chinese\_characters\_16X16(72,4,an);

Chinese\_characters\_16X16(88,4,cun);

Chinese\_characters\_16X16(24,6,qu);

Chinese\_characters\_16X16(40,6,wu);

Chinese\_characters\_16X16(56,6,qing3);

Chinese\_characters\_16X16(72,6,an);

Chinese\_characters\_16X16(88,6,qu);

}

else

{

Chinese\_characters\_16X16(32,2,ci);

Chinese\_characters\_16X16(48,2,gui);

Chinese\_characters\_16X16(64,2,yi);

Chinese\_characters\_16X16(80,2,man);

}

}

else if(state==1) //存物界面

{

Chinese\_characters\_16X16(8,0,qing3);

Chinese\_characters\_16X16(24,0,she);

Chinese\_characters\_16X16(40,0,zhi);

Chinese\_characters\_16X16(56,0,shu\_zi\_liu);

Chinese\_characters\_16X16(72,0,wei);

Chinese\_characters\_16X16(88,0,mi);

Chinese\_characters\_16X16(104,0,ma); ///提示语，请输入六位密码

Chinese\_characters\_16X16(0,2,bing);

Chinese\_characters\_16X16(16,2,an);

Chinese\_characters\_16X16(32,2,xia);

Chinese\_characters\_16X16(48,2,que);

Chinese\_characters\_16X16(64,2,ren);

Chinese\_characters\_16X16(80,2,jian);

Chinese\_characters\_16X16(96,2,que);

Chinese\_characters\_16X16(112,2,ding);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,4,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,4,Ascii\_9); break;

}

Chinese\_characters\_16X16(16,6,mi);

Chinese\_characters\_16X16(32,6,ma);

Chinese\_characters\_8X16(48,6,Ascii\_mao\_hao);

if(password\_count>=1) Chinese\_characters\_8X16(56,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(56,6,Ascii\_kong);

if(password\_count>=2) Chinese\_characters\_8X16(64,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(64,6,Ascii\_kong);

if(password\_count>=3) Chinese\_characters\_8X16(72,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(72,6,Ascii\_kong);

if(password\_count>=4) Chinese\_characters\_8X16(80,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(80,6,Ascii\_kong);

if(password\_count>=5) Chinese\_characters\_8X16(88,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(88,6,Ascii\_kong);

if(password\_count>=6) Chinese\_characters\_8X16(96,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(96,6,Ascii\_kong);

}

else if(state==2) //取物界面-输入箱号

{

Chinese\_characters\_16X16(8,0,qing3);

Chinese\_characters\_16X16(24,0,shu);

Chinese\_characters\_16X16(40,0,ru);

Chinese\_characters\_16X16(56,0,qu);

Chinese\_characters\_16X16(72,0,wu);

Chinese\_characters\_16X16(88,0,xiang);

Chinese\_characters\_16X16(104,0,hao);

Chinese\_characters\_16X16(0,2,bing);

Chinese\_characters\_16X16(16,2,an);

Chinese\_characters\_16X16(32,2,xia);

Chinese\_characters\_16X16(48,2,que);

Chinese\_characters\_16X16(64,2,ren);

Chinese\_characters\_16X16(80,2,jian);

Chinese\_characters\_16X16(96,2,jixu\_ji);

Chinese\_characters\_16X16(112,2,xu);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,4,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,4,Ascii\_9); break;

}

Chinese\_characters\_16X16(40,6,xiang);

Chinese\_characters\_16X16(56,6,hao);

Chinese\_characters\_8X16(72,6,Ascii\_mao\_hao);

if(password\_count==0)

{

if(s0)

{

Chinese\_characters\_8X16(80,6,Ascii\_xia\_hua\_xian);

}

else

{

Chinese\_characters\_8X16(80,6,Ascii\_kong);

}

}

else

{

switch(Box\_number%10)

{

case 0 : Chinese\_characters\_8X16(80,6,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(80,6,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(80,6,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(80,6,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(80,6,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(80,6,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(80,6,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(80,6,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(80,6,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(80,6,Ascii\_9); break;

}

}

}

else if(state==3) //取物界面-输入密码

{

Chinese\_characters\_16X16(8,0,qing3);

Chinese\_characters\_16X16(24,0,shu);

Chinese\_characters\_16X16(40,0,ru);

Chinese\_characters\_16X16(56,0,qu);

Chinese\_characters\_16X16(72,0,wu);

Chinese\_characters\_16X16(88,0,mi);

Chinese\_characters\_16X16(104,0,ma);

Chinese\_characters\_16X16(0,2,bing);

Chinese\_characters\_16X16(16,2,an);

Chinese\_characters\_16X16(32,2,xia);

Chinese\_characters\_16X16(48,2,que);

Chinese\_characters\_16X16(64,2,ren);

Chinese\_characters\_16X16(80,2,jian);

Chinese\_characters\_16X16(96,2,que);

Chinese\_characters\_16X16(112,2,ding);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,4,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,4,Ascii\_9); break;

}

Chinese\_characters\_16X16(16,6,mi);

Chinese\_characters\_16X16(32,6,ma);

Chinese\_characters\_8X16(48,6,Ascii\_mao\_hao);

if(password\_count>=1) Chinese\_characters\_8X16(56,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(56,6,Ascii\_kong);

if(password\_count>=2) Chinese\_characters\_8X16(64,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(64,6,Ascii\_kong);

if(password\_count>=3) Chinese\_characters\_8X16(72,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(72,6,Ascii\_kong);

if(password\_count>=4) Chinese\_characters\_8X16(80,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(80,6,Ascii\_kong);

if(password\_count>=5) Chinese\_characters\_8X16(88,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(88,6,Ascii\_kong);

if(password\_count>=6) Chinese\_characters\_8X16(96,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(96,6,Ascii\_kong);

}

else if(state==4) //存物成功界面，存物成功，箱号XX 请您牢记密码

{

Chinese\_characters\_16X16(32,0,cun);

Chinese\_characters\_16X16(48,0,wu);

Chinese\_characters\_16X16(64,0,cheng);

Chinese\_characters\_16X16(80,0,gong);

Chinese\_characters\_16X16(16,2,qing3);

Chinese\_characters\_16X16(32,2,nin);

Chinese\_characters\_16X16(48,2,lao);

Chinese\_characters\_16X16(64,2,ji);

Chinese\_characters\_16X16(80,2,mi);

Chinese\_characters\_16X16(96,2,ma);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,4,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,4,Ascii\_9); break;

}

Chinese\_characters\_16X16(40,6,xiang);

Chinese\_characters\_16X16(56,6,hao);

Chinese\_characters\_8X16(72,6,Ascii\_mao\_hao);

switch(Box\_number%10)

{

case 0 : Chinese\_characters\_8X16(80,6,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(80,6,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(80,6,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(80,6,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(80,6,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(80,6,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(80,6,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(80,6,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(80,6,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(80,6,Ascii\_9); break;

}

}

else if(state==5) //取物界面-取物成功界面 取物成功， 欢迎再次光临

{

Chinese\_characters\_16X16(32,0,qu);

Chinese\_characters\_16X16(48,0,wu);

Chinese\_characters\_16X16(64,0,cheng);

Chinese\_characters\_16X16(80,0,gong);

switch(Box\_number%10)

{

case 0 : Chinese\_characters\_8X16(24,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(24,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(24,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(24,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(24,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(24,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(24,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(24,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(24,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(24,2,Ascii\_9); break;

}

Chinese\_characters\_16X16(32,2,xiang);

Chinese\_characters\_16X16(48,2,men);

Chinese\_characters\_16X16(64,2,yi);

Chinese\_characters\_16X16(80,2,kai);

Chinese\_characters\_16X16(16,6,huan);

Chinese\_characters\_16X16(32,6,ying);

Chinese\_characters\_16X16(48,6,zai);

Chinese\_characters\_16X16(64,6,xia\_ci\_ci);

Chinese\_characters\_16X16(80,6,guang);

Chinese\_characters\_16X16(96,6,lin);

}

else if(state==6) //存物 密码

{

Chinese\_characters\_16X16(32,0,cun);

Chinese\_characters\_16X16(48,0,wu);

Chinese\_characters\_16X16(64,0,shi);

Chinese\_characters\_16X16(80,0,bai);

Chinese\_characters\_8X16(24,2,Ascii\_1);

Chinese\_characters\_16X16(32,2,xiang);

Chinese\_characters\_16X16(48,2,men);

Chinese\_characters\_16X16(64,2,yi);

Chinese\_characters\_16X16(80,2,kai);

Chinese\_characters\_16X16(16,6,huan);

Chinese\_characters\_16X16(32,6,ying);

Chinese\_characters\_16X16(48,6,zai);

Chinese\_characters\_16X16(64,6,xia\_ci\_ci);

Chinese\_characters\_16X16(80,6,guang);

Chinese\_characters\_16X16(96,6,lin);

}

else if(state==7) //管理员登录界面

{

Chinese\_characters\_16X16(24,0,guan);

Chinese\_characters\_16X16(40,0,li);

Chinese\_characters\_16X16(56,0,yuan);

Chinese\_characters\_16X16(72,0,deng);

Chinese\_characters\_16X16(88,0,lu);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,4,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,4,Ascii\_9); break;

}

Chinese\_characters\_16X16(16,6,mi);

Chinese\_characters\_16X16(32,6,ma);

Chinese\_characters\_8X16(48,6,Ascii\_mao\_hao);

if(password\_count>=1) Chinese\_characters\_8X16(56,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(56,6,Ascii\_kong);

if(password\_count>=2) Chinese\_characters\_8X16(64,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(64,6,Ascii\_kong);

if(password\_count>=3) Chinese\_characters\_8X16(72,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(72,6,Ascii\_kong);

if(password\_count>=4) Chinese\_characters\_8X16(80,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(80,6,Ascii\_kong);

if(password\_count>=5) Chinese\_characters\_8X16(88,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(88,6,Ascii\_kong);

if(password\_count>=6) Chinese\_characters\_8X16(96,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(96,6,Ascii\_kong);

}

else if(state==8) //管理员登录界面,箭头指向第一个选项“管理员取物”

{

Chinese\_characters\_16X16(24,0,guan);

Chinese\_characters\_16X16(40,0,li);

Chinese\_characters\_16X16(56,0,yuan);

Chinese\_characters\_16X16(72,0,jie);

Chinese\_characters\_16X16(88,0,mian);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,2,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,2,Ascii\_9); break;

}

Chinese\_characters\_8X16(8,4,Ascii\_1);

Chinese\_characters\_8X16(16,4,Ascii\_mao\_hao);

Chinese\_characters\_16X16(32,4,guan);

Chinese\_characters\_16X16(48,4,li);

Chinese\_characters\_16X16(64,4,yuan);

Chinese\_characters\_16X16(80,4,qu);

Chinese\_characters\_16X16(96,4,wu);

Chinese\_characters\_8X16(112,4,Ascii\_zuo\_jiantou);

Chinese\_characters\_8X16(8,6,Ascii\_2);

Chinese\_characters\_8X16(16,6,Ascii\_mao\_hao);

Chinese\_characters\_16X16(32,6,geng);

Chinese\_characters\_16X16(48,6,gai);

Chinese\_characters\_16X16(64,6,mi);

Chinese\_characters\_16X16(80,6,ma);

}

else if(state==9) //管理员界面,箭头指向第2个选项“更改密码”

{

Chinese\_characters\_16X16(24,0,guan);

Chinese\_characters\_16X16(40,0,li);

Chinese\_characters\_16X16(56,0,yuan);

Chinese\_characters\_16X16(72,0,jie);

Chinese\_characters\_16X16(88,0,mian);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,2,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,2,Ascii\_9); break;

}

Chinese\_characters\_8X16(8,4,Ascii\_1);

Chinese\_characters\_8X16(16,4,Ascii\_mao\_hao);

Chinese\_characters\_16X16(32,4,guan);

Chinese\_characters\_16X16(48,4,li);

Chinese\_characters\_16X16(64,4,yuan);

Chinese\_characters\_16X16(80,4,qu);

Chinese\_characters\_16X16(96,4,wu);

Chinese\_characters\_8X16(8,6,Ascii\_2);

Chinese\_characters\_8X16(16,6,Ascii\_mao\_hao);

Chinese\_characters\_16X16(32,6,geng);

Chinese\_characters\_16X16(48,6,gai);

Chinese\_characters\_16X16(64,6,mi);

Chinese\_characters\_16X16(80,6,ma);

Chinese\_characters\_8X16(112,6,Ascii\_zuo\_jiantou);

}

else if(state==10) //管理员取物界面

{

Chinese\_characters\_16X16(24,0,guan);

Chinese\_characters\_16X16(40,0,li);

Chinese\_characters\_16X16(56,0,yuan);

Chinese\_characters\_16X16(72,0,qu);

Chinese\_characters\_16X16(88,0,wu);

Chinese\_characters\_16X16(0,2,an);

Chinese\_characters\_16X16(16,2,xia);

Chinese\_characters\_16X16(32,2,dui);

Chinese\_characters\_16X16(48,2,ying4);

Chinese\_characters\_16X16(64,2,xiang);

Chinese\_characters\_16X16(80,2,hao);

Chinese\_characters\_16X16(96,2,kai);

Chinese\_characters\_16X16(112,2,xiang);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,4,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,4,Ascii\_9); break;

}

if(Box\_state[0]==0) Chinese\_characters\_16X16(0,6,num1);

else Chinese\_characters\_16X16(0,6,numEnd);

if(Box\_state[1]==0) Chinese\_characters\_16X16(16,6,num2);

else Chinese\_characters\_16X16(16,6,numEnd);

if(Box\_state[2]==0) Chinese\_characters\_16X16(32,6,num3);

else Chinese\_characters\_16X16(32,6,numEnd);

if(Box\_state[3]==0) Chinese\_characters\_16X16(48,6,num4);

else Chinese\_characters\_16X16(48,6,numEnd);

if(Box\_state[4]==0) Chinese\_characters\_16X16(64,6,num5);

else Chinese\_characters\_16X16(64,6,numEnd);

if(Box\_state[5]==0) Chinese\_characters\_16X16(80,6,num6);

else Chinese\_characters\_16X16(80,6,numEnd);

if(Box\_state[6]==0) Chinese\_characters\_16X16(96,6,num7);

else Chinese\_characters\_16X16(96,6,numEnd);

if(Box\_state[7]==0) Chinese\_characters\_16X16(112,6,num8);

else Chinese\_characters\_16X16(112,6,numEnd);

}

else if(state==11) //管理员取物界面

{

Chinese\_characters\_16X16(16,0,geng);

Chinese\_characters\_16X16(32,0,gai);

Chinese\_characters\_16X16(48,0,guan);

Chinese\_characters\_16X16(64,0,li);

Chinese\_characters\_16X16(80,0,mi);

Chinese\_characters\_16X16(96,0,ma);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,2,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,2,Ascii\_9); break;

}

Chinese\_characters\_16X16(0,4,shu);

Chinese\_characters\_16X16(16,4,ru);

Chinese\_characters\_16X16(32,4,mi);

Chinese\_characters\_16X16(48,4,ma);

Chinese\_characters\_8X16(64,4,Ascii\_mao\_hao);

if(password\_count>=1) Chinese\_characters\_8X16(72,4,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(72,4,Ascii\_kong);

if(password\_count>=2) Chinese\_characters\_8X16(80,4,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(80,4,Ascii\_kong);

if(password\_count>=3) Chinese\_characters\_8X16(88,4,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(88,4,Ascii\_kong);

if(password\_count>=4) Chinese\_characters\_8X16(96,4,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(96,4,Ascii\_kong);

if(password\_count>=5) Chinese\_characters\_8X16(104,4,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(104,4,Ascii\_kong);

if(password\_count>=6) Chinese\_characters\_8X16(112,4,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(112,4,Ascii\_kong);

Chinese\_characters\_16X16(0,6,zai);

Chinese\_characters\_16X16(16,6,ci);

Chinese\_characters\_16X16(32,6,shu);

Chinese\_characters\_16X16(48,6,ru);

Chinese\_characters\_8X16(64,6,Ascii\_mao\_hao);

if(password\_count>=7) Chinese\_characters\_8X16(72,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(72,6,Ascii\_kong);

if(password\_count>=8) Chinese\_characters\_8X16(80,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(80,6,Ascii\_kong);

if(password\_count>=9) Chinese\_characters\_8X16(88,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(88,6,Ascii\_kong);

if(password\_count>=10) Chinese\_characters\_8X16(96,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(96,6,Ascii\_kong);

if(password\_count>=11) Chinese\_characters\_8X16(104,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(104,6,Ascii\_kong);

if(password\_count>=12) Chinese\_characters\_8X16(112,6,Ascii\_xing\_hao);

else Chinese\_characters\_8X16(112,6,Ascii\_kong);

}

else if(state==12) //提示界面，输入的箱号，没有存物界面

{

Chinese\_characters\_16X16(32,0,ti);

Chinese\_characters\_16X16(48,0,ti\_shi\_shi);

Chinese\_characters\_16X16(64,0,jie);

Chinese\_characters\_16X16(80,0,mian);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,4,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,4,Ascii\_9); break;

}

Chinese\_characters\_16X16(16,6,ci);

Chinese\_characters\_16X16(32,6,xiang);

Chinese\_characters\_16X16(48,6,mei);

Chinese\_characters\_16X16(64,6,you);

Chinese\_characters\_16X16(80,6,cun);

Chinese\_characters\_16X16(96,6,wu);

}

else if(state==13) //提示界面，取物失败，密码错误

{

Chinese\_characters\_16X16(32,0,qu);

Chinese\_characters\_16X16(48,0,wu);

Chinese\_characters\_16X16(64,0,shi);

Chinese\_characters\_16X16(80,0,bai);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,2,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,2,Ascii\_9); break;

}

Chinese\_characters\_16X16(32,4,mi);

Chinese\_characters\_16X16(48,4,ma);

Chinese\_characters\_16X16(64,4,cuo\_wu\_cuo);

Chinese\_characters\_16X16(80,4,cuo\_wu\_wu);

Chinese\_characters\_16X16(96,4,dou\_hao);

Chinese\_characters\_16X16(112,4,yi\_wang\_yi);

Chinese\_characters\_16X16(0,6,yi\_wang\_wang);

Chinese\_characters\_16X16(16,6,mi);

Chinese\_characters\_16X16(32,6,ma);

Chinese\_characters\_16X16(48,6,lian);

Chinese\_characters\_16X16(64,6,xi);

Chinese\_characters\_16X16(80,6,guan);

Chinese\_characters\_16X16(96,6,li);

Chinese\_characters\_16X16(112,6,yuan);

}

else if(state==14) //管理员更改密码成功界面 ying

{

Chinese\_characters\_16X16(16,0,geng);

Chinese\_characters\_16X16(32,0,gai);

Chinese\_characters\_16X16(48,0,guan);

Chinese\_characters\_16X16(64,0,li);

Chinese\_characters\_16X16(80,0,mi);

Chinese\_characters\_16X16(96,0,ma);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,2,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,2,Ascii\_9); break;

}

Chinese\_characters\_16X16(32,6,geng);

Chinese\_characters\_16X16(48,6,gai);

Chinese\_characters\_16X16(64,6,cheng);

Chinese\_characters\_16X16(80,6,gong);

}

else if(state==15) //管理员更改密码失败界面

{

Chinese\_characters\_16X16(16,0,geng);

Chinese\_characters\_16X16(32,0,gai);

Chinese\_characters\_16X16(48,0,guan);

Chinese\_characters\_16X16(64,0,li);

Chinese\_characters\_16X16(80,0,mi);

Chinese\_characters\_16X16(96,0,ma);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,2,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,2,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,2,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,2,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,2,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,2,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,2,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,2,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,2,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,2,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,2,Ascii\_9); break;

}

Chinese\_characters\_16X16(32,6,geng);

Chinese\_characters\_16X16(48,6,gai);

Chinese\_characters\_16X16(64,6,shi);

Chinese\_characters\_16X16(80,6,bai);

}

else if(state==16) //管理员登录界面

{

Chinese\_characters\_16X16(24,0,guan);

Chinese\_characters\_16X16(40,0,li);

Chinese\_characters\_16X16(56,0,yuan);

Chinese\_characters\_16X16(72,0,deng);

Chinese\_characters\_16X16(88,0,lu);

switch(system\_time/10%10)

{

case 0 : Chinese\_characters\_8X16(56,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(56,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(56,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(56,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(56,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(56,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(56,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(56,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(56,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(56,4,Ascii\_9); break;

}

switch(system\_time%10)

{

case 0 : Chinese\_characters\_8X16(64,4,Ascii\_0); break;

case 1 : Chinese\_characters\_8X16(64,4,Ascii\_1); break;

case 2 : Chinese\_characters\_8X16(64,4,Ascii\_2); break;

case 3 : Chinese\_characters\_8X16(64,4,Ascii\_3); break;

case 4 : Chinese\_characters\_8X16(64,4,Ascii\_4); break;

case 5 : Chinese\_characters\_8X16(64,4,Ascii\_5); break;

case 6 : Chinese\_characters\_8X16(64,4,Ascii\_6); break;

case 7 : Chinese\_characters\_8X16(64,4,Ascii\_7); break;

case 8 : Chinese\_characters\_8X16(64,4,Ascii\_8); break;

case 9 : Chinese\_characters\_8X16(64,4,Ascii\_9); break;

}

Chinese\_characters\_16X16(24,6,mi);

Chinese\_characters\_16X16(40,6,ma);

Chinese\_characters\_16X16(56,6,cuo\_wu\_cuo);

Chinese\_characters\_16X16(72,6,cuo\_wu\_wu);

}

}

void Time0\_int()

{

TMOD = 0x01; //设置定时器模式

TL0 = 0x00; //设置定时初值

TH0 = 0x4C; //设置定时初值

TR0 = 1; //定时器0开始计时

ET0 = 1;

}

void Uart\_int()

{

PCON &= 0x7F; //波特率不倍速

SCON = 0x50; //8位数据,可变波特率

AUXR &= 0xBF; //定时器1时钟为Fosc/12,即12T

AUXR &= 0xFE; //串口1选择定时器1为波特率发生器

TMOD &= 0x0F; //清除定时器1模式位

TMOD |= 0x20; //设定定时器1为8位自动重装方式

TL1 = 0xFD; //设定定时初值

TH1 = 0xFD; //设定定时器重装值

ET1 = 0; //禁止定时器1中断

TR1 = 1; //启动定时器1

}

/\*

4\*4矩阵键盘处理函数

对于当前程序与硬件来说的键值：

1→0xE7 2→0xD7 3→0xB7 存 →0x77

4→0xEB 5→0xDB 6→0xBB 取 →0x7B

7→0xED 8→0xDD 9→0xBD 管理→0x7D

1→0xEE 1→0xDE 1→0xBE 选择→0x7E

\*/

void key\_back\_dispose()

{

unsigned char Key\_Flsh1=0,Key\_Flsh2=0;

P1=0x0F;

delay(55);

if(P1!=0x0F)

{

if(key\_flag)

{

key\_flag=0;

Key\_Flsh1=P1;

P1=0xf0;

if(P1!=0xf0) //

{

Key\_Flsh2=P1|Key\_Flsh1;

switch(Key\_Flsh2)

{

case 0xE7 : key\_back=1; break;

case 0xD7 : key\_back=2; break;

case 0xB7 : key\_back=3; break;

case 0x77 : key\_back=12; break;

case 0xEB : key\_back=4; break;

case 0xDB : key\_back=5; break;

case 0xBB : key\_back=6; break;

case 0x7B : key\_back=13; break;

case 0xED : key\_back=7; break;

case 0xDD : key\_back=8; break;

case 0xBD : key\_back=9; break;

case 0x7D : key\_back=14; break;

case 0xEE : key\_back=10; break;

case 0xDE : key\_back=0; break;

case 0xBE : key\_back=11; break;

case 0x7E : key\_back=15; break;

}

beep\_time=1;

// while(P1!=0x0F) //按键死循环释放

// {

// P1=0x0f; //先赋值，赋值之后短延时稳定一下数据，然后再判断数据

// delay(333);

// };

if(state!=0)system\_time=10;

}

}

}

else

{

key\_flag=1;

}

}

void key\_dispose()

{

if(key\_back!=16)

{

if(state==0) //如果说是待机界面 ,只有存取按键和管理界面可用

{

if(key\_back==12) //存按键按下

{

key\_back=16; //清除按键值

system\_time=10;

state=1;

password\_count=0;

}

else if(key\_back==13) //取按键按下

{

key\_back=16; //清除按键值

system\_time=10;

state=2;

}

else if(key\_back==14) //管理按键按下

{

key\_back=16; //清除按键值

system\_time=10;

state=7;

}

}

else if(state==1) //跳转到存物界面，按键部分只有0-9，删除/退出 确定 按键可用

{

if(key\_back>=0&&key\_back<=9) //按键按下0-9，说明是输入密码

{

if(password\_count<6) //判断密码输入位数够不够六位

{

password[password\_count]=key\_back;

password\_count++;

}

else

{

beep\_time=4;

}

key\_back=16; //清除按键值

}

else if(key\_back==10) //退格按键按下，只要密码不为空，就退一位

{

key\_back=16; //清除按键值

if(password\_count>0) password\_count--;

}

else if(key\_back==11) //确认按键按下

{ key\_back=16; //清除按键值

if(password\_count==6)

{

if(Box\_state[0]==0)

{ Box\_state[0]=1;

memory\_Box(); Box\_number=1;

memory\_Pass(password[0],password[1],password[2],password[3],password[4],password[5],Box\_number);

led0\_time=40; led0=0;

}

else if(Box\_state[1]==0)

{ Box\_state[1]=1;

memory\_Box(); Box\_number=2;

memory\_Pass(password[0],password[1],password[2],password[3],password[4],password[5],Box\_number);

led1\_time=40; led1=0;

}

else if(Box\_state[2]==0)

{ Box\_state[2]=1;

memory\_Box(); Box\_number=3;

memory\_Pass(password[0],password[1],password[2],password[3],password[4],password[5],Box\_number);

led2\_time=40; led2=0;

}

else if(Box\_state[3]==0)

{Box\_state[3]=1;

memory\_Box(); Box\_number=4;

memory\_Pass(password[0],password[1],password[2],password[3],password[4],password[5],Box\_number);

led3\_time=40; led3=0;

}

else if(Box\_state[4]==0)

{ Box\_state[4]=1;

memory\_Box(); Box\_number=5;

memory\_Pass(password[0],password[1],password[2],password[3],password[4],password[5],Box\_number);

led4\_time=40; led4=0;

}

else if(Box\_state[5]==0)

{ Box\_state[5]=1;

memory\_Box(); Box\_number=6;

memory\_Pass(password[0],password[1],password[2],password[3],password[4],password[5],Box\_number);

led5\_time=40; led5=0;

}

else if(Box\_state[6]==0)

{ Box\_state[6]=1;

memory\_Box(); Box\_number=7;

memory\_Pass(password[0],password[1],password[2],password[3],password[4],password[5],Box\_number);

led6\_time=40; led6=0;

}

else if(Box\_state[7]==0)

{ Box\_state[7]=1;

memory\_Box(); Box\_number=8;

memory\_Pass(password[0],password[1],password[2],password[3],password[4],password[5],Box\_number);

led7\_time=40; led7=0;

}

system\_time=3;

state=4;

}

else

{

beep\_time=4;

}

}

else if(key\_back==15) //回到主界面按键按下

{ key\_back=16; //清除按键值

password\_count=0; //计数变量也清零

state=0; //回到主界面变量

}

}

else if(state==2) //跳转到取物界面，按键部分只有0-9，删除/退出 按键可用 这里是取物箱号选定的数据处理部分

{

if(key\_back>=0&&key\_back<=9) //按键按下0-9，说明是输入密箱子号

{

if(password\_count<=1) //判断密码输入位数够不够一位

{

Box\_number=key\_back;

if(password\_count!=1) password\_count++;

}

else

{

beep\_time=4;

}

key\_back=16; //清除按键值

}

else if(key\_back==10) //退格按键按下，只要密码不为空，就退一位

{

key\_back=16; //清除按键值

if(password\_count>0) password\_count--;

}

else if(key\_back==15) //回到主界面按键按下

{ key\_back=16; //清除按键值

password\_count=0; //计数变量也清零

state=0; //回到主界面变量

}

else if(key\_back==11) //确认按键按下

{ key\_back=16; //清除按键值

if(password\_count==1&&Box\_number!=0&&Box\_number!=9) //先判断是否有正确的箱号输入

{

if(Box\_state[Box\_number-1]==1) //确定一下这个箱子里面有东西

{

state=3;

password\_count=0;

}

else //这里欠缺一个界面，显示出箱子原本是空的界面处理

{

beep\_time=4; system\_time=1; state=12;

}

}

else //这里还可以添加一个界面，显示出，填写的密码不正确的界面

{

beep\_time=10;

}

}

}

else if(state==3)

{

if(key\_back>=0&&key\_back<=9) //按键按下0-9，说明是输入密码

{

if(password\_count<6) //判断密码输入位数够不够六位

{

password[password\_count]=key\_back;

password\_count++;

}

else

{

beep\_time=4;

}

key\_back=16; //清除按键值

}

else if(key\_back==10) //退格按键按下，只要密码不为空，就退一位

{

key\_back=16; //清除按键值

if(password\_count>0) password\_count--;

}

else if(key\_back==11) //确认按键按下

{ key\_back=16; //清除按键值

if(password\_count==6)

{

read\_Pass(Box\_number);

if(password\_flsh[0]==password[0]&&password\_flsh[1]==password[1]&&password\_flsh[2]==password[2]&&password\_flsh[3]==password[3]&&password\_flsh[4]==password[4]&&password\_flsh[5]==password[5])

{

memory\_Pass(0,0,0,0,0,0,Box\_number); //密码确认正确，清空密码

Box\_state[Box\_number-1]=0;

state=5;

system\_time=1;

memory\_Box();

switch(Box\_number)

{

case 1 : led0\_time=40; led0=0; break;

case 2 : led1\_time=40; led1=0; break;

case 3 : led2\_time=40; led2=0; break;

case 4 : led3\_time=40; led3=0; break;

case 5 : led4\_time=40; led4=0; break;

case 6 : led5\_time=40; led5=0; break;

case 7 : led6\_time=40; led6=0; break;

case 8 : led7\_time=40; led7=0; break;

}

}

else

{

state=13; system\_time=2;

}

}

else

{

beep\_time=4;

}

}

else if(key\_back==15) //回到主界面按键按下

{ key\_back=16; //清除按键值

password\_count=0; //计数变量也清零

state=0; //回到主界面变量

}

}

else if(state==4||state==15||state==14||state==16)

{

if(key\_back!=16)

{

if(state==15||state==14)

{

state=8;

}

else

{

state=0;

}

key\_back=16;

}

}

else if(state==7) //跳转到管理界面, 按键部分只有0-9，删除/退出 还有选项按键可用

{

if(key\_back>=0&&key\_back<=9) //按键按下0-9，说明是输入密码

{

if(password\_count<6) //判断密码输入位数够不够六位

{

password[password\_count]=key\_back;

password\_count++;

}

else

{

beep\_time=4;

}

key\_back=16; //清除按键值

}

else if(key\_back==10) //退格按键按下，只要密码不为空，就退一位

{

key\_back=16; //清除按键值

if(password\_count>0) password\_count--;

}

else if(key\_back==11) //确认按键按下

{ key\_back=16; //清除按键值

if(password\_count==6)

{

if((password[0]==1&&password[1]==3&&password[2]==1&&password[3]==4&&password[4]==2&&password[5]==0)||(password[0]==admin[0]&&password[1]==admin[1]&&password[2]==admin[2]&&password[3]==admin[3]&&password[4]==admin[4]&&password[5]==admin[5]))

{

password\_count=0;

state=8;

}

else

{

state=16; system\_time=2;

}

}

}

else if(key\_back==15) //回到主界面按键按下

{ key\_back=16; //清除按键值

password\_count=0; //计数变量也清零

state=0; //回到主界面变量

}

}

else if(state==8||state==9)

{

if(key\_back==1) //选择按键，选择第一个子菜单

{

state=8;

key\_back=16;

}

else if(key\_back==2) //选择按键，选择第二个子菜单

{

state=9;

key\_back=16;

}

else if(key\_back==11) //确认按键按下

{ key\_back=16;

if(state==8) //如果处于第一个子菜单，就跳转到，指定开箱的界面

{

state=10;

}

else

{

state=11; Modify=0;

}

}

else if(key\_back==15) //回到主界面按键按下

{ key\_back=16; //清除按键值

password\_count=0; //计数变量也清零

state=0; //回到主界面变量

}

}

else if(state==10)

{

if(key\_back>=1&&key\_back<=8)

{

Box\_number=key\_back;

memory\_Pass(0,0,0,0,0,0,Box\_number); //密码确认正确，清空密码

Box\_state[Box\_number-1]=0;

memory\_Box();

switch(Box\_number)

{

case 1 : led0\_time=40; led0=0; break;

case 2 : led1\_time=40; led1=0; break;

case 3 : led2\_time=40; led2=0; break;

case 4 : led3\_time=40; led3=0; break;

case 5 : led4\_time=40; led4=0; break;

case 6 : led5\_time=40; led5=0; break;

case 7 : led6\_time=40; led6=0; break;

case 8 : led7\_time=40; led7=0; break;

}

key\_back=16;

}

else if(key\_back==11) //确认按键按下

{ key\_back=16;

state=8;

}

else if(key\_back==15) //回到主界面按键按下

{ key\_back=16; //清除按键值

password\_count=0; //计数变量也清零

state=0; //回到主界面变量

}

else key\_back=16;

}

else if(state==11)

{

if(key\_back>=0&&key\_back<=9) //按键按下0-9，说明是输入密箱子号

{

if(Modify==0) //修改数据标志位为0，说明正在填写要修改的密码

{

if(password\_count<6) //这里对记录第一次输入的密码记录

{

password[password\_count]=key\_back;

password\_count++;

}

}

else //这里对记录第二次输入的密码记录

{

if(password\_count>=6&&password\_count<12)

{

Password\_flsh2[password\_count-6]=key\_back;

password\_count++;

}

}

key\_back=16; //清除按键值

}

else if(key\_back==10) //退格按键按下，只要密码不为空，就退一位

{

key\_back=16; //清除按键值

if(password\_count>0) password\_count--;

if(password\_count<=5) Modify=0;

}

else if(key\_back==15) //回到主界面按键按下

{ key\_back=16; //清除按键值

password\_count=0; //计数变量也清零

state=0; //回到主界面变量

}

else if(key\_back==11) //确认按键按下

{

key\_back=16; //清除按键值

if(password\_count==6)

{

Modify=1;

}

if(password\_count==12)

{

if(password[0]==Password\_flsh2[0]&&password[1]==Password\_flsh2[1]&&password[2]==Password\_flsh2[2]&&password[3]==Password\_flsh2[3]&&password[4]==Password\_flsh2[4]&&password[5]==Password\_flsh2[5])

{

admin[0]=password[0];

admin[1]=password[1];

admin[2]=password[2];

admin[3]=password[3];

admin[4]=password[4];

admin[5]=password[5];

memory\_admin();

state=14; system\_time=2;

}

else

{

state=15; system\_time=2;

}

}

}

}

}

}

void led\_dispose()

{

if(led0\_time!=0) led0\_time--;

else led0=1;

if(led1\_time!=0) led1\_time--;

else led1=1;

if(led2\_time!=0) led2\_time--;

else led2=1;

if(led3\_time!=0) led3\_time--;

else led3=1;

if(led4\_time!=0) led4\_time--;

else led4=1;

if(led5\_time!=0) led5\_time--;

else led5=1;

if(led6\_time!=0) led6\_time--;

else led6=1;

if(led7\_time!=0) led7\_time--;

else led7=1;

}

void main()

{

Time0\_int();

// Uart\_int();

EA=1;

LCD\_initialize();

LCD\_clear();

read\_memory\_Box();

read\_memory\_admin();

while(1)

{

key\_back\_dispose();

dispaly();

key\_dispose();

}

}

void time0() interrupt 1

{

TH0=0x3C;

TL0=0xb0;

ms++;

led\_dispose();

if(beep\_time!=0)

{

if(beep\_time!=0)

{

beep\_time--;

beep=~beep;

}

}else beep=1;

if(ms%10==0) s0=~s0;

if(ms>=20)

{

ms=0;

if(system\_time!=0) system\_time--;

else

{

if(state==15||state==14)

{

system\_time=10; state=8;

}

else state=0;

}

}

}

//void uart() interrupt 4

// {

// if(RI)

// {

// RI=0;

// }

// }

#ifndef \_LCD12864\_No\_font\_library\_H\_

#define \_LCD12864\_No\_font\_library\_H\_

#define LCD\_DATA P2

#define uchar unsigned char

#define uint unsigned int

sbit LCD\_RS=P0^2;

sbit LCD\_RW=P0^1;

sbit LCD\_E=P0^0;

sbit LCD\_CS2=P0^4; //右屏选择（左右屏有时候相反）

sbit LCD\_CS1=P0^3; //左屏选择

//sbit LCD\_RST=P3^7;

void LCD\_check\_busy()

{

unsigned char temp;

LCD\_RS=0;

LCD\_RW=1;

do

{

LCD\_DATA=0xff;

LCD\_E=1;

temp=LCD\_DATA;

LCD\_E=0;

}while((temp&0x80)==0x80);

}

//写指令代码（cs为0选左屏，cs为1选右屏）

void LCD\_W\_code(unsigned char tpcode,bit cs)

{

LCD\_RS=0;

LCD\_RW=0;

LCD\_CS2=~cs;

LCD\_CS1=cs;

LCD\_DATA=tpcode;

LCD\_E=1;

\_nop\_();

LCD\_E=0;

}

//写显示数据（cs为0选左屏，cs为1选右屏）

void LCD\_W\_data(unsigned char tpdata,bit cs)

{

LCD\_check\_busy();

LCD\_RS=1;

LCD\_RW=0;

LCD\_CS2=~cs;

LCD\_CS1=cs;

LCD\_DATA=tpdata;

LCD\_E=1;

\_nop\_();

LCD\_E=0;

}

//LCD初始化函数

void LCD\_initialize()

{

// LCD\_RST=0;

\_nop\_();

\_nop\_();

// LCD\_RST=1;

LCD\_W\_code(0x3f,0); //开显示设置

LCD\_W\_code(0xc0,0); //设置显示起始行为第一行

LCD\_W\_code(0xb8,0); //页面地址设置

LCD\_W\_code(0x40,0); //列地址设为0

LCD\_W\_code(0x3f,1);

LCD\_W\_code(0xc0,1);

LCD\_W\_code(0xb8,1);

LCD\_W\_code(0x40,1);

}

//LCD显示字符串函数（word表示要显示的字符串,

//length表示要显示的字符串宽度,

//x表示首字符所在行数,

//y表示首字符所在列数）

void LCD\_display\_word(unsigned char word[],

unsigned int length,

unsigned char x,

unsigned char y)

{

unsigned char i;

for(i=0;i<length;i++)

{

LCD\_W\_code(0xb8+x,0);

LCD\_W\_code(0xb8+x,1);

if(y+i<64)

{

LCD\_W\_code(0x40+y+i,0);

LCD\_W\_data(word[i],0);

}

else

{

LCD\_W\_code(y+i,1);

LCD\_W\_data(word[i],1);

}

}

}

//LCD画全屏函数

void LCD\_full\_draw(unsigned char word[])

{

unsigned char i,j;

for(i=0;i<8;i++)

{

LCD\_W\_code(0xb8+i,0);

LCD\_W\_code(0x40,0);

for(j=0;j<64;j++)

{

LCD\_W\_data(word[i\*128+j],0);

}

LCD\_W\_code(0xb8+i,1);

LCD\_W\_code(0x40,1);

for(j=0;j<64;j++)

{

LCD\_W\_data(word[i\*128+64+j],1);

}

}

}

//LCD显示一个字节函数（

//x表示x坐标, X轴范围是 0 - 128

//y表示y坐标, Y轴范围是 0 - 7

//tpdata表示要显示的数据）

void LCD\_display\_byte(unsigned char x,

unsigned char y,

unsigned char tpdata)

{

if(x<64)

{

LCD\_W\_code(0xb8+y,0);

LCD\_W\_code(0x40+x,0);

LCD\_W\_data(tpdata,0);

}

else

{

LCD\_W\_code(0xb8+y,1);

LCD\_W\_code(x,1);

LCD\_W\_data(tpdata,1);

}

}

void LCD\_draw(unsigned char word[])

{

unsigned char i,j;

for(i=0;i<8;i++)

{

LCD\_W\_code(0xb8+i,1);

LCD\_W\_code(0x40+20,1);

for(j=0;j<44;j++)

{

LCD\_W\_data(word[i\*44+j],1);

}

}

}

//LCD清屏函数

void LCD\_clear()

{

unsigned char i,j;

for(i=0;i<8;i++)

{

for(j=0;j<128;j++)

{

LCD\_display\_byte(j,i,0x00);

}

}

}

#endif

#ifndef \_zi\_ku\_H\_

#define \_zi\_ku\_H\_

#define uchar unsigned char

#define uint unsigned int

/\*

这段是用到的一些字符

\*/

uchar code Ascii\_mao\_hao[]={0x00,0x00,0x00,0xC0,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x00,0x00,0x00,/\*":",10\*/};

uchar code Ascii\_0[]={0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,0x00,0x0F,0x10,0x20,0x20,0x10,0x0F,0x00,/\*"0",9\*/};

uchar code Ascii\_1[]={0x00,0x10,0x10,0xF8,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00,/\*"1",0\*/};

uchar code Ascii\_2[]={0x00,0x70,0x08,0x08,0x08,0x88,0x70,0x00,0x00,0x30,0x28,0x24,0x22,0x21,0x30,0x00,/\*"2",1\*/};

uchar code Ascii\_3[]={0x00,0x30,0x08,0x88,0x88,0x48,0x30,0x00,0x00,0x18,0x20,0x20,0x20,0x11,0x0E,0x00,/\*"3",2\*/};

uchar code Ascii\_4[]={0x00,0x00,0xC0,0x20,0x10,0xF8,0x00,0x00,0x00,0x07,0x04,0x24,0x24,0x3F,0x24,0x00,/\*"4",3\*/};

uchar code Ascii\_5[]={0x00,0xF8,0x08,0x88,0x88,0x08,0x08,0x00,0x00,0x19,0x21,0x20,0x20,0x11,0x0E,0x00,/\*"5",4\*/};

uchar code Ascii\_6[]={0x00,0xE0,0x10,0x88,0x88,0x18,0x00,0x00,0x00,0x0F,0x11,0x20,0x20,0x11,0x0E,0x00,/\*"6",5\*/};

uchar code Ascii\_7[]={0x00,0x38,0x08,0x08,0xC8,0x38,0x08,0x00,0x00,0x00,0x00,0x3F,0x00,0x00,0x00,0x00,/\*"7",6\*/};

uchar code Ascii\_8[]={0x00,0x70,0x88,0x08,0x08,0x88,0x70,0x00,0x00,0x1C,0x22,0x21,0x21,0x22,0x1C,0x00,/\*"8",7\*/};

uchar code Ascii\_9[]={0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,0x00,0x00,0x31,0x22,0x22,0x11,0x0F,0x00,/\*"9",8\*/};

uchar code Ascii\_xing\_hao[]={0x40,0x40,0x80,0xF0,0x80,0x40,0x40,0x00,0x02,0x02,0x01,0x0F,0x01,0x02,0x02,0x00,/\*"\*",0\*/};

uchar code Ascii\_kong[]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,/\*" ",0\*/};

uchar code Ascii\_xia\_hua\_xian[]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,/\*"\_",0\*/};

uchar code Ascii\_zuo\_jiantou[]={0x00,0x00,0x80,0x40,0x20,0x10,0x08,0x00,0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x00,/\*"<",0\*/};

uchar code Ascii\_dou\_hao[]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0xB0,0x70,0x00,0x00,0x00,0x00,0x00,/\*",",0\*/};

/\*

这段子模是箱子的子模

\*/

uchar code num1[]={0xFE,0x02,0x02,0x02,0x02,0x12,0xFA,0x02,0x02,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0x3F,0x20,0x20,0x20,0x28,0x28,0x2F,0x28,0x28,0x20,0x20,0x20,0x3F,0x00,0x00,0x00,/\*"未命名文件",0\*/};

uchar code num2[]={0xFE,0x02,0x02,0x32,0x0A,0x0A,0x8A,0x8A,0x72,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0x3F,0x20,0x20,0x2C,0x2A,0x29,0x28,0x28,0x2C,0x20,0x20,0x20,0x3F,0x00,0x00,0x00,/\*"未命名文件",0\*/};

uchar code num3[]={0xFE,0x02,0x02,0x1A,0x06,0x46,0x46,0xA6,0x1A,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0x3F,0x20,0x20,0x24,0x28,0x28,0x28,0x24,0x23,0x20,0x20,0x20,0x3F,0x00,0x00,0x00,/\*"未命名文件",0\*/};

uchar code num4[]={0xFE,0x02,0x02,0x82,0x42,0x22,0x12,0xFA,0x02,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0x3F,0x20,0x21,0x21,0x21,0x25,0x25,0x27,0x25,0x25,0x20,0x20,0x3F,0x00,0x00,0x00,/\*"未命名文件",0\*/};

uchar code num5[]={0x00,0x00,0x00,0xFE,0x02,0x02,0x02,0xFA,0x8A,0x4A,0x4A,0x8A,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0x3F,0x20,0x20,0x20,0x26,0x28,0x28,0x28,0x24,0x23,0x20,0x20,0x3F,/\*"未命名文件",0\*/};

uchar code num6[]={0x00,0x00,0x00,0xFE,0x02,0x02,0x02,0xFA,0x86,0x46,0x46,0x8E,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0x3F,0x20,0x20,0x20,0x27,0x28,0x28,0x28,0x24,0x23,0x20,0x20,0x3F,/\*"未命名文件",0\*/};

uchar code num7[]={0x00,0x00,0x00,0xFE,0x02,0x02,0x02,0x1A,0x0A,0x0A,0xCA,0x3A,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0x3F,0x20,0x20,0x20,0x20,0x20,0x2F,0x20,0x20,0x20,0x20,0x20,0x3F,/\*"未命名文件",0\*/};

uchar code num8[]={0x00,0x00,0x00,0xFE,0x02,0x02,0x02,0x32,0x4A,0x8A,0x8A,0x4A,0x32,0x02,0x02,0xFE,0x00,0x00,0x00,0x3F,0x20,0x20,0x20,0x26,0x29,0x28,0x28,0x29,0x26,0x20,0x20,0x3F,/\*"未命名文件",0\*/};

uchar code numEnd[]={0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0x00,0x00,0x00,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x3F,0x00,0x00,0x00,/\*"未命名文件",0\*/};

/\*

下面这段是本次设计用到的汉字

\*/

uchar code huan[] ={0x04,0x24,0x44,0x84,0x64,0x9C,0x40,0x30,0x0F,0xC8,0x08,0x08,0x28,0x18,0x00,0x00,0x10,0x08,0x06,0x01,0x82,0x4C,0x20,0x18,0x06,0x01,0x06,0x18,0x20,0x40,0x80,0x00,/\*"欢",0\*/};

uchar code ying[] ={0x40,0x40,0x42,0xCC,0x00,0x00,0xFC,0x04,0x02,0x00,0xFC,0x04,0x04,0xFC,0x00,0x00,0x00,0x40,0x20,0x1F,0x20,0x40,0x4F,0x44,0x42,0x40,0x7F,0x42,0x44,0x43,0x40,0x00,/\*"迎",1\*/};

uchar code guang[] ={0x40,0x40,0x42,0x44,0x58,0xC0,0x40,0x7F,0x40,0xC0,0x50,0x48,0x46,0x40,0x40,0x00,0x80,0x80,0x40,0x20,0x18,0x07,0x00,0x00,0x00,0x3F,0x40,0x40,0x40,0x40,0x78,0x00,/\*"光",2\*/};

uchar code lin[] ={0x00,0xF8,0x00,0x00,0xFF,0x40,0x20,0x18,0x0F,0x18,0x68,0x08,0x08,0x08,0x08,0x00,0x00,0x1F,0x00,0x00,0xFF,0x00,0x00,0x7F,0x21,0x21,0x3F,0x21,0x21,0x7F,0x00,0x00,/\*"临",3\*/};

uchar code sheng[] ={0x48,0x4A,0xFA,0x0A,0xFE,0x09,0xF9,0x49,0xA8,0x00,0x00,0xF0,0x00,0x00,0xFF,0x00,0x22,0x11,0x0B,0x04,0xFF,0x04,0x09,0x12,0x23,0x00,0x00,0x0F,0x40,0x80,0x7F,0x00,/\*"剩",0\*/};

uchar code yu[] ={0x80,0x80,0x40,0x20,0x50,0x48,0x44,0xC3,0x44,0x48,0x50,0x20,0x40,0x80,0x80,0x00,0x00,0x40,0x22,0x1A,0x02,0x42,0x82,0x7F,0x02,0x02,0x02,0x0A,0x12,0x60,0x00,0x00,/\*"余",1\*/};

uchar code xiang[] ={0x20,0x90,0x8C,0x87,0xEC,0x94,0x84,0x14,0xC8,0x47,0x44,0x4C,0x54,0xC4,0x04,0x00,0x40,0x20,0x18,0x06,0xFF,0x04,0x18,0x00,0xFF,0x92,0x92,0x92,0x92,0xFF,0x00,0x00,/\*"箱",2\*/};

uchar code zi[] ={0x80,0x82,0x82,0x82,0x82,0x82,0x82,0xE2,0xA2,0x92,0x8A,0x86,0x82,0x80,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x40,0x80,0x7F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,/\*"子",3\*/};

uchar code wei[] ={0x00,0x80,0x60,0xF8,0x07,0x10,0x90,0x10,0x11,0x16,0x10,0x10,0xD0,0x10,0x00,0x00,0x01,0x00,0x00,0xFF,0x40,0x40,0x41,0x5E,0x40,0x40,0x70,0x4E,0x41,0x40,0x40,0x00,/\*"位",4\*/};

uchar code guan[] ={0x88,0x64,0x23,0xA2,0xA6,0xAA,0xA2,0xBA,0xA4,0xA3,0xA2,0x26,0x2A,0xA2,0x62,0x00,0x00,0x00,0x00,0xFF,0x4A,0x4A,0x4A,0x4A,0x4A,0x4A,0x4B,0xF8,0x00,0x00,0x00,0x00,/\*"管",5\*/};

uchar code li[] ={0x04,0x84,0x84,0xFC,0x84,0x84,0x00,0xFE,0x92,0x92,0xFE,0x92,0x92,0xFE,0x00,0x00,0x20,0x60,0x20,0x1F,0x10,0x10,0x40,0x44,0x44,0x44,0x7F,0x44,0x44,0x44,0x40,0x00,/\*"理",6\*/};

uchar code yuan[] ={0x00,0x00,0xC0,0x5E,0x52,0x52,0x52,0x52,0x52,0x52,0x52,0x5E,0xC0,0x00,0x00,0x00,0x80,0x80,0x8F,0x40,0x40,0x20,0x10,0x0E,0x10,0x10,0x20,0x20,0x4F,0x80,0x00,0x00,/\*"员",7\*/};

uchar code she[] ={0x40,0x40,0x42,0xCC,0x00,0x40,0xA0,0x9E,0x82,0x82,0x82,0x9E,0xA0,0x20,0x20,0x00,0x00,0x00,0x00,0x3F,0x90,0x88,0x40,0x43,0x2C,0x10,0x28,0x46,0x41,0x80,0x80,0x00,/\*"设",8\*/};

uchar code zhi[] ={0x00,0x17,0x15,0xD5,0x55,0x57,0x55,0x7D,0x55,0x57,0x55,0xD5,0x15,0x17,0x00,0x00,0x40,0x40,0x40,0x7F,0x55,0x55,0x55,0x55,0x55,0x55,0x55,0x7F,0x40,0x40,0x40,0x00,/\*"置",9\*/};

uchar code cun[] ={0x04,0x04,0x84,0xE4,0x1C,0x07,0x24,0x24,0x24,0x24,0xA4,0x64,0x24,0x04,0x04,0x00,0x02,0x01,0x00,0xFF,0x00,0x02,0x02,0x42,0x82,0x7F,0x02,0x02,0x02,0x02,0x02,0x00,/\*"存",0\*/};

uchar code qu[] ={0x02,0x02,0xFE,0x92,0x92,0x92,0xFE,0x02,0x06,0xFC,0x04,0x04,0x04,0xFC,0x00,0x00,0x08,0x18,0x0F,0x08,0x08,0x04,0xFF,0x04,0x84,0x40,0x27,0x18,0x27,0x40,0x80,0x00,/\*"取",1\*/};

uchar code wu[] ={0x40,0x3C,0x10,0xFF,0x10,0x10,0x20,0x10,0x8F,0x78,0x08,0xF8,0x08,0xF8,0x00,0x00,0x02,0x06,0x02,0xFF,0x01,0x01,0x04,0x42,0x21,0x18,0x46,0x81,0x40,0x3F,0x00,0x00,/\*"物",2\*/};

uchar code ci[] ={0x00,0x00,0xF8,0x00,0x00,0xFF,0x20,0x20,0x00,0xFF,0x40,0x20,0x10,0x08,0x00,0x00,0x20,0x60,0x3F,0x20,0x10,0x1F,0x10,0x10,0x00,0x3F,0x40,0x40,0x40,0x40,0x78,0x00,/\*"此",0\*/};

uchar code gui[] ={0x10,0x10,0xD0,0xFF,0x90,0x10,0x00,0xFE,0x22,0x22,0x22,0x22,0x22,0xE2,0x02,0x00,0x04,0x03,0x00,0xFF,0x00,0x03,0x00,0x7F,0x44,0x44,0x44,0x44,0x44,0x47,0x40,0x00,/\*"柜",1\*/};

uchar code yi[] ={0x00,0x00,0xE2,0x82,0x82,0x82,0x82,0x82,0x82,0x82,0x82,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x78,0x00,0x00,/\*"已",2\*/};

uchar code man[] ={0x10,0x60,0x02,0x8C,0x00,0x24,0x24,0x2F,0xE4,0x24,0x24,0xE4,0x2F,0x24,0x24,0x00,0x04,0x04,0x7E,0x01,0x00,0xFF,0x11,0x09,0x07,0x19,0x09,0x07,0x49,0x91,0x7F,0x00,/\*"满",3\*/};

uchar code mi[] ={0x10,0x8C,0x44,0x04,0xE4,0x04,0x95,0xA6,0x44,0x24,0x14,0x84,0x44,0x94,0x0C,0x00,0x02,0x02,0x7A,0x41,0x41,0x43,0x42,0x7E,0x42,0x42,0x42,0x43,0xF8,0x00,0x00,0x00,/\*"密",0\*/};

uchar code ma[] ={0x04,0x84,0xE4,0x5C,0x44,0xC4,0x00,0x02,0xF2,0x82,0x82,0x82,0xFE,0x80,0x80,0x00,0x02,0x01,0x7F,0x10,0x10,0x3F,0x00,0x08,0x08,0x08,0x08,0x48,0x88,0x40,0x3F,0x00,/\*"码",1\*/};

uchar code shu[] ={0x88,0x68,0x1F,0xC8,0x08,0x10,0xC8,0x54,0x52,0xD1,0x12,0x94,0x08,0xD0,0x10,0x00,0x09,0x19,0x09,0xFF,0x05,0x00,0xFF,0x12,0x92,0xFF,0x00,0x5F,0x80,0x7F,0x00,0x00,/\*"输",0\*/};

uchar code ru[] ={0x00,0x00,0x00,0x00,0x00,0x01,0xE2,0x1C,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x40,0x20,0x10,0x0C,0x03,0x00,0x00,0x00,0x03,0x0C,0x30,0x40,0x80,0x80,0x00,/\*"入",1\*/};

uchar code qing3[] ={0x40,0x42,0xCC,0x00,0x00,0x44,0x54,0x54,0x54,0x7F,0x54,0x54,0x54,0x44,0x40,0x00,0x00,0x00,0x7F,0x20,0x10,0x00,0xFF,0x15,0x15,0x15,0x55,0x95,0x7F,0x00,0x00,0x00,/\*"请",0\*/};

uchar code qing1[] ={0x10,0x60,0x02,0x8C,0x00,0x44,0x54,0x54,0x54,0x7F,0x54,0x54,0x54,0x44,0x40,0x00,0x04,0x04,0x7E,0x01,0x00,0x00,0xFF,0x15,0x15,0x15,0x55,0x95,0x7F,0x00,0x00,0x00,/\*"清",0\*/};

uchar code an[] ={0x10,0x10,0x10,0xFF,0x90,0x20,0x98,0x88,0x88,0xE9,0x8E,0x88,0x88,0xA8,0x98,0x00,0x02,0x42,0x81,0x7F,0x00,0x00,0x80,0x84,0x4B,0x28,0x10,0x28,0x47,0x80,0x00,0x00,/\*"按",0\*/};

uchar code bing[] ={0x00,0x10,0x10,0x11,0xF6,0x10,0x10,0x10,0x10,0x10,0xF4,0x13,0x10,0x10,0x00,0x00,0x02,0x82,0x42,0x32,0x0F,0x02,0x02,0x02,0x02,0x02,0xFF,0x02,0x02,0x02,0x02,0x00,/\*"并",0\*/};

uchar code xia[] ={0x02,0x02,0x02,0x02,0x02,0x02,0xFE,0x02,0x02,0x42,0x82,0x02,0x02,0x02,0x02,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x01,0x06,0x00,0x00,0x00,/\*"下",1\*/};

uchar code que[] ={0x04,0x84,0xE4,0x5C,0x44,0xC4,0x20,0x10,0xE8,0x27,0x24,0xE4,0x34,0x2C,0xE0,0x00,0x02,0x01,0x7F,0x10,0x10,0x3F,0x80,0x60,0x1F,0x09,0x09,0x3F,0x49,0x89,0x7F,0x00,/\*"确",2\*/};

uchar code ren[] ={0x40,0x40,0x42,0xCC,0x00,0x00,0x00,0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0x90,0x48,0x20,0x18,0x07,0x00,0x07,0x18,0x20,0x40,0x80,0x00,/\*"认",3\*/};

uchar code jian[] ={0x40,0x30,0xEF,0x24,0x24,0x80,0xE4,0x9C,0x10,0x54,0x54,0xFF,0x54,0x7C,0x10,0x00,0x01,0x01,0x7F,0x21,0x51,0x26,0x18,0x27,0x44,0x45,0x45,0x5F,0x45,0x45,0x44,0x00,/\*"键",4\*/};

uchar code ding[] ={0x10,0x0C,0x44,0x44,0x44,0x44,0x45,0xC6,0x44,0x44,0x44,0x44,0x44,0x14,0x0C,0x00,0x80,0x40,0x20,0x1E,0x20,0x40,0x40,0x7F,0x44,0x44,0x44,0x44,0x44,0x40,0x40,0x00,/\*"定",6\*/};

uchar code hao[] ={0x80,0x80,0x80,0xBE,0xA2,0xA2,0xA2,0xA2,0xA2,0xA2,0xA2,0xBE,0x80,0x80,0x80,0x00,0x00,0x00,0x00,0x06,0x05,0x04,0x04,0x04,0x44,0x84,0x44,0x3C,0x00,0x00,0x00,0x00,/\*"号",0\*/};

uchar code jixu\_ji[] ={0x20,0x30,0xAC,0x63,0x10,0x00,0xFE,0x48,0x50,0x40,0xFF,0x40,0x50,0x48,0x00,0x00,0x22,0x67,0x22,0x12,0x12,0x00,0x7F,0x44,0x42,0x41,0x7F,0x41,0x42,0x44,0x40,0x00,/\*"继",1\*/};

uchar code xu[] ={0x20,0x30,0xAC,0x63,0x20,0x18,0x20,0x24,0xA4,0x24,0x3F,0xA4,0x24,0xA4,0x60,0x00,0x22,0x67,0x22,0x12,0x12,0x12,0x88,0x8A,0x4C,0x29,0x18,0x0F,0x18,0x28,0xC8,0x00,/\*"续",2\*/};

uchar code cheng[] ={0x00,0x00,0xF8,0x88,0x88,0x88,0x88,0x08,0x08,0xFF,0x08,0x09,0x0A,0xC8,0x08,0x00,0x80,0x60,0x1F,0x00,0x10,0x20,0x1F,0x80,0x40,0x21,0x16,0x18,0x26,0x41,0xF8,0x00,/\*"成",0\*/};

uchar code gong[] ={0x08,0x08,0x08,0xF8,0x08,0x08,0x08,0x10,0x10,0xFF,0x10,0x10,0x10,0xF0,0x00,0x00,0x10,0x30,0x10,0x1F,0x08,0x88,0x48,0x30,0x0E,0x01,0x40,0x80,0x40,0x3F,0x00,0x00,/\*"功",1\*/};

uchar code nin[] ={0x20,0x10,0x08,0xFC,0x23,0x10,0x88,0x67,0x04,0xF4,0x04,0x24,0x54,0x8C,0x00,0x00,0x40,0x30,0x00,0x77,0x80,0x81,0x88,0xB2,0x84,0x83,0x80,0xE0,0x00,0x11,0x60,0x00,/\*"您",0\*/};

uchar code lao[] ={0x10,0x0C,0x04,0xE4,0x84,0x84,0x85,0xF6,0x84,0x84,0x84,0x84,0x84,0x14,0x0C,0x00,0x08,0x0A,0x09,0x08,0x08,0x08,0x08,0xFF,0x08,0x08,0x08,0x08,0x08,0x08,0x08,0x00,/\*"牢",1\*/};

uchar code ji[] ={0x40,0x40,0x42,0xCC,0x00,0x00,0x00,0x84,0x84,0x84,0x84,0x84,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0x20,0x10,0x00,0x3F,0x40,0x40,0x40,0x40,0x41,0x40,0x70,0x00,/\*"记",0\*/};

uchar code men[] ={0x00,0xF8,0x01,0x06,0x00,0x00,0x02,0x02,0x02,0x02,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x40,0x80,0x7F,0x00,0x00,/\*"门",13\*/};

uchar code kai[] ={0x80,0x82,0x82,0x82,0xFE,0x82,0x82,0x82,0x82,0x82,0xFE,0x82,0x82,0x82,0x80,0x00,0x00,0x80,0x40,0x30,0x0F,0x00,0x00,0x00,0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,/\*"开",10\*/};

uchar code xia\_ci\_ci[] ={0x00,0x02,0x0C,0x80,0x60,0x80,0x40,0x30,0x0F,0xC8,0x08,0x08,0x28,0x18,0x00,0x00,0x02,0x02,0x7E,0x01,0x80,0x40,0x20,0x18,0x06,0x01,0x06,0x18,0x20,0x40,0x80,0x00,/\*"次",3\*/};

uchar code zai[] ={0x02,0x02,0xF2,0x92,0x92,0x92,0x92,0xFE,0x92,0x92,0x92,0x92,0xF2,0x02,0x02,0x00,0x04,0x04,0xFF,0x04,0x04,0x04,0x04,0x07,0x04,0x04,0x44,0x84,0x7F,0x04,0x04,0x00,/\*"再",0\*/};

uchar code shu\_zi\_liu[] ={0x20,0x20,0x20,0x20,0x20,0x20,0x21,0x22,0x2C,0x20,0x20,0x20,0x20,0x20,0x20,0x00,0x00,0x40,0x20,0x10,0x0C,0x03,0x00,0x00,0x00,0x01,0x02,0x04,0x18,0x60,0x00,0x00,/\*"六",0\*/};

uchar code deng[] ={0x00,0x00,0x8A,0x52,0x22,0x5A,0x46,0x40,0x47,0x58,0x24,0x52,0x88,0x04,0x00,0x00,0x01,0x41,0x40,0x4F,0x59,0x69,0x49,0x49,0x49,0x69,0x59,0x4F,0x40,0x41,0x01,0x00,/\*"登",3\*/};

uchar code lu[] ={0x80,0x80,0x82,0x92,0x92,0x92,0x92,0x92,0x92,0x92,0x92,0xFE,0x80,0x80,0x80,0x00,0x40,0x40,0x22,0x24,0x10,0x48,0x80,0x7F,0x04,0x08,0x18,0x24,0x22,0x40,0x40,0x00,/\*"录",4\*/};

uchar code shi[] ={0x00,0x40,0x30,0x1E,0x10,0x10,0x10,0xFF,0x10,0x10,0x10,0x10,0x10,0x00,0x00,0x00,0x81,0x81,0x41,0x21,0x11,0x0D,0x03,0x01,0x03,0x0D,0x11,0x21,0x41,0x81,0x81,0x00,/\*"失",0\*/};

uchar code bai[] ={0x00,0xFE,0x02,0xFA,0x02,0xFE,0x40,0x20,0xD8,0x17,0x10,0x10,0xF0,0x10,0x10,0x00,0x80,0x47,0x30,0x0F,0x10,0x67,0x80,0x40,0x21,0x16,0x08,0x16,0x21,0x40,0x80,0x00,/\*"败",1\*/};

uchar code jie[] ={0x00,0x00,0x00,0xFE,0x92,0x92,0x92,0xFE,0x92,0x92,0x92,0xFE,0x00,0x00,0x00,0x00,0x08,0x08,0x04,0x84,0x62,0x1E,0x01,0x00,0x01,0xFE,0x02,0x04,0x04,0x08,0x08,0x00,/\*"界",0\*/};

uchar code mian[] ={0x02,0x02,0xE2,0x22,0x22,0xF2,0x2E,0x22,0x22,0xE2,0x22,0x22,0xE2,0x02,0x02,0x00,0x00,0x00,0xFF,0x40,0x40,0x7F,0x49,0x49,0x49,0x7F,0x40,0x40,0xFF,0x00,0x00,0x00,/\*"面",1\*/};

uchar code zuo\_jiantou[] ={0x80,0x80,0xC0,0xC0,0xE0,0xA0,0x90,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x00,0x00,0x00,0x01,0x01,0x03,0x02,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,/\*"←",0\*/};

uchar code geng[] ={0x02,0x02,0xF2,0x92,0x92,0x92,0x92,0xFE,0x92,0x92,0x92,0x92,0xF2,0x02,0x02,0x00,0x80,0x80,0x87,0x4C,0x54,0x24,0x34,0x4F,0x44,0x44,0x84,0x84,0x87,0x80,0x80,0x00,/\*"更",0\*/};

uchar code gai[] ={0x04,0x84,0x84,0x84,0x84,0xFC,0x40,0x30,0xCC,0x0B,0x08,0x08,0xF8,0x08,0x08,0x00,0x00,0x7F,0x20,0x10,0x10,0x08,0x80,0x40,0x21,0x16,0x08,0x16,0x21,0x40,0x80,0x00,/\*"改",1\*/};

uchar code xuan[] ={0x40,0x40,0x42,0xCC,0x00,0x50,0x4E,0xC8,0x48,0x7F,0xC8,0x48,0x48,0x40,0x00,0x00,0x00,0x40,0x20,0x1F,0x20,0x50,0x4C,0x43,0x40,0x40,0x4F,0x50,0x50,0x5C,0x40,0x00,/\*"选",0\*/};

uchar code ze[] ={0x10,0x10,0xFF,0x10,0x00,0x82,0x86,0x4A,0x52,0xA2,0x52,0x4A,0x86,0x80,0x80,0x00,0x42,0x82,0x7F,0x01,0x00,0x10,0x12,0x12,0x12,0xFF,0x12,0x12,0x12,0x10,0x00,0x00,/\*"择",1\*/};

uchar code dui[] ={0x00,0x08,0x48,0x88,0x08,0xC8,0x38,0x00,0x90,0x10,0x10,0xFF,0x10,0x10,0x10,0x00,0x00,0x20,0x10,0x0C,0x03,0x04,0x18,0x00,0x00,0x43,0x80,0x7F,0x00,0x00,0x00,0x00,/\*"对",2\*/};

uchar code ying4[] ={0x00,0x00,0xFC,0x04,0x44,0x84,0x04,0x25,0xC6,0x04,0x04,0x04,0x04,0xE4,0x04,0x00,0x40,0x30,0x0F,0x40,0x40,0x41,0x4E,0x40,0x40,0x63,0x50,0x4C,0x43,0x40,0x40,0x00,/\*"应",3\*/};

uchar code ti[] ={0x10,0x10,0x10,0xFF,0x10,0x90,0x00,0xFE,0x92,0x92,0x92,0x92,0x92,0xFE,0x00,0x00,0x04,0x44,0x82,0x7F,0x01,0x80,0x42,0x3A,0x42,0x82,0xFE,0x92,0x92,0x92,0x82,0x00,/\*"提",0\*/};

uchar code ti\_shi\_shi[] ={0x40,0x40,0x42,0x42,0x42,0x42,0x42,0xC2,0x42,0x42,0x42,0x42,0x42,0x40,0x40,0x00,0x20,0x10,0x08,0x06,0x00,0x40,0x80,0x7F,0x00,0x00,0x00,0x02,0x04,0x08,0x30,0x00,/\*"示",1\*/};

uchar code mei[] ={0x10,0x60,0x02,0xCC,0x80,0x40,0x20,0x1E,0x02,0x02,0x02,0x3E,0x40,0x40,0x40,0x00,0x04,0x04,0x7E,0x01,0x80,0x80,0x83,0x4D,0x51,0x21,0x51,0x49,0x87,0x80,0x80,0x00,/\*"没",0\*/};

uchar code you[] ={0x04,0x04,0x04,0x84,0xE4,0x3C,0x27,0x24,0x24,0x24,0x24,0xE4,0x04,0x04,0x04,0x00,0x04,0x02,0x01,0x00,0xFF,0x09,0x09,0x09,0x09,0x49,0x89,0x7F,0x00,0x00,0x00,0x00,/\*"有",1\*/};

uchar code yi\_wang\_yi[] ={0x40,0x40,0x42,0xCC,0x20,0x20,0xAE,0xAA,0xAA,0xBF,0xAA,0xAA,0xAE,0x20,0x20,0x00,0x00,0x40,0x20,0x1F,0x20,0x40,0x67,0x50,0x48,0x46,0x48,0x50,0x67,0x40,0x40,0x00,/\*"遗",0\*/};

uchar code yi\_wang\_wang[] ={0x04,0x04,0x04,0xFC,0x84,0x84,0x85,0x86,0x84,0x84,0x84,0x84,0x84,0x04,0x04,0x00,0x40,0x38,0x00,0x00,0x3C,0x40,0x40,0x42,0x4C,0x40,0x40,0x70,0x04,0x08,0x30,0x00,/\*"忘",1\*/};

uchar code dou\_hao[] ={ 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x58,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,/\*"，",0\*/};

uchar code ju\_hao[] ={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x24,0x24,0x18,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,/\*"。",0\*/};

uchar code cuo\_wu\_cuo[] ={0x40,0x30,0xEF,0x24,0x64,0x48,0x48,0x7F,0x48,0x48,0x48,0x7F,0x48,0x48,0x40,0x00,0x01,0x01,0x7F,0x21,0x11,0x00,0xFF,0x49,0x49,0x49,0x49,0x49,0xFF,0x00,0x00,0x00,/\*"错",0\*/};

uchar code cuo\_wu\_wu[] ={0x40,0x42,0xCC,0x00,0x00,0x80,0x9E,0x92,0x92,0x92,0x92,0x92,0x9E,0x80,0x00,0x00,0x00,0x00,0x7F,0x20,0x94,0x84,0x44,0x24,0x14,0x0F,0x14,0x24,0x44,0x84,0x84,0x00,/\*"误",1\*/};

uchar code lian[] ={0x02,0xFE,0x92,0x92,0xFE,0x02,0x00,0x10,0x11,0x16,0xF0,0x14,0x13,0x10,0x00,0x00,0x10,0x1F,0x08,0x08,0xFF,0x04,0x81,0x41,0x31,0x0D,0x03,0x0D,0x31,0x41,0x81,0x00,/\*"联",0\*/};

uchar code xi[] ={0x00,0x00,0x22,0x32,0x2A,0xA6,0xA2,0x62,0x21,0x11,0x09,0x81,0x01,0x00,0x00,0x00,0x00,0x42,0x22,0x13,0x0B,0x42,0x82,0x7E,0x02,0x02,0x0A,0x12,0x23,0x46,0x00,0x00,/\*"系",1\*/};

#endif

#ifndef \_24C02\_H\_

#define \_24C02\_H\_

#include<intrins.h>

sbit SCL\_24C02=P0^7; //24C02串行时钟

sbit SDA\_24C02=P0^6; //24C02串行数据

void mDelay(uint t) //延时

{

uchar i;

while(t--)

{

for(i=0;i<125;i++)

{;}

}

}

void NOP\_24C02(void) //空操作

{

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

}

/\*起始条件\*/

void Start\_24C02(void)

{

SDA\_24C02=1;

SCL\_24C02=1;

NOP\_24C02();

SDA\_24C02=0;

NOP\_24C02();

}

/\*停止条件\*/

void Stop\_24C02(void)

{

SDA\_24C02=0;

SCL\_24C02=1;

NOP\_24C02();

SDA\_24C02=1;

NOP\_24C02();

}

/\*应答位\*/

void Ack\_24C02(void)

{

SDA\_24C02=0;

NOP\_24C02();

SCL\_24C02=1;

NOP\_24C02();

SCL\_24C02=0;

}

/\*反向应答位\*/

void NoAck\_24C02(void)

{

SDA\_24C02=1;

NOP\_24C02();

SCL\_24C02=1;

NOP\_24C02();

SCL\_24C02=0;

}

/\*发送数据子程序，Data为要求发送的数据\*/

void Send\_24C02(uchar Data)

{

uchar BitCounter=8;

uchar temp;

do

{

temp=Data;

SCL\_24C02=0;

NOP\_24C02();

if((temp&0x80)==0x80)

SDA\_24C02=1;

else

SDA\_24C02=0;

SCL\_24C02=1;

temp=Data<<1;

Data=temp;

BitCounter--;

}

while(BitCounter);

SCL\_24C02=0;

}

/\*读一字节的数据，并返回该字节值\*/

uchar Read\_24C02(void)

{

uchar temp=0;

uchar temp1=0;

uchar BitCounter=8;

SDA\_24C02=1;

do{

SCL\_24C02=0;

NOP\_24C02();

SCL\_24C02=1;

NOP\_24C02();

if(SDA\_24C02)

temp=temp|0x01;

else

temp=temp&0xfe;

if(BitCounter-1)

{

temp1=temp<<1;

temp=temp1;

}

BitCounter--;

}

while(BitCounter);

return(temp);

}

void Write\_dat\_24C02(uchar Data[],uchar Address,uchar Num)

{

uchar i;

uchar \*PData;

PData=Data;

for(i=0;i<Num;i++)

{

Start\_24C02();

Send\_24C02(0xa0);

Ack\_24C02();

Send\_24C02(Address+i);

Ack\_24C02();

Send\_24C02(\*(PData+i));

Ack\_24C02();

Stop\_24C02();

mDelay(20);

}

}

void Read\_dat\_24C02(uchar Data[],uchar Address,uchar Num)

{

uchar i;

uchar \*PData;

PData=Data;

for(i=0;i<Num;i++)

{

Start\_24C02();

Send\_24C02(0xa0);

Ack\_24C02();

Send\_24C02(Address+i);

Ack\_24C02();

Start\_24C02();

Send\_24C02(0xa1);

Ack\_24C02();

\*(PData+i)=Read\_24C02();

SCL\_24C02=0;

NoAck\_24C02();

Stop\_24C02();

}

}

#endif