#include<reg51.h>

#include "intrins.h"

#define uchar unsigned char

#define uint unsigned int

#define ulong unsigned long

sbit wei1 = P1^0; //数码管的位断开关

sbit wei2 = P1^1;

sbit wei3 = P1^2;

sbit wei4 = P1^3;

sbit wei5 = P1^4;

sbit wei6 = P1^5;

sbit wei7 = P1^6;

sbit wei8 = P1^7;

bit flag0;

uchar key; //键顺序码

uchar dis\_buf; //显示缓存

uchar temp;

unsigned int Num1=0;

uchar code table\_dm[17]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71,0x00};//共阴极数码管断码表

//uchar code table\_dm[11]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0xff};//共阳极数码管断码表

uint Display\_Buffer[8]={16,16,16,16,16,16,16,16};

uint a,n;

void delayms(uint xms)

{

int i,j;

for(i=xms;i>0;i--)

for(j=110;j>0;j--);

}

void system\_Ini()

{

TMOD = 0x00; //选择模式0 13位计数 最大计数8192

TH0 = (8192 - 1000)/32;

TL0 = (8192 - 1000)%32 ;

IE = 0x8A; //中断

TR0 = 1; //开启T0定时器

}

void keyscan(void)

{

uchar i;

temp = 0;

P2=0xF0; //高四位输入 列为高电平 行为低电平

delayms(1); //延时

temp=P2; //读P1口

temp=temp&0xF0; //屏蔽低四位

temp=~((temp>>4)|0xF0);

if(temp==1) // p1.4 被拉低

key=0; //第1个按键键值

else if(temp==2) // p1.5 被拉低

key=1; //第2个按键键值

else if(temp==4) // p1.6 被拉低

key=2; //第3个按键键值

else if(temp==8) // p1.7 被拉低

key=3; //第4个按键键值

else

key = 16;

P2=0x0F; //低四位输入 行为高电平 列为低电平

delayms(1); //延时

temp=P2; //读P1口

temp=temp&0x0F;

temp=~(temp|0xF0);

if(temp==1) //第一行 p1.1 被拉低(直接temp==2 是因为我们P1.0空的，是3\*4的矩阵键盘)

key=key+0;

else if(temp==2) //第一行 p1.1 被拉低(直接temp==2 是因为我们P1.0空的，是3\*4的矩阵键盘)

key=key+4;

else if(temp==4) //第二行 p1.2 被拉低

key=key+8;

else if(temp==8) //第三行 p1.3 被拉低

key=key+12;

else

key = 16;

dis\_buf = key; //键值入显示缓存

if(dis\_buf!=16&&a<3)

a++;

if(dis\_buf!=16&&flag0==0&&a>=2)

{

flag0=1;

if(dis\_buf==0)

dis\_buf=1;

else if(dis\_buf==1)

dis\_buf=2;

else if(dis\_buf==2)

dis\_buf=3;

else if(dis\_buf==3)

dis\_buf=10;

else if(dis\_buf==7)

dis\_buf=11;

else if(dis\_buf==8)

dis\_buf=7;

else if(dis\_buf==9)

dis\_buf=8;

else if(dis\_buf==10)

dis\_buf=9;

else if(dis\_buf==11)

dis\_buf=12;

else if(dis\_buf==12)

dis\_buf=14;

else if(dis\_buf==13)

dis\_buf=0;

else if(dis\_buf==14)

dis\_buf=15;

else if(dis\_buf==15)

dis\_buf=13;

for(i=7;i>0;i--)

Display\_Buffer[i]=Display\_Buffer[i-1];

Display\_Buffer[0]=dis\_buf;

}

if(dis\_buf==16)

{

flag0=0;

a=0;

}

}

void main(void)

{

system\_Ini();

while(1)

{

keyscan();

}

}

void T0zd(void) interrupt 1 //3定时器1的中断号 1定时器0的中断号 0外部中断1 2外部中断2 4串口中断

{

TH0 = (8192 - 1000)/32;

TL0 = (8192 - 1000)%32;

Num1++;

if(Num1>2)

{

Num1=0;

n++;

wei1=wei2=wei3=wei4=wei5=wei6=wei7=wei8=1;

if(n>7)

n=0;

if(n==0)

{

P3=table\_dm[Display\_Buffer[n]];

wei1=0;

wei2=1;

wei3=1;

wei4=1;

wei5=1;

wei6=1;

wei7=1;

wei8=1;

}

else if(n==1)

{

P3=table\_dm[Display\_Buffer[n]];

wei1=1;

wei2=0;

wei3=1;

wei4=1;

wei5=1;

wei6=1;

wei7=1;

wei8=1;

}

else if(n==2)

{

P3=table\_dm[Display\_Buffer[n]];

wei1=1;

wei2=1;

wei3=0;

wei4=1;

wei5=1;

wei6=1;

wei7=1;

wei8=1;

}

else if(n==3)

{

P3=table\_dm[Display\_Buffer[n]];

wei1=1;

wei2=1;

wei3=1;

wei4=0;

wei5=1;

wei6=1;

wei7=1;

wei8=1;

}

else if(n==4)

{

P3=table\_dm[Display\_Buffer[n]];

wei1=1;

wei2=1;

wei3=1;

wei4=1;

wei5=0;

wei6=1;

wei7=1;

wei8=1;

}

else if(n==5)

{

P3=table\_dm[Display\_Buffer[n]];

wei1=1;

wei2=1;

wei3=1;

wei4=1;

wei5=1;

wei6=0;

wei7=1;

wei8=1;

}

else if(n==6)

{

P3=table\_dm[Display\_Buffer[n]];

wei1=1;

wei2=1;

wei3=1;

wei4=1;

wei5=1;

wei6=1;

wei7=0;

wei8=1;

}

else if(n==7)

{

P3=table\_dm[Display\_Buffer[n]];

wei1=1;

wei2=1;

wei3=1;

wei4=1;

wei5=1;

wei6=1;

wei7=1;

wei8=0;

}

}

}