//从左向右矩阵键盘输出数字

#include<reg51.h>

#define GPIO\_DIG P0

#define GPIO\_KEY P1

#define uint unsigned int

sbit LSA=P2^2;

sbit LSB=P2^3;

sbit LSC=P2^4;

unsigned char code DIG\_CODE[17]={

0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,

0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};

unsigned char KeyValue;

unsigned char KeyState;

unsigned char DisplayData[8];

void Delay(uint i)

{

while(i--);

}

void KeyDown();

void DigDisplay();

void main(void)

{

KeyState=0;

while(1)

{

KeyDown();

if(KeyState==1)

{

DisplayData[7]=DisplayData[6];

DisplayData[6]=DisplayData[5];

DisplayData[5]=DisplayData[4];

DisplayData[4]=DisplayData[3];

DisplayData[3]=DisplayData[2];

DisplayData[2]=DisplayData[1];

DisplayData[1]=DisplayData[0];

DisplayData[0]=DIG\_CODE[KeyValue];

KeyState=0;

}

DigDisplay();

}

}

void DigDisplay()

{

unsigned char i;

for(i=0;i<8;i++)

{

switch(i) //??,????????,

{

case(0):

LSA=0;LSB=0;LSC=0; break;//???0?

case(1):

LSA=1;LSB=0;LSC=0; break;//???1?

case(2):

LSA=0;LSB=1;LSC=0; break;//???2?

case(3):

LSA=1;LSB=1;LSC=0; break;//???3?

case(4):

LSA=0;LSB=0;LSC=1; break;//???4?

case(5):

LSA=1;LSB=0;LSC=1; break;//???5?

case(6):

LSA=0;LSB=1;LSC=1; break;//???6?

case(7):

LSA=1;LSB=1;LSC=1; break;//???7?

}

GPIO\_DIG=DisplayData[i];//????

Delay（10）;

GPIO\_DIG=0x00;//??

}

}

void KeyDown(void)

{

unsigned int a=0;

GPIO\_KEY=0x0f;

if(GPIO\_KEY!=0x0f)

{

Delay(1000);

if(GPIO\_KEY!=0x0f)

{

KeyState=1;//?????

switch(GPIO\_KEY)

{

case(0X07): KeyValue=0;break;

case(0X0b): KeyValue=1;break;

case(0X0d): KeyValue=2;break;

case(0X0e): KeyValue=3;break;

}

GPIO\_KEY=0XF0;

Delay(1000);

switch(GPIO\_KEY)

{

case(0X70): KeyValue=KeyValue;break;

case(0Xb0): KeyValue=KeyValue+4;break;

case(0Xd0): KeyValue=KeyValue+8;break;

case(0Xe0): KeyValue=KeyValue+12;break;

}

while((a<500)&&(GPIO\_KEY!=0xf0)) //??????

{

Delay(1000);

a++;

}

a=0;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* ? ? ? : Delay10ms

\* ???? : ????,??10ms

\* ? ? : ?

\* ? ? : ?

//从右向左矩阵键盘输出数字（可以用于计算器显示）

#include<reg51.h>

#define GPIO\_DIG P0

#define GPIO\_KEY P1

//#define uint unsigned int

sbit LSA=P2^2;

sbit LSB=P2^3;

sbit LSC=P2^4;

unsigned char code DIG\_CODE[17]={

0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,

0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};

unsigned char KeyValue; //存放按键的值

unsigned char KeyState; //主函数显示按键是否按下 0未按下 1按下

unsigned char DisplayData[8]; //用来存放每一位的数字

void Delay(unsigned int i) //延迟函数

{

while(i--);

}

void KeyDown(); //为每一个按键赋值 先列后行

void DigDisplay(); //动态数码管的操作

void main(void)

{

KeyState=0;

while(1)

{

KeyDown();

if(KeyState==1)

{

DisplayData[7]=DisplayData[6];

DisplayData[6]=DisplayData[5];

DisplayData[5]=DisplayData[4];

DisplayData[4]=DisplayData[3];

DisplayData[3]=DisplayData[2];

DisplayData[2]=DisplayData[1];

DisplayData[1]=DisplayData[0];

DisplayData[0]=DIG\_CODE[KeyValue];

KeyState=0; //判断按键是否松开 若没有此句数码管将全亮

}

DigDisplay();

}

}

void DigDisplay()

{

unsigned char i;

for(i=0;i<8;i++)

{

switch(i)

{

case(0):

LSA=1;LSB=1;LSC=1; break;//第七位

case(1):

LSA=0;LSB=1;LSC=1; break;//第六位

case(2):

LSA=1;LSB=0;LSC=1; break;//第五位

case(3):

LSA=0;LSB=0;LSC=1; break;//第四位

case(4):

LSA=1;LSB=1;LSC=0; break;//第三位

case(5):

LSA=0;LSB=1;LSC=0; break;//第二位

case(6):

LSA=1;LSB=0;LSC=0; break;//第一位

case(7):

LSA=0;LSB=0;LSC=0; break;//第零位

}

GPIO\_DIG=DisplayData[i]; // DisplayData[0]对应第七位即数码管最后一位（第8位）

Delay(100); //3000从左往右闪烁 2000几乎一块闪1000加快

//500闪烁趋于稳定500之后越来越暗

GPIO\_DIG=0x00; //消隐

}

}

void KeyDown(void)

{

unsigned int a=0;

GPIO\_KEY=0x0f;

if(GPIO\_KEY!=0x0f)

{

Delay(1000);

if(GPIO\_KEY!=0x0f)

{

KeyState=1; //按键按下

switch(GPIO\_KEY)

{

case(0X07): KeyValue=0;break;

case(0X0b): KeyValue=1;break;

case(0X0d): KeyValue=2;break;

case(0X0e): KeyValue=3;break;

}

//赋值行

GPIO\_KEY=0XF0;

Delay(1000);

switch(GPIO\_KEY)

{

case(0X70): KeyValue=KeyValue;break;

case(0Xb0): KeyValue=KeyValue+4;break;

case(0Xd0): KeyValue=KeyValue+8;break;

case(0Xe0): KeyValue=KeyValue+12;break;

}

while((a<500)&&(GPIO\_KEY!=0xf0)) //一直按了此处循环500次或者松手跳

{ //出此函数

Delay(1000);

a++;

}

a=0;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///从00:00:00开始计时

#include<reg51.h>

#define GPIO\_DIG P0

#define uint unsigned int

#define uchar unsigned char

sbit LSA=P2^2;

sbit LSB=P2^3;

sbit LSC=P2^4;

uchar code DIG\_CODE[17]={

0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,

0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71};

uchar shi=0;

uchar fen=0;

uchar miao=0;

uchar DisplayData[8]; //??????????

void Delay(uint i) //????

{

while(i--);

}

void DigDisplay(); //????????

void time0Init()

{

TMOD|=0x01;

TH0=0xFC;

TL0=0x18;

ET0=1;

EA=1;

TR0=1;

}

void main(void)

{

time0Init();

while(1)

{

DigDisplay();

}

}

void DigDisplay()

{

uchar i;

for(i=0;i<8;i++)

{

switch(i)

{

case(0):

LSA=1;LSB=1;LSC=1; break;//???

case(1):

LSA=0;LSB=1;LSC=1; break;//???

case(2):

LSA=1;LSB=0;LSC=1; break;//???

case(3):

LSA=0;LSB=0;LSC=1; break;//???

case(4):

LSA=1;LSB=1;LSC=0; break;//???

case(5):

LSA=0;LSB=1;LSC=0; break;//???

case(6):

LSA=1;LSB=0;LSC=0; break;//???

case(7):

LSA=0;LSB=0;LSC=0; break;//???

}

GPIO\_DIG=DisplayData[i]; // DisplayData[0]?????????????(?8?)

Delay(100); //3000?????? 2000?????1000??

//500??????500??????

GPIO\_DIG=0x00; //??

}

}

void timer0() interrupt 1

{

static uint i;

TH0=0xFC;

TL0=0x18;

i++;

if(i==1000) //Ò»Ãë

{

i=0;

miao++;

if(miao==60)

{

miao=0;

fen++;

if(fen==60)

{

fen=0;

shi++;

if(shi==24)

{

shi=0;

}

}

}

DisplayData[0]=DIG\_CODE[miao%10];

DisplayData[1]=DIG\_CODE[miao/10];

DisplayData[3]=DIG\_CODE[fen%10];

DisplayData[4]=DIG\_CODE[fen/10];

DisplayData[6]=DIG\_CODE[shi%10];

DisplayData[7]=DIG\_CODE[shi/10];

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

按键 K1 K2 K3 K4 分别控制 选择 加 减 确定 00.00

#include<reg52.h>

#define uchar unsigned char

#define uint unsigned int

sbit wei1=P3^0;

sbit wei2=P3^1;

sbit wei3=P3^2;

sbit wei4=P3^3;

sbit set=P1^0;

sbit add=P1^1;

sbit cut=P1^2;

sbit ok=P1^3;

uchar code duan[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};

uchar aa,miao,fen,flag,cont,\_miao,\_fen;

bit flag1;

void delay(uchar t)

{

uchar i,j;

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

for(j=124;j>0;j--);

}

void display(uchar dat,uchar dat1)

{

P2=duan[dat/10];

wei1=1;

delay(5);

wei1=0;

P2=duan[dat%10]+0x80;

wei2=1;

delay(5);

wei2=0;

P2=duan[dat1/10];

wei3=1;

delay(5);

wei3=0;

P2=duan[dat1%10];

wei4=1;

delay(5);

wei4=0;

}

void display1(uchar dat,uchar dat1)

{

if(flag1==1)

{

if(flag==2)

{

P2=duan[dat1/10];

wei3=1;

delay(5);

wei3=0;

P2=duan[dat1%10];

wei4=1;

delay(5);

wei4=0;

}

else if(flag==1)

{

P2=duan[dat/10];

wei1=1;

delay(5);

wei1=0;

P2=duan[dat%10]+0x80;

wei2=1;

delay(5);

wei2=0;

}

}

else

{

P2=duan[dat/10];

wei1=1;

delay(5);

wei1=0;

P2=duan[dat%10]+0x80;

delay(5);

wei2=0;

P2=duan[dat1/10];

wei3=1;

delay(5);

wei3=0;

P2=duan[dat1%10];

wei4=1;

delay(5);

wei4=0;

}

}

void init()

{

TMOD=0X01;

TH0=(65535-50000)/256;

TL0=(65535-50000)%256;

ET0=1;

EA=1;

TR0=1;

}

void anjian()

{

if(set==0)

{

delay(2);

if(set==0)

{

display1(\_fen,\_miao);

if(flag<2) flag++;

else flag=1;

while(!set);

}

}

if(flag==0)

display(fen,miao);

else display1(\_fen,\_miao);

if(flag==1)

{

if(add==0)

{

delay(2);

if(add==0)

{

while(!add)

display1(\_fen,\_miao);

\_miao++;

if(\_miao==60) \_miao=0;

else if(\_miao<0) \_miao=59;

}

}

if(cut==0)

{

delay(2);

if(cut==0)

{

while(!cut)

display1(\_fen,\_miao);

\_miao--;

if(\_miao==0) \_miao=59;

}

}

}

if(flag==2)

{

delay(2);

if(add==0)

{

delay(2);

if(add==0)

{

while(!add)

display1(\_fen,\_miao);

\_fen++;

if(\_fen==60) \_fen=0;

}

}

if(cut==0)

{

delay(2);

if(cut==0)

{

while(!cut)

display1(\_fen,\_miao);

\_fen--;

if(\_fen==0)

\_fen=59;

}

}

}

if(ok==0)

{

delay(2);

if(ok==0)

{

flag=0;

fen=\_fen;

miao=\_miao;

display(fen,miao);

while(!ok);

}

}

}

void main()

{

init();

while(1)

{

anjian();

if(flag==1)

{

if(flag1==1) display1(\_fen,\_miao);

else display(\_fen,\_miao);

}

else if(flag==2)

{

if(flag1==1) display1(\_fen,\_miao);

else display(\_fen,\_miao);

}

else display(fen,miao);

}

}

void T0\_int0() interrupt 1

{

TH0=(65535-50000)/256;

TL0=(65535-50000)%256;

aa++;

cont++;

if(cont==10)

{

flag1=!flag1;

cont=0;

}

if(aa==20)

{

miao++;

aa=0;

if(miao==60)

{

fen++;

miao=0;

if(fen==60)

fen=0;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///普中开发板 独立按键控制时间 00.00.00 k1 k2 k3 k4 选择 加 减 确定

#include<reg51.h>

#define GPIO\_DIG P0

#define uint unsigned int

#define uchar unsigned char

sbit LSA=P2^2;

sbit LSB=P2^3;

sbit LSC=P2^4;

sbit set=P1^0;

sbit add=P1^1;

sbit cut=P1^2;

sbit ok=P1^3;

uchar code DIG\_CODE[17]={

0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,

0x7f,0x6f,};//0x77,0x7c,0x39,0x5e,0x79,0x71

uchar shi=0;

uchar fen=0;

uchar miao=0;

uchar aa,flag,cont,\_shi,\_fen,\_miao;

bit flag1;

uchar DisplayData[8]; //ÓÃÀ´´æ·ÅÊý×Ö

void Tdisplay(uchar dat,uchar dat1,uchar dat2); //Ê±¼äµÄÊ®Î»¸öÎ»·Ö±ðÈ¡³ö

void Tdisplay1(uchar dat,uchar dat1,uchar dat2);

void anjian();

void Delay(uint i) //ÑÓ³Ù

{

while(i--);

}

void DigDisplay(); //38ÒëÂëÆ÷µÄÑ¡Î»º¯Êý

void time0Init() //¶¨Ê±Æ÷0ÖÐ¶Ï

{

TMOD|=0x01;

TH0=0xFC; //1ms

TL0=0x18;

ET0=1;

EA=1;

TR0=1;

}

void main(void)

{

time0Init();

while(1)

{

anjian();

if(flag==1)

{

if(flag1==1) Tdisplay1(\_shi,\_fen,\_miao);

else Tdisplay(\_shi,\_fen,\_miao);

}

else if(flag==2)

{

if(flag1==1) Tdisplay1(\_shi,\_fen,\_miao);

else Tdisplay(\_shi,\_fen,\_miao);

}

else if(flag==3)

{

if(flag1==1) Tdisplay1(\_shi,\_fen,\_miao);

else Tdisplay(\_shi,\_fen,\_miao);

}

else Tdisplay(shi,fen,miao);

DigDisplay();

}

}

void Tdisplay(uchar dat,uchar dat1,uchar dat2)

{

DisplayData[1]=DIG\_CODE[dat%10]+0x80;

DisplayData[0]=DIG\_CODE[dat/10];

DisplayData[4]=DIG\_CODE[dat1%10]+0x80;

DisplayData[3]=DIG\_CODE[dat1/10];

DisplayData[7]=DIG\_CODE[dat2%10];

DisplayData[6]=DIG\_CODE[dat2/10];

}

void Tdisplay1(uchar dat,uchar dat1,uchar dat2)

{

if(flag1==1)

{

if(flag==3)

{

DisplayData[1]=DIG\_CODE[dat%10];

DisplayData[0]=DIG\_CODE[dat/10];

}

else if(flag==2)

{

DisplayData[4]=DIG\_CODE[dat1%10];

DisplayData[3]=DIG\_CODE[dat1/10];

}

else if(flag==1)

{

DisplayData[7]=DIG\_CODE[dat2%10];

DisplayData[6]=DIG\_CODE[dat2/10];

}

}

else

{

DisplayData[1]=DIG\_CODE[dat%10]+0x80;

DisplayData[0]=DIG\_CODE[dat/10];

DisplayData[4]=DIG\_CODE[dat1%10]+0x80;

DisplayData[3]=DIG\_CODE[dat1/10];

DisplayData[7]=DIG\_CODE[dat2%10];

DisplayData[6]=DIG\_CODE[dat2/10];

}

}

void anjian()

{

if(set==0)

{

Delay(200);

if(set==0)

{

Tdisplay1(\_shi,\_fen,\_miao);

if(flag<3) flag++;

else flag=1;

while(!set);

}

}

if(flag==0)

Tdisplay(\_shi,fen,miao);

else Tdisplay1(\_shi,\_fen,\_miao);

if(flag==1)

{

if(add==0)

{

Delay(200);

if(add==0)

{

while(!add)

Tdisplay1(\_shi,\_fen,\_miao);

\_miao++;

if(\_miao==60) \_miao=0;

else if(\_miao<0) \_miao=59;

}

}

if(cut==0)

{

Delay(2);

if(cut==0)

{

while(!cut)

Tdisplay1(\_shi,\_fen,\_miao);

\_miao--;

if(\_miao==0) \_miao=59;

}

}

}

if(flag==2)

{

Delay(200);

if(add==0)

{

Delay(200);

if(add==0)

{

while(!add)

Tdisplay1(\_shi,\_fen,\_miao);

\_fen++;

if(\_fen==60) \_fen=0;

}

}

if(cut==0)

{

Delay(200);

if(cut==0)

{

while(!cut)

Tdisplay1(\_shi,\_fen,\_miao);

\_fen--;

if(\_fen==0)

\_fen=59;

}

}

}

if(flag==3)

{

Delay(200);

if(add==0)

{

Delay(200);

if(add==0)

{

while(!add)

Tdisplay1(\_shi,\_fen,\_miao);

\_shi++;

if(\_shi==24) \_shi=0;

}

}

if(cut==0)

{

Delay(200);

if(cut==0)

{

while(!cut)

Tdisplay1(\_shi,\_fen,\_miao);

\_shi--;

if(\_shi==0)

\_shi=24;

}

}

}

if(ok==0)

{

Delay(200);

if(ok==0)

{

flag=0;

shi=\_shi;

fen=\_fen;

miao=\_miao;

Tdisplay(\_shi,\_fen,\_miao);

while(!ok);

}

}

}

void DigDisplay()

{

uchar i;

for(i=0;i<8;i++)

{

switch(i)

{

case(0):

LSA=0;LSB=0;LSC=0; break;//???

case(1):

LSA=1;LSB=0;LSC=0; break;//???

case(2):

LSA=0;LSB=1;LSC=0; break;//???

case(3):

LSA=1;LSB=1;LSC=0; break;//???

case(4):

LSA=0;LSB=0;LSC=1; break;//???

case(5):

LSA=1;LSB=0;LSC=1; break;//???

case(6):

LSA=0;LSB=1;LSC=1; break;//???

case(7):

LSA=1;LSB=1;LSC=1; break;//???

}

GPIO\_DIG=DisplayData[i]; // DisplayData[0]?????????????(?8?)

Delay(100); //3000?????? 2000?????1000??

//500??????500??????

GPIO\_DIG=0x00; //ÏûÒþ

}

}

void timer0() interrupt 1

{

static uint i;

TH0=0xFC;

TL0=0x18;

i++;

cont++;

if(cont==1000)

{

flag1=~flag1;

cont=0;

}

if(i==1000) //Ò»Ãë

{

i=0;

miao++;

if(miao==60)

{

miao=0;

fen++;

if(fen==60)

{

fen=0;

shi++;

if(shi==24)

{

shi=0;

}

}

}

}

}