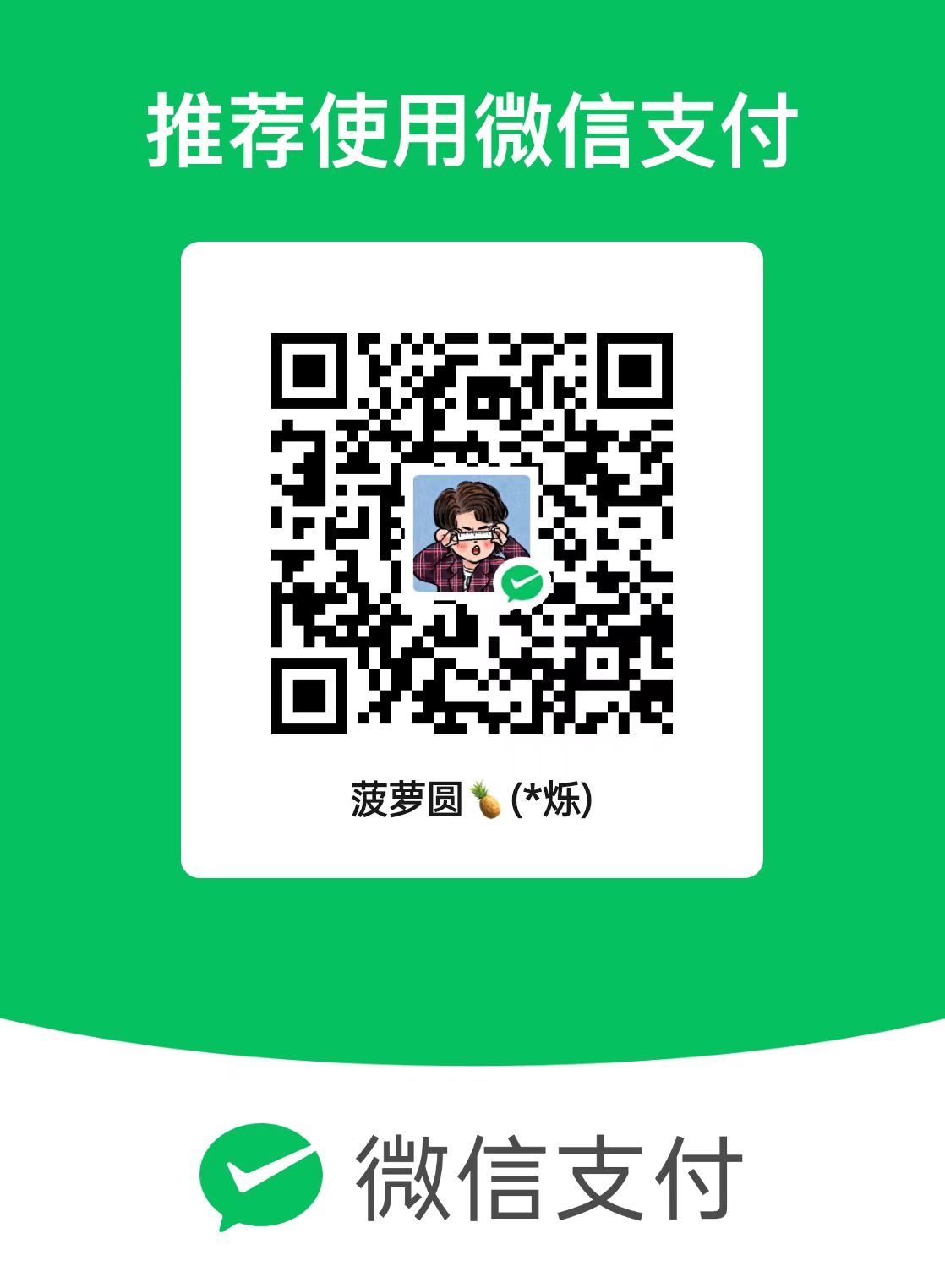
朋友们，写代码不易，使用源码请打赏支持喔~

量力而行，无论5元10元甚至是50元，都是对up的辛苦劳动成果的认可，都很感谢你！^\_^

如果有什么问题，请私信up，只要有空都会回复你哒！（up: aSerialKiller）

谢谢~祝你天天开心！o(\*￣▽￣\*)ブ

****

**//注：up使用的开发板为普中51A2单核，若有不同需要根据原理图自行修改相关引脚代码，否则可能出错。常见问题及解决方案见20221111增强改进版时钟文件。**

//------------基础定义------------

#include<reg51.h>

#include<intrins.h>

#define uchar unsigned char

#define uint unsigned int

//----------矩阵键盘matkey函数-----------

#define KeyOut P1

unsigned char KeyValue; //用来存放读取到的键值

unsigned char KeyState; //用来存放按键状态

//--------------定时器--------------

//晶振频率11.0592MHZ/s

#define T0time 46080

// TH0=(65536-T0time)/256;

// TL0=(65536-T0time)%256;

uchar count=0; //计时1秒 T0需要循环的次数

//--------------数码管---------------

uchar table[]={~0xC0,~0xF9,~0xA4,~0xB0,~0x99,~0x92,~0x82,~0xF8,~0x80,~0x90,~0xBF};

//共阴数码管 0 1 2 3 4 5 6 7 8 9 -

uchar table\_dot[]={0xbf,0x86,0xdb,0xcf,0xe6,0xed,0xfd,0x87,0xff,0xef,~0xBF};

//共阴数码管 带小数点的0~9、-

sbit smg0=P2^2; //数码管位选

sbit smg1=P2^3;

sbit smg2=P2^4;

#define duan P0 //数码管段选

//---------------闹钟----------------

//1·1 进入闹钟设置界面/返回时钟运行界面

//1·2 闹钟开/关

//1·3 闹钟时间减

//1·4 闹钟时间加

sbit BEEP=P1^5;//蜂鸣器引脚

uchar shi1=0,fen1=0,miao1=10;//闹钟时、分、秒初始数据

bit alarmOrNot;

/\*

表明是否在闹钟设置界面的标志符号。

0:不在闹钟界面,时钟正常运行;

1:在闹钟界面,时钟停止运行。

\*/

bit alarmOnOff;//表明闹钟是否开启的标志符号

uchar alarmSetLocation=3;//记录闹钟设置时的所选位置

//闹钟设置时选中位置闪烁

uchar alarmBlankCount=255;//空白显示 计数标志

uchar alarmNormalCount=255;//正常显示 计数标志

//---------------时钟------------------

uchar shi=0,fen=0,miao=0; //时钟时、分、秒 初始数据

bit clockSetOrNot;

/\*

表明是否在时钟设置模式的标志符号。

0:不在设置模式,时钟正常运行;

1:在设置模式,时钟停止运行。

\*/

sbit keyClockSetOrNot=P3^1;//开启/退出设置模式按钮

sbit keyClockSetLocation=P3^0;//设置时or分or秒

uchar clockSetLocation=3;//记录时钟设置时的所选位置

sbit keyClockSetDec=P3^2;//时间减

sbit keyClockSetAdd=P3^3;//时间加

//时钟设置时选中位置闪烁

uchar clockBlankCount=255;//空白显示 计数标志

uchar clockNormalCount=255;//正常显示 计数标志

//-----------------------------------------------------------------------

void delay\_10us(uint ten\_us)//ten\_us为1时，大约延时10us

{

while(ten\_us--);

}

void delay\_10ms(uchar c) //矩阵键盘用延时函数 10ms

{

uchar a,b;

for(;c>0;c--)

for(b=38;b>0;b--)

for(a=130;a>0;a--);

}

void delay() //短延时

{

uchar j;

for(j=0; j<64; j++);

}

void long\_delay() //长延时

{

uchar i;

uchar j;

for(i=0; i<255; i++)

for(j=0; j<64; j++);

}

void timer0\_initial()//--------------------------计时器T0初始化

{

TMOD=0x01;

TH0=(65536-T0time)/256;//晶振为11.0592MHZ

TL0=(65536-T0time)%256;

EA=1;

ET0=1;

TR0=1;

TR1=0;

}

void serviceTimer0() interrupt 1//---------------------------中断程序

{

TH0=(65536-T0time)/256;//定时器重新赋值

TL0=(65536-T0time)%256;

count++;

if(20==count)//计时器次数达到20次，秒+1

{

miao++;

count=0;//计时器次数清零

}

if(60==miao)//秒计数达到60次，分+1

{

fen++;

miao=0;//秒计数清零

}

if(60==fen)//分计数达到60次，时+1

{

shi++;

fen=0;//分计数清零

}

if(24==shi)//时计数达到24次，时清零

{

shi=0;

}

}

void smg(uchar wei,uchar number)//数码管单个位置亮一下

{

switch(wei)//译码并位选

{

case 7:smg2=1;smg1=1;smg0=1;break;

case 6:smg2=1;smg1=1;smg0=0;break;

case 5:smg2=1;smg1=0;smg0=1;break;

case 4:smg2=1;smg1=0;smg0=0;break;

case 3:smg2=0;smg1=1;smg0=1;break;

case 2:smg2=0;smg1=1;smg0=0;break;

case 1:smg2=0;smg1=0;smg0=1;break;

case 0:smg2=0;smg1=0;smg0=0;break;

}

duan=number;//段选

delay();//短延时

duan=0x00;//消影

}

void clock\_display()//--------------------------------数码管动态显示时钟界面

{

//时的十位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(0==clockSetLocation&&clockBlankCount>0)

{

smg(7,0x00);

clockBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(0==clockSetLocation&&clockNormalCount>0)

{

smg(7,table[shi/10]);

clockNormalCount--;

}else if(0==clockSetLocation)

{

clockBlankCount=255;//闪烁标志初始化

clockNormalCount=255;

}else

smg(7,table[shi/10]);

//时的个位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(0==clockSetLocation&&clockBlankCount>0)

{

smg(6,0x00);

clockBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(0==clockSetLocation&&clockNormalCount>0)

{

smg(6,table[shi%10]);

clockNormalCount--;

}else if(0==clockSetLocation)

{

clockBlankCount=255;//闪烁标志初始化

clockNormalCount=255;

}else

smg(6,table[shi%10]);

//-

smg(5,table[10]);

//分的十位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(1==clockSetLocation&&clockBlankCount>0)

{

smg(4,0x00);

clockBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(1==clockSetLocation&&clockNormalCount>0)

{

smg(4,table[fen/10]);

clockNormalCount--;

}else if(1==clockSetLocation)

{

clockBlankCount=255;//闪烁标志初始化

clockNormalCount=255;

}else

smg(4,table[fen/10]);

//分的个位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(1==clockSetLocation&&clockBlankCount>0)

{

smg(3,0x00);

clockBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(1==clockSetLocation&&clockNormalCount>0)

{

smg(3,table[fen%10]);

clockNormalCount--;

}else if(1==clockSetLocation)

{

clockBlankCount=255;//闪烁标志初始化

clockNormalCount=255;

}else

smg(3,table[fen%10]);

//-

smg(2,table[10]);

//秒的十位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(2==clockSetLocation&&clockBlankCount>0)

{

smg(1,0x00);

clockBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(2==clockSetLocation&&clockNormalCount>0)

{

smg(1,table[miao/10]);

clockNormalCount--;

}else if(2==clockSetLocation)

{

clockBlankCount=255;//闪烁标志初始化

clockNormalCount=255;

}else

smg(1,table[miao/10]);

//秒的个位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(2==clockSetLocation&&clockBlankCount>0)

{

smg(0,0x00);

clockBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(2==clockSetLocation&&clockNormalCount>0)

{

smg(0,table[miao%10]);

clockNormalCount--;

}else if(2==clockSetLocation)

{

clockBlankCount=255;//闪烁标志初始化

clockNormalCount=255;

}else

smg(0,table[miao%10]);

}

void clock\_set()//-----------------------按键功能：判断是否进入设置模式，并且进行时间初始化设置。

{

if(0==keyClockSetOrNot) //如果按下了设置模式总开关

{

delay\_10ms(3);//消抖

if(0==keyClockSetOrNot)//再次检测按键状态

{

clockSetOrNot=!clockSetOrNot;//标志符号，说明进入/退出设置模式

if(1==clockSetOrNot) //若进入设置模式（标识符号为1），时钟停止运行。

{

TR0=0;

clockSetLocation=0;

}else //否则时钟正常运行。

{

TR0=1;

clockSetLocation=3;

}

while(0==keyClockSetOrNot);//长按停留在此

delay\_10ms(3);//消抖

}

}

if(1==clockSetOrNot) //进入了设置模式，设置按钮可以使用了

{

if(0==keyClockSetLocation)//如果按下调整位置按钮

{

delay\_10ms(3);//消抖

if(0==keyClockSetLocation)//再次检测按键状态

{

clockSetLocation++;//位置标志+1

if(3==clockSetLocation)clockSetLocation=0;

while(0==keyClockSetLocation);//长按停留在此

delay\_10ms(3);//消抖

}

}

if(0==clockSetLocation)//调整小时位

{

if(0==keyClockSetAdd)//按下“加”按键

{

delay\_10ms(3);//消抖

if(0==keyClockSetAdd)//再次检测按键状态

{

shi++;

if(24==shi)shi=0;//小时位加到24则重置为0

while(0==keyClockSetAdd);//长按停留在此

delay\_10ms(3);//消抖

}

}

if(0==keyClockSetDec)//按下“减”按键

{

delay\_10ms(3);//消抖

if(0==keyClockSetDec)//再次检测按键状态

{

if(0==shi)shi=24;//小时位减到0则重置为23

shi--;

while(0==keyClockSetDec);//长按停留在此

delay\_10ms(3);//消抖

}

}

}

if(1==clockSetLocation)//调整分钟位

{

if(0==keyClockSetAdd)//按下“加”按键

{

delay\_10ms(3);//消抖

if(0==keyClockSetAdd)//再次检测按键状态

{

fen++;

if(60==fen)fen=0;//分钟位加到60则重置为0

while(0==keyClockSetAdd);//长按停留在此

delay\_10ms(3);//消抖

}

}

if(0==keyClockSetDec)//按下“减”按键

{

delay\_10ms(3);//消抖

if(0==keyClockSetDec)//再次检测按键状态

{

if(0==fen)fen=60;//分钟位减到0则重置为60

fen--;

while(0==keyClockSetDec);//长按停留在此

delay\_10ms(3);//消抖

}

}

}

if(2==clockSetLocation)//调整秒钟位

{

if(0==keyClockSetAdd)//按下“加”按键

{

delay\_10ms(3);//消抖

if(0==keyClockSetAdd)//再次检测按键状态

{

miao++;

if(60==miao)miao=0;//秒钟位加到60则重置为0

while(0==keyClockSetAdd);//长按停留在此

delay\_10ms(3);//消抖

}

}

if(0==keyClockSetDec)//按下“减”按键

{

delay\_10ms(3);//消抖

if(0==keyClockSetDec)//再次检测按键状态

{

if(0==miao)miao=60;//秒钟位减到0则重置为60

miao--;

while(0==keyClockSetDec);//长按停留在此

delay\_10ms(3);//消抖

}

}

}

}

}

void matkey(void)//-----------------------------矩阵键盘函数

//函数功能：

//1、检测是否有按键按下

//2、如果有，则状态标志KeyState置1并判断按下的是哪个键

//3、返回键值到KeyValue(键值是从0开始的)

{

uchar a;//用于松手检测

KeyOut=0x0f;

if(KeyOut!=0x0f)//如果有按键按下，则开始判断是按下了哪个键

{

delay\_10ms(3);//消抖

if(KeyOut!=0x0f)

{

KeyState=1;

//测试列

KeyOut=0X0F;

// delay\_10ms(3);

switch(KeyOut)

{

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

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

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

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

// default: KeyValue=17; //检测出错回复17意思是把数码管全灭掉。

}

//测试行

KeyOut=0XF0;

delay\_10ms(3);

switch(KeyOut)

{

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

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

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

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

// default: KeyValue=17;

}

while((a<50)&&(KeyOut!=0xf0)) //检测按键松手检测

{

delay\_10ms(3);

a++;

}

a=0;

}

}

}

void alarm\_set()//------------------------------------------闹钟设计

{

matkey();//检测矩阵键盘

if(1==KeyState)//如果有矩阵键盘按下

{

KeyState=0;//矩阵键盘状态重置为0

if(0==KeyValue)//如果按下矩阵键盘0

{

alarmOrNot=!alarmOrNot;//是否处在闹钟界面的标志符号取反

alarmSetLocation=3;//重置闹钟界面的设置位置

}

if(1==alarmOrNot)//如果处在闹钟界面，那么设置闹钟的按钮就可以使用了

{

if(1==KeyValue)//如果按下矩阵键盘1

{

alarmOnOff=!alarmOnOff;//是否开启闹钟的标志符号取反

alarmSetLocation=3;//重置闹钟界面的设置位置

}

if(4==KeyValue)//如果按下调整位置按钮

{

alarmSetLocation++;//位置标志+1

if(alarmSetLocation>=4)alarmSetLocation=0;

}

if(0==alarmSetLocation)//调整小时位

{

if(6==KeyValue)//按下“加”按键

{

shi1++;

if(24==shi1)shi1=0;//小时位加到24则重置为0

}

if(5==KeyValue)//按下“减”按键

{

if(0==shi1)shi1=24;//小时位减到0则重置为24

shi1--;

}

}

if(1==alarmSetLocation)//调整分钟位

{

if(6==KeyValue)//按下“加”按键

{

fen1++;

if(60==fen1)fen1=0;//分钟位加到60则重置为0

}

if(5==KeyValue)//按下“减”按键

{

if(0==fen1)fen1=60;//分钟位减到0则重置为60

fen1--;

}

}

if(2==alarmSetLocation)//调整秒钟位

{

if(6==KeyValue)//按下“加”按键

{

miao1++;

if(60==miao1)miao1=0;//秒钟位加到60则重置为0

}

if(5==KeyValue)//按下“减”按键

{

if(0==miao1)miao1=60;//秒钟位减到0则重置为60

miao1--;

}

}

}

}

}

void alarm\_display()//----------------------------数码管动态显示闹钟界面

{

//smg(7,0xf7);//'A.'表示ALARM,闹钟界面

if(0==alarmOnOff)//如果闹钟关闭

smg(7,0xf3);//'P.'表示STOP,闹钟关闭

if(1==alarmOnOff)//如果闹钟开启

smg(7,0xf9);//'E.'表示OPEN,闹钟开启

//显示出闹钟的设定时间

//时的十位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(0==alarmSetLocation&&alarmBlankCount>0)

{

smg(5,0x00);

alarmBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(0==alarmSetLocation&&alarmNormalCount>0)

{

smg(5,table[shi1/10]);

alarmNormalCount--;

}else if(0==alarmSetLocation)

{

alarmBlankCount=255;//闪烁标志初始化

alarmNormalCount=255;

}else

smg(5,table[shi1/10]);

//时的个位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(0==alarmSetLocation&&alarmBlankCount>0)

{

smg(4,0x00);

alarmBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(0==alarmSetLocation&&alarmNormalCount>0)

{

smg(4,table\_dot[shi1%10]);

alarmNormalCount--;

}else if(0==alarmSetLocation)

{

alarmBlankCount=255;//闪烁标志初始化

alarmNormalCount=255;

}else

smg(4,table\_dot[shi1%10]);

//分的十位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(1==alarmSetLocation&&alarmBlankCount>0)

{

smg(3,0x00);

alarmBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(1==alarmSetLocation&&alarmNormalCount>0)

{

smg(3,table[fen1/10]);

alarmNormalCount--;

}else if(1==alarmSetLocation)

{

alarmBlankCount=255;//闪烁标志初始化

alarmNormalCount=255;

}else

smg(3,table[fen1/10]);

//分的个位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(1==alarmSetLocation&&alarmBlankCount>0)

{

smg(2,0x00);

alarmBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(1==alarmSetLocation&&alarmNormalCount>0)

{

smg(2,table\_dot[fen1%10]);

alarmNormalCount--;

}else if(1==alarmSetLocation)

{

alarmBlankCount=255;//闪烁标志初始化

alarmNormalCount=255;

}else

smg(2,table\_dot[fen1%10]);

//秒的十位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(2==alarmSetLocation&&alarmBlankCount>0)

{

smg(1,0x00);

alarmBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(2==alarmSetLocation&&alarmNormalCount>0)

{

smg(1,table[miao1/10]);

alarmNormalCount--;

}else if(2==alarmSetLocation)

{

alarmBlankCount=255;//闪烁标志初始化

alarmNormalCount=255;

}else

smg(1,table[miao1/10]);

//秒的个位

//如果正在调整此位置且闪烁空白计数不为0，则此位置继续空白显示

if(2==alarmSetLocation&&alarmBlankCount>0)

{

smg(0,0x00);

alarmBlankCount--;

}

//如果正在调整此位置且闪烁正常计数不为0，则此位置继续正常显示

else if(2==alarmSetLocation&&alarmNormalCount>0)

{

smg(0,table[miao1%10]);

alarmNormalCount--;

}else if(2==alarmSetLocation)

{

alarmBlankCount=255;//闪烁标志初始化

alarmNormalCount=255;

}else

smg(0,table[miao1%10]);

}

void delay\_ms(uint x)//闹钟执行用 毫秒延时函数

{

uint i,j;

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

for(j=0;j<115;j++);

}

void alarm\_exe()//---------------------------------------------闹钟执行

{

if( (shi==shi1&&fen==fen1&&miao==miao1) && 1==alarmOnOff && 0==clockSetOrNot && 0==alarmOrNot)

//只有当以下条件【都满足】的时候，闹钟才会响起：

//1、时钟的时分秒=闹钟的时分秒

//2、闹钟处于开启状态

//3、当前不处在时钟设置界面

//4、当前不处在闹钟设置界面

{

while((1==keyClockSetOrNot)&&(1==keyClockSetLocation)&&(1==keyClockSetAdd)&&(1==keyClockSetDec))

{

matkey();

BEEP=~BEEP;

smg(7,0x77);//'A'

BEEP=~BEEP;

smg(6,0xb8);//'L.'

BEEP=~BEEP;

smg(5,table[shi1/10]);

BEEP=~BEEP;

smg(4,table\_dot[shi1%10]);

BEEP=~BEEP;

smg(3,table[fen1/10]);

BEEP=~BEEP;

smg(2,table\_dot[fen1%10]);

BEEP=~BEEP;

smg(1,table[miao1/10]);

BEEP=~BEEP;

smg(0,table[miao1%10]);

BEEP=~BEEP;

}

while((0==keyClockSetOrNot)||(0==keyClockSetLocation)||(0==keyClockSetAdd)||(0==keyClockSetDec));

delay\_10ms(3);

}

}

void start\_show()//---------------------------------------开机展示

{

uint t=1000;

while(t--)//LHL ELEC.

{

smg(7,0x38);//L

smg(6,0x76);//H

smg(5,0x38);//L

smg(4,0x00);//空

smg(3,0x79);//E

smg(2,0x38);//L

smg(1,0x79);//E

smg(0,0xb9);//C.

}

t=1000;//t重置

while(t--)//CLO.

{

smg(7,0x39);//C

smg(6,0x38);//L

smg(5,0x3f);//O

smg(4,0xb9);//C.

smg(3,0);//空

smg(2,0);//空

smg(1,0);//空

smg(0,0);//空

}

t=1000;//t重置

while(t--)//20221101（完成时钟部分的日期）

{

smg(7,table[2]);

smg(6,table[0]);

smg(5,table[2]);

smg(4,table[2]);

smg(3,table[1]);

smg(2,table[1]);

smg(1,table[0]);

smg(0,table[1]);

}

t=1000;//t重置

while(t--)//ALA.

{

smg(7,0x77);//A

smg(6,0x38);//L

smg(5,0xf7);//A.

smg(4,0);

smg(3,0);

smg(2,0);

smg(1,0);

smg(0,0);

}

t=1000;//t重置

while(t--)//20221105（完成闹钟部分的日期）

{

smg(7,table[2]);

smg(6,table[0]);

smg(5,table[2]);

smg(4,table[2]);

smg(3,table[1]);

smg(2,table[1]);

smg(1,table[0]);

smg(0,table[5]);

}

t=1000;//t重置

while(t--)//good job（对自己鼓励的话语）

{

smg(7,0x6f);//g

smg(6,0x5c);//o（下方）

smg(5,0x5c);//o（下方）

smg(4,0x5e);//d

smg(3,0x00);//空

smg(2,0x0e);//j

smg(1,0x5c);//o（下方）

smg(0,0x7c);//b

}

t=800;//t重置

while(t--)//go

{

smg(7,0x6f);//g

smg(6,0x63);//o（上方）

smg(5,0x00);//空

smg(4,0x00);//空

smg(3,0x00);//空

smg(2,0x00);//空

smg(1,0x00);//空

smg(0,0x00);//空

}

}

void main()

{

start\_show(); //开机展示

timer0\_initial(); //定时器T0初始化

clockSetOrNot=0; //设置模式标志符号初始化为0，时钟正常工作

KeyState=0; //矩阵键盘状态符号初始化为0

alarmOrNot=0; //闹钟界面标志符号初始化为0，默认在时钟界面

alarmOnOff=0; //闹钟默认关闭

while(1)

{

if(0==alarmOrNot)//如果不在闹钟界面，则来到动态数码管时钟显示

{

clock\_display(); //数码管动态显示时钟

clock\_set(); //时钟时间设置

}

if(0==clockSetOrNot)//如果不在时钟设定界面，则可以进行闹钟的设定

alarm\_set();

if(1==alarmOrNot)//如果如果在闹钟界面，则来到动态数码管闹钟显示

{

alarm\_display();//数码管动态显示闹钟界面

}

alarm\_exe();//闹钟执行

}

}