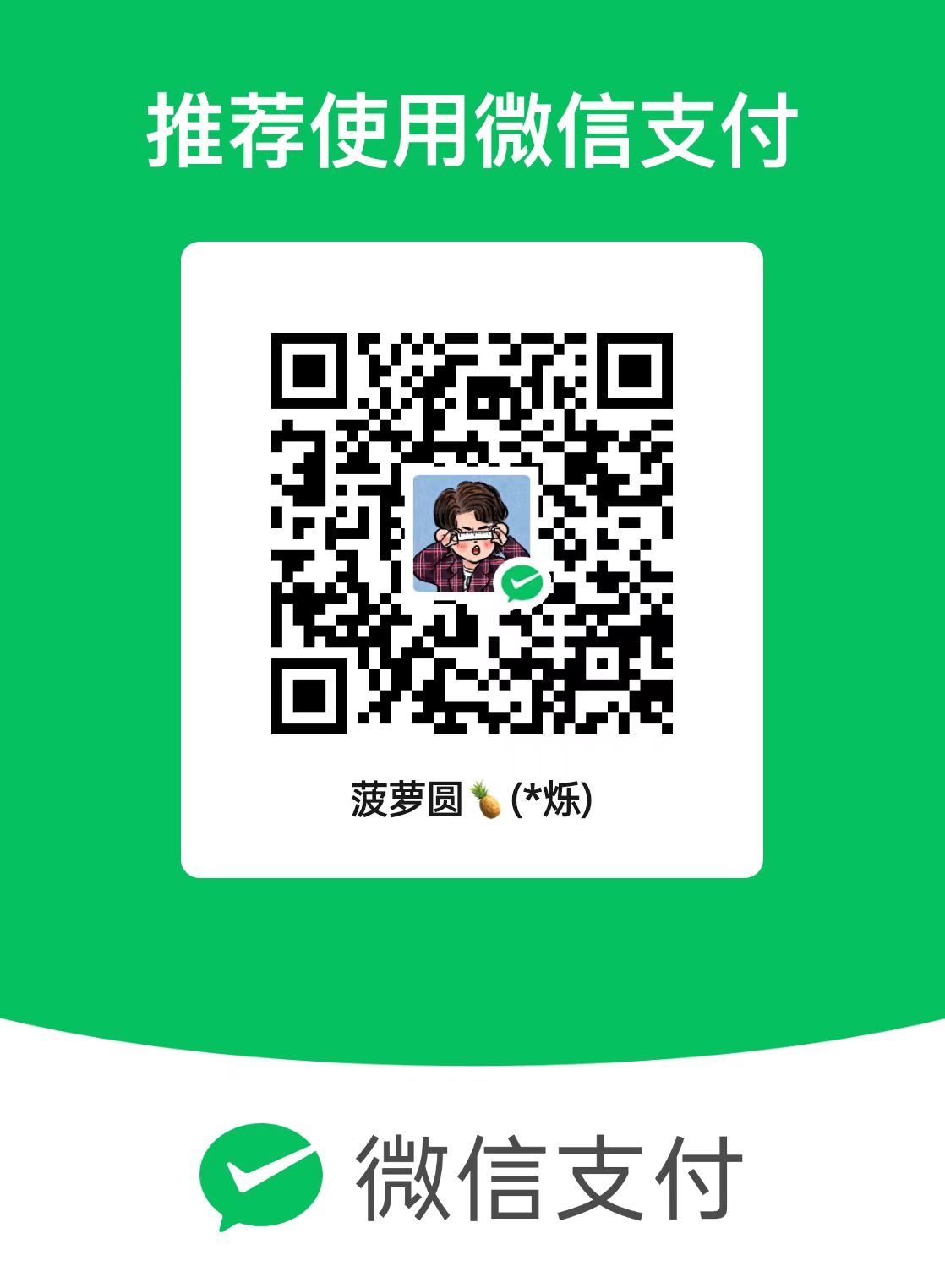
朋友们，写代码不易，请给一些赞赏支持喔~

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



目录

[一、简介 2](#_Toc20285)

[二、源码 3](#_Toc6426)

[三、常见问题及解决方案（出现问题先看这里） 28](#_Toc25829)

[四、（可选）按键功能指示灯示意图 29](#_Toc22800)

#### 一、简介

【数字电子时钟】

1、功能：时钟运行、时间调整、闹钟设置、日历设置；

2、交互：由四个独立按键控制，每个按键两种功能，分时复用；

3、显示：基于数码管动态显示。

【使用的芯片及开发板】

1、芯片：STC89C52RC（只要是**51单片机**都差不多）

2、开发板：**普中51-单核-A2**（若开发板不同引脚连接可能不同，需要自行 修改**引脚相关代码**。请认真看原理图喔。出现问题请先点击[常见问题](#_三、常见问题及解决方案（出现问题先看这里）)。）

【相较于初代时钟，此版改进了三个方面】

1、减少了按键控制的数量。从原先的9个按键（4个独立按键+5个矩阵按 键）变成了由4个独立按键控制。

2、增加了日历（可调日期）功能。

3、按键功能分配方式全新升级，更清晰、更合理，并可选搭配[外接LED](#_四、（可选）按键功能指示灯示意图)指 示灯来指示按键功能。

各位小伙伴请支持一下up叭~

写代码不易，码字也不易。

大家一起学习，一起进步！

#### 二、源码（为了方便，整个项目的代码都集中在一个.C文件中，将以下代码全部复制粘贴到KEIL中即可）

#include <reg51.h>

#include <intrins.h> //包含\_nop\_( )空函数指令的头文件

#define uchar unsigned char

#define uint unsigned int

#define BEEP\_PIN P1^5 //蜂鸣器引脚（若蜂鸣器不响，试试修改这里）

//\*\*\*\*\*\*\*\*\*\* 按键功能指示灯（可选）\*\*\*\*\*\*\*\*\*\*

sbit led0=P1^0;//需要外接LED灯

sbit led1=P1^1;//需要外接LED灯

bit led;//按键功能指示灯

//\*\*\*\*\*\*\*\*\*\* 数码管 \*\*\*\*\*\*\*\*\*\*

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

//共阴数码管 0 1 2 3 4 5 6 7 8 9 -

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

//共阴数码管 带小数点的0~9、-

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

//\*\*\*\*\*\*\*\*\*\* 定时器 \*\*\*\*\*\*\*\*\*\*

//晶振频率11.0592MHZ/s

#define T0time 46070

// 理论值为46080

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

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

uchar count=0; //计时1秒 T0需要循环的次数

//\*\*\*\*\*\*\*\*\*\* 蜂鸣器 \*\*\*\*\*\*\*\*\*\*

sbit BEEP=BEEP\_PIN;

//\*\*\*\*\*\*\*\*\*\* 全局标志 \*\*\*\*\*\*\*\*\*\*

//表明现在显示哪个界面。

uchar whereNow; //0:时钟界面；1:调时界面；2:闹钟界面；3:日历界面；

sbit K1=P3^1; //四个独立按键

sbit K2=P3^0;

sbit K3=P3^2;

sbit K4=P3^3;

//选中位置闪烁计数标志

#define shanCount 128

uchar blankCount;//空白显示 计数标志

uchar normalCount;//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

//\*\*\*\*\*\*\*\*\*\* 调时 \*\*\*\*\*\*\*\*\*\*

uchar clockSetLocation;//记录调时选中的位置 0,不闪烁,且无法使用加减按钮

//\*\*\*\*\*\*\*\*\*\* 闹钟 \*\*\*\*\*\*\*\*\*\*

uchar alarmSetLocation;//记录闹钟设置时的所选位置

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

bit alarmOnOff;//表明闹钟是否开启

bit alarmSetOrNot;//表明是否正在设置闹钟

//\*\*\*\*\*\*\*\*\*\* 日历 \*\*\*\*\*\*\*\*\*\*

uchar cldSetLocation;//记录闹钟设置时的所选位置

uint year=2023; //日历 年 初始数据

uchar month=11,day=11;//日历 月、日 初始数据

#define minYear 2020

#define maxYear 2050

//设定日历可显示的年份上下限

bit isLeapYear; //表明是否是闰年 1，是；0，否。

bit cldSetOrNot;//表明是否正在设置日历

uchar commonYearTable[]={0,31,28,31,30,31,30,31,31,30,31,30,31};//记录平年每个月有多少天

uchar leapYearTable[]={0,31,29,31,30,31,30,31,31,30,31,30,31}; //记录闰年每个月有多少天

//数组的第一个元素是用来凑index[0]的，这样在写程序的时候直接index[month]，方便些。

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

void delay() //短延时

{

uchar i;

for(i=0;i<128;i++);

}

void delay\_10ms(uchar c) //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 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 timer0\_initial()//--------------------------定时器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 is\_leap\_year()//--------------------------判断是否是闰年(我觉得做成函数比较直观)

{

if((year%4==0 && year%100!=0) || (year%400==0))

isLeapYear=1;

else

isLeapYear=0;

}

void serviceTimer0() interrupt 1//-----------------------中断服务函数

{

TH0=(65536-T0time)/256;//定时器重新赋值

TL0=(65536-T0time)%256;

count++;//定时器计数标志+1

if(20==count)//定时器次数达到20次，秒+1

{

miao++;

count=0;//定时器次数清零

}

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，日+1

{

day++;

shi=0;//时计数清零

}

if(0==isLeapYear)//如果是平年

{

if(day>commonYearTable[month])

{

day=1;//如果日计数超过本月天数，则变为1

month++;

}

}else //如果是闰年

{

if(day>leapYearTable[month])

{

day=1;//如果日计数超过本月天数，则变为1

month++;

}

}

if(month>12)

{

month=1;//如果月计数超过12，则变为1

year++;

}

if(year>maxYear)

{

year=minYear;//如果年计数超过一定限度，则变为最低限度年份

}

}

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(1==clockSetLocation&&blankCount>0)

{

smg(7,0x00);

blankCount--;

}

//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&&normalCount>0)

{

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

normalCount--;

}else if(1==clockSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}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(1==clockSetLocation&&blankCount>0)

{

smg(6,0x00);

blankCount--;

}

//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&&normalCount>0)

{

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

normalCount--;

}else if(1==clockSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}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(2==clockSetLocation&&blankCount>0)

{

smg(4,0x00);

blankCount--;

}

//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&&normalCount>0)

{

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

normalCount--;

}else if(2==clockSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}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(2==clockSetLocation&&blankCount>0)

{

smg(3,0x00);

blankCount--;

}

//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&&normalCount>0)

{

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

normalCount--;

}else if(2==clockSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}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(3==clockSetLocation&&blankCount>0)

{

smg(1,0x00);

blankCount--;

}

//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(3==clockSetLocation&&normalCount>0)

{

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

normalCount--;

}else if(3==clockSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}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(3==clockSetLocation&&blankCount>0)

{

smg(0,0x00);

blankCount--;

}

//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(3==clockSetLocation&&normalCount>0)

{

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

normalCount--;

}else if(3==clockSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}else

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

}

void clock\_set()//--------------------------调时(与时钟同界面)

{

if(0==K1 && 0==whereNow) //如果此时在时钟界面并按下了调时按键K1

{

delay\_10ms(3);//消抖

if(0==K1)//再次检测按键状态

{

while(0==K1);//长按等待

delay\_10ms(3);//消抖

whereNow=1;

clockSetLocation=1;

TR0=0;

}

}

if(1==whereNow) //进入调时界面，4个独立按键化身为调时相关按钮

{

if(0==K1)//如果按下K1:调整位置

{

delay\_10ms(3);//消抖

if(0==K1)//再次检测按键状态

{

while(0==K1);//长按停留在此

delay\_10ms(3);//消抖

clockSetLocation++;//位置标志+1

if(clockSetLocation>3)clockSetLocation=1;

}

}

if(1==clockSetLocation)//进行“时”加减

{

if(0==K2)//按下K2：数字-1

{

delay\_10ms(3);//消抖

if(0==K2)//再次检测按键状态

{

while(0==K2);//长按停留在此

delay\_10ms(3);//消抖

if(0==shi)shi=24;

shi--;

}

}

if(0==K3)//按下K3：数字+1

{

delay\_10ms(3);//消抖

if(0==K3)//再次检测按键状态

{

while(0==K3);//长按停留在此

delay\_10ms(3);//消抖

shi++;

if(24==shi)shi=0;

}

}

}

if(2==clockSetLocation)//进行“分”加减

{

if(0==K2)//按下K2：数字-1

{

delay\_10ms(3);//消抖

if(0==K2)//再次检测按键状态

{

while(0==K2);//长按停留在此

delay\_10ms(3);//消抖

if(0==fen)fen=60;

fen--;

}

}

if(0==K3)//按下K3：数字+1

{

delay\_10ms(3);//消抖

if(0==K3)//再次检测按键状态

{

while(0==K3);//长按停留在此

delay\_10ms(3);//消抖

fen++;

if(60==fen)fen=0;

}

}

}

if(3==clockSetLocation)//进行“秒”加减

{

if(0==K2)//按下K2：数字-1

{

delay\_10ms(3);//消抖

if(0==K2)//再次检测按键状态

{

while(0==K2);//长按停留在此

delay\_10ms(3);//消抖

if(0==miao)miao=60;

miao--;

}

}

if(0==K3)//按下K3：数字+1

{

delay\_10ms(3);//消抖

if(0==K3)//再次检测按键状态

{

while(0==K3);//长按停留在此

delay\_10ms(3);//消抖

miao++;

if(60==miao)miao=0;

}

}

}

if(0==K4)//如果按下K4：确认并退出调时模式，然后返回时钟界面

{

delay\_10ms(3);//消抖

if(0==K4)//再次检测按键状态

{

while(0==K4);//长按停留在此

delay\_10ms(3);//消抖

clockSetLocation=0;

whereNow=0;

TR0=1;

}

}

}

}

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

{

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

//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(1==alarmSetLocation&&blankCount>0)

{

smg(5,0x00);

blankCount--;

}

//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&&normalCount>0)

{

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

normalCount--;

}else if(1==alarmSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}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(1==alarmSetLocation&&blankCount>0)

{

smg(4,0x00);

blankCount--;

}

//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&&normalCount>0)

{

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

normalCount--;

}else if(1==alarmSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}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(2==alarmSetLocation&&blankCount>0)

{

smg(3,0x00);

blankCount--;

}

//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&&normalCount>0)

{

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

normalCount--;

}else if(2==alarmSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}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(2==alarmSetLocation&&blankCount>0)

{

smg(2,0x00);

blankCount--;

}

//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&&normalCount>0)

{

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

normalCount--;

}else if(2==alarmSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}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(3==alarmSetLocation&&blankCount>0)

{

smg(1,0x00);

blankCount--;

}

//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(3==alarmSetLocation&&normalCount>0)

{

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

normalCount--;

}else if(3==alarmSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}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(3==alarmSetLocation&&blankCount>0)

{

smg(0,0x00);

blankCount--;

}

//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(3==alarmSetLocation&&normalCount>0)

{

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

normalCount--;

}else if(3==alarmSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}else

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

}

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

{

if(0==whereNow && 0==K2)//如果此时在时钟界面并按下了闹钟按键K2

{

delay\_10ms(3);//消抖

if(0==K2)//再次检测按键状态

{

while(0==K2);//长按等待

delay\_10ms(3);//消抖

whereNow=2;

}

}

if(2==whereNow)//如果此时在闹钟界面，那么四个独立按键化身为闹钟相关按键

{

if(0==K1)//如果按下K1:调整位置

{

delay\_10ms(3);//消抖

if(0==K1)//再次检测按键状态

{

while(0==K1);//长按停留在此

delay\_10ms(3);//消抖

alarmSetLocation++;//位置标志+1

if(4==alarmSetLocation)alarmOnOff=!alarmOnOff;

if(alarmSetLocation>3)alarmSetLocation=0;

}

}

if(1==alarmSetLocation)//进行“时”加减

{

if(0==K2)//按下K2：数字-1

{

delay\_10ms(3);//消抖

if(0==K2)//再次检测按键状态

{

while(0==K2);//长按停留在此

delay\_10ms(3);//消抖

if(0==shi1)shi1=24;

shi1--;

}

}

if(0==K3)//按下K3：数字+1

{

delay\_10ms(3);//消抖

if(0==K3)//再次检测按键状态

{

while(0==K3);//长按停留在此

delay\_10ms(3);//消抖

shi1++;

if(24==shi1)shi1=0;

}

}

}

if(2==alarmSetLocation)//进行“分”加减

{

if(0==K2)//按下K2：数字-1

{

delay\_10ms(3);//消抖

if(0==K2)//再次检测按键状态

{

while(0==K2);//长按停留在此

delay\_10ms(3);//消抖

if(0==fen1)fen1=60;

fen1--;

}

}

if(0==K3)//按下K3：数字+1

{

delay\_10ms(3);//消抖

if(0==K3)//再次检测按键状态

{

while(0==K3);//长按停留在此

delay\_10ms(3);//消抖

fen1++;

if(60==fen1)fen1=0;

}

}

}

if(3==alarmSetLocation)//进行“秒”加减

{

if(0==K2)//按下K2：数字-1

{

delay\_10ms(3);//消抖

if(0==K2)//再次检测按键状态

{

while(0==K2);//长按停留在此

delay\_10ms(3);//消抖

if(0==miao1)miao1=60;

miao1--;

}

}

if(0==K3)//按下K3：数字+1

{

delay\_10ms(3);//消抖

if(0==K3)//再次检测按键状态

{

while(0==K3);//长按停留在此

delay\_10ms(3);//消抖

miao1++;

if(60==miao1)miao1=0;

}

}

}

if(0==K4)//如果按下K4：确认并退出闹钟界面，然后返回时钟界面

{

delay\_10ms(3);//消抖

if(0==K4)//再次检测按键状态

{

while(0==K4);//长按停留在此

delay\_10ms(3);//消抖

alarmSetLocation=0;

whereNow=0;

}

}

}

}

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

//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、当前在时钟运行界面

//

//注：按下独立按键中的任意一个并松开，可停止闹钟

{

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

{

while(1==K1 && 1==K2 && 1==K3 && 1==K4)

{

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==K1 || 0==K2 || 0==K3 || 0==K4);//等待，直至松开独立按键

delay\_10ms(3);

}

}

void cld\_display()//------------------------数码管动态显示日历界面

{

//年的千位

//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==cldSetLocation&&blankCount>0)

{

smg(7,0x00);

blankCount--;

}

//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==cldSetLocation&&normalCount>0)

{

smg(7,table[year/1000]);

normalCount--;

}else if(1==cldSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}else

smg(7,table[year/1000]);

//年的百位

//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==cldSetLocation&&blankCount>0)

{

smg(6,0x00);

blankCount--;

}

//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==cldSetLocation&&normalCount>0)

{

smg(6,table[year/100%10]);

normalCount--;

}else if(1==cldSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}else

smg(6,table[year/100%10]);

//年的十位

//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==cldSetLocation&&blankCount>0)

{

smg(5,0x00);

blankCount--;

}

//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==cldSetLocation&&normalCount>0)

{

smg(5,table[year%100/10]);

normalCount--;

}else if(1==cldSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}else

smg(5,table[year%100/10]);

//年的个位

//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==cldSetLocation&&blankCount>0)

{

smg(4,0x00);

blankCount--;

}

//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==cldSetLocation&&normalCount>0)

{

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

normalCount--;

}else if(1==cldSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}else

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

//月的十位

//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==cldSetLocation&&blankCount>0)

{

smg(3,0x00);

blankCount--;

}

//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==cldSetLocation&&normalCount>0)

{

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

normalCount--;

}else if(2==cldSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}else

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

//月的个位

//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==cldSetLocation&&blankCount>0)

{

smg(2,0x00);

blankCount--;

}

//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==cldSetLocation&&normalCount>0)

{

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

normalCount--;

}else if(2==cldSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}else

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

//日的十位

//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(3==cldSetLocation&&blankCount>0)

{

smg(1,0x00);

blankCount--;

}

//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(3==cldSetLocation&&normalCount>0)

{

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

normalCount--;

}else if(3==cldSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}else

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

//日的个位

//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(3==cldSetLocation&&blankCount>0)

{

smg(0,0x00);

blankCount--;

}

//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(3==cldSetLocation&&normalCount>0)

{

smg(0,table\_dot[day%10]);

normalCount--;

}else if(3==cldSetLocation)

{

blankCount=shanCount;//闪烁标志初始化

normalCount=shanCount;

}else

smg(0,table\_dot[day%10]);

}

void cld\_set()//---------------------------日历设置

{

if(0==whereNow && 0==K3)//如果此时在时钟界面并按下了日历按键K3

{

delay\_10ms(3);//消抖

if(0==K3)//再次检测按键状态

{

while(0==K3);//长按等待

delay\_10ms(3);//消抖

whereNow=3;

}

}

if(3==whereNow)//如果此时在日历界面，那么四个独立按键化身为日历相关按键

{

if(0==K1)//如果按下K1:调整位置

{

delay\_10ms(3);//消抖

if(0==K1)//再次检测按键状态

{

while(0==K1);//长按停留在此

delay\_10ms(3);//消抖

cldSetLocation++;//位置标志+1

if(cldSetLocation>3)cldSetLocation=1;

}

}

if(1==cldSetLocation)//进行“年”加减

{

if(0==K2)//按下K2：数字-1

{

delay\_10ms(3);//消抖

if(0==K2)//再次检测按键状态

{

while(0==K2);//长按停留在此

delay\_10ms(3);//消抖

if(minYear==year)year=minYear+1;//设定日历年份减到最低限度则无法再减

year--;

}

}

if(0==K3)//按下K3：数字+1

{

delay\_10ms(3);//消抖

if(0==K3)//再次检测按键状态

{

while(0==K3);//长按停留在此

delay\_10ms(3);//消抖

year++;

if(maxYear+1==year)year=maxYear;//设定日历年份加到最大限度则无法再加

}

}

}

if(2==cldSetLocation)//进行“月”加减

{

if(0==K2)//按下K2：数字-1

{

delay\_10ms(3);//消抖

if(0==K2)//再次检测按键状态

{

while(0==K2);//长按停留在此

delay\_10ms(3);//消抖

if(1==month)month=13;

month--;

}

}

if(0==K3)//按下K3：数字+1

{

delay\_10ms(3);//消抖

if(0==K3)//再次检测按键状态

{

while(0==K3);//长按停留在此

delay\_10ms(3);//消抖

month++;

if(13==month)month=1;

}

}

}

if(3==cldSetLocation)//进行“日”加减

{

if(0==K2)//按下K2：数字-1

{

delay\_10ms(3);//消抖

if(0==K2)//再次检测按键状态

{

while(0==K2);//长按停留在此

delay\_10ms(3);//消抖

if(day<=1)

{

if(0==isLeapYear)//如果是平年

day=commonYearTable[month]+1;

if(1==isLeapYear)//如果是闰年

day=leapYearTable[month]+1;

}

day--;

}

}

if(0==K3)//按下K3：数字+1

{

delay\_10ms(3);//消抖

if(0==K3)//再次检测按键状态

{

while(0==K3);//长按停留在此

delay\_10ms(3);//消抖

day++;

if(0==isLeapYear)//如果是平年

{

if(day>commonYearTable[month])day=1;

}

if(1==isLeapYear)//如果是闰年

{

if(day>leapYearTable[month])day=1;

}

}

}

}

if(0==K4)//如果按下K4：确认并退出日历界面，然后返回时钟界面

{

delay\_10ms(3);//消抖

if(0==K4)//再次检测按键状态

{

while(0==K4);//长按停留在此

delay\_10ms(3);//消抖

cldSetLocation=0;

whereNow=0;

}

}

}

}

//void led\_display()//-----------------按键功能指示灯（可选）

//{

// if(0==whereNow)

// {

// led0=1;

// led1=0;

// }

// else

// {

// led0=0;

// led1=1;

// }

//}

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

{

uint t=1000;

while(t--)//20221111（完成此作品的日期）

{

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[1]);

smg(0,table[1]);

}

t=1000;//t重置

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.

}

}

void main()

{

start\_show(); //开机展示（可选）

clockSetLocation=0; //

alarmSetOrNot=0; //

alarmSetLocation=0; //

cldSetOrNot=0; //

cldSetLocation=0; //各界面位置标志与是否调整标志初始化为0

alarmOnOff=0; //闹钟默认关闭

whereNow=0; //默认最开始在时钟界面

blankCount=shanCount; //

normalCount=shanCount;//闪烁计数标志初始化

timer0\_initial(); //定时器T0初始化

led=led0;

while(1)

{

is\_leap\_year(); //判断是否是闰年

if(0==whereNow || 1==whereNow)clock\_display();//数码管动态显示时钟界面

clock\_set(); //调时(与时钟同界面)

if(2==whereNow)alarm\_display(); //数码管动态显示闹钟界面

alarm\_set(); //闹钟设置

alarm\_exe(); //闹钟执行

if(3==whereNow)cld\_display();//数码管动态显示日历界面

cld\_set(); //日历设置

//led\_display(); //按键功能指示灯（可选）

}

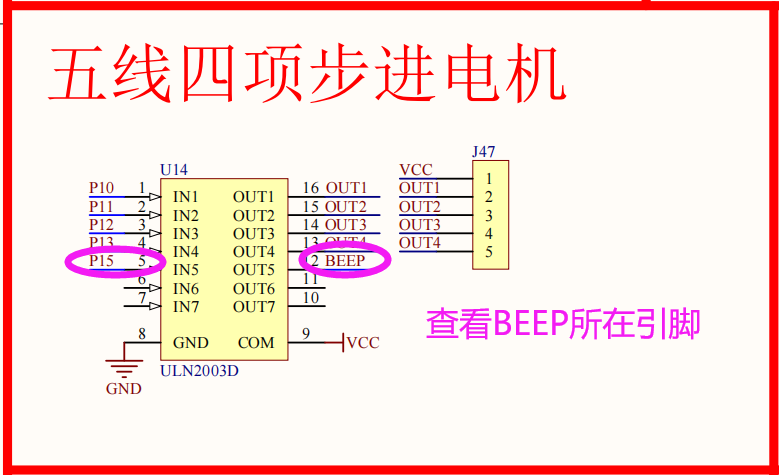
}

#### 三、常见问题及解决方案（出现问题先看这里）

1、问题：闹钟该响的时候不响，不该响的时候响。（蜂鸣器相关问题）

解决方案：请查看您的开发板原理图 -> 找到“五线四项步进电机”模块 -> 查看“BEEP”所在引脚是否为P15（源代码写的是P1^5）

很多小伙伴的开发板虽然也是普中51A2，但BEEP所在引脚是P2^5。如果是这样，您可以将源代码中对蜂鸣器引脚的定义（在代码最前面几行）改成P2^5



2、问题：源代码中有“按键功能指示灯”（void led\_display()函数），这是什么？

up主解答：这就是一个方便使用者的辅助灯，可以指示独立按键目前对应什么功能，是可选的。如果您不需要这些花里胡哨的，完全可以不管它，不用修改任何代码（或者直接把led\_display函数注释掉）。如果您需要，请看后面的[示意图](#_四、（可选）按键功能指示灯示意图)。

3、问题：数码管出现乱码怎么办？

解决方案：请您看看是否把LCD插到开发板上了。（因为引脚冲突，LCD和数码管不能同时使用）

#### 四、（可选）按键功能指示灯示意图

指示灯功能表及外接LED安放位置示意图：

(电烙铁在学校没有带回家，所以做了这么一个简陋的外接led，用电工胶带接触不良，请小伙伴们还是用电烙铁焊接噢)

