

设计题目：

基于 51 单片机的电子时钟设计

摘 要

单片机,是集 CPU ,RAM ,ROM ,定时器,计数器和多种接口于一体的微控制器。自 20 世纪 70 年代问世以来,以其极高的性能价格比,受到人们的重视和关注。它体积小、重量轻、抗干扰能力强、环境要求不高、价格低廉、可靠性高、灵活性好、开发较为容易,广泛应用于智能生产和工业自动化上。

本系统为基于DS12C887 的多功能电子钟,以STC89C51单片机作为主控芯,采用实时时钟芯片DS12C887,使用 1602 液晶作为显示输出。该系统走时精确,具有闹钟设置,时间模式切换,秒表以及可同时显示时间、日期等多种功能。本文将详细介绍STC89C51 单片机和DS12C887 时钟芯片的基本原理,从软件和硬件电路的实现两大方面进行分析

【关键词】

STC89C51 单片机 液晶屏 时钟芯片 蜂鸣器

目 录

前 言	1
一、方案选型:	2
二、系统硬件设计	3
2.1 51 单片机最小系统设计	3
2.2 电源供电电路设计	3
2.3 串口通信电路设计	4
2.4 时钟芯片电路设计	4
2.5 LCD显示电路设计	6
2.6 报警电路设计	6
2.7 键控电路设计	6
三、系统软件设计	7
3.1 系统程序流程图设计	7
3.2 系统程序设计 (见附录)	9
四、总结	9
4.1 作品功能、特色	9
4.2 综合设计的体会	9
参考文献	11
附录	12

前 言

随着科学技术的不断发展，人们对时间计量的要求越来越高。在当今社会，电子时钟已经得到相当广泛的应用，产品多样，发展更是多元化。本作品是以STC89C51单片机作为主控芯片，使用12MHZ的晶振，使用专用时钟日历芯片DS12C887产生时间信息，时间精确。软件部分以C语言为主体，用1602LCD液晶屏显示输出信息，输出信息量多，更直观、人性化。该时钟可实现人机交互，可通过提供的键盘对其进行调整。系统具有以下功能：年、月、日、时、分、秒显示；12小时/24小时模式切换，在12小时模式中，用AM和PM区分上午和下午；秒表功能；整点闹铃和报时功能，且闹钟可设置多组。本次设计的电子时钟系统由单片机最小系统，1602LCD液晶屏，时钟芯片，调整按键，蜂鸣器，电源五大部分组成。

一、方案选型：

我们在设计电子时钟时遇到了芯片选型的问题，以下是三个设计方案：

方案一：DS1302+数码管

DS1302 的使用非常方便，而且价格便宜而数码管显示的也很清楚，特别是显示时间很直观。但在制作过程中我们发现了这方案的一些问题。DS1302 是不自带电池，虽然可以通过外接纽扣电池来达到断电走时继续的目的，但在实际调试中会发现这是比较困难的。因为 DS1302 上电需要复位，而复位就会把正确的走时清零。如果不复位，DS1302 会出现各种各样的问题，如不走时、读出乱码等。要解决这个问题需要增加如 2402 等存储器，上电后先存储时间值，再复位。这么做无疑增加了电路设计和软件设计的复杂度。而使用数码管显示，虽然价格也便宜，显示效果好，但多位的数码管在动态扫描的时候会出现闪烁。如果少用几位，用切换的方法查看日期，时间等信息又显得麻烦。

方案二：DS12C887+1602LCD 液晶屏

DS12C887 时钟芯片功能丰富价格适中，能够自动产生世纪、年、月、日、时、分、秒以及时间模式转换等的时间信息，芯片内部增加了世纪存储器，从而利用硬件电路解决了“千年”的问题。DS12C887 时钟芯片中还自带有锂电池，单片机掉电后时钟芯片内部的时间信息可以保持十年之久。1602LCD 液晶屏可以输出 2 行，每行显示 16 个字符。虽然 1602LCD 液晶屏较昂贵，但是该液晶屏显示清晰且不会闪烁，由于液晶屏是数字式的，因此和单片机系统的接口简单，操作方便。1602LCD 液晶屏的功耗主要消耗在其内部的电极和驱动 IC 上，因而耗电量比其它显示器要少得多，功耗较低。以上两种元件的程序编写简单，适用于多方面的应用。

方案三：SD2068+1602LCD 液晶屏

SD2068 实时时钟芯片功能更加丰富，它除了具备有 DS12C887 时钟芯片的功能，另外还内置时钟精度数字调整功能，可以在很宽的范围内校正时钟的偏差；内置上电复位电路及指示位；内置电源稳压，内部计时电压可低至 1.5V。该芯片为工业级产品，是在选用实时时钟 IC 时的理想选择。但是该芯片是一款新型的芯片，使用并不广泛，操作不方便，可能会出现芯片不稳定等的因素。

因此由以上三种方案进行比较，我们选择方案二来设计电子时钟。以 STC89C51

为主控芯片，DS12C887 为时钟芯片，1602LCD 液晶屏作为显示器。程序控制 DS12C887 时钟芯片实现小时,分,秒和年,月,日的计时，并在 1602LCD 液晶屏上显示出来。通过按键对 12 小时/24 小时显示模式切换。当时间走到程序所设定的时间时，蜂鸣器响起，起到闹铃功能。当要显示秒表计时时，可以通过按键切换来实现。

二、系统硬件设计

2.1 51 单片机最小系统设计

单片机最小系统如下图 1-1 所示：

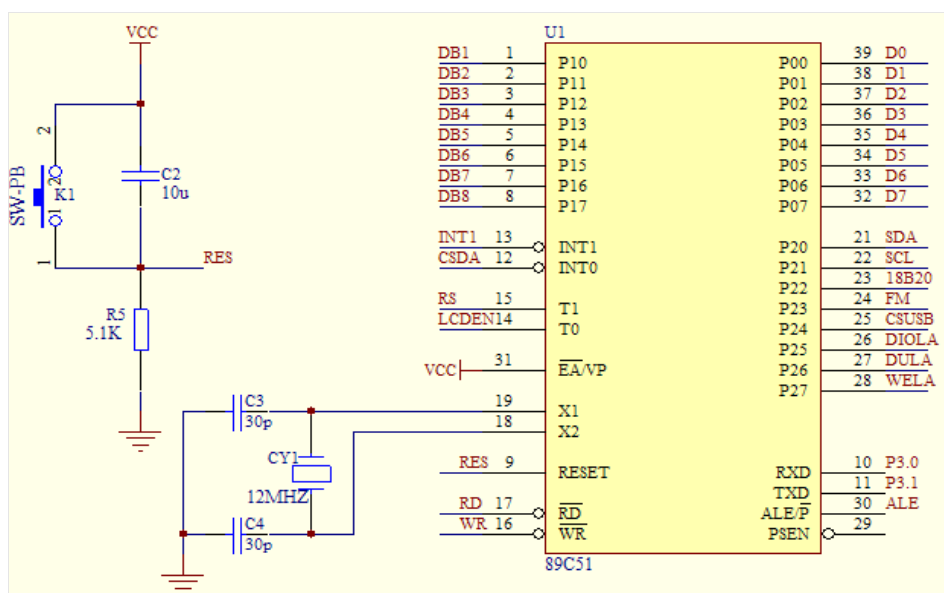


图 1-1

以 STC89C51 单片机为核心，选用 12MHZ 的晶振，由于晶振的频率越高，单片机的运行速度就越快，但考虑到单片机的运行速度快会导致对存储器的要求就会变高，因此 12MHZ 晶振为最佳选择。外接电容的值虽然没有严格的要求，但是外接电容的大小会影响振荡器的频率高低、振荡器的稳定性和起振的快速性，因此我们选用 30pF 的电容作为起振电容。复位电路为按键高电平复位，当按键按下，RES 端为高电平，当高电平持续 4us 的时间就可以使单片机复位。

2.2 电源供电电路设计

电源供电电路如下图 1-2 所示：

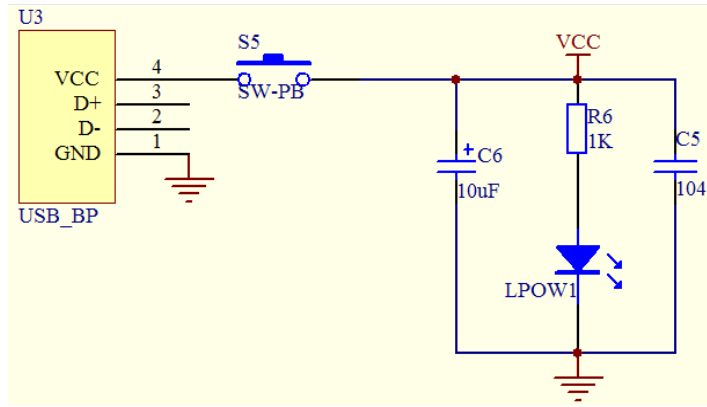


图 1-2

我们采用外接 USB 端口的方式为单片机供电，LPOW1 为电源显示灯，当按键 S5 按下，显示灯亮，表示给单片机供+5V 电压。

2.3 串口通信电路设计

串口通信电路如下图 1-3 所示：

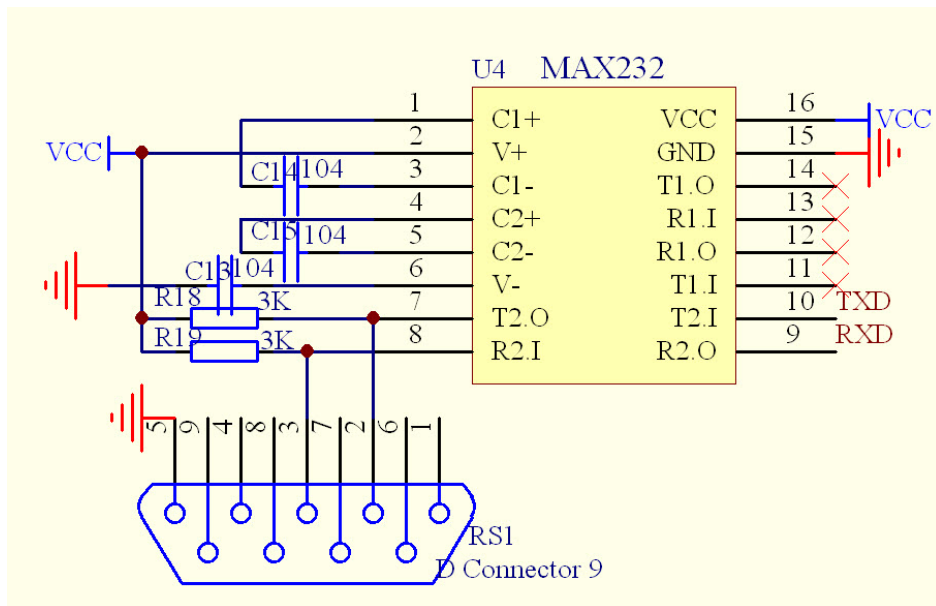


图 1-3 串口通信电路图

图中通过 MAX232 进行 TTL 电平和 232 电平转换，从而单片机和上位机之间通信提供通道。通信电路的目的就是让通信双发的电平匹配，单片机用的是 TTL 电平，上位机的串口用的是 232 电平。TTL 电平的逻辑 1 的电压范围是+3.3V 到+5V，逻辑 0 的电压范围是 0 到+3.3V；232 电平的逻辑 1 的电压范围是+5V 到+15V，逻辑 0 的电压范围是-5V 到-15V。因此设计串口通信电路就是让这两种电平统一。

2.4 时钟芯片电路设计

时钟芯片电路如下图 1-4 所示：

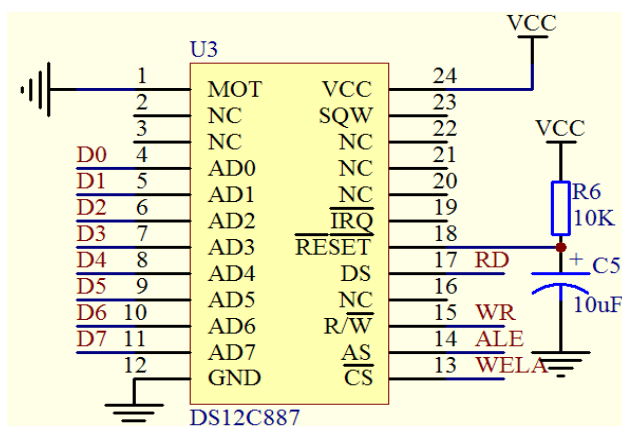


图 1-4

我们采用 DS12C887 时钟芯片定时及计时功能，DS12C887 时钟芯片共需要 13 条信号线。GND、VCC：直流电源，其中 VCC 接+5V 输入，GND 接地，当 VCC 输入为+5V 时，用户可以访问 DS12C887 内 RAM 中的数据，并可对其进行读、写操作；当 VCC 的输入小于+4.25V 时，禁止用户对内部 RAM 进行读、写操作，此时用户不能正确获取芯片内的时间信息；当 VCC 的输入小于+3V 时，DS12C887 会自动将电源发换到内部自带的锂电池上，以保证内部的电路能够正常工作。MOT：模式选择脚。SQW：方波输出脚，当供电电压 VCC 大于 4.25V 时，SQW 脚可进行方波输出。AD0~AD7：复用地址数据总线，该总线采用分时复用技术，在总线周期的前半部分，出现在 AD0~AD7 上的是地址信息，可用以选通 DS12C887 内的 RAM，总线周期的后半部分出现在 AD0~AD7 上的数据信息。AS：地址选通输入脚。DS/RD：数据选择或读输入脚，该引脚有两种工作模式，当 MOT 接 VCC 时，选用 Motorola 工作模式，在这种工作模式中，每个总线周期的后一部分的 DS 为高电平。在读操作中，DS 的上升沿使 DS12C887 将内部数据送往总线 AD0~AD7 上，以供外部读取。在写操作中，DS 的下降沿将使总线 AD0~AD7 上的数据锁存在 DS12C887 中。当 MOT 接 GND 时，选用 Intel 工作模式，在该模式中，该引脚是读允许输入脚。R/W：读/写输入端，该管脚也有 2 种工作模式，当 MOT 接 VCC 时，R/W 工作在 Motorola 模式。CS：片选输入，低电平有效。IRQ：中断请求输入，低电平有效，该脚有效对 DS12C887 内的时钟、日历和 RAM 中的内容没有任何影响，仅内部的控制寄存器有影响，在典型的应用中，RESET 可以直接接 VCC，这样可以保证 DS12C887 在掉电时，其内部控制寄存器不受影响。

2.5 LCD显示电路设计

1602LCD 液晶屏显示电路如下图 1-5 所示：

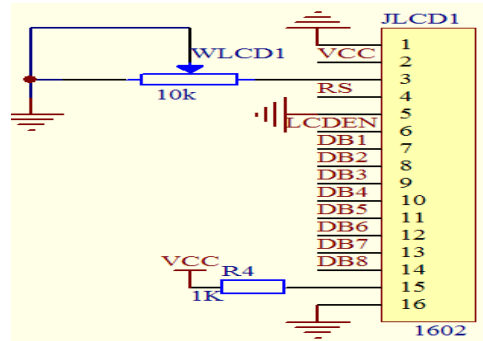


图 1-5

1602LCD 液晶屏为 5V 电压驱动，带背光，可显示两行，每行 16 个字符，不能显示汉字。液晶 1, 2 端为电源；15, 16 为背光电源；为防止直接加 5V 而烧坏背光灯，在 15 脚串联一个 1k 电阻用于限流。液晶 3 端为液晶对比度调节端，通过一个 10k 的变位器来调节液晶显示对比度。液晶 4 端为向液晶控制器写数据/写命令选择端，接单片机的 P3.5 口。液晶 5 端为读/写选择端，因为我们不从液晶中读取数据，只向其写入命令和显示数据，因此此端始终选择为写状态，即低电平接地。液晶 6 端为使能信号，是操作时必须的信号，接单片机的 P3.4 口。

2.6 报警电路设计

蜂鸣器闹铃电路如下图 1-6 所示

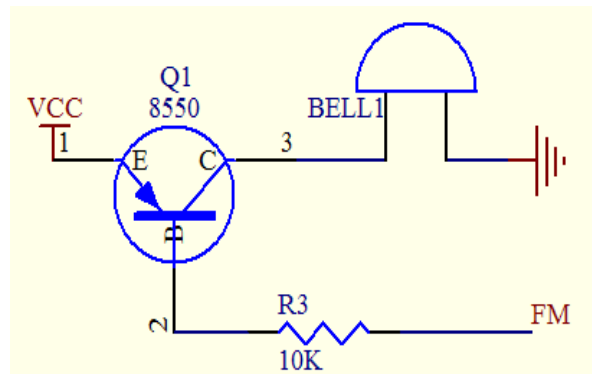


图 1-6

蜂鸣器电路接在单片机的 P2.3 引脚上，当给该引脚一个低电平，三极管导通，蜂鸣器发出声音作为闹铃。

2.7 键控电路设计

按键调整电路如下图 1-7 所示：

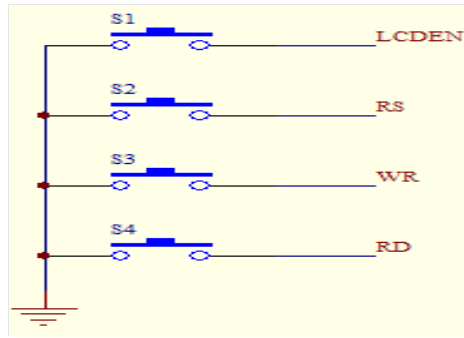


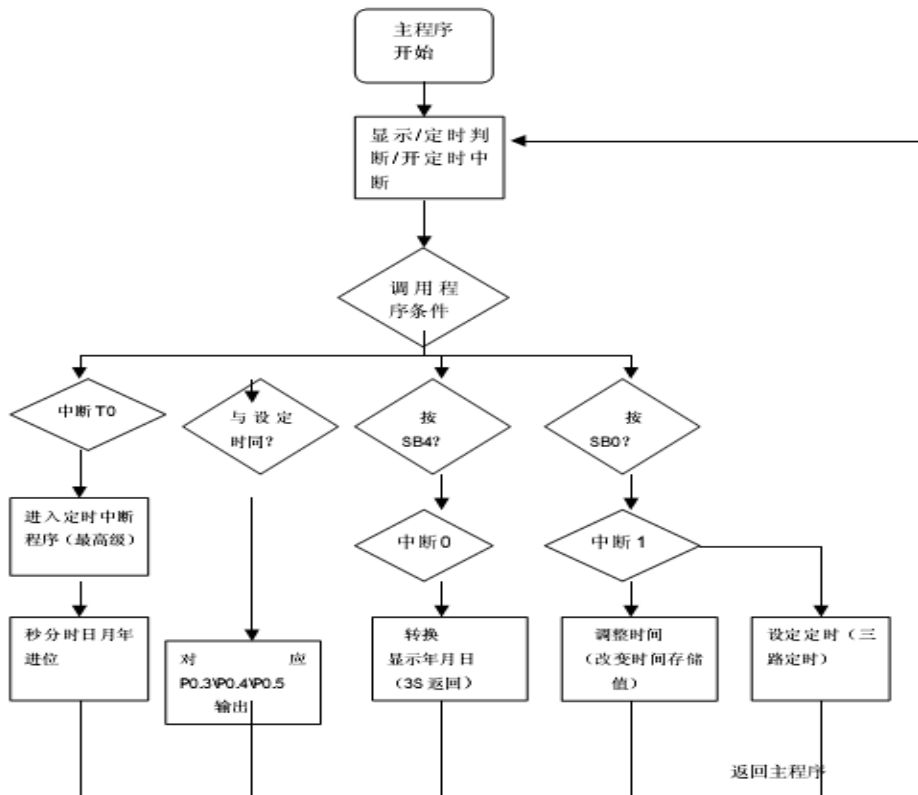
图 1-7

四个独立键盘均采用查询方式，我们将按键的一端接地，另一端各接一根输入线直接与 STC89C51 的 I/O 口相连，当按键闭合时，相当于该 I/O 口通过按键与地相连，变成低电平，单片机通过检测 I/O 口的电平状态，即可以识别出按下的键。通过四个按键实现各方式的切换，S2 用于 12 小时/24 小时的模式切换，S3 用于秒表功能与时钟功能间的切换，S4 用于实现参数设置和调节功能。

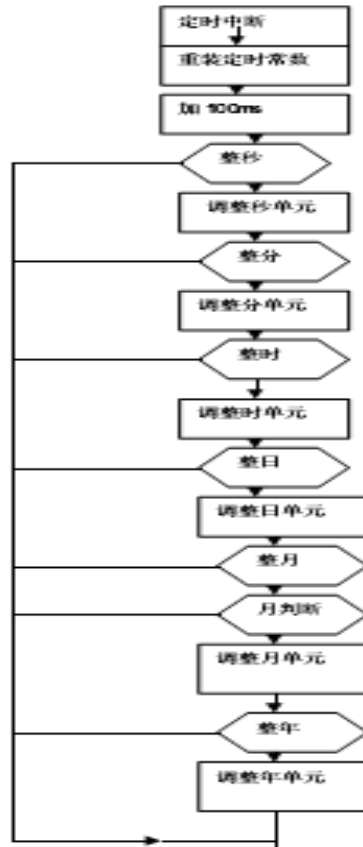
三、系统软件设计

3.1 系统程序流程图设计

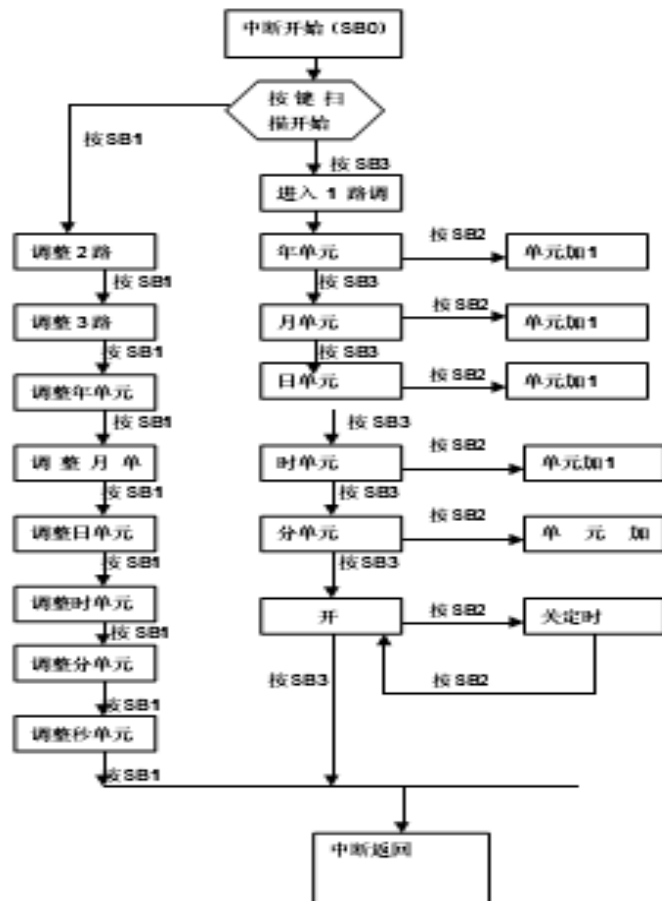
流程图 1：实验主程序流程图



流程图 2：定时中断程序流程图



流程图 3：调时功能流程图



3.2 系统程序设计 （见附录）

四、总结

4.1 作品功能、特色

本系统实现了人机交互功能，有夏令时制以及万年历功能。可对年、月、日、时、分、秒进行任意设定，多组闹钟设定任你行，具有整点报时功能，模式切换随你意，秒表功能同时有。

特色：本系统采用 1602LCD 显示信息，相比其他电子时钟，其显示信息更丰富，可同时显示时间，日期，星期等。秒表功能精确到毫秒级，比普通具有秒表功能的时钟要高一个数量级。闹钟可设置多组，解决了普通电子钟只能设置一组给大家带来的烦恼。系统具有断电保护功能，芯片可以保存掉电时候的信息并继续工作，其睡眠工作时间可长达 12 小时

4.2 综合设计的体会

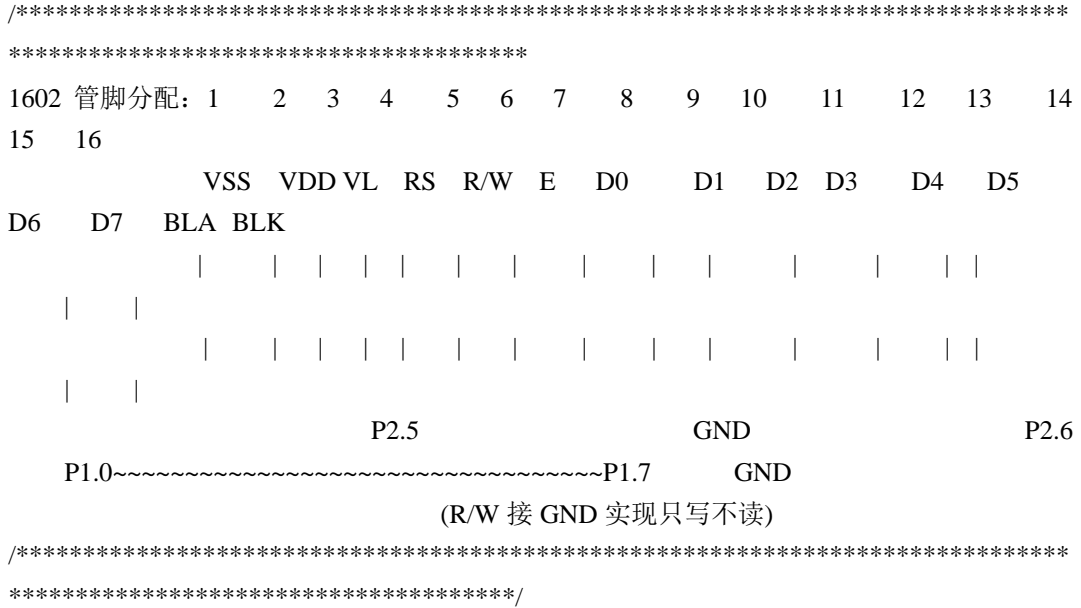
本次大赛很好的锻炼了我们的动手能力以及团队合作能力，让我们享受到电子设计的快乐。让我们开阔了视野，同时也增长了见识，在课余时间发挥想象和积极思考，锻炼了自己的思维能力。在元件的选择和电路的设计上我们大胆创新，敢于尝试，并且积极搜查资料，认真分析，选择出最优的电路组合，使整个系统能够按照设计要求

稳定运行。

参考文献

- [1] 李广弟等编 单片机基础北京航空航天大学出版社 2001
- [2] 吴金戌等编 8051 单片机实践与应用清华大学出版社 2001
- [3] 王洪庆编 微型计算机控制技术机械工业出版社 2006
- [4] 付家才编 单片机控制工程实践技术化学工业出版社 2004
- [5] 鲍宏亚等编 MCS-51 系列单片机应用系统设计及实用技术中国宇航出版社
2005
- [6] 杨文龙.单片机原理及应用[M].西安电子科技大学出版社, 2000
- [7] 赵文博, 刘文涛.单片机语言 C51 程序设计[M].人民邮电出版社, 2006

附录



```

#include<reg52.h>

#define uchar unsigned char
#define uint unsigned int

sbit lcd_en=P2^6;           //LCD1602 使能信号 E 位定义
sbit lcd_rs=P2^5;         //LCD1602 读/写 (H/L) 控制信号 RS 位定义

uchar ii,ll;             //在 1602 上写字符串所用的变量

sbit beep=P2^3;          //蜂鸣器实现闹钟报警
sbit dscs=P2^7;
sbit dsas=P2^4;
sbit dsrw=P3^6;
sbit dsds=P3^7;
sbit dsirq=P3^2;        //闹钟申请

//按键部分
sbit s1=P2^0;
sbit s2=P2^1;

```

```

sbit s3=P2^2;
sbit s4=P3^0;

uchar code table4[]=" 20  -  -   Mon";
uchar code table5[]="      :  :  ";

uchar count,s3_num,s4_num,flag,flag1;
uchar miao,shi,fen,nian,yue,ri,xinqi,n_miao,n_shi,n_fen;

//显示字符串
uchar code table1[]=" 2009-07-13  Mon";
uchar code table2[]="      00:00:00";
uchar code table3[]=" Alarm Clock      ";//闹钟字符串

//void display()
//{
//wrt
//}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
uchar code xq1[]="Mon";    //星期一的字符串
uchar code xq2[]="Tue";   //星期二的字符串
uchar code xq3[]="Wed";   //星期三的字符串
uchar code xq4[]="Thu";   //星期四的字符串
uchar code xq5[]="Fri";   //星期五的字符串
uchar code xq6[]="Sat";   //星期六的字符串
uchar code xq7[]="Sun";   //星期天的字符串

//函数申明
void write_ds(uchar,uchar);           //INTEL 总线写时序
void set_alarm();    //设置闹钟
uchar read_ds(uchar);                 //intel 总线读时序
void set_time();                       //设置时间
void write_nyr(uchar add,uchar date);  //在 LCD1602 上刷新年月日
void didi();

```

```

//1602 显示操作程序
void LCD_1602_str(uchar hang1,char *ttt1); //每次调用都会清屏，在液晶屏上面显示字符串
void LCD_1602_t(uchar hang1,char *ttt1); //在上次显示内容的基础上替换特定位置的字符串
void LCD_1602_num(uchar hang1,uchar nnn1); //在特定的位置替换为一位数字（必须是0~9的数字，否则显示的将不是想要显示的数字）

void delay(uint k); //延迟子函数

void write_com(uchar com); //1602 写指令
void write_data(uchar data0); //1602 写数据
void init00(); //1602 初始化子程序

void int0(); //定时器初始化函数
char k[]=" ."; //k[]="____."
char j[]="s"; //j[]="s"
uchar sta; //temp 记录产生的1ms基准中断次数,sta 存放系统状态（0时为暂停，1时为启动）
uint temp,ms,s,s_100; //毫秒变量ms，秒变量s，百秒变量s_100
uint ms_bai,ms_shi,ms_ge,s_shi,s_ge,s_100_shi,s_100_ge;

sbit stop = P2^1; //停表或者启动 按键s2 实现停止秒表计数
sbit rest = P2^0; //清零 按键s1 实现清零秒表计数

void miao1(void);

void key_s4_scan();

////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////设置闹钟//////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////

void set_alarm()
{
    if(n_shi==read_ds(4)&& n_fen==read_ds(2)&& n_miao==read_ds(0))

        didi();
}

```



```

void didi()
{
    beep=0;
    delay(500);
    beep=1;
    delay(1000);
    beep=0;
    delay(500);
    beep=1;
    beep=0;
    delay(500);
    beep=1;
    delay(1000);
    beep=0;
    delay(500);
    beep=1;
    beep=0;
    delay(500);
    beep=1;
    delay(1000);
    beep=0;
    delay(500);
    beep=1;
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////1602 显示子函数/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

void LCD_1602_str(uchar hang1,char *ttt1)
{
    init00(); //1602 初始化子函数
    for(ll=0,ii=0;ttt1[ii];ii++,ll++);//“求第一行字符串长度”,当 ttt1[ii]为最后一位\0 时其 ascII
    码为 0, 所以此时跳出循环, 从而求出字符串长度 ll
    write_com(0x80+hang1);
    for(ii=0;ii<ll;ii++) //显示第一行字符串

```

```

        write_data(ttt1[ii]);

    }

void LCD_1602_t(uchar hang1,char *ttt1)
{
    for(ll=0,ii=0;ttt1[ii];ii++,ll++);//求第一行字符串长度,当 ttt1[ii]为最后一位\0 时其 ascII 码
    为 0, 所以此时跳出循环, 从而求出字符串长度 ll
    write_com(0x80+hang1);
    for(ii=0;ii<ll;ii++)          //显示第一行字符串
    write_data(ttt1[ii]);

}

//在特定的位置替换为一位数字 (必须是 0~9 的数字, 否则显示的将不是想要显示的数字)
void LCD_1602_num(uchar hang1,uchar nnn1)
{
    write_com(0x80+hang1);
    write_data(nnn1+48);          //将数字转化为 ascII 码, 数字+48=相应数字的 ascII 码
    0x30=48

}

////////////////////////////////////
//////////////////////////////////////延时子程序////////////////////////////////////
////////////////////////////////////

void delay(uint k)
{
    uint x,y;
    for(x=k;x>0;x--)
        for(y=1;y>0;y--);
}

////////////////////////////////////
//////////////////////////////////////写指令子程序////////////////////////////////////
////////////////////////////////////

void write_com(uchar com)
{
    lcd_rs=0;                    //RS 为 L, RW 为 L 时为写指令, 开发板硬件已将 RW 接地了,
    所以不用控制 RW 了
    P1=com;                      //送指令代码到 P1 口
}

```

```

    delay(1);          //延时 1ms, 时序要求
    lcd_en=1;         //使能信号给高脉冲, 使得 1602 读取 P0 口上的指令, 高脉冲的
    脉宽有要求, 不同厂家的 1602 要求也不一样
    delay(1);
    lcd_en=0;
    delay(1);
    lcd_rs=1;
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////写数据子程序/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

void write_data(uchar data0)
{
    lcd_rs=1;          //RS 为 H, RW 为 L 时为写数据, 开发板硬件已将 RW 接地了,
    所以不用控制 RW 了
    P1=data0;         //送数据到 P1 口
    delay(1);
    lcd_en=1;         //使能信号给高脉冲, 使得 1602 读取 P0 口上的数据, 高脉冲的
    脉宽有要求, 不同厂家的 1602 要求也不一样
    delay(1);
    lcd_en=0;
    delay(1);
    lcd_rs=1;
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////1602 初始化子程序/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

void init00()
{
//  dula=1;          //关数码管段锁存器, 开发板上数码管和液晶公用一个数
    据线 P0
//  wela=0;         //关数码管位锁存器
    write_com(0x38); //设置 1602 为 16X2 显示, 5X7 点阵, 8 为数据口
    write_com(0x0c); //设置显示开/关及光标设置, 0x0c 为开显示, 不显示光标但不
    闪烁。00001DCB: D=1, 开显示, D=0 关显示; C=1 显示光标, C=0 不显示光标; B=1 光标
    闪烁, B=0 光标不闪烁
    write_com(0x06); //0x06 为当写一个字符后地址指针加一, 且光标加一, 整屏不
    移动。000001NS: N=1 写一个字符后地址指针加一, 光标加一, N=0 写入一个字符后地址指
    针减一, 光标减一; S=1 写入一个字符后, 整屏左移 (N=1) 或者右移 (N=0), 以得到光标
    不移动而屏幕移动的效果, S=0 写入一个字符后整屏显示但不移动。
}

```

```

    write_com(0x80);    //数据指针设置, 0x80+地址码 (0~27H, 40~67H), 00H~0fH 和
40H~4fH 为可显示区间, 其余区间为寄存器
    write_com(0x01);    //显示清屏
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////DS12887 时钟芯片初始化函数/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

void init()
{
    write_ds(0x0B,0x26); //向寄存器 B 写数据, 设置 DS12887
    write_ds(0x0A,0x20); //向寄存器 A 写数据, 设置 DS12887 让其开始工作
    read_ds(0x0c);      //读寄存器 c 清除 irq 引脚输出
    EA=1;
    EX1=1;    //允许外部中断
    IT1=0;    //跳变沿触发
    LCD_1602_str(0,table1); //显示 table1[]="_2009-07-13__Mon";table2[]="___00:00:00";
    LCD_1602_str(0x40,table2);
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////在 LCD1602 上刷新时分秒/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

void write_sfm(uchar add,uchar date)
{
    uchar shi,ge;
    shi=date/10;
    ge=date%10;
    LCD_1602_num(0x40+3+add,shi);
    LCD_1602_num(0x40+4+add,ge);
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////在 LCD1602 上刷新年月日/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

void write_nyr(uchar add,uchar date)
{
    uchar shi,ge;
    shi=date/10;
    ge=date%10;
}

```

```

LCD_1602_num(1+add,shi);
LCD_1602_num(2+add,ge);

}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////向 DS12887 写数据/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

void write_ds(uchar add,uchar date)
{
    dscs=0;
    dsas=1;
    dsds=1;
    dsrw=1;
    P0=add;
    dsas=0;
    dsrw=0;
    P0=date;
    dsrw=1;
    dsas=1;
    dscs=1;
}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////从 DS12887 读数据/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

uchar read_ds(uchar add)
{
    uchar ds_date;
    dsas=1;
    dsds=1;
    dsrw=1;
    dscs=0;
    P0=add;
    dsas=0;
    dsds=0;
    P0=0xff;
    ds_date=P0;
    dsds=1;
    dsas=1;
    dscs=1;
    return ds_date;
}

```

```

}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////设置起始时间/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

void set_time()
{
    s1=1;//为读数据作准备
    s2=1;
    s3=1;
    if(s3_num==0)

        write_com(0x0c);//正常工作状态，不开光标，关光标闪烁
    else
        write_com(0x0e);//设定时间状态，开光标，开光标闪烁

    if(s3==0)
    {
        delay(600);
        if(s3==0)
        {
            s3_num++;

            if(flag==1)
            {
                s3_num=255; //按键 s3 进行清闹钟闹钟标志位
                flag=0;
            }
            if(s3_num>=12)
            {
                LCD_1602_t(0,table1);
                LCD_1602_t(0x40,table2);
                s3_num=0;
            }
            while(!s3);
        }
    }
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////设置秒/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

if(s3_num==1)
{
    write_com(0x80+0x40+11); //地址指针指向 miao 显示位置处
    //秒加
    if(s1==0)
    {
        delay(600);//延时去抖
        if(s1==0)
        {
            miao++;
            if(miao==60)
            miao=0;
            write_sfm(7,miao);
            while(!s1);
        }
    }
    //秒减
    if(s2==0)
    {
        delay(600);//延时去抖
        if(s2==0)
        {
            miao--;
            if(miao==-1)
            miao=59;
            write_sfm(7,miao);
            while(!s2);
        }
    }
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////设置分/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```
if(s3_num==2)
{
    write_com(0x80+0x40+8);    //地址指针指向 fen 显示位置处
    //分加
    if(s1==0)
    {
        delay(600);//延时去抖
        if(s1==0)
        {
            fen++;
            if(fen==60)
            fen=0;
            write_sfm(4, fen);
            while(!s1);
        }
    }
    //分减
    if(s2==0)
    {
        delay(600);//延时去抖
        if(s2==0)
        {
            fen--;
            if(fen== -1)
            fen=59;
            write_sfm(4, fen);
            while(!s2);
        }
    }
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////设置时////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
if(s3_num==3)
{
    write_com(0x80+0x40+5);    //地址指针指向 shi 显示位置处
    //时加
    if(s1==0)
    {
```



```

        delay(600); //延时去抖
        if(s1==0)
        {
            shi++;
            if(shi==24)
            shi=0;
            write_sfm(1,shi);
            while(!s1);
        }
    }
    //时减
    if(s2==0)
    {
        delay(600); //延时去抖
        if(s2==0)
        {
            shi--;
            if(shi==-1)
            shi=23;
            write_sfm(1,shi);
            while(!s2);
        }
    }
}

```

```

/////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////// 设置星期 ///////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

if(s3_num==4)
{
    write_com(0x80+13); //地址指针指向 xingqi 显示位置处
    //星期加
    if(s1==0)
    {
        delay(600); //延时去抖
        if(s1==0)
        {
            xinqi++;
            if(xinqi==8)
            xinqi=1;
            if(xinqi==1) //显示在 LCD1602 上

```

```

        LCD_1602_t(13,xq1);
    if(xinqi==2)
        LCD_1602_t(13,xq2);
    if(xinqi==3)
        LCD_1602_t(13,xq3);
    if(xinqi==4)
        LCD_1602_t(13,xq4);
    if(xinqi==5)
        LCD_1602_t(13,xq5);
    if(xinqi==6)
        LCD_1602_t(13,xq6);
    if(xinqi==7)
        LCD_1602_t(13,xq7);
    while(!s1);
}
}
//星期減
if(s2==0)
{
    delay(600);//延时去抖
    if(s2==0)
    {
        xinqi--;
        if(xinqi==0);
        xinqi=7;
        if(xinqi==1)           //显示在 LCD1602 上
            LCD_1602_t(13,xq1);
        if(xinqi==2)
            LCD_1602_t(13,xq2);
        if(xinqi==3)
            LCD_1602_t(13,xq3);
        if(xinqi==4)
            LCD_1602_t(13,xq4);
        if(xinqi==5)
            LCD_1602_t(13,xq5);
        if(xinqi==6)
            LCD_1602_t(13,xq6);
        if(xinqi==7)
            LCD_1602_t(13,xq7);
        while(!s2);
    }
}
}
}

```

```

//////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////设置日//////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////

```

```

if(s3_num==5)
{
    write_com(0x80+10); //地址指针指向 ri 显示位置处
    //日加
    if(s1==0)
    {
        delay(600);//延时去抖
        if(s1==0)
        {
            ri++;
            if(ri>=32)
                ri=1;
            write_nyr(8,ri);
            while(!s1);
        }
    }
    //日减
    if(s2==0)
    {
        delay(600);//延时去抖
        if(s2==0)
        {
            ri--;
            if(ri==0)
                ri=31;
            write_nyr(8,ri);
            while(!s2);
        }
    }
}

```

```

//////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////设置月//////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////

```

```

if(s3_num==6)
{
    write_com(0x80+7);//地址指针指向 yue 显示位置处
    //月加
    if(s1==0)
    {
        delay(600);//延时去抖
        if(s1==0)
        {
            yue++;
            if(yue==13)
            yue=1;
            write_nyr(5,yue);
            while(!s1);
        }
    }
    //月减
    if(s2==0)
    {
        delay(600);//延时去抖
        if(s2==0)
        {
            yue--;
            if(yue==0)
            yue=12;
            write_nyr(5,yue);
            while(!s2);    //松键检测
        }
    }
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////设置年////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

if(s3_num==7)    //年只调节后两位
{
    write_com(0x80+4);//地址指针指向 nian 显示位置处
    //年加

```

```

if(s1==0)
{
    delay(600);//延时去抖
    if(s1==0)
    {
        nian++;
        if(nian==100)
        nian=0;
        write_nyr(2,nian);
        while(!s1);
    }
}
//年减
if(s2==0)
{
    delay(600);//延时去抖
    if(s2==0)
    {
        nian--;
        if(nian==-1)
        nian=99;
        write_nyr(2,nian);
        while(!s2);
    }
}
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////  进  入  闹  铃  设  置  区  显  示  提  示
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

if(s3_num==8)
{
    LCD_1602_t(0,table3);
    write_sfm(7,n_miao);
    write_sfm(4,n_fen);
    write_sfm(1,n_shi);
    s3_num++;
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////设置闹钟秒/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

if(s3_num==9)
{
    write_com(0x80+0x40+11); //地址指针指向 miao 显示位置处
    //秒加
    if(s1==0)
    {
        delay(600);//延时去抖
        if(s1==0)
        {
            n_miao++;
            if(n_miao==60)
            n_miao=0;
            write_sfm(7,n_miao);
            while(!s1);
        }
    }
    //秒减
    if(s2==0)
    {
        delay(600);//延时去抖
        if(s2==0)
        {
            n_miao--;
            if(n_miao==-1)
            n_miao=59;
            write_sfm(7,n_miao);
            while(!s2);
        }
    }
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////设置闹钟分/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

if(s3_num==10)
{
    write_com(0x80+0x40+8); //地址指针指向 fen 显示位置处
    //分加
    if(s1==0)
    {
        delay(600);//延时去抖
        if(s1==0)
        {
            n_fen++;
            if(fen==60)
            n_fen=0;
            write_sfm(4,n_fen);
            while(!s1);
        }
    }
    //分减
    if(s2==0)
    {
        delay(600);//延时去抖
        if(s2==0)
        {
            n_fen--;
            if(n_fen==-1)
            n_fen=59;
            write_sfm(4,n_fen);
            while(!s2);
        }
    }
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////设置闹钟时////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

if(s3_num==11)
{
    write_com(0x80+0x40+5); //地址指针指向 shi 显示位置处
    //时加
    if(s1==0)
    {

```

```

delay(600);//延时去抖
if(s1==0)
{
    n_shi++;
    if(n_shi==24)
    n_shi=0;
    write_sfm(1,n_shi);
}
}
//时减
if(s2==0)
{
    delay(600);//延时去抖
    if(s2==0)
    {
        n_shi--;
        if(n_shi==-1)
        n_shi=23;
        write_sfm(1,n_shi);
    }
}
}

```

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////等到闹钟时设置完成后再将设置时间送入 DS12887，从而保证退出设置
状态时刻的时间与设定时间相等/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

write_ds(9,nian);//将初值送入 DS12887
write_ds(8,yue);//将初值送入 DS12887
write_ds(7,ri);//将初值送入 DS12887
write_ds(6,xinqi);//将初值送入 DS12887
write_ds(4,shi);//将初值送入 DS12887
write_ds(2,fen);//将初值送入 DS12887
write_ds(0,miao);//将初值送入 DS12887

```

```

write_ds(5,n_shi);//将初值送入 DS12887 //闹钟部分
write_ds(3,n_fen);//将初值送入 DS12887
write_ds(1,n_miao);//将初值送入 DS12887

```

```

}

```



```

}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////从 DS12887 中读时钟，并将读到的时钟信息显示到 LCD1602 上
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

void r_x()
{
    LCD_1602_t(0," 20  -  -   Mon");           //uchar   code
table1[] =" 2009-07-13  Mon";
                                           //uchar   code
table2[]="    00:00:00";
        LCD_1602_t(0x40,"      :  :  ");
    //读时间
    miao=read_ds(0);
    fen=read_ds(2);
    shi=read_ds(4);
    xinqi=read_ds(6);
    ri=read_ds(7);
    yue=read_ds(8);
    nian=read_ds(9);

    //将读到的时间显示到 LCD 上
    write_sfm(7,miao);
    write_sfm(4,fen);
    write_sfm(1,shi);
    write_nyr(2,nian);
    write_nyr(5,yue);
    write_nyr(8,ri);

    //显示星期
    if(xinqi==1)
        LCD_1602_t(13,xq1);
    if(xinqi==2)
        LCD_1602_t(13,xq2);
    if(xinqi==3)
        LCD_1602_t(13,xq3);
    if(xinqi==4)
        LCD_1602_t(13,xq4);
    if(xinqi==5)
        LCD_1602_t(13,xq5);
}

```

```

if(xinqi==6)
    LCD_1602_t(13,xq6);
if(xinqi==7)
    LCD_1602_t(13,xq7);
}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////          主          函          数
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void main()
{
    init();    //初始化
    while(1) //不断读取并刷新显示时间
    {
        set_alarm();
        key_s4_scan();
        if(s4_num==0)
        {
            set_time(); //设置时钟

            if(s3_num==0)
            {
                r_x(); //读取并显示时钟 5
            }
            delay(1000);/////////////////////////////////////////////////////////////////
            //闹钟响
            if(flag==1)
            {
                didi();
                flag=0;
                if(s1==0)
                {
                    delay(600);
                    if(s1==0)
                    {
                        beep=1;
                        while(!s1);
                    }
                }
            }
        }
    }
}

```

```
        }

    }
}
else
    miao1();

//-----
delay(600);
delay(600);
delay(600);
delay(600);
//-----
}
}
```

```
//外部中断 1 服务程序，闹铃时间到
void nao() interrupt 2
{
    flag=1;//置位，表示定时时间到
    read_ds(0x0c); //清 DS12887 定时标志
}
```

```
////////////////////////////////////
////////////////////////////////////    秒表部分    //////////////////////////////////////
////////////////////////////////////
void key_s4_scan()
{
    if(s4==0)
    {
        delay(1000);
        if(s4==0)
        {   s4_num++;
            while(!s4);
        }
        if(s4_num>=2)    //如果连按两次 s4 键，则退出秒表界面
            s4_num=0;
    }
}
```

```

    }
}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////秒表部分显示程序/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

/////sbit stop = P2^1; //停表或者启动  按键 s2 实现停止秒表计数
/////sbit rest = P2^0; //清零          按键 s1 实现清零秒表计数

void miao1(void)
{   init00();      //1602 初始化
    int0();        //定时器 0 初始化

    LCD_1602_str(0,k);      //初始化 LCD1602 并在第一行第五列显示小数点"."
    k[]="____."
    LCD_1602_t(8,j);      //在 LCD1602 第一行第九列显示时间单位秒“s”
    j[]="s"

    while(1)
    {
        if(stop==0)      //检测停止或者启动按键是否按下
        {
            delay(10);    //延时 ， 去抖动作用
            if(stop==0)
            {
                sta++;
                if(sta==2)//在启动状态下连续按下 stop 两次则 进入时改变状态为暂停
                sta = 0;
                while(!stop);//检测按键是否松开， 否则停在这里
            }
        }
        if(rest==0)      //检测复位按键是否按下
        {
            delay(10);//延时 100ms ， 去抖动作用
            if(!rest)
            {
                ms=0;    //所有变量清零
                s=0;
                s_100=0;
            }
        }
    }
}

```

```

        sta=0;    //置状态为暂停
        temp  = 0;
        ET0   = 0;    //关中断
        TR0   = 0;    //关定时器
        TH0   = 0xFC; //给定时器 0 高 8 位装初值
        TL0   = 0xC9; //给定时器 0 低 8 位装初值
        while(!rest);    //检测按键是否松开，否则停在这里
    }
}

if(sta==1)    //当系统处于启动状态时
{
    ET0   = 1; //开定时器 0 中断;
    TR0   = 1; //启动定时器 0;
}
else    //当系统处于暂停状态时
{
    ET0   = 0; //关中断
    TR0   = 0; //关定时器
}

//求每位显示数据    “__00_000s”
ms_bai=ms/100;
ms_shi=(ms/10)%10;
ms_ge=ms%10;

s_shi=(s/10)%10;
s_ge=s%10;

s_100_shi=((s_100)/10)%10;
s_100_ge=(s_100)%10;

//LCD1602 显示 毫秒值
LCD_1602_num(5,ms_bai);
LCD_1602_num(6,ms_shi);
LCD_1602_num(7,ms_ge);

//LCD1602 显示 秒值
LCD_1602_num(2,s_shi);
LCD_1602_num(3,s_ge);

```

```

//LCD1602 显示 百秒值
if(s_100<100&& s_100>=10)
{
    LCD_1602_num(0,s_100_shi);
    LCD_1602_num(1,s_100_ge);
}
if(s_100<10)
{
    LCD_1602_num(0,112); //112+48 为不显示字符,保证最高位不为0;
    LCD_1602_num(1,s_100_ge);
}
if(s_100==0)
{
    LCD_1602_num(0,112); //112+48 为不显示字符, 保证最高位不为 0
    LCD_1602_num(1,112); //112+48 为不显示字符, 保证最高位不为 0
}

key_s4_scan();

if(s4_num==0)
{
    write_com(0x01);
    //LCD_1602_str(0,table1); //uchar code table1[] ="
2009-07-13 Mon";
//uchar code
table2[]=" 00:00:00";
// LCD_1602_str(0x40,table2);
write_ds(9,nian);//将初值送入 DS12887
write_ds(8,yue);//将初值送入 DS12887
write_ds(7,ri);//将初值送入 DS12887
write_ds(6,xinqi);//将初值送入 DS12887
write_ds(4,shi);//将初值送入 DS12887
write_ds(2,fen);//将初值送入 DS12887
write_ds(0,miao);//将初值送入 DS12887

    r_x();

    LCD_1602_t(0,table4); //uchar code table1[] ="
2009-07-13 Mon";
//uchar code
table2[]=" 00:00:00";
LCD_1602_t(0x40,table5);

```

```

        break;
    }

}

}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////  定  时  器  0  方  式  1  初  始  化  程  序
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

void int0()
{
    TMOD=0x01; //定时器 0 方式 1, 16 位手动装载模式;
    TH0=0xFC; //给定时器 0 高 8 位装初值
    TL0=0xC9; //给定时器 0 低 8 位装初值
    EA=1; //开总中断;
    ET0=1; //开定时器 0 中断;
//    TR0=1; //启动定时器 0;
    P2=0xff;//端口初始化
}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////  定  时  器  0  中  断  服  务  程  序
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

extern void Timer0(void) interrupt 1
{
    ET0= 0; //关定时器中断
    TR0= 0; //关定时器计数
    temp++; //每产生 1ms 定时中断 temp 加一
    ms=(temp%1000);//得到毫秒级数据
    if(temp%1000==0)
    {
        s++; //满 1000ms 向秒级变量进 1
        if(s==100)
        {
            s=0; //满 100s 秒级变量清零
            s_100++; //满 100s 向百秒级变量进 1
        }
    }
}

```

```

        if(s_100==100)
        {
            s_100=0; //满 10000s 百秒级变量清零
        }
    }
}

if(temp==1000) //当 1ms 定时中断次数为 1000 时，temp 清零
temp=0;

TH0 = 0xFC; //给定时器 0 高 8 位装初值 减去中断服务程序消耗的时间 0.177ms
TL0 = 0xC9; //给定时器 0 低 8 位装初值
ET0 = 1; //开定时器 0 中断;
TR0 = 1; //启动定时器 0;
}

```