

基于1602显示的电子钟的设计

信工1601_蒋文韬_2016014452

目录

- 1、问题重述
- 2、顶层模块设计
- 3、具体细节
- 4、优化与改进
- 5、具体实现展示

1、问题重述

一、题目：

【设计要求】

合理使用 STC 单片机内的定时器资源，并使用 1602 实现数字钟的功能，显示方式 xx: xx: xx（时：分：秒）。

基本部分：

能在 1602 上以 xx: xx: xx 的形式显示时间，符合真实工作情况（40 分）。

提高部分：

（1）能通过一个按键将 1602 切换到显示年月日，显示格式 xx/xx/xx（年/月/日）（20 分）。

（2）通过按键可以 调整时、分、秒（30 分）。

发挥部分：

完善电子钟的功能（10 分）。

注：

（1）设计的电子钟，使用最少的按键，按照电子表，最多使用 3 个按键。

（2）时钟工作时，其进位应该与真实的电子钟相同。

（3）显示时间和显示年月日之间的进位关系符合实际。

2、顶层模块设计

1. 顶层模块设计

模式类型	实现功能
0（基本部分）	正常显示时间（时：分：秒）
1（提高部分）	正常显示日期（年.月.日）
2（提高部分）	更改时间——小时
3（提高部分）	更改时间——分钟
4（提高部分）	更改时间——秒
5（发挥部分）	更改日期——年
6（发挥部分）	更改日期——月
7（发挥部分）	更改日期——日

只使用两个按键：SW17，SW18
按键SW17：控制模式切换
按键SW18：控制更改时间或日期（加）

3、具体细节

3-1：关于分频

参考例程11-1进行设计

注意：分频后的频率 = $(\text{IRC频率}/8/12) / (65536 - \text{TIMS}) / 2$

3、具体细节

3-2：时间和日期变量的声明和定义

```
28 //声明结构体变量，时间time，包含时分秒，整型
29 struct time
30 {
31     int hr;//时钟
32     int min;//分钟
33     int sec;//秒钟
34 };
35 //声明结构体变量，日期date，包含年月日，整型
36 struct date
37 {
38     int yr;//年
39     int m;//月
40     int d;//日
41 };
42 struct time mytime;//定义全局变量mytime
43 struct date mydate;//定义全局变量mydate
```

3、具体细节

3-2-1: 中断1——定时器0中断程序

```
void timer_0() interrupt 1//定时器0中断程序
{
    flag_show = 1;
    /*****注释掉的部分是模拟调节时间时，时间暂停，不走，这种情况只在少数电子表中出现*****/
    // if(flag_trans != 2 && flag_trans != 3 && flag_trans != 4)
    // {
    //     mytime.sec++;
    // }
    mytime.sec++;//调节时间时，时间继续走，不停表
    if(mytime.sec == 60)
    {
        mytime.sec = 0;
        mytime.min++;
    }
    if(mytime.min == 60)
    {
        mytime.min = 0;
        mytime.hr++;
    }
    if(mytime.hr == 24)
    {
        mytime.hr = 0;
        flag_leap = is_leap(mydate.yr);
    }
}
```

3、具体细节

3-2-2: 中断1——定时器0中断程序

```
int commen_mon[13] = {-1,31,28,31,30,51,30,31,31,30,31,30,31};//月份数组第0个元素不用
```

```
int flag_leap = 0;//默认为平年
```

```
/*判断平年和闰年*/  
int is_leap(int year)  
{  
    return ((year%4==0&&year%100!=0)|| (year%400==0)) ? 1:0;  
}
```


3、具体细节

3-2-3: 中断1——定时器0中断程序

```
if(mytime.hr == 24)
{
    mytime.hr = 0;
    flag_leap = is_leap(mydate.yr);
    if( (mydate.d < commen_mon[mydate.m]) || (mydate.d == commen_mon[mydate.m] && mydate.m == 2 && flag_leap == 1) )
    {
        mydate.d++;
    }
    else
    {
        mydate.d = 1;
        if(mydate.m<12)
        {
            mydate.m++;
        }
        else
        {
            mydate.m = 1;
            mydate.yr++;
        }
    }
}
```

3、具体细节

3-3: 外部中断0——按下SW17, 切换模式

```
int flag_trans = 0;
```

```
void trans() interrupt 0//切换不同的功能
{
    flag_trans++;
    lcdwritcmd(0x01);//清屏命令
    if(flag_trans == 8)
    {
        flag_trans = 0;
    }
}
```

3、具体细节

3-4： 外部中断1——按下SW18， 更改时间日期

```
void add() interrupt 2//加1
{
    if(flag_trans == 2)
    {
        mytime.hr++;//小时+1
    }
    else if(flag_trans == 3)
    {
        mytime.min++;//分钟+1
    }
    else if(flag_trans == 4)
    {
        mytime.sec++;//秒+1
    }
    else if(flag_trans == 5)
    {
        mydate.yr++;//年+1
    }
    else if(flag_trans == 6)
    {
        mydate.m++;//月+1
    }
    else if(flag_trans == 7)
    {
        mydate.d++;//日+1
    }
}
```

3、具体细节

3-5: 时间显示函数

```
/*显示时间函数*/
void show_time()
{
    /****数字型转化为字符型****/
    if(mytime.sec<10)
    {
        time_sec[0] = '0';
        sprintf(time_sec + 1, "%d", mytime.sec);
    }
    else
    {
        sprintf(time_sec, "%d", mytime.sec);
    }
    if(mytime.min<10)
    {


---


    else
    {


---


        if(mytime.hr<10)
        {


---


        else
        {


---


            lcdshowstr(0,1,time_hr); //显示小时
            lcdshowstr(2,1,":");
            lcdshowstr(3,1,time_min); //显示分钟
            lcdshowstr(5,1,":");
            lcdshowstr(6,1,time_sec); //显示秒
        }
    }
}
```

3、具体细节

3-6: 日期显示函数

```
/*显示日期函数*/
void show_date()
{
    if(mydate.d<10)
    {
        date_d[0]='0';
        sprintf(date_d + 1, "%d", mydate.d);
    }
    else
    {
        sprintf(date_d, "%d", mydate.d);
    }
    if(mydate.m<10)
    {
        date_m[0] = '0';
        sprintf(date_m + 1, "%d", mydate.m);
    }
    else
    {
        sprintf(date_m, "%d", mydate.m);
    }
    sprintf(date_yr, "%d", mydate.yr);
    lcdshowstr(0, 1, date_yr);
    lcdshowstr(4, 1, ".");
    lcdshowstr(5, 1, date_m);
    lcdshowstr(7, 1, ".");
    lcdshowstr(8, 1, date_d);
}
```

3、具体细节

3-7-1: while(1)中有什么？

```
while(1)
{
    if(flag_trans == 0 || flag_trans == 2 || flag_trans == 3 || flag_trans == 4)
    {
        if(flag_trans == 0)
        {
            lcdshowstr(0,0,Time_Show_Str);
        }
        else if(flag_trans == 2)
        {
            lcdshowstr(0,0,Ajust_Hr_Str);
        }
        else if(flag_trans == 3)
        {
            lcdshowstr(0,0,Ajust_Min_Str);
        }
        else if(flag_trans == 4)
        {
            lcdshowstr(0,0,Ajust_Sec_Str);
        }

        if(flag_show == 1)
        {
            flag_show = 0;
            show_time();
        }
    }
}
```

```
char Time_Show_Str[] = "Now,the time is: ";
char Date_Show_Str[] = "Now,the date is: ";
char Ajust_Hr_Str[] = "Now,plz add hour: ";
char Ajust_Min_Str[] = "Now,plz add min: ";
char Ajust_Sec_Str[] = "Now,plz add sec: ";
char Ajust_Yr_Str[] = "Now,plz add year: ";
char Ajust_M_Str[] = "Now,plz add mon: ";
char Ajust_D_Str[] = "Now,plz add day: ";
```

3、具体细节

3-7-2: while(1)中有什么？

```
else if(flag_trans == 1 || flag_trans == 5 || flag_trans == 6 || flag_trans == 7)
{
    if(flag_trans == 1)
    {
        lcdshowstr(0,0,Date_Show_Str);
    }
    else if(flag_trans == 5)
    {
        lcdshowstr(0,0,Ajust_Yr_Str);
    }
    else if(flag_trans == 6)
    {
        lcdshowstr(0,0,Ajust_M_Str);
    }
    else if(flag_trans == 7)
    {
        lcdshowstr(0,0,Ajust_D_Str);
    }
    if(flag_show == 1)
    {
        flag_show = 0;
        show_date();
    }
}
```

```
char Time_Show_Str[] = "Now,the time is: ";
char Date_Show_Str[] = "Now,the date is: ";
char Ajust_Hr_Str[] = "Now,plz add hour: ";
char Ajust_Min_Str[] = "Now,plz add min: ";
char Ajust_Sec_Str[] = "Now,plz add sec: ";
char Ajust_Yr_Str[] = "Now,plz add year: ";
char Ajust_M_Str[] = "Now,plz add mon: ";
char Ajust_D_Str[] = "Now,plz add day: ";
```

4、优化与改进

- 1、尽可能少使用IF-ELSE语句，避免程序臃肿
 - 2、时间或日期字符串显示时，只定义一个变量
 - 3、模块化设计的思想要贯穿设计全过程
-

5、具体实现展示

5-1：调节小时



5、具体实现展示

5-2：调节分钟



5、具体实现展示

5-3：调节秒（不停表）



5、具体实现展示

5-3：调节时间（停表）



5、具体实现展示

5-4：调节年



5、具体实现展示

5-5：调节月



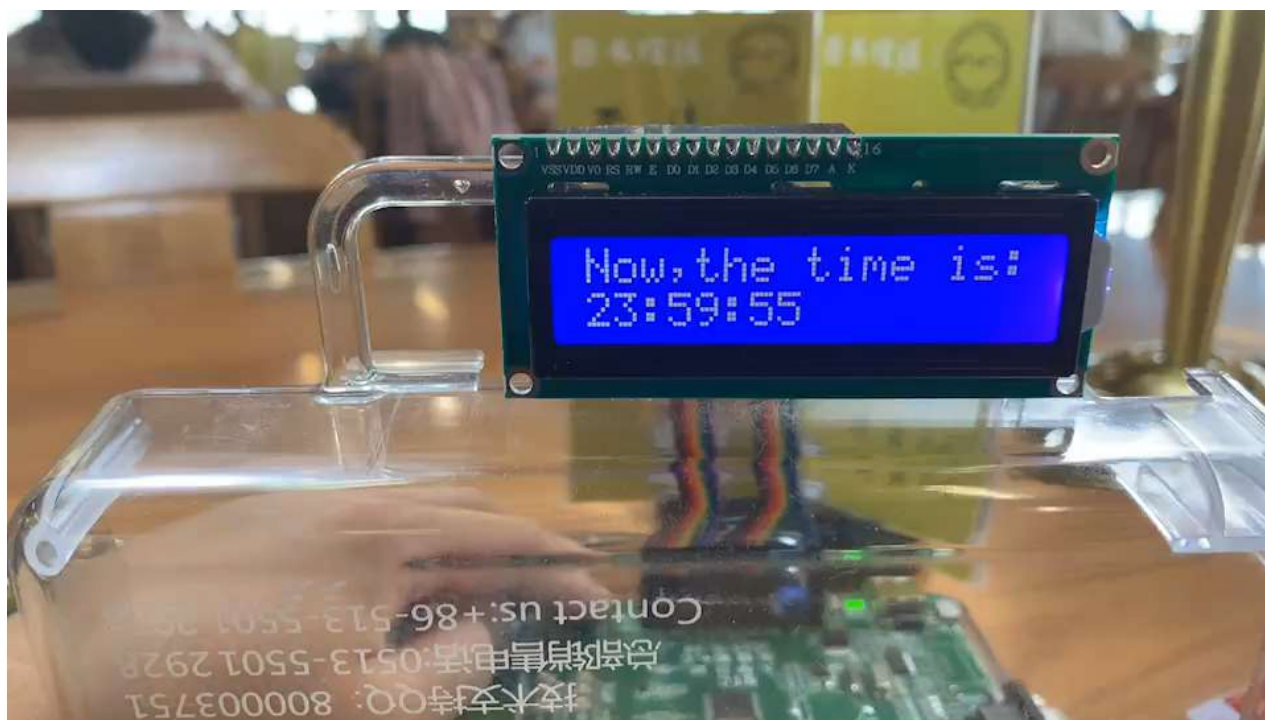
5、具体实现展示

5-7：调节日



5、具体实现展示

5-8: 闰年二月



5、具体实现展示

5-9: 平年二月



谢谢大家

不足之处恳请批评指正