单片机实验（四）

【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，对多使用 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。

【设计思路】

基础部分：

主要考察的是我们对于**定时器模式的理解**和对**定时器中断的理解**，怎么样**通过设置初值**和**分频系数**的方式来产生 1s 的时钟。

提高部分：

控制部分：我设计了两个界面，一个用来显示时间，一个用来显示日期，这两个界面的切换分别由外部中断 0 控制。除此之外，我还设置了外部中断 1 的设置时间和日期功能，这个设置功能只能够在显示时间的界面触发，且在设置时间的操作下，时钟暂停计时。

逻辑部分：增加了大月小月不同月份下的进制，闰年和非闰年下二月的进制，增加了光标显示的功能

【遇到问题&解决方法】

1. 如何设置定时器使得其产生 1s 的时钟

这部分我是翻阅了书才得到答案的

计数器/定时器寄存器组
--定时器/计数器工作模式寄存器TMOD

■ M1和M0（TMOD.1和TMOD.0），定时器/计数器0模式选择。

M1	M0	工作模式
0	0	16位自动重新加载模式。当溢出时，将RL_TH0和RL_TL0的值自动重新加载到TH0和TL0中

对于在 16 重加载模式的定时器 0 来说，当溢出时，RL_TH0 和 RL_TL0 的值会自动加载到 TH0 和 TL0 中

■ 定时器/计数器1的计数初值寄存器TH1和TL1，它们用于保存定时器/计数器1的计数初值。

名字	地址	复位值	B7	B6	B5	B4	B3	B2	B1	B0
TH1	0x8D	00000000	计数初值的高8位							
TL1	0x8B	00000000	计数初值的低8位							

在编程中我们可以把定时器的初值放入 TH,TL 寄存器中

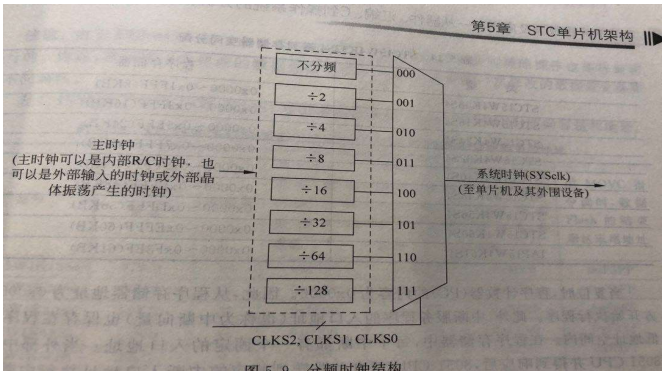
计数器/定时器寄存器组
--辅助寄存器AUXR

定时器0/1的速度控制位T0x12/T1x12。

■ 当该位为0时，定时器0/1是12分频；当该位为1时，定时器0/1不分频。

名字	地址	复位值	B7	B6	B5	B4	B3	B2	B1	B0
AUXR	0x8E	00000001	T0x12	T1x12	UART_M0x6	T2R	T2_C/T	T2x12	EXTRAM	S1ST2

同时我们我们可以设置 AUXR 寄存器，当该寄存器的第一位是 0 时为 12 分频，当该位为 1 时，则不分频。



除此之外，我们还可以设置 CLK_DIV 寄存器，来对 RC 时钟进行分频操作。

根据以上的知识，综合考虑

我设置了系统时钟为 RC 时钟/8，定时器接收时钟为 SYSCLK/12

所以要产生 1s 的时钟，需要赋给的初值是：65535-1000000/2/8+1=3036

【设计细节】

1.如何简洁的设计时钟

我在上一步时就以及设计好了时间间隔为 1s 的定时器中断，然后我还需要做的就是把时，分，秒具体化。我的思路可能和其他人不太一样。我尽量使得代码简洁，并且在中断尽可能地少写代码。

```
void timer_0() interrupt 1
{
  if(!Modle_CH)//不在设置状态
  Time++;
}
```

我在外部设置了一个全局变量 Time 来记录我的总时间，然后在定时器中断中对它进行++的操作。然后在外部设置了一个对秒，分，小时的判断

```
136  void Timeset(void)
137  {
138    hour=Time/3600;
139    min=Time%3600/60;
140    sec=Time%3600%60;
141  }
```

这样写的好处是：不需要再用 if 语句进行判断，是否满 60 进位，得到的时间就是自动进位的

```
if(Time==3600*24)//24小时进位
{
  Time=0;
  day++;
}
```

同时我们让时间到 24h 后归零

2.关于大月小月和润年的判断

自我感觉这一段可以更简单的，主要是采用了 if 语句和或的操作。

```
if(month==1|month==3|month==5|month==7|month==8|month==10|month==12)//大月
{
  if(day==32)
  {
    day=1;
    month++;
  }
}
else if(month==4|month==6|month==9|month==11)//小月
{
  if(day==31)
  {
    day=1;
    month++;
  }
}
```

闰年的判断稍微复杂了一点需要判断是否是 100 的倍数

```
else if(month==2)//二月份
{
  if(year%400==0)
  {
    if(day==30)
    {
      day=1;
      month++;
    }
  }
  else if(year%4==0&year%100!=0)
  {
    if(day==30)
    {
      day=1;
      month++;
    }
  }
  else if(1)
  {
    if(day==29)
    {
      day=1;
      month++;
    }
  }
}
```

这里关于月份的判断我都采取了 else if 的操作，保证了不会出现触发多次的情况

3.设置时间时候暂停和光标显示

光标显示是吕蒙同学给我的灵感，我觉得挺好的，所以就之间用了。

主要思路就是：在第一行要改的时间下面打上^^。

```
if(Dis_Model)
{
  switch(Modle_CH)//进行高亮显示
  {
    case 0:
      Display_Time();
      lcdshowstr(0,1,"                ");
    break;
    case 1:
      Display_Time();
      lcdshowstr(0,1,"                ");
      lcdshowstr(11,1,"^^");
    break;
```

然后就是设置时间的时候暂停，就是在外部设置个标志位，当标志位改变时，定时器中的 TIME 就不再增加。

```
void timer_0() interrupt 1
{
  if(!Modle_CH)//不在设置状态
  Time++;
}
```

【实验结果】

把代码下载到单片机后：RC 时钟为 12MHZ

界面 1 显示时间

按下 SW17 后切换到界面 2

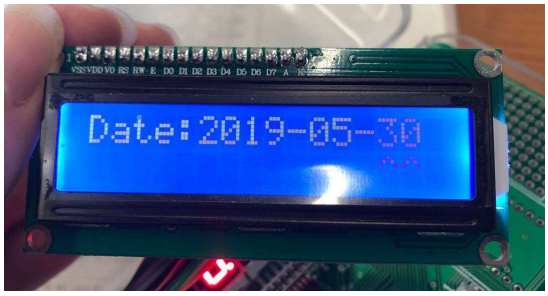
再按下 SW17 返回界面 2 后，再按下 SW18 进入设置模式：

（光标由于反光，不清楚）

之后每按一下 SW18 切换一个位：

（分钟）

（小时）



（天）

（月份）

（年份）

下面验证一下大月，小月、24 小时归零和二月：

（24 小时归零）

（10s 后）

大月：

（大月 31 日）

小月：

（小月 30 日）

非闰年二月：

（二月 28 日）

闰年二月：

【程序代码】

作业：

```
#include "reg51.h"
#include "stdio.h"
#include "led1602.h"
#define TIMS 3036
sfr AUXR  =0x8E;
sfr AUXR2 =0x8F;
sfr CLK_DIV=0x97;
xdata long Time=0;//×ÜÃëÊý
unsigned int  hour;//Ð¡Ê±
unsigned int  min;//·ÖÖÓ
unsigned int  sec;//Ãë
unsigned int  year=2019;//Äê
unsigned int  month=5;//ÔÂ·Ý
unsigned int  day=30;//ÌìÊý
unsigned int  Time_flag=0;//ÊÇ·ñ,úÐÂÊ±¼ä
unsigned int  Modle_CH=0;//,Ä±äÃ£Ê½
xdata unsigned char str[20];
unsigned int Dis_Model=1;//1Îª Ê±¼äÏÔÊ¾£¬0Îª ÈÕÆÚÏÔÊ¾
void Timeset(void);
void Display_Time(void);
void Display_Date(void);
void Time_Jud(void);
void timer_0() interrupt 1
{
   if(!Modle_CH)//²»ÔÚÉèÖÃ×´Ì¬
   Time++;
}
service_int0() interrupt 0
{
   if(Modle_CH==0)
   Dis_Model^=1;
   else
```

```
        switch(Modle_CH)//Ôö¼ÓÊ±¼ä
        {
            case 1:
                Time++;
            break;
            case 2:
                Time+=60;
            break;
            case 3:
                Time+=3600;
            break;
            case 4:
                day++;
            break;
            case 5:
                month++;
            break;
            case 6:
                year++;
            break;
        }
}
service_int1() interrupt 2
{
   if(Dis_Model)
       Modle_CH++;
   if(Modle_CH==7)
       Modle_CH=0;
}
void main()
{
   unsigned int i;
   CLK_DIV=0x04; //Ö÷Ê±ÖÓ16·ÖÆµ
   TL0=TIMS;
```

```
TH0=TIMS>>8;
AUXR&=0x7F;//AUXR×î,ßÎ»ÖÃ0£¬SYSclk/12×÷¶¨Ê±Æ÷Ê±ÖÓ
AUXR2|=0x00;//
TMOD=0x00;//¶¨Ê±Æ÷0¹¦×÷Ä£Ê½Ê16Î»×Ô¯ÖØ¼ÓÔØÄ£Ê½
TR0=1;//Æô¶¯¶¨Ê±Æ÷/¼ÆÊýÆ÷0
ET0=1;//Ê¹ÄÜ¶¨Ê±Æ÷/¼ÆÊýÆ÷0
IT1=1;//Ê¹ÄÜÍâ²¿ÖÐ¶ÏÌ1
EX1=1;
IT0=1;//Ê¹ÄÜÍâ²¿ÖÐ¶ÏÌ0
EX0=1;
EA=1;//Ê¹ÄÜCPUÈ«¾ÖÖÐ¶ÏÌ

P0M0=0;
P0M1=0;
P2M0=0;
P2M1=0;
for(i=0;i<10000;i++);
lcdwait();
lcdinit();
while(1)
{
    Time_Jud();
    Timeset();
    if(Dis_Model)
    {
        switch(Modle_CH)//½øÐÐ,ßÁÁÎÔÊ¾
        {
            case 0:
                Display_Time();
                lcdshowstr(0,1,"                ");
            break;
            case 1:
                Display_Time();
                lcdshowstr(0,1,"                ");
```

```
                lcdshowstr(11,1,"^^");
            break;
            case 2:
                Display_Time();
                lcdshowstr(0,1,"                ");
                lcdshowstr(8,1,"^^");
            break;
            case 3:
                Display_Time();
                lcdshowstr(0,1,"                ");
                lcdshowstr(5,1,"^^");
            break;
            case 4:
                Display_Date();
                lcdshowstr(0,1,"                ");
                lcdshowstr(13,1,"^^");
            break;
            case 5:
                Display_Date();
                lcdshowstr(0,1,"                ");
                lcdshowstr(10,1,"^^");
            break;
            case 6:
                Display_Date();
                lcdshowstr(0,1,"                ");
                lcdshowstr(5,1,"^^^^");
            break;
        }
    }
    if(!Dis_Model)
    {
        Display_Date();
    }
}
```

```
}
void Timeset(void)
{
    hour=Time/3600;
    min=Time%3600/60;
    sec=Time%3600%60;
}
void Display_Time(void)
{
    sprintf((char*)str,"Time:%02d:%02d:%02d   ",hour,min,sec);
    lcdshowstr(0,0,str);
}
void Display_Date(void)
{
    sprintf((char*)str,"Date:%04d-%02d-%02d   ",year,month,day);
    lcdshowstr(0,0,str);
}
void Time_Jud(void)
{
    if(Time>=86400)//24Ð¡Ê±½øÎ»
    {
        Time=0;
        day++;
    }
    Time_flag=1;
    if(month==1|month==3|month==5|month==7|month==8|month==10|month==1
2)//´óÔÂ
    {
        if(day==32)
        {
            day=1;
            month++;
        }
    }
```

```
else if(month==4|month==6|month==9|month==11)//Ð¡ÔÂ
{
    if(day==31)
    {
        day=1;
        month++;
    }
}
else if(month==2)//¶þÔÂ·Ý
{
    if(year%400==0)
    {
        if(day==30)
        {
            day=1;
            month++;
        }
    }
    else if(year%4==0&year%100!=0)
    {
        if(day==30)
        {
            day=1;
            month++;
        }
    }
    else if(1)
    {
        if(day==29)
        {
            day=1;
            month++;
        }
    }
```

```
        }
        if(month==13)
        {
            month=1;
            year++;
        }
    }
```