西安交通大學

XI'AN JIAOTONG UNIVERSITY

电子系统设计

实验报告

班级：xxxx

姓名：xxxx

学号：xxxxxxxxx

数字钟设计

一、实验目的

1.掌握定时器的结构及工作原理，并能在编程中熟练运用。
2.掌握中断系统的概念与工作原理，并能熟练运用。
3.进一步熟悉数码管的显示及单片机其他外设。

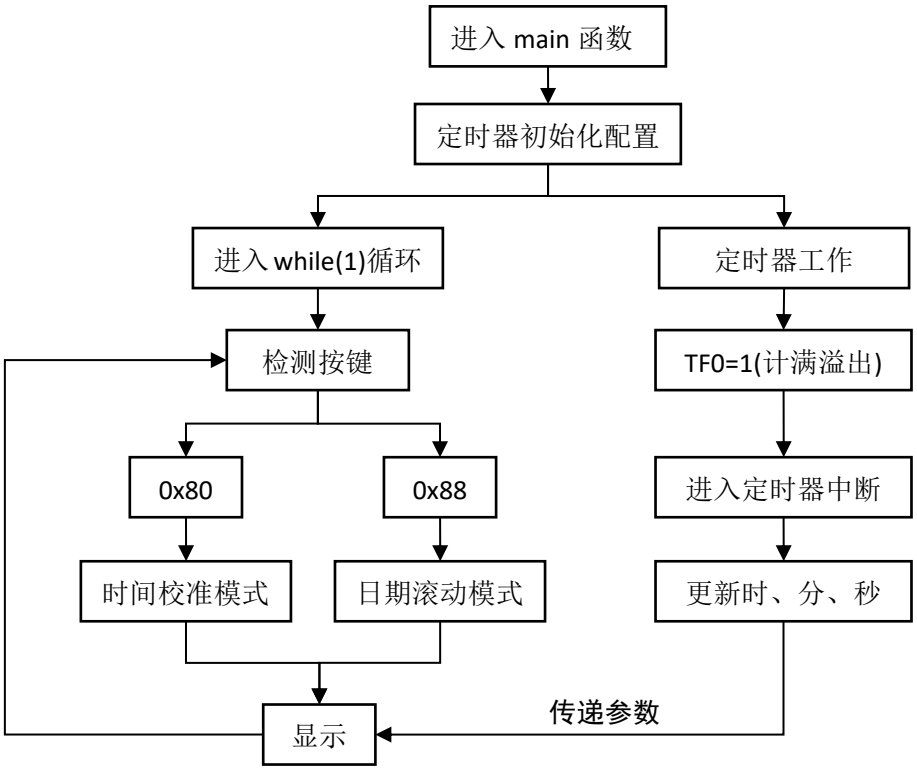
二、实验要求

1.时-分-秒（2 位-2 位-2 位）显示
2.可通过按键置入时间值(参照电子表设置时间工作模式）
3.可通过按键控制在 LED 上从右向左滚动显示年_月_ 日 3 次，如：2013_01_20 空空 2013_01_20

三、系统方案设计

1.总体方案设计

1）系统总体框图



2）系统设计思路

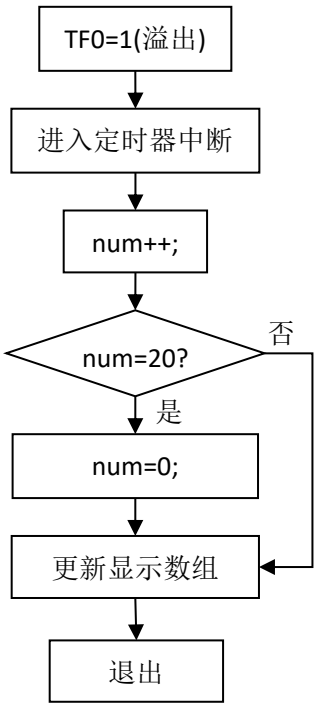
本实验中采用定时器中断实现时、分、秒的计时与更新，利用数码管的动态扫描实现时间与日期的显示。为了实现时间校准功能，在主函数的 while(1)循环中设置了按键检测，若没有按键按下或无效按键按下，则执行

显示函数，显示实时时间；若有效按键按下，则根据按下的按键选择时间校准模式或日期滚动显示模式，执行完该子函数后，程序跳回主函数，继续按键检测和实时显示时间。

2. 各模块方案设计

1）定时器中断设计

定时器中断模块流程图如下：



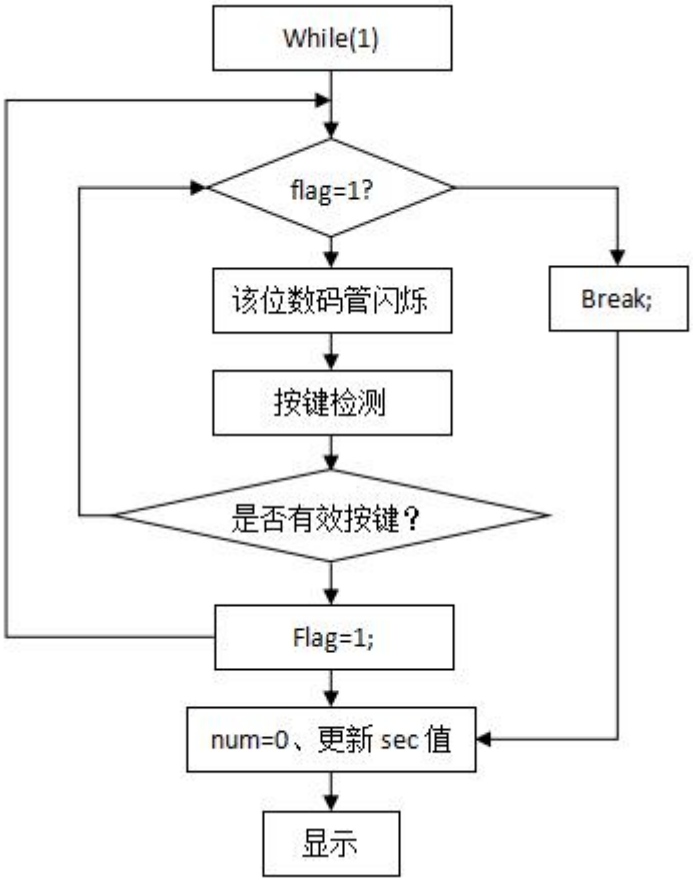
程序中，为实现 1s 的定时，采用定时器 0、设定为工作方式 1（16 位）。定时器的计数初值设定为 TH0=0x3C、TL0=0xB0，即 15536。由于计满时为 65536，故计数器溢出一次时，计数次数为 50000，对应 50ms。这样，设置计数变量为 num，每计满一次进入定时器中断函数，则 num++，当 num 计满 20 时即为 1s。此时，秒计时变量 sec++，同时根据秒计时变量更新分、时的计数变量 min、hour，并根据当前对应的时刻值更新显示数组，即可实现时钟的实时显示。

2）时间校准模块设计

本设计中，可实现时、分、秒按位顺序调整，用闪烁模式来提醒用户当前调整位。由于每一位都遵循了同样的程序设计步骤，故在此以调整秒的个位为例，说明算法流程。流程图如下：

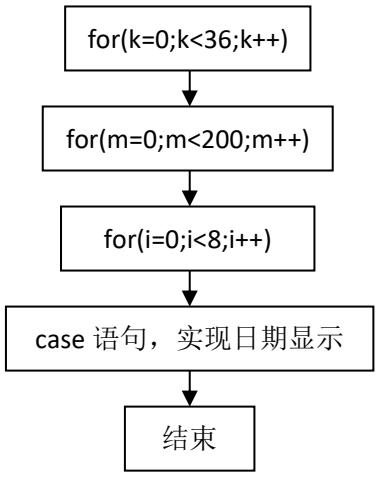
用 while(1)循环来实现实时按键检测和数码管显示，当无按键按下或者按下的按键无效时，数码管实时显示当前时间，并在当前调整位上采取闪烁模式，以提醒用户。当有效的校准按键按下时，将 flag 赋值为 1，并跳出循环。此时，根据校准时间重新设定时、分、秒的值，并在数码管上显示。

需要注意的是，由于设定时间时计时器还在计数，故计时器中断中的 num 为历史累计值，这直接导致了置入的秒计时个位值不准，故在对秒进行校准时，需将 num 计数值清零。



3）日期滚动模式设计

流程图如下：



其中，最外层的 k 循环用于控制字母滚动位置，第二层的 m 循环用于延时，控制字幕滚动的速度，最内层的 i 循环用于动态扫描，使八个数码管能同时显示不同的数字。

4）数据存储结构设计

本实验中，变量均以 unsigned char 型或 int 型定义，数组以 unsigned char 数组定义。程序中用到三个数组，分别为 seg_table、show_table 和 date。其中，

seg_table 为不可更改类型，按顺序存储了 0--9 的十个数字所对应的数码管显示的真值表，用于查找；show_table 用于存储实时的显示数据信息。依次取出 hour、min、sec 的十位与个位，到 seg_table 中查找，并赋值给 show_table，让数码管显示 show_table 的内容，即为实时时间。date 用于存储日期信息。

四、实验结果

1. 时-分-秒（2 位-2 位-2 位）显示

上电，即可实现时钟功能，并实时显示时间。

2. 可通过按键置入时间值(参照电子表设置时间工作模式）

按下按键 0x80，即进入时间校准模式，进入后不需再按其他按键，只需按位调整时间。当前所调整的数码管会闪烁数字，以提示用户当前调整位。当最后一位校准结束后，自动恢复实时时钟模式。

3. 可通过按键控制在 LED 上从右向左滚动显示年_月_ 日 3 次，如：2013_01_20 空空 2013_01_20

按下按键 0x88，即进入日期滚动模式，日期从右向左按规定的格式滚动 3 次，结束后自动恢复实时时钟模式 。

五、资源使用情况

资源使用情况为：

Data=67.0
Xdata=0
Code=2801

说明占用的内部 RAM 为 67 字节，外部 RAM 为 0 字节，程序占用的程序存储器空间为 2801 字节。

附录

完整代码如下：

```
#include <REG51.H>                  //special function register declarations
#include <absacc.h>
#include <stdio.h>

unsigned char seg_table[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
                                    //查找表
unsigned char show_table[8]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
                                    //计时信息显示表
unsigned char code
date[]={0x5b,0x3f,0x06,0x6f,0x40,0x3f,0x06,0x40,0x06,0x5b,0x00,0x00};
                                    //日期信息表

int num=0;                          //num 满 20 为 1s，用于更新时间
int hour=0,min=0,sec=0;             //计时变量

void Delaymini();                   //子函数声明
void timer0();
void twinkle(int sel,int n);
void display();
void delay(unsigned char  p);
unsigned char getkeycode();
void reset();
void scan();

/**********************************主函数**********************************/
void  main()
{
    TMOD=0x01;      //模式控制 方式 1（16 位）
    TH0= 0x3C;      //初始值 3CB0：15536   终值 65535   计数值 50000    计数周期
0.05s
    TL0 = 0xB0;
    TR0=1;          //启动定时器工作
    EA=1;           //CPU 中断允许位
    ET0=1;          //T0 中断允许位
    while(1)        //TF0=1:溢出标志位
    {
        switch(getkeycode())
        {
            case 0x84:reset();break;              //第三行第 4 列
            case 0x88:scan();break;               //第四行第 4 列
            default: break;
```

```
        }
        display();
    }
}

/**********************************延时函数**********************************/
void Delaymini()
{
    unsigned char i,j;
    for(i=1;i>0;i--)
        for(j=30;j>0;j--);
}

void delay(unsigned char  p)
{
    unsigned char  i,j;
    for(;p>0;p--)
        for(i=181;i>0;i--)
            for(j=181;j>0;j--);
}

/**********************************定时器中断**********************************/
void timer0() interrupt 1
{
    TMOD=0x01;
    TH0=0x3C;
    TL0=0xB0;
    num++;

    if(num==20)
    {
        num=0;
        sec++;
            if(sec==60)
            {
                sec=0;
                min++;
                if(min==60)
                {
                    min=0;
                    hour++;
                    if(hour==24)
                        hour=0;
                }
```

```
            }
    }
    show_table[0]=seg_table[hour/10];
    show_table[1]=seg_table[hour%10];
    show_table[2]=0x40;
    show_table[3]=seg_table[min/10];
    show_table[4]=seg_table[min%10];
    show_table[5]=0x40;
    show_table[6]=seg_table[sec/10];
    show_table[7]=seg_table[sec%10];
}

/**********************************闪烁与显示**********************************/
void twinkle(sel,n)
{
    int i;
    for(i=0;i<8;i++)
    {
        XBYTE[0x8000]=  0x01<<i;
        if(i==sel)
        {
            if(n%600<300)
                XBYTE[0x9000]= 0x00;
            else
                XBYTE[0x9000]= show_table[7-i];
        }
        else
            XBYTE[0x9000]= show_table[7-i];
        Delaymini();
        XBYTE[0x9000]= 0x00; //消隐
    }
}

void display()
{
    int i;
    for(i=0;i<8;i++)
    {
        XBYTE[0x8000]=  0x01<<i;
        XBYTE[0x9000]= show_table[7-i];
        Delaymini();
        XBYTE[0x9000]= 0x00; //消隐
    }
}
```

```
/*********************************按键扫描*************************************/
unsigned char getkeycode()                  //键盘扫描函数，返回获得键码
{
    unsigned char line=0x00;                //行码
    unsigned char col=0x00;                 //列码
    unsigned char scancode=0x01;            //行扫描码
    unsigned char keycode;                  //键号

    XBYTE[0x8000]=0xff;
    col=XBYTE[0x8000]&0x0f;                 //从列端口读入四位列码
    if (col==0x00)
        keycode=0x00;
    else
    {
        while((scancode&0x0f)!=0)           //取 scancode 的低四位，没变为全 0，
循环
        {
            line=scancode;                  //行号
            XBYTE[0x8000]=scancode;         //给行赋扫描码，第一行为 0x01
            if((XBYTE[0x8000]&0x0f)==col)   //检测按键所在的行跳出循环
                break;
            scancode=scancode<<1;           //行扫描码左移一位，转下一行
        }
        col=col<<4;                         //把列码移到高四位
        keycode=col|line;
    }
    return keycode;
}

/*********************************校准函数*************************************/
void reset()
{
    int i,n,m1=0,n1=0,m2=0,n2=0,m3=0,n3=0;
    int flag1=0,flag2=0,flag3=0,flag4=0,flag5=0,flag6=0;
    while(1)              //时校准
    {
        if(flag1==1)
        {
            n=0;
            break;
        }
        if(n==6000)      //定期清零，防止 n 过大而溢出
            n=0;
```

```
    twinkle(7,n);
    switch(getkeycode())
    {
        case 0x11:m1=1;flag1=1;break;              //第一行第1列
        case 0x21:m1=2;flag1=1;break;              //第一行第2列
        case 0x24:m1=0;flag1=1;break;              //第三行第2列
        default:break;
    }
    n++;
}
hour=m1*10+(hour%10);
for(i=0;i<1000;i++)
    display();

while(1)
{
    if(flag2==1)
    {
        n=0;
        break;
    }
    if(n==6000)
        n=0;
    twinkle(6,n);
    switch(getkeycode())
    {
        case 0x11:n1=1;flag2=1;break;              //第一行第1列
        case 0x21:n1=2;flag2=1;break;              //第一行第2列
        case 0x41:n1=3;flag2=1;break;              //第一行第3列
        case 0x81:n1=4;flag2=1;break;              //第一行第4列

        case 0x12:n1=5;flag2=1;break;              //第二行第1列
        case 0x22:n1=6;flag2=1;break;              //第二行第2列
        case 0x42:n1=7;flag2=1;break;              //第二行第3列
        case 0x82:n1=8;flag2=1;break;              //第二行第4列

        case 0x14:n1=9;flag2=1;break;              //第三行第1列
        case 0x24:n1=0;flag2=1;break;              //第三行第2列
        default:break;
    }
    n++;
}
hour=(hour/10)*10+n1;
for(i=0;i<1000;i++)
```

```
    display();

while(1)            //分校准
{
    if(flag3==1)
    {
        n=0;
        break;
    }
    if(n==6000)
        n=0;
    twinkle(4,n);
    switch(getkeycode())
    {
        case 0x11:m2=1;flag3=1;break;              //第一行第 1 列
        case 0x21:m2=2;flag3=1;break;              //第一行第 2 列
        case 0x41:m2=3;flag3=1;break;              //第一行第 3 列
        case 0x81:m2=4;flag3=1;break;              //第一行第 4 列
        case 0x12:m2=5;flag3=1;break;              //第二行第 1 列
        case 0x24:m2=0;flag3=1;break;              //第三行第 2 列
        default:break;
    }
    n++;
}
min=m2*10+(min%10);
for(i=0;i<1000;i++)
    display();

while(1)
{
    if(flag4==1)
    {
        n=0;
        break;
    }
    if(n==6000)
        n=0;
    twinkle(3,n);
    switch(getkeycode())
    {
        case 0x11:n2=1;flag4=1;break;                //第一行第 1 列
        case 0x21:n2=2;flag4=1;break;                //第一行第 2 列
        case 0x41:n2=3;flag4=1;break;                //第一行第 3 列
        case 0x81:n2=4;flag4=1;break;                //第一行第 4 列
```

```
        case 0x12:n2=5;flag4=1;break;                //第二行第 1 列
        case 0x22:n2=6;flag4=1;break;                //第二行第 2 列
        case 0x42:n2=7;flag4=1;break;                //第二行第 3 列
        case 0x82:n2=8;flag4=1;break;                //第二行第 4 列

        case 0x14:n2=9;flag4=1;break;                //第三行第 1 列
        case 0x24:n2=0;flag4=1;break;                //第三行第 2 列
        default:break;
    }
    n++;
}
min=(min/10)*10+n2;
for(i=0;i<1000;i++)
    display();

while(1)              //秒校准
{
    if(flag5==1)
    {
        n=0;
        break;
    }
    if(n==6000)
        n=0;
    twinkle(1,n);
    switch(getkeycode())
    {
        case 0x11:m3=1;flag5=1;break;                //第一行第 1 列
        case 0x21:m3=2;flag5=1;break;                //第一行第 2 列
        case 0x41:m3=3;flag5=1;break;                //第一行第 3 列
        case 0x81:m3=4;flag5=1;break;                //第一行第 4 列

        case 0x12:m3=5;flag5=1;break;                //第二行第 1 列
        case 0x24:m3=0;flag5=1;break;                //第三行第 2 列
        default:break;
    }
    n++;
}
sec=m3*10+(sec%10);
for(i=0;i<1000;i++)
    display();

while(1)
```

```
    {
        if(flag6==1)
        {
            n=0;
            break;
        }
        if(n==6000)
            n=0;
        twinkle(0,n);
        switch(getkeycode())
        {
            case 0x11:n3=1;flag6=1;break;                //第一行第 1 列
            case 0x21:n3=2;flag6=1;break;                //第一行第 2 列
            case 0x41:n3=3;flag6=1;break;                //第一行第 3 列
            case 0x81:n3=4;flag6=1;break;                //第一行第 4 列

            case 0x12:n3=5;flag6=1;break;                //第二行第 1 列
            case 0x22:n3=6;flag6=1;break;                //第二行第 2 列
            case 0x42:n3=7;flag6=1;break;                //第二行第 3 列
            case 0x82:n3=8;flag6=1;break;                //第二行第 4 列

            case 0x14:n3=9;flag6=1;break;                //第三行第 1 列
            case 0x24:n3=0;flag6=1;break;                //第三行第 2 列
            default:break;
        }
        n++;
    }
    sec=(sec/10)*10+n3;
    num=0;                  //将之前的秒计时清零，从校准结束时刻算起
    for(i=0;i<1000;i++)
        display();
}

/*********************************日期滚动显示*********************************/
void scan(void)
{
    int i,m,k=0;
    for(k=0;k<36;k++)
        for(m=0;m<200;m++)
            for(i=0;i<8;i++)
            {
                XBYTE[0x8000]=0x01<<i;
                switch(i)
                {
```

```
                    case(0):if((k-i)<0) XBYTE[0x9000]= 0x00; else XBYTE[0x9000]=
date[(k+i)%12];Delaymini();break;
                    case(1):if((k-i)<0) XBYTE[0x9000]= 0x00; else XBYTE[0x9000]=
date[(k+i-2)%12];Delaymini();break;
                    case(2):if((k-i)<0) XBYTE[0x9000]= 0x00; else XBYTE[0x9000]=
date[(k+i-4)%12];Delaymini();break;
                    case(3):if((k-i)<0) XBYTE[0x9000]= 0x00; else XBYTE[0x9000]=
date[(k+i-6)%12];Delaymini();break;
                    case(4):if((k-i)<0) XBYTE[0x9000]= 0x00; else XBYTE[0x9000]=
date[(k+i-8)%12];Delaymini();break;
                    case(5):if((k-i)<0) XBYTE[0x9000]= 0x00; else XBYTE[0x9000]=
date[(k+i-10)%12];Delaymini();break;
                    case(6):if((k-i)<0) XBYTE[0x9000]= 0x00; else XBYTE[0x9000]=
date[(k+i-12)%12];Delaymini();break;
                    case(7):if((k-i)<0) XBYTE[0x9000]= 0x00; else XBYTE[0x9000]=
date[(k+i-14)%12];Delaymini();break;
                }
                XBYTE[0x9000]= 0x00;
            }
}
```