

**《开放式单片机应用系统专题设计》**

**专题名称：电脑时钟**

**班级：集卓2001**

**姓名：薛谦**

**学号：U202014228**

**指导教师：吴裕斌，刘冬生**

**光学与电子信息学院**

**2022.6**

一、任务要求

利用实验板资源设计一个电脑时钟

【基本要求】：

1.显示模式：

年月日，时间，星期信息同屏显示

2.可设置2个以上闹钟，具体要求如下

屏幕要显示闹钟已设置标识

可以设定闹钟日期

可以设定闹铃时长、间隔和闹铃次数

闹点到时用声光提示

3. 整点声光提示。

二、功能特点与使用说明

**（一）功能特点**

具有十分全面的闹钟功能设计，用户交互友好清晰，通过键盘输入，LCD屏幕提示各个按键的作用。 除了可以切换显示方式的显示模式外，提供闹钟功能，倒计时功能和整点报时功能，具有良好的交互性。同时，用户可自行设定时间与日期，有较高的自由度。

**（二）使用说明**

上电后，程序自动进入显示模式，默认显示方式为24小时制。在LCD屏的第一行左侧显示时间，在第二行左侧显示日期与当前的星期几。在第一行右侧从左到右依次为当前是上午还是下午的标识，当前已开启闹钟数量，当前是否开启了整点报时功能。

用户可通过按钮K8或0进入分别进入两个功能表。当按下K8后，程序进入功能表1，此时可以通过按键，按照显示在LCD屏幕上的提示，选择进入：闹钟设置；时间设置；整点报时；设置闹钟状态四种功能。在功能表1界面按下按键F2，或在显示模式按下按键0，进入功能表2，此时可选择进入显示方式切换；倒计时设置；倒计时状态；设置日期四种功能。

在进入功能表1后，按下按键1，进入闹钟设置流程。系统共提供5个可设置时钟，每个时钟可按照LCD提示，依次设置是否按照日期闹铃，每周的哪几天闹铃，是否打开声光提示，声光提示频率，闹铃时长、间隔和闹铃次数，最后设置闹铃时间。按下按键2，进入设置闹钟状态，可通过选择对应闹钟序号打开或关闭对应闹钟。按下按键3，进入整点报时功能，按照提示选择打开或者关闭整点报时功能，按下按键4，进入时间设置功能，按照提示设置时间，按下F8确认后将按照当前设定时间显示。在设置完成后，系统默认返回功能表1页面。如要返回则再次按下F8。

在进入功能表2后，按下按键5，进入设置倒计时功能模式，按照提示设置倒计时的时长，按下F8后倒计时将立刻开始。按下按键6后，进入倒计时状态检查模式，可查询当前进行中的倒计时剩余时间，若按下F1，当前倒计时关闭，按下F8，退出当前功能。按下按键7后，进入日期设置模式，可按照提示设置日期，系统将在设置完成后自动计算对应星期。按下按键8后，系统进入12/24显示方式切换，按下F1选择12小时制，按下F2选择24小时制。在设置完成后，系统默认返回功能表2页面。如要返回则再次按下F8。

在闹铃，整点报时，倒计时响应上，系统设置的优先级为：

整点报时>倒计时>闹铃1>闹铃2>闹铃3>闹铃4>闹铃5，同一分钟内只允许一次响应。

闹铃时，按下按键=直到两个灯全亮可关闭本次闹铃，按下F7可关闭剩余闹铃。整点报时时长10s后自动关闭，倒计时响应10s，期间可通过按下F7关闭。

三、方案设计

**（一）设计思路**

该程序采用传统的状态机与中断立FLAG - 主循环查询的模式。主循环中通过状态不同进行不同的FLAG循环查询来调用不同的函数。用户的键入在定时器的中断服务函数中以定时扫描的形式进行读取。键入信号是驱动本程序改变状态的根本信号以及各类函数的起始输入信号。

主要的函数模块有：1.显示模式函数。2.闹钟设置函数。3.倒计时设置函数。4.整点报时设置函数。5.响铃响应函数。6.时间及日期设置函数。7. LCD显示模块。8.键盘扫描函数。下面分函数简述

1．**显示模式函数：**

用定时器1负责计数，设置每次定时1ms，每1000次秒标志位为1，秒数加1，每60秒分钟标志位为1，分钟数加1，每60分钟，小时数加1，每24小时日标志位为1，日加1。主程序检查日标志位，通过日期检查函数确定当前的年月日和星期。

2．**闹钟设置函数：**

通过对闹钟每一个信息分别设置对应的数组来存储相应的特征。在闹钟设置函数中，通过显示屏提示，先确定要使用的是哪个闹钟，即数组中第几个元素的数据，随后依次设置是否按照日期闹铃，每周的哪几天闹铃，是否打开声光提示，声光提示频率，闹铃时长、间隔和闹铃次数，最后设置闹铃时间。

3. **倒计时设置函数：**

在定时器1中设置两个变量分别存放当前时间的运行时长，在倒计时中设置的倒计时时长加上设置时的运行时间，即为倒计时目标时间，存入对应变量中。

4. **整点报时设置函数：**

设置对应标志位，标志位为1则检查当前时间是否为整点，是的话就报时。否则不检查。

5.**响应函数：**

依次检查整点报时，倒计时，闹钟的标志位，按照相应设置响应。其中闹钟的标志位用定时器4扫描给出。

6. **时间及日期设置函数：**

对响应变量的修改操作，不赘述。

7. **LCD显示模块：**

按照手册中的时序图和相关指令说明使用即可，不赘述。

8.**键盘扫描函数：**

用定时0每20ms扫描一次。采用软件延时去抖。按键扫描时通过对74HCT74操作依次进行行扫描，按照芯片的时序使用即可。

四、流程图 （主程序、子程序的详细流程）



图1 主控循环流程图



图2 设置整点报时流程图



图3闹钟设置流程图



图4倒计时设置流程图



图6 响应判断流程图

五、源程序 （含文件头说明、资源使用说明、语句行注释）

见附录。

六、程序测试方法与结果

（一）程序测试方法

1.关闭闹铃

2.设置一个闹铃（时间为13：59；响铃间隔4次，每次间隔1min，响铃时长30s，开声光提示）

3.打开整点闹铃

4.打开一个25s的倒计时

5.设置时间为13：58

6.第3个闹钟时使用消除重复按键结束闹铃

7.设置时间为12小时制。

（二）实验现象

13：59：19倒计时正常运行，按下F7后关闭。

13：59，闹铃按照设置响起，按下按键=关闭

14：00，整点报时响起，闹钟无响应

14：01，闹钟再次响铃，按下F7关闭

14：02，闹钟未响应，设置为12小时显示，显示无问题。

（三）实验结果

实验程序执行正常。

七、心得与体会

如果说这次专题设计难度最大的困难在哪里，那应该就是在对用户行为的对策上，一旦试图去思考用户可能做出的种种随意的设置及其可能带来的种种影响，实在是让人胃痛不已。但也是在这不断对程序的完善过程中，加强了我对程序结构的整体性管理。

同时，本次设计让我十分清晰地感受到，在长时间使用C语言编程以来，当前在我的编程思维里，面向过程的编程思维的确是占据了相当程度的强势地位。而这种感受暴露出来后，让我第一次十分强烈的感受到在做具有一定交互的工程时，面向用户的语言将会在效率和编写难度上远远甩开C语言。在接下来的大学时间里掌握一门以上的面向对象的编程语言想必还是很有利于编程思维的全面发展的。

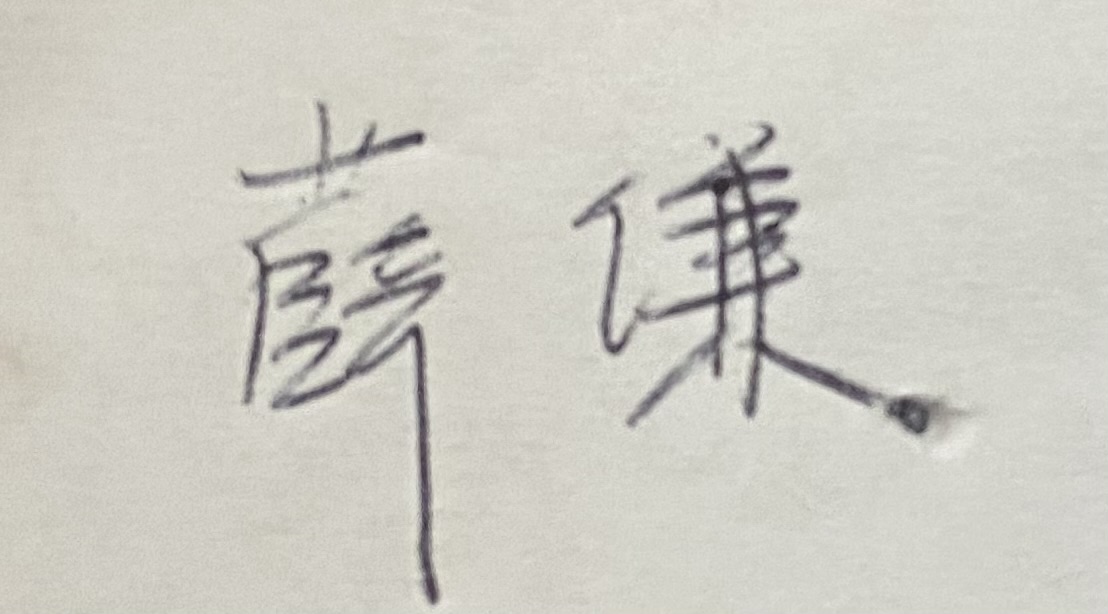
由于在此之前一直在使用功能强大很多的串口屏，因此在本次实验中使用LCD1602颇为感到束手束脚。然而话说回来，1602的时序简单，实际操作还方便不少。但由于屏幕容量有限，所以不得不在状态转移，甚至一个状态中反复清屏、切换显示界面、循环改变显示变量，中间标志位的设置与处理十分的繁琐。调试的过程实在是算不上心平气和。但在这些交互调试到位之后，协助我进行测试的舍友表示交互性还是很人性化的，可读性也挺不错。这也让我意识到，在做产品时，只要能够满足甲方的需求，达到客户的标准，那还是尽可能的选择更高预算的选择，否则最后吃最多的苦的人大概率会是我自己。

当然，完成这个实验内心还是十分开心，虽然工程比较简单，但是能尽可能把每个细节考虑到，同时考虑一些人的因素，也算没有浪费两周的时间。

感谢帮助我测试的舍友，帮我发现了茫茫多的我自己写代码时完全没料到的逻辑错误，以及某某在我代码写的暴躁的时候让我冷静下来。

最后，十分感谢各位老师和助教的耐心解答以及费心地准备这一次专题实验。

设计人签名：



**附录：源码**

**User**

**main.c**

#include "main.h"

void main()

{

    Sys\_Init();

    Timer0\_init();

    Timer1\_init();

    Timer4\_init();

    while(1)

    {

        if(checkday\_flag == 1)

        {

            checkday\_flag = 0;

            checkday();

            time\_show();

        }

        if(Flag1 == 1)

        {

            time\_show();

            Flag1 = 0;

        }

        if(Flag3 == 1)

        {

            time\_showplus();    //进入显示模式

            Flag3 = 0;

        }

        func\_choose();

        point\_ring0();

        if\_count();

    }

}

**main.h**

#ifndef \_MAIN\_H\_

#define \_MAIN\_H\_

//首先是C的一些库和这个片子的

#include "stdio.h"

#include "stdlib.h"

#include "string.h"

#include "math.h"

#include "C8051F020.h"

#include "lcd.h"

#include "delay.h"

#include "key.h"

#include "buzzer.h"

#include "timer.h"

#include "systeminit.h"

#include "display.h"

#include "led.h"

#include "clock.h"

#include "ring.h"

#include "countdown.h"

#define uint        unsigned int

#define uchar       unsigned char

**clock.c**

#include "main.h"

int num\_clc = 0;

int quantity = 1;

int this\_moment = 0;

char \*fre0[] = {" NOT SET","Very Low Fre","Low Fre      ",

                "Middle Fre     ","High Fre     "

                ,"Very High Fre"};

int open[5] = {1,1,0,0,0};       //是否开启了闹钟

int date\_whe[5] = {1,0,0,0,0};//是否按照日期设置

int date\_d[5] = {1,1,1,1,1} ;  //设置的日期

int  clc\_week[5][7] = {{1,1,1,1,1,1,1},

                      {1,1,1,1,1,1,1},

                      {1,1,1,1,1,1,1},

                      {1,1,1,1,1,1,1},

                      {1,1,1,1,1,1,1}};    //设置每周哪几天闹铃

int length[5] = {40,30,30,30,30} ;     //响铃时间时长

int inter[5]  = {1,5,5,5,5};   //响铃间隔时长

int repeat[5] = {1,0,0,0,0}; //是否重复,0为重复0次,最多5次

int fre[5] = {4,1,1,1,1} ;    //声光频率

int light[5] = {1,1,1,1,1}; //光

int sound[5] ={0,1,1,1,1} ;  //声

int clc\_min[5] = {0,0,0,0,0}; //分钟

int clc\_hour[5] = {0,0,0,0,0};  //小时

// CLC clock;本来打算用结构体，然后不知道为什么复用性不太好，干脆直接用数组还好控制一点，结构体数组终究是还没彻底搞成啊，技术不够，惭愧惭愧。

// void clock\_init()

// {

//   int i = 0;

//   for(i = 0;i<=4;++i)

//   {

//      open[i] = 0;

//      date\_d[i] = 10;

//      date\_whe[i]=0;

//      fre[i] = 1;

//       clc\_week[i][0] = 1;

//       clc\_week[i][1] = 1;

//       clc\_week[i][2] = 1;

//       clc\_week[i][3] = 1;

//       clc\_week[i][4] = 1;

//       clc\_week[i][5] = 1;

//       clc\_week[i][6] = 1;

//

//      inter[i] = 10;

//      length[i] = 10;

//      light[i] = 0;

//      repeat[i] = 3;

//      sound[i] = 0;

//

//   }

////    inter[0] = 10;

////    inter[1] = 10;

////    inter[2] = 10;

////    inter[3] = 10;

////    inter[4] = 10;

//

// }

void pass()

{

  while(1)

  {

    if(key == 18)

    break;

  }

}

void func\_choose()

{

  if(key == 19)

  {

    fun\_init();

    while( 1 )

    {

      set\_if();

      count\_clo();

      if\_date0();

      changeshow();

      If\_set();

      close\_clc();

      point();

      if\_ctl\_time();

      if(key == 19)

      {

        break;

      }

      if(key == 13)

      {

        fun\_re();

      }

      if(key == 12)

      {

        count\_re();

        Delay\_us(500);

      }

    }

    LCD\_Init();

    Delay\_ms(10);

    time\_show();

  }

}

void if\_date0()

{

  if(key == 7)

  {

    // LCD\_Init();

    // Delay\_ms(10);

    // LCD\_ShowString(1,1," F7 TO Quit ");

    set\_date0();

    count\_re();

  }

}

void set\_date0()

{

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"Set the Date");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F1 t0 INC date!");

  LCD\_ShowString(2,1,"F2 to DES date!");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F5 t0 INC Month!");

  LCD\_ShowString(2,1,"F6 to DES Month!");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F3 t0 INC YEAR!");

  LCD\_ShowString(2,1,"F4 to DES YEAR!");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F8 t0 set");

  LCD\_ShowString(2,1,"    -  -");

  LCD\_ShowNum(2,1,year,4);

  LCD\_ShowNum(2,6,month,2);

  LCD\_ShowNum(2,9,day,2);

  while(1)

  {

    if(key == 14)

    {

      year ++;

      if(year == 2200)

      {

        year--;

      }

      if((year %  4 == 0 && year % 100 != 0) || (year % 400 == 0) && (month == 2)&&(day == 28))

      {

        day = 29;

        LCD\_ShowNum(2,9,day,2);

      }

      LCD\_ShowNum(2,1,year,4);

      Delay\_us(500);

    }

    if(key == 15)

    {

      year --;

      if(year == 2000)

      {

        year++;

      }

      if((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0) && (month == 2)&&(day == 28))

      {

        if((month == 2)&&(day == 28))

        day = 29;

        LCD\_ShowNum(2,9,day,2);

      }

      else if((month == 2)&&(day == 29))

      {

        day = 28;

        LCD\_ShowNum(2,9,day,2);

      }

      LCD\_ShowNum(2,1,year,4);

      Delay\_us(500);

    }

    if(key == 16)

    {

      month++;

      if(month == 13)

      {

        month = 1;

      }

      if(day >= 31)

      {

        if((month == 4)||(month == 6)|| (month == 9) || (month == 11)||(month == 2))

        {

          day = 1;

        }

      }

      if(day == 30)

      {

        if ((month == 2))

        {

            day = 1;

        }

      }

      if(day == 29)

      {

        if((month == 2))

        {

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

          {

            day = 29;

          }

          else

          {

            day = 1;

          }

        }

      }

    LCD\_ShowNum(2,6,month,2);

    LCD\_ShowNum(2,9,day,2);

    Delay\_us(500);

    }

    if(key == 17)

    {

      month--;

      if(month == 0)

      {

        month = 12;

      }

      if(day >= 31)//很多很麻烦调整起来相当折磨的日期判断，甚至这里的逻辑问题在设置的时候比闹铃的时候还多。

      {

        if((month == 4)||(month == 6)|| (month == 9) || (month == 11)||(month == 2))

        {

          day = 1;

        }

      }

      if(day == 30)

      {

        if ((month == 2))

        {

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

          {

            day = 1;

          }

        }

      }

      if(day == 29)

      {

        if((month == 2))

        {

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

          {

            day = 29;

          }

          else

          {

            day = 1;

          }

        }

      }

    LCD\_ShowNum(2,6,month,2);

    LCD\_ShowNum(2,9,day,2);

    Delay\_us(500);

    }

    if(key == 12)

    {

      day++;

      if((month ==1)|| (month ==3) || (month == 5)||(month == 7)||(month == 8)||(month == 10))

      {

        if(day > 31)

        {

          day = 1;

          month++;

          LCD\_ShowNum(2,6,month,2);

        }

      }

      else  if((month == 4)||(month == 6)|| (month == 9) || (month == 11))

      {

        if(day > 30)

        {

          day = 1;

          month++;

          LCD\_ShowNum(2,6,month,2);

        }

      }

      else  if(month ==12)

      {

        if(day > 31)

        {

          day = 1;

          month = 1;

          LCD\_ShowNum(2,6,month,2);

        }

      }

      else

      {

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

        {

          if(day >29)

          {

            day = 1;

            month++;

            LCD\_ShowNum(2,6,month,2);

          }

        }

        else

        {

          if(day > 28)

          {

            day = 1;

            month++;

            LCD\_ShowNum(2,6,month,2);

          }

        }

      }

      LCD\_ShowNum(2,9,day,2);

      Delay\_us(500);

    }

    if(key == 13)

    {

      day --;

      if(day == 0)

      {

        if((month ==12) || (month == 5)||(month == 7)||(month == 10))

        {

            day = 30;

            month--;

        }

        else  if((month == 4)|| (month ==2)||(month == 6)||(month == 8)|| (month == 9) || (month == 11))

        {

          month--;

          day = 31;

        }

        else  if(month ==1)

        {

            day = 31;

            month = 12;

        }

        else

        {

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

          {

              day = 29;

              month = 2;

          }

          else

          {

              day = 28;

              month = 2;

          }

        }

      }

      LCD\_ShowNum(2,9,day,2);

      LCD\_ShowNum(2,6,month,2);

      Delay\_us(500);

    }

    if(key == 19)

    {

        break;

    }

  }

  checkday();

}

void if\_ctl\_time()

{

  if(key == 4)

  {

    TR1 = 0;

    ctl\_time();

    Congratulation1();

    fun\_re();

    TR1 = 1;

  }

}

void ctl\_time()

{

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"Set the Time");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F3 t0 INC MIN!");

  LCD\_ShowString(2,1,"F4 to DES MIN!");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F5 t0 INC HOUR!");

  LCD\_ShowString(2,1,"F6 to DES HOUR!");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F8 t0 set");

  LCD\_ShowString(2,7," : ");

  LCD\_ShowNum(2,5, hour,2);

  LCD\_ShowNum(2,10, min,2);

  while(1)

  {

      if(key == 14)

      {

        min +=1;

        if(min >= 60)

        {

          min = 0;

        }

        LCD\_ShowNum(2,10, min,2);

        Delay\_us(500);

      }

      if(key == 15)

      {

        if(min == 0 )

        {

          min = 59;

        }

        else

        {

          min -=1;

        }

        LCD\_ShowNum(2,10, min,2);

        Delay\_us(500);

      }

      if(key == 16)

      {

        hour++;

        if(hour >=24)

        {

          hour = 0;

        }

        LCD\_ShowNum(2,5, hour,2);

        Delay\_us(500);

      }

      if(key == 17)

      {

        if(hour == 0 )

        {

          hour = 23;

        }

        else

        {

          hour--;

        }

        LCD\_ShowNum(2,5, hour,2);

        Delay\_us(500);

      }

      if(key == 19)

      {

          break;

      }

  }

}

void close\_clc()

{

  if(key == 2)

  {

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"Which To Close");

    LCD\_ShowString(2,1,"F8 To Quit");

    pass();

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"1 2 3 4 5");

    LCD\_ShowNum(2,1,open[0],1);

    LCD\_ShowNum(2,3,open[1],1);

    LCD\_ShowNum(2,5,open[2],1);

    LCD\_ShowNum(2,7,open[3],1);

    LCD\_ShowNum(2,9,open[4],1);

    while(1)

    {

      if(key == 1)

      {

        if(open[0] == 1)

        {

          open[0] = 0;

          quantity--;

        }

        else if(open[0] == 0)

        {

          open[0] = 1;

          quantity++;

        }

        LCD\_ShowNum(2,1,open[0],1);

        Delay\_ms(20);

      }

      if(key == 2)

      {

        if(open[1] == 1)

        {

          open[1] = 0;

          quantity--;

        }

        else if(open[1] == 0)

        {

          open[1] = 1;

          quantity++;

        }

        LCD\_ShowNum(2,3,open[1],1);

        Delay\_ms(20);

      }

      if(key == 3)

      {

        if(open[2] == 1)

        {

          open[2] = 0;

          quantity--;

        }

        else if(open[2] == 0)

        {

          open[2] = 1;

          quantity++;

        }

        LCD\_ShowNum(2,5,open[2],1);

        Delay\_ms(20);

      }

      if(key == 4)

      {

        if(open[3] == 1)

        {

          open[3] = 0;

          quantity--;

        }

        else if(open[3] == 0)

        {

          open[3] = 1;

          quantity++;

        }

        LCD\_ShowNum(2,7,open[3],1);

        Delay\_ms(20);

      }

      if(key == 5)

      {

        if(open[4] == 1)

        {

          open[4] = 0;

          quantity--;

        }

        else if(open[4] == 0)

        {

          open[4] = 1;

          quantity++;

        }

        LCD\_ShowNum(2,9,open[4],1);

        Delay\_ms(20);

      }

      if(key == 19)

      {

        break;

      }

    }

    fun\_re();

  }

}

void If\_set()

{

  if(key == 1)

  {

    T4CON &= 0xFB;

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"Whether to Set");

    pass();

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"F1 For Yes!");

    Delay\_ms(10);

    LCD\_ShowString(2,1,"F2 For No !");

    Delay\_ms(10);

    while(1)

    {

        if(key==12)

        {

            Whi\_ch();

            Congratulation();

            break;

        }

        if(key == 13)

        {

          break;

        }

    }

    fun\_re();

    T4CON |= 0x04;

  }

}

void Whi\_ch()

{

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"Which (1-5)");

    Delay\_ms(10);

    LCD\_ShowString(2,1,"Make the choice");

    Delay\_ms(10);

    while(1)

    {

      if(key == 1)

      {

          num\_clc = 0;

          break;

      }

      if(key == 2)

      {

          num\_clc = 1;

          break;

      }

      if(key == 3)

      {

          num\_clc = 2;

          break;

      }

      if(key == 4)

      {

          num\_clc = 3;

          break;

      }

      if(key == 5)

      {

          num\_clc = 4;

          break;

      }

    }

    open[num\_clc] = 1;

    quantity++;

    If\_date();

}

void If\_date()

{

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"Obey  date?");

    pass();

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"F1 For Yes!");

    Delay\_ms(10);

    LCD\_ShowString(2,1,"F2 For No !");

    Delay\_ms(10);

    while(1)

    {

        if(key==12)

        {

          date\_whe[num\_clc] = 1;

          set\_date();

          break;

        }

        if(key == 13)

        {

           date\_whe[num\_clc] = 0;

          set\_week();

          break;

        }

    }

}

void set\_date()

{

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"Set the Date");

    pass();

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"F3 t0 INC!");

    LCD\_ShowString(2,1,"F4 to DES!");

    pass();

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"F8 t0 set");

    LCD\_ShowString(2,1,"Date:");

  while(1)

  {

    LCD\_ShowNum(2,6, date\_d[num\_clc],2);

    Delay\_us(50);

    if(key == 14)

    {

      date\_d[num\_clc] +=1;

      if(  date\_d[num\_clc] > 31)

      {

        date\_d[num\_clc] = 1;

      }

    }

    if(key == 15)

    {

      if(date\_d[num\_clc] ==0)

      {

        date\_d[num\_clc] = 31;

      }

      else

      {

        date\_d[num\_clc]-=1;

      }

    }

     if(key == 19)

    {

        break;

    }

  }

  set\_week();

}

void set\_week()

{

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"Choose the day");

  Delay\_ms(10);

  LCD\_ShowString(2,1,"1-OPEN 0-CLOSE");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"1 2 3 4 5 6 7");

  Delay\_ms(10);

  LCD\_ShowString(2,1,"1 1 1 1 1 1 1");

  while(1)//按下对应的按键即可选择对应的周几到底是开不开闹铃

  {

    if(key == 1)

    {

      if(  clc\_week[num\_clc][1] == 0)

      {

          clc\_week[num\_clc][1] = 1;

      }

      else if(  clc\_week[num\_clc][1] == 1)

      {

          clc\_week[num\_clc][1] = 0;

      }

      Delay\_us(500);

      LCD\_ShowNum(2,1,  clc\_week[num\_clc][1],1);

    }

    if(key == 2)

    {

      if(  clc\_week[num\_clc][2] == 0)

      {

          clc\_week[num\_clc][2] = 1;

      }

      else if(  clc\_week[num\_clc][2] == 1)

      {

          clc\_week[num\_clc][2] = 0;

      }

      Delay\_us(500);

      LCD\_ShowNum(2,3,  clc\_week[num\_clc][2],1);

    }

    if(key == 3)

    {

      if(  clc\_week[num\_clc][3] == 0)

      {

          clc\_week[num\_clc][3] = 1;

      }

      else if(  clc\_week[num\_clc][3] == 1)

      {

          clc\_week[num\_clc][3] = 0;

      }

      Delay\_us(500);

      LCD\_ShowNum(2,5,  clc\_week[num\_clc][3],1);

    }

    if(key == 4)

    {

      if(  clc\_week[num\_clc][4] == 0)

      {

          clc\_week[num\_clc][4] = 1;

      }

      else if(  clc\_week[num\_clc][4] == 1)

      {

          clc\_week[num\_clc][4] = 0;

      }

      Delay\_us(500);

      LCD\_ShowNum(2,7,  clc\_week[num\_clc][4],1);

    }

    if(key == 5)

    {

      if(  clc\_week[num\_clc][5] == 0)

      {

          clc\_week[num\_clc][5] = 1;

      }

      else if(  clc\_week[num\_clc][5] == 1)

      {

          clc\_week[num\_clc][5] = 0;

      }

      Delay\_us(500);

      LCD\_ShowNum(2,9,  clc\_week[num\_clc][5],1);

    }

    if(key == 6)

    {

      if(  clc\_week[num\_clc][6] == 0)

      {

          clc\_week[num\_clc][6] = 1;

      }

      else if(  clc\_week[num\_clc][6] == 1)

      {

          clc\_week[num\_clc][6] = 0;

      }

      Delay\_us(500);

      LCD\_ShowNum(2,11,  clc\_week[num\_clc][6],1);

    }

    if(key == 7)

    {

      if(  clc\_week[num\_clc][7] == 0)

      {

          clc\_week[num\_clc][7] = 1;

      }

      else if(  clc\_week[num\_clc][7] == 1)

      {

          clc\_week[num\_clc][7] = 0;

      }

      Delay\_us(500);

      LCD\_ShowNum(2,13,  clc\_week[num\_clc][7],1);

    }

     if(key == 19)

    {

      break;

    }

  }

  if((clc\_week[num\_clc][7] == 0)&&

  (clc\_week[num\_clc][6] == 0)&&

  (clc\_week[num\_clc][5] == 0)&&

  (clc\_week[num\_clc][4] == 0)&&

  (clc\_week[num\_clc][3] == 0)&&

  (clc\_week[num\_clc][2] == 0)&&

  (clc\_week[num\_clc][1] == 0)&&

  (clc\_week[num\_clc][0] == 0)

  )

  {

    open[num\_clc] = 0;

  }

  set\_length();

}

void set\_length()

{

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"Set the length");

    pass();

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"F3 t0 INC!");

    LCD\_ShowString(2,1,"F4 to DES!");

    pass();

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"F8 t0 set");

    LCD\_ShowString(2,6,"s");

    while(1)

    {

        LCD\_ShowNum(2,3, length[num\_clc],3);

        if(key == 14)

        {

          if(  length[num\_clc] < 295)

          {

              length[num\_clc] +=10;

          }

        }

        if(key == 15)

        {

          if(  length[num\_clc] > 10)

          {

              length[num\_clc] -=10;

          }

        }

         if(key == 19)

        {

            break;

        }

    }

    set\_inter();

}

void set\_inter()

{

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"Set the inter");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F3 t0 INC!");

  LCD\_ShowString(2,1,"F4 to DES!");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F8 t0 set");

  LCD\_ShowString(2,6,"min");

  inter[num\_clc] = (length[num\_clc] / 60) +5;

  while(1)

  {

      LCD\_ShowNum(2,3, inter[num\_clc],2);

      Delay\_us(50);

      if(key == 14)

      {

        if(  inter[num\_clc] < 31)

        {

              inter[num\_clc] +=1;

        }

      }

      if(key == 15)

      {

        if( ( inter[num\_clc] > ((length[num\_clc] / 60)+1) )

          && ( inter[num\_clc] != 1))

        {

              inter[num\_clc] -=1;

        }

      }

        if(key == 19)

      {

          break;

      }

  }

  set\_repeat();

}

void set\_repeat()

{

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"Set the repeat");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F3 t0 INC!");

  LCD\_ShowString(2,1,"F4 to DES!");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F8 t0 set");

  LCD\_ShowString(2,6,"times");

  while(1)

  {

      LCD\_ShowNum(2,3, repeat[num\_clc],2);

      if(key == 14)

      {

        if(  repeat[num\_clc] < 5)

        {

              repeat[num\_clc] +=1;

        }

      }

      if(key == 15)

      {

        if(  repeat[num\_clc] >=1 )

        {

              repeat[num\_clc] -=1;

        }

      }

       if(key == 19)

      {

          break;

      }

  }

  set\_remind();

}

void set\_light()

{

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"Whether to Set");

  Delay\_ms(10);

  LCD\_ShowString(2,1,"LIGHT");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F1 For Yes!");

  Delay\_ms(10);

  LCD\_ShowString(2,1,"F2 For No !");

  Delay\_ms(10);

  while(1)

  {

      if(key==12)

      {

         light[num\_clc] = 1;

        break;

      }

      if(key == 13)

      {

         light[num\_clc] = 0;

        break;

      }

  }

}

void set\_sound()

{

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"Whether to Set");

  Delay\_ms(10);

  LCD\_ShowString(2,1,"Sound");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F1 For Yes!");

  Delay\_ms(10);

  LCD\_ShowString(2,1,"F2 For No !");

  Delay\_ms(10);

  while(1)

  {

      if(key==12)

      {

        sound[num\_clc] = 1;

        break;

      }

      if(key == 13)

      {

        sound[num\_clc] = 0;

        break;

      }

  }

}

void set\_remind()

{

  set\_light();

  set\_sound();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"Set the frequency");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F3 t0 INC!");

  LCD\_ShowString(2,1,"F4 to DES!");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F8 t0 set");

  LCD\_ShowString(2,6,"min");

  while(1)

  {

      LCD\_ShowString(2,1,fre0[fre[num\_clc]]);

      if(key == 14)

      {

        if(  fre[num\_clc] < 5)

        {

              fre[num\_clc] +=1;

        }

      }

      if(key == 15)

      {

        if(  fre[num\_clc] >1 )

        {

            fre[num\_clc] -=1;

        }

      }

       if(key == 19)

      {

        break;

      }

  }

  set\_time();

}

void set\_time()

{

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"Set the Time");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F3 t0 INC MIN!");

  LCD\_ShowString(2,1,"F4 to DES MIN!");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F5 t0 INC HOUR!");

  LCD\_ShowString(2,1,"F6 to DES HOUR!");

  pass();

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"F8 t0 set");

  LCD\_ShowString(2,7," : ");

  LCD\_ShowNum(2,10, clc\_min[num\_clc],2);

  LCD\_ShowNum(2,5, clc\_hour[num\_clc],2);

  settime();

  final\_check();

}

void final\_check()

{

  if((clc\_min[num\_clc] == 0) && Flag2[0] && Flag2[1])

  {

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"Have Set the");

    LCD\_ShowString(2,1,"Whole Point");

    pass();

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"Please Reset the time");

    Delay\_ms(150);

    set\_time();

  }

}

void Congratulation()

{

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"CLOCK  IS SET");

  Delay\_ms(10);

  LCD\_ShowNum(1,6,(num\_clc+1),1);

  Delay\_ms(100);

}

void Congratulation1()

{

  second = 0;

  LCD\_Init();

  Delay\_ms(10);

  LCD\_ShowString(1,1,"Time  IS SET");

  Delay\_ms(100);

}

void settime()

{

  while(1)

  {

      if(key == 14)

      {

        clc\_min[num\_clc] +=1;

        if(clc\_min[num\_clc] >= 60)

        {

          clc\_min[num\_clc] = 0;

        }

        LCD\_ShowNum(2,10, clc\_min[num\_clc],2);

        Delay\_us(500);

      }

      if(key == 15)

      {

        if(clc\_min[num\_clc] == 0 )

        {

          clc\_min[num\_clc] = 59;

        }

        else

        {

           clc\_min[num\_clc] -=1;

        }

        LCD\_ShowNum(2,10, clc\_min[num\_clc],2);

        Delay\_us(500);

      }

      if(key == 16)

      {

        clc\_hour[num\_clc]++;

        if(clc\_hour[num\_clc] >=24)

        {

          clc\_hour[num\_clc] = 0;

        }

        LCD\_ShowNum(2,5, clc\_hour[num\_clc],2);

        Delay\_us(500);

      }

      if(key == 17)

      {

        if(clc\_hour[num\_clc] == 0 )

        {

          clc\_hour[num\_clc] = 23;

        }

        else

        {

          clc\_hour[num\_clc]--;

        }

        LCD\_ShowNum(2,5, clc\_hour[num\_clc],2);

        Delay\_us(500);

      }

      if(key == 19)

      {

          break;

      }

  }

}

void set\_point()

{

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"Open point ring?");

    pass();

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"F1 For OPEN!");

    Delay\_ms(10);

    LCD\_ShowString(2,1,"F2 For CLOSE!");

    Delay\_ms(10);

    while(1)

    {

        if(key==12)

        {

          LCD\_Init();

          Delay\_ms(10);

          LCD\_ShowString(1,1," Open success");

          Delay\_ms(100);

          // LCD\_Init();

          // Delay\_ms(10);

          // LCD\_SetCursor(1,16);

          // LCD\_WriteData(0xEF);

          // Cursor\_Home();

          Flag2[1] = 1;

          break;

        }

        if(key == 13)

        {

          LCD\_Init();

          Delay\_ms(10);

          LCD\_ShowString(1,1," Close success");

          Delay\_ms(100);

          Flag2[1] = 0;

          // LCD\_Init();

          // Delay\_ms(10);

          // LCD\_SetCursor(1,16);

          // LCD\_WriteData(0xFE);

          // Cursor\_Home();

          break;

        }

    }

}

**clock.h**

#ifndef \_CLOCK\_H

#define \_CLOCK\_H

extern int num\_clc;

extern int quantity;

//  typedef struct clock\_structure

//  {

extern    int open[5];       //是否开启了闹钟

extern    int date\_whe[5] ;//是否按照日期设置

extern    int date\_d[5] ;  //设置的日期

extern    int clc\_week[5][7];    //设置每周哪几天闹铃

extern    int length[5] ;     //响铃时间时长

extern    int inter[5] ;   //响铃间隔时长

extern    int repeat[5] ; //是否重复，0为重复0次，最多5次

extern    int fre[5] ;    //声光频率

extern    int light[5]; //光

extern    int sound[5] ;  //声

extern int clc\_hour[5];

extern int clc\_min[5];

extern int this\_moment;

//  }\*CLC;

//void clock\_init();

void pass();//用于每次显示的间隔判断函数

void if\_date0();//是否你要设置日期

void set\_date0();//设置日期

void func\_choose();//功能表，进入你想要的功能

void if\_ctl\_time();//是否要设置时间

void ctl\_time();//设置时间

void point();//设置要不要开启整点报时

void If\_set();//是否要设置闹钟

void Whi\_ch();//你要设置那个闹钟

void If\_date();//是否要按照日期设置

void set\_date();//设置日期

void set\_week();//设置每周哪几天闹铃

void set\_length();//设置闹铃时长

void set\_inter();//设置响铃间隔

void set\_repeat();//设置要闹铃几次

void set\_light();//设置是否开启灯光提示

void set\_sound();//设置是否要开启声音提示

void set\_remind();//设置提示的频率

void set\_time();//设置闹铃的时间

void Congratulation();//恭喜设置时间成功

void Congratulation1();//恭喜设置闹铃成功

void settime();//真的要设置闹铃时间

void set\_point();//设置整点闹铃的一些设置

void final\_check();//再次检查是不是闹钟设置重复的

void close\_clc();//选择关闭闹钟

#endif

**ring.c**

#include "main.h"

int FRE\_CHOOSE[6] = {3,250,100,50,15,3} ;

int ring\_repeat[5] = {0,0,0,0,0};//控制重复闹铃的变量

int ring\_flag = 0;//当前有闹铃在闹的标志位

int ini\_min[5] = {0,0,0,0,0};

int ini\_hour[5] = {0};

void ring\_down()

{

    int temp\_sec;

    if(countflag == 1)

    {

        countflag = 0;

        if((check\_min <= global\_min)

        && (check\_sec <= global\_second))

        {

        if(this\_moment == 0)

        {

            this\_moment = 1;

            LCD\_Init();

            Delay\_ms(10);

            LCD\_ShowString(1,1," Count Reached");

            temp\_sec = global\_second;

            while( (global\_second-temp\_sec)<=10 )

            {

                Led0\_ON();

                Led1\_OFF();

                buz = 1;

                Delay\_ms(FRE\_CHOOSE[fre[3]]);

                Led0\_OFF();

                Led1\_ON();

                buz = 0;

                Delay\_ms(FRE\_CHOOSE[fre[3]]);

                if(key == 18)

                {

                    break;

                }

            }

            buz = 0;

            Led0\_OFF();

            Led1\_OFF();

            LCD\_Init();

            Delay\_ms(10);

            time\_show();

        }

    }

    }

}

void point()

{

        if(key == 3)

        {

            set\_point();

            fun\_re();

        }

}

void point\_ring0()

{

    if(Flag2[0] == 1)

    {

        Flag2[0] = 0;

        point\_ring();

    }

    ring\_down();

    Check\_wh1();

}

void rinning(int fre)

{

    Led0\_ON();

    buz = 1;

    Delay\_ms(fre);

    Led0\_OFF();

    buz = 0;

    Delay\_ms(fre);

}

void point\_ring()//若调用本函数，将在整点闹铃，即声光提示

{

    int temp\_sec;

    if(Flag2[1] == 1)

    {

        LCD\_Init();

        Delay\_ms(10);

        LCD\_ShowString(1,1,"  :00 reached!");

        LCD\_ShowNum(1,1,hour,2);

        if(min < 3)

        {

            temp\_sec = global\_second;

            while((global\_second-temp\_sec)<=10)

            {

                    rinning(FRE\_CHOOSE[4]);

            }

            this\_moment = 1;

            buz = 0;

            Led0\_OFF();

            Led1\_OFF();

        }

        LCD\_Init();

        Delay\_ms(10);

        time\_show();

    }

}

void confirm()

{

    T4CON &= 0xFB;

    if(open[0] || open[1] || open[2] || open[3] || open[4])

        {

            ring\_flag = 1;

    }

    T4CON |= 0x04;

}

void Check\_wh1()

{

        if(ring\_flag == 1)

        {

            ring\_flag = 0;

            if(open[0] == 1)

            {

                    Check\_time(0);

            }

            Check\_wh2();

        }

}

void Check\_wh2()

{

    if(open[1] == 1)

    {

        Check\_time(1);

    }

    Check\_wh3();

}

void Check\_wh3()

{

    if(open[2] == 1)

    {

        Check\_time(2);

    }

    Check\_wh4();

}

void Check\_wh4()

{

    if(open[3] == 1)

    {

        Check\_time(3);

    }

    Check\_wh5();

}

void Check\_wh5()

{

    if(open[4] == 1)

    {

        Check\_time(4);

    }

}

void Check\_time(int order)//该函数确定闹钟模式

{

    if(date\_whe[order] == 1)

    {

        ring\_date(order);

    }

    else

    {

        zhende\_ring(order);

    }

}

void ring\_date(int order)

{

    if(date\_d[order] == day)

    {

        zhende\_ring(order);

    }

}

void zhende\_ring(int order)

{

    if((clc\_week[order][week-1]) )

    {

            if((min == clc\_min[order]) && (hour == clc\_hour[order]))

            {

                if(repeat[order] == 0)

                {

                    just\_ring(order);

                }

                else

                {

                    if(ring\_repeat[order] < repeat[order])

                    {

                    ready\_ring(order);

                    ring\_repeat[order]++;

                    }

                    else if(ring\_repeat[order] == repeat[order])

                    {

                    ready\_ring(order);

                    ring\_repeat[order] = 0;

                    clc\_min[order] = ini\_min[order];

                    clc\_hour[order] = ini\_hour[order];

                    }

                }

            }

    }

}

void ready\_ring(int order)

{

        if(ring\_repeat[order] == 0)

        {

            ini\_min[order] =  clc\_min[order];

            ini\_hour[order] = clc\_hour[order];

        }

    if(ring\_repeat[order] == repeat[order])

    {

        just\_ring(order);

    }

    else if(ring\_repeat[order] < repeat[order])

    {

        just\_ring(order);

        reload(order);

    }

}

void reload(int order)

{

    if((clc\_min[order] + inter[order]) >= 60)

    {

        clc\_min[order] = (clc\_min[order] + inter[order] - 60);

        clc\_hour[order] ++;

        if(clc\_hour[order] == 24)

        {

            clc\_hour[order] = 0;

        }

    }

    if((clc\_min[order] + inter[order]) < 60)

    {

        clc\_min[order] = (clc\_min[order] + inter[order]);

    }

}

void just\_ring(int order)

{

    int temp\_sec;

    if(this\_moment == 0)

    {

        if((min == clc\_min[order]) && (hour == clc\_hour[order]) && (second < 3))

            {

                this\_moment = 1;

                ring\_show(order);

                temp\_sec = global\_second;

                while( (global\_second-temp\_sec)<=length[order] )

                {

                    if(light[order] == 1)

                    {

                        Led0\_ON();

                        Led1\_OFF();

                    }

                    if(sound[order] == 1)

                    {

                    buz = 1;

                    }

                    Delay\_ms(FRE\_CHOOSE[fre[order]]);

                    Led0\_OFF();

                    Led1\_ON();

                    buz = 0;

                    Delay\_ms(FRE\_CHOOSE[fre[order]]);

                    if(key == 11)//关闭当前正在显示的闹铃

                    {

                        break;

                    }

                    if(key == 18)//按下F7后，不再重复闹铃，即使后面还有

                    {

                        ring\_repeat[order] = repeat[order];

                        break;

                    }

                    LCD\_ShowNum(1,4,min,2);

                    LCD\_ShowNum(1,7,second,2);

                }

                buz = 0;

                Led0\_OFF();

                Led1\_OFF();

                LCD\_Init();

                Delay\_ms(10);

                time\_show();

            }

    }

}

**ring.h**

#ifndef \_RING\_H

#define \_RING\_H

extern int FRE\_CHOOSE[6];

extern int ring\_repeat[5];

extern int ring\_flag;

void ring\_down();//倒计时响铃函数

void rinning(int fre);//倒计时真的在响铃时调用的函数

void confirm();//确认当前有没有开了的闹钟

void Check\_wh1();//检查第一个闹铃开没开

void point\_ring();//整点报时判断

void point\_ring0();//整点报时执行

void Check\_wh2();//检查第二个闹铃开没开

void Check\_wh3();//检查第三个闹铃开没开

void Check\_wh4();//检查第四个闹铃开没开

void Check\_wh5();//检查第五个闹铃开没开

void Check\_time(int order);//检查当前是否满足时间要求

void ring\_date(int order);//日期检查

void zhende\_ring(int order);//真的进入闹铃函数了啊

void ready\_ring(int order);//准备闹铃函数

void reload(int order);//装载下一次闹铃时间

void just\_ring(int order);//仅仅是控制闹铃的函数

#endif

**countdown.c**

#include "main.h"

int count\_min = 0;

int count\_sec = 30;

int check\_min = 0;

int check\_sec = 0;

int countflag = 0;

void if\_count()

{

  if(key == 7)

  {

        count\_down();

        while (1)

        {

            set\_if();

            count\_clo();

            if\_date0();

            changeshow();

    if(key == 19)

        {

            break;

        }

        }

        LCD\_Init();

        Delay\_ms(10);

        time\_show();

    }

}

void count\_clo()

{

    int left\_min = 0;

    int left\_sec = 0;

    if(key == 6)

    {

        if(countflag != 0)

        {

            LCD\_Init();

            Delay\_ms(10);

            LCD\_ShowString(1,1,"F1 To CLOSE");

            LCD\_ShowString(2,1,"F8 t0 Quit");

            pass();

            LCD\_Init();

            Delay\_ms(10);

            LCD\_ShowString(1,1,"TIMER LEFT");

            LCD\_ShowString(2,7," : ");

            left\_min = (count\_sec- global\_second)/60 ;

            left\_sec = count\_sec - global\_second-(count\_min\*60);

            LCD\_ShowNum(2, 10, left\_sec,2);

            LCD\_ShowNum(2, 5,  left\_min,2);

            while(1)

            {

                left\_min = (check\_sec- global\_second)/60 ;

                left\_sec = check\_sec - global\_second-(count\_min\*60);

                LCD\_ShowNum(2, 10, left\_sec,2);

                LCD\_ShowNum(2, 5,  left\_min,2);

                Delay\_us(50);

                if(key == 12)

                {

                  countflag = 0;

                  LCD\_Init();

                  Delay\_ms(10);

                  LCD\_ShowString(1,1,"Count Closed");

                  Delay\_ms(200);

                  break;

                }

                if(key == 19)

                {

                  break;

                }

                if(check\_sec <= global\_second)

                {

                  countflag = 0;

                  LCD\_Init();

                  Delay\_ms(10);

                  LCD\_ShowString(1,1,"Count Closed");

                  Delay\_ms(200);

                  break;

                }

            }

        }

        else

        {

            LCD\_Init();

            Delay\_ms(10);

            LCD\_ShowString(1,1,"None is running");

            LCD\_ShowString(2,1,"Quit to set one");

            Delay\_ms(100);

        }

        count\_re();

    }

}

void set\_if()

{

    if(key == 5)

    {

        countflag = 0;

        count\_min = 0;

        count\_sec = 30;

        LCD\_Init();

        Delay\_ms(10);

        LCD\_ShowString(1,1," SET CountDown");

        pass();

        LCD\_Init();

        Delay\_ms(10);

        LCD\_ShowString(1,1,"F3 t0 INC MIN!");

        LCD\_ShowString(2,1,"F4 to DES MIN!");

        pass();

        LCD\_Init();

        Delay\_ms(10);

        LCD\_ShowString(1,1,"F1 t0 INC SEC!");

        LCD\_ShowString(2,1,"F2 to DES SEC!");

        pass();

        LCD\_Init();

        Delay\_ms(10);

        LCD\_ShowString(1,1,"F8 t0 set");

        LCD\_ShowString(2,7," : ");

        LCD\_ShowNum(2, 5, count\_min,2);

        LCD\_ShowNum(2, 10,  count\_sec,2);

        set\_count\_time();

        LCD\_Init();

        Delay\_ms(10);

        LCD\_ShowString(1,1,"START RUNNING");

        Delay\_ms(100);

        check\_min = global\_min + count\_min ;

        check\_sec = global\_second + count\_sec + (count\_min\*60);

        countflag = 1;

        count\_re();

    }

}

void set\_count\_time()

{

    while(1)

    {

      if(key == 16)

      {

        count\_sec +=1;

        if(count\_sec >= 60)

        {

          count\_sec = 10;

        }

        LCD\_ShowNum(2,10, count\_sec,2);

        Delay\_us(500);

      }

      if(key == 17)

      {

        if(count\_sec <= 10 )

        {

          count\_sec = 59;

        }

        else

        {

           count\_sec -=1;

        }

        LCD\_ShowNum(2,10, count\_sec,2);

        Delay\_us(500);

      }

      if(key == 14)

      {

        count\_min++;

        if(count\_min >=31)

        {

          count\_min = 0;

        }

        LCD\_ShowNum(2,5, count\_min,2);

        Delay\_us(500);

      }

      if(key == 15)

      {

        if(count\_min == 0 )

        {

          count\_min = 30;

        }

        else

        {

          count\_min--;

        }

        LCD\_ShowNum(2,5, count\_min,2);

        Delay\_us(500);

      }

      if(key == 19)

      {

          break;

      }

    }

}

**countdown.h**

#ifndef COUNTDOWN\_H\_

#define COUNTDOWN\_H\_

extern int count\_min ;

extern int count\_sec ;

extern int check\_min ;

extern int check\_sec ;

extern int countflag;

void if\_count();//设置是否要看当前状态

void set\_if();//设置是否要更改倒计时时长

void count\_clo();//检查倒计时状态函数

void set\_count\_time();//设置倒计时时长

#endif

**display.c**

#include "main.h"

int weekflag = 0;

extern int  day ;

extern int  month ;

extern int  year ;

extern int week;

int showhour = 0;

int timeflag = 1;

void time\_show()

{

    LCD\_ShowString(2,1,"    -  -");

    LCD\_ShowString(1,1,"  :  :");

    // LCD\_ShowString(2,11,"Week");

    LCD\_ShowNum(2,1,year,4);

    LCD\_ShowNum(2,6,month,2);

    LCD\_ShowNum(2,9,day,2);

    LCD\_ShowString(2,13,wek[week-1]);

    if(timeflag == 1)

    {

        LCD\_ShowNum(1,1,hour,2);

        LCD\_ShowString(1,9," ");

    }

    else

    {

        showhour = (hour > 12)? (hour - 12) : hour;

        LCD\_ShowNum(1,1,showhour,2);

        if((hour > 12) || (hour == 0))

        {

            LCD\_ShowString(1,9,"p");

        }

        else

        {

            LCD\_ShowString(1,9,"a");

        }

    }

    LCD\_ShowNum(1,4,min,2);

    LCD\_ShowNum(1,7,second,2);

    LCD\_ShowString(1,11,"CLC:");

    quantity = (open[0] + open[1]+open[2] +open[3] + open[4]);

    LCD\_ShowNum(1,14,quantity,1);

    if(Flag2[1] == 1)

    {

        LCD\_SetCursor(1,16);

        LCD\_WriteData(0xEF);

        Cursor\_Home();

    }

    else

    {

        LCD\_SetCursor(1,16);

        LCD\_WriteData(0xFE);

        Cursor\_Home();

    }

}

void changeshow()

{

    int temp = 0;

    if(key == 8)

    {

        LCD\_Init();

    Delay\_ms(10);

        LCD\_ShowString(1,1," SET TIME SHOW");

        pass();

        LCD\_Init();

        Delay\_ms(10);

        LCD\_ShowString(1,1,"F1 For 12!");

        Delay\_ms(10);

        LCD\_ShowString(2,1,"F2 For 24!");

        Delay\_ms(10);

        pass();

        LCD\_Init();

        Delay\_ms(10);

        LCD\_ShowString(1,1,"Current Mode");

        while(1)

        {

        temp = (timeflag \* 12) +12;

        LCD\_ShowNum(2,9,temp,2);

        if(key==12)

        {

        timeflag = 1;

        Delay\_us(500);

        }

        if(key == 13)

        {

        timeflag = 0;

        Delay\_us(500);

        }

        if(key == 19)

        {

            LCD\_Init();

            Delay\_ms(10);

            LCD\_ShowString(1,1," SET     NOW ");

            temp = (timeflag \* 12) +12;

            LCD\_ShowNum(1,7,temp,2);

            Delay\_ms(150);

            break;

        }

        }

        count\_re();

    }

}

void time\_showplus()

{

    LCD\_ShowNum(1,7,second ,2);

}

void time\_showmin()

{

    LCD\_ShowNum(2,1,year,4);

    LCD\_ShowNum(2,6,month,2);

    LCD\_ShowNum(2,9,day,2);

    LCD\_ShowNum(2,15,week,1);

}

void ring\_show(int order)

{

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"  :  :");

    LCD\_ShowString(2,7,"CLOCK");

    LCD\_ShowNum(2,12,(order+1), 1);

    LCD\_ShowNum(1,1,hour,2);

}

void  checkday()

{

        if((month ==1)|| (month ==3) || (month == 5)||(month == 7)||(month == 8)||(month == 10))

        {

            if(day > 31)

            {

                day = 1;

                month++;

            }

        }

        else    if((month == 4)||(month == 6)|| (month == 9) || (month == 11))

        {

            if(day > 30)

            {

                day = 1;

                month++;

            }

        }

        else    if(month ==12)

        {

            if(day > 31)

            {

                day = 1;

                month = 1;

                year ++;

            }

        }

        else

        {

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

            {

                if(day >29)

                {

                    day = 1;

                    month++;

                }

            }

            else

            {

                if(day > 28)

                {

                    day = 1;

                    month++;

                }

            }

        }

    week = TimeToWeek(year,month,day);

    if(week == 0)

    {

        week = 7;

    }

}

int TimeToWeek(int year,int month,int day)

{

    int week = 0;

    int AllDay = 0;

    int AllYearDay = 0;

    int AllMonthDay = 0;

    int i = 0;

    //计算年的总天数

    for(i=1970;i<year;i++)

    {

        if(((i%4==0)&&(i%100!=0))||(i%400==0))  //润年

        {

            AllYearDay += 366;    //31天--7个月-217  30天--4个月-120  2月-29

        }

        else

        {

            AllYearDay += 365;

        }

    }

    //计算月的总天数

    for(i = 1;i < month;i++)

    {

        if((i==1)||(i==3)||(i==5)||(i==7)||(i==8)||(i==10)||(i==12))

        {

            AllMonthDay += 31;

        }

        else if(i==2)

        {

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

            {

                AllMonthDay += 29;

            }

            else

            {

                AllMonthDay += 28;

            }

        }

        else

        {

            AllMonthDay += 30;

        }

    }

    //计算1970-01-01到当前时间所过的总天数

    AllDay = AllYearDay+AllMonthDay+day;

    //计算当前为周几，0为周日 1～6对应周一至周六

    week = (AllDay-4)%7;  //1970-01-01 为周4

    return week;

}

void fun\_init()

{

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1," F8 TO Quit ");

    while(1)

    {

        if(key == 18)

        break;

    }

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"1: SET");

    LCD\_ShowString(1,9,"2: CLOSE");

    LCD\_ShowString(2,1,"3: POINT");

    LCD\_ShowString(2,9,"4: TIME");

}

void fun\_re()

{

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"1: SET");

    LCD\_ShowString(1,9,"2: CLOSE");

    LCD\_ShowString(2,1,"3: POINT");

    LCD\_ShowString(2,9,"4: TIME");

}

void count\_down()

{

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1," F8 TO Quit ");

    pass();

    LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"5: SET");

    LCD\_ShowString(1,9,"6: STOP");

    LCD\_ShowString(2,1,"7: DATE");

    LCD\_ShowString(2,9,"8: 12/24");

}

void count\_re()

{

        LCD\_Init();

    Delay\_ms(10);

    LCD\_ShowString(1,1,"5: SET");

    LCD\_ShowString(1,9,"6: STOP");

    LCD\_ShowString(2,1,"7: DATE");

    LCD\_ShowString(2,9,"8: 12/24");

}

}

**display.h**

#ifndef \_DISPLAY\_H

#define \_DISPLAY\_H

extern int timeflag;

extern int showhour ;

void time\_show();

void time\_showplus();

int     TimeToWeek(int year,int month,int day);//计算当前是星期几

void    checkday();//检查日期是否正常

void    time\_showmin();//分钟刷新

void    ring\_show(int order);//闹铃时显示的页面

void    fun\_init();//功能表1初始进入

void    fun\_re();//功能表1再次进入

void    count\_down();//功能表2初始进入

void    count\_re();//功能表2再次进入

void    changeshow();//小时制更改函数

#endif

**Hardware**

**buzzer.c**

#include "main.h"

//sbit buz = P3^5;

void buzz(int time)

{

    buz = 1;

    Delay\_ms(time);

    buz = 0;

}

**buzzer.h**

#ifndef \_BUZZER\_H

#define \_BUZZER\_H

sbit buz = P3^5;

void buzz(int time);

#endif

**key.c**

#include "main.h"

sbit DAT=P3^6;

sbit CLK=P3^7;

int key = 20;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \* @brief  按键初始化

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Init\_Key()                         //set shift register to all 0

{

    int i;

    for(i=0;i<8;i++)

    {

            DAT=0;

            CLK=0;

            CLK=1;

    }

}

void release(void)

{

        int button,temp=0;

        int j=1;         //循环标志

        TH0 = (65536 - 60000) / 256;

      TL0 = (65536 - 60000) % 256;

        button=P4&0x0f;

        while(j)

        {

           button=P4&0x0f;

            if(button==0x0f)        //有键按下

            {

                Delay\_ms (5);        //消抖

                button=P4&0x0f;

                if(button==0x0f)

                {

                    j=0;

                }

            }

   }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \* @brief  读取当前按键值 ，未按下时为20

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int Key\_Scan()

{

    int button,temp = 0;    //读P4

        Send(0x00);

        button=P4&0x0f;

        if(button!=0x0f)        //有键按下

        {

//          LCD\_ShowNum(2,1,10,2);

//          Cursor\_Home();

            Delay\_ms (5);        //消抖

          button=P4&0x0f;

            if(button!=0x0f)//第一行 在上升沿写入一位低电平，开始逐行检测 若P4再次变0，说明按下的键在该行，然后再根据之前存的BUUTON值求键值

            {

                Send(0xFE);

                temp=P4&0x0f;

        if(temp!=0x0f)  //若P4再次变1，说明按下的键在该行，然后再根据之前存的BUUTON值求键值

                {

                    switch(button)

                    {

                        case 0x0e:  return 0;

                        case 0x0d:  return 1;

                        case 0x0b:  return 2;

                        case 0x07:  return 3;

                        default:        break;

                    }

              }

                else

                {

                    //第二行

                    Send(0xFD);

                    temp=P4&0x0f;

                    if(temp!=0x0f)

                    {

                        switch(button)

                        {

                            case 0x0e:  return 4;

                            case 0x0d:  return 5;

                            case 0x0b:  return 6;

                            case 0x07:  return 7;

                            default:

                            break;

                        }

                     }

                    else

                    {

                        //第三行

                    Send(0xFB);

                    temp=P4&0x0f;

                    if(temp!=0x0f)

                    {

                        switch(button)

                        {

                            case 0x0e:   return 8;

                            case 0x0d:   return 9;

                            case 0x0b:   return 10;

                            case 0x07:   return 11;

                            default: break;

                        }

                     }

                    else

                    {

                                //第四行

                            Send(0xF7);

                            temp=P4&0x0f;

                            if(temp!=0x0f)

                            {

                                switch(button)

                                {

                                    case 0x0e:   return 12;

                                    case 0x0d:   return 13;

                                    case 0x0b:   return 14;

                                    case 0x07:   return 15;

                                    default: break;

                                }

                             }

                            else

                            {

                                    //第五行

                                Send(0xEF);

                                temp=P4&0x0f;

                                if(temp!=0x0f)

                                {

                                    switch(button)

                                    {

                                        case 0x0e:   return 16;

                                        case 0x0d:   return 17;

                                        case 0x0b:   return 18;

                                        case 0x07:   return 19;

                                        default: break;

                                    }

                                }

                                else

                                {

                                    return 20;

                                }

                            }

                        }

                    }

                }

            }

            else return 20;

      }

        else return 20;

        return 20;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \* @brief  控制74HCT164

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Send(char keyscan)

{

    int i=0;

    for(i=0;i<8;i++)

    {

        if(keyscan&0x80)

        {

            DAT=1;

            CLK=0;

            CLK=1;

        }

      else

        {

            DAT=0;

            CLK=0;

            CLK=1;

        }

        keyscan=keyscan<<1;

    }

}

**key.h**

#ifndef \_KEY\_H

#define \_KEY\_H

extern int      key;

void Init\_Key();

int     Key\_Scan();

void    release(void);

void    Send(char keyscan);

#endif

**lcd.c**

#include "main.h"

/\*\*

  \* @brief  移动光标到指定位置

  \* @param  Line 行位置，范围：1~2

  \* @param  Column 列位置，范围：1~16

  \* @retval 无

  \*/

void LCD\_SetCursor(unsigned char Line,unsigned char Column)

{

    int i = 0;

    if(Line==1)

    {

        if(Column > 1)

        {

        for(i = 0; i<Column-1;i++)

        LCD\_WriteCommand(0x14);

        }

    }

    else if(Line==2)

    {

        for(i = 0; i<40; i++)

        LCD\_WriteCommand(0x14);

        if(Column > 1)

        {

        for(i = 0; i<Column-1;i++)

        LCD\_WriteCommand(0x14);

        }

    }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \* @brief  清屏

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Clear()

{

    LCD\_WriteCommand(0x01);

}

void LCD\_WriteCommand(unsigned char Command)

{

    LCD\_RS=0;

    LCD\_RW=0;

    LCD\_DataPort=Command;

    LCD\_EN=1;

    LCD\_Delay();

    LCD\_EN=0;

    LCD\_Delay();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \* @brief 光标回到初始位置

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Cursor\_Home()

{

    LCD\_WriteCommand(0x02);

}

void LCD\_Delay()

{

    Delay\_us(10);

}

void LCD\_WriteData(unsigned char Data)

{

    LCD\_RS=1;

    LCD\_RW=0;

    LCD\_EN=1;

    LCD\_DataPort=Data;

    LCD\_EN=1;

    LCD\_Delay();

    LCD\_EN=0;

    LCD\_Delay();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  \* @brief  将LCD屏幕初始化

  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LCD\_Init()

{

    LCD\_EN=0;

    LCD\_WriteCommand(0x01);//光标复位，清屏

    LCD\_WriteCommand(0x06);//数据读写操作后，光标自动加一，画面不动

    LCD\_WriteCommand(0x0c);//显示开，光标关，闪烁关

    LCD\_WriteCommand(0x38);//八位数据接口，两行显示，5\*7点阵

}

/\*\*

  \* @brief  在LCD1602指定位置上显示一个字符

  \* @param  Line 行位置，范围：1~2

  \* @param  Column 列位置，范围：1~16

  \* @param  Char 要显示的字符

  \* @retval 无

  \*/

void LCD\_ShowChar(unsigned char Line,unsigned char Column,char Char)

{

    LCD\_SetCursor(Line,Column);

    LCD\_WriteData(Char);

}

/\*\*

  \* @brief  在LCD1602指定位置开始显示所给字符串

  \* @param  Line 起始行位置，范围：1~2

  \* @param  Column 起始列位置，范围：1~16

  \* @param  String 要显示的字符串

  \* @retval 无

  \*/

void LCD\_ShowString(unsigned char Line,unsigned char Column,char \*String)

{

    unsigned char i;

    LCD\_SetCursor(Line,Column);

    for(i=0;String[i]!='\0';i++)

    {

        LCD\_WriteData(String[i]);

    }

    Cursor\_Home();

}

void LCD\_number(unsigned char Line,unsigned char Column,int Char)

{

    LCD\_SetCursor(Line,Column);

    LCD\_WriteData(Char);

}

void LCD\_Shownum1(unsigned char Line,unsigned char Column,int \*String)

{

    unsigned char i;

    LCD\_SetCursor(Line,Column);

    for(i=0;String[i]!='\0';i++)

    {

        LCD\_WriteData(String[i]);

    }

    Cursor\_Home();

}

/\*\*

  \* @brief  返回值=X的Y次方

  \*/

int LCD\_Pow(int X,int Y)

{

    unsigned char i;

    int Result=1;

    for(i=0;i<Y;i++)

    {

        Result\*=X;

    }

    return Result;

}

/\*\*

  \* @brief  在LCD1602指定位置开始显示所给数字

  \* @param  Line 起始行位置，范围：1~2

  \* @param  Column 起始列位置，范围：1~16

  \* @param  Number 要显示的数字，范围：0~65535

  \* @param  Length 要显示数字的长度，范围：1~5

  \* @retval 无

  \*/

void LCD\_ShowNum(unsigned char Line,unsigned char Column,unsigned int Number,unsigned char Length)

{

    unsigned char i;

    LCD\_SetCursor(Line,Column);

    for(i=Length;i>0;i--)

    {

        LCD\_WriteData(Number/LCD\_Pow(10,i-1)%10+'0');

    }

    Cursor\_Home();

}

**lcd.h**

#ifndef \_LCD\_H\_

#define \_LCD\_H\_

#include "main.h"

#define LCD\_DataPort P2

sbit LCD\_EN = P3^2; //input enable;

sbit LCD\_RW = P3^1; //H=read; L=write;

sbit LCD\_RS = P3^0; //H=data; L=command;

void LCD\_Init();

void Clear();

void LCD\_Delay();

void LCD\_WriteCommand(unsigned char Command);

void LCD\_WriteData(unsigned char Data);

void Cursor\_Home();

void LCD\_ShowChar(unsigned char Line,unsigned char Column,char Char);

void LCD\_SetCursor(unsigned char Line,unsigned char Column);

void LCD\_ShowString(unsigned char Line,unsigned char Column,char \*String);

int LCD\_Pow(int X,int Y);

void LCD\_ShowNum(unsigned char Line,unsigned char Column,unsigned int Number,unsigned char Length);

#endif

**LED.c**

#include "main.h"

void Led0\_ON()

{

    P4 |= 0x10;

}

void Led0\_OFF()

{

    P4 &= 0xEF;

}

void Led1\_ON()

{

    P4 |= 0x20;

}

void LED1\_OFF()

{

    P4 &= 0xDF;

}

**LED.h**

#ifndef \_LED\_H

#define \_LED\_H

void Led0\_ON();//开0号灯

void Led0\_OFF();//关0号灯

void Led1\_ON();//开1号灯

void Led1\_OFF();//开0号灯

#endif

**Software**

**delay.c**

#include "main.h"

void Delay\_us (int us)

{

  int a,b,c,d,e;

  for(e=0;e<us;e++)

  {

    for(d=1;d>0;d--)

        for(c=1;c>0;c--)

            for(b=1;b>0;b--)

                for(a=1;a>0;a--);

  }

}

void Delay\_ms (int ms)

{

  int a,b,c;

  for(c = 0; c<ms;c++)

  {

    for(b=35;b>0;b--)

      for(a=227;a>0;a--);

  }

}

**delay.h**

#ifndef \_\_DELAY\_H\_\_

#define \_\_DELAY\_H\_\_

void Delay\_us (int us);//延时，微妙级

void Delay\_ms(int ms);//延时，毫秒级

#endif

**timer.c**

#include "main.h"

char  \*xskey;

int check = 0;

int Flag0    = 0;

int Flag1    = 0;

int Flag2[2] = {0};//用于指定是否整点报时

int Flag3 = 0;

int checkday\_flag = 0;//用于表示是否一天过去了，需要更新星期数

int  count = 0;

int  second = 0;

int  min = 0;

int  hour = 0;

int  day = 0;

int  month = 0;

int  year = 0;

int  week = 0;

char \*wek[] = {"Mon","Tue","Wed","Fur","Fri","Sat","Sun"};

 int global\_second = 0;//计算程序运行时间，单位为秒

 int global\_min = 0;//计算程序运行时间单位为分钟

void Timer0\_init()

{

    TMOD = 0x11;      // 定时器0和定时器1同时设置为16位定时方式

    TH0 = 0xF0;       // 放在高八位当中

    TL0 = 0x60;

    EA = 1;

    ET0 = 1;

    TR0 =1;         //中断设置

    PT0 = 0;

}

void Timer1\_init()

{

    TMOD = 0x11;

    TH1 = 0x0FA;

  TL1 = 0x0CB;

    EA = 1;

    ET1 = 1;

    TR1 = 1;

    PT1 = 1;

}

void Timer4\_init()

{

    RCAP4H =0XFA;                                       // 1ms

    RCAP4L =0XCB;

    TH4    =0XFA;

    TL4    =0XCB;

    EIE2  |=0X04;

    T4CON  =0X04;

}

void Timer0SP(void) interrupt 1

{

  key=Key\_Scan();

    TH0 = 0xF0;       // 放在高八位当中 定时20ms

    TL0 = 0x60;

}

void Timer1SP(void) interrupt 3

{

    TH1 = 0x0FA;

    TL1 = 0x0CB;

        count +=1;

    if(count >= 1125)

    {

            count = 0;

            second++;

            global\_second++;

            Flag3 = 1;

            if(second >= 60)

            {

                second = 0;

                min++;

                global\_min++;

                Flag1 = 1;//分钟更新标志位

                if(this\_moment == 1)

                {

                    this\_moment = 0;//响应标志位

                }

                if(min >= 60)

                {

                    min = 0;

                    hour ++;

                    Flag2[0] = 1;//整点报时标志位

                    if(hour >= 24)

                    {

                        hour = 0;

                        Flag0 = 1;

                        if(Flag0 == 1)

                        {

                            day++;

                            checkday\_flag = 1;//日期更新标志位

                        }

                    }

                }

            }

    }

}

void Timer4\_ISR (void) interrupt 16

{

    T4CON &= ~0x80;//重新装载时间

    check += 1;

    if(check >= 300)

    {

        check = 0;

        confirm();

    }

}

**timer.h**

#ifndef \_TIMER\_H

#define \_TIMER\_H

extern char \*xskey;

extern int  count;

extern int  check;

extern int  second ;

extern int  min ;

extern int  hour ;

extern int  day;

extern int  month;

extern int  year;

extern int  week;

extern int  Flag0;

extern int  Flag1;

extern int  Flag2[2];

extern  int  global\_second;

extern  int  global\_min;

extern int Flag3;

extern int checkday\_flag;

extern char \*wek[];

void Timer0\_init();//初始化定时器0

void Timer1\_init();//初始化定时器1

void Timer4\_init();//初始化定时器4

#endif

**systeminit.c**

#include "main.h"

void PORT\_Init()

{

    WDTCN=0xde;                                             //disable watchdog

    WDTCN=0xad;

    XBR0=0x00;                                          //default

    XBR1=0x00;                                          //default

    XBR2=0x40;                                          //enable weak-pull up

    P74OUT=0x02;                                        //p4,p5 low nipple as open-drain, high nipple as push-pull

    P4 = 0xFF;

    P0MDOUT=0x1f;                                       //0

    P1MDIN=0xff;                                        //P1 as intput

    P2MDOUT=0xff;

    P3MDOUT=0xf7;                                       //P3.0,1,2,5,6,7 as push-pull

    buz=0;                                              //close the fucking buzzer

}

void time\_init()

{

    second = 59;

    min = 59;

    hour = 23;

    day = 31;

    month = 12;

    year = 2022;

    week = TimeToWeek(year,month,day);

    if(week == 0)

    week = 7;

    time\_show();

}

void Oscillator\_Init()

{

    OSCICN  = 0x07;                 //Internal 16MHz

        CKCON       =   0x00;

}

void Interrupts\_Init()

{

    IE = 0x84;

        Led0\_OFF();

        Led1\_OFF();

}

void Sys\_Init()

{

    PORT\_Init();

    Oscillator\_Init();

    Interrupts\_Init();

    Init\_Key();

    LCD\_Init();

    //clock\_init();

    time\_init();

}

**systeminit.h**

#ifndef SYSTEM\_INIT

#define SYSTEM\_INIT

void PORT\_Init();//端口初始化

void Oscillator\_Init();//时钟初始化，配置为8M

void Interrupts\_Init();//中断寄存器初始化

void Sys\_Init();//系统初始化，

void time\_init();//初始时间设置

#endif