

## 一、电子时钟简介

本设计利用 Atmel 公司的 AT89C51 单片机对电子时钟进行开发,设计了实现所需功能的硬件电路,应用 C 语言进行软件编程,并用 Proteus 和 KEIL 联调进行仿真演示、验证。主要介绍用单片机内部的定时/计数器来实现电子时钟的方法,本设计由单片机 80C51 芯片和 LED 数码管为核心,辅以必要的电路,构成了一个单片机的数字电子时钟。它的计时周期为 24 小时,显满刻度为“23 时 59 分 59 秒”,且配有 6 个独立键盘,可以灵活地调节时间,并具有一定的扩展性。

### 1、电子时钟的基本特点

现在高精度的计时工具大多数都使用了石英晶体振荡器,由于电子钟、石英钟、石英表都采用了石英技术,因此走时精度高,稳定性好,使用方便,不需要经常调试,数字式电子钟用集成电路计时时,译码代替机械式传动,用 LED 显示器代替指针显示进而显示时间,减小了计时误差,这种表具有时、分、秒显示时间的功能,还可以进行时和分的校对,片选的灵活性好。

### 2、电子时钟的原理

该电子时钟由 89C51, BUTTON, 7 段数码管等构成,采用晶振电路作为驱动电路,由延时程序和循环程序产生的一秒定时,达到时分秒的计时,六十秒为一分钟,六十分钟为一小时,满二十四小时为一天。按键处理设置为:如果没有按键,则熄频,按一下 key 时钟正常走时.按 key 第二下秒表走时,按 key1 两下暂停,三下去清零,然后再按一下秒表正常走时,K1 用于调时,按下停止走时,K3 用于时钟调秒、调分、调时的功能切换选择。定义一个变量来记录 K3 按键的次数,K3=0,调秒;K3=1,调分;K3=2,调时;按键 K4 为增 1 键,对选定的被调项的当前值增 1;按键 K5 为减 1 键,对选定的被调项的当前值减 1。

## 二、单片机简介

单片机是指一个集成在一块芯片上的完整计算机系统。尽管他的大部分功能集成在一块小芯片上，但是它具有一个完整计算机所需要的大部分部件：CPU、内存、内部和外部总线系统，目前大部分还会具有外存。同时集成诸如通讯接口、定时器，实时时钟等外围设备。而现在最强大的单片机系统甚至可以将声音、图像、网络、复杂的输入输出系统集成在一块芯片上。

### 1、单片机的特点

(1) 单片机的存储器 ROM 和 RAM 时严格区分的。ROM 称为程序存储器，只存放程序，固定常数，及数据表格。RAM 则为数据存储器，用作工作区及存放用户数据。

(2) 采用面向控制的指令系统。为满足控制需要，单片机有更强的逻辑控制能力，特别是单片机具有很强的位处理能力。

(3) 单片机的 I/O 口通常时多功能的。由于单片机芯片上引脚数目有限，为了解决实际引脚数和需要的信号线的矛盾，采用了引脚功能复用的方法，引脚处于何种功能，可由指令来设置或由机器状态来区分。

(4) 单片机的外部扩展能力很强。在内部的各种功能部件不能满足应用的需求时，均可在外部进行扩展，与许多通用的微机接口芯片兼容，给应用系统设计带来了很大的方便。

### 2、89C51 单片机介绍

VCC：电源。

GND：接地。

P0 口：P0 口为一个 8 位漏级开路双向 I/O 口，每脚可吸收 8TTL 门电流。当 P1 口的管脚第一次写 1 时，被定义为高阻输入。P0 能够用于外部程序数据存储器，它可以被定义为数据/地址的第八位。在 FIASH 编程时，P0 口作为原码输入口，当 FIASH 进行校验时，P0 输出原码，此时 P0 外部必须被拉高。

P1 口：P1 口是一个内部提供上拉电阻的 8 位双向 I/O 口，P1 口缓冲器能接收输

出 4TTL 门电流。P1 口管脚写入 1 后，被内部上拉为高，可用作输入，P1 口被外部下拉为低电平时，将输出电流，这是由于内部上拉的缘故。在 FLASH 编程和校验时，P1 口作为第八位地址接收。

P2 口：P2 口为一个内部上拉电阻的 8 位双向 I/O 口，P2 口缓冲器可接收，输出 4 个 TTL 门电流，当 P2 口被写“1”时，其管脚被内部上拉电阻拉高，且作为输入。并因此作为输入时，P2 口的管脚被外部拉低，将输出电流。这是由于内部上拉的缘故。P2 口当用于外部程序存储器或 16 位地址外部数据存储器进行存取时，P2 口输出地址的高八位。在给出地址“1”时，它利用内部上拉优势，当对外部八位地址数据存储器进行读写时，P2 口输出其特殊功能寄存器的内容。P2 口在 FLASH 编程和校验时接收高八位地址信号和控制信号。

P3 口：P3 口管脚是 8 个带内部上拉电阻的双向 I/O 口，可接收输出 4 个 TTL 门电流。当 P3 口写入“1”后，它们被内部上拉为高电平，并用作输入。作为输入，由于外部下拉为低电平，P3 口将输出电流（ILL）这是由于上拉的缘故。

P3 口也可作为 AT89C51 的一些特殊功能口，如下表所示：

口管脚备选功能

P3.0 RXD（串行输入口）

P3.1 TXD（串行输出口）

P3.2/INT0（外部中断 0）

P3.3/INT1（外部中断 1）

P3.4 T0（记时器 0 外部输入）

P3.5 T1（记时器 1 外部输入）

P3.6/WR（外部数据存储器写选通）

P3.7/RD（外部数据存储器读选通）

P3 口同时为闪烁编程和编程校验接收一些控制信号。

RST：复位输入。当振荡器复位器件时，要保持 RST 脚两个机器周期的高电平时间。

ALE/PROG：当访问外部存储器时，地址锁存允许的输出电平用于锁存地址的地位字节。在 FLASH 编程期间，此引脚用于输入编程脉冲。在平时，ALE 端以不变的频率周期输出正脉冲信号，此频率为振荡器频率的 1/6。因此它可用作对外部输出的脉冲或用于定时目的。然而要注意的是：每当用作外部数据存储器时，将跳过一个 ALE 脉冲。如想禁止 ALE 的输出可在 SFR8EH 地址上置 0。此时，ALE 只有在执行 MOVX，MOVC

指令时 ALE 才起作用。另外，该引脚被略微拉高。如果微处理器在外部执行状态 ALE 禁止，置位无效。

**PSEN:** 外部程序存储器的选通信号。在由外部程序存储器取指期间，每个机器周期两次/PSEN 有效。但在访问外部数据存储器时，这两次有效的/PSEN 信号将不出现。

**EA/VPP:** 当/EA 保持低电平时，则在此期间外部程序存储（0000H-FFFFH），不管是否有内部程序存储器。注意加密方式 1 时，/EA 将内部锁定为 RESET；当/EA 端保持高电平时，此间内部程序存储器。在 FLASH 编程期间，此引脚也用于施加 12V 编程电源（VPP）。

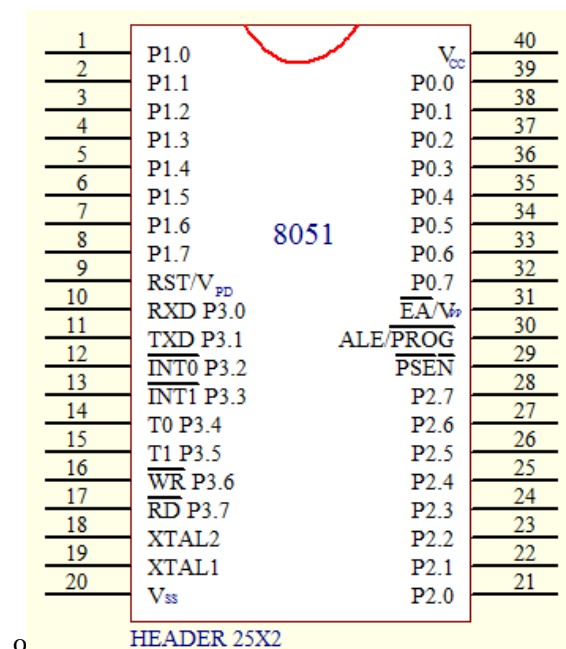
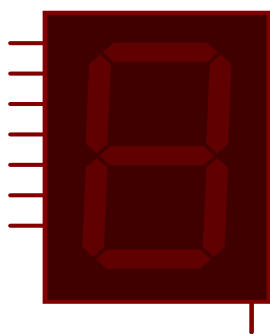


图 2.1 89C51 单片机

### 三、硬件设计

#### 1、数码管显示工作原理

数码管是一种把多个 LED 显示段集成在一起的显示设备。有两种类型，一种是共阳型，一种是共阴型。共阳型就是把多个 LED 显示段的阳极接在一起，又称为公共端。共阴型就是把多个 LED 显示段的阴极接在一起，即为公共端。阳极即为二极管的正极，又称为正极，阴极即为二极管的负极，又称为负极。通常的数码管又分为 8 段，即 8 个 LED 显示段，这是为工程应用方便如设计的，分别为 A、B、C、D、E、F、G、DP，其中 DP 是小数点位段。而多位数码管，除某一位的公共端会连接在一起，不同位的数码管的相同端也会连接在一起。即，所有的 A 段都会连在一起，其它的段也是如此，这是实际最常用的用法。数码管显示方法可分为静态显示和动态显示两种。静态显示就是数码管的 8 段输入及其公共端电平一直有效。动态显示的原理是，各个数码管的相同段连接在一起，共同占用 8 位段引管线；每位数码管的阳极连在一起组成公共端。利用人眼的视觉暂留性，依次给出各个数码管公共端加有效信号，在此同时给出该数码管加有效的数据信号，当全段扫描速度大于视觉暂留速度时，显示就会清晰显示出来。



共阴数码管

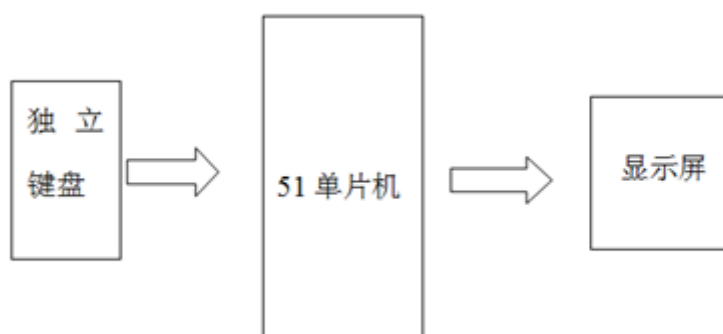
## 2、键盘电路设计

按键处理设置为：如果没有按键，则熄频，按一下 key 时钟正常走时.按 key 第二下秒表走时，按 key1 两下暂停，三下去清零，然后再按一下秒表正常走时，K1 用于调时，按下停止走时，K3 用于时钟调秒、调分、调时的功能切换选择。定义一个变量来记录 K3 按键的次数，K3=0，调秒；K3=1，调分；K3=2，调时；按键 K4 为增 1 键，对选定的被调项的当前值增 1;按键 K5 为减 1 键，对选定的被调项的当前值减 1。

## 3、系统工作原理

- (1) 单片机发送的信号通过程序控制最终在数码管上显示出来。
- (2) 单片机通过输出各种电脉冲信号来驱动控制各部分正常工作。
- (3) 为使时钟走时与标准时间一致，校时电路是必不可少的，键盘用来校正数码管上显示的时间。
- (4) 设计的电路主要由三模块构成：单片机控制电路，显示电路、及校正电路。

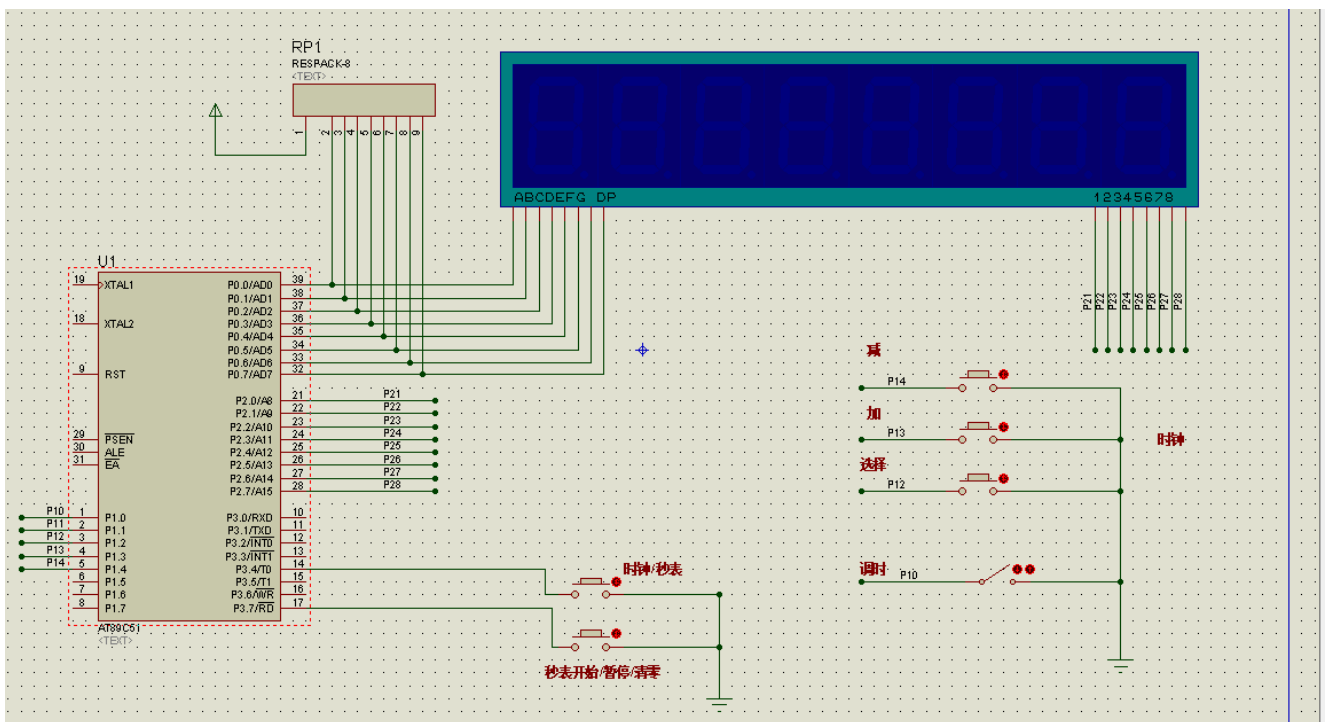
此设计原理框图如图 3-3 所示，此电路包括以下三个部分：单片机，键盘及显示电路。



设计原理框图

本设计采用 C 语言程序设计，使单片机控制数码管显示时、分、秒，当秒计数计满 60 时就向分进位，分计数器计满 60 后向时计数器进位，小时计数器按“23 翻 0”规律计数。时、分、秒的计数结果经过数据处理可直接送显示器显示。当计时发生误差的时候可以用校时电路进行校正。设计采用的是时、分、秒显示，单片机对数据进行处理同时在数码管上显

## 4、电路原理图



## 四、控制系统的软件设计

### 1、程序设计

本系统的软件系统主要可分为主程序、定时计数中断程序、时间调整程序、延时程序四大模块。在程序设计过程中，加强了部分软件抗干扰措施，下面对部分模块作介绍。我们用定时器 1 采用方式 0 定时，定时时间为 10ms。根据需要开始定时器/计数器工作——将 TR0 置“1”。

定时计数中断程序：

```
//T0 中断控制器时钟运行
void T0_INT() interrupt 1

{

    TH0 =(65536-10000)/256;
    TL0 =(65536-10000)%256;
    scount++;
    count++;
    set_miaobiao();
    if(count == 100)
    {
        count = 0;
        timecontrol();
    }
}
```

//秒表  
//时钟  
//判定1s时间是否已到

时间调整程序：

---

```
}
void keyscan_settime() //时间设置模块
{
    uchar set_hour,set_min,set_sec;
    if(K1 == 0)
    {
        delayMS(10);
        if(K1 == 0)
        {
            set_sec = second,set_min = minute, set_hour = hour;
            TR0 = 0; //关闭定时器
            while(K1 == 0) //等待用户按键
            {
                display(set_sec,set_min,set_hour);
            }
        }
    }
    if(K3 == 0)
    {
        delayMS(10);
        if(K3 == 0)
        {
            {
```



```

if(K3 == 0)
{
    delayMS(10);
    if(K3 == 0)
    {
        while(!K3);

        select++;           //选择时分秒
        if(select == 3)
            select=0;
    }
}

if(K4 == 0)
{
    delayMS(10);
    if(K4 == 0)
    {
        while(!K4);
        switch(select)

```

```

        while(!K4);
        switch(select)
        {
            case 0:if(set_sec<59) set_sec++;
                    else set_sec = 0;break;

            case 1:if(set_min<59) set_min++;
                    else set_min = 0;break;

            case 2:if(set_hour<23) set_hour++;
                    else set_hour = 0; break;

            default;;
        }
    }
}

```

```

if(K5 == 0)
{
    delayMS(10);
    if(K5 == 0)
    {
        while(!K5);
        switch(select)
        {
            case 0:if(set_sec > 0) set_sec--;
                    else set_sec = 59;break;

            case 1:if(set_min > 0 ) set_min--;
                    else set_min = 59;break;

            case 2:if(set_hour > 0) set_hour--;
                    else set_hour = 23; break;

            default;;
        }
    }
}

```

```

//通过count获得hour, min, sec变量值
void timecontrol()
{
    second++;
    if(second==60)
    {
        second=0;
        minute++;
        if(minute==60)
        {
            minute=0;
            hour++;
            if(hour==24)
            hour=0;
        }
    }
}

```

## 2、仿真图及结果分析