

单片机原理及应用



项目九：显示接口设计

知识目标

1. 掌握七段数码管的结构；
2. 掌握七段数码管的工作原理
3. 掌握七段数码管与单片机的接口技术

技能目标

- ；
4. 掌握单片机控制七段数码管的编程方法；
 1. 会阐述七段数码管的结构、工作原理；
 2. 会设计七段数码管与单片机的接口的电路；

能力目标

3. 会使用单片机控制七段数码管的编程方法；
1. 能分析设计任务，正确使用七段数码管；
2. 能够正确设计七段数码管与单片机的接口电路；
2. 能使用 PROTUES 软件绘制电路原理图；
3. 能使用 Keil 软件编译程序对七段数码管进行控制，并与 PROTUES 软件联调，实现控制电路仿真；
4. 能够完成 24 秒倒计时电路设计和仿真；
5. 能够使用七段数码管去解决实际问题。

课时建议

4 课时

1.1 任务提出

设计一个篮球 24 秒倒计时系统，用于对球员持球时间 24 秒限制。一旦球员的持球时间超过了 24 秒，自动报警，判定球员的犯规。系统具备显示 24 秒计时功能，计时为递减方式，间隔 1s，当递减到 0 时声光报警。

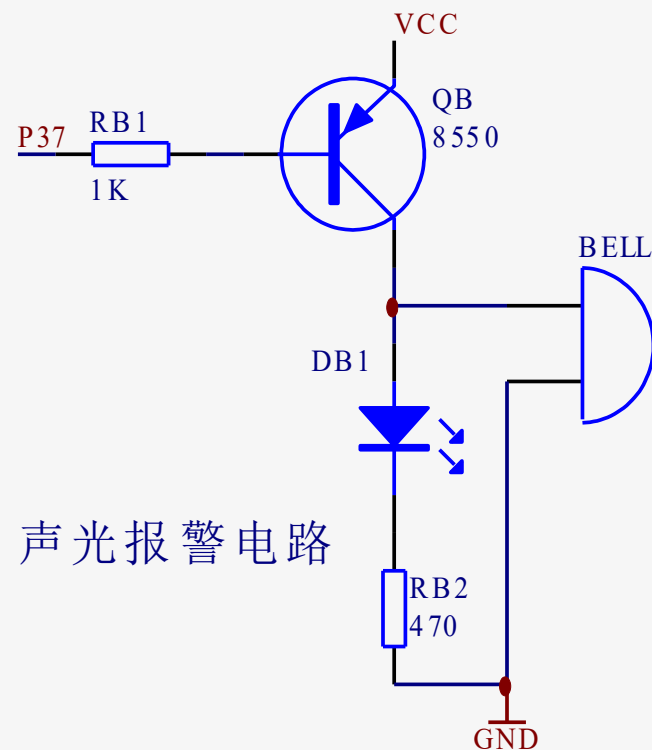


1.2 方案设计

1. 单片机选择。采用 ATMEL 公司的 AT89C51 作为系统控制器的 CPU 方案。单片机算术运算功能强，软件编程灵活、自由度大，可以用软件编程实现各种算法和逻辑控制，并且由于其功耗低、体积小、技术成熟和成本低等优点，使其在各个领域应用广泛。

2. 显示电路的选择。根据题目要求，能够直观显示倒计时时间。采用 LED 数码管显示，LED 数码管价格适中，显示控制也方便。

3. 声光报警电路的选择。由于系统要求同时具有声音和光报警，采用蜂鸣器和发光二极管，配合其它外围元件共同完成。具体电路如图所示。此电路元件少，编程控制简单，只需要单片机输出相应的电平即可驱动电路工作。



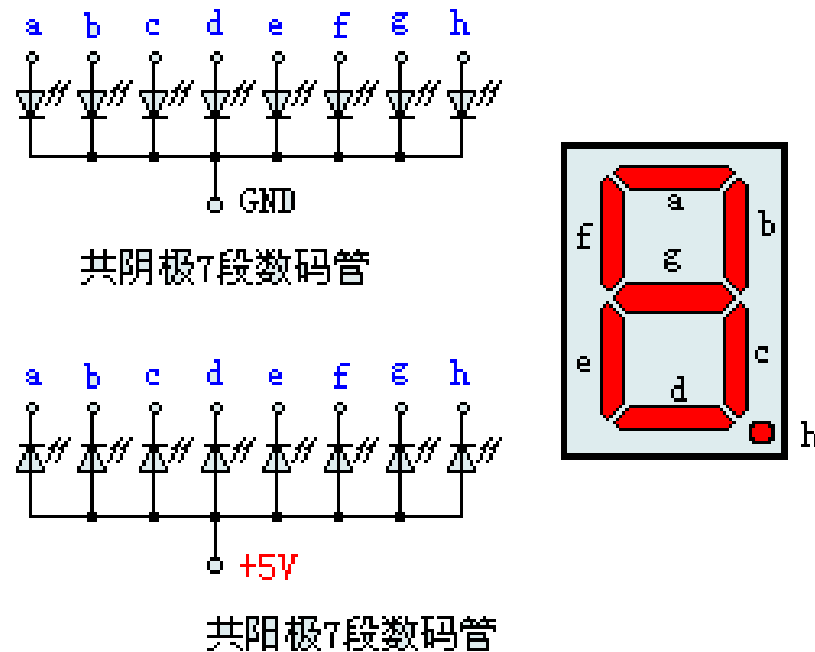
1.3 硬件设计

1. 数码管

数码管是由发光二极管按照一定的顺序排列组合在一起的一种半导体发光器件，按有无小数点可分为七段数码管和八段数码管；按显示的位数可以分为1位、2位、3位、4位、5位、6位、7位等数码管。通过对其不同的引脚施加相对电流就可以显示出不同的数据，比如时间、日期等信息；数码管凭借其价格便宜、使用简单，在家用电器得到了广泛地应用。

1.1 结构原理

数码管根据 LED 的接法不同分为共阴极数码管和共阳数码管两类，共阴和共阳极数码管的接法和其它内部电路如图 9-2 所示。共阴极就是将八个 LED 的阴极连在一起，让其接地，给任何一个 LED 的另一端高电平，便能点亮。而共阳极就是将八个 LED 的阳极连在一起。颜色有红，绿，蓝，黄等几种。



1.3 硬件设计

1.2 数码管驱动方式

(1) 静态驱动方式

数码管要用驱动电路来驱动数码管的各个段码才能正常显示。静态驱动也称直流驱动。静态驱动是指每个数码管的每一个段码都由一个单片机的 I/O 端口进行驱动，或者使用如 BCD 码二 - 十进制译码器译码进行驱动。静态驱动的优点是编程简单，显示亮度高，缺点是占用 I/O 端口多，如驱动 1 个数码管静态显示则需要 8 根 I/O 端口来驱动。实际应用时必须增加译码驱动器进行驱动，硬件电路较复杂。

(2) 动态显示

所谓动态显示就是一位一位地轮流点亮各位显示器（扫描），对于显示器的每一位而言，每隔一段时间点亮一次。在轮流显示过程中，每位数码管的点亮时间为 $1 \sim 2\text{ms}$ ，但利用人眼的视觉暂留效应和发光二极管熄灭时的余辉效应，给人的印象就是一组稳定的显示数据，不会有闪烁感。显示器亮度既与点亮时的导通电流有关，也与点亮时间和间隔时间的比例有关。

动态显示器的优点是节省硬件资源，成本较低，但在控制系统运行过程中，要保证显示器正常显示，CPU 必须每隔一段时间执行一次显示子程序，这占用了 CPU 的大量时间，降低了 CPU 工作效率，同时显示亮度较静态显示器低。

1.3 硬件设计

1.2 数码管显示原理

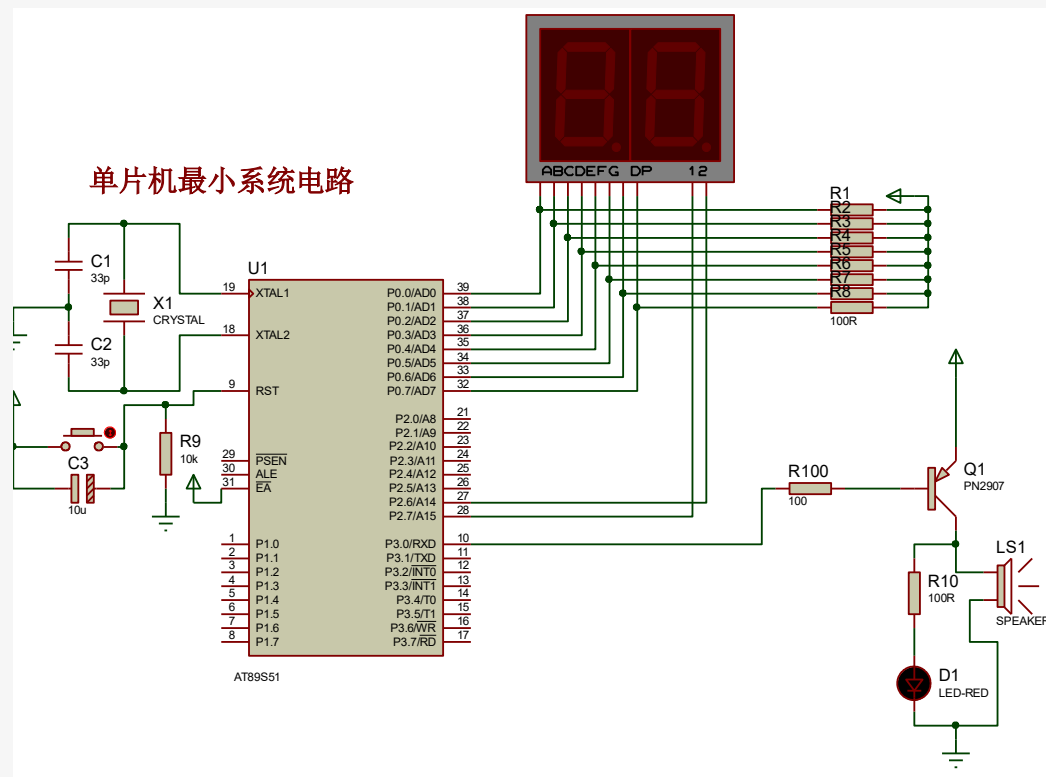
不管是共阴极数码管还是共阳极数码管，想要点亮数码管，都需要对其引脚 **a**、**b**、**c**、**d**、**e**、**f**、**g** 各引脚输入高 / 低电平。对于共阴极数码管，只要引脚输入为高电平，对应的二极管就会发亮，而对于共阳极极数码管，各引脚输入低电平，对应的二极管就会发亮。通过点亮不同的发光段可组成不同的字形。输入到数码管 **dp**、**g**、**f**、**e**、**d**、**c**、**b**、**a** 的二进制码称为字段码（或称字形码），数码管显示的结果为字形。表 是显示字形与共阳极和共阴极两种接法的字段码对应关系。

h	g	f	e	d	c	b	a	共阳极 段码	共阴极 段码	功能 说明
1	1	0	0	0	0	0	0	0xc0	0x3f	显示“0”
1	1	1	1	1	0	0	1	0xf9	0x6h	显示“1”
1	0	1	0	0	1	0	0	0xa4	0x5b	显示“2”
1	0	1	1	0	0	0	0	0xb0	0x4f	显示“3”
1	0	0	1	1	0	0	1	0x99	0x06	显示“4”
1	0	0	1	0	0	1	0	0x92	0x6d	显示“5”
1	0	0	0	0	0	1	0	0x82	0x7d	显示“6”
1	1	1	1	1	0	0	0	0xf8	0x07	显示“7”
1	0	0	0	0	0	0	0	0x80	0x7f	显示“8”
1	0	0	1	0	0	0	0	0x90	0x6f	显示“9”
1	0	0	0	1	0	0	0	0x88	0x77	显示“A”
1	0	0	0	0	0	1	1	0x83	0x7c	显示“B”
1	1	0	0	0	1	1	0	0xc6	0x39	显示“C”
1	0	1	0	0	0	0	1	0xa1	0x5e	显示“D”
1	0	0	0	0	1	1	0	0x86	0x79	显示“E”
1	0	0	0	1	1	1	0	0x8e	0x71	显示“F”

1.3 硬件设计

2. 硬件电路

篮球 24 秒倒计时系统仍然选用 AT89S51 单片机作为控制核心，1 个两位一体的共阳数码管作为输出显示端。AT89S51 的 P0 口接数码管的段码控制，其中 P0.0 ~ P0.6 分别连接数码管的 A ~ G 引脚，P0.7 连接 DP 端，低电平有效。P2 口接数码管位码选通部分，P2.7 口控制第 1 个数码管，一直到 P2.6 口控制第 2 个数码管，高电平有效。要使 2 位数码管显示实现从 24 到 0 的动态计数，实际上就是通过 P2 口输出控制信号轮流选通数码管，共阳型数码管公共端为高电平方可选通，因此要求 P2 口由 P2.7 到 P2.0 依次输出高电平，然后在数码管段码控制端口 P0 按照一定规律送出要显示的数字 0 ~ 9。由于 P0 口带负载能力较小，因此仿真电路中 P0 接入一排上拉电阻。



1.3 硬件设计

3. 软件设计

根据数码管的动态显示方式，要显示的数码管的 8 个笔画端接在单片机的 P0 端口的 8 位上，而单片机的另一个端口 P2 的各个位分别控制 8 位数码管的公共端，控制数码管是否点亮选通。在程序的控制下依次快速输出要显示的各个数，并同时控制对应的数码管工作，在这里选择最左侧两个数码管输出。

1. 计数程序

在程序设计中，采用减 1 程序来达到计数的目的。采用 1 位计数的计算方法完成 59 的计数程序如下：

```
uchar time[]={25,0,0};  
void calc ()  
{ time[0]--;  
if(time[0]==0) time[0]=0;  
}
```

首先把这个数据分成个位、十位，然后把个位、十位所对应的数码管数组中读取控制代码，送单片机端口，进行显示。数据转换程序如下所示。

```
void convert()  
{ display_data[2]=time[0]/10; // 十位  
  display_data[3]=time[0]%10; // 个位  
}
```

1.4 软件设计

3. 软件设计

2. 报警程序

当 24 秒计时到 0 时就进入声光报警程序。这里采用是的死循环。

```
if(time[0]==0)
{
    while(1)
    {
        beep=0;
        delay1();
        beep=1;
        delay1();
    }
}
```

首先将要显示内容的 0 ~ 9 十个数字存放在数组中，程序开始后，数组中的元素不断地逐个送到数码管的段码端口 P0，同时位码控制 P2 端选通要显示的数码管，则数字显示在选中的数码管上。从段码端口来看，动态显示程序显示段码输出的过程实际上和静态显示 10 个数字是一样的，依次输出各个段码；从位码端口看，要哪只数码管显示个位，就在那只数码管的对应位输出 1，显示完从 0 到 9 的一组数字后，下一位输出 1，选通下一个数码管，输出十位数，这与广告灯闪烁控制过程类似。

3. 总体程序见教材或电子材料

```
#include "reg51.h"
#define uchar unsigned char
uchar display_code[]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0
xF8,0x80,0x90,0x88,0x83,0xC6,0xA1,0x86,0x8E};
uchar display_data[8]={0,0,0,0,0,0,0,0};

uchar time[]={25,0,0};

sbit beep = P3^0;

void delay1(void)
{
    uchar i,j;
    for(i=250;i>0;i--)
        for(j=200;j>0;j--);
}
```

```
void delay(void)
{
    uchar i;
    for(i=250;i>0;i--);
}

void display()
{
    uchar i,k;
    k=0x80;
    for(i=0;i<8;i++)
    {
        P2=0;
        P0=display_code[display_data[i]];
        P2=k;
        k=k>>1;
        delay();
    }
    P2=0;
}
```

```
void calc()
{
    --time[0];
}
```

```
void convert()
{
    display_data[0]=time[0]/10;
    display_data[1]=time[0]%10;
}
```

```
void main(void)
{
    uchar i;
    while(1)
    {
        calc();
        if(time[0]==0)
        {
            while(1)
            {
                beep=0;
                delay1();
                beep=1;
                delay1();
            }
        }

        for(i=0;i<250;i++)
        {
            display();
        }
        convert();
    }
}
```

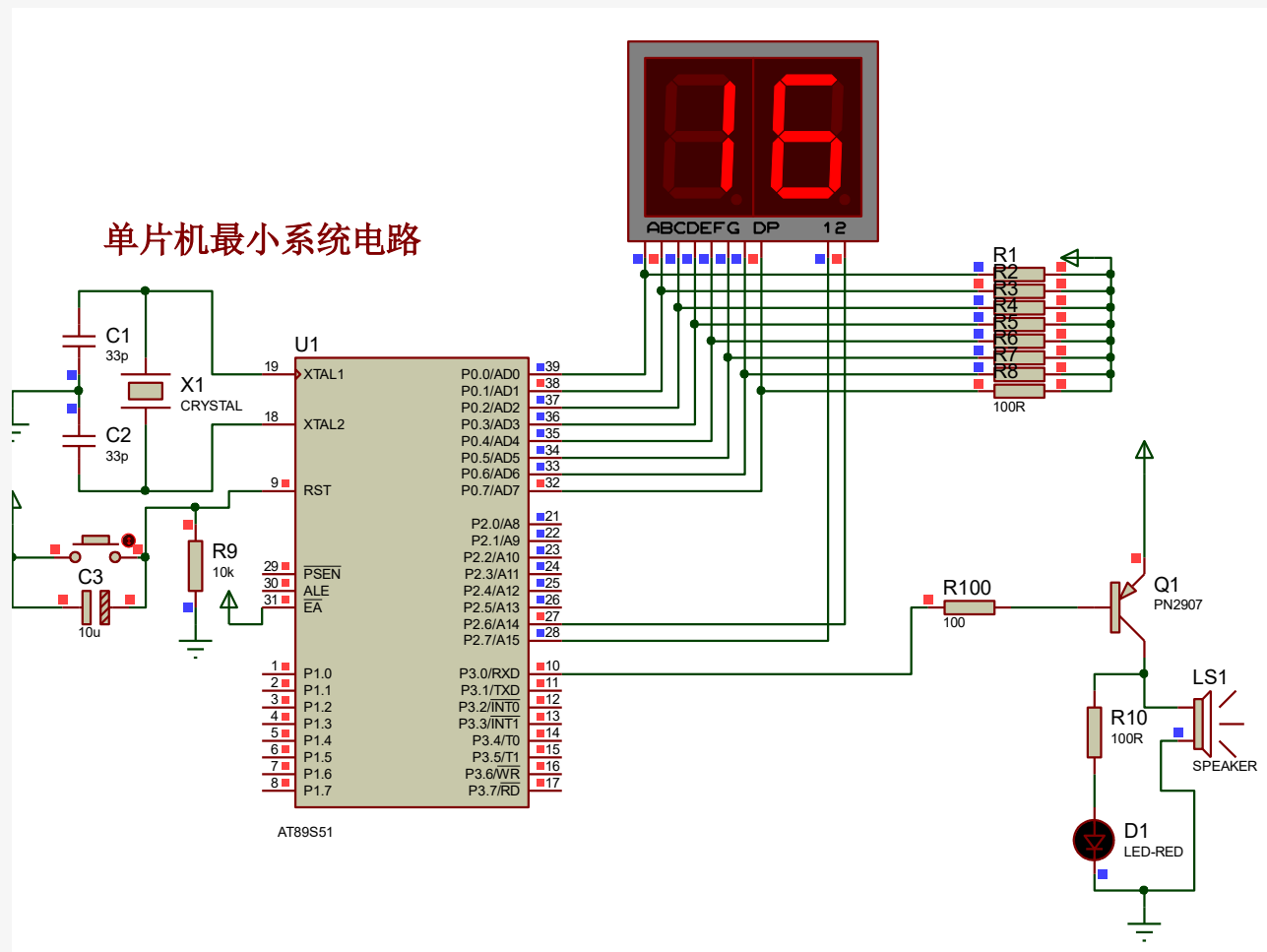
1.5 软件仿真

1. 使用 Proteus 软件，绘制硬件电路图，并保存到指定的位置。

2. 使用 KeilC51 建立一个 24jishuqi 工程项目，建立一个 .c 文件，在编辑区中输入上述的源程序，并以“24jishuqi.c”为文件名存放 24jishuqi 文件夹中；进行编译，排除所有程序故障，直到无错误为止，目标代码文件“24jishuqi.hex”。

3. 使用 Proteus 打开绘制好的按键电路设计图，双击电路图中 AT89S51 单片机，把编译好的“24jishuqi.hex”文件下载到单片机中。点击模拟调试按钮，就可以看到数码管从 24 秒倒时计数，当计数到 0 时，就进行声光报警。

1.5 软件仿真



1.6 项目总结

本项目实现了篮球 24 秒倒计时系统的设计，介绍了数码管结构、驱动方式和显示原理，并阐述了数码管与单片机的接口设计以及编程控制的方法。

项目提升

- 1、数码管从内部结构上分为哪几类？
 - 2、数码管从显示方式上分为哪几种？
 - 3、数码管静态显示方式原理是什么？它有什么优缺点？
 - 4、数码管动态显示方式原理是什么？它有什么优缺点？
 - 5、数码管显示原理是什么？
 - 6、在使用共阴极数码管的显示电路中，如果把共阴极数码管修改成共阳极数码管，能正常显示吗？为什么？电路和程序如何修改？
 - 7、如何设计 0-999 的计数显示电路？
-

知识目标

1. 掌握 LCD 液晶显示器的结构；
2. 掌握 LCD 液晶显示器的工作原理
3. 掌握 LCD 液晶显示器与单片机的接口技术

；

技能目标

4. 掌握单片机控制 LCD 液晶显示器的编程方法
1. 会阐述 LCD 液晶显示器的结构、工作原理；
2. 会设计 LCD 液晶显示器与单片机的接口的电路；

能力目标

3. 会使用单片机控制 LCD 液晶显示器的编程方法
1. 能分析设计任务，正确使用 LCD 液晶显示器；
2. 能够正确设计 LCD 液晶显示器与单片机的接口电路；
2. 能使用 PROTUES 软件绘制电路原理图；
3. 能使用 Keil 软件编译程序对 LCD 液晶显示器进行控制，并与 PROTUES 软件联调，实现控制电路仿真；
4. 能够完成秒表电路设计和仿真；
5. 能够使用 LCD 液晶显示器去解决实际问题。

课时建议

4 课时

1.1 任务提出

设计一个秒表系统，显示时间为 0~59s，满 60s 时，秒表自动清零并重新从 0 开始显示。在液晶屏幕上第一行显示 “Welcome”，第二行显示秒表时间。



1.2 方案设计

1. 单片机选择。采用 ATMEL 公司的 AT89C51 作为系统控制器的 CPU 方案。单片机算术运算功能强，软件编程灵活、自由度大，可以用软件编程实现各种算法和逻辑控制，并且由于其功耗低、体积小、技术成熟和成本低等优点，使其在各个领域应用广泛。

2. 显示电路的选择。根据题目要求，能够直观显示倒计时时间，采用 LCD 液晶显示。采用 LCD 液晶显示屏，液晶显示屏显示的功能强大，可显示大量文字，图形，显示多样，清晰可见。与数码管显示相比，再直观程度上和亮度清晰度上都存在和多优势，并且现在液晶显示已成为主流，被人们普遍接受，符合大众口味。

1.3 硬件设计

1. LCD 模块

液晶显示器在日常生活中经常见到，应用在很多的电子产品中，如计算器、万用表、电子表等。液晶显示器显示的主要是数字、专用符号和图形液晶显示器与发光管、数码管相比具有显示质量高、数字输出接口、体积小、重量轻和功耗低等优点。

液晶显示器是利用液晶的物理特性显示的，在其显示区域施加一定的电压，从而可以控制其显示相应的图形、汉字和字符等。

液晶显示器按其显示方式分为段式、字符式、点阵式等；按其驱动方式来分，可以分为静态驱动、单纯矩阵驱动和主动矩阵驱动。

1.3 硬件设计

1.1 LCM1602

字符型液晶显示模块是一种专门用于显示字母、数字、符号等的点阵式 LCD，其型号通常为

XXX1602、XXX1604、XXX2002、XXX2004 等。其中 XXX 为商标名称，16 代表液晶每行可显示 16 个字符；其后的 02 表示共有两行；20 代表液晶每行可显示 20 字符，其后的 02 表示共有两行。LCM1602 液晶点阵字符显示器用 5×7 点阵图形来显示西文字符，可显示 2 行×16 个西文字符。LCM1602 的主要由三部分组成：LCD 控制器、LCD 控制器、LCD 显示装置

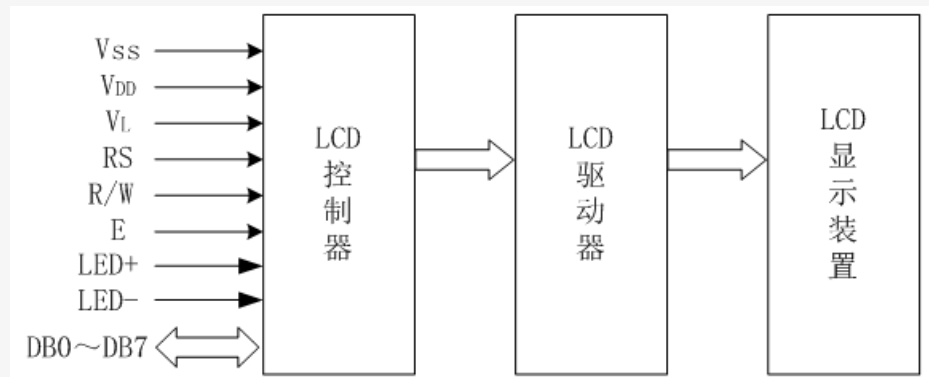


图 9-5 LCM1602 组成

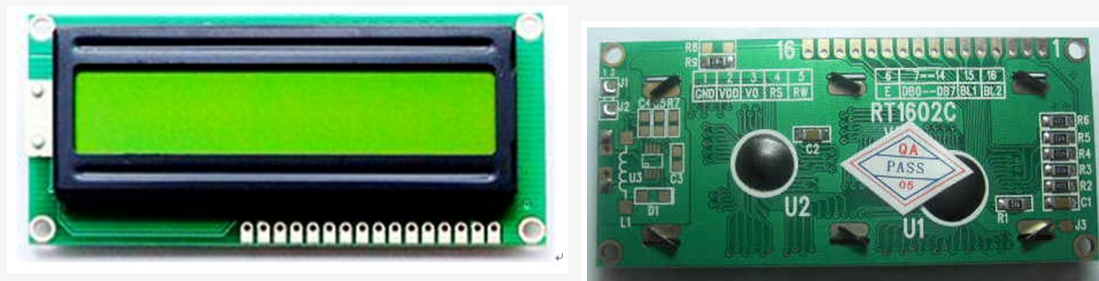


图 9-6 LCM1602 实物图

1.3 硬件设计

1.1 LCM1602 引脚及功能

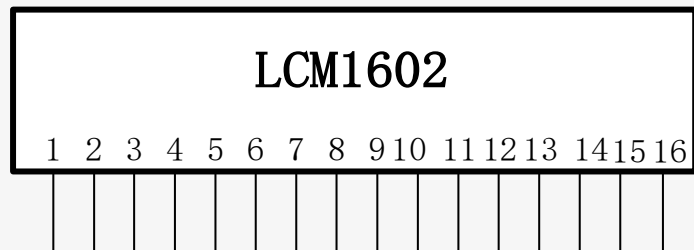


图 9-7 LCM1602 引脚示意图

其主要参数如下：

1602LCD 主要技术参数：

- (1) 显示容量:16×2 个字符
- (2) 芯片工作电压:4.5-5.5V
- (3) 工作电流:2.0mA(5.0V)
- (4) 模块最佳工作电压:5.0V
- (5) 字符尺寸

寸:2.95×4.35(W×H)mm

表 9-2 LCM1602 管脚说明

管脚编号	名称	方向	功能	操作
1	V _{SS}	电源	电源接地	0V
2	V _{DD}	电源	电源正极	+5V
3	VL	电源	LCD 亮度调整电压输入	电压越低, 屏幕越亮
4	RS	输入	寄存器选择信号	1=选择数据寄存器 0=选择指令寄存器
5	R/W	输入	Read/Write	1=Read/读取 0=Write/写入
6	E	输入	LCD 响应信号	1=响应 LCD 0=禁用 LCD
7~10	DB0~DB3	输入/输出	低四位总线	可用 4bit 输入数据、命令及地址
11~14	DB4~DB7	输入/输出	高四位总线	配合 DB0~DB3 的 8 位输入数据、命令及地址
15	LED+	输入	背光源正极	+5V
16	LED-	输入	背光源负极	0V

1.3 硬件设计

1.2 LCM1602 指令

LCM 模块使用单片机来控制十分方便。LCM 中有一个 IR（指令寄存器）和 DR（数据寄存器 DR），受 RS 引脚控制。在对指令寄存器或数据寄存器的存取时均需检查 LCM 内部的忙碌标志 BF 的状态，如果此时处于忙碌状态，不允许接收任何控制命令。只有当不忙碌时，才可以写入指令寄存器或数据寄存器。LCM 控制器共有 11 种指令

表 9-3 LCD 指令控制码表

序号	指令操作	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	执行
1	清除显示屏	0	0	0	0	0	0	0	0	0	*	1.64us
2	光标回到原点	0	0	0	0	0	0	0	0	1	*	1.64us
3	进入模式设定	0	0	0	0	0	0	0	1	I/D	S	40us
4	显示 ON/OFF	0	0	0	0	0	0	1	D	C	B	40us
5	显示/光标移位	0	0	0	0	0	1	S/C	R/L	*	*	40us
6	功能设定	0	0	0	0	1	DL	N	F	*	*	40us
7	设定字符发生器 (CGRAM)地址	0	0	0	1	A5	A4	A3	A2	A1	A0	40us
8	设置(DD RAM)显示地址	0	0	1	A6	A5	A4	A3	A2	A1	A0	40us
9	忙碌标志位 BF	0	1	BF	D6	D5	D4	D3	D2	D1	D0	40us
10	写入数据寄存器 (显示数据)	1	0	D7	D6	D5	D4	D3	D2	D1	D0	40us
11	读取数据寄存器	1	1	D7	D6	D5	D4	D3	D2	D1	D0	40us

1.3 硬件设计

1.2 LCM1602 指令

(1) 清除显示屏 (Clear Display)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

指令码 01H，光标复位到地址 00H 位置。

1.3 硬件设计

1.2 LCM1602 指令

(2) 光标回原点 (左上角)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	×

指令代码为 02H，光标复位，光标返回到地址 00H。

1.3 硬件设计

1.2 LCM1602 指令

(3) 设定进入模式

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

光标和显示模式设置指令：

I/D: 光标移动方向，高电平右移，低电平左移，当 I/D=1，表示当读或写完一个数据操作后，地址指针 AC 加 1，且光标加 1(光标右移一格)

I/D=0，表示当读或写完一个数据操作后，地址指针 AC 减 1，且光标减 1(光标左移一格)

S: 屏幕上所有文字是否左移或者右移。高电平表示有效，低电平则无效。

S=1 表示当写一个数据操作时，整屏显示左移 (I/D=1) 或右移 (I/D=0)，以得到光标不移动而屏幕移的效果。

S=0 表示当写一个数据操作时，整屏显示不移动。

1.3 硬件设计

1.2 LCM1602 指令

(4) 显示屏开关 (Display ON/OFF)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

D: 控制整体显示的开与关，高电平表示开显示，低电平表示关显示

C: 控制光标的开与关，高电平表示有光标，低电平表示无光标

B: 控制光标是否闪烁，高电平闪烁，低电平不闪烁。

1.3 硬件设计

1.2 LCM1602 指令

(5) 显示 / 光标移位 (Display/Cursor shift)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	×	×

“×”表示该位可以为0或1。

S/C: 滚动对象的选择，高电平时移动显示的文字，低电平时移动光标。

S/C=1 画面滚动

S/C=0 光标滚动

R/L: 滚动方向的选择

R/L=1 向右滚动

R/L=0 向左滚动

1.3 硬件设计

1.2 LCM1602 指令

(6) 功能设定 (Function Set)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	×	×

“×”表示该位可以为 0 或 1。

DL: 低电平时为 4 位总线，高电平时为 8 位总线

N: 低电平时为单行显示，高电平时双行显示

F: 低电平时显示 5x7 的点阵字符，高电平时显示 5x11 的点阵字符。

1.3 硬件设计

1.2 LCM1602 指令

(7) 字符产生器 RAM(CGRAM) 地址设定

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	A5	A4	A3	A2	A1	A0

设定下一个要读 / 写数据的 CGRAM 地址，地址由 (A5 ~ A0) 给出，可设定 00 ~ 3FH 共 64 个地址。

1.3 硬件设计

1.2 LCM1602 指令

(8) 显示数据 RAM(DDRAM) 地址设定

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	A6	A5	A4	A3	A2	A1	A0

设定下一个要读 / 写数据的 DDRAM 地址，地址由 (A6 ~ A0) 给出，可设定 00 ~ 7FH 共 128 个地址。N = 0 一行显示 A6 ~ A0 = 00 ~ 4FH，N = 1 两行显示，首行 A6 ~ A0 = 00H ~ 2FH 次行 A6 ~ A0 = 40H ~ 67H

(9) 忙碌标志 / 地址计数器读取 (Busy Flag/Address Counter)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	BF	A6	A5	A4	A3	A2	A1	A0

BF: 忙标志位，高电平表示忙，此时模块不接收命令或者数据，如为低，表示不忙。

1.3 硬件设计

1.2 LCM1602 指令

(10) 写入数据寄存器

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

先设定 CGRAM 或 DDRAM 地址，再将数据写入 DB7 ~ DB0 中，以使 LCD 显示出字型，也可使使用者创的图形存入 CGRAM 中。

(11) 读取数据寄存器

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

先设定好 CGRAM 或 DDRAM 地址，再读取其中的数据。

1.3 硬件设计

1.2 LCM1602 指令

(10) 写入数据寄存器

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

先设定 CGRAM 或 DDRAM 地址，再将数据写入 DB7 ~ DB0 中，以使 LCD 显示出字型，也可使使用者创的图形存入 CGRAM 中。

(11) 读取数据寄存器

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

先设定好 CGRAM 或 DDRAM 地址，再读取其中的数据。

1.3 硬件设计

1.3 初始化

(1) 初始化设置

- 1) 显示器清屏
- 2) 显示器开 / 关及光标设置
- 3) 显示光标移动设置

(2) 数据控制

控制器内部设有一个数据地址指针，用户可通过它们来访问内部全部 80 字节 RAM

。

1) 数据指针设置

数据地址指针：80H + 地址码 (00H ~ 27H , 40H ~ 67H)

- 2) 读数据
- 3) 写数据

1.3 硬件设计

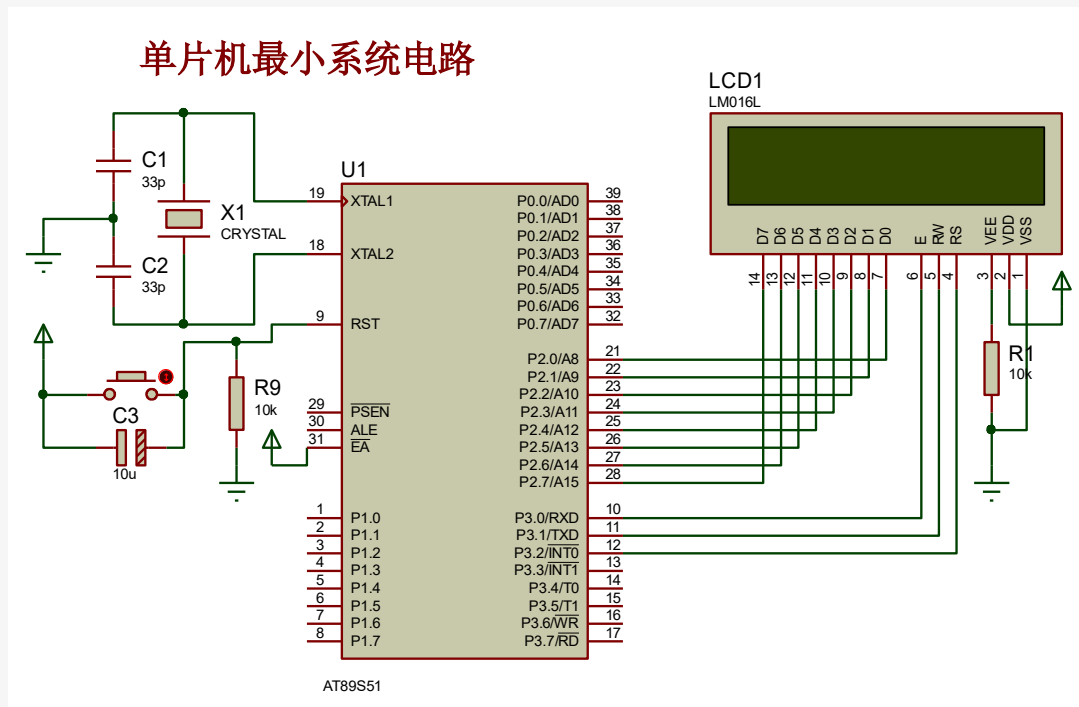
1.4 复位过程

- (1) 延时 15ms
- (2) 写指令 38H (不检测忙信号)
- (3) 延时 5ms
- (4) 写指令 38H (不检测忙信号)
- (5) 延时 5ms
- (6) 写指令 38H (不检测忙信号)
- (7) 以后每次写指令、读 / 写数据操作之前均需检测忙信号
- (8) 写指令 38H : 显示模式设置
- (9) 写指令 08H : 显示关闭, 不显示光标
- (10) 写指令 01H : 显示清屏
- (11) 写指令 06H : 显示光标移动设置, 写一个字符后, $N = 1$, 地址加 1, 光标加 1, $S = 0$, 整屏显示。
- (12) 写指令 0CH : 显示开及光标设置, $D = 1$ 开显示, $C = 1$ 不显示光标, $B = 0$ 光标不闪。

1.3 硬件设计

2. 硬件电路

秒表系统由单片机最小系统和 LCM 组成。单片机的 P2 端口接 LCM 的 D0-D7（要注意接线的正确性，单片机端口的低引脚 P20 接 LCM 的 D0 引脚，依次类推），E、R/W 和 RS 分别接单片机的 P30、P31 和 P32 引脚。



1.3 硬件设计

3. 软件设计

1. LCD 显示过程

软件程序设计要严格按照 LCM1602 液晶显示器的读操作时序和写操作时序来编写，具体见图 9-9、图 9-10 和表 9-4。单片机所用的晶体振荡器频率不同，在编写延时程序时延时参数要作适当的修改，使之符合 LCM1602 的时序要求。

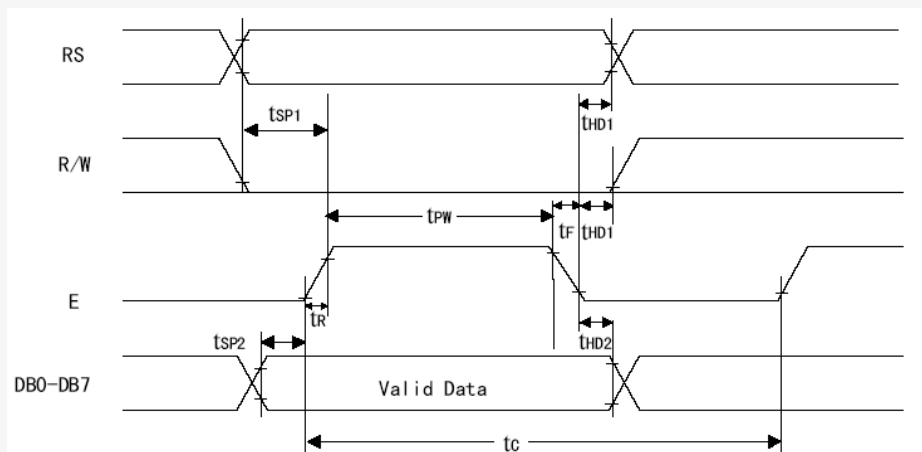


图 9-9 写操作时序

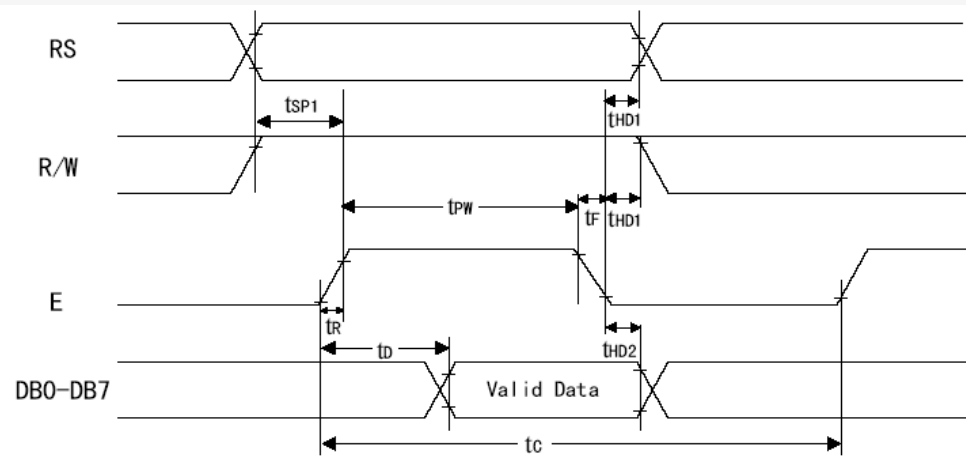


图 9-10 读操作时序

1.3 硬件设计

3. 软件设计

1. LCD 显示过程

表 9-4 时序参数表

时序参数	符号	极限值			单位	测试条件
		最小值	典型值	最大值		
E 信号周期	t _C	400	—	—	ns	引脚 E
E 脉冲宽度	t _{PW}	150	—	—	ns	
E 上升沿/下降沿时间	t _R , t _F	—	—	25	ns	
地址建立时间	t _{SP1}	30	—	—	ns	引脚 E、RS、R/W
地址保持时间	t _{HD1}	10	—	—	ns	
数据建立时间(读操作)	t _D	—	—	100	ns	引脚 DB0~DB7
数据保持时间(读操作)	t _{HD2}	20	—	—	ns	
数据建立时间(写操作)	t _{SP2}	40	—	—	ns	
数据保持时间(写操作)	t _{HD2}	10	—	—	ns	

要使液晶显示器 LCM1602 显示出数字，必须把数字 0 — 9 变换成 ASCII 码表中对应的 ASCII 码。从 ASCII 码表中可以看出，要把数字 0 — 9 转换成 ASCII 码，只要把数字 0 — 9 分别加上十六进制数 0x30 就行了。

2. 总体程序见教材或电子材料

```
#include "reg51.h"
#define uchar unsigned char
#define uint unsigned int

uchar code table[]="Welcome!";

void delay1(void)
{
    uchar i,j,k;
    for(i=250;i>0;i--)
        for(k=10;k>0;k--)
            for(j=200;j>0;j--);
}

//LCD 定义引脚
sbit lcden=P2^6; // E
sbit lcdwrite=P2^5; // R/W
sbit lcdrs=P2^4; // RS

void delay(uint z)
{
    uint x,y;
    for(x=z;x>0;x--)
        for(y=110;y>0;y--);
}
```

```
//lcd 写命令
void write_com(uchar com)
{
    lcdrs=0;
    lcden=0;
    P0=com;
    //P2=com;
    delay(5);
    lcden=1;
    delay(5);
    lcden=0;
}

//lcd 写指令
void write_data(uchar da)
{
    lcdrs=1;
    lcden=0;
    P0=da;
    //P2=da;
    delay(5);
    lcden=1;
    delay(5);
    lcden=0;
}
```

```

void init()
{
    uchar num;
    lcdwrite=0;
    lcden=0;
    write_com(0x38); //16*2 显示 ,5*7 点阵 ,8
    write_com(0x0c); // 显示开, 关光标
    write_com(0x06); // 移到光标
    write_com(0x01); // 清除 LCD 的显示内容
    for (num=0;num<8;num++)
    {
        write_data(table[num]);
        delay(5);
    }
}

```

位数据

```

// 主程序
void main ()
{
    uchar sec=0;
    init();
    while(1)
    {
        sec++;
        if(sec==60) sec=0;
        write_com(0x80+0x
        40);
        write_data(sec/10+
        0x30);
        write_data(sec%10
        +0x30);
        delay1();
    }
}

```

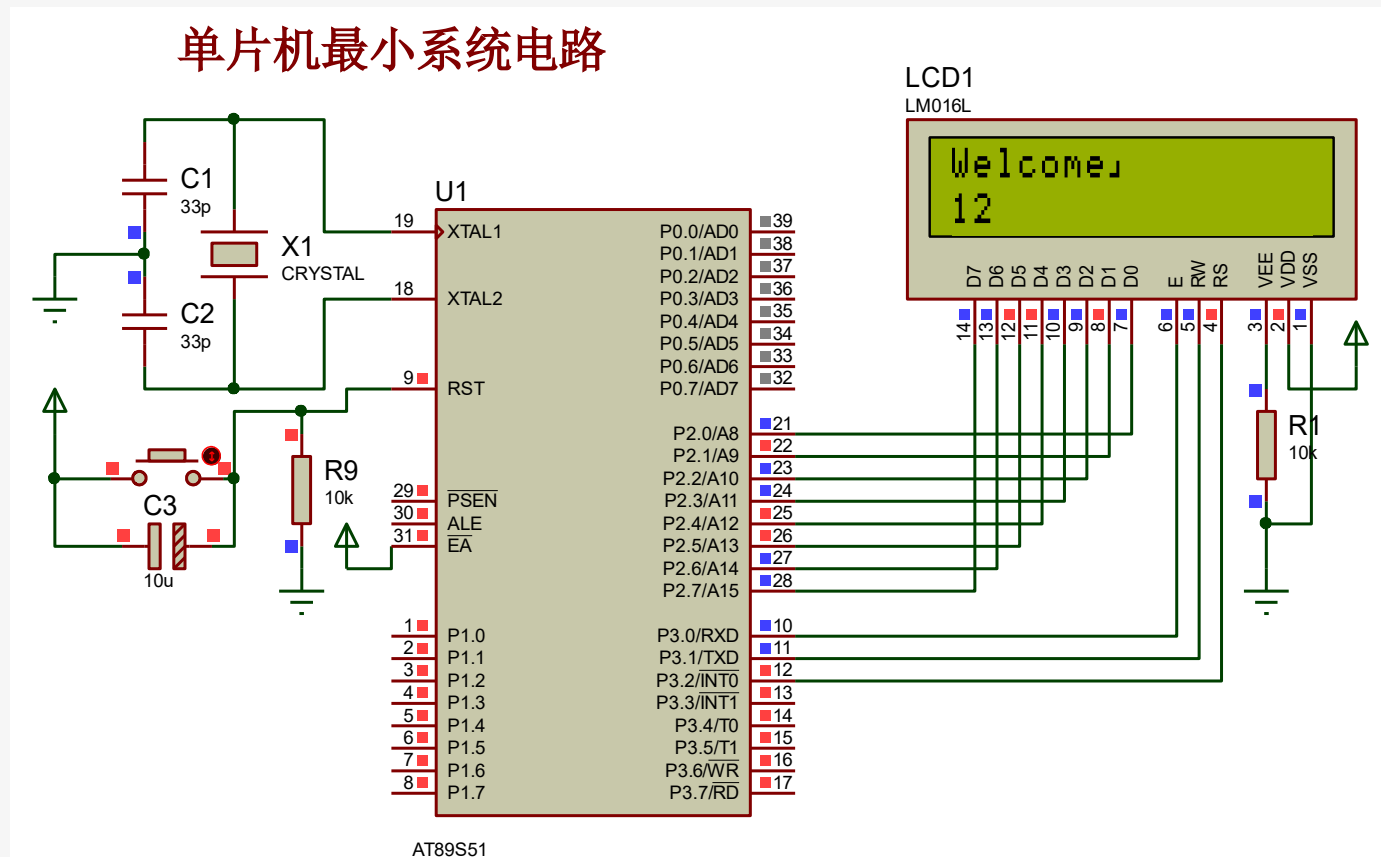
1.5 软件仿真

1. 使用 Proteus 软件，绘制的硬件电路图，并保存到指定的位置。

2. 使用 KeilC51 建立一个秒表工程项目，建立一个 .c 文件，在编辑区中输入上述的源程序，并以“miaobiao.c”为文件名存放 miaobiao 文件夹中；进行编译，排除所有程序故障，直到无错误为止，目标代码文件“miaobiao.hex”。

3. 使用 Proteus 打开绘制好的按键电路设计图，双击电路图中 AT89S51 单片机，把编译好的“miaobiao.hex”文件下载到单片机中。点击模拟调试按钮，就可以看到液晶屏幕上第一行显示“Welcome”，第二行显示秒表时间，显示时间为 0~59s，满 60s 时，秒表自动清零并重新从 0 开始显示。

1.5 软件仿真



1.6 项目总结

本项目实现了秒表的设计和实现，介绍了 LCM 结构、引脚功能、显示原理以及指令功能，并阐述了 LCM 与单片机的接口设计以及编程控制的方法。

项目提升

- 1、LCD1602 的清屏命令是什么？
- 2、LCM 复位过程是什么？
- 3、LCM1602 有哪几部分组成？
- 4、液晶显示的类型有几种？
- 5、LCD1602 能显示中文吗？
- 6、LCD1602 各个引脚的作用是什么？