

## AT89C51 单片机 LED 汉字显示屏设计方案

本文研究了基于 AT89C51 单片机  $16 \times 16$  LED 汉字滚动显示屏的设计与运用 Proteus 软件的仿真实现。主要介绍了 LED 汉字显示屏的硬件电路、汇编程序设计及调试、Proteus 软件仿真等方面的内容,本显示屏的设计具有体积小、硬件少、电路结构简单及容易实现等优点。能帮助广大电子爱好者了解汉字的点阵显示原理,认识单片机的基本结构、工作原理及应用方法,并提高单片机知识技术的运用能力。

### 0 引言

LED 显示屏是利用发光二极管点阵模块或像素单元组成的平面式显示屏幕。它具有发光率高、使用寿命长、组态灵活、色彩丰富以及对室内外环境适应能力强等优点。并广泛的用于公交汽车、商店、体育场馆、车站、学校、银行、高速公路等公共场所的信息发布和广告宣传。LED 显示屏发展较快,本文讲述了基于 AT89C51 单片机  $16 \times 16$  LED 汉字点阵滚动显示的基本原理、硬件组成与设计、程序编写与调试、Proteus 软件仿真等基本环节和相关技术。

### 1 硬件电路组成及工作原理

本产品采用以 AT89C51 单片机为核心芯片的电路来实现,主要由 AT89C51 芯片、时钟电路、复位电路、列扫描驱动电路(74HC154)、 $16 \times 16$  LED 点阵 5 部分组成,如图 1 所示。其中,AT89C51 是一种带 4kB 闪烁可编程可擦除只读存储器(Falsh Programmable and Erasable Read OnlyMemory, FPEROM)的低电压、高性能 CMOS 型 8 位微处理器,俗称单片机。该器件采用 ATMEL 高密度非易失存储器制造技术制造,工业标准的 MCS-51 指令集和输出管脚相兼容。由于将多功能 8 位 CPU 和闪烁存储器组合在单个芯片中,能够进行 1 000 次写/擦循环,数据保留时间为 10 年。他是一种高效微控制器,为很多嵌人式控制系统提供了一种灵活性高且价廉的方案。因此,在智能化电子设计与制作过程中经常用到 AT89C51 芯片。时钟电路由 AT89C51 的 18, 19 脚的时钟端(XTAI 1 及 XTAL2)以及 12 MHz 晶振 X、电容 C2、C3 组成,采用片内振荡方式。复位电路采用简易的上电复位电路,主要由电阻 R, R2, 电容 C, 开关 K 组成,分别接至 AT89C51 的 RST 复位输入端。LED 点阵显示屏采用  $16 \times 16$  共 256 个象素的点阵,可通过万用表检测发光二极管的方法测试判断出该点阵的引脚分布。

我们把行列总线接在单片机的 I/O 口,然后把上面分析到的扫描代码送入总线,就可以得到显示的汉字了。但是若将 LED 点阵的行列端口全部直接接入 89S51 单片机,则需要使用 32 条 I/O 口,这样会造成 I/O 口资源的耗尽,系统也再无扩充的余地。因此,我们在实际应用中只是将 LED 点阵的 16 条行线直接接在 P0 口和 P2 口,至于列选扫描信号则是由 4-16 线译码器 74HC154 来选择控制,这样一来列选控制只使用了单片机的 4 个 I/O 口,节约了很多 I/O 口资源,为单片机系统扩充使用功能提供了条件。考虑到 P0 口必需设置上拉电阻,我们采用  $4.7 \text{ k}\Omega$  排电阻作为上拉电阻。汉字扫描显示的基本过程是这样的:通电后由于电阻 R, 电容 c1 的作用,使单片机的 RST 复位脚电平先高后低,从而达到复位;

之后，在 C、C3、X 以及单片机内部时钟电路的作用下，单片机 89C51 按照设定的程序在 P0 和 P2 接口输出与内部汉字对应的代码电平送至 LED 点阵的行选线（高电平驱动），同时在 P1.1, P1.2, P1.3, P1.4 接口输出列选扫描信号（低电平驱动），从而选中相应的像素 LED 发光，并利用人眼的视觉暂留特性合成整个汉字的显示。再改变取表地址实现汉字的滚动显示。

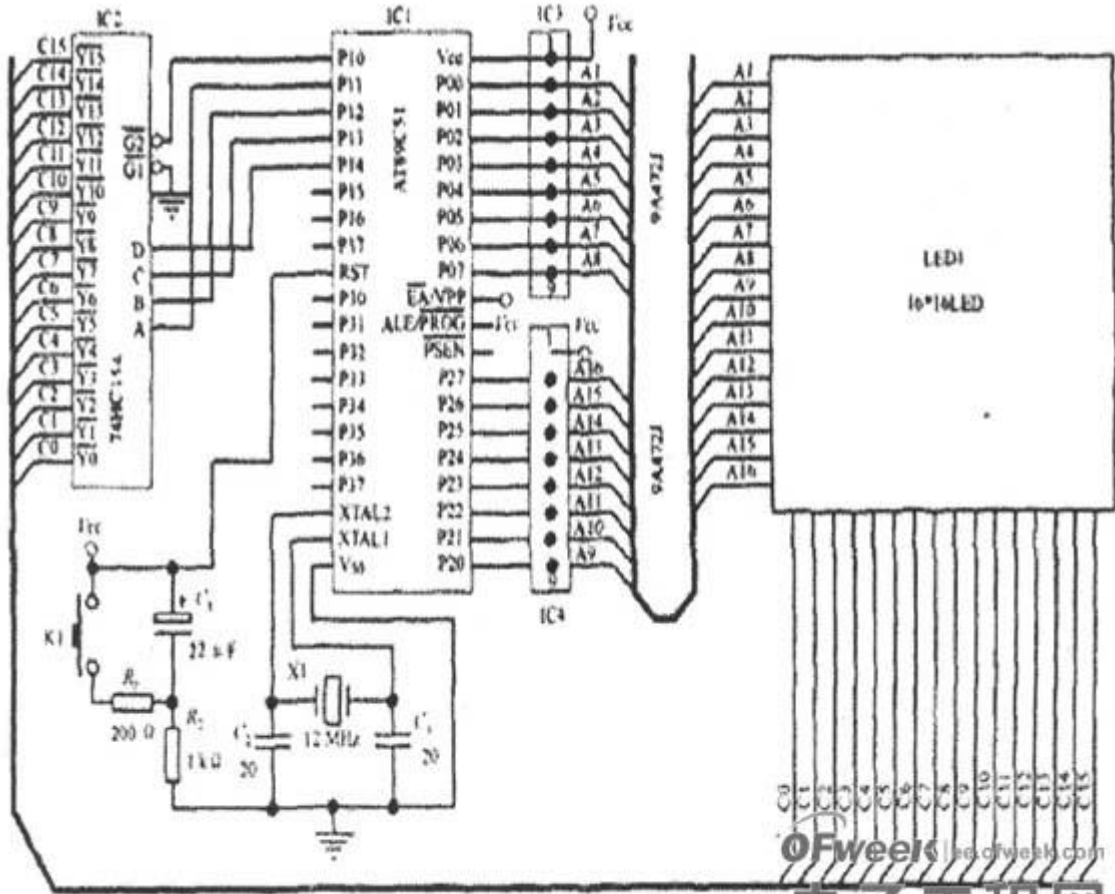


图1 硬件电路组成

## 2 汉字的点阵显示原理及字库代码获取方法

我们以 UCDO5 中文宋体字库为例，每一个字由 16 行 16 列的点阵组成显示。即国标汉字库中的每一个字均由 256 点阵来表示。我们可以把每一个点理解为一个像素，而把每一个字的字形理解为一幅图像。事实上这个汉字屏不仅可以显示汉字，也可以显示在 256 像素范围内的任何图形。如查用 8 位的 AT89C51 单片机控制，由于单片机的总线由 8 位，一个字需要拆分为 2 个部分，如图 2 所示

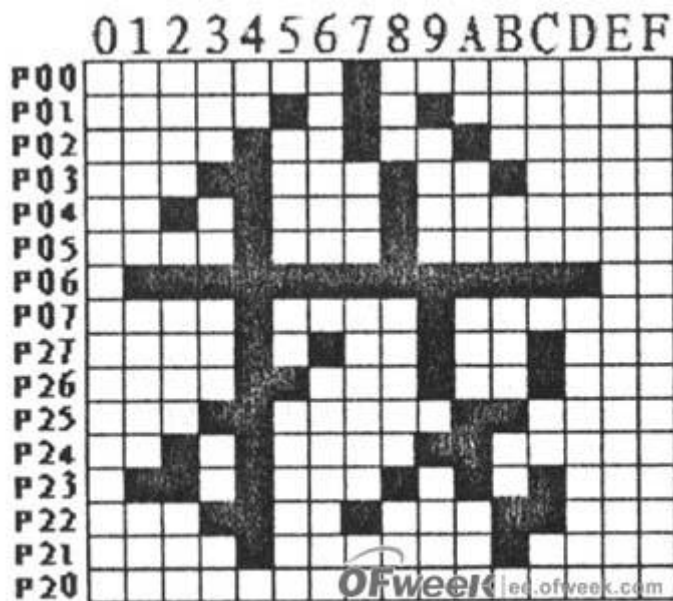


图2 汉字显示原理

为了弄清楚汉字的点阵组成规律，首先通过列扫描方法获取汉字的代码。汉字可拆分为上部 and 下部，上部由  $8 \times 16$  点阵组成，下部也由  $8 \times 16$  点阵组成。本例通过列扫描方法首先显示左上角的第一列的上半部分，即第 0 列的 P00~P07 口，方向为 P00 到 P07，显示汉字“我”时，为全灭，第一列的下半部分也为全灭。第二列的上半部分 P06、点亮，由上往下排列，为：P0.0 灭，P0.1 灭，P0.2 灭 P0.3 灭，P0.4 灭，P0.5 灭，P0.6 亮，P0.7 灭。即二进制 00000010，转换为十六进制为 02h。上半部第二列完成后，继续扫描下半部的第二列，为了接线的方便，我们仍设计成由上往下扫描，即从 P27 向 P20 方向扫描，从图 3 可以看到，这一列 P23 亮，即为 00001000，十六进制则为 08h。依照这个方法转向第三列、第四列，……，直至第十六列的扫描，一共扫描 32 个 8 位，可以得出汉字“我”的扫描代码为：

00H, 02H, 08H, 06H, 28H, 02H, 24H, 22H

0FCH, 3FH, 24H, 2 1H, 20H, 10H, 3CH, 08H

0E2H, 07H, 20H, 0AH, 0E4H, 11H, 0A8H, 20H

20H, 30H, 00H, 00H, 00H, 00H, 00H, 00H

由这个原理可以看出，无论显示何种字体或图像，都可以用这个方法来分析出他的扫描代码从而显示在屏幕上。上述方法虽然能够让我们弄清楚汉字点阵代码的获取过程，但是依靠人工方法获取汉字代码是一件非常繁琐的事情。为此，我们经常采用字库软件查找字符代码，软件打开后输入汉字，点“检取”，十六进制数据的汉字代码即可自动生成，把我們所需要的竖排数据复制到程序中即可，如图 3 所示。

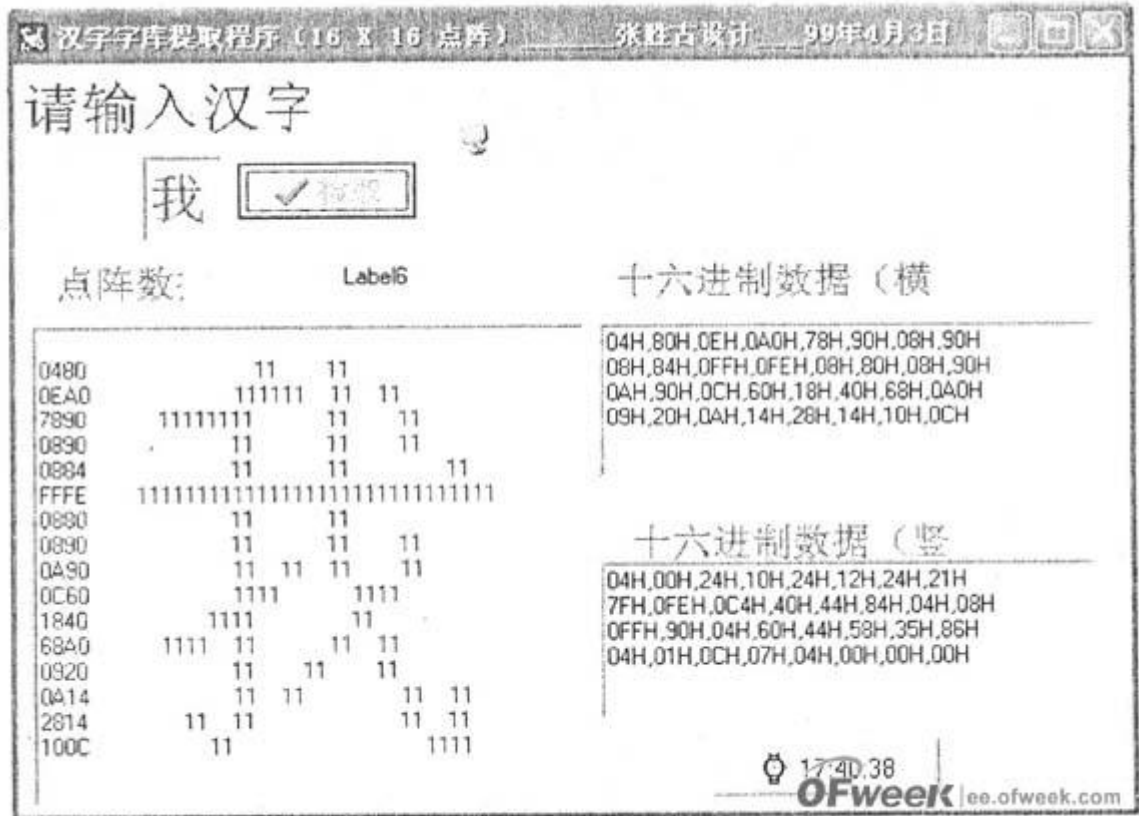


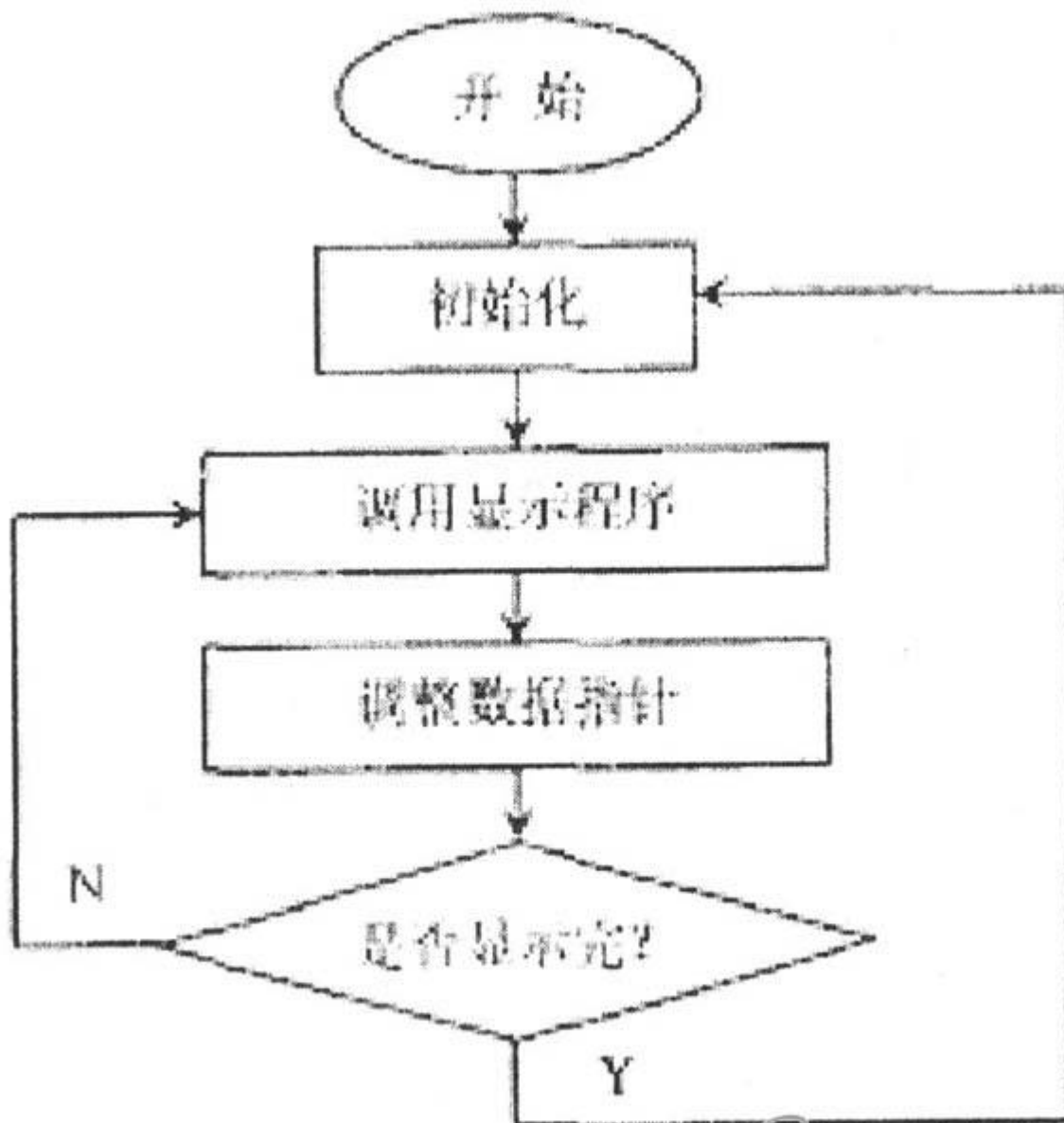
图3 字库提取程序 电子工程网

可见，汉字点阵显示一般有点扫描、行扫描和列扫描3种。为了符合视觉暂留要求，点扫描方法的扫描频率必须大于  $16 \times 64 - 1024$  Hz，周期小于 1 ms 即可。行扫描和列扫描方法的扫描频率必须大于  $16 \times 8 - 128$  Hz，周期小于 7.8 ms 即可，但是一次驱动一列或一行（8 颗 LED）时需外加驱动电路提高电流，否则 LED 亮度会不足。

### 3 在 Keil 环境中程序设计与调试

软件程序主要由开始、初始化、主程序、字库组成。其中主程序的流程图如图 4 所示。下面的程序能够用来实现滚动显示“我爱你——祖国”。汉字的显示。





OFweek | ee.ofweek.com

图4 主程序流程图 **电子工程网**

程序清单如下:

```
=====
;
ORG 0000H
AJMP MAIN
ORG 0030H
```

```
=====
;
MAIN:MOV DPTR, #TAB ;字码表初址赋值
      MOV R1, #00H ;列控制码
      MOV R4, #96 ;移动“ ”及“单片机位真”6个字符,
```

共96列

OFweek 电子工程网

OFweek 电子工程网

```

CM: MOV R5, #5 ;每屏反复显示5次
    MOV R3, #16 ;列数
C1: MOV R2, #0 ;取码指针
C16: MOV P0, #00H
    MOV P2, #00H ;关显示
    CLR P3.0
    MOV A, R2
    MOVC A, @A+DPTR;取当前列显示字码的第一个字节
    MOV P0, A ;送1~8行控制口
    INC R2
    MOV A, R2
    MOVC A, @A+DPTR;取当前列的显示字码的第二个字节
    MOV P2, A ;送9~15行控制口
    INC R2
    MOV P1, R1 ;送列控制码
    INC R1
    ACALL D1MS ;显示2MS
    ACALL D1MS
    DJNZ R3, C16 ;一屏16列是否显示完
    MOV R3, #16
    DJNZ R5, C1 ;未显示5次, 继续
    INC DPTR;一屏反复显示5次完, 字码表初值加2
    INC DPTR
    DJNZ R4, CM ;96列未移动完, 继续
    AJMP MAIN;96列移动完, 返回, 重新从" "开始显示

```

```

;
D1MS: MOV R6, #2
    MOV R7, #248
    DJNZ R7, $
    DJNZ R6, $-4
    RET

```

```

;
TAB:
DB 00H, 00H, 20H, 06H, 28H, 02H, 24H, 22H, 0FCH, 3FH, 24H, 21H, 20H, 10H, 3CH, 08H;
DB 0E2H, 07H, 20H, 0AH, 0E4H, 11H, 0A8H, 20H, 20H, 30H, 00H, 00H; "我"

```

在 keil 软件中完成程序编写、调试和编译之后，生成能让单片机运行的 Hex 文件，如图 5 所示。



图5 在keil软件中编程和调试

#### 4 元器件选择

本设计所需元器件如表 1 所示。



表 1 元器件选择

| 元件编号                            | 规格参数              | 作用简介        |
|---------------------------------|-------------------|-------------|
| U1                              | AT89C51           | 核心控制芯片      |
| LED1                            | 16×16LED 点阵       | 组成汉字显示屏     |
| IC2                             | 74HC154           | 4 线-16 线译码器 |
| IC3, IC4                        | 1 kΩ 电阻           | 上拉电阻        |
| R <sub>1</sub>                  | 200 Ω             |             |
| R <sub>2</sub>                  | 1 kΩ              | 复位电路        |
| C <sub>1</sub>                  | 22 μF/16 V        |             |
| K <sub>1</sub>                  | 按键开关              |             |
| X <sub>1</sub>                  | 12 MHz 晶振         | 时钟电路        |
| C <sub>2</sub> , C <sub>3</sub> | 20 pF             |             |
| PCB 板                           | 10 cm × 10 cm 环氧板 |             |

## 5 运用 Proteus 软件仿真 LED 汉字显示屏

Proteus 与其它单片机仿真软件不同的是, 它不仅能仿真单片机外围电路或没有单片机参与的其它电路的工作情况, 也能仿真单片机 CPU 的工作情况。因此在仿真和程序调试时, 是从工程的角度直接看程序运行和电路工作的过程和结果。从某种意义上讲 Proteus 仿真, 基本接近与工程应用。本次基于 AT89C51 单片机 16×16LED 汉字滚动显示屏的设计已运用 Proteus 软件仿真实现, 如图 6 所示。

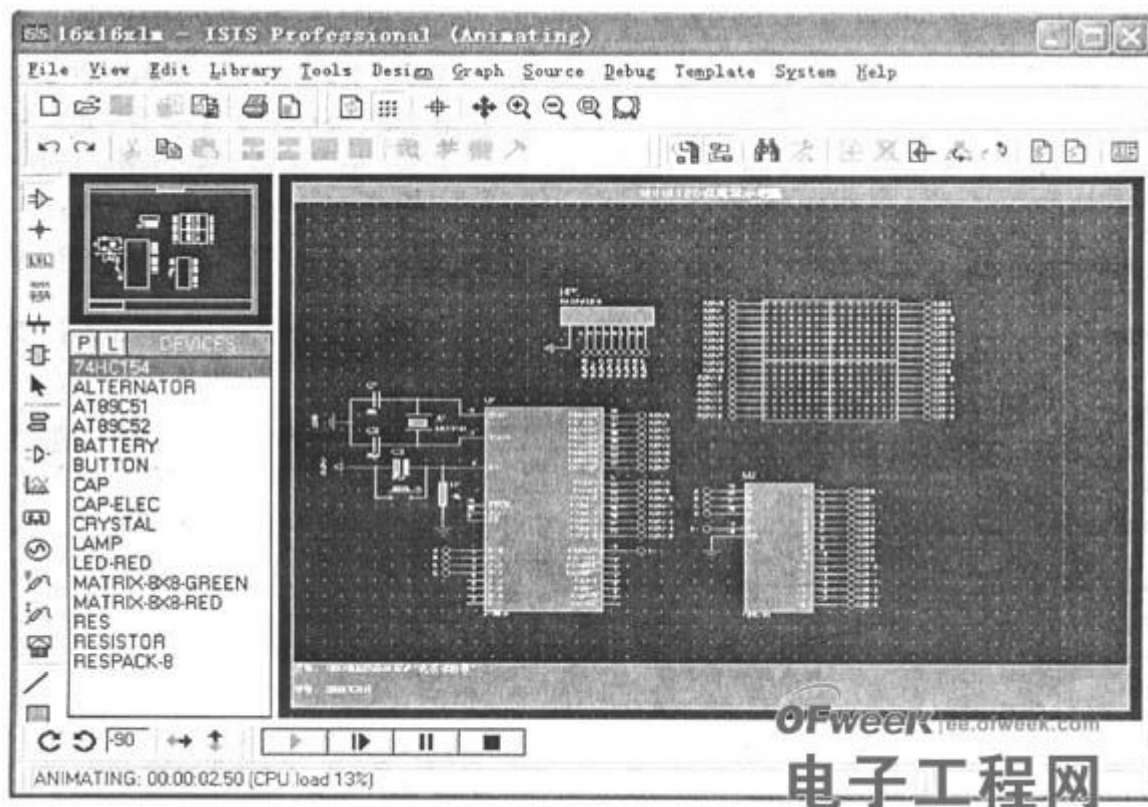


图6 Proteus仿真LED汉字显示屏

虽然本设计只使用了一块  $16 \times 16$  LED 点阵，电路简单，但是已经包涵了 LED 汉字滚动显示屏的电路基本原理、基本程序和 Proteus 软件仿真，只要扩展单片机的 10 接口，并增加一些 LED 点阵和相关芯片，就能设计出更大面积、更多花样的 LED 显示屏。因此本文对同类设计具有一定的理论和实践参考价值。