

16 × 16 点阵的制作

◎ 李 高

最近，在电子论坛里看见有很多朋友在找 DIY 点阵的资料。可是网上资料以 8 × 8 点阵居多，而根据我的制作经验，对于单片机的爱好者和 DIY 迷，16 × 16 点阵是比较合适的选择。

本点阵用的驱动芯片是 74HC595。74HC595 具有 8 位锁存、串—并移位寄存器和三态输出，每路最大输出 35mA 的电流(非恒流)。显示屏行业中常用 Motorola、Philips 及 ST 等厂家的芯片，其中 Motorola 的产品性能较好。

16 × 16 点阵需要 4 只 74HC595、256 只 LED (建议不要用高亮度的，一般的小 LED 效果最好)，用一个 51 单片机最小系统即可控制。外围元件很少，方便业余制作。

以下介绍我制作的 16 × 16 点阵的硬件和源程序，可以滚动显示字符和图形，显示内容可以根据 <http://www.c51bbs.com/c51blog/index.html> 提供的字模提取软件设置。本程序中显示的是：我爱电子(带图形)，最多可以有 200 多字。

源程序如下：

```
#include <at89x51.h>
#include <intrins.h>
sbit SDATA_595 = P2^0 ;
/* 串行数据输入 */
sbit SCLK_595 = P2^1;
```

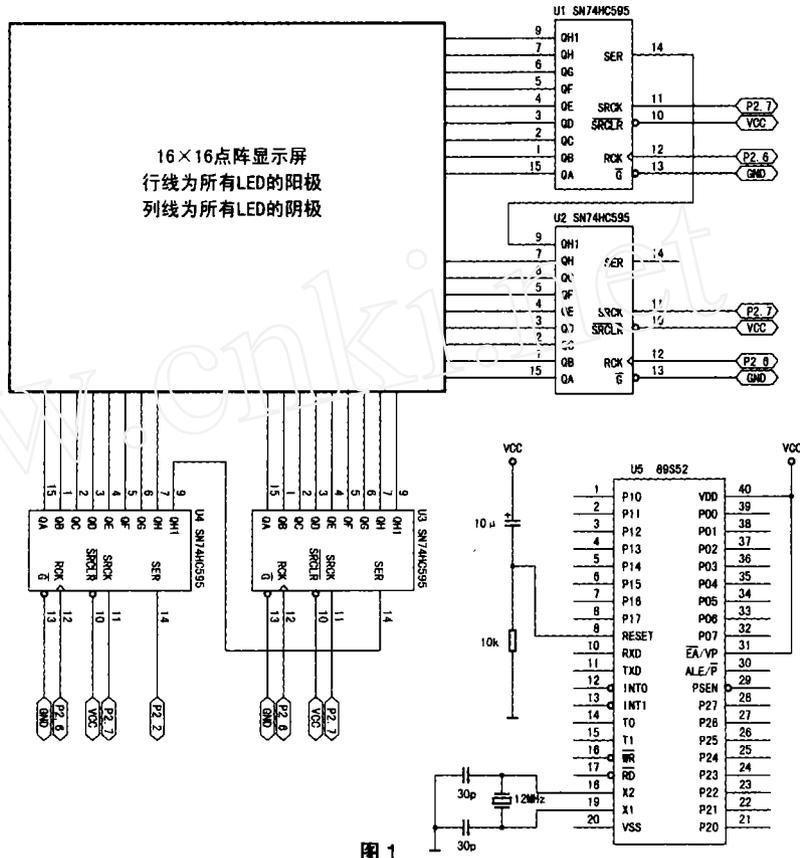


图 1

/* 移位时钟脉冲 */

sbit RCK_595 = P2^2; /* 输出锁存器控制脉冲 */

// 共阳段码表 // 字节的高位对应数码管的 H 段, 依次类推 //

static unsigned char code number_Tab[]={

/*-- 我, 宋体 12; 对应的点阵为: 宽 × 高 = 16 × 16 --*/

```
0x04,0x20,0x04,0x20,0x44,0x42,0x44,0x41,0x7F,0xFE,0x84,0x80,0x84,0x80,0x04,0x08,0x04,0x08,0xFF,0x10,0x04,0xE0,0x44,0x58,0x35,0x84,0x04,0x02,0x04,0x0F,0x00,0x00,
```

/*-- 爱, 宋体 12; 对应的点阵为: 宽 × 高 = 16 × 16 --*/

```
0x00,0x02,0x02,0x04,0x4D,0x05,0x69,0x09,0x59,0x32,0x49,0xE2,0x6F,0x54,0x59,0x54,0xC9,0x48,0x89,0x58,0x99,0x64,0xE9,0x44,0x89,0x20,0x09,0x03,0x0C,0x02,0x00,0x00,
```

/*-- 电, 宋体 12; 对应的点阵为: 宽 × 高 = 16 × 16 --*/

```

0x00,0x00,0x00,0x00,0x1F,0xF0,0x12,0x20,0x12,0x20,0x1
2,0x20,0x12,0x20,0xFF,0xFC,0x12,0x22,0x12,0x22,0x12,0
x22,0x12,0x22,0x1F,0xF2,0x00,0x02,0x00,0x0E,0x00,0x00,
/*-- 子, 宋体 12; 对应的点阵为: 宽 × 高 = 16 × 16
--*/
0x00,0x80,0x00,0x80,0x40,0x80,0x40,0x80,0x40,0x80,0x4
0,0x82,0x40,0x81,0x47,0xFE,0x48,0x80,0x50,0x80,0x60,0
x80,0x40,0x80,0x00,0x8
0,0x01,0x80,0x00,0x80,0x00,0x00,
/*-- 调入一幅图像: 心形 --*/
/*-- 宽 × 高 = 16 × 16 --*/
0x0F,0x00,0x10,0x80,0x20,0x40,0x20,0x20,0x10,0x10,0x0
8,0x08,0x04,0x04,0x02,0x02,0x02,0x02,0x04,0x04,0x08,0
x08,0x10,0x10,0x20,0x20,0x20,0x40,0x10,0x80,0x0F,0x00,
}; //H L = 共阳段码表
unsigned int i,j,k=0,b,count=0;
/* 向 74595 送一字节数据子程序 */
static void WR_595(unsigned char data num2 )
{
    unsigned char data count1;
    for(count1=0;count1<=7;
count1++)
    {
        if((num2&0x80)==0x80)
/* 最高位为 1,则向 SDATA_595 发送 1*/
        SDATA_595=1;
/* 送数据最高位 */
        else
            SDATA_595=0;
            num2<<=1;
/* 右移位 */
            SCLK_595=0;
/* 产生上升沿 */
            SCLK_595=1;
        } }
/* 打开锁存,更新 74595 输出子程序 */
static void Out_595(void)
{
    RCK_595=0;
    _nop_();
    _nop_();
    RCK_595=1; /* 上升沿锁存数据 */
} /* 显示函数 */
void disp(void)
{unsigned char move=0x80;
for(i=0;i<=30;i++)
{if(k<32+count)
{for(j=0;j<8;j++)
{ WR_595(number_Tab[k++]);
WR_595(number_Tab[k++]);
WR_595(~move);
WR_595(0xFF);
Out_595();
move>>=1;
if(move==0x00)
move=0x80;
}
for(j=0;j<8;j++)
{ WR_595(number_Tab[k++]);
WR_595(number_Tab[k++]);
WR_595(0xFF);
WR_595(~move);
Out_595();
move>>=1;
if(move==0x00)
move=0x80;
}}
k=count;
}}
/* 主程序 */
void main(void)
{ while(1)
{
    disp();
    count+=2;
    if(count==1600000)
    count=0;
}}

```