

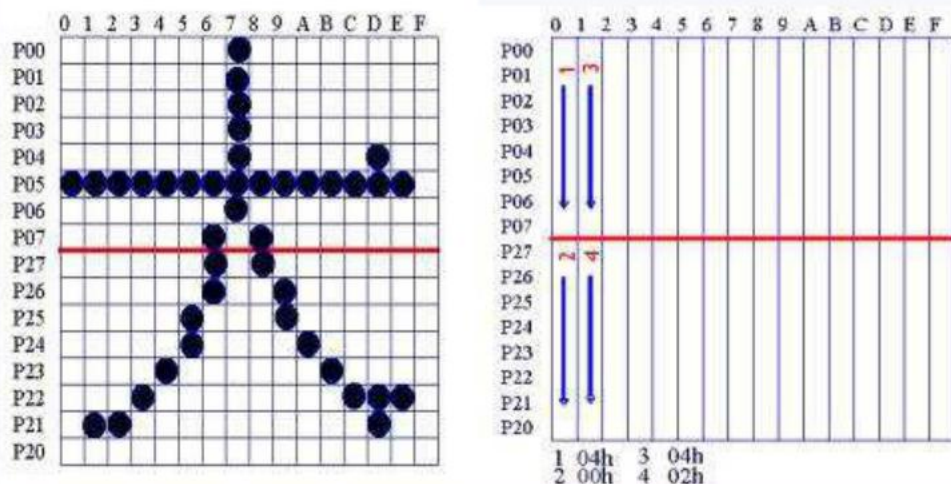
点阵式汉字 LED 显示屏电路原理与制作

2008-08-22 10:23:44 阅读(26) 发表评论

汉字显示屏广泛应用与汽车报站器，广告屏等。本文介绍一种实用的汉字显示屏的制作，考虑到电路元件的易购性，没有使用 8*8 的点阵发光管模块，而是直接使用了 256 个高亮度发光管，组成了 16 行 16 列的发光点阵。同时为了降低制作难度，仅作了一个字的轮流显示，实际使用时可根据这个原理自行扩充显示的字数。

1 汉字显示的原理：

我们以 UC DOS 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。



用 8 位的 AT89C51 单片机控制，由于单片机的总线为 8 位，一个字需要拆分为 2 个部分。

一般我们把它拆分为上部和下部，上部由 8*16 点阵组成，下部也由 8*16 点阵组成。

在本例中单片机首先显示的是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“大”时，p05 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 亮，p0.6 灭，p0.7 灭。即二进制 00000100，转换为 16 进制为 04h。

上半部第一列完成后，继续扫描下半部的第一列，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，从上图可以看到，这一列全部为不亮，即为 00000000，16 进制则为 00h。

然后单片机转向上半部第二列，仍为 p05 点亮，为 00000100，即 16 进制 04h。

这一列完成后继续进行下半部分的扫描，p21 点亮，为二进制 00000010，即 16 进制 02h。

依照这个方法，继续进行下面的扫描，一共扫描 32 个 8 位，可以得出汉字“大”

的扫描代码为：

04H, 00H, 04H, 02H, 04H, 02H, 04H, 04H

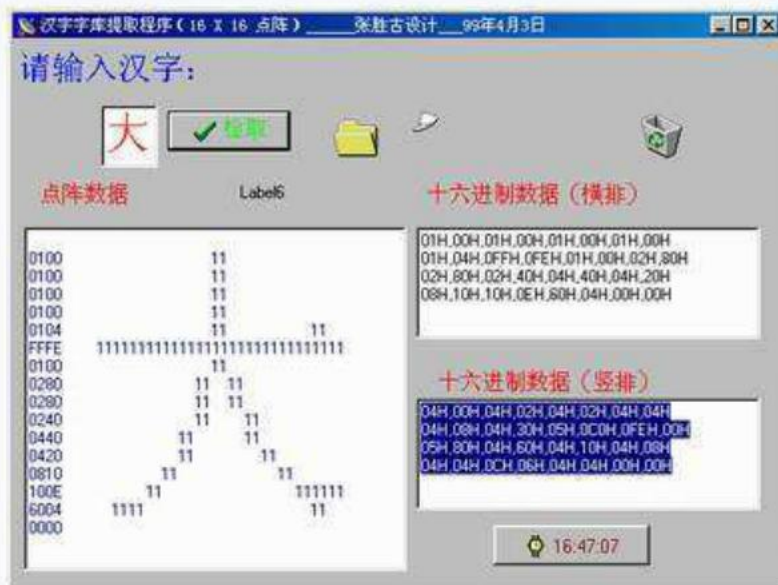
04H, 08H, 04H, 30H, 05H, 0C0H, 0FEH, 00H

05H, 80H, 04H, 60H, 04H, 10H, 04H, 08H

04H, 04H, 0CH, 06H, 04H, 04H, 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它的扫描代码从而显示在屏幕上。

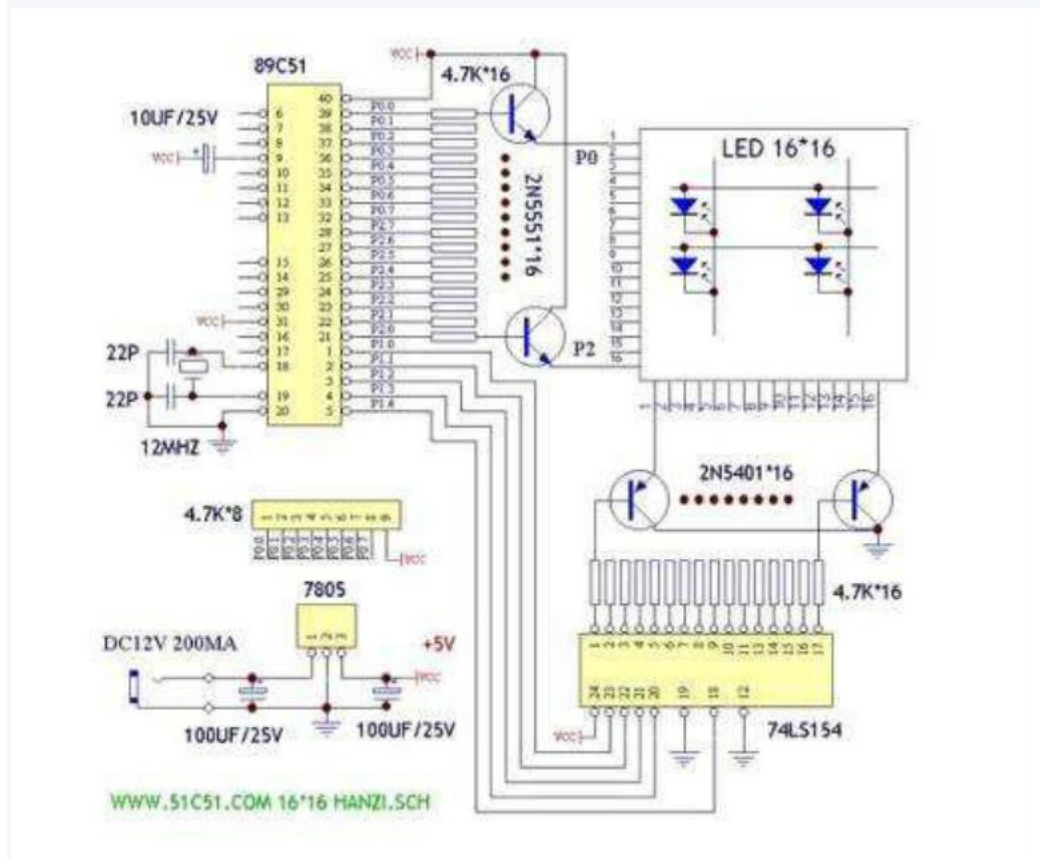
不过现在有很多现成的汉字字模生成软件，我们就不必自己去画表格算代码了。



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，把我们的需要的竖排数据复制到我们的程序中即可。

我们把行列总线接在单片机的 i0 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。在这个例子里，由于一共用到 16 行，16 列， 如果将其全部接入 89c51

单片机， 一共使用 32 条 io 口， 这样造成了 io 资源的耗尽， 系统也再无扩充的余地。实际应用中我们使用 4-16 线译码器 74ls154 来完成列方向的显示。 而行方向 16 条线则接在 p0 口和 p2 口。



程序清单：

双击代码全选

```

1  ORG  00H
2  LOOP: MOV  A, #0FFH ; 开机初始化，清除画面
3  MOV  P0, A    ; 清除 P0 口
4      ANL  P2, #00    ; 清除 P2 口
5  MOV  R2, #200
6  D100MS: MOV R3, #250 ; 延时 100 毫秒
7      DJNZ R3, $
8      DJNZ R2, D100MS
9      MOV 20H, #00H ; 取码指针的初值
10 L100:  MOV R1, #100 ; 每个字的停留时间
11 L16:  MOV R6, #16 ; 每个字 16 个码
12      MOV R4, #00H ; 扫描指针清零
13      MOV R0, 20H ; 取码指针存入 R0
    
```

```

14 L3:   MOV A, R4   ; 扫描指针存入 A
15      MOV P1, A  ; 扫描输出
16      INC R4     ; 扫描指针加 1, 扫描下一个
17      MOV A, R0  ; 取码指针存入 A
18      MOV DPTR, #TABLE ; 取数据表的上半部分的代码
19      MOVC A, @A+DPTR
20      MOV P0, A  ; 输出到 P0
21      INC R0     ; 取码指针加 1, 取下一个码。
22      MOV A, R0
23      MOV DPTR, #TABLE ; 取数据表下半部份的代码
24      MOVC A, @A+DPTR
25      MOV P2, A  ; 输出到 P2 口
26      INC R0
27      MOV R3, #02 ; 扫描 1 毫秒
28 DELAY2: MOV R5, #248 ;
29         DJNZ R5, $
30         DJNZ R3, DELAY2
31         MOV A, #00H ; 清除屏幕
32         MOV P0, A
33         ANL P2, #00H
34         DJNZ R6, L3 ; 一个字 16 个码是否完成?
35         DJNZ R1, L16 ; 每个字的停留时间是否到了?
36         MOV 20H, R0 ; 取码指针存入 20H
37         CJNE R0, #0FFH, L100 ; 8 个字 256 个码是否完成?
38         JMP LOOP ; 反复循环
39 TABLE :
40 ; 汉字“倚”的代码
41 db 01H, 00H, 02H, 00H, 04H, 00H, 1FH, 0FFH
42 db 0E2H, 00H, 22H, 00H, 22H, 0FCH, 26H, 88H
43 db 2AH, 88H, 0F2H, 88H, 2AH, 0FAH, 26H, 01H
44 db 63H, 0FEH, 26H, 00H, 02H, 00H, 00H, 00H
45 ; 以下分别输入天, 一, 出, 宝, 刀, 屠, 龙, 的代码, 略。
46 end

```

电路中行方向由 p0 口和 p2 口完成扫描, 由于 p0 口没有上拉电阻, 因此接一个 4.7k*8 的排阻上拉。如没有排阻, 也可用 8 个普通的 4.7k 1/8w 电阻。为提供负载能力, 接 16 个 2n5551 的 NPN 三极管驱动。

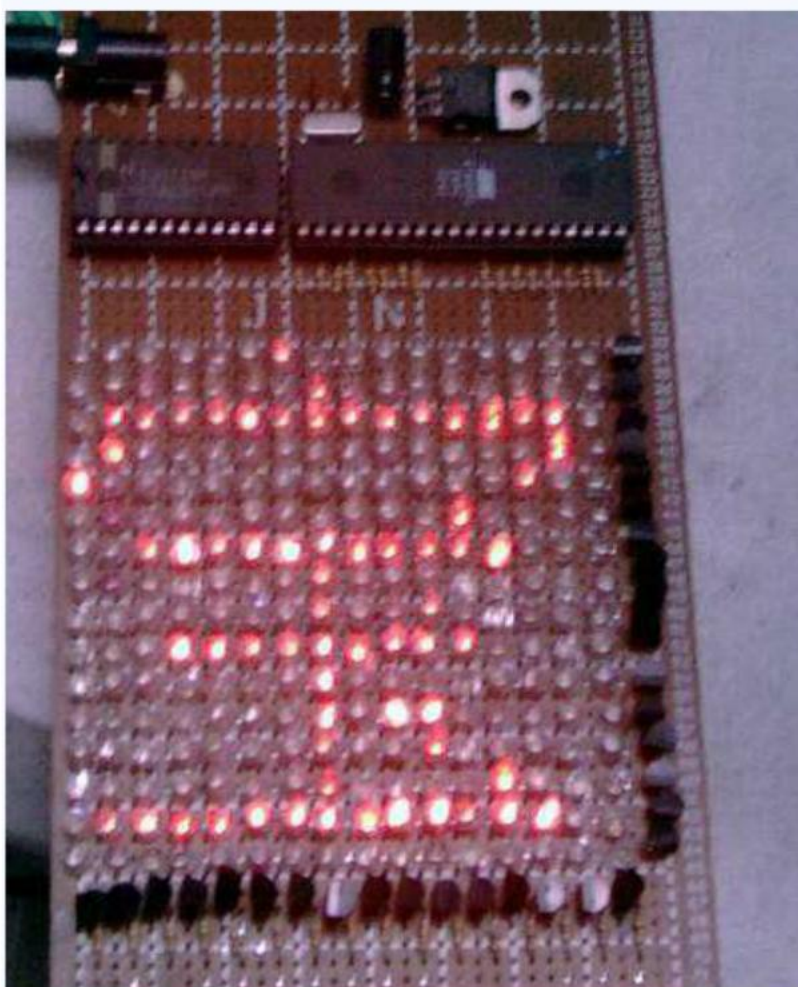
列方向则由 4—16 译码器 74LS154 完成扫描, 它由 89C51 的 P1.0—P1.3 控制。同样, 驱动部分则是 16 个 2N5401 的三极管完成的。

电路的供电为一片 LM7805 三端稳压器, 耗电电流为 100Ma 左右。

采用一块 12*20cm 的万能电路板，应当选用质量好些的发光管，（否则有坏点现象，更换起来较麻烦）首先将 256 个发光管插入电路板，注意插入方向，同时使高度一致，行方向直接焊接起来，列方向则搭桥架空焊接，完成后用万用表测试一下如有不亮的更换掉。

然后找一个电脑硬盘的数据线，截取所需的长度，分别将行，列线引出至电路的相关管脚即可。原理图为了简洁，故只画出了示意图，行列方向只画出了 2 个三极管，屏幕只画出 4 个发光管，实际上发光管为 256 只，三极管行列方向各 16 只，一共 32 只。焊接过程认真仔细一天时间即可完成全部制作。将程序编译后烧写入 89c51，插入 40pin Ic 座，即可看到屏幕轮流显示：“倚天一出宝刀屠龙”。

当然，你可将程序的汉字代码部分更换为您所需要的代码即可显示你所需要的汉字



元件清单：

名称	数量	规格
4.7k 1/8w	32	电阻
4.7k*8 排阻	1	
2n5551	16	小功率 NPN 三极管

2n5401	16	小功率 PNP 三极管
led	256	3mm 白发红高亮度
22P	2	瓷片电容
10uf/50v	1	电解电容
100uf/25v	2	电解电容
AT89C51	1	或 AT89S51
40pin Ic 座	1	插 89c51 用
12M	1	晶体
74LS154	1	或 74HC154
LM7805	1	稳压 IC
电源插座	1	
稳压电源	1	