

# 自制汉字库的原理与应用

## 自制汉字库的原理与应用

**摘要** 本文论述了自制汉字库的基本原理和具体制作步骤。并利用89C51 作为主控CPU，通过TFT控制板来点320240液晶屏。给出了在液晶屏上显示“国”字过程。

**关键词** 硬件汉字库 自制汉字库 液晶显示 89C51

虽然硬件汉字库具有信息量大、使用方便等特点，但是其必须扩展一个Flash 存储器，这样必然会带来硬件成本的提高和设计复杂度的增大。对于众多的仪器仪表、工业控制设备来讲，往往只需要几十个或几百个汉字就足够了，本文介绍了一种自制汉字库的方法，在硬件上节省了Flash 存储器。

### 一 汉字库的基本知识

汉字可以看成是有若干点所组成，如16x 16 点阵形式，每个点用一个二进制位表示，存1 的点当显示时可以在屏上显示一个亮点，存0 的点则在屏上不显示，这样就把存某字的16x 16 点阵信息直接用来在显示器上按上述原则显示则将出现对应的汉字，如一个国字的16x 16 点阵字模如图2 所示，当用存储单元存储该字模信息时将需32 个字节，在图2 的右边写出了该字模对应的字节值。

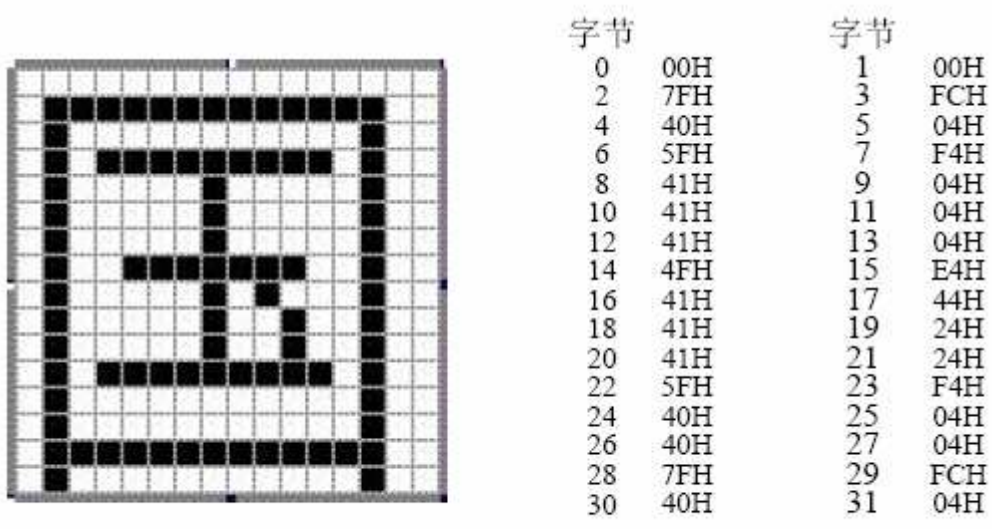


图2 “国”字的16×16点阵字模

### 二 自制汉字库的制作

自制汉字库的制作非常简单，分为以下几步：

1 首先统计所有需要用到的汉字，并进行编号，使每个汉字都有一个唯一的号码。编号从0开始，依次加一。

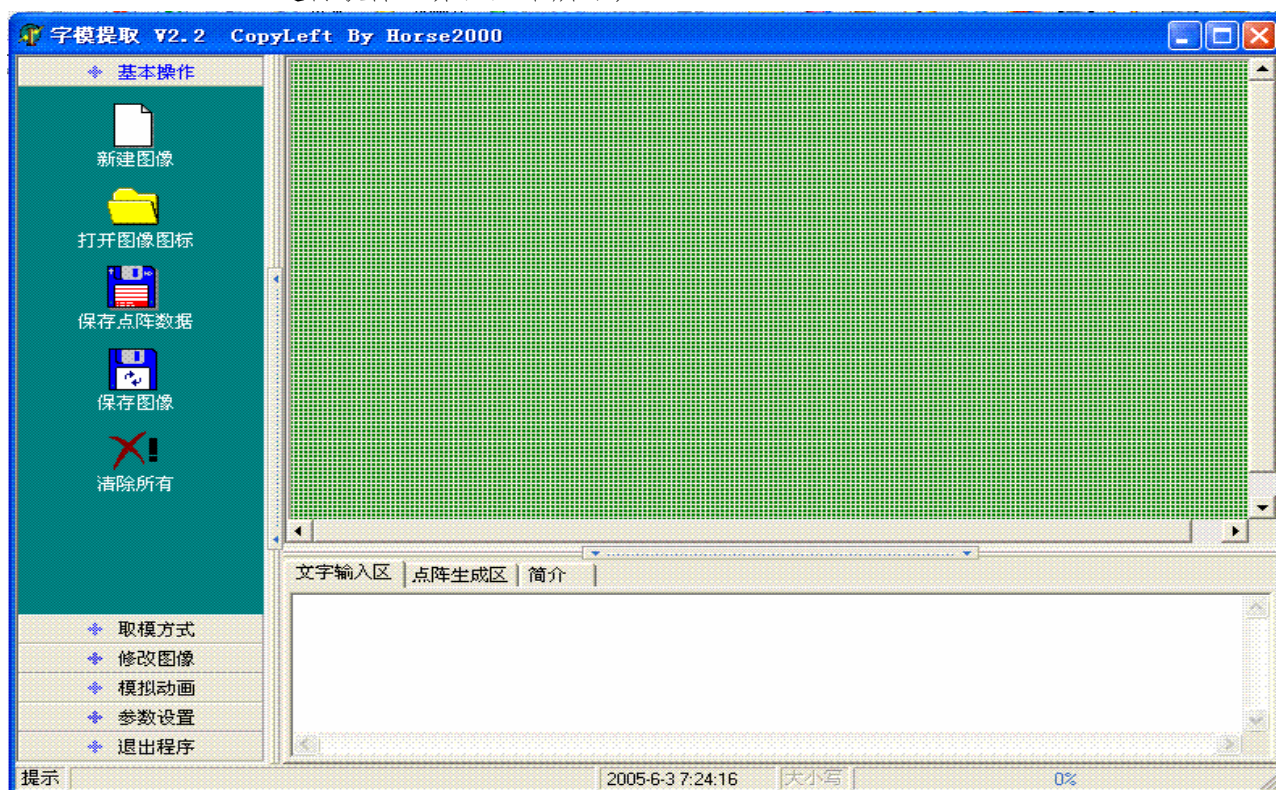
2 利用字库生成工具，将第一步统计的汉字按顺序逐个进行输入，按后选择字库的输出格式，转化后每个汉字都得到一个32字节的字库。字库格式有两种：C51

## 自制汉字库的原理与应用

格式和A51格式，如果用C语言要用C51格式，用汇编语言则用A51格式。这里给大家推荐一款字库提取软件：字库提取V2.2（作者：horse2000）。

软件使用说明：

（1）运行软件，界面如下所示；



（2）将你需要的一个或多个汉字输入到右下方的“文字输入区”内，输入完汉字后按ctrl+enter 键，表示输入结束，在右上角窗口内会出现输入的汉字；

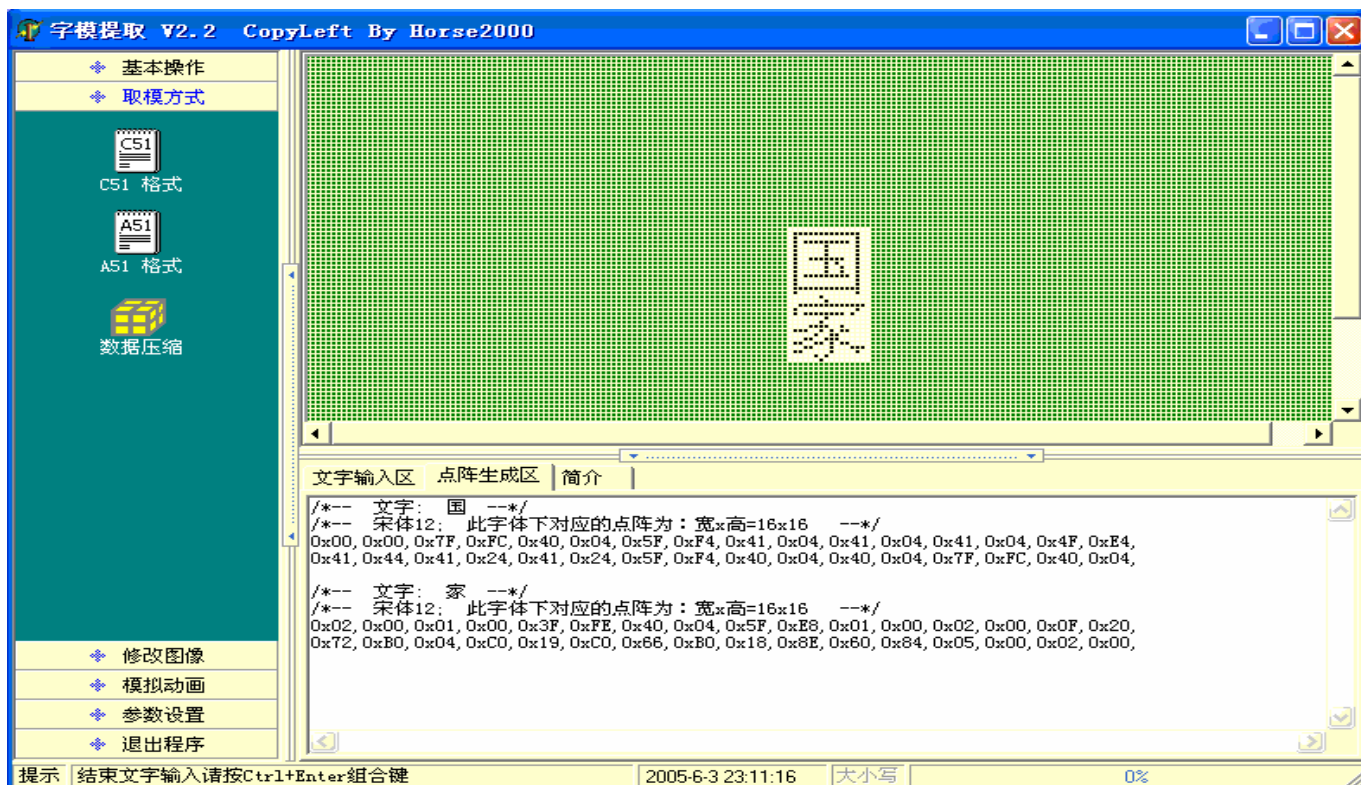




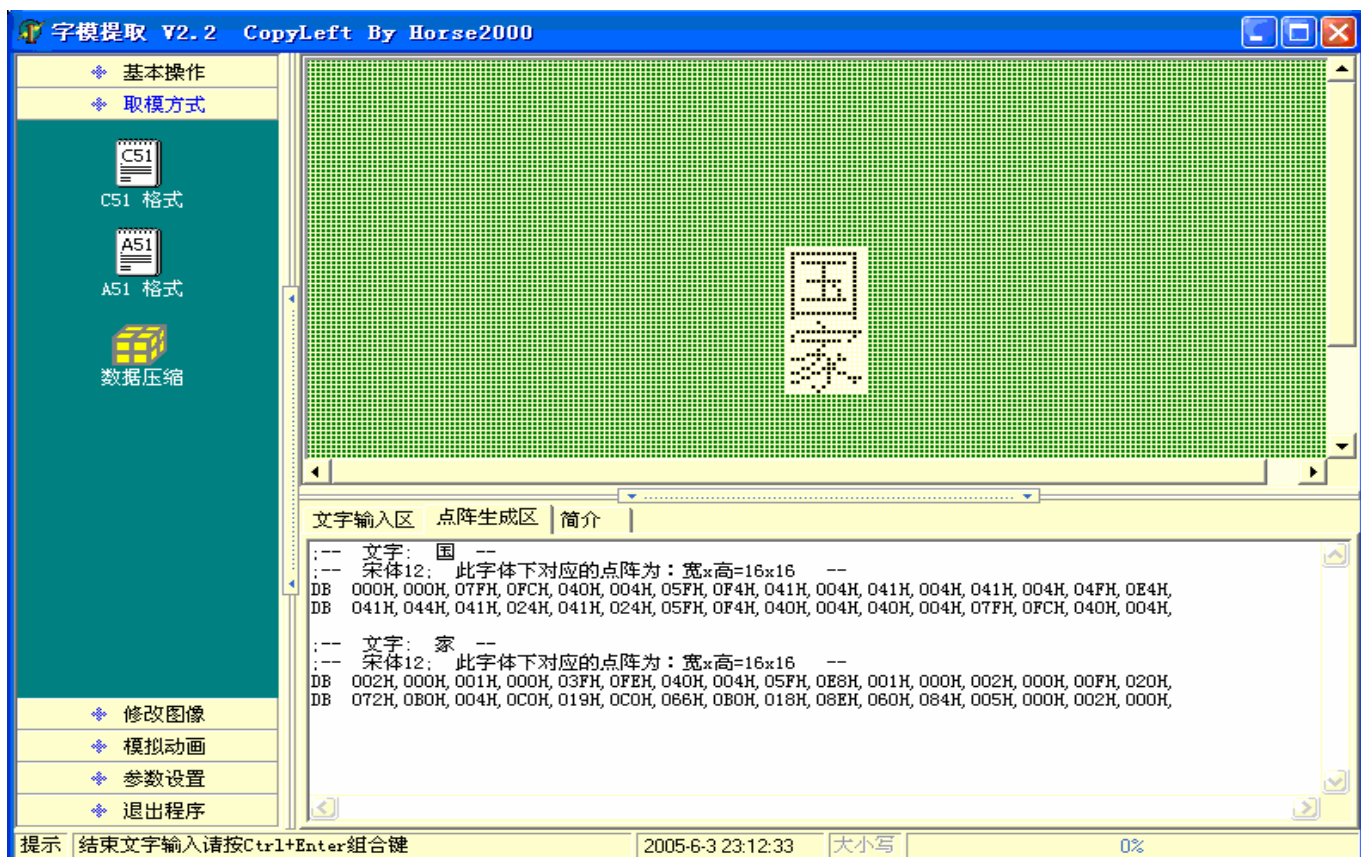
## 自制汉字库的原理与应用

(3) 点击左边窗口的“取模方式”，点击左边的“C51格式”或“A51格式”就会在右下脚的“点阵生成区”内生成字库点阵。

C51 格式:



A51 格式:



# 自制汉字库的原理与应用

(4) 将点阵生成区内的汉字点阵复制到你的程序里，就大功告成了。

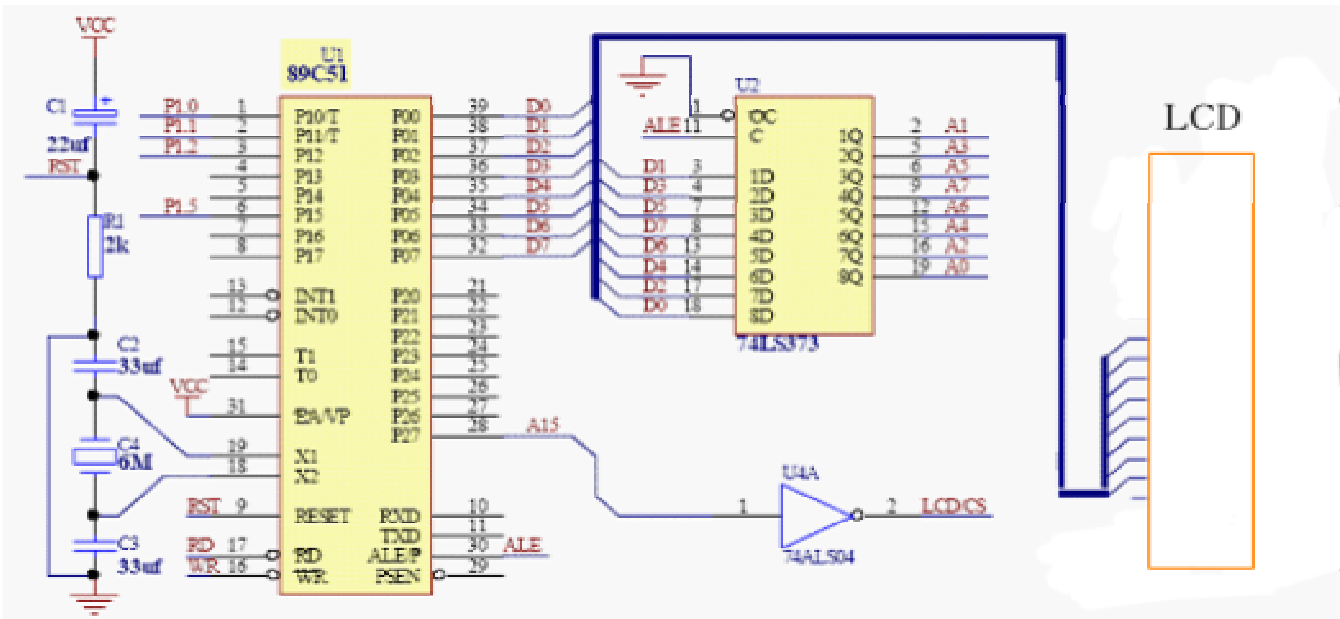
## 三 自制汉字库的使用电路

在本节中笔者使用 ATMEL 公司的产品 89C51 作为主控 CPU，使用 TFT 控制板来控制 320x240 点阵图形液晶显示器，将汉字“国家”送到液晶屏的 (2, 0) 处显示。

TFT 控制与单片机的连接信号定义如下：

引脚	符号	功能	备注
1	5V	液晶屏逻辑电源	5V
2	5V	液晶屏逻辑电源	5V
3	D0	数据总线	INOUT 3.3/5V
4	A0	地址线 0	IN 3.3/5V
5	D1	数据总线	INOUT 3.3/5V
6	A1	地址线 1	IN 3.3/5V
7	D2	数据总线	INOUT 3.3/5V
8	CSJ	片选信号，低电平对屏操作有效	IN 3.3/5V
9	D3	数据总线	INOUT 3.3/5V
10	GND	液晶屏逻辑地	0V
11	D4	数据总线	INOUT 3.3/5V
12	GND	液晶屏逻辑地	0V
13	D5	数据总线	INOUT 3.3/5V
14	GND	液晶屏逻辑地	0V
15	D6	数据总线	INOUT 3.3/5V
16	GND	液晶屏逻辑地	0V
17	D7	数据总线	INOUT 3.3/5V
18	RDJ	读操作信号，低电平有效	IN 3.3/5V
19	WRJ	写操作信号，低电平有效	IN 3.3/5V
20	NC		

具体电路如图 3 所示



# 自制汉字库的原理与应用

硬件设计说明：A15 反相后作为液晶显示器的片选LCD/CS。

## 四 显示原理与软件编制

将“国家”显示在（2，0）处，设置TFT控制板为X方向地址自动加一。因此，在每次设置XY地址后，可以连续写两个字节。详细程序如下所示。

```
#include <reg51.h>
#include <absacc.h>
#define X_ADDR XBYTE[0x8000]
#define Y_ADDR XBYTE[0x8001]
#define CMD XBYTE[0x8002]
#define DAT XBYTE[0x8003]
/*
#define X_ADDR XBYTE[0x0000]
#define Y_ADDR XBYTE[0x0100]
#define CMD XBYTE[0x0200]
#define DAT XBYTE[0x0300]
*/
main()
{
    unsigned int i;
    unsigned char cnt,x,y,z;
    unsigned char code zk_0[32] = {
/*-- 文字: 国 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=16x16 --*/
0x00,0x00,0x7F,0xFC,0x40,0x04,0x5F,0xF4,0x41,0x04,0x41,0x04,0x41,0x04,0x4F,0xE4,
0x41,0x44,0x41,0x24,0x41,0x24,0x5F,0xF4,0x40,0x04,0x40,0x04,0x7F,0xFC,0x40,0x04
};
    unsigned char code zk_1[32] = {
/*-- 文字: 家 --*/
/*-- 宋体12; 此字体下对应的点阵为: 宽x高=16x16 --*/
0x02,0x00,0x01,0x00,0x3F,0xFE,0x40,0x04,0x5F,0xE8,0x01,0x00,0x02,0x00,0x0F,0x20,
0x72,0xB0,0x04,0xC0,0x19,0xC0,0x66,0xB0,0x18,0x8E,0x60,0x84,0x05,0x00,0x02,0x00
};
    //clear panel
    CMD = 0x04;//x inc
    for (y=0;y<240;y++)
    { Y_ADDR = y;X_ADDR = 0;
      for (x=0;x<40;x++)
      {
          DAT = 0;
      }
    }
    //写“国”
    for ( cnt=0; cnt<=15; cnt++ )
```

## 自制汉字库的原理与应用

```
{X_ADDR = 2;  
Y_ADDR = cnt;  
DAT = zk_0[cnt*2];  
DAT = zk_0[cnt*2+1];  
}  
//写“家”  
for ( cnt=0; cnt<=15; cnt++ )  
{X_ADDR = 4;  
Y_ADDR = cnt;  
DAT = zk_1[cnt*2];  
DAT = zk_1[cnt*2+1];  
}  
while(1);}
```