

# 启光电子 ELHI

## 点阵取模篇


在 GUI 函数中 用到了许多汉字的显示方法

例如 16x16 32x32 等 自制的汉字字库 或者是字符库

他们都是由取模软件生成的

### 教程

找到应用软件  应用软件

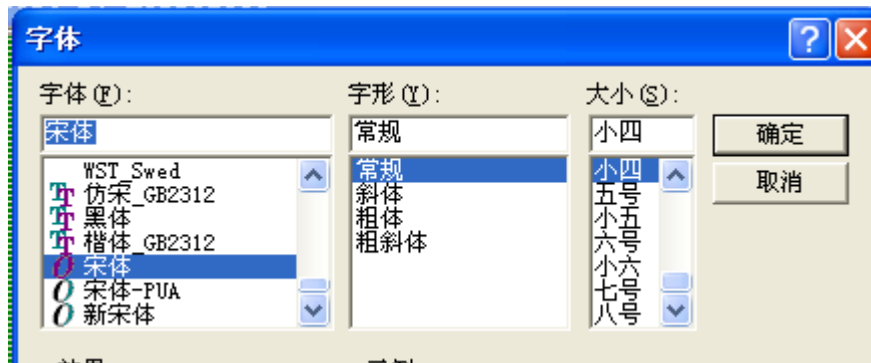
打开取模软件  LCD点阵提取工具zimo221.EXE



这里以取模 16x16 为例 取模字是 我爱单片机

先选择  参数设置

点击  文字输入区字体选择



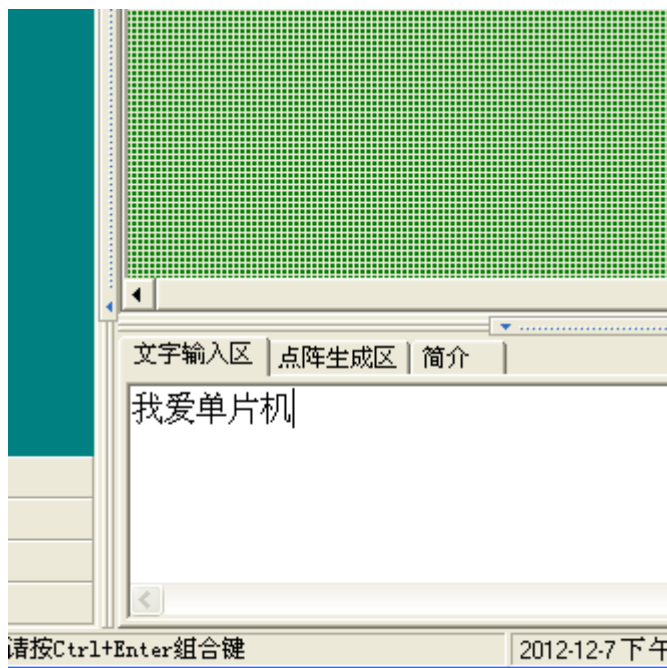
在这里 选择汉字的大小 就可以生成不同比例的点阵形式

不多说 16x16 点阵是 小四 32x32 点阵是 小一 (但是试过 小一生成的是 32x33 的 这样就多了下面的 4 个字节 把最后的 4 个字节去了就可以了)

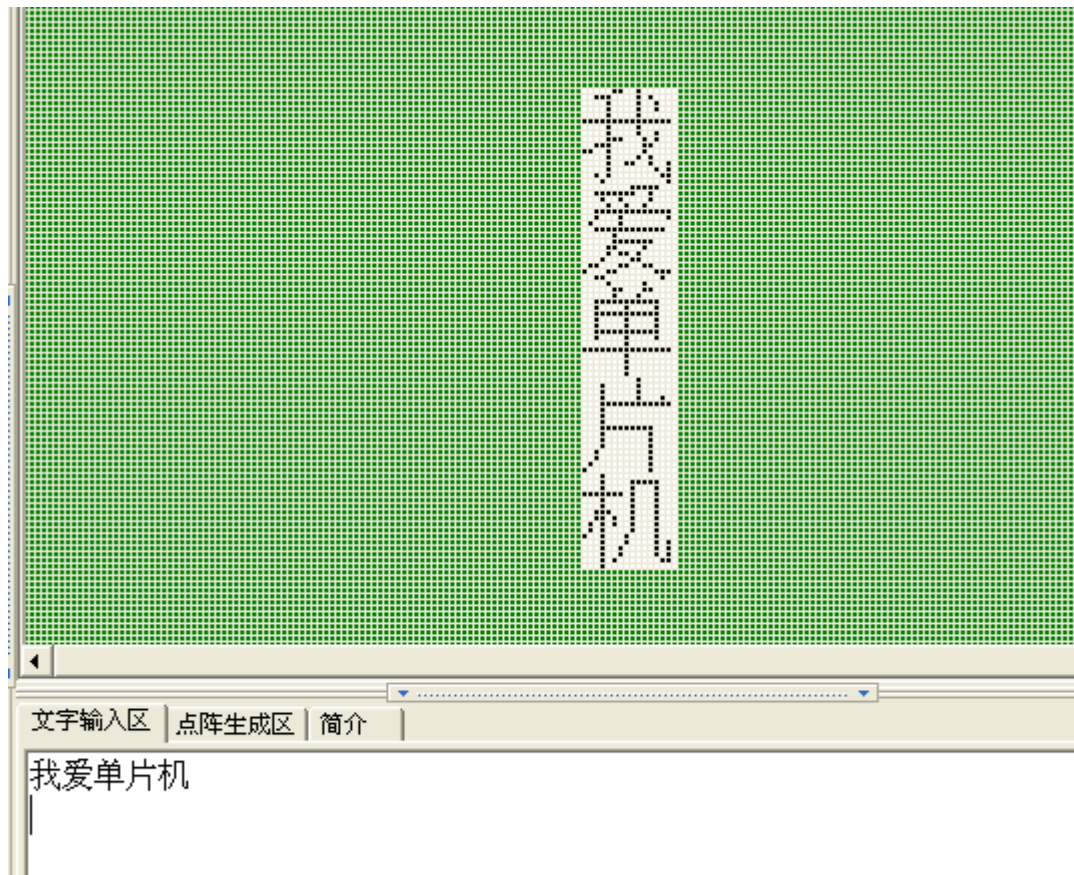
这里在强调一点 就是不同的系统取模的效果也不同, XP 和 W7 就有区别, 所以大家在使用的时候, 想要达到的效果有的时候还是要自己去尝试的, 不能只局限于我们的介绍!



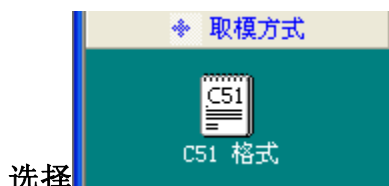
在右下角的框中输入要输入的字 我们这里 输入 我爱单片机



然后看最下面 有句 请按 **Ctrl+Enter** 组合键 按下后就会在上面的格里生成点阵

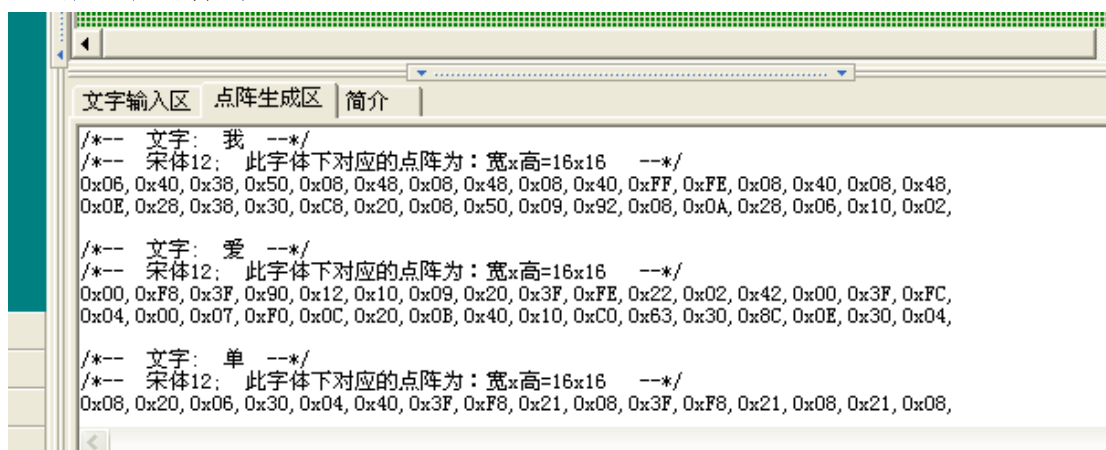


生成后就是导入代码了



选择

点击后 就会看到



然后将生成的点阵字库 放入程序的指定位置 在做简单处理 就可以应用了

其他的汉字大家就要自己找了 这里直接举个例子 为大家学习做参考

## 下面说点程序上的处理

```
code struct  typFNT_HZ16 codeHZ_16[] =          // 数据表
{
/*-- 文字: 云 --*/
/*-- 宋体12: 此字体下对应的点阵为: 宽x高=16x16 --*/
"云", 0x00, 0x00, 0x00, 0x30, 0x3F, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0C, 0xFF, 0xFE, 0x03, 0x00,
0x07, 0x00, 0x06, 0x40, 0x0C, 0x20, 0x18, 0x10, 0x31, 0xF8, 0x7F, 0x0C, 0x20, 0x08, 0x00, 0x00,

/*-- 文字: 销 --*/
/*-- 宋体12: 此字体下对应的点阵为: 宽x高=16x16 --*/
"销", 0x10, 0x20, 0x11, 0x24, 0x3E, 0xA4, 0x20, 0x68, 0x41, 0xFC, 0x7D, 0x04, 0x91, 0x04, 0x11, 0xFC,
0xFD, 0x04, 0x11, 0x04, 0x11, 0xFC, 0x11, 0x04, 0x11, 0x04, 0x15, 0x04, 0x19, 0x14, 0x11, 0x08,

/*-- 文字: 雨 --*/
/*-- 宋体12: 此字体下对应的点阵为: 宽x高=16x16 --*/
"雨", 0x00, 0x00, 0xFF, 0xFE, 0x01, 0x00, 0x01, 0x00, 0x3F, 0xF8, 0x31, 0x88, 0x29, 0x48, 0x25, 0x28,
0x31, 0x88, 0x29, 0x48, 0x25, 0x28, 0x21, 0x08, 0x21, 0x08, 0x21, 0x08, 0x21, 0x28, 0x20, 0x10,
```

这是实验里的 GUI 汉字调用 16x16

这个和我们生成的代码是有些不同的

这里在每个点阵之间 都加上了相应的汉字

例如第一个 “云”， 这个是这个字库的索引 也是结构体

在函数调用里 查 云 这个字的点阵是很难的 所以加了一个索引

直接查 这个字在什么位置 那么它的下个字节 就是它的点阵开始的第一个字节 所以 在导入生成的点阵前 都要记得 加上这个字的 索引