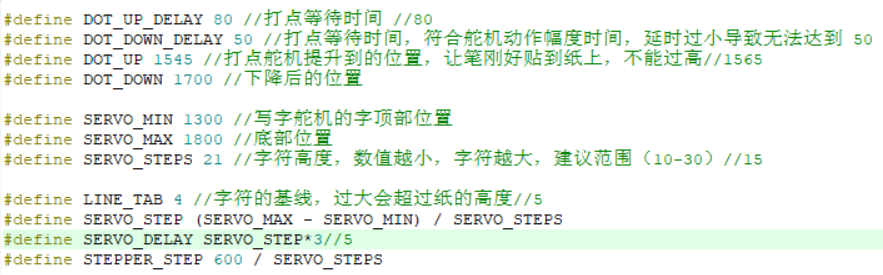
解析打字代码的原理

原版本的打字机是8\*8的，现在修改为16\*16的，所以能打中文及其他更大的字符，理论上可以打一条长条的16\*n山水画之类的，这里只提供想法，没写代码实现。

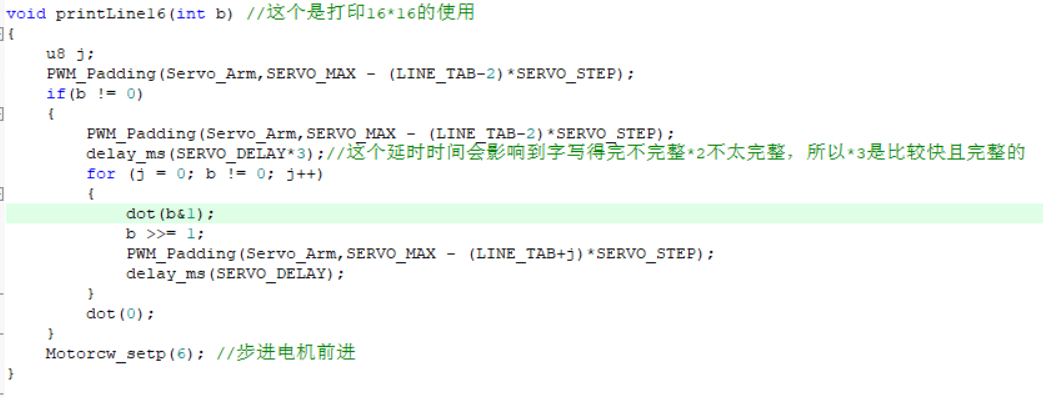
为此我修改了参数部分及增加了两个函数，如下所示：

1. 参数修改



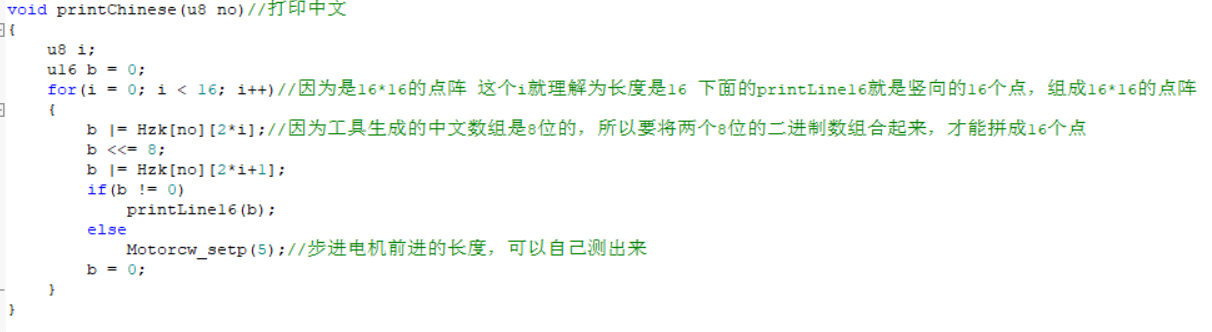
在做这个之前我建议你们先把英文的打出来，这样再接下去会比较简。这是根据我硬件修改的值，你们也根据自己实际的板子调节下，打点时间（带delay的）参照原版本的就好，主要调节SERVO\_MIN、SERVO\_MAX、SERVO\_STEPS、LINE\_TAB的值，SERVO\_MIN要基本碰到纸条的上边缘，SERVO\_MAX要基本能触碰纸条的下边缘，SERVO\_STEPS大概要写多大的字，我的参数应该可以拿去用，LINE\_TAB基线应该要靠下一点才能写出完整的字体，参数设置基本就是这样。

1. 打印长度为16的线的函数



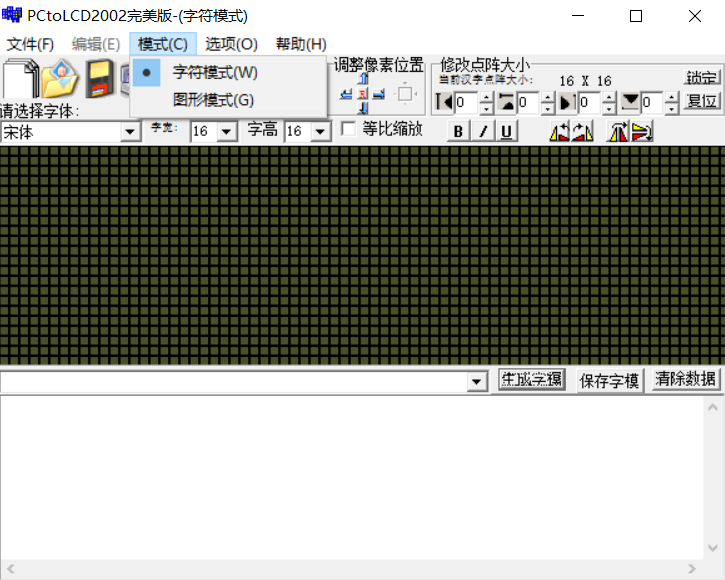
仔细比对是跟8\*8的函数差不多的，几乎一摸一样，就是那个延时的地方不一样，原本是\*20，现在改为\*3，加快它的速度。

1. 打印中文的函数（其他文字只要能建模就可以打出来）

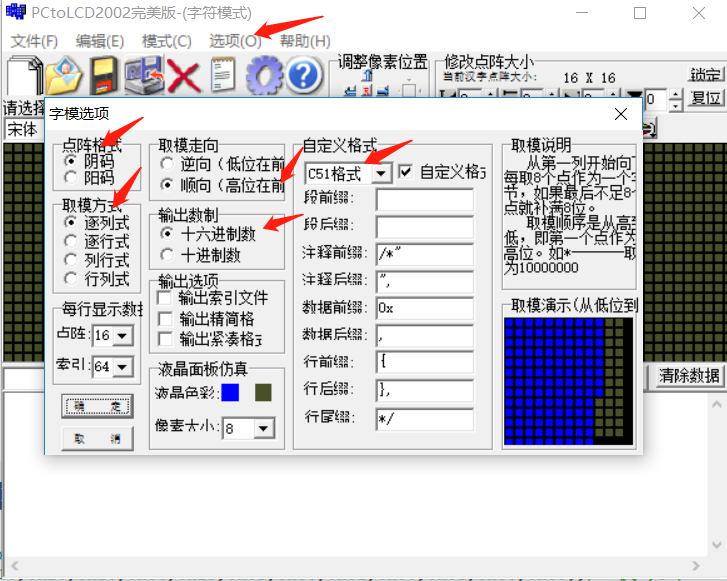


当你移植到这一步，恭喜你基本完成了，现在你只差一个中文字模数组了，上面截图中Hzk[][],就是存放字符字模的二维数组，接下来就是教你如何用软件建模来创建字符数组

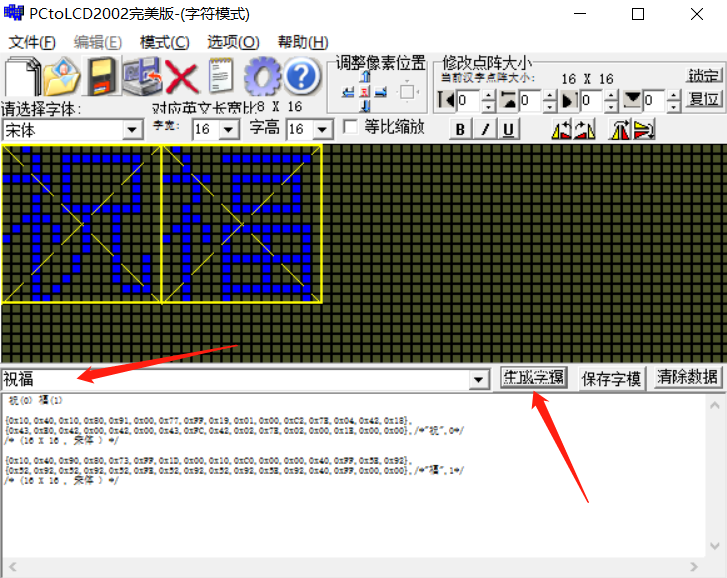
此代码字库文件的拓展方法：

使用软件PCtoLCD2002完美版 对你要打印的东西进行建模，软件配置如下：

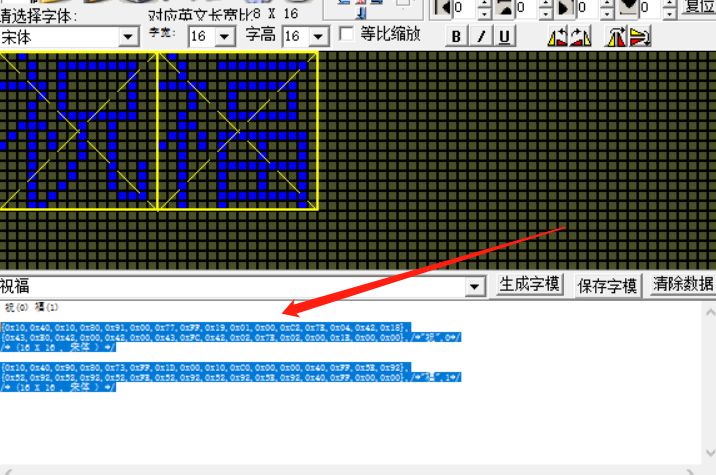
1. 如图选择字符模式



1. 按照这个设置配置好



1. 输入你想要的打印的字符，可以是中文及其他语言，只要他能建模，应该都是可以打出来的



1. 复制这些数据并创建一个二维数组去保存它，这里提供C语言的创建方法：

Static char Hzk[][32] = {

{0x10,0x40,0x10,0x80,0x91,0x00,0x77,0xFF,0x19,0x01,0x00,0xC2,0x7E,0x04,0x42,0x18,

0x43,0xE0,0x42,0x00,0x42,0x00,0x43,0xFC,0x42,0x02,0x7E,0x02,0x00,0x1E,0x00,0x00},

{0x04,0x01,0x04,0x01,0x04,0x02,0x04,0x04,0x04,0x08,0x04,0x30,0x04,0xC0,0xFF,0x00,

0x04,0xC0,0x04,0x30,0x04,0x08,0x04,0x04,0x04,0x02,0x04,0x01,0x04,0x01,0x00,0x00},

{0x08,0x02,0x34,0x92,0x24,0x92,0x24,0x94,0x25,0x24,0x25,0x28,0xA6,0x4A,0x65,0x91,

0x24,0xFE,0x24,0x20,0x24,0x50,0x24,0x88,0x25,0x08,0x2C,0x04,0x30,0x04,0x00,0x00}

}

注意每一个字都由32个16进制数组成，刚刚复制出来的是跟这个不一样的，大家要按照我这样的格式，去除一些多余的大括号，每个大括号和大括号用逗号隔开，最后一个数据不用逗号。

至此，就可以增加到数组内，并调用函数打字

我模仿了arduino的语法写了那两个函数的arduino版本，你们可以适当看看，在文件夹的新建文本文档中

如果你们也想用蓝牙之类的方式让他写中文，那就需要字库，要用一个内存芯片去储存所有字体的字模数组，我已经把软件（字库制作软件）放这了，可以生成字库放进储存芯片，要注意字体的编码格式，具体的实现方法自行寻找资料

**作者：obito （https: https://github.com/obito0）**