

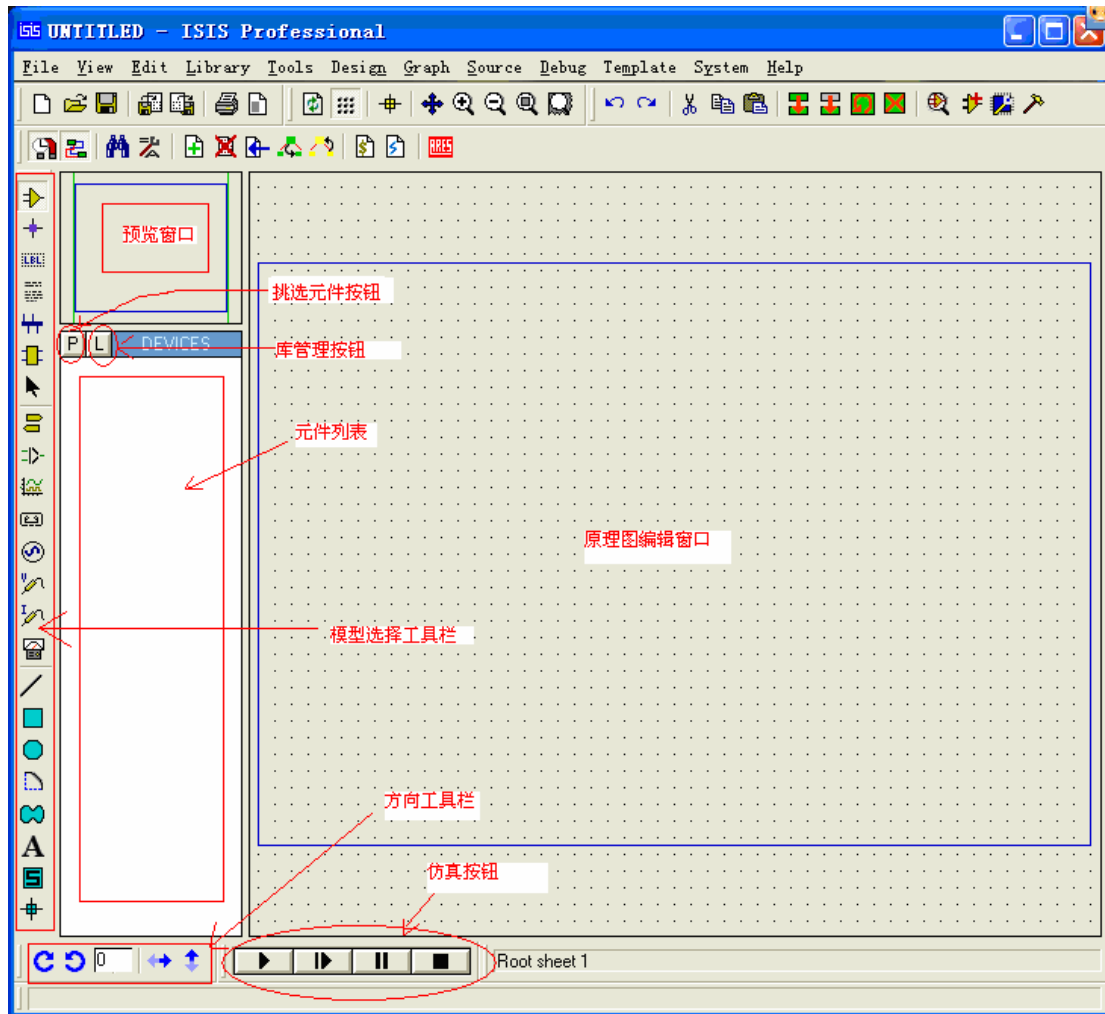
Proteus 入门教程

作者：archeng504 日期：2006-1-2

本文将简单介绍一下 Proteus 的使用。在这里，我用的 Proteus 版本是 Proteus 6.7 sp3 Professional。

一、Proteus 6 Professional 界面简介

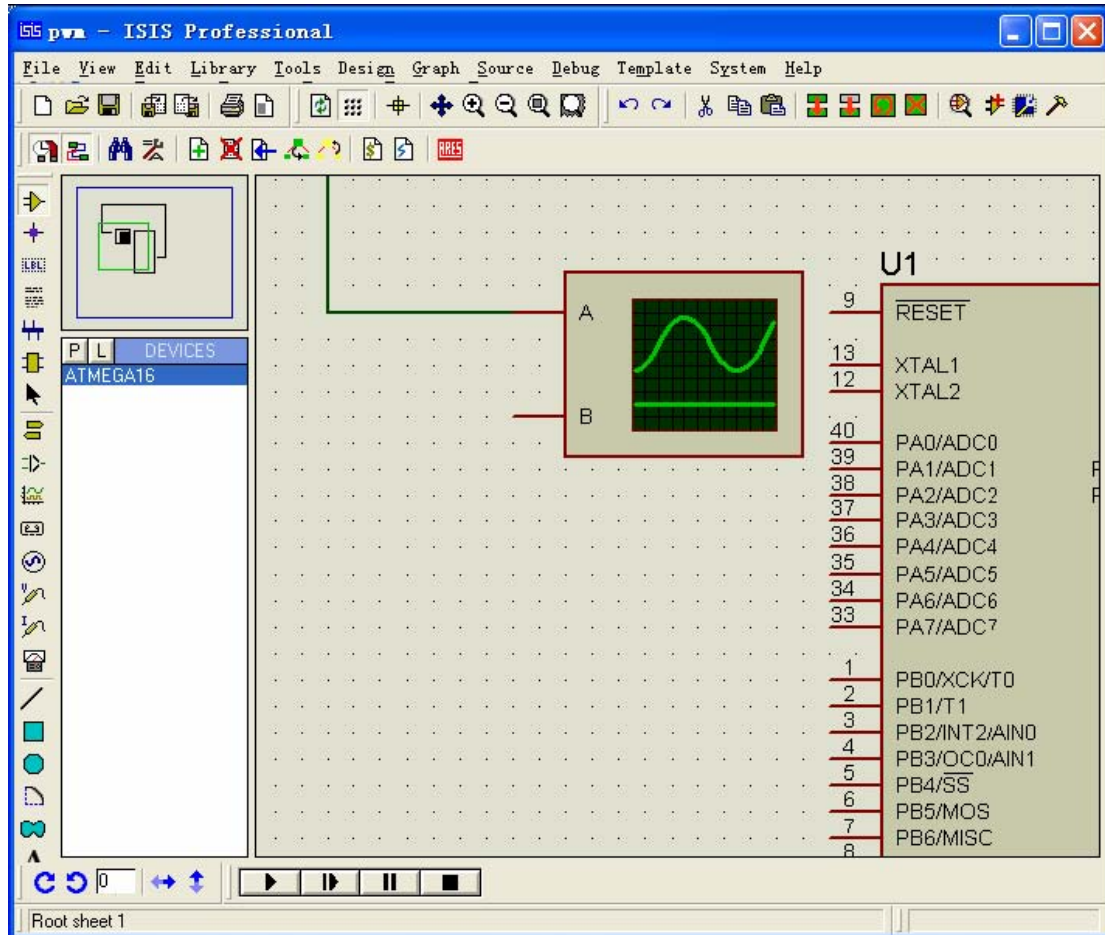
安装完 Proteus 后，运行 ISIS 6 Professional，会出现以下窗口界面：




为了方便介绍，我分别对窗口内各部分进行中文说明（见上图）。下面简单介绍各部分的功能：

- 原理图编辑窗口 (The Editing Window):** 顾名思义，它是用来绘制原理图的。蓝色方框内为可编辑区，元件要放到它里面。注意，这个窗口是没有滚动条的，你可用预览窗口来改变原理图的可视范围。
- 预览窗口 (The Overview Window):** 它可显示两个内容，一个是：当你在元件列表中选择一个元件时，它会显示该元件的预览图；另一个是，当你的鼠标焦点落在原理图编辑窗口时（即放置元件到原理图编辑窗口

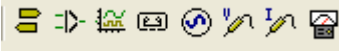
后或在原理图编辑窗口中点击鼠标后)，它会显示整张原理图的缩略图，并会显示一个绿色的方框，绿色的方框里面的内容就是当前原理图窗口中显示的内容，因此，你可用鼠标在它上面点击来改变绿色的方框的位置，从而改变原理图的可视范围。



3. 模型选择工具栏 (Mode Selector Toolbar):

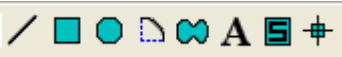
主要模型 (Main Modes): 

- 1* 选择元件 (components) (默认选择的)
- 2* 放置连接点
- 3* 放置标签 (用总线时会用到)
- 4* 放置文本
- 5* 用于绘制总线
- 6* 用于放置子电路
- 7* 用于即时编辑元件参数 (先单击该图标再单击要修改的元件)

配件 (Gadgets): 

- 1* 终端接口 (terminals): 有 VCC、地、输出、输入等接口
- 2* 器件引脚: 用于绘制各种引脚

- 3* 仿真图表 (graph): 用于各种分析, 如 Noise Analysis
- 4* 录音机
- 5* 信号发生器 (generators)
- 6* 电压探针: 使用仿真图表时要用到
- 7* 电流探针: 使用仿真图表时要用到
- 8* 虚拟仪表: 有示波器等


2D 图形 (2D Graphics): 


- 1* 画各种直线
- 2* 画各种方框
- 3* 画各种圆
- 4* 画各种圆弧
- 5* 画各种多边形
- 6* 画各种文本
- 7* 画符号
- 8* 画原点等

4. 元件列表 (The Object Selector):

用于挑选元件 (components)、终端接口 (terminals)、信号发生器 (generators)、仿真图表 (graph) 等。举例, 当你选择“元件 (components)”, 单击“P”按钮会打开挑选元件对话框, 选择了一个元件后 (单击了“OK”后), 该元件会在元件列表中显示, 以后要用到该元件时, 只需在元件列表中选择即可。

5. 方向工具栏 (Orientation Toolbar):

旋转:  旋转角度只能是 90 的整数倍。

翻转:  完成水平翻转和垂直翻转。

使用方法: 先右键单击元件, 再点击 (左击) 相应的旋转图标。

6. 仿真工具栏

 仿真控制按钮

- 1* 运行
- 2* 单步运行
- 3* 暂停
- 4* 停止

二、操作简介

1、绘制原理图: 绘制原理图要在原理图编辑窗口中的蓝色方框内完成。原理图编辑窗口的操作是不同于常用的 WINDOWS 应用程序的, 正确的操作是: 用左键放置元件; 右键选择元件; 双击右键删除元件; 右键拖选多个元件; 先右键后

左键编辑元件属性；先右键后左键拖动元件；连线用左键，删除用右键；改连接线：先右击连线，再左键拖动；中键放缩原理图。具体操作见下面例子。

2、定制自己的元件：有三个实现途径，一是用 PROTEUS VSM SDK 开发仿真模型，并制作元件；另一个是在已有的元件基础上进行改造，比如把元件改为 bus 接口的；还有一个是利用已制作好（别人的）的元件，我们可以到网上下载一些新元件并把它们添加到自己的元件库 里面。由于我没有 PROTEUS VSM SDK，所以我只介绍后两个。

3、Sub-Circuits 应用：用一个子电路可以把部分电路封装起来，这样可以节省原理图窗口的空间。

三、实例

（一）、AVR 单片机的仿真实例

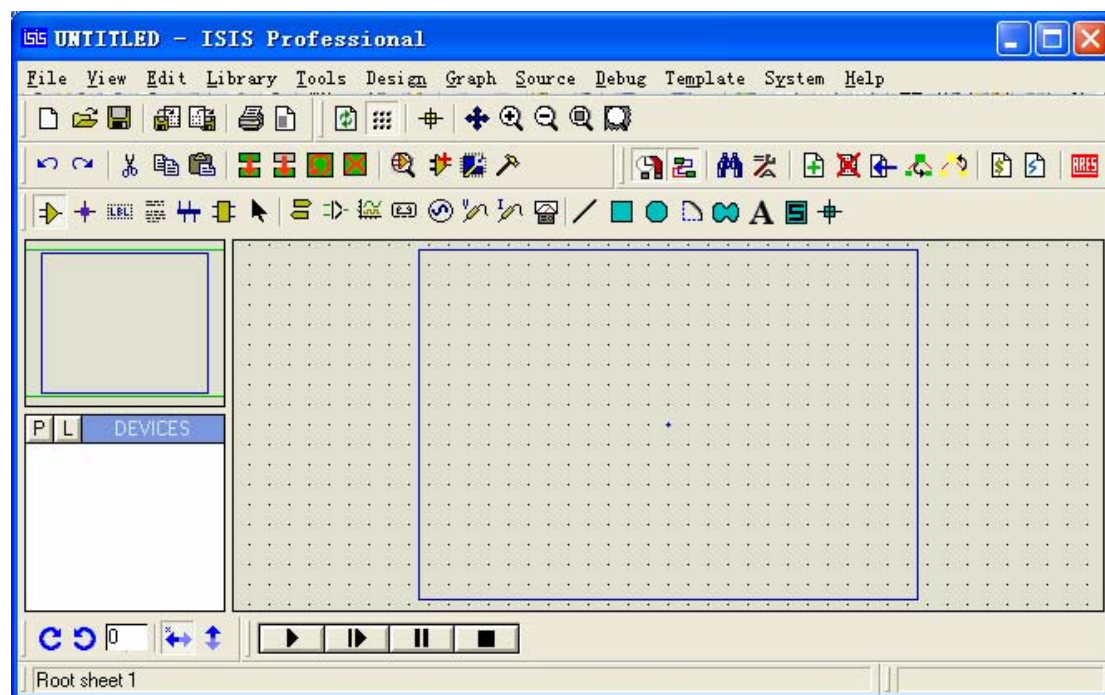
本例是实现 AVR 驱动 LCD1602，并用示波器监测数据线。

开始前先要准备好仿真文件，就是用编译器编译连接产生的调试或下载文件，不同编译器产生的文件格式是不同的，如 ICC 是 COF、IAR 是 D90，GCC 是 COF、ELF。Proteus 6.7sp3 支持的有 COF、D90、HEX 等，ELF 暂不支持。本例用的是：lcd_C.hex。

开始了!!!!!!

运行 Proteus 6 Professional (ISIS6 Professional)

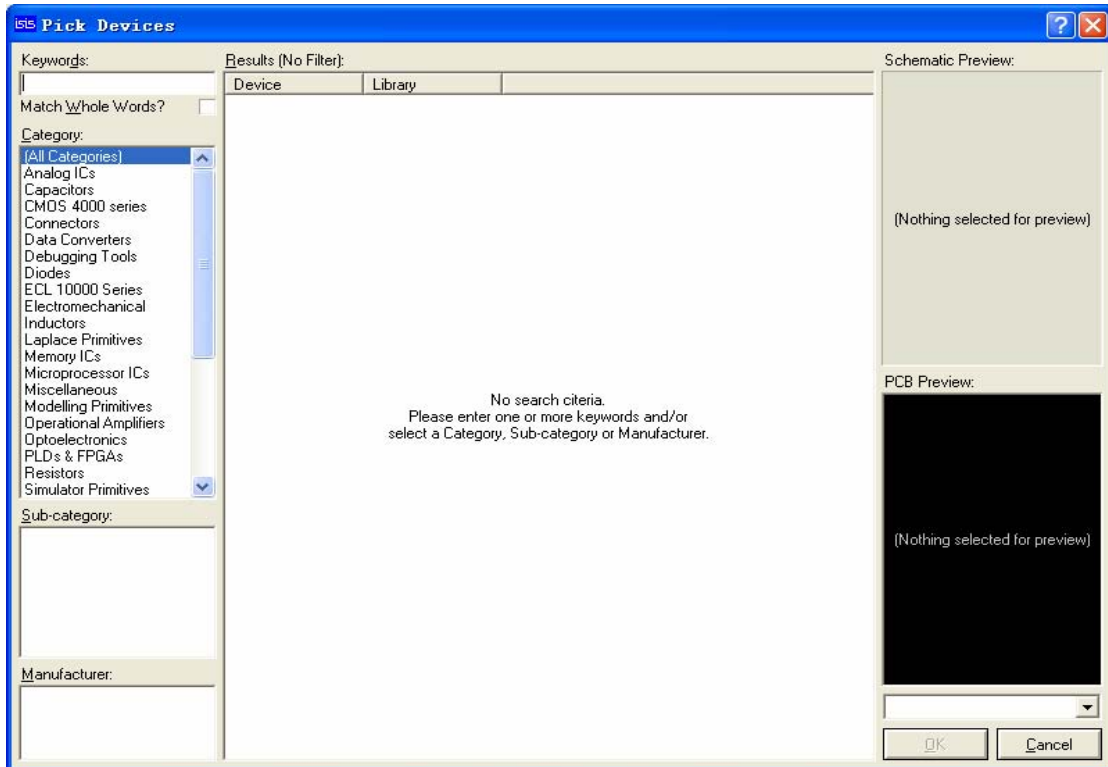
出现下面窗口：



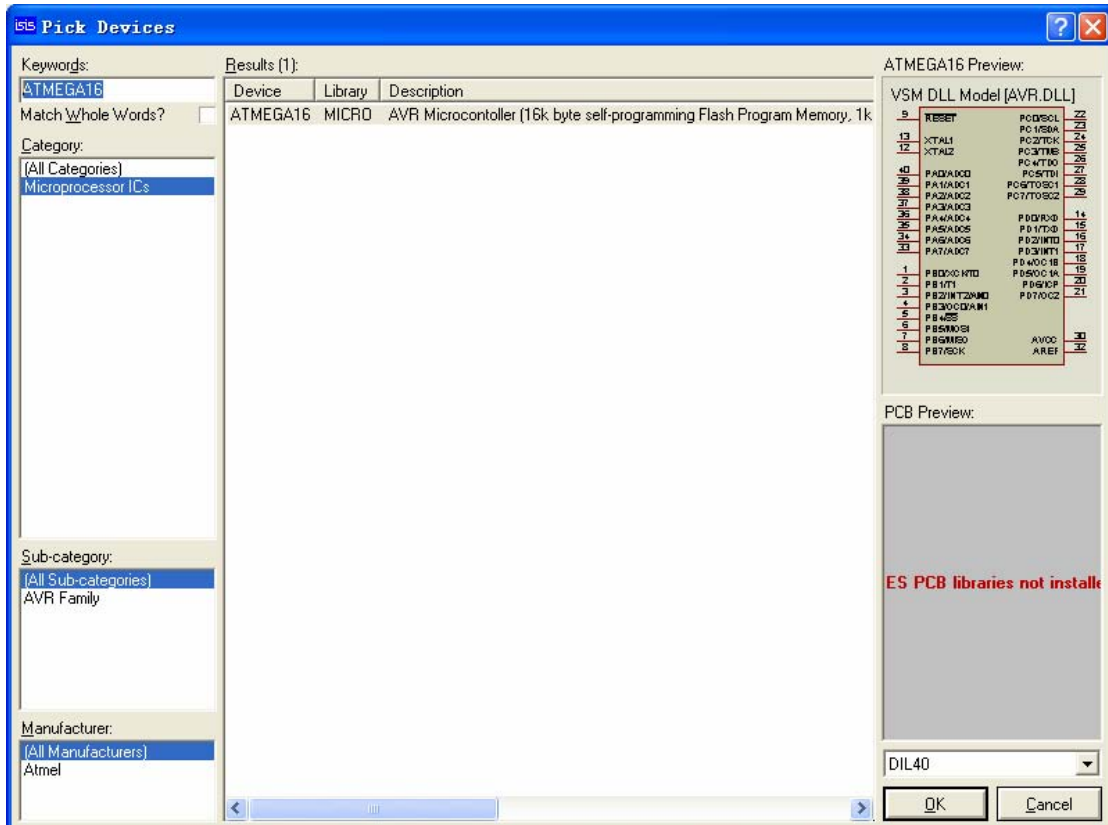
1、添加元件到元件列表中：本例要用到的元件有：ATMEGA16、LM016L (LCD1602)、“地”、示波器。

单击“P”按钮，出现挑选元件对话框

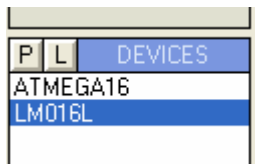




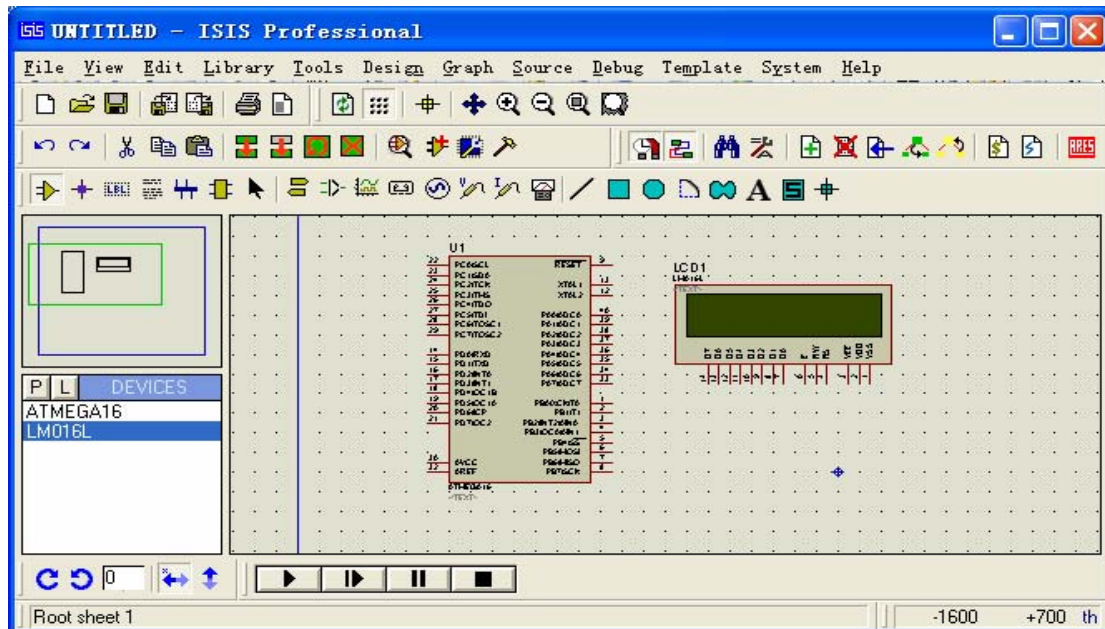
在对话框的 KEYWORDS 中输入 ATMEGA16，得到以下结果：




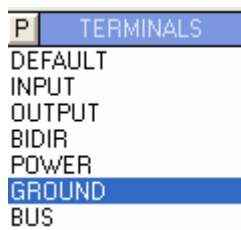
单击 OK，关闭对话框，这时元件列表中列出 ATMEGA16，同样找出 LM016L。最终结果：




2、放置元件：在元件列表中左键选取 ATMEGA16，在原理图编辑窗口中单击左键，这样 ATMEGA16 就被放到原理图编辑窗口中了。同样放置 LM016L。

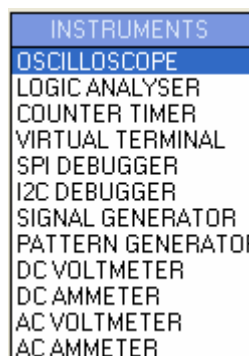


添加“地”：左键选择模型选择工具栏中的  图标，出现：



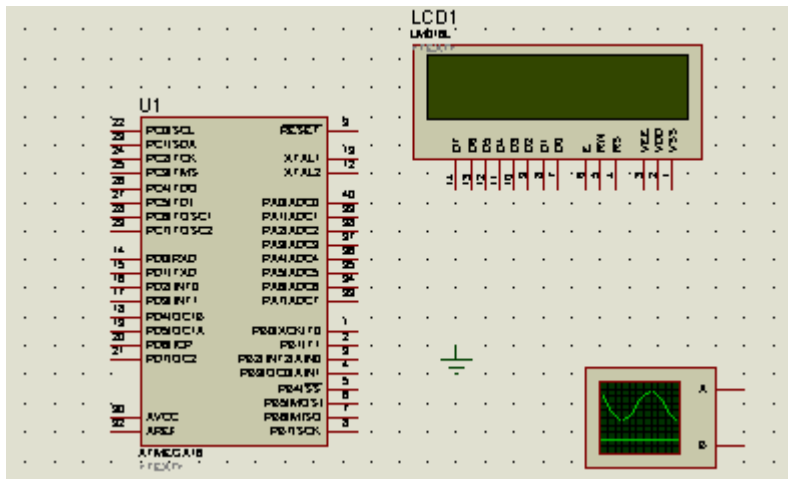
左键选择 GROUND，并在原理图编辑窗口中左击，这样“地”就被放置到原理图编辑窗口中了。






添加示波器：左键选择模型选择工具栏中的  图标，出现：



左键选择 OSCILLOSCOPE，并在原理图编辑窗口中左击，这样示波器就被放置

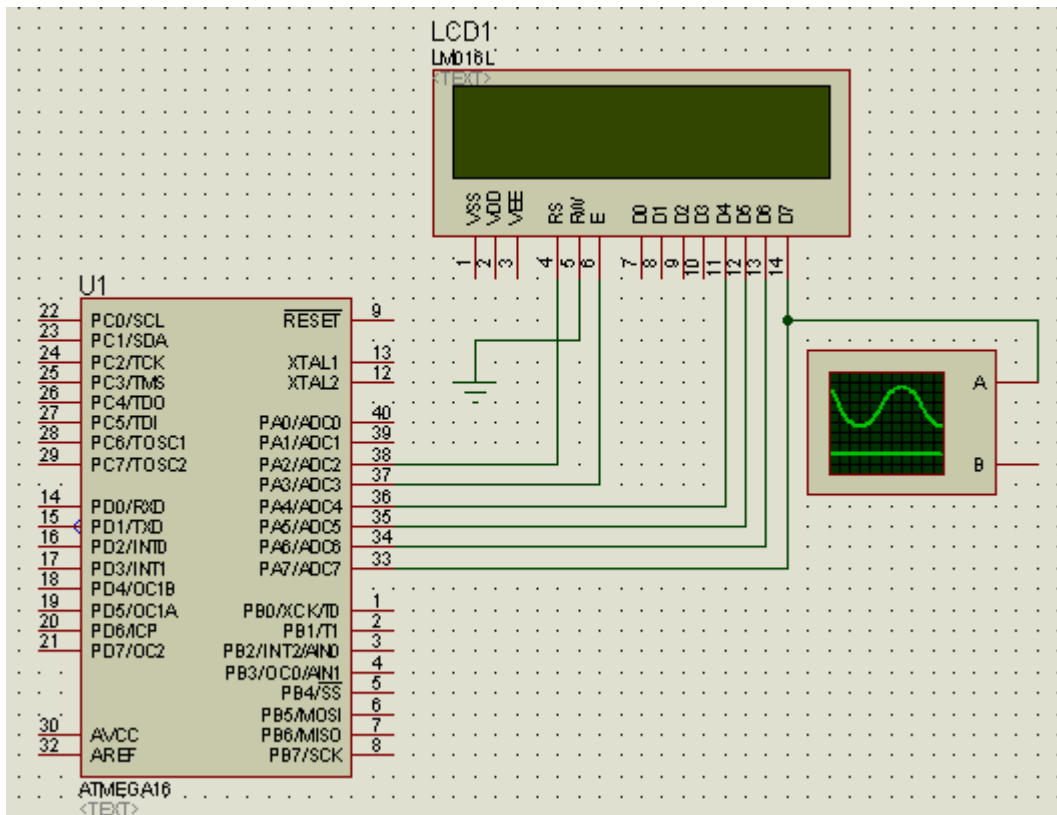
到原理图编辑窗口中了。



补充: 放置元件时要注意所放置的元件应放到蓝色方框内, 如果不小心放到外面, 由于在外面鼠标用不了, 要用到菜单“Edit”的“Tidy”清除, 方法很简单只需单击“Tidy”即可。操作中可能要整体移动部分电路, 操作方法: 先用右键拖选, 再单击     中的 , 这时这部分电路会随鼠标移动, 在目标位置单击左键, 这部分电路将被放到该处。

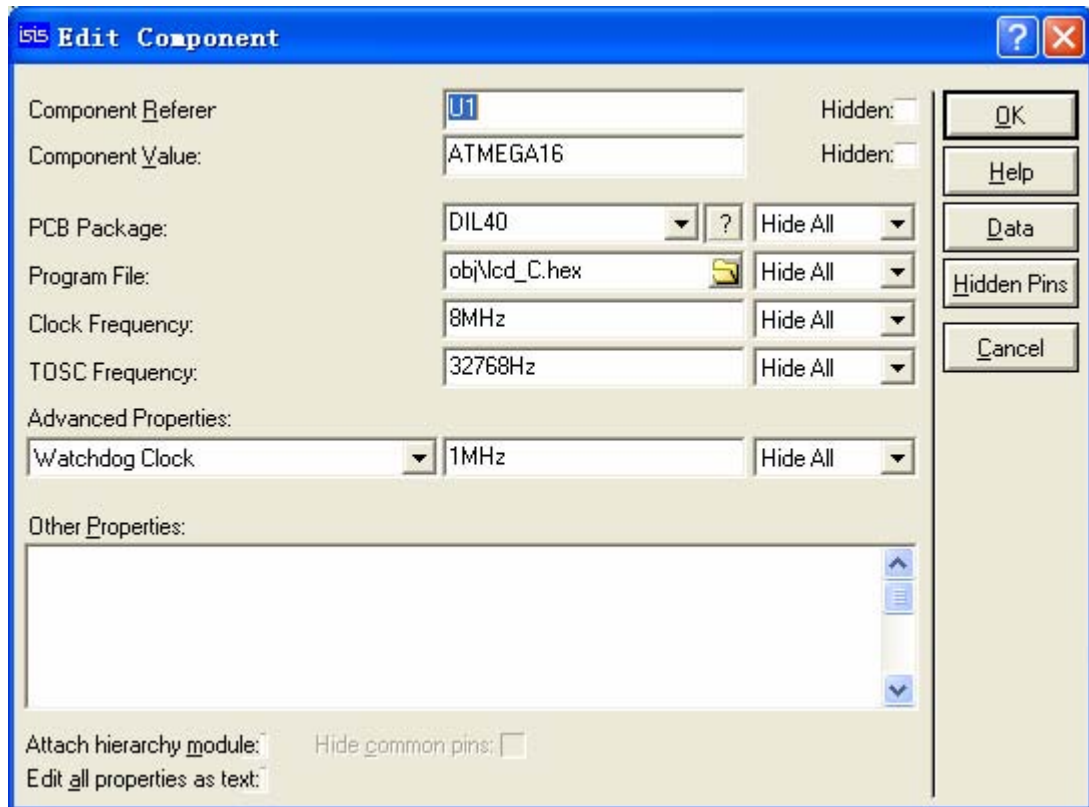
3. 连线


AVR、LCD 的 VSS、VDD、VEE 不需连接, 默认 VSS=0V、VDD=5V、VEE= -5V、GND=0V



4. 添加仿真文件

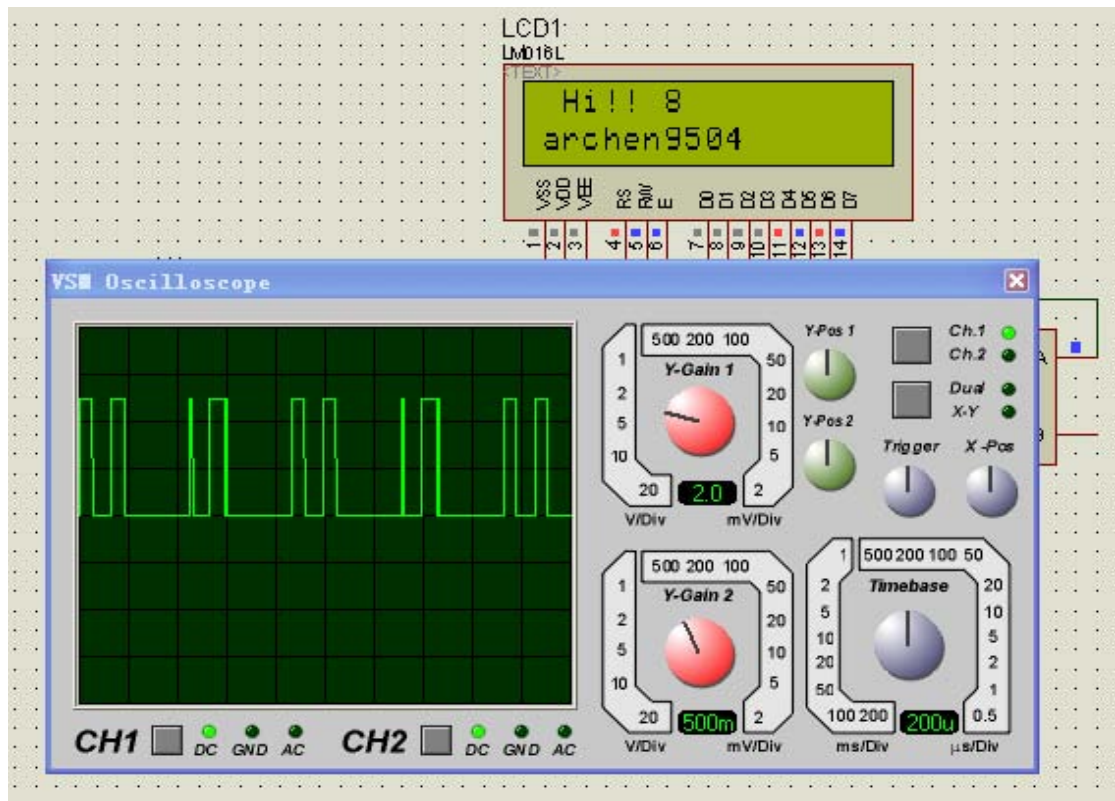
先右键 ATMEGA16 再左键, 出现



在 Program File 中单击  出现文件浏览对话框，找到 lcd_C.hex 文件，单击确定完成添加文件，在 Clock Frequency 中把频率改为 8MHz，单击 OK 退出。

5. 仿真

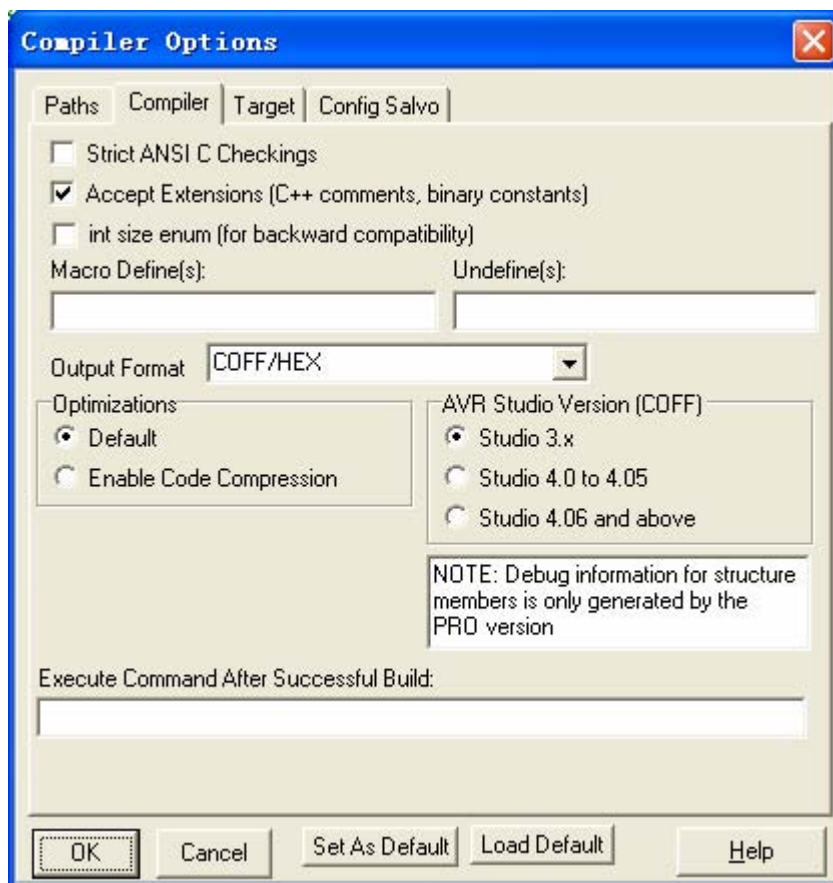
单击  开始仿真。




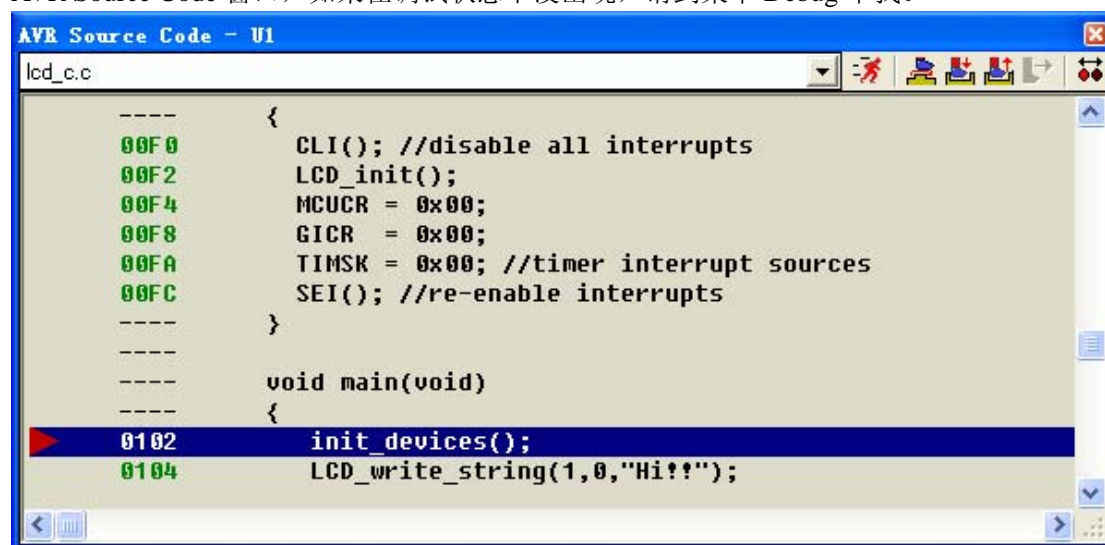
说明：红色代表高电平，蓝色代表低电平，灰色代表不确定电平（floating）。运行时，在 **Debug** 菜单中可以查看 AVR 的相关资源。

6、源代码调试


先声明一下，我是用 ICC 的，版本 ICC6.31a Pro。Proteus 是支持 COF 文件调试的。在用 ICC 时，你应该改一下编译器选项，AVR Studio Version 选 Studio 3.x，这样在 Proteus 中可看到原代码了。



在 Proteus 中，完成绘制原理图和添加调试文件后（COF 文件），单击  会出现 AVR Source Code 窗口，如果在调试状态下没出现，请到菜单 Debug 中找。

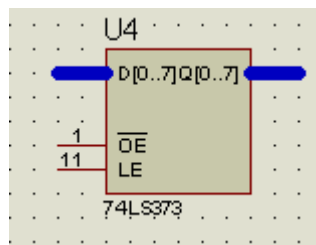
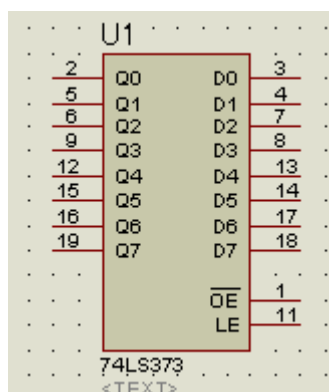


说一下      ，下面依次说明

- 1* 连续运行，会退出单步调试状态，并关闭 AVR Source Code 窗口
 - 2* 单步运行，遇到子函数会直接跳过
 - 3* 单步运行，遇到子函数会进入其内部
 - 4* 跳出当前函数，当用 3*进入到函数内部，使用它会立即退出该函数返回上一级函数，可见它应该与 3*配合使用
 - 5* 运行到鼠标所在行
 - 6* 添加或删除断点，设置了断点后用  程序会停在断点处。
- 另外，你可 AVR Source Code 窗口中右击，会出现右键菜单，上面的功能在这里不作说明。

(二)、把 74LS373 改成 bus 接口

我们可以修改 Proteus 中的元件，比如把 74LS373 改成 bus 接口的，有两种实现方案。

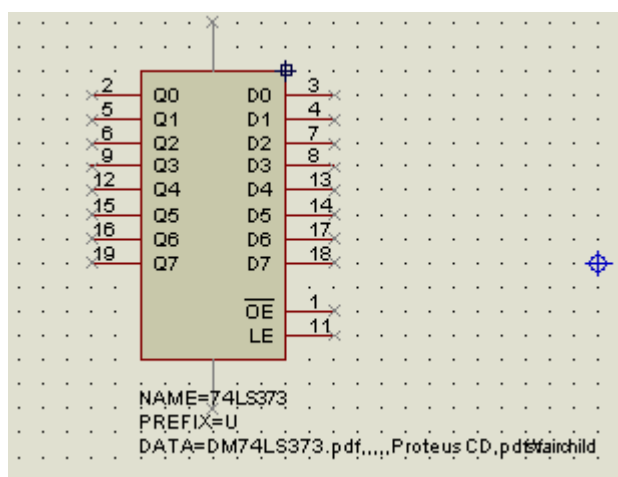


方案 (1) 利用现有元件

步骤:

- 1、“拆”元件

先把添加 74LS373 (左上图的)，再右键选择 74LS373，再单击工具栏的  出现

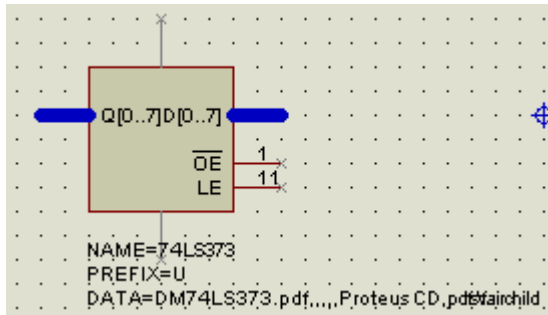


- 2、修改

先把 Q0 至 Q7 、D0 至 D7 的管脚删掉

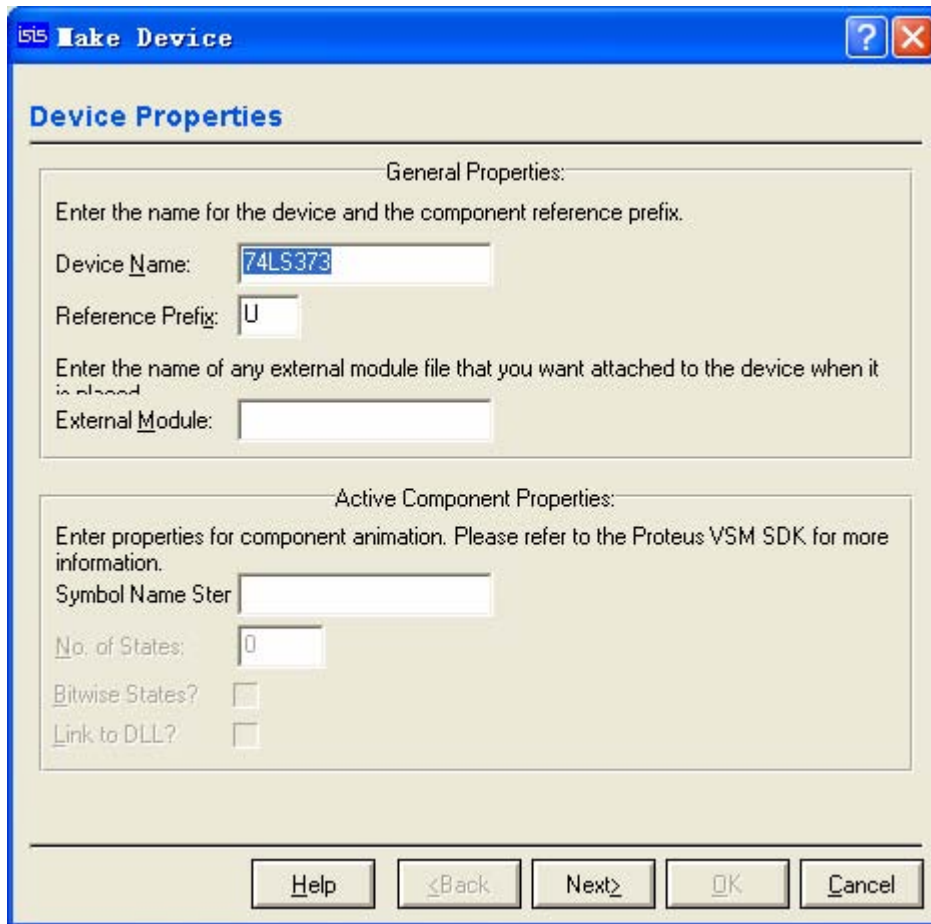
添加 BUS 的具体方法见方案 (2) 的相关介绍。

效果如下:

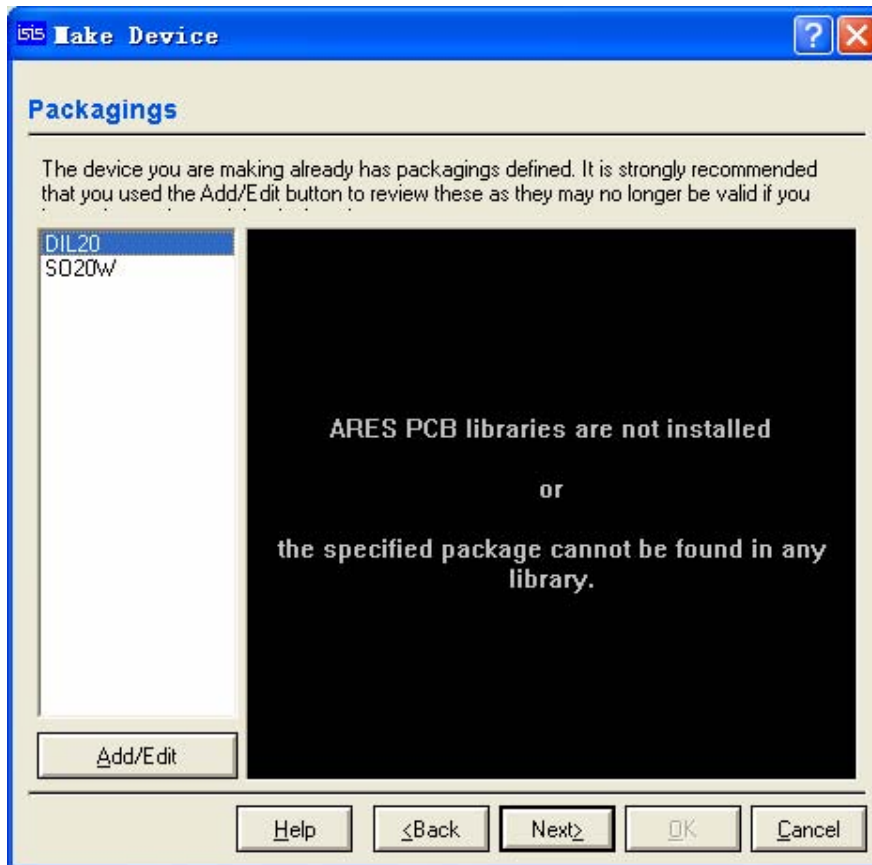


3、重新 Make Device

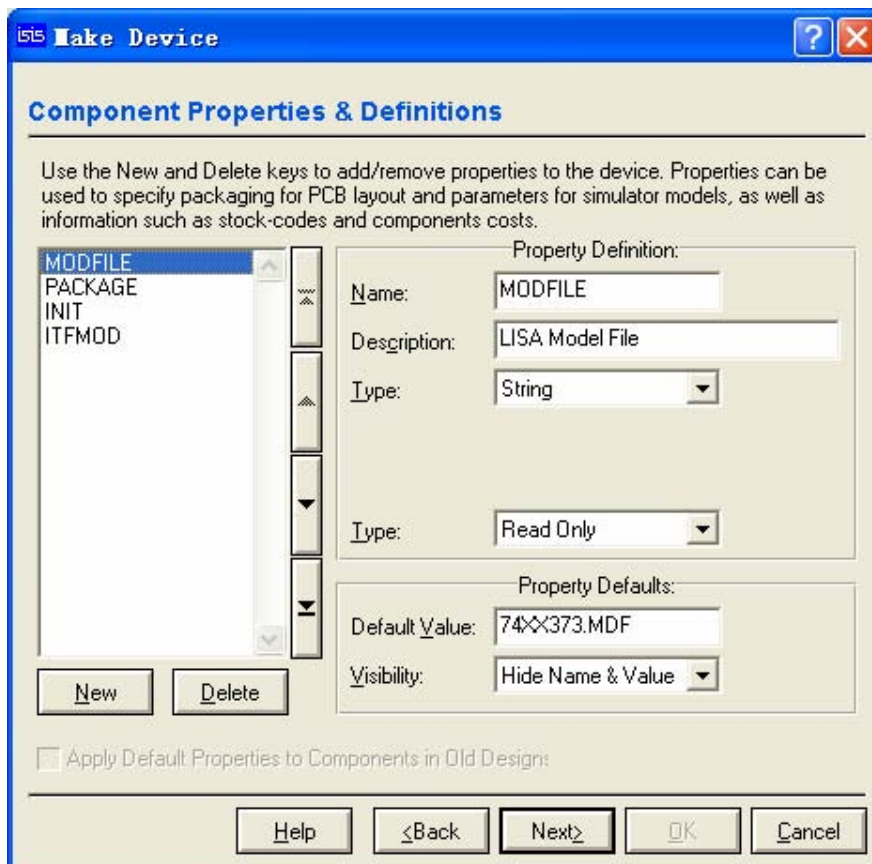
用右键拖选整个元件，选择菜单“Library”→“Make Device”，出现下面对话框



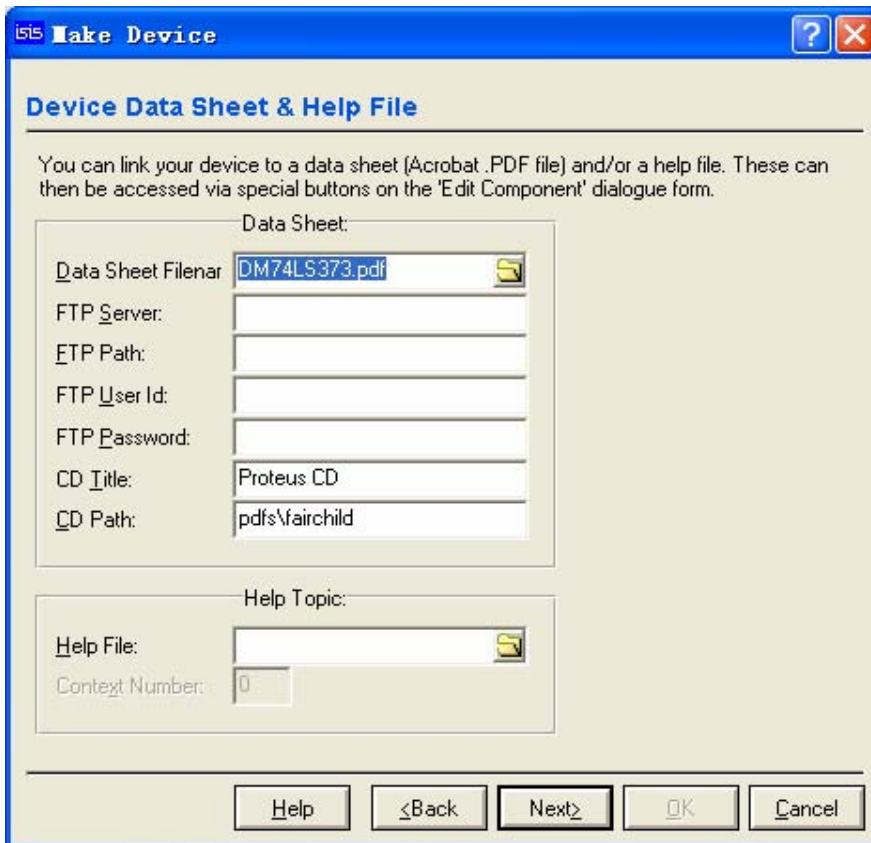
将 74LS373 改为 74LS373.bus 其他不变，Next



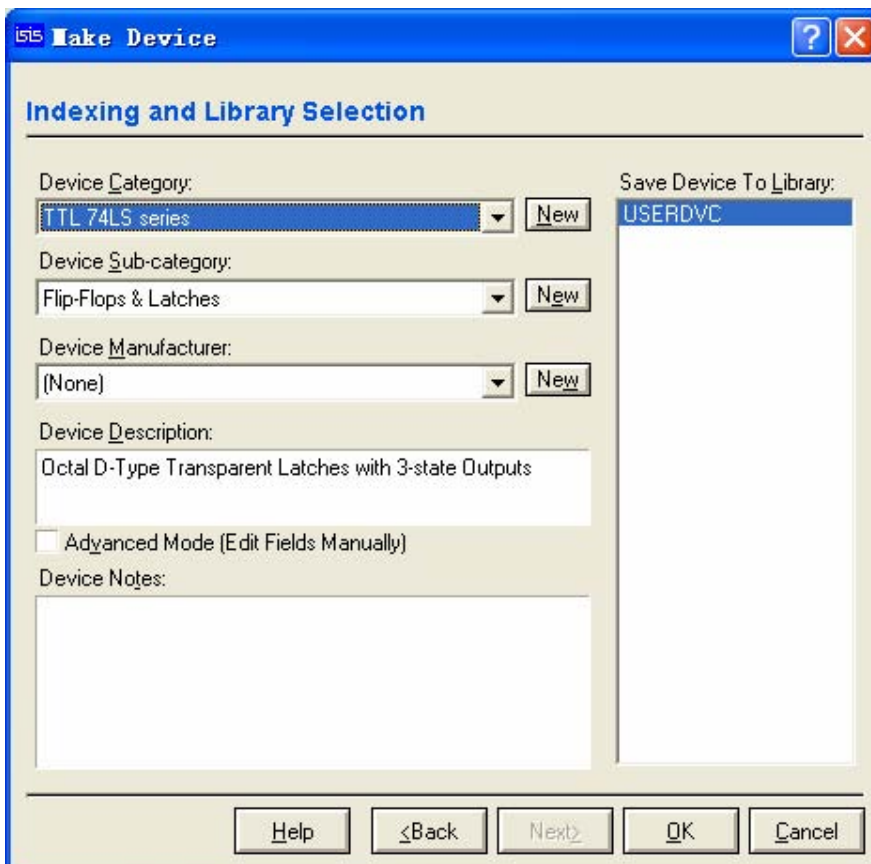
这个不用改，Next



这个不用改，Next




这个不用改，Next

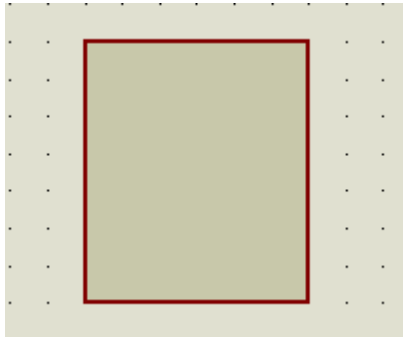


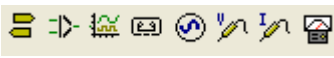

这个最好要改，第一个可改为 74LS BUS，第二个不变，OK 结束

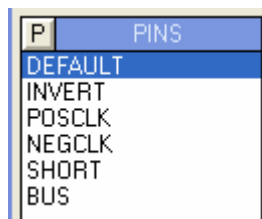
方案（2）重新绘制元件

步骤：

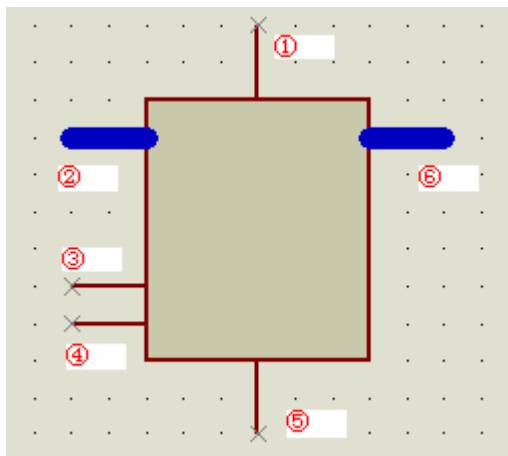
1. 用 2D GRAPHICS 中的  中的  绘制 Device Body



2. 用  中的  绘制引脚



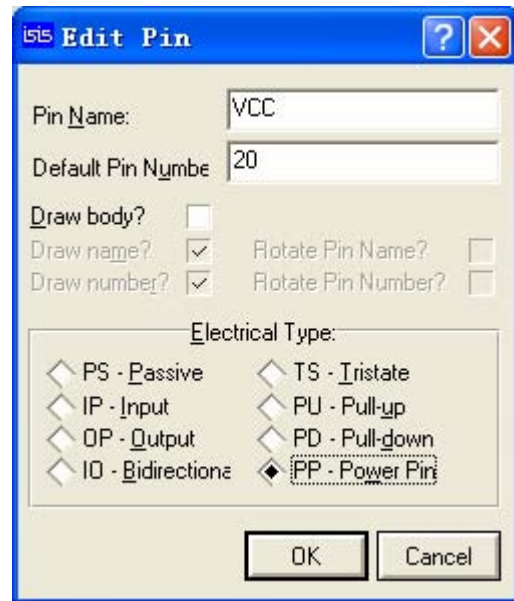
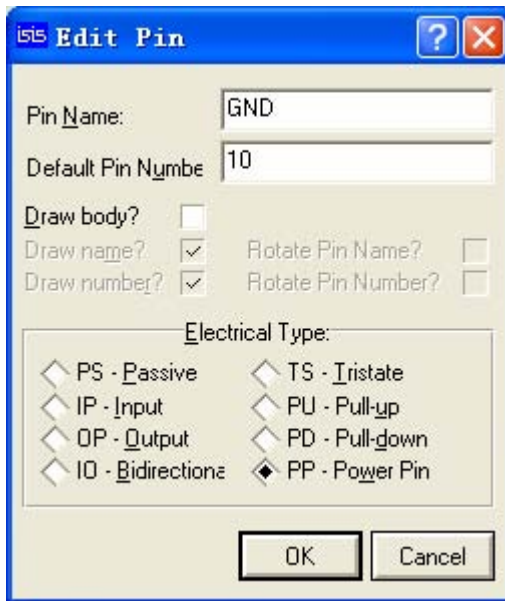
DEFAULT 为普通引脚 , BUS 为总线 



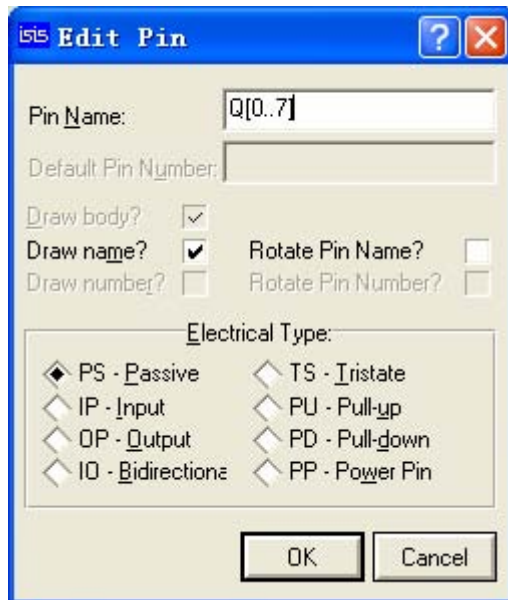
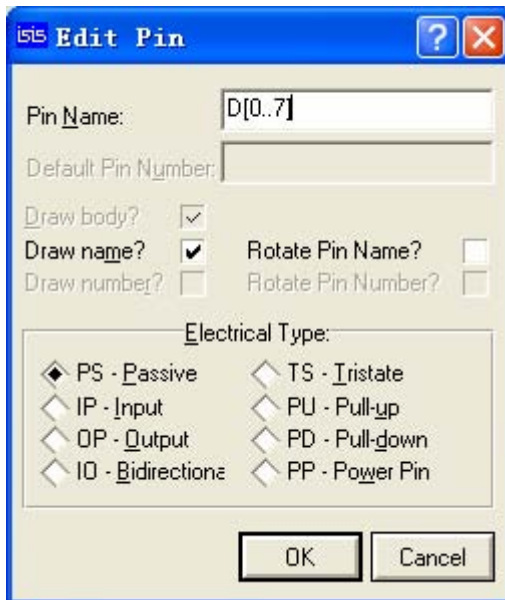
3. 修改引脚属性

说明：①为 GND, PIN10; ②为 D[0..7]; ③为 OE, PIN1; ④为 LE, PIN11; ⑤为 VCC, PIN20; ⑥为 Q[0..7]。

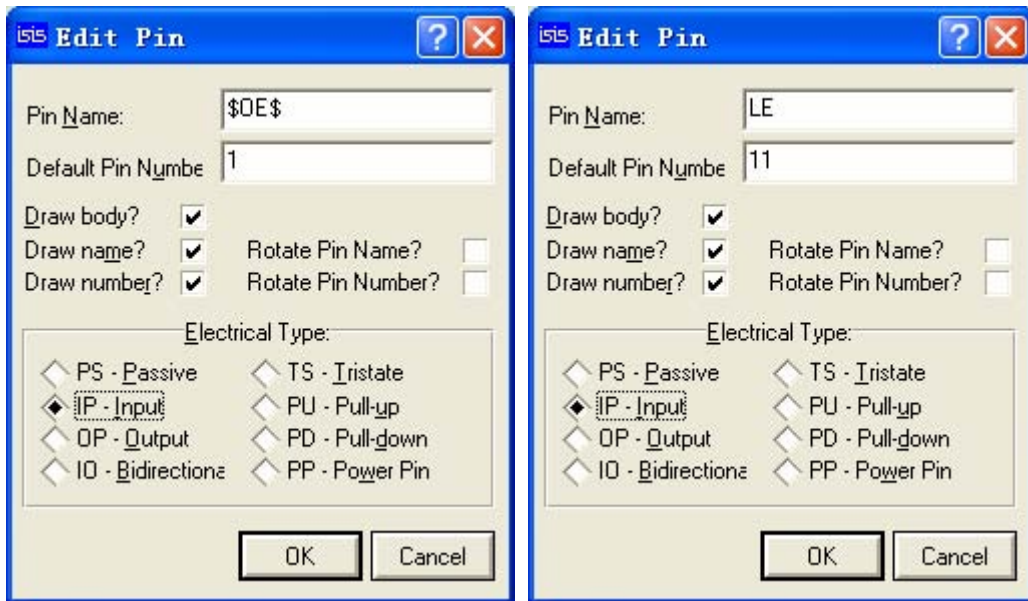
先右击后左击①，在出现的对话框中输入下面数据；⑤是类似的。GND、VCC 需要隐藏，故 Draw body 不选。



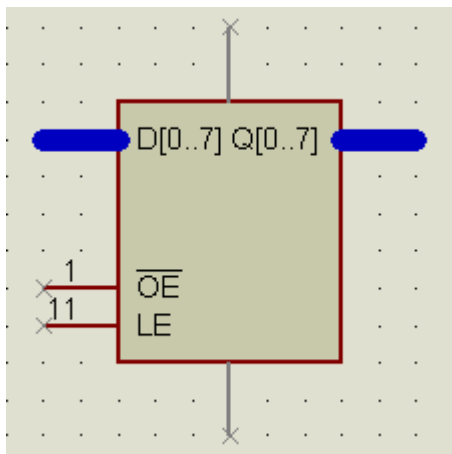
先右击后左击②，在出现的对话框中输入下面数据；⑥是类似的



先右击后左击③，在出现的对话框中输入下面数据；④是类似的

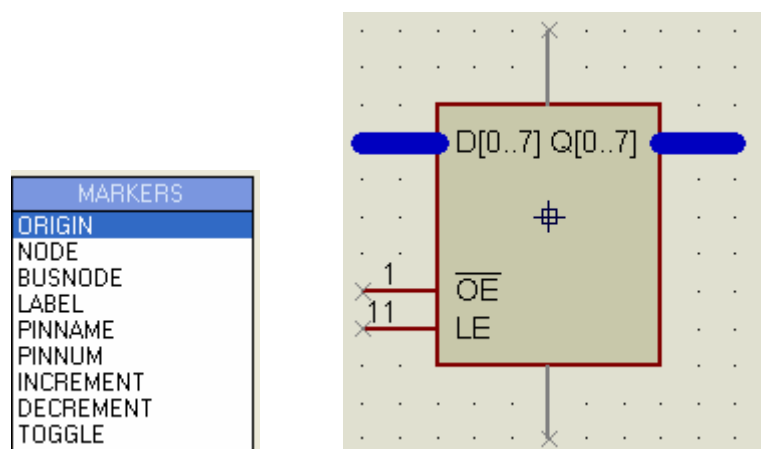


最终效果



4. 添加中心点

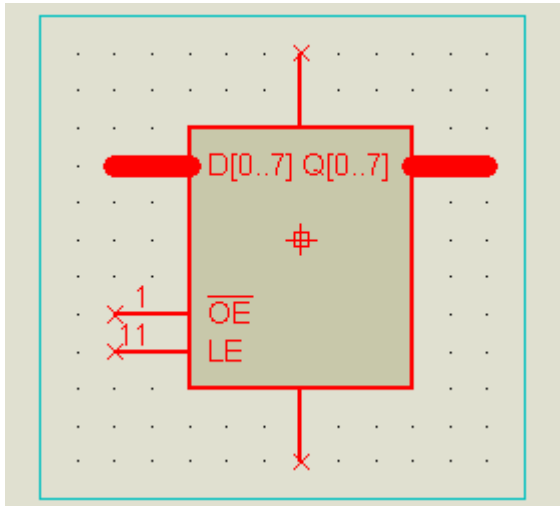
选择 中的 绘制中心点，选择 ORIGIN



中心点的位置可任意放。

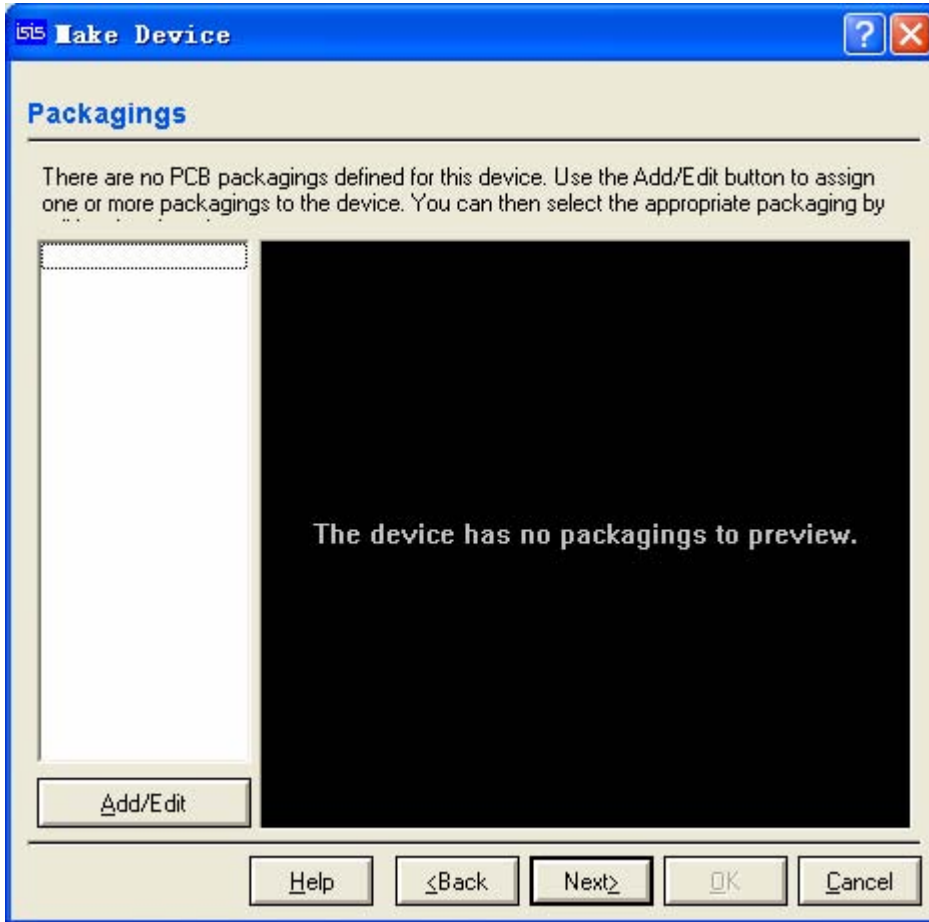
5. 封装入库

先用右键选择整个元件

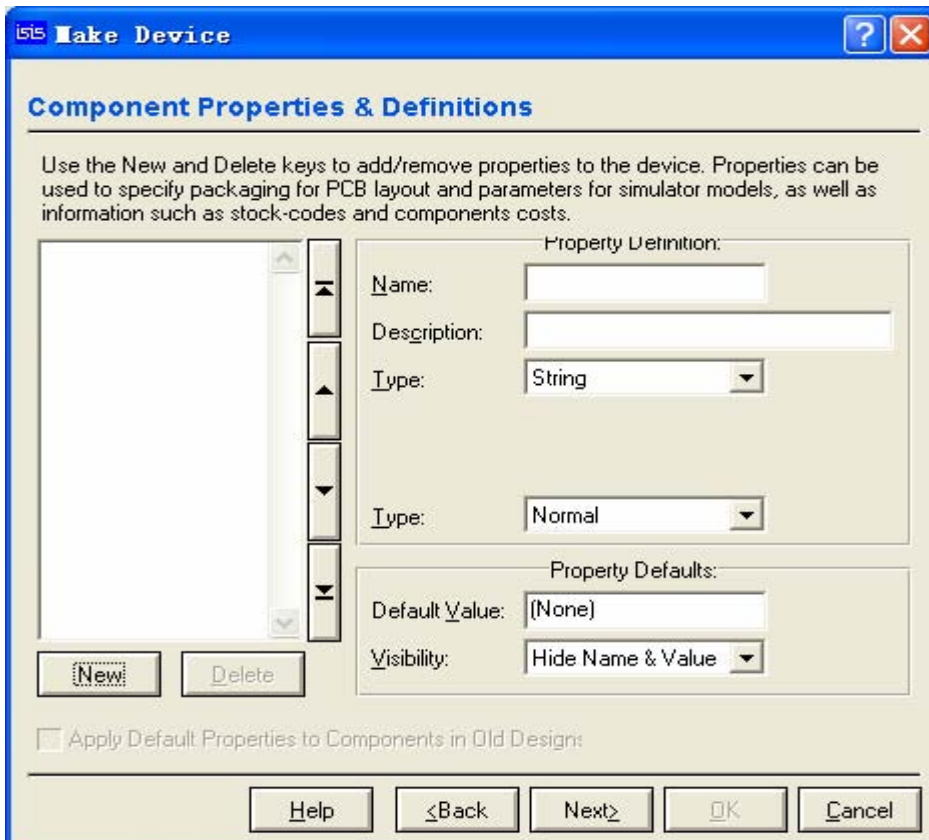


然后，选择菜单“Library” → “Make Device”，出现下面对话框，并输入下面内容

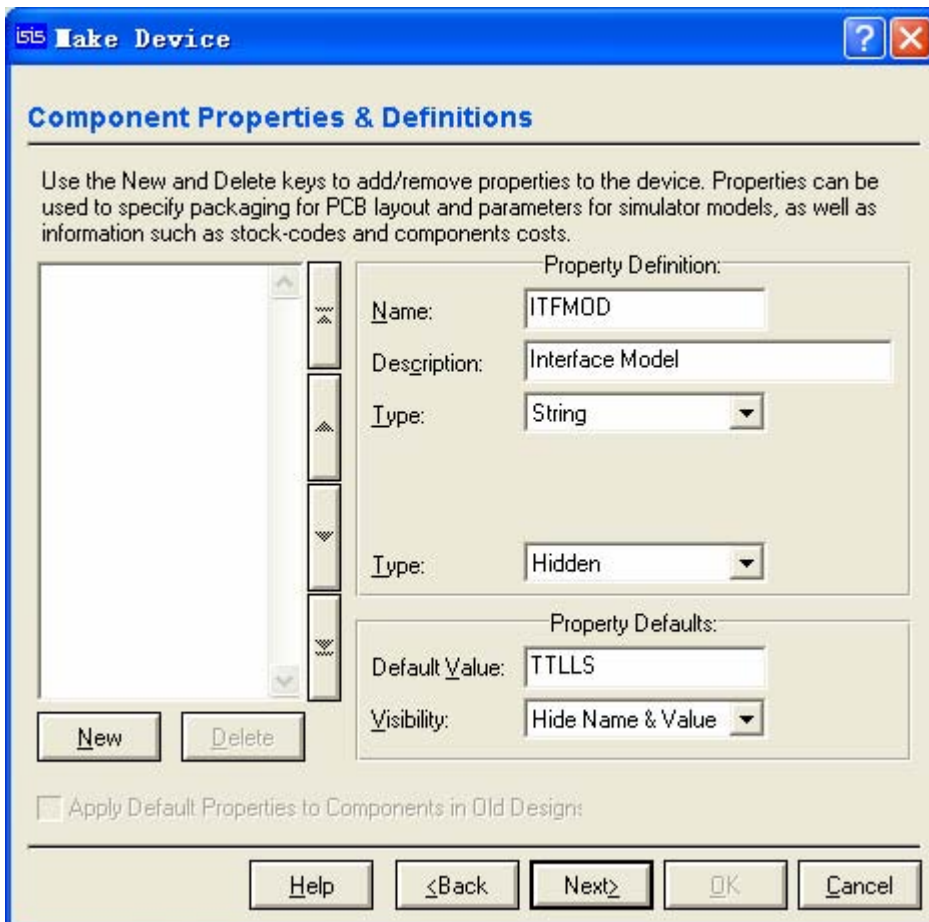
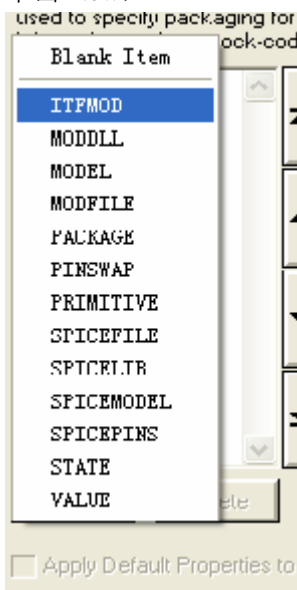
Next，选择 PCB 封装，可不管他



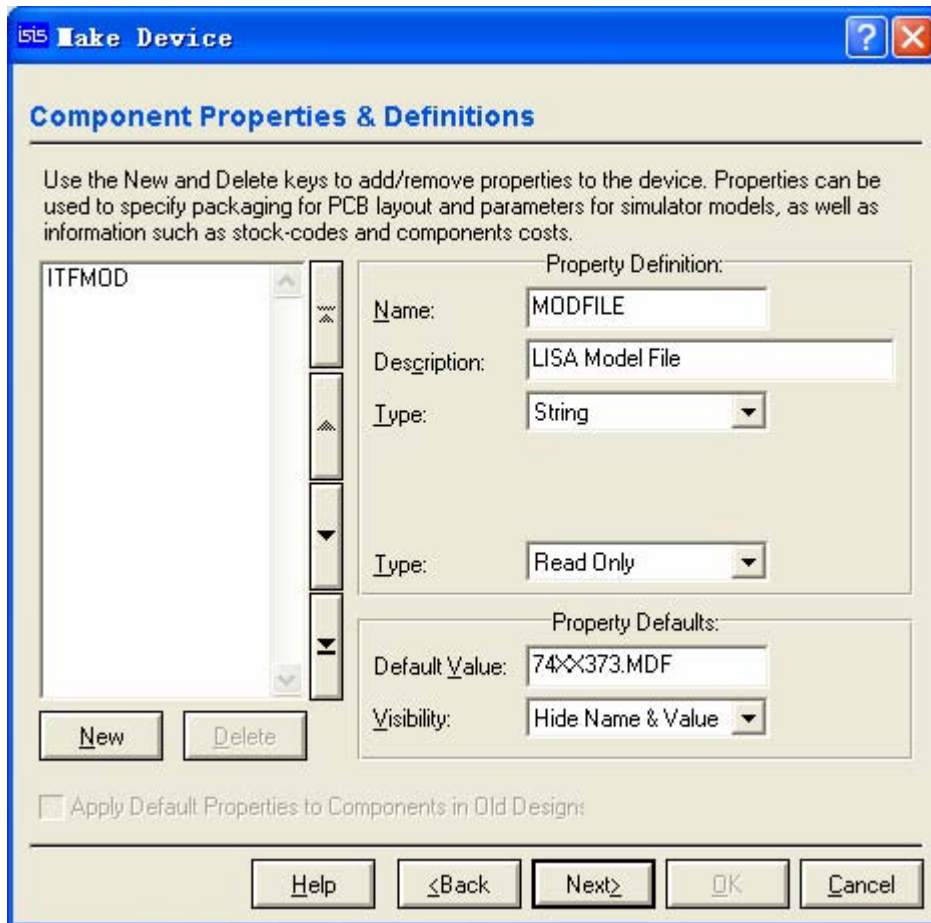
Next, 设置元件的参数



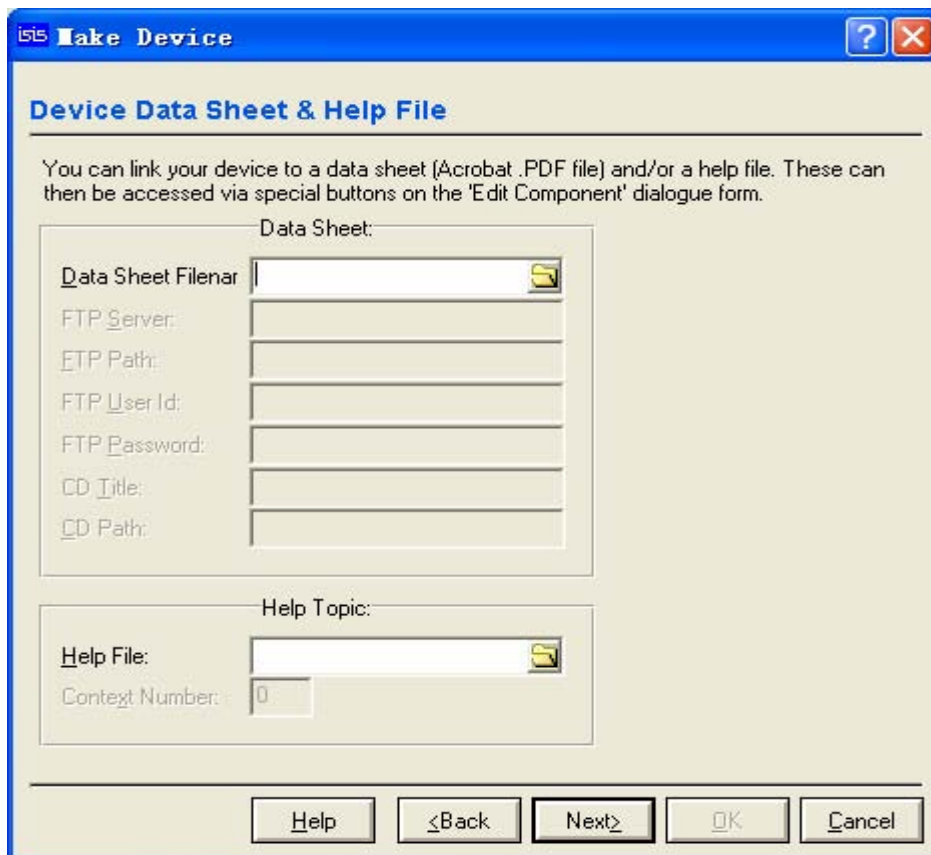
我们需要添加两个属性：{ITFMOD=TTLS}、{MODFILE=74XX373.MDF}
单击 New，



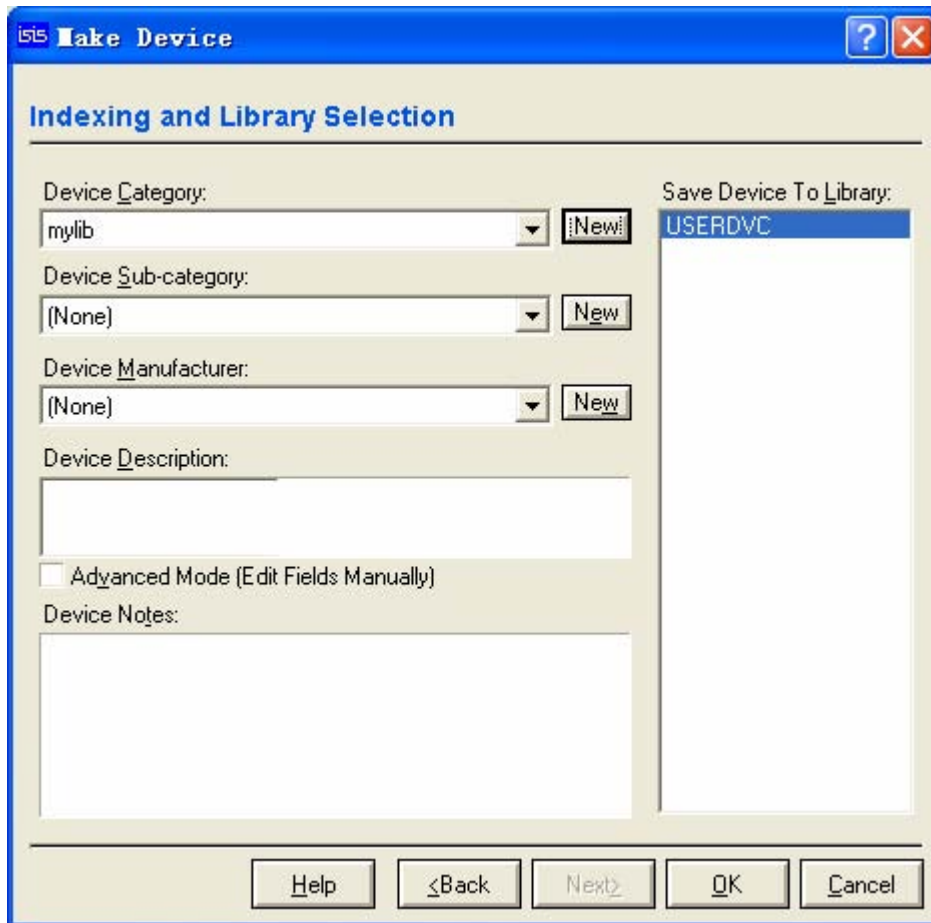
再单击 New



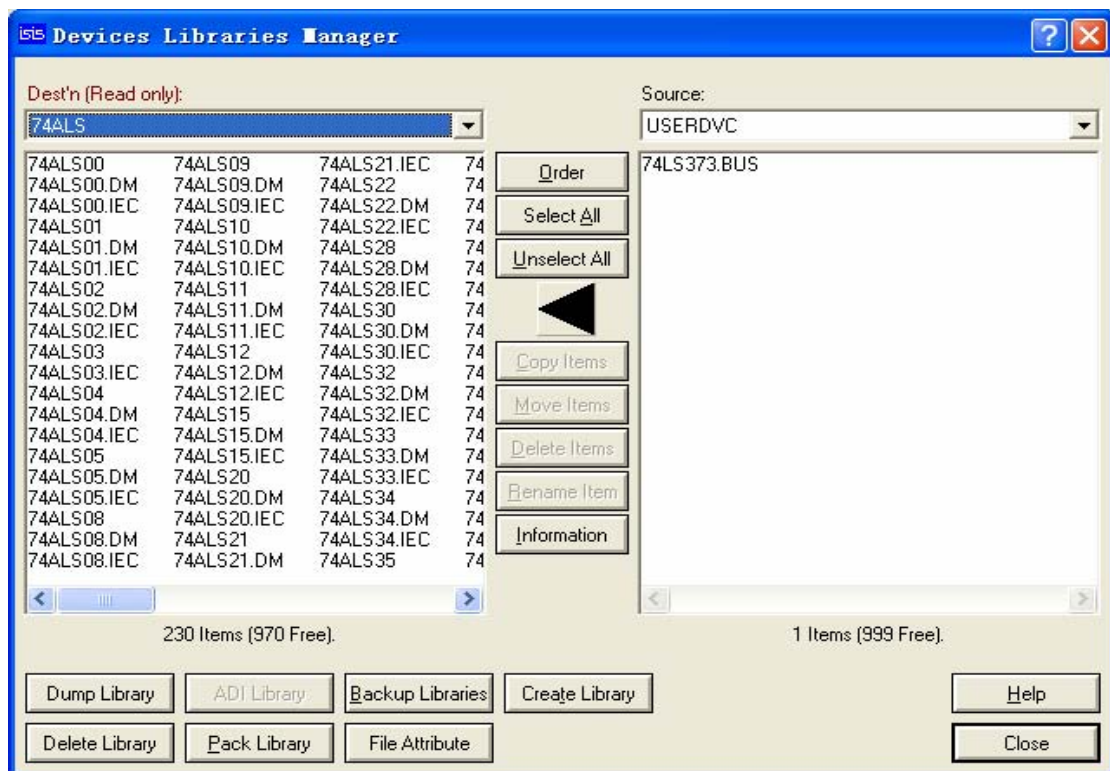
Next, 这不用理



Next, 选择元件存放位置，默认是放在 USERDVC 中的，左边是选择类别，最好自己新建一个，如 mylib。



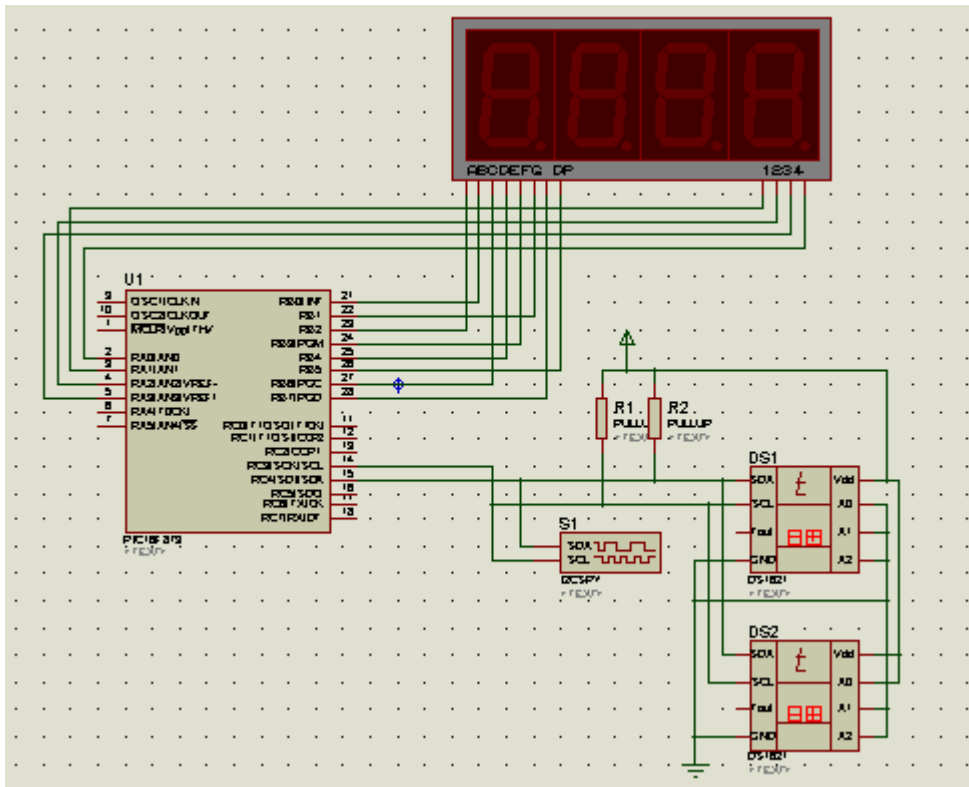
到此为止已经完成元件修改。你可用库管理器自己的元件。



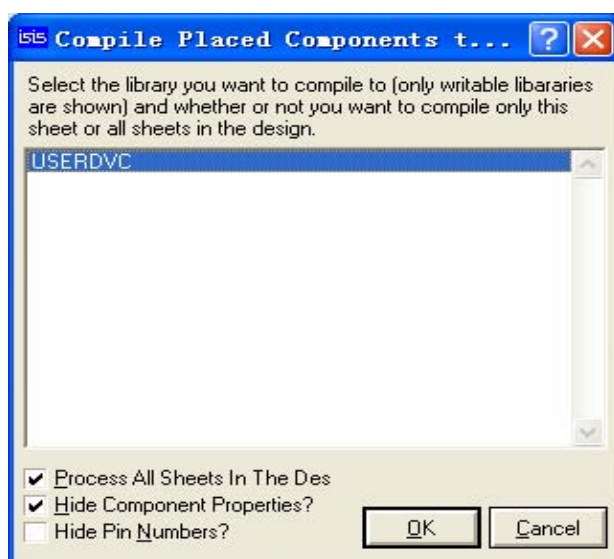
(三) 利用已制作好（别人的）的元件

其实很简单，仿真模型提供者一般会给出三样东西：模型文件（一般为 dll 文件）、例子、库文件。我们要做的工作是：先把 dll 文件拷贝到 Proteus 安装目录下的 MODELS 文件夹里，这样附带的例子就可运行了！如果还附带库文件的话，就把.lib 文件拷贝到 Proteus 安装目录下的 LIBRARY 文件夹里，这样你就可以 Proteus 的库管理器中看到该库文件。如果没有附带库文件，你就要把它添加到你自己的苦里面，方法下面介绍。

先到<http://www.callbus.ru/models.html>下载I2C Spy和DS1621 两个，我们的目标是I2C Spy，但它的例子中要用到DS1621，所以也把DS1621 给下了。于是，我们得到下面文件：I2CSPY.dll、DS1621.dll和test_i2c.DSN等文件。先把I2CSPY.dll和DS1621.dll文件拷贝到Proteus 安装目录下的MODELS文件夹里，运行test_i2c.DSN。

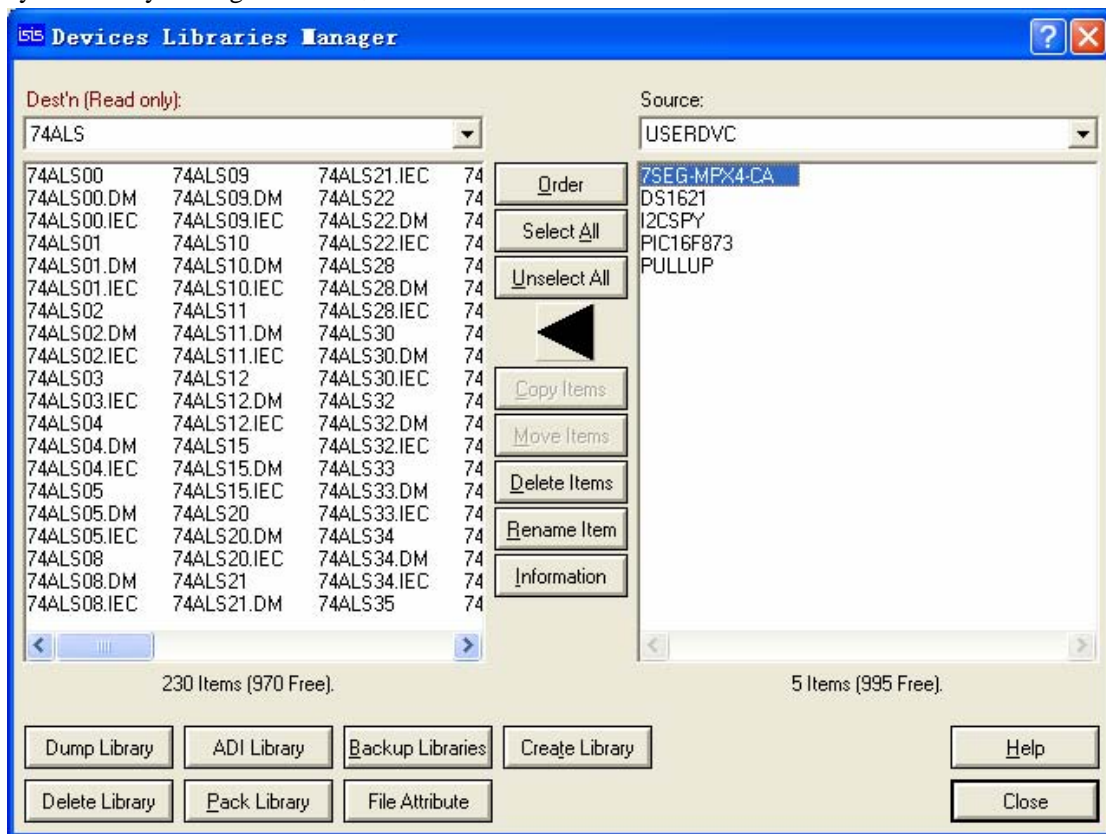


到菜单 Library 的 Compile to library 项，出现下面对话框，单击 OK，这样原理图的



所有元件将被添加到 USERDVC.LIB 中，我们到库管理器中把不需要的删除。菜单


Library 的 Library Manager 项，出现

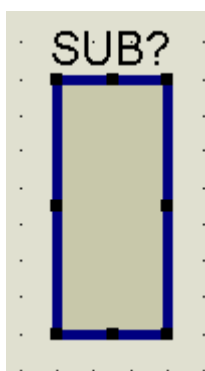



用 Delete Items 把除 I2C Spy 外的元件删除，单击 Close 完成。另外你也用上实例的方案 (1) 的方法。

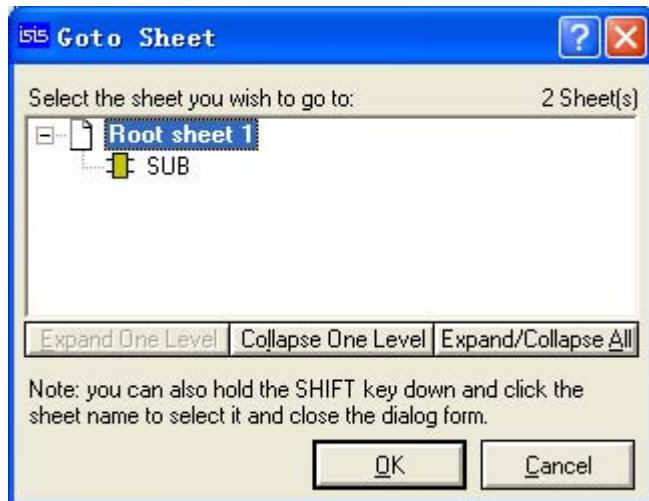
(四) Sub-Circuits 应用

以一个例子介绍，这个例子是将实例一改为 **Sub-Circuits** 形式。

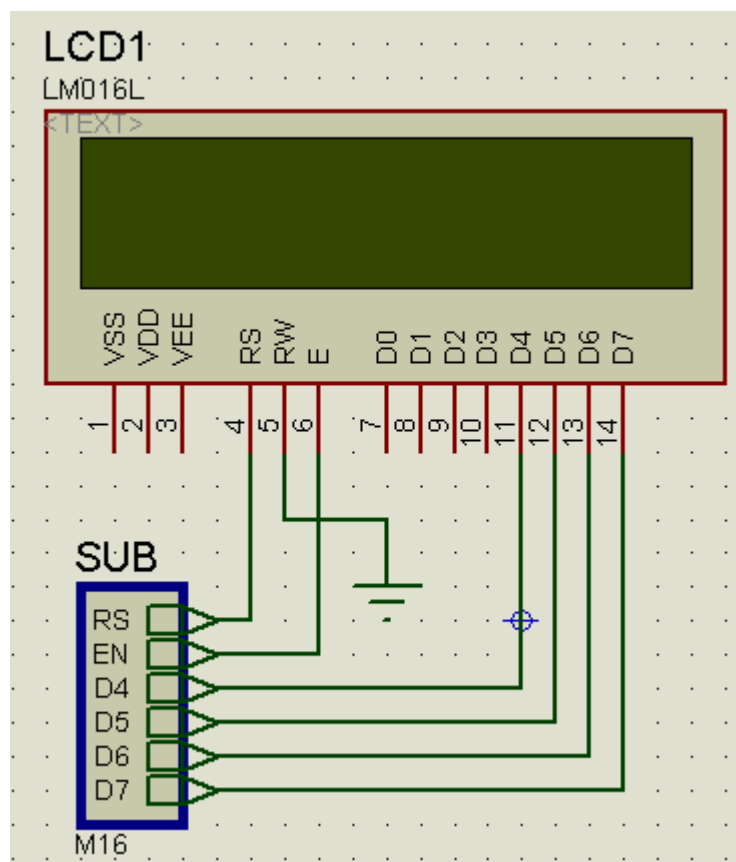
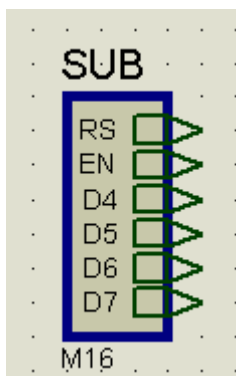
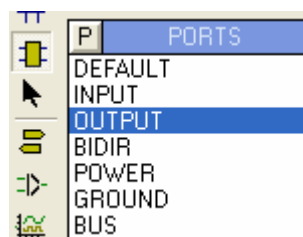
1、绘制 **Sub-Circuits** 符号：点击 ，到原理图窗口拖一个方框修改属性，如右图




这时用工具栏的  查看原理图的层次结构，我们可以看到两张原理图，我们刚才操作的是 Boot sheet1，SUB 是刚生成的，它的名字就是上一步的 Name 属性。我们在右图中选择 SUB 后按 OK 即可切换到 SUB 原理图中。

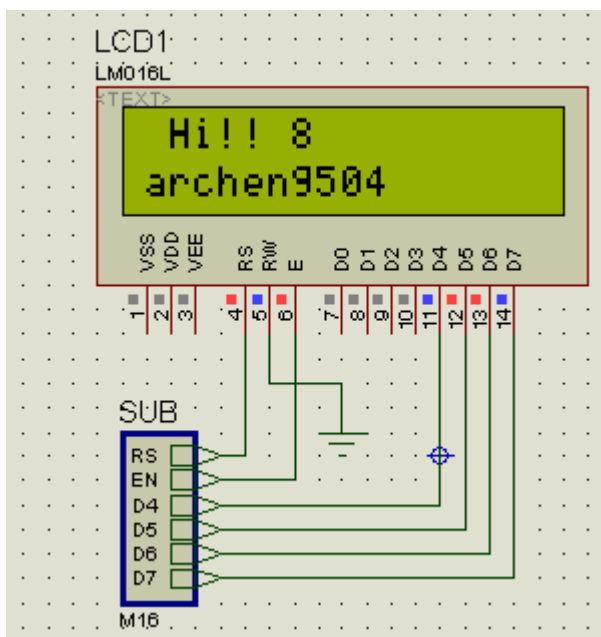
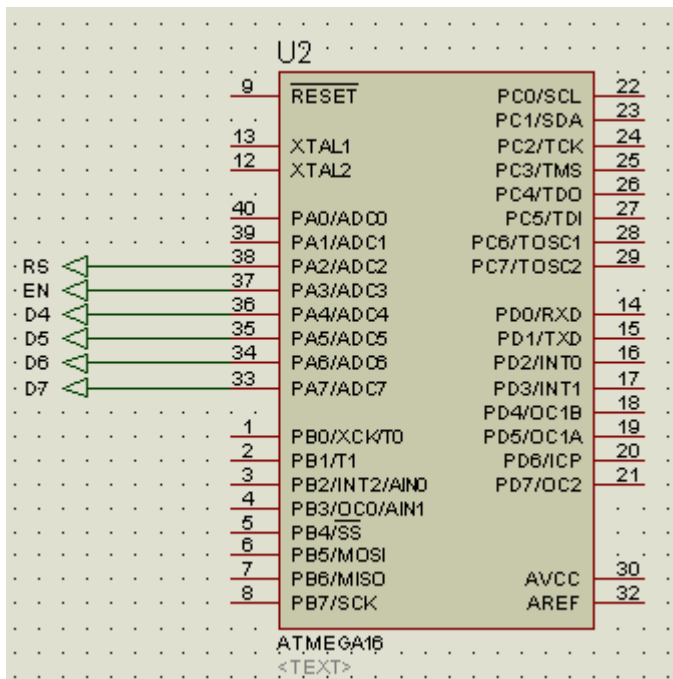
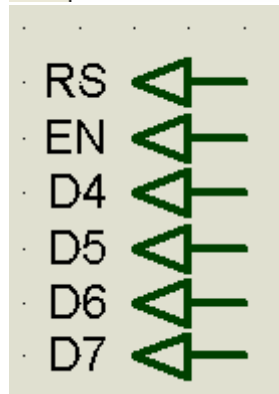
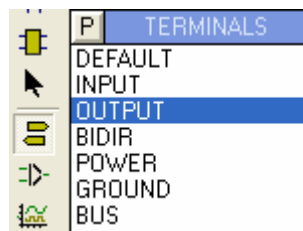


回到 Boot sheet1，我们打算在 Boot sheet1 中绘制 LCD，在 SUB 中绘制 MEGA16。下面，我们来添加引脚，由于这个例子的数据传输方向是单向的，即 M16 到 LCD。因此，我们只用到 OUTPUT。到元件列表选取 OUTPUT 放到 SUB 符号的右边（注：引脚只能放置到 SUB 符号的左右两边），添加 6 个并设置它们的标签名（方法：右击已放置的 OUTPUT 再左击，在 String 中输入标签名），连接 LCD，最终效果



2、绘制 SUB 原理图：切换到 SUB（上面有介绍），这是一个空的原理图，我们先添加

ATMEGA16, 接着单击到元件列表中选择 OUTPUT, 添加 6 个并标上标签名(它们应该跟刚才绘制 SUB 符号的引脚时用到的标签名一致), 连接 ATMEGA16, 最终效果



至此, 这个例子结束了。

注意: 整个原理图的所有元件应该有唯一确定的名称, 如 ATMEGA16 的名称为 U2 不能为 U?, 否则仿真不了!!