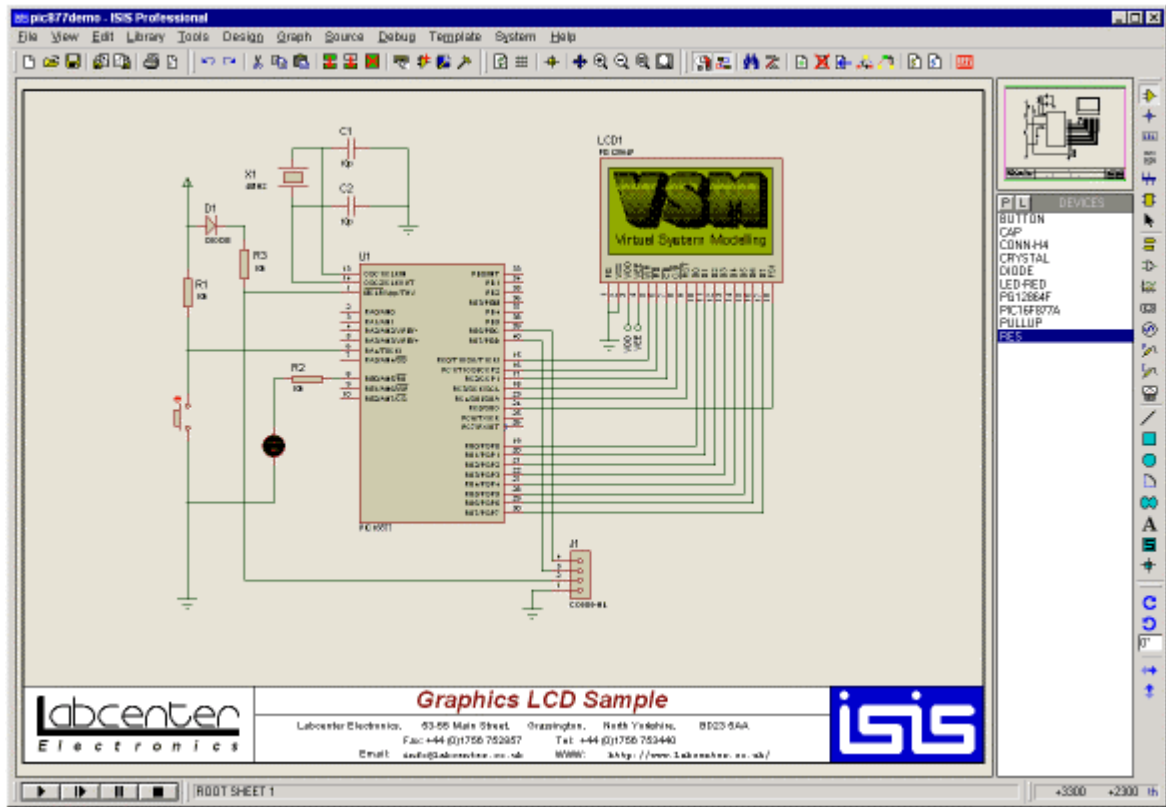
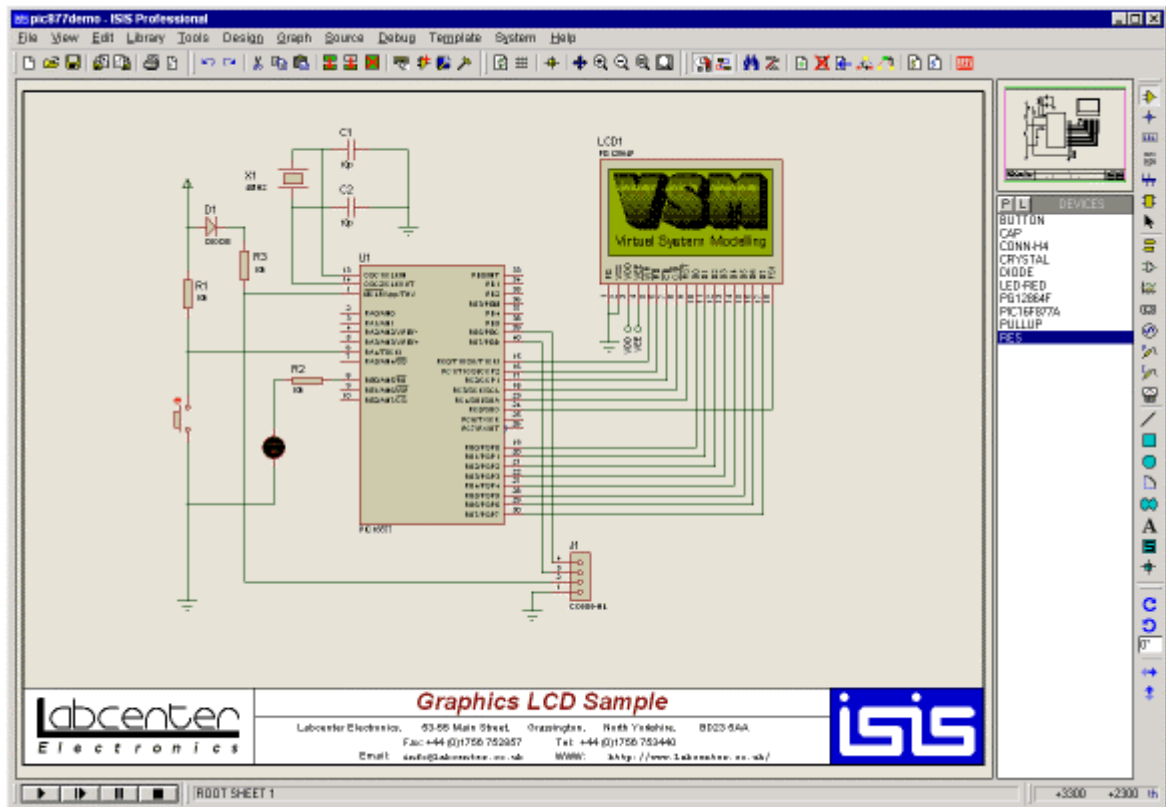


# Keil 与 proteus 完美结合教程

## Proteus 单片机仿真软件



可与 Keil 或 Pic 的 Mplab 联接使用，单片机初学者的福音，有非常丰富的外围器件可供仿真使用



## 简介:

Proteus6.7 是目前最好的模拟单片机外围器件的工具，真的很不错。可以仿真 51 系列、AVR, PIC 等常用的 MCU 及其外围电路（如 LCD, RAM, ROM, 键盘, 马达, LED, AD/DA, 部分 SPI 器件, 部分 IIC 器件, ...）其实 proteus 与 multisim 比较类似，只不过它可以仿真 MCU！

当然，软件仿真精度有限，而且不可能所有的器件都找得到相应的仿真模型，用开发板和仿真器当然是最好选择，可是估计初学者有的可能性比较小吧？

如果你在学 51 单片机，如果你想自己动手做做 LCD, LED, AD/DA, 直流马达, SPI, IIC, 键盘, ... 的小实验的话，试一下吧，不会让你失望的！用 51 不管你是用汇编或是 C 编程当然要用 keil 啦，uvisoin3 有不少新特性呢！使用 keil c51 v7.50 + proteus 6.7 可以像使用仿真器一样调试程序，一般而言，微机实验中用万利仿真器+电工系自己做的实验板的实验都可以做得到吧！当然，硬件实践还是必不可少的!!!

本方案只是在没有硬件的情况下让你能像 pspice 仿真模拟/数字电路那样仿真 MCU 及外围电路。另外，即使有硬件，在程序编写早期用软件仿真一下也不错的！

需要 Proteus6.7 SP3 和 keil c51 v7.50 请到本站：[www.51c51.net](http://www.51c51.net) 下载。本站提供的是完全解密版和专业版。放心使用。

为了使大家能最快的进入 keil c51 v7.50 + proteus 6.7 仿真本站特别编写了本教程共享，希望对本站大力支持！

以下文章 作者：■ 凉山州机电学校 马正强

### 1 Proteus 介绍

Proteus 与其它单片机仿真软件不同的是，它不仅能仿真单片机 CPU 的工作情况，也能仿真单片机外围电路或没有单片机参与的其它电路的工作情况。因此在仿真和程序调试时，关心的不再是某些语句执行时单片机寄存器和存储器内容的改变，而是从工程的角度直接看程序运行和电路工作的过程和结果。对于这样的仿真实验，从某种意义上讲，是弥补了实验和工程应用间脱节的矛盾和现象。

#### (1)proteus 的工作过程

运行 proteus 的 ISIS 程序后，进入该仿真软件的主界面。在工作前，要设置 view 菜单下的捕捉对齐和 system 下的颜色、图形界面大小等项目。通过工具栏中的 p(从库中选择元件命令)命令，在 pick devices 窗口中选择电路所需的元件，放置元件并调整其相对位置，元件参数设置，元器件间连线，编写程序；在 source 菜单的 Define code generation tools 菜单命令下，选择程序编译的工具、路径、扩展名等项目；在 source 菜单的 Add/remove source files 命令下，加入单片机硬件电路的对应程序；通过 debug 菜单的相应命令仿真程序和电路的运行情况。

#### (2)Proteus 软件所提供的元件资源

Proteus 软件所提供了 30 多个元件库，数千种元件。元件涉及到数字和模拟、交流和直流等。

#### (3)Proteus 软件所提供的仪表资源

对于一个仿真软件或实验室，测试的仪器仪表的数量、类型和质量，是衡量实验室是否合格的一个关键因素。在 Proteus 软件包中，不存在同类仪表使用数量的问题。

Proteus 还提供了一个图形显示功能，可以将线路上变化的信号，以图形的方式实时地显示出来，其作用与示波器相似但功能更多。

#### (4) Proteus 软件所提供的调试手段

Proteus 提供了比较丰富的测试信号用于电路的测试。这些测试信号包括模拟信号和数字信号。

对于单片机硬件电路和软件的调试, Proteus 提供了两种方法: 一种是系统总体执行效果, 一种是对软件的分步调试以看具体的执行情况。

对于总体执行效果的调试方法, 只需要执行 debug 菜单下的 execute 菜单项或 F12 快捷键启动执行, 用 debug 菜单下的 pause animation 菜单项或 pause 键暂停系统的运行; 或用 debug 菜单下的 stop animation 菜单项或 shift-break 组合键停止系统的运行。其运行方式也可以选择工具栏中的相应工具进行。

对于软件的分步调试, 应先执行 debug 菜单下的 start/restart debugging 菜单项命令, 此时可以选择 step over、step into 和 step out 命令执行程序(可以用快捷键 F10、F11 和 ctrl+F11), 执行的效果是单句执行、进入子程序执行和跳出子程序执行。在执行了 start / restart debugging 命令后, 在 debug 菜单的下面要出现仿真中所涉及到的软件列表和单片机的系统资源等, 可供调试时分析和查看。

## 2 实验的组织

在单片机的理论教学和实验中, 所涉及到的内容包括了四方面, 即单片机系统资源、软件技术、硬件接口电路、软件和硬件结合的应用系统。因此教师在进行教学时, 应该充分考虑课程的特点并作合理的模块划分, 在每次实验课程前作适当的准备工作, 以使教学任务能集中和突出。以下例子中所涉及到的单片机均为 51 系列。

### (1) 系统资源的实验教学

任何一种单片机均提供了一定的系统资源。对于 51 系列单片机来讲, 其所提供的资源是以寄存器和存储器的方式体现出来的。对于寄存器内容的查看, 可以采用多种可以模拟仿真 51 单片机的软件来实现。对于 Proteus 软件来讲, 可以执行在 debug 菜单下 registers 菜单项命令打开相应的窗口。对于系统内部存储器的查看, 同样是执行 debug 菜单下的 internal memory 菜单项命令打开相应的窗口。

### (2) 软件技术

在软件技术的实验中所涉及到的内容主要包括对语言中具体语句的执行效果验证、算法的具体实现、程序中语法错误的检查和逻辑错误的验证等内容。对于纯软件的实验教学内容, 教师在实验设置时应充分考虑到学生的能力并作好具体的准备工作, 以达到预期的教学目的。

例如, 对于两个 16 位二进制数的乘法运算  $R4R5 \times R6R7 = R3R4R5R6R7$  的汇编语言实现。为了较为直观地验证程序执行的正确性, 教师在之前应该设计一个硬件电路, 能完成数据的输入(带显示)和结果的输出(显示), 并设计主程序、bcd 码到二进制的转换、二进制码到 bcd 码的转换、bcd 码到显示码的转换、显示等程序的设计, 学生只在实验中进行乘法运算的算法设计并作为子程序加入系统中即可。

### (3) 硬件接口电路

单片机的接口电路所涉及到的知识涵盖了电子电路设计的相关领域, 也涉及到单片机应用系统具体的工作需要。接口电路部分涉及到线路的连接、元器件的选择、工作方式的设置、电路工作的稳定性和可靠性、电路的空间需求等等许多方面的知识和内容; 但就学生实验来讲, 所涉及到的并不需要这么全面。作为 Proteus 仿真软件, 其仿真电路时所模拟的元件真实工作情况, 和单片机应用系统的实际工作环境还是有一定的差别。因此只能

根据教学的需要,让学生自行设计电路并设置其工作方式。其实验可以通过两种方式组织实施,一是根据具体的要求,让学生设计接口电路并编写相应的程序;一是根据程序的要求,设计单片机接口电路。

例如,对于键盘输入和显示(数码管)接口电路的设计,可以采用比较多的方法来实现。选择何种方法,与键盘中键的数量和数码管的个数以及显示的方式等都有关系。选择一种采用几种方法都能实现的键盘键数(如 12 个)和显示位数(如 4 位或 4 位半),提出具体的诸如必须预留的口地址、成本控制等要求,让学生自行设计接口电路。当然,对于如 A/D 转换等接口电路的设计,教师应该在事先设计出单片机电路的其它部分,并以模块的方式进行处理,给学生提供电路的接口信号。

#### (4) 软件和硬件结合的应用系统

软件和硬件的结合,就是一个单片机的应用系统了。这一步的实验可以在课程设计中以课题的方式提出来处理,也可以将其分解成不同的接口部分在接口电路的多次实验中分次完成。在这一阶段,硬件电路的设计已经不是最为关键的了,而软件系统的设计、调试和运行才是实验的主要内容。因此可以以建议性的意见给出具体的硬件电路,并提出该电路所需要完成的具体工作,由学生进行软件的设计和调试。

#### (5) 观摩实验

学生自行设计的硬件系统和软件,一般存在着某方面的不足,当给出了相应的实验设计任务后,学生就算能顺利的完成,但其中所涉及到的有些方式方法并不能在自己的设计中充分领悟。Protues 系统提供了相应的一些范例,比如单片机系统中的计算器示例,当教师讲述了其中涉及到的所有(子)程序的功能并将其合理地组织在一起的时候,结合该系统的硬件,将对学生进行程序的设计和系统硬件的设计方法,产生比较大的影响。

### 3 用 Proteus 软件虚拟单片机实验的优点

采用 Proteus 仿真软件进行虚拟单片机实验,具有比较明显的优势,如涉及到的实验实习内容全面、硬件投入少、学生可自行实验、实验过程中损耗小、与工程实践最为接近等。当然其存在的缺点也是有的。

#### (1) 内容全面

内容全面包括其能实验的内容包括软件部分的汇编、C51 等语言的调试过程,也包括硬件接口电路中的大部分类型。对同一类功能的接口电路,可以采用不同的硬件来搭建完成,因此采用 Protues 仿真软件进行实验教学,克服了用单片机实验教学板教学中硬件电路固定、学生不能更改、实验内容固定等方面的局限性,可以扩展学生的思路和提高学生的学习兴趣。

#### (2) 硬件投入少,经济优势明显

对于传统的采用单片机实验教学板的教学实验,由于硬件电路的固定,也就将单片机的 CPU 和具体的接口电路固定了下来。在单片机的实际教学中,如果要涉及到 51 系列,也要涉及到 PIC16 系列,那么为了教学必然要投入两种单片机的实验教学板;同时在教学过程中所涉及到的接口电路,也需要有较大的投入和储备,以利于实验的进行和在实验过程中元件损毁后的更换。Protues 所提供的元件库中,大部分可以直接用于接口电路的搭建,同时该软件所提供的仪表,不管在质量还是数量上,都是可靠和经济的。如果在实验教学中投入这样的真实的仪器仪表,仅仪表的维护来讲,其工作量也是比较大的。因此采用软件的方式进行教学,其经济优势是比较明显的。

#### (3) 学生可自行实验,锻炼解决实际工程问题的能力

实验能力和实验设计能力的培养，是工科学生解决实际工程问题能力中较为重要的。传统的实验教学中，忽视了学生实验能力的培养，对于实验设计能力的培养，则很少涉及到。因此学生学习了理论，要想将其应用到实际的工程实践中，其难度是比较大的。还有，学生毕业后想对单片机控制技术或智能仪表等有较深的研究和学习，如果采用传统的实验教学方法，则学生需要购置的设备比较多，增加了他们学习和研究的投入。采用仿真软件后，学习的投入变得比较的小，而实际工程问题的研究，也可以先在软件环境中模拟通过，再进行硬件的投入，这样处理，不仅省时省力，也可以节省因方案不正确所造成的硬件投入的浪费。

#### (4) 实验过程中损耗小，基本没有元器件的损耗问题

在传统的实验教学过程中，都涉及到因操作不当而造成的元器件和仪器仪表的损毁，也涉及到仪器仪表等工作时所造成的能源消耗。采用 Protues 仿真软件进行的实验教学，则不存在上述的问题，其在实验的过程中是比较安全的。

#### (5) 与工程实践最为接近，可以了解实际问题的解决过程

在进行课程设计或进行大实验的时候，可以具体的在 Protues 中做一个工程项目，并将其最后移植到一个具体的硬件电路中，让学生了解将仿真软件和具体的工程实践如何结合起来，利于学生对工程实践过程的了解和学习。

#### (6) 大量的范例，可供学生参考处理

在系统的设计时，存在对已有资源的借鉴和引用处理，而该仿真系统所提供的较多的比较完善的系统设计方法和设计范例，可供学生参考和借鉴。同时也可以在原设计上进行修改处理。

#### (7) 协作能力的培养和锻炼

一个比较大的工程设计项目，是由一个开发小组协作完成的。了解和把握别人的设计意图和思维模式，是团结协作的基础。在 Proteus 中进行仿真实验时，所涉及到的内容并不全是学生独立设计完成的，因此对于锻炼学生的团结协作意识，是有好处的。

### 4 一个具体的课程实践

该实验为对 8051 单片机外围电路进行写数据仿真的电路。课程要求学生设计电路并提供仿真和验证的方法。

#### (1) 电 路

为了简化电路，将单片机的时钟电路直接用 CLOCK 信号处理；对于数据的写入，采用逻辑分析的方法进行，电路如图 1 所示。

#### (2) 验证方法

逻辑仿真验证程序执行的正确性和电路工作的情况，可通过时序图的方式进行验证，如图 2 所示。

#### (3) 程 序

对 1234H 单元写入数据 AAH，循环进行。

```
ORG 0
loop1: MOV DPTR, #1234H
MOV A, #0AAH
MOVX @DPTR, A
MOVX A, @DPTR
JMP LOOP1
END
```

结 语

Proteus 不仅可以作为学校单片机(电子等)实验的模拟仿真,也可以作为个人工作室的仿真实验。作为电子技术或控制类相关专业的学生和工程技术人员,在学习了该软件后,可以充分地利用它所提供的资源,帮助自己提高工程应用能力。

在教学中利用该软件,除了能教单片机的具体课程内容外,还可以教会学生应用该工具的能力,为今后的工作打下基础。

以上大家对 Proteus 有了一定的了解,下面让我们进入:

## Keil 与 proteus 完美结合

### ● Keil 与 proteus 连接调试

### Proteus 与 Keil 联机 (一)

步骤如下:

**下载**

keil c51 v7.50 软件和 proteus 6.7 SP3 软件, 分别进行安装。安装完毕以后进行如下置:

- 1、 系统安装上 TCP/IP 协议
- 2、 把 proteus 安装目录下 VDM51.dll ( C:\Program Files\Labcenter Electronics\Proteus 6 Professional\MODELS) 文件复制到 Keil 安装目录的 \C51\BIN 目录中。
- 3、 编辑 C51 里 tools.ini 文件,加入: TDRV1=BIN\VDM51.DLL("PROTEUS VSM MONITOR 51 DRIVER")
- 4、 keil 里设置: project-->options for project-->debug tab
- 5、 选中 use proteus VSM monitor 51( 如果想用两台电脑仿真, 双击 setting, 输入 IP 地址 或者 DNS name.
- 6、 载入 proteus 文件
- 7、 proteus 里 DEBUG-->use remote debug monitor

进入 KEIL 的 project 菜单 option for target '工程名'。在 DEBUG 选项中右栏上部的下拉菜单选中 Proteus VSM Monitor-51 Driver。在进入 seting, 如果同一台机 IP 名为 127.0.0.1, 如不是同一台机则填另

一台的 IP 地址。端口号一定为 8000 注意：可以在一台机器上运行 keil，另一台中运行 proteus 进行远程仿真哦~。

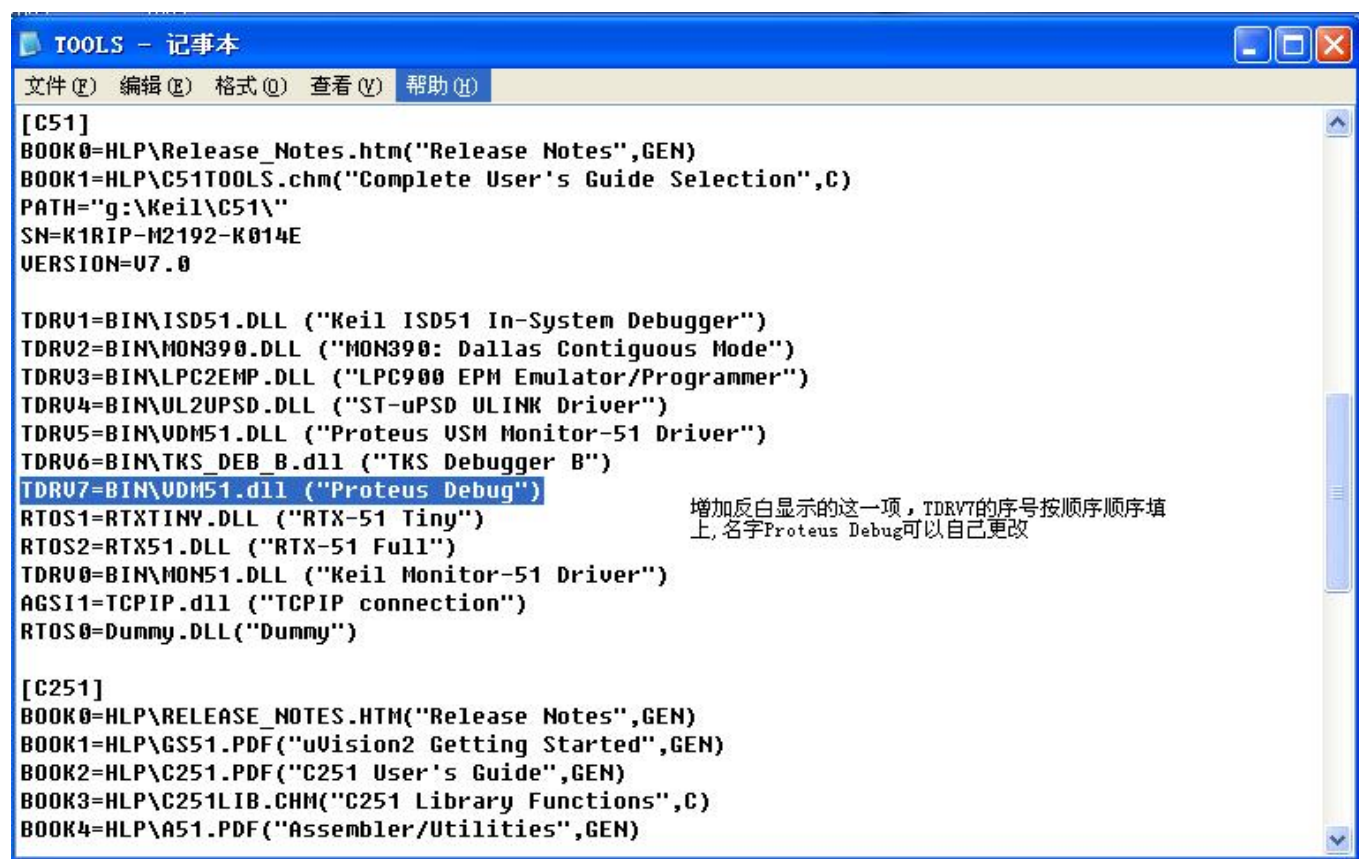
8、打开 KEIL，按 F5 开始仿真

## Proteus 与 Keil 联机（二）

摘自零距离单片机实验室：

<http://www.mkdz.cn/Announce/announce.asp?BoardID=100&ID=1098&Ar=840&AUpflag=1&Ap=1&Aq=1>

- 1.把压缩文件中的 VDM51.DLL 解压到 Keil 目录下的 c51\BIN
- 2.修改 Keil 目录下的 TOOLS.ini 文件如图



```
TOOLS - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

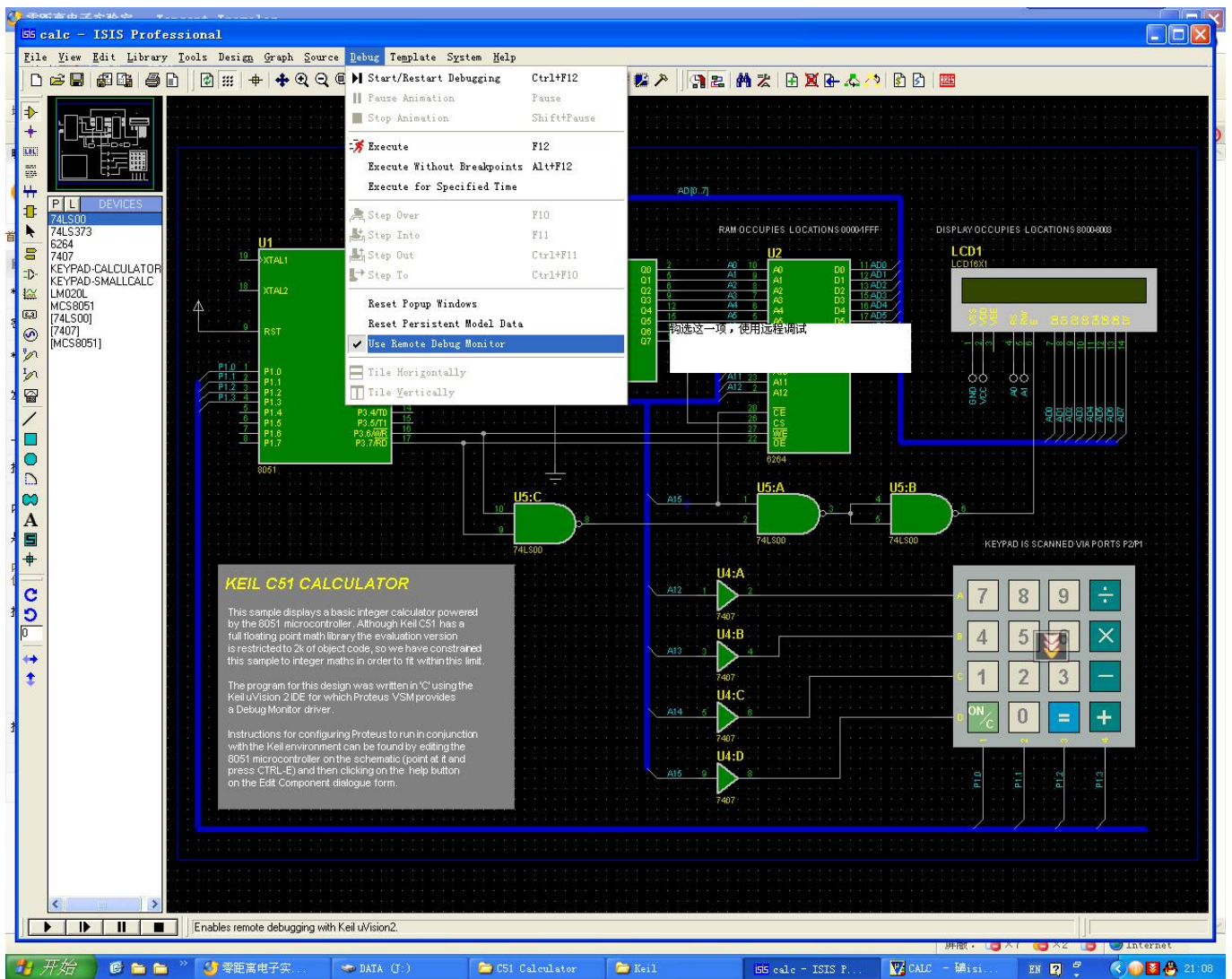
[C51]
BOOK0=HLP\Release_Notes.htm("Release Notes",GEN)
BOOK1=HLP\C51TOOLS.chm("Complete User's Guide Selection",C)
PATH="g:\Keil\C51\"
SN=K1RIP-M2192-K014E
VERSION=V7.0

TDRU1=BIN\ISD51.DLL ("Keil ISD51 In-System Debugger")
TDRU2=BIN\MON390.DLL ("MON390: Dallas Contiguous Mode")
TDRU3=BIN\LPC2EMP.DLL ("LPC900 EPM Emulator/Programmer")
TDRU4=BIN\UL2UPSD.DLL ("ST-uPSD ULINK Driver")
TDRU5=BIN\VDM51.DLL ("Proteus USM Monitor-51 Driver")
TDRU6=BIN\TKS_DEB_B.dll ("TKS Debugger B")
TDRU7=BIN\VDM51.dll ("Proteus Debug")
RTOS1=RTXTINY.DLL ("RTX-51 Tiny")
RTOS2=RTX51.DLL ("RTX-51 Full")
TDRU0=BIN\MON51.DLL ("Keil Monitor-51 Driver")
AGS11=TCPIP.dll ("TCP/IP connection")
RTOS0=Dummy.DLL("Dummy")

[C251]
BOOK0=HLP\RELEASE_NOTES.HTM("Release Notes",GEN)
BOOK1=HLP\GS51.PDF("uVision2 Getting Started",GEN)
BOOK2=HLP\C251.PDF("C251 User's Guide",GEN)
BOOK3=HLP\C251LIB.CHM("C251 Library Functions",C)
BOOK4=HLP\A51.PDF("Assembler/Utilities",GEN)
```

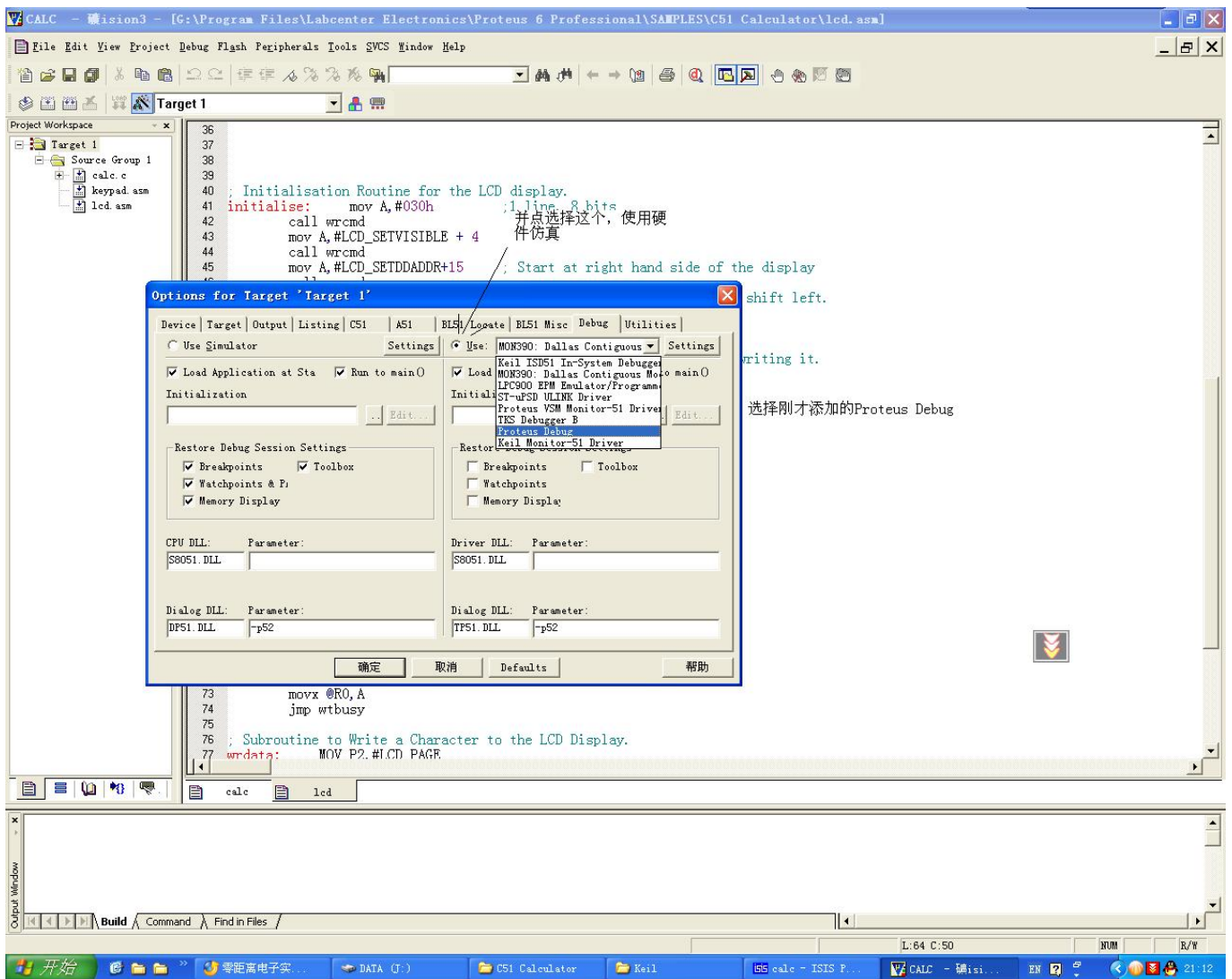
如果 Keil 正在运行，则需重新启动

- 3.使用 ISIS 打开 Labcenter Electronics\Proteus 6 Professional\SAMPLES\C51 Calculator 原理图，



4. 打开 Keil 选择 Labcenter Electronics\Proteus 6 Professional\SAMPLES\C51 Calculator 目录下的 Keil 工程





5.在 Keil 下重新编译一下，并按调试按钮

**Project Workspace**

Register	Value
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x00
b	0x00
sp	0x35
dptr	0x0000
PC	0x467b
states	369
sec	0.00369
psw	0x00

```

84: wtbody: MOV R1,#LCD_BUSY_RD
C:0x4674 7902 MOV R1,#0x02
85: MOVX A,@R1
C:0x4676 E3 MOVX A,@R1
86: JB ACC.7,wtbody
C:0x4677 20E7FA JB ACC.7,wtbody(C:4674)
87: ret
C:0x467A 22 RET
20: VOID main (VOID)
21: //Initialise our variables and call the
22: //Assembly routine to initialise the LCD display.
23: { lvalue = 0;
C:0x467B E4 CLR A
C:0x467C F52E MOV 0x2E,A
C:0x467E F52D MOV 0x2D,A
C:0x4680 F52C MOV 0x2C,A
C:0x4682 F52B MOV 0x2B,A
24: rvalue = 0;
C:0x4684 F532 MOV 0x32,A
C:0x4686 F531 MOV 0x31,A
C:0x4688 F530 MOV 0x30,A
C:0x468A F52F MOV 0x2F,A
25: currtoken = '=';
C:0x468C 75343D MOV 0x34,#0x3D
26: lasttoken = '0';
C:0x468F 753330 MOV 0x33,#0x30
27: initialise(); // Initialize the LCD
C:0x4692 124640 LCALL INITIALISE(C:4640)
28: calc_output(OK);
C:0x4695 E4 CLR A
C:0x4696 FF MOV R7,A
C:0x4697 FE MOV R6,A
C:0x4698 12469E LCALL calc_output(C:469E)
29: calc_evaluate();
C:0x469B 0243C9 LJMP calc_evaluate(C:43C9)
161: VOID calc_output (INT status)
162: // Output according to the status of the operation.
163: // *Sleep* is used for the first op press after a full cancel
164: // or on startup.
165: { switch (status)
166: { case OK : calc_display(calc_decascii(lvalue)); break;
C:0x469E EE MOV A,R6
    
```

**Output Window**

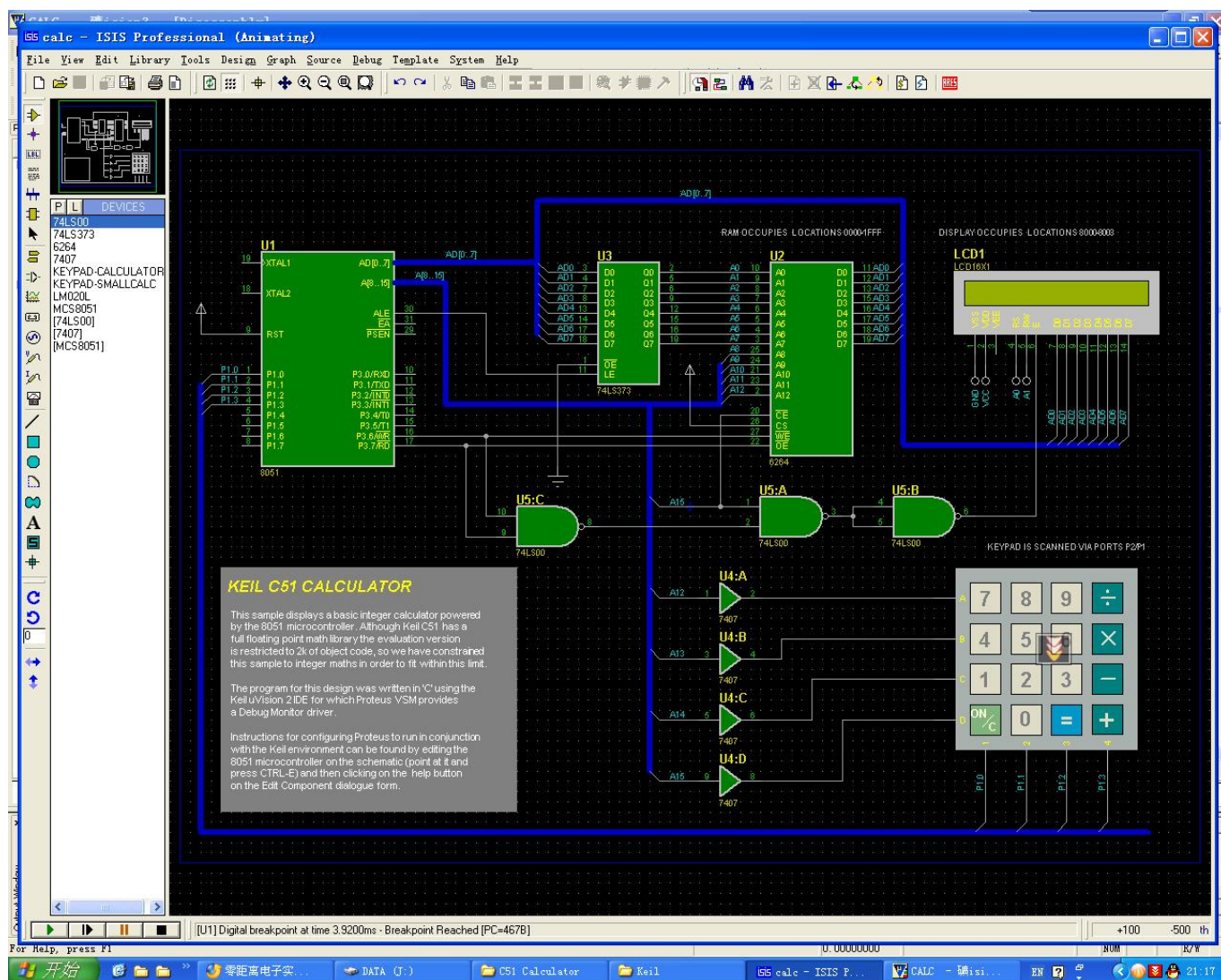
```

VDM51 target initialized.
Load "G:\Program Files\Labcenter Electronics\Proteus 6 Professional\SAMPLES\C51 Calculator\calc"
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go
    
```

For Help, press F1

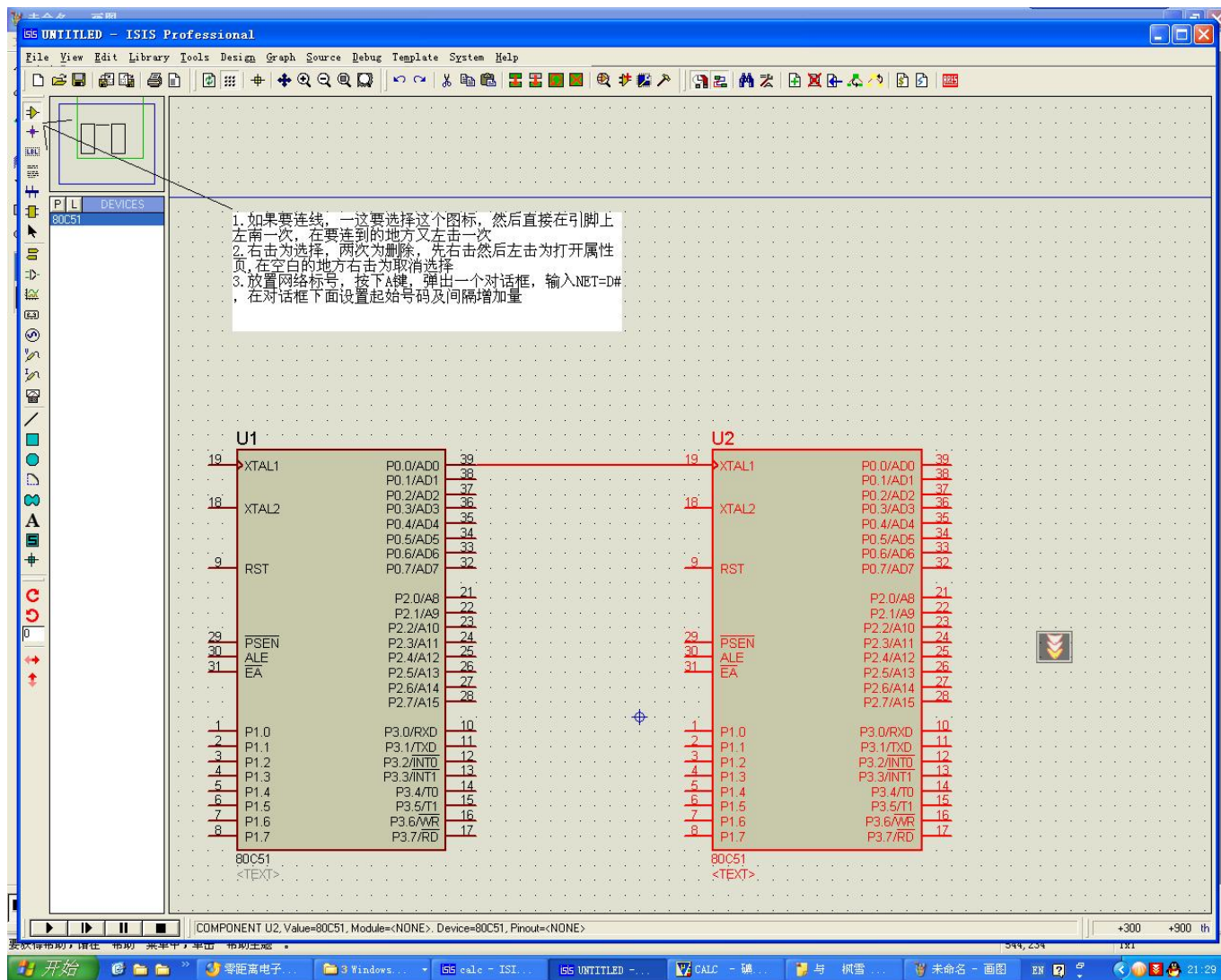
0.00000000 NUM E/W

开始 零阳电子实... DATA (J) C51 Calculator Keil 65 calc - ISIS P... CALC - 编... EN 21:16



是不是 Keil 与 Proteus 都一起运行起来了，和用硬件仿真差不多吧！

使用 Proteus 一点点的烂经验：



## 基本概念

### 坐标系统 (CO-ORDINATE SYSTEM)

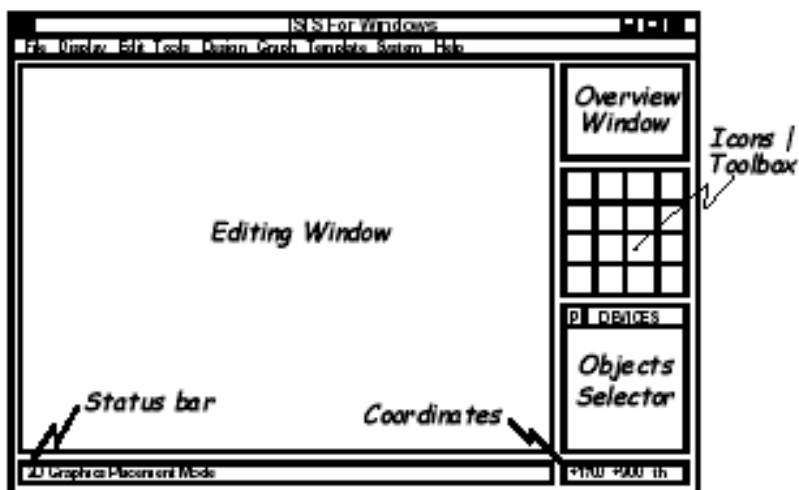
ISIS中坐标系统的基本单位是10nm，主要是为了和ARES保持一致。但坐标系统的识别 (read-out) 单位被限制在1 thou。坐标原点位于工作区的中间，所以既有正坐标值，又有负坐标值。坐标位置指示器位于屏幕的右下角。

一旦实时捕捉 (Real-Time Snap) 功能使能，当鼠标指针指向管脚末端或者导线时，X、Y坐标之一或二者都被加亮显示，加亮显示说明鼠标指针位置已经被捕捉。例如，如果鼠标指针在一条横线附近，它将会被捕捉到Y轴，Y坐标将会被加亮。

### 屏幕外观 (SCREEN LAYOUT)

#### 概述 (Introduction)

整个屏幕被分成三个区域——编辑窗口 (Editing Window)、预览窗口 (Overview Window)、工具箱，如下图所示：



编辑窗口显示你正在编辑的电路原理图，可以通过View菜单的Redraw命令来刷新显示内容，同时预览窗口中的内容也将被刷新。当执行其它命令导致显示错乱时可以使用该特性恢复显示。

要使编辑窗口显示一张大的电路图的其他部分，可以通过如下几种方式：

- 用鼠标左键点击预览窗口中想要显示的位置，这将使编辑窗口显示以鼠标点击处为中心的内容。
- 在编辑窗口内移动鼠标，按下SHIFT键，用鼠标“撞击”边框，这会使显示平移。我们把这称为*Shift-Pan*。
- 用鼠标指向编辑窗口并按 缩放键（见下文），会以鼠标指针位置为中心重新显示。

### 缩放（Zooming）

按F6键可以放大电路图（连续按会不断放大直到最大），按F7键可以缩小电路图（类似的连续按会不断缩小直到最小），这两种情况无论哪种都会使显示以当前鼠标位置为中心重新显示。按F8键可以把一整张图缩放到完全显出来。图的大小可以通过View菜单的Zoom命令或者是上述的功能键控制。无论何时你都可以使用功能键控制缩放，即便是在滚动和拖放对象时。

另外，按着SHIFT键，同时在一个特定的区域用鼠标左键拖一个框，则框内的部分就会被放大，这个框可以是在编辑窗口内拖，也可以是在预览窗口内拖。

### 点状栅格（The Dot Grid）

编辑窗口内有点状的栅格，可以通过View菜单的Grid命令在打开和关闭间切换。点与点之间的间距由当前捕捉的设置决定？？？？？？？？？？？？

### 捕捉到栅格（Snapping to a Grid）

你会注意到鼠标在编辑窗口内移动时，坐标值是以固定的步长的增长的——初始设定是100。这称为捕捉，能够使你把元件按栅格对齐。捕捉的尺度可以由View菜单的Snap命令设置，或者直接使用快捷键F4、F3、F2和CTRL+F1。

如果你想要确切地看到捕捉位置，可以使用View菜单的X-Cursor命令，选中后将会在捕捉点显示一个小的或大的交叉十字。

### 实时捕捉（Real Time Snap）

当鼠标指针指向管脚末端或者导线时，鼠标指针将会被捕捉到这些物体，这种功能被称为实时捕捉，该功能可以使你方便的实现导线和管脚的连接。可以通过Tools菜单的*Real Time Snap* 命令或者是CTRL+S切换该功能。

### 预览窗口（The Overview Window）

该窗口通常显示整个电路图的缩略图，上面有一个half-inch的格子。The cyan box 标示出图的边框，同时窗口上的绿框标出在编辑窗口的中显示的区域。

在预览窗口上点击鼠标左键，将会以点击位置为中心刷新编辑窗口。

其他情况下，预览窗口显示将要放置的对象的预览。这种*Place Preview*特性在下列情况下被激活：

- 当一个对象在选择器中被选中
- 当使用旋转或镜像按钮时
- 当为一个可以设定朝向的对象选择类型图标时（例如：*Component icon, Device Pin icon*等等）

当放置对象或者执行其他非以上操作时，place preview会自动消除

### 工具箱（The Toolbox）

该区域包括一些图标 (icons) 和一个项目选择器 (item selector)。

上部的八组图标用来选择放置不同的对象,下部的八组图标进行相应的控制。比较特殊的,第三行的前三个图标对上部的图标进行不同对象组的切换。在手册的前面部分的快速参考中有图标的所有功能和使用方式。

对象选择器 (Object Selector) 根据由图标决定的当前状态显示不同的内容。显示对象的类型包括:设备,终端,管脚,图形符号,标注和图形。

在某些状态下,对象选择器有一个Pick切换按钮,点击该按钮可以弹出库元件选取窗体。通过该窗体可以选择元件并置入对象选择器,在今后绘图时使用。

## 概述 (Introduction)

## 文件命令 (FILING COMMANDS)

ISIS使用了下列的文件类型:

设计文件 (Design Files)

部分文件 (Section Files)

模块文件 (Module Files)


库文件 (Library Files)

设计包含一个电路的所有信息,文件扩展名为“DSN”。ISIS以前的版本曾用过的扩展名有“ISS”,“IDS”,“IWS”,如果在目录PROTEUS下安装了文件转换器IDSCVT40.DLL和(或)IWSCVT40.DLL,这些扩展名的文件就可以自动转换过来。

部分的原理图可以倒出为部分文件,然后可以读入到其它文件。扩展名是SEC。可以用文件 (File) 菜单中的“导入(Import)”和“导出(Export)”命令来读和写。

模块文件的扩展名是“MOD”,可以和它的功能一起使用来实现层次设计。更详细的情况参照“层次设计(HIERARCHICAL DESIGNS)”。

符号和器件的库文件的后缀是“LIB”。

 Proteus VSM仿真系统也可以使用其它的文件类型。详细内容请参照VSM用户手册。

## 开始一个新的设计 (Starting a New Design)

命令“New Design”将清除所有原有的设计数据,出现一张空的A4纸。新设计的缺省名字为UNTITLED.DSN,“Save Design”命令会把设计以这个名字存入磁盘文件中,文件的其他选项也会使用它作为缺省名。

如果你要在开始新设计的时候给这个设计命名,那就使用“装入设计”命令,输入新的文件名。

## 装入设计 (Loading a Design)

装入设计的方法有两种:

- 1.在DOS提示符下输入: ISIS<my design>
- 2.在ISIS系统中使用“装入设计”命令。

## 保存设计 (Saving the Design)

你可以在退出ISIS系统或者其它任何时候保存设计,使用“Save Design”命令。两种情况下,设计都被存到装入时的文件中。旧的版本就会在名字前加了前缀“Backup of”。

“另存为”命令可以把设计保存到另一个文件中。

## 导入/导出部分文件 (Import / Export Section)

文件菜单中的“导出”命令可以就当前选中的对象生成一个部分文件。这个文件可以使用“导入”命令读入另一个设计。部分文件的操作类似于“块拷贝”。

## 退出ISIS

想要退出ISIS系统,用菜单项中的“退出”命令或者键入“Q”。如果文件修改过,系统会出现对话框问是否保存文件。

## 基本的编辑工具 (GENERAL EDITING FACILITIES)

### 对象放置 (Object Placement)

ISIS支持多种类型的对象,每一类型对象的具体作用和功能将在下一章给出。虽然类型不同,但放置对象的基本步骤都是一样的。

### 放置对象的步骤如下 (To place an object:)

1. 根据对象的类别在工具箱选择相应模式的图标 (mode icon)。
2. Select the sub-mode icon for the specific type of object.
- 2、根据对象的具体类型选择子模式图标 (sub-mode icon)。
- 3、如果对象类型是元件、端点、管脚、图形、符号或标记，从选择器里 (selector) 选择你想要的对象的名字。对于元件、端点、管脚和符号，可能首先需要从库中调出。
- 4、如果对象是有方向的，将会在预览窗口显示出来，你可以通过点击旋转和镜象图标来调整对象的朝向。
- 5、最后，指向编辑窗口并点击鼠标左键放置对象。对于不同的对象，确切的步骤可能略有不同，但你会发现和别的图形编辑软件是类似的，而且很直观。

### 选中对象 (Tagging an Object)

用鼠标指向对象并点击右键可以选中该对象。该操作选中对象并使其高亮显示，然后可以进行编辑。

- 选中对象时该对象上的所有连线同时被选中。
- 要选中一组对象，可以通过依次在每个对象右击选中每个对象的方式。也可以通过右键拖出一个选择框的方式，但只有完全位于选择框内的对象才可以被选中。
- 在空白处点击鼠标右键可以取消所有对象的选择。

### 删除对象 (Deleting an Object)

用鼠标指向选中的对象并点击右键可以删除该对象，同时删除该对象的所有连线。

### 拖动对象 (Dragging an Object)

用鼠标指向选中的对象并用左键拖曳可以拖动该对象。该方式不仅对整个对象有效，而且对对象中单独的labels也有效。

- 如果Wire Auto Router功能被使能的话，被拖动对象上所有的连线将会重新排布或者'fixed up'。这将花费一定的时间（10秒左右），尤其在对象有很多连线的情况下，这时鼠标指针将显示为一个沙漏。
- 如果你误拖动一个对象，所有的连线都变成了一团糟，你可以使用Undo命令撤消操作恢复原来的状态。

### 拖动对象标签 (Dragging an Object Label)

许多类型的对象有一个或多个属性标签附着。例如，每个元件有一个“reference”标签和一个“value”标签。可以很容易地移动这些标签使你的电路图看起来更美观。

### 移动标签的步骤如下 (To move a label)

- 1、选中对象
- 2、用鼠标指向标签，按下鼠标左键。
- 3、拖动标签到你需要的位置。如果想要定位的更精确的话，可以在拖动是改变捕捉的精度（使用F4、F3、F2、CTRL+F1键）。

#### 1. 释放鼠标

### 调整对象大小 (Resizing an Object)

子电路 (Sub-circuits)、图表、线、框和圆可以调整大小。当你选中这些对象时，对象周围会出现白色小方块叫做“手柄”，可以通过拖动这些“手柄”来调整对象的大小。

### 调整对象大小的步骤如下 (To resize an object)

- 1、选中对象
  - 2、如果对象可以调整大小，对象周围会出现白色小方块，叫做“手柄”。
  - 3、用鼠标左键拖动这些“手柄”到新的位置，可以改变对象的大小。在拖动的过程中手柄会消失以便不和对象的显示混叠。
- ☞ 可以参照④调整二维图形的大小这一节来获得使用“手柄”调整对象大小的更进一步的信息。

### 调整对象的朝向 (Reorienting an Object)

许多类型的对象可以调整朝向为0°, 90°, 270°, 360°或通过x轴y轴镜象。当该类型对象被选中后，“Rotation and Mirror”图标会从兰色变为红色，然后就可以来改变对象的朝向。

### 调整对象朝向的步骤如下 (To reorient an object)

- 1、选中对象
- 2、用鼠标左键点击Rotation图标可以使对象逆时针旋转，用鼠标右键点击Rotation图标可以使对象顺时针旋转。

3、用鼠标左键点击*Mirror*图标可以使对象按x轴镜像，用鼠标右键点击*Mirror*图标可以使对象按y轴镜像。

毫无疑问当*Rotation and Mirror*图标是红色时，操作他们将回改变某个对象，即便你当前没有看到它，实际上，这中颜色的指示在你想对将要放置的新对象操作时是格外有用的。当图标是红色时，首先取消对象的选择，此时图标会变成蓝色，说明现在可以“安全”调整新对象了。

### 编辑对象 (Editing an Object)

许多对象具有图形或文本属性，这些属性可以通过一个对话框进行编辑，这是一中很常见的操作，有多种实现方式。

**编辑单个对象的步骤是 (To edit a single object using the mouse) :**

- 1、选中对象
- 2、用鼠标左键点击对象。

**连续编辑多个对象的步骤是 (To edit a succession of objects using the mouse)**

- 1、选择*Main Mode*图标，再选择*Instant Edit*图标。
- 2、依次用鼠标左键点击各个对象

**以特定的编辑模式编辑对象的步骤是 (To edit an object and access special edit modes:)**


- 1、指向对象
- 2、使用键盘CTRL+'E'

对于文本脚本来说，这将启动外部的文本编辑器。如果鼠标没有指向任何对象的话，该命令将对当前的图进行编辑。

**通过元件的名称编辑元件的步骤如下 (To edit a component by name:)**

- 1、键入'E'
- 2、在弹出的对话框中输入元件的名称 (part ID)。

确定后将会弹出该项目中任何元件的编辑对话框，并非只限于当前sheet的元件。编辑完后，画面将会以该元件为中心重新显示。你可以通过该方式来定位一个元件，即便你并不想对其进行编辑。

 在OBJECT SPECIFICS这一节中将详细说明对应于每种对象类型的具体编辑操作方式。

### 编辑对象标签 (Editing An Object Label)

元件、端点、线和总线标签都可以象元件一样编辑。

**编辑单个对象标签的步骤是 (To edit a single object label using the mouse:)**

- 1、选中对象标签。
- 2、用鼠标左键点击对象。

**连续编辑多个对象标签的步骤是 (To edit a succession of object labels using the mouse:)**

- 1、选择*Main Mode*图标，再选择*Instant Edit*图标。
- 2、依次用鼠标左键点击各个标签。

任何一种方式，都将弹出一个带有*Label and Style*栏的对话框窗体。可以参照指南中*Editing Local Styles*这一节得到编辑*local*文本类型的详细内容。

### 拷贝所有选中的对象 (Copying all Tagged Objects)

**拷贝一整块电路的方式 (To copy a section of circuitry:)**


- 1、选中需要的对象，具体的方式参照上文的*Tagging an Object*部分。
- 2、用鼠标左键点击*Copy*图标。
- 3、把拷贝的轮廓拖到需要的位置，点击鼠标左键放置拷贝。
- 4、重复步骤[3]放置多个拷贝。
- 5、点击鼠标右键结束

当一组元件被拷贝后，他们的标注自动重置为随机态，用来为下一步的自动标注做准备，防止出现重复的元件标注。

### 移动所有选中的对象 (Moving all Tagged Objects)

**移动一组对象的步骤是 (To move a set of objects:)**

- 1、选中需要的对象，具体的方式参照上文的*Tagging an Object*部分。
- 2、把轮廓拖到需要的位置，点击鼠标左键放置。

 你可以使用块移动的方式来移动一组导线，而不移动任何对象。更进一步的讨论可以参照*Dragging Wires*这一节。

### 删除所有选中的对象 (Deleting all Tagged Objects)



### 删除一组对象的步骤是 (To delete a group of objects:)

- 1、选中需要的对象，具体的方式参照上文的Tagging an Object部分。
- 2、用鼠标左键点击Delete图标。

如果错误删除了对象，可以使用Undo命令来恢复原状。

## 画线 (WIRING UP)

### 画线 (Wire Placement)

你一定发现没有画线的图标按钮。这是因为ISIS的智能化足以在你想要画线的时候进行自动检测。这就省去了选择画线模式的麻烦。

### 在两个对象间连线 (To connect a wire between two objects)

- 1、左击第一个对象连接点。
- 2、如果你想让ISIS自动定出走线路径，只需左击另一个连接点。另一方面，如果你想自己决定走线路径，只需在想要拐点处点击鼠标左键。

一个连接点可以精确的连到一根线。在元件和终端的管脚末端都有连接点。一个圆点从中心出发有四个连接点，可以连四根线。

由于一般都希望能连接到现有的线上，ISIS也将线视作连续的连接点。此外，一个连接点意味着3根线汇于一点，ISIS提供了一个圆点，避免由于错漏点而引起的混乱。

(注：原文是Since it is common to wish to connect to existing wires, ISIS also treats wires as continuous connection points. Furthermore, as such a junction invariably means that 3 wires are meeting at a point it also places a dot for you. This completely avoids ambiguities that could otherwise arise from missing dots.)

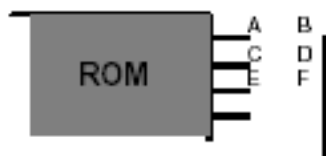
在此过程的任何一个阶段，你都可以按ESC来放弃画线。

### 线路自动路径器 (Wire Auto-Router)

线路自动路径器(WAR)为你省去了必须标明每根线具体路径的麻烦。该功能默认是打开的，但可通过两种途径方式略过该功能。如果你只是在两个连接点左击，WAR将选择一个合适的线径。但如果你点了一个连接点，然后点一个或几个非连接点的位置，ISIS将认为你在手工定线的路径，将会让你点击线的路径的每个角。路径是通过左击另一个连接点来完成的。WAR可通过使用工具菜单里的WAR命令来关闭。这功能在你想在两个连接点间直接定出对角线时是很有用的。

### 重复布线 (Wire Repeat)

假设你要连接一个8字节ROM数据总线到电路图主要数据总线，你已将ROM，总线和总线插入点如下放置。



首先左击A，然后左击B，在AB间画一根水平线。双击C，重复布线功能会被激活，自动在CD间布线。双击E、F，以下类同。

重复布线完全复制了上一根线的路径。如果上一根线已经是自动重复布线将仍旧自动复制该路径。另一方面，如果上一根线为手工布线，那么将精确复制用于新的线 (原文: then its exact route will be offset and used for the new wire)。

### 拖线 (Dragging Wires)

尽管线一般使用连接和拖的方法，但也有一些特殊方法可以使用。

如果你拖动线的一个角，那该角就随着鼠标指针移动。

如果你鼠标指向一个线段的中间或两端，就会出现一个角，然后可以拖动。注意：为了使后者能够工作，线所连的对象不能有标示，否则ISIS会认为你想拖该对象。

也可使用块移动命令来移动线段或线段组。

## 移动线段或线段组 (To move a wire segment or a group of segments)

- 1、在你想移动的线段周围拖出一个选择框。若该“框”为一个线段旁的一条线也是可以的。
- 2、左击“移动”图标（在工具箱里）。
- 3、如图标所示的相反方向垂直于线段移动“选择框”（tag-box）。
- 4、左击结束。

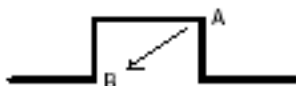
如果操作错误，可使Undo命令返回。



由于对象被移动后节点可能仍留在对象原来位置周围，ISIS提供一项技术来快速删除线中不需要的节点。

## 从线中移走节点 (To remove a kink from a wire)

- 1、选中 (Tag) 要处理的线。
- 2、用鼠标指向节点一角，按下左键。
- 3、拖动该角和自身重合（如下图）。
- 4、松开鼠标左键。ISIS将从线中移走该节点。



## 其他 (MISCELLANEOUS)

### 页边界 (The Sheet Border)

当创建新的一页时，无论是使用缺省的首页，还是用Design菜单中*New Sheet*命令，页面的大小总是由System菜单的*Set Sheet Sizes*的设置决定。页面的扩展部分用兰黑页面显示，但不会在实际的打印输出纸张上显示出来。

如果在输出时需要显示页边界，必须在页边框上设置制图框。

由于*Set Sheet Sizes*命令有两个不同用途，该命令还可进一步讨论。

要改变当前页的大小，只要激活该命令并选择所需的大小。

要重新定义页的大小，就要选择相应的数据输入区域并输入新的尺寸。如果改变了当前页的尺寸，就会马上影响当前页，但不会影响其他页。要影响其他页，就必须激活页面大小设置，并依次改变其他页的尺寸。.

页面大小设置对话框允许修改五种标准页面大小 (A4到A0) 和非标准，可以通过该对话框选择当前页的尺寸。每页尺寸(A4-A0及用户)都有两个编辑字段。左字段定义宽度，右字段定义高度。当ISIS打开一个页面，其尺寸不是标准尺寸时，就将该尺寸放入用户字段，并设置为该页的默认尺寸。

标准尺寸默认值对大多数打印机而言都是可以的，但要注意所有的打印机都有不能打印的空白处，如果你想打印的东西位于那些空白处是打不出来。打印机无法打印的空白处尺寸因不同的打印机而易。最好的方法是在你选择的页面尺寸周围画一个框，然后打印。如果框的一条边或多条边未能显示出来，那你的页面尺寸就扩展到了打印空白处，那你就应该将它缩减。

另外还有一点需注意：设计中的每页都有其实际尺寸在设计文档中。如果设计被载入不同尺寸配置的ISIS中，将不会有任何作用，除非使用页面尺寸设置命令。

## 标题部分 (The Header Block)

一般在每页都会有一个标题部分说明该页的具体内容如设计、标题、文件、版本和页码，及设计作者。为控制这些信息如何出现，标题部分被定义为一个符号库输入 (a symbol library entry)，称作HEADER。这通常是通过制图对象设置，标示，然后激活设置标志命令实现的。最重要的是，例如，如果需要的话，在显示或打印的时候目前的设计中带有@DTITLE 的文本对象自动替换为实际的设计标题。

这些关键字的完整清单如下：

**@STITLE** 设计标题从*Edit Design Property*窗体得到。不要与*Sheet*名相混。

**@DOCNO** 设计文件号从*Edit Design Property*窗体得到。

**@REV** 设计修改号从*Edit Design Property*窗体得到。

**@AUTHOR** 设计者从*Edit Design Property*窗体得到。

**@CDATE** 设计产生日期自动生成并附以固定的版式。

**@MDATE** 设计修改日期自动生成并附以固定的版式。

**@WS\_CDATE** 设计产生日期自动生成，格式根据Windows“短日期格式”（见下）

**@WL\_CDATE** 设计产生日期自动生成，格式根据Windows“长日期格式”（见下）

**@WS\_MDATE** 设计修改日期自动生成，格式根据Windows“短日期格式”（见下）

**@WL\_MDATE** 设计修改日期自动生成，格式根据Windows“长日期格式”（见下）

**@CTIME** 设计产生时间自动生成，格式根据Windows“时间格式”（见下）

**@MTIME** 设计修改时间自动生成，格式根据Windows“时间格式”（见下）

**@PAGENUM** 当前的*Sheet*的页数在设计中

**@PAGECOUNT** 总*Sheet*数在设计中

**@PAGE** 在设计中的*Sheet*页数表示为x/y，x是*Sheet*页数，y是设计中总的页数

**@FILENAME** 当前设计的文件名

**@PATHNAME** 完整的路径和当前设计文件的文件名

Windows长短日期格式和时间格式在Windows控制面板里使用地域设置或国际的applets

注意上面的关键词出现正文行前空2格处，这一行不应包括添加的内容。例如，不放置形式行：

```
AUTHORED BY @AUTHOR ON @WS_MDATE
```

为完成上面的，你需要在每一处放置四行。(AUTHORED BY, @AUTHOR, ON and @WS\_MDATE)

使用2D图形对象可以定义任何形式你想要的标题块。特别是你可以将你公司的标识语合并入标题。一旦定义，标题可以被放置在像任何图形符号的制图板上。

## 置于顶层/置于底层

有时，几个对象重叠（特别是图形），你很难指定你想要的。ISIS默认最近指定过的，但你能使用置于顶层和置于底层命令。命令在最近选中的对象上有效。

当进行新的设计时，你能使用这些命令编辑新的器件。例如，如果你画了一个新的运放符号，你可能需要把运放的主体（三角形）放在后面，以确保它的填充图案不会遮住输入端的‘+’和‘-’，当然，输入端的‘+’和‘-’要求在‘前面’。

## 自动标注器 (The Automatic Annotator)

ISIS能为设计的所有或部分成分自动选择相关的成分-这一步骤叫做自动标注。

可以使用*Tools*菜单的Global Annotator命令来使能该功能

注意：Global Annotator 不能标注不同类的多元素部分。这是因为存在几个非注解的接替卷和接触点，例如，没办法知道什么伴随什么（原文：Note that the Global Annotator cannot annotate heterogeneous multi-element parts. This is because with several un-annotated relay coils and contacts, for example, there is no way for it to know what goes with what.）。

## 数值标注

这一工具用于分析程序计算一些你想输入的标准电路的数值。下面是个例子：

VALUES

R1,10k

C1,100n

END

R1的数值将被调整到10K,C1至100n.

## 图形和文本格式

### 简介

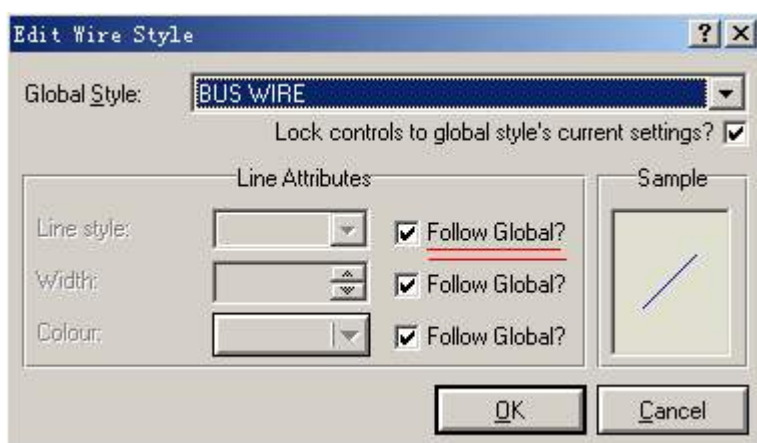
ISIS 有一个很完善的图形系统允许你自定义原理图所包含项目的外观比如线条格式,填充色,文本的字体,文本的效果等等... 这个系统非常强大并且允许你自己定义部分或者全部的原理图的全部外观,同时允许加载某些对象到你本地的外观属性.

在 ISIS 中所有的图形对象都是根据图形格式所画出. 图形格式(graphics style)是一个完整的描述,关于怎样去画出和填充一个图形(比如一条线条,一个方框,圆或别的)并且包含线条的格式(实心线,点线,虚线等等),宽度,颜色,填充格式,填充的前台色和背景色,等等..同样,所有的标号(label)和文字块(text blocks)在 ISIS(终端标号,管脚名,等等)都是根据文本格式所画出来.文本格式(text style)是一个完整的描述,关于怎样去画出一些文本和包含字体的属性(比如:亚洲字体,罗马字体,等等),字符的高度,宽度,颜色,等等...

在 ISIS 中,大多数的对象,例如 2D 图形,线条,终端标号,等等...每一个都有属于自己的格式以便他们能被定义,也就是说,比如,一条线条和另外一条线条有不同的外观. 这些项目作为这个对象的格式被设定,别的对象比如管脚名,子电路体,等等 是早已经被预先定义好的格式,因此这些对象只能定义要么全有要么全无的特性,也就是说,比如,子电路体可以有不同的你所想要的外观,但是所有的子电路体必须有相同的外观.

大多数的对象都有自己的格式当他们被放置后,都是由全局的格式最适合的设置的.比如,当你在布了一个终端,他的标号格式是自动由终端标号的格式设置,并且当你布了一条线条,他的图形格式是自动由线条格式设置. 2D 图形对象有些细微不同,对这些对象,右击这些对象可以选择适合的图形格式并且新放的图形对象是根据你所选择的格式来初始化的.

现在出现了聪明的器件,每一个本地格式都和通常设置他们全局格式保持联系.另外,每一个本地格式都有



一个 Follow Global 的选项对应它的属性 如图:

这个 Follow Global 选项当被选后,指示这个属性的属性应该是和全局的格式的设置是有关联的,当没有选择时,这个本地选择的格式属性就被使用了,在默认情况下,所有的新布的对象,所有的 Follow Global?选项都是被选的,这样新布的元件的外观完全是按照全局格式.

有本地和全局格式以及本地格式可以跟随一些或者全部全局属性的格式的好处是:

- 它允许你编辑所有的设计外观通过一个简单的全局格式的编辑-你不用分别编辑所有的元件属性。

- 它允许你定义库的符号可以自动的调配你所放置元件的外观
- 它允许你修改一些或者所有的部分在一个元件或者其他的元件

比如，假使你创建了一个元件并且放入了元件库，倘若你画了这个元件的图形在元件格式，然后当这个元件随后被调用到一个新的图中，他的格式会自动跟随这个图中元件的格式。

### 模板和模板菜单

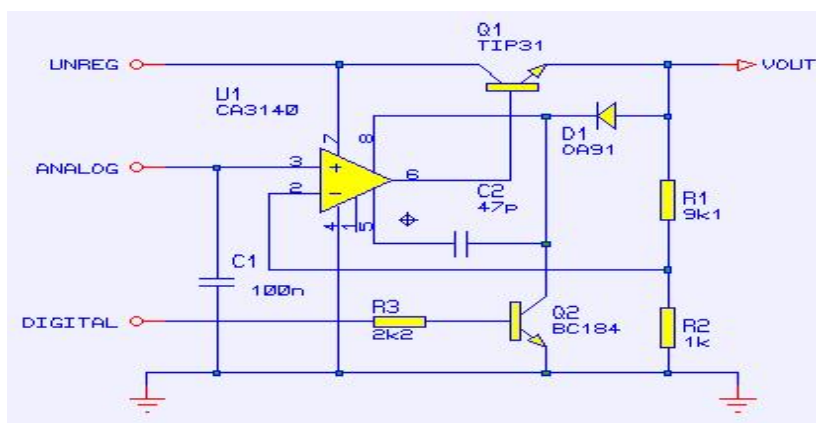
所有的控制电路图外观的信息，比如图形格式，文本格式，设计颜色，线条连接点大小和图形等 都是模板中的项目。所有的更改模板命令都是模板菜单的内容。

注意模板的改变仅仅只影响到当前运行的 ISIS，尽管这些模板有可能被保存并且在别的设计中调用。为了使下次你开始一个设计的时候这个改变依然有效，你必须用保存为默认模板命令去更新默认的模板，这个命令在模板菜单的下。

### 教程

#### 1. 概述

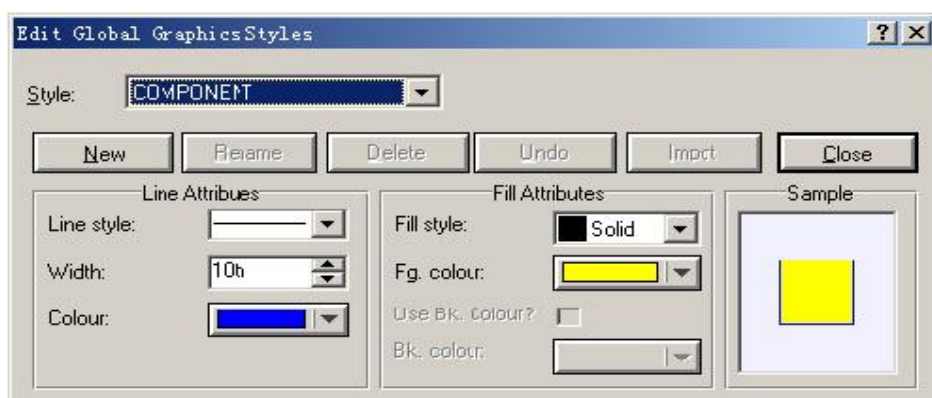
在开始图形和文本格式的教程前，载入电路图例子 STYLETUT.DSN 从 Samples 目录，象你所看见的，这个电路图的基本颜色是蓝色外框和填充为黄色的元件以及红色的终端，等并且如果你放大一些文本，你将会看见所有的都在 Labcenter Electronics 字体中。如图：



#### 2. 编辑全局格式

我们将要开始怎样编辑全局格式。这些格式就象它们名字所暗示的一样，在全局设计和编辑这些格式的时候容许全部改变这些电路的外观。

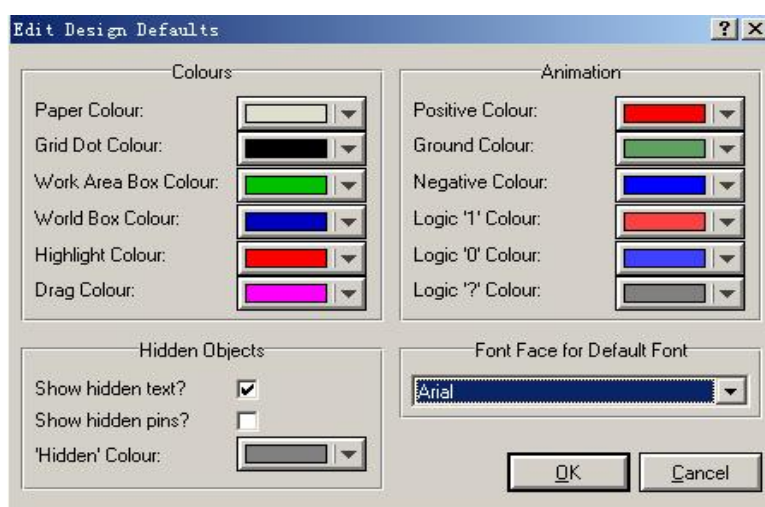
让我们开始改变一点这个电路的元件的颜色，这个设计中的所有元件都是从标准库中选取。这些元件的图形是在图形格式中设定好的。我们可以改变这个格式用 Set Graphics Styles 命令，这个命令在 Template 菜单下。选择这个命令然后你将会看见编辑全局图形格式的对话框显现出来。这个窗体可以访问所有在当前设计中的格式—下拉式的格式选择可以看见所有的选项，在格式下面的按钮暗示有可能创建，编辑并且删除新的格式除了编辑原来预定义的格式外。



确认 COMPONENT 格式在格式(Style)列表中被选中。你就会看到 COMPONENT 格式下面的 Line 属性和 Fill 属性，在 Line 属性中选择一个红色并且在 Fill 属性中选择 none 选项。改变完成后你将会看见右边的 Sample 对话框中会显示一个更新了的新的样式。

现在选择 TERMINAL 格式在格式(Style)列表中。改变后原先的 COMPONENT 格式会自动保存并且 Sample 对话框会显示出 TERMINAL 的样式。这时，在 Line 属性中选择一个蓝色并且在 Fill 属性中选择 Solid 选项,同时确信你选择了 Fg Colour 中的 yellow 格式。用关闭按钮关上窗体然后可以在电路图上显示出所做的改变。在白色的背景下新改变颜色格式的电路图可能看起来更好而且你也可能是打印在白纸上，让我们选择白色作为我们的纸张颜色，从 Template 菜单下选择 Set Design Defaults 命令。这个 Edit Design Defaults 的对话框容许你编辑在 ISIS 中所使用的大部分的颜色包括背景色和和纸张色。改变这个为浅灰色或者白色并且关闭窗体。

因此大家都看见了怎样去改变图形的格式。那文本的格式怎么改变呢？？改变文本格式和改变图形格式差不多除了你所要改变的格式属性不同外。然而，在我们改变每一个独立元件的文本格式前，让我们把文本的字体格式改变为别的，要改变默认文本字体，再次选中 Set Design Defaults 从 Template 菜单中，会显示一个 Edit Design Defaults 对话框。Font Face For Default Font 下你会看见选中的字体，如图：



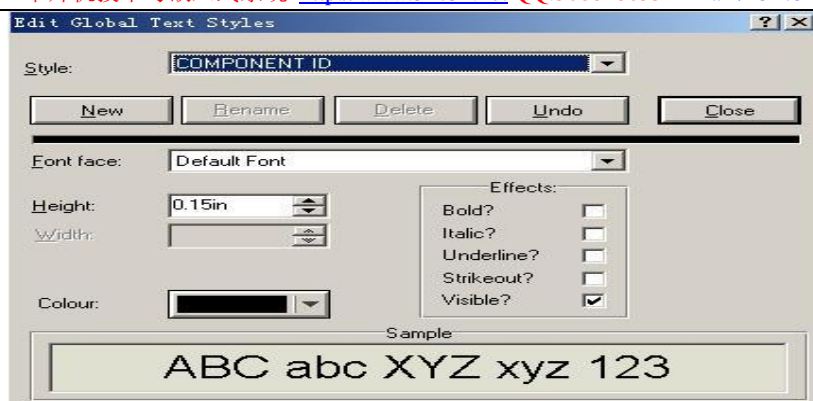
改变字体成标准的 windows 的 Times New Roman 字体然后关闭窗体。现在正在画的电路中的所有文本全部被改变为 Times New Roman 字体。

不论 ISIS 提供了什么类型的字体总是会有 2 个额外的字体在列表的最上面，就是默认的字体和 Vector Font 字体。默认字体不论你在 Edit Design Defaults 选中什么字体都会强制成默认字体，同时 Vector Font 是选中在 Labcenter Electronics 中内置的 Vector Font 字体，在 STYLETUT.DSN 这个电路图中当你改变了 Edit Design Defaults 中的字体，立刻就会在电路中显示所要的结果。

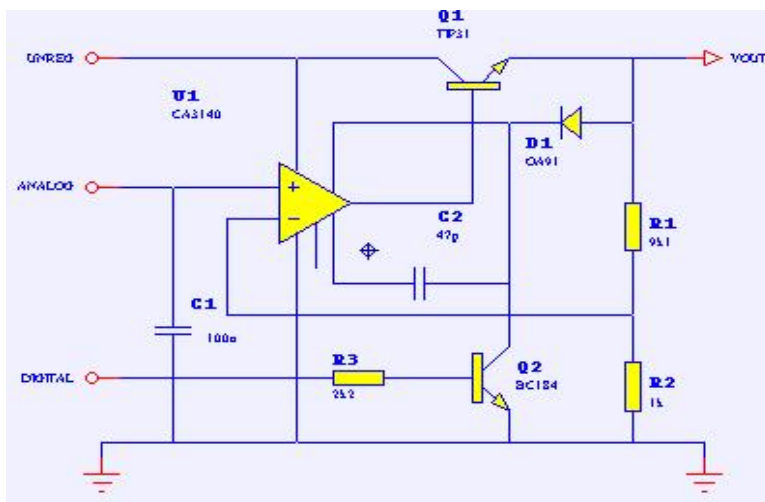
Vector Font 字体的用处就是当你需要输出一个备份到一个外设的时候，有可能 windows 不支持外设所用的字体，这时候就会调用 Vector Font 做为定义的字体，Vector Font 字体有个额外的优点就是保证在屏幕上所看到的和你输出到外设的大小一样。在 TrueType fonts 字体中是不容许的。

另外一方面来说，如果你使用一个彩色的外设（打印机），有可能不支持你所看到的 TrueType fonts 的字体，这样它就会用超过两种的字体混合看起来可能有点单调。

在我们把字体改变成 Times New Roman 后一些元件的文本格式字体会改变。然后从 Template 菜单中选择 Set Text Styles 命令，会显示出一个 Edit Global Text Styles 窗体，注意这个窗体和以前所看的窗体的相似处，如图：



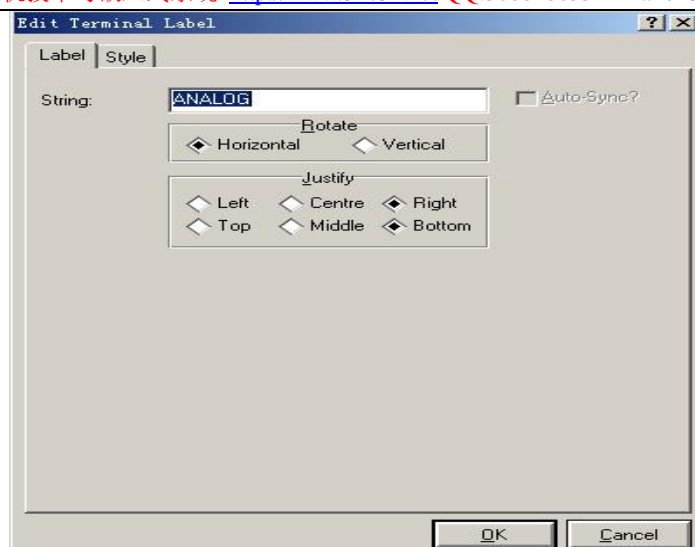
在默认状态下，COMPONENT ID 是 Style 所选中编辑的项目并且你会看见 Default Font 中的 Default Font 选项，就象我们所解释的一样，这个字体的选择锁死了 Edit Design Defaults 中的选择，当前的字体是 Times New Roman.，因为这样，此对话框中的地方使用了 True Type font 字体，比如字符的宽度 (Width)，现在是灰色的，选择 Font face 为 Courier New 并且选择 Bold?选项在 Effects 属性下。现在选择 PIN NUMBER 格式在格式列表中，(先前在 COMPONENT 中所做的修改会自动保存) 把在 Effects 属性下的 Visible?选项不选择，然后关闭对话框。你将会看见所画好的电路图的元件的 ID(component ID)现在变成 Courier New 字体并且 OP-AMP 的 pin number 不见了。如图：



### 编辑本地格式

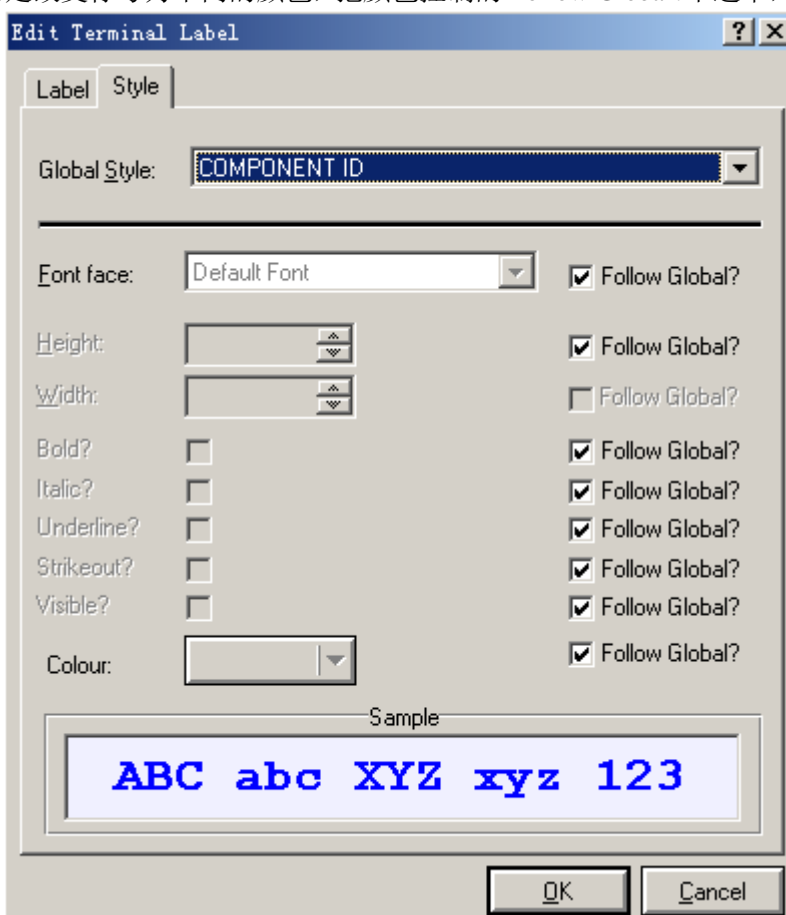
以上我们所讲的只能修改全局的格式，这些改变能通过电路图立刻反映出来，现在我们去查看怎样去改变一个本地格式，编辑 ANALOG 终端的标号。

让我们查看本地格式：先右键单击 ANALOG 这个标号，然后左键单击，一个 Edit Terminal Label 对话框将会显示出来。如图：



最后, 选择 Style 查看 Style 的设置。你将会看到如下: 一个全局 Style 选择框在顶部, 一个 Style 的属性设置 (Font face, Width, 等等) 在左边, 右边每一项都有 Follow Global? 属性被选中。所有的字体和图形都是这个形式。当 Follow Global? 是被选中的时候暗示这个格式是和全局格式是有联系的。未选中的话, 容许你自己定义一些格式, 这样, 在 TrueType font 字体中 Width 是没有意义的。如图:

我们所要做的是改变标号为不同的颜色, 把颜色控制的 Follow Global? 不选中, 你将会看见颜色控制可以使用了,



并且显示出当前所用的颜色。改变色彩为暗蓝色, 然后关闭这个对话框, 然后看见 ANALOG 终端的标号变为暗蓝色。为了验证这个标号的颜色是真的改变并且与全局格式无关, 从 Template 菜单中选择 Set Text Styles 命令, 改变 COMPONENT ID 的颜色为紫红色 (magenta), 选中 Italic 选项然后关闭窗口体。这个 ANALOG 终端的标号依然没有跟随全局格式改变而改变, 所有的其他元件都变了颜色和成为斜体了。



最后，我们会演示怎样去修改一个本地的图形格式。选中 **Graphic Mode icon** 按钮和 **Box icon** 按钮。鼠标在电路图中单击左键然后拖动画出一个方框。我们需要把这个本地的方框的格式改成透明的。右键点击方框的线条然后左键再点击上面，一个 **Edit Box's Graphic Style** 对话框将会出现。它和前面所见过的 **Edit Terminal Label** 对话框的很相似，只有个别的属性不同。如图：



操作和以前的一样，不选中 **Fill Style** 的 **Follow Global?**选项，**Fill Style** 就会显示出来，改变成 **none** 选项，然后按下 **This Graphic Only** 按钮。

这是一个简单的例子，这个功能主要的作用是在 **2D graphic** 中（**lines, boxes**,等等）当你创建了一个新的库，建议你避免去设置本地格式除非很有必要。当你做完上面的工作后，你所修改的元件的外观和你所编辑的一样，当应用到一个库文件中去的时候，有些元件可能不会混淆。在正确使用来说，通常要注意的是，比如：一个晶体管的是实心的填充色，如果更改的话，会导致在不同色彩的电路图中或者使用者中不能正确的显示元件。

### 利用模板工作

我们最终提到了模板的对象。你所看见的一切信息包括文本格式，图形格式，连接点的外型和大小，纸张颜色等等，事实上所有的改变都可以通过模板菜单来下命令，被称为设计模板（**design template**）。当安装 **ISIS** 的时候有个默认的模板 **DEFAULT.DTF** 被安装到 **proteus** 的库文件下，当 **ISIS** 被启动或者你在 **File** 菜单下选择了 **New** 命令，**ISIS** 就会从这个文件中调用默认的模板，当你在 **Template** 菜单下选择 **Save Default Template** 命令就会把所做的任何修改保存到文件中。这个文件是没有备份的，因此不要随意使用 **Save Default Template** 命令，除非你确信这个模板在将来你也会用到他。

为了保持电路图的可持续性，**ISIS** 保存了一份当前的模板拷贝，当这个电路图被调用的时候就会恢复模板设置。假设你早已经在前面的教程中载入了一个电路图，如果你后来有载入了一些别的电路图，你会看见在 **ISIS** 中它们有不同的格式。因此很高兴的是如果你设置了一个格式并且用了 **Save Default Template** 命令设置成默认的模板，如果你决定改变这个格式，最简单的方法是从 **Template** 菜单下选择 **Apply Template From Design** 或者 **Apply Default Template**，这两个命令都可以调入一个新的模板到当前的设计中，不管是从别的电路图中还是默认模板中得到的模板都不能改变已经改变成本地格式的设置。在这种方式下，你可以设置一个新的默认的模板并且把它调用到你所有以前的设计中，相反，如果你丢失了你的默认模板，你可以从你以前的设计中恢复过来然后用 **Save Default Template** 保存成默认模板。

### 第三章 属性

#### 1. 简介

ISIS 中的属性有非常大的用处。一个特定的对象的属性是由一些关键字组成，比如，在 ISIS 中，我们使用封装的属性与 PCB 的封装关联。

对象，管脚，电路图都有自己的属性，如果你想很把这个功能强大的软件用好你必须很清楚他们之间的联系，这个软件和以前你所用过的画电路的软件有些不同。

#### 2. 对象属性

对象属性有两个类型—系统属性和使用属性。在 ISIS 中的这些功能是由一些保留字所组成，不管是内部的程序比如 ARES 和 VSM，或者你自己所使用的软件都是有关联的。

##### (1) 系统属性

系统属性在 ISIS 中是由一些特殊含义的关键字所构成。比如，一个元件中的 DEVICE 属性是根据元件库在分配时候定义的。这些属性一般是文本的—比如元件的 REF 和 VALUE 属性可以直接从 Edit Component 对话框中访问，但是别的比如 DEVICE 属性就是做为图形操作所生成的结果。

通常上，如果你关心的只是希望去读取它们的数值 (values) 通过 search and tag 命令，或者用 Assignment Tool 去修改它们的属性。比如，你可能希望选中在这个设计中的所有的 7400 元件。这个需要你知这个你所要选中的元件的系统属性。

每一个对象的系统属性的细节部分都在 OBJECT SPECIFICS 中给出。

##### (2) 使用属性

元件，子电路和 VSM 的器件可以加载除了本身的标准属性外没有限制的额外的属性，这些使用属性是由一个文本块构成一个属性块，它包含很多，比如：

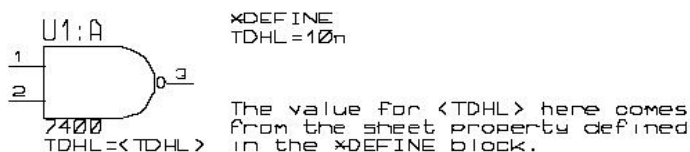
SUPPLIER=XYZ Electronics

你可以直接编辑属性块通过对象的对话框，和用 Property Assignment Tool 生成的一样好。

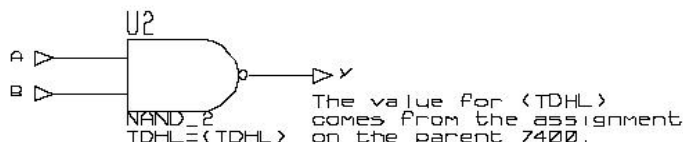
##### 为了编辑一个对象的属性

1. 选中这个所要编辑的元件并且点击左键会出现对象的对话框。
2. 如果这个对象有它自己的使用属性，这个对话框的将会有有一个文本的编辑框标号为 Properties,鼠标移到已经存在的文本的下面并且按左键。
3. 如果需要的话编辑这个文本。每一个属性都有一些关键词和数值构成，中间用等号分开如图：

#### PARENT SHEET



#### CHILD SHEET



#### DESIGN PROPERTIES

使用属性的关键字应该是简单的由一个数值，一些单词，数字和下划线组成。一般还包括空格，逗号，双引号或者等号 (,"=)，为了和 ISIS 常用的方式一致，这些属性可以用大括号('{ '和 '})'来关闭，使它们不能在屏幕上显示。比如，敲入：

```
{PRIMITIVE=DIGITAL}
```

用来定义这个对象的所需要的仿真模式是数字方式，但是这个文本不会显示。当然了，有的人只想让数值显示，这样的话你可以这样做：

```
{MODFILE=}OPAMP
```

理论上来说你可以把大括号放在任何地方。但是，当使用 **Property Assignment Tool** 修改属性块的时候，它是假设你所使用的，包括大括号中不被显示的同时修改。如果你把大括号放到别的地方，这样你可能得不到你想要的结果。

## (2) 属性的定义(PROPDEFS)

它可以提供一个详细的解释关于器件的使用属性。比如，通常使用的元件属性是 **PACKAGE** 和 **MODFILE**。创建一个元件库的时候要给出一个合适的元件定义，这些属性将会在它们自己的 **Edit Component** 对话框中显示出来。这些属性定义包括属性的数据类型描述（比如整数，浮点数或者字符）和在适合范围的数值。默认的数值一般都已经定义过。

这个设计使查看一个属性在特殊模式下是否合法和查看这些属性的意思变的很容易。它也可以支持不同的封装类型和在不同的模式下的仿真方式。

对于一些未知特殊的元件的属性也可以用前面介绍的方法写成本属性格式，这个设计向前兼容以前的电路图（proteus 老版本的）或者那些没有属性定义的库元件。

## 3. 电路(sheet) 属性

### 简介

每一个电路图的属性都设定了一些属性的赋值。这些可以考虑做为常量定义（数值或者文本式）在对象属性的赋值中是很有用的。对它们自身来说可能不是非常有用，但是它们真正的作用是在对象属性表达式中。比如，如果一个电路的属性定义成如下：

```
*DEFINE
```

```
PI=3.142
```

接着你可以用下面的表达式定义一个电阻：

```
VAL=EVAL(500/PI)
```

在生成网络报表的时候，这个语句引起网络报表编译程序估算一个数值，并且这个电阻数值将会显示在网络报表中或者材料清单中，数值为 159.134

进一步的信息关于属性表达式请看属性的表达式这章。

### 定义电路 (sheet) 属性

电路的属性可以被定义成如下的形式：

- \* 直接的方式，使用一个默认的 **DEFINE script block**，你可以使用它去定义表达式中所要用的常量，就象上面的例子一样。

- \* 作为一个参数的映射用 **MAP ON script block**，这样，利用 **MAP ON script** 的参数从一个父电路图中获得所要的数据值，这是非常有用的在创建通用的仿真模式的时候，每一个元件在仿真方式上除了在元件属性的不一样外都使用相同的电路。

进一步的论述在 **VSM** 的手册中给出

- \*从父对象中继承。换句话说，如果父对象有如下的赋值：

```
R3=10k
```

然后子元件会自动获得这个属性。这个为参数电路提供了基础，在参数电路方式下通过继承来的不同的元件值构成图表形式。

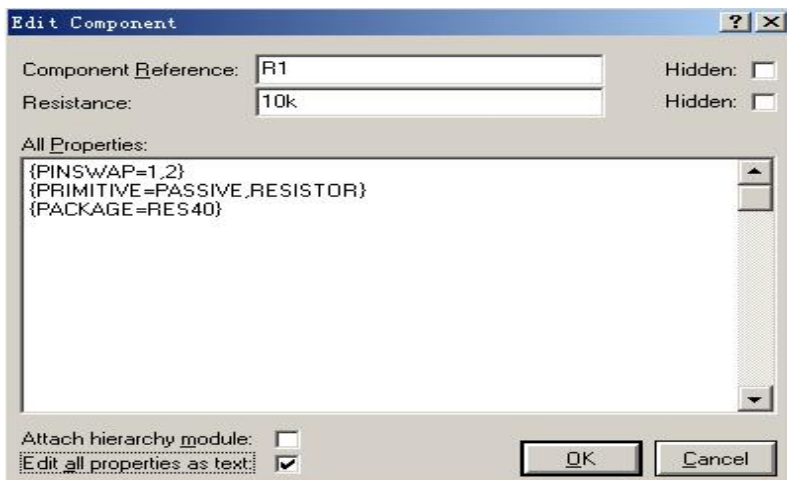
如果一个属性用 **DEFINE** 或者 **MAP ON block** 在子电路中定义并且在父电路中也定义了，从父电路来的属性就会传给子电路。在任何需要的地方都可以用这个方法提供默认的数值。

### 电路属性规则和范围

重要的一点是要清楚电路属性只是一个参考在它们所定义的地方。特别是，在父电路中的电路属性不能被它的子电路访问除了通过父电路的电路属性块。如果需要用这种方式去访问一个属性，你可以加入下面一行：

```
TDHL=<TDHL>
```

对父元件来说，如果 TDHL 被定义成一个电路属性在父电路中，它也将变成父元件的属性，并且可以对子电路的属性定义，那儿会有更进一步的属性表达式。这种排列与编写 C 程序有些相似。



这个给出设计中的属性是根据所有的电路属性累积后所决定的。因此有的电路没有父电路，这样仅仅只要用 DEFINE script blocks 来定义设计的属性。

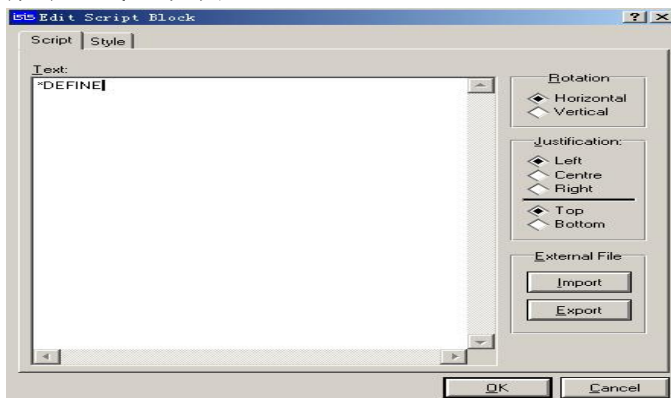
对于网络报表 SDF 格式的输出，当任何应用程序解读的时候，这个设计属性就会显示出来，在 proteus 的 VSM 方式下，设计属性一般定义成仿真模式比如 the number of steps, the operating temperature 等等。这些都在 VSM 的手册中给出。

为了创建一个设计属性的列表：

1. 从 Tools 菜单下选择 Goto Sheet 命令返回你设计的主电路。
2. 选择 Main Mode 和 Script 按钮
3. 在你想要定义的地方按左键
4. 在开头键入下面行：

\*DEFINE

5. 键入所要的属性表达式 如图：



注意设计属性对于电路图来说也是电路属性。然而，对电路属性的标准规则依然适用，设计属性对于别的电路来说是私有的。

## 参数电路 简介

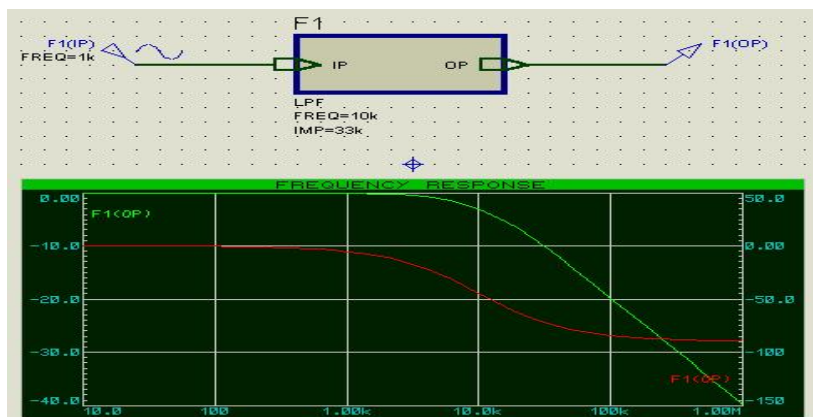
ISIS 有一个独特的和非常强大的特性可以组合电路属性，元件属性和继承属性构成参数电路。一个参数电路里面的元件值可以根据公式给出胜于用常量。自然，公式要包含变量或者参数和从构成这个参数电路的电路属性已经定义好的数值。因此，从同一个电路中继承来的不同参数，在参数电路中有不同的元件的值

更多的信息关于继承设计看 HIERARCHICAL DESIGNS，如果你不清楚继承设计是什么，我们建议你跳过

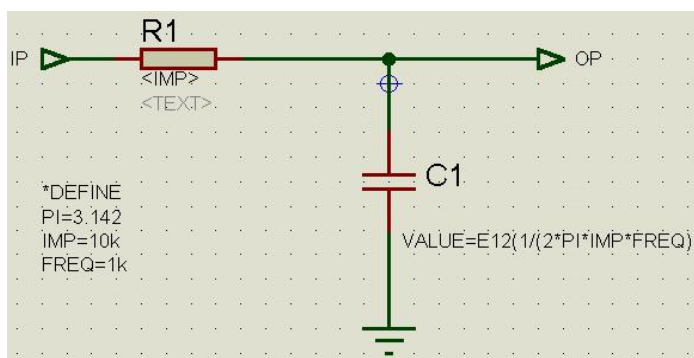
这章。

一个例子：

在 SAMPLE 目录下你可以发现一个例子设计 LPF.DSN,这可以很好的解释参数电路。主电路的如下：



它包含一个简单的子电路，这个子电路有两个使用属性定义了频率和这个滤波器的阻抗。这个子电路的真实的电路是：



这有几点要注意一下：

\*这个 DEFINE 块定义了电路属性：PI.这是为电容属性的表达式服务的常量

\* 电阻的 value field 中包含字符<IMP>, 这个带尖括号的语句使网络编译器用给出的父电路 IMP 的属性替代这个<IMP>, 这种情况下为 33k,这没有执行表达式的估算—这个替换纯粹是文字上的替换。

\*这个电容有一个使用属性的表达式。这个 E12 函数要求表达式不仅用网络编译器估算，并且要四舍五入取到最接近 E12 的值。其它的选项是 EVAL(没有四舍五入)和 E24(四舍五入到 E24 的值)。

PI,IMP 和 FREQ 是所有的电路属性。PI 的值从 DEFINE 块得来同时 IMP 和 FREQ 从父电路得来

如果你生成材料清单，你将会看见如下：

QTY	PART-REFS	VALUE
-----	-----------	-------

-----

Resistors

-----

1	R1	33k
---	----	-----

Capacitors

-----

1	C1	470p
---	----	------

ISIS 计算  $1/(2*3.142*33000*10000)$  近似到 0.00000000482 并且四舍五入到接近 E12 的值-470p

上面的例子中实际上有两个截然不同的过程—属性替代和属性表达式的求值。它们都有自己的优点和缺点，在下面的部分会详细论述

### 属性替代

这个功能特点就是机械的操作，当用来对电阻的赋值和网络编译器碰到用尖括号括起来的关键字的时候会替换掉原来的值。如果关键字是一个电路的属性，会用数值取代要替换的地方。如果没有电路属性存在，这样网络编译器就会产生一个警告并且把这个属性从对象中去除。

在两个方面属性的替代很有用：

\*你可以在参数电路中使用而不用给出参数的数值。对于上面的例子的元件的封装---可以很方便的计算出这个电容所使用的电容值为 470pF，但是你在 PCB 设计中仍然需要封装它，你可以把下面的话附在电容的使用属性中：

PACKAGE=<C1\_PACKAGE>

然后你加上这个属性到子电路：

C1\_PACKAGE=CAP10

当网络编译的时候，这个 C1 将会有显示出这个属性

PACKAGE=CAP10

属性的表达式计算不能用在这个地方，因为 CAP10 不可能算为一个数值。

\*属性替代另外一个主要的作用是利用 proteus 的 VSM 设置扫描分析。这种情况下你想仿真器去计算这个表达式。在这里通过属性替代建立的元件属性比属性的表达式计算适合。更进一步的论述在 VSM 手册中给出。

### 属性表达式计算

与属性替代截然相反的是属性的表达式计算。通过公式计算出一个合适的数值并且用这个值取代原来的表达式。另外 ISIS 中的最终结果近似成 E12 或者 E24 的数值。

这儿有三种形式的语法：

EVAL (...)

E12 (...)

E24 (...)

在所有的情况下，这些参数包含在一个数学表达式中，表达式还含有加，减，乘除的操作和常量或者属性名。乘除的优先级别高但是在有括号的地方，括号中的最先执行。

一些表达式的例子如下：

EVAL(1/(A+B))	A 和 B 都是电路属性
E12(20k+2*F*PI)	20K 自动转换成 20000
E24(3+4*5)	估算为 24

尽管在某些方面比属性替换功能强大，但是还是有些限制：

- \* 等式仅仅由数值构成,加入字符是不容许的。
- \* 电路属性中的公式只是一个参考—你不能用它来访问别的电路属性或者别的元件值。
- \* 这个不支持数学函数（比如：sin,cos,square-root）等等。

我们会在将来的版本中弥补这些不足。

### 四舍五入的函数 E12(),E24()

这个属性表达式支持把数值结果近似成 E12 或者 E24 的值。这个功能防止参数电路由于没有可用值或者浮点数而终止操作。

你需要注意：

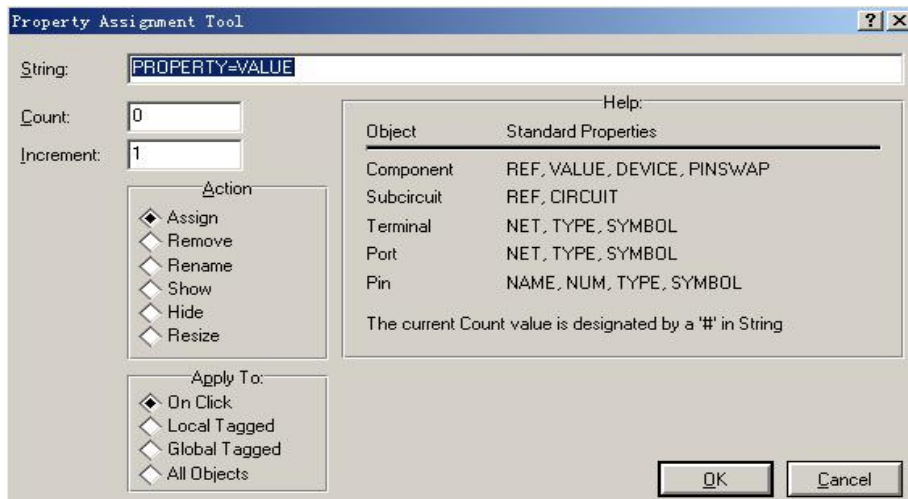
- \* 数值为 0 或者负值的时候这个近似功能禁止。在任何情况下不可能有负的电阻和电容，所以这个不是问题。
- \* 这些函数用几何基础比算术基础好，因此在 3k3 和 4k7 之间的取值近似为 3.94。
- \* 如果在一个参数电路中包含不少的近似值，运算的时候是难以忍受的。对于滤波器的设计或者别的，你最好手动计算这些数值然后使用参数替代在你的电路中取代。

当然，如果你有 proteus VSM,你可以运行仿真程序和查看不同的数值对电路的影响。

### PAT 的对话框

PAT 中含有比较复杂的对话框，其中有如下的内容：

- String 对于要改变对象的属性的赋值，或者属性的关键字。
- Count 计数的初始值。每次执行 PAT 都会自动增加。这个当前的数值可以用一个 # 符号代入前面的 String 定义中。
- Action 这个 Action 就是你想要执行的操作。在 PAT Action 中有不同的 Action。
- Apply 这个应用模式就是你想要 PAT 所做的操作。在 PAT Apply 中有不同的 Apply。



如图：

### PAT Actions

这个 Property Assignment Tool 可以执行如下的功能：

赋值 赋值属性包含如下形式：

**keyword=value**

并且这些属性会被赋值到所选择的对象。

如果你想连续的赋值，比如 D0,D1,D2 等等。可以使用('#')字符代替这个值并且设置初始值。

使用属性和系统属性都有可能要赋值，对系统属性赋值可以改变你所画的图形。

去除 (REMOVE) 这个地方必须有一个属性的关键字并且这个属性将会从所选的对象中去除。只有使用属性才可以被去除。

改名 (RENAME) 这个地方包含如下的形式的字符：

**current\_keyword=new\_keyword**

左边的字符是你想要改名的属性，右边的是你要改成的名字。

只有使用属性才可以改名

显示 (SHOW) 这个地方必须要包含属性的关键字并且这个属性是所选择对象要显示的。

所有的使用属性和系统属性都可以显示

隐藏 (HIDE) 这个地方必须要包含属性的关键字并且这个属性是所选择对象要隐藏的。

所有的使用属性和隐藏属性都可以被隐藏。

重定义尺寸 ( RESIZE) 这个地方包含如下形式：

**REF=20,16**

这样所选择的元件的尺寸属性将会被赋值成一个新的高和宽。只有文本的系统属性可以缩放。

### PAT 应用模式

这个 PAT action 选择后有下列方法应用到电路中。

点击 (ON CLICK) 一旦按可了 ok 确认后，这个对话框消失同时返回 Main Mode，并且 Instant Edit 按键自动使用，当你用鼠标点在所要执行的对象上就会改变

当你选择不同的按键后，这个 PAT 功能就被取消掉了

这是唯一的方法应用 PAT 到连线的标号的赋值上。

LOCAL TAGGED 这个选项是执行所有在当前电路中选中的对象。你可以单独选择或者用 Search & Tag 命令选择对象。

GLOBAL TAGGED 这个选项是执行所有的在这个电路中选中的对象。你可以单独选择或者用 Search & Tag 命令选择对象。

ALL OBJECTS 在这个电路中所有的对象都会执行这个操作。

### The Search and Tag 命令

这个 search and tag 命令对于选择大量特殊的对象很有用，尤其是在用 PAT 的 Local Tagged or Global Tagged 功能的时候可以连续操作。

这三个搜索命令

SEARCH 这是一个常用的搜索操作在你开始一个新的搜索任务的时候。设置所要搜寻对象的条件和清除原先选中的对象

AND SEARCH 这个操作可以消除当前已经选中对象的。它除去所有选中不符合条件的对象只留下符合条件的。

OR SEARCH 这个操作是用来加入对象到所选中对象中。它把所有符合条件都设置为选中状态。

### 例子

这个 Property Assignment Tool 和 Search & Tag 命令可以提供非常强大和自由的功能，当用来巧妙的处理对象的属性时候。然而，在开始你可能有些不太会用，所以我们用这个例子一步步教你怎样去使用。

**去标注一个总线的分接头。**

1. 键入“A”键调用 PAT
2. 设置 string 中内容为 NET=D#然后按 OK,这个操作默认为赋值并且模式是点击式的。
3. 在你想要标注的连线上点击。可以用鼠标点也可以用回车键点击。这个些线条将会被标注成 D0,D1,D2 等等。

**对电路中的所有 BC108s 加入封装。**

1. 键入“T”键调用 Search and Tag 命令。
2. 设置 property 为 Value 和 string 为 BC108.然后按 OK。默认模式是等于。所有元件的 value 为 BC108 的全部会被选中。
3. 键入‘T’调用 PAT
4. 设置这个 string 为 PACKAGE=TO18 然后按 OK,这个操作默认为赋值和默认模式为 Global Tagged，所有选中的 BC108s 将会得到一个封装属性。

**把所有的 ITEM 属性改名为 CODE 属性。**

1. 键入‘A’键调用 PAT
2. 设置 string 为 ITEM=CODE，操作设为 Rename 和模式改为 All Objects，然后按 OK,所有的对象的属性都会把 ITEM=value 替换为 CODE=value。

**隐藏所有的封装属性。**

1. 键入‘A’调用 PAT.
2. 设置 string 为 PACKAGE,这个操作设为 Hide 和模式为 All Objects，然后按 OK,所有封装属性都会隐藏。

**改变元件的 references**

1. 键入‘A’调用 PAT
2. 设置 string 为 REF=10.8，这个操作设为 Resize 和模式为 All Objects，然后按 Ok，所有的元件 references 都会收缩成新大小。

**对 1000uF 的大电容赋封装**

1. 键入‘T’调用 Search & Tag 命令。
2. 设置 property 为 DEVICE 和 string 为 CAP ELEC。然后按 OK,这将会选择所有的电解电容。
3. 从 Tools 菜单中调用 AND Search 命令。
4. 设置 property 为 VALUE，string 为 1000u，选择模式为 Begins，然后按 OK，这将会选中所有 values 为 1000u 或者 1000uF 的电容。



5. 键入'A'调用 PAT

6. 设置 string 为 PACKAGE=ELEC-RAD30 然后按 OK,这个操作使用默认的赋值和模式为 Global Tagged,所有选中的电容将会换成新封装。

### 属性的定义

创建属性的定义

属性的定义完全是使用了 Make Device 的对话框中的属性表。更详细的信息是使用对每个项目的帮助

默认的属性定义

大部分的属性在你所创建的大多数元件都是可以使用的。比如,如何元件转换成 PCB 格式都要 PACKAGE 属性,任何元件要能仿真就需要 MODFILE, MODEL 和 SPICEMODEL 属性。

你可能想要应用你自己定义的属性比如 STOCKCODE, SUPPLIER 或者 COST 在你所创建的元件中。这是很容易的,在 System 菜单下使用 Set Property Definitions 命令可以对默认的属性定义。这些属性定义是在 Make Device 对话框中会显示的属性。

这些信息利用这个命令巧妙的用 proteus 系统目录下的 PROPDEFS.INI 文件保存着。

本站将提供更详细的说明资料。整理中... ..

## 实例教程