

LabVIEW 入门讲义

背景介绍

LabVIEW 是实验室虚拟仪器集成环境 (Laboratory Virtual Instrument Engineering Workbench) 的简称, 是美国国家仪器公司 (NATIONAL INSTRUMENTS, 简称 NI) 的创新软件产品, 也是目前应用最广、发展最快、功能最强的图形化软件集成开发环境。目前它广泛地被工业界、学术界和研究实验室所接受, 视为一个标准的数据采集和仪器控制软件。

LabVIEW 集成了与满足 GPIB、VXI、RS-232 和 RS-485 协议的硬件及数据采集卡通讯的全部功能。它还内置了便于应用 TCP/IP、ActiveX 等软件标准的库函数。这是一个功能强大且灵活的软件。利用它可以方便地建立自己的虚拟仪器, 其图形化的界面使得编程及使用过程都生动有趣。图形化的程序语言, 又称为“G”语言。其编写的程序称为虚拟仪器 VI (Virtual Instrument), 以 .VI 后缀。使用这种语言编程时, 基本上不写程序代码, 取而代之的是流程图或流程图。它尽可能利用了技术人员、科学家、工程师所熟悉的术语、图标和概念, 因此, LabVIEW 是一个面向最终用户的工具。它可以增强你构建自己的科学和工程系统的能力, 提供了实现仪器编程和数据采集系统的便捷途径。使用它进行原理研究、设计、测试并实现仪器系统时, 可以大大提高工作效率。

以下的课程内容可以在 wiki 上找到:

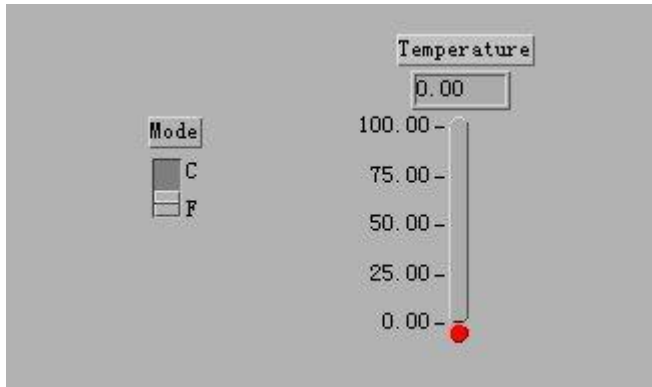
<http://phylab.fudan.edu.cn/doku.php?id=home:whyx:labview>

第一课 LabVIEW 概述

第一节 虚拟仪器 (VI) 的概念

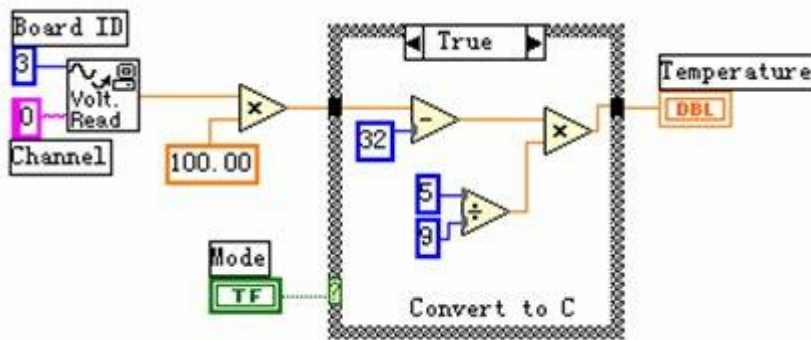
使用 LabVIEW 开发平台编制的程序称为虚拟仪器程序, 简称为 VI。VI 包括三个部分: 程序前面板、框图程序和图标/连接器。

程序前面板用于设置输入数值和观察输出量, 用于模拟真实仪表的前面板。在程序前面板上, 输入量被称为控制 (Controls), 输出量被称为显示 (Indicators)。控制和显示是以各种图标形式出现在前面板上, 如旋钮、开关、按钮、图表、图形等, 这使这得前面板直观易懂。下面是一个温度计程序

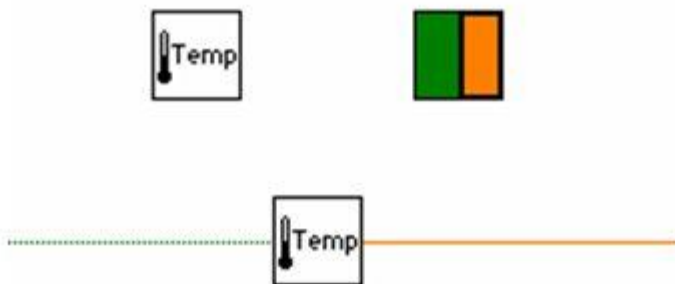


(Thermometer VI) 的前面板。

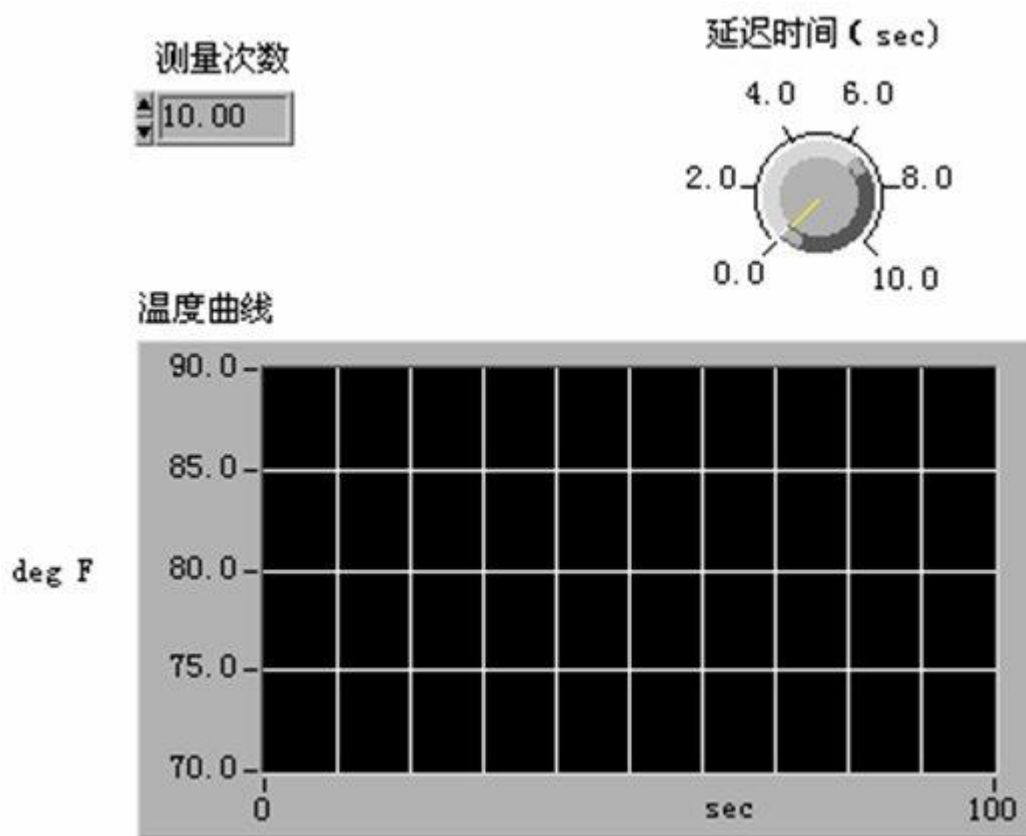
每一个程序前面板都对应着一段框图程序。框图程序用 LabVIEW 图形编程语言编写，可以把它理解成传统程序的源代码。框图程序由端口、节点、图框和连线构成。其中端口被用来同程序前面板的控制和显示传递数据，节点被用来实现函数和功能调用，图框被用来实现结构化程序控制命令，而连线代表程序执行过程中的数据流，定义了框图内的数据流动方向。上述温度计程序 (Thermometer VI) 的框图程序如下：



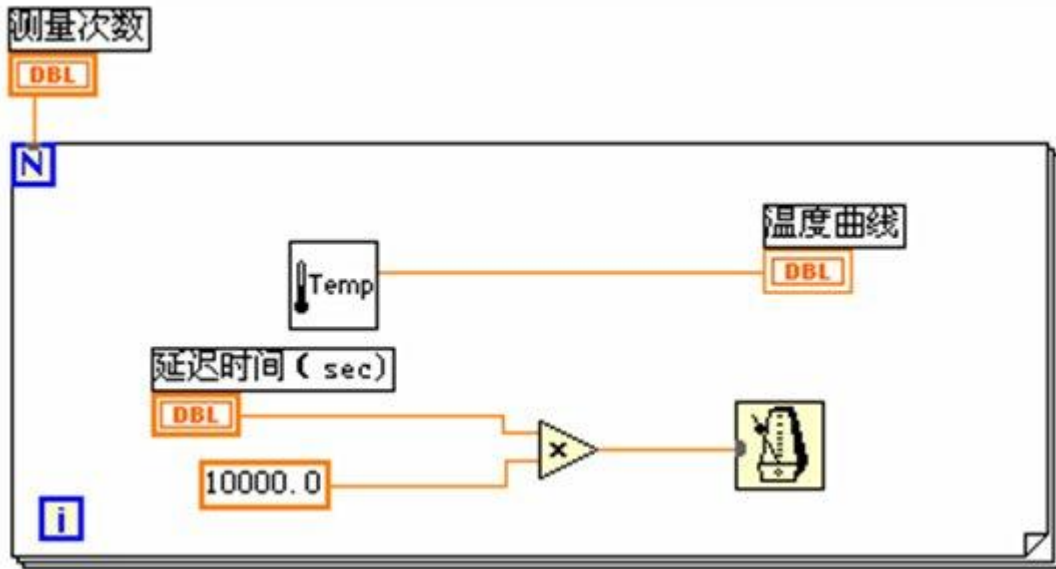
图标/连接器是子 VI 被其它 VI 调用的接口。图标是子 VI 在其他程序框图中被调用的节点表现形式；而连接器则表示节点数据的输入/输出，就象函数的参数。用户必须指定连接器端口与前面板的控制和显示一一对应。下面图示为温度计程序(Thermometer VI)的图标和连接器。连接器一般情况下隐含不显示，除非用户选择打开观察它。



LabVIEW 的强大功能归因于它的层次化结构，用户可以把创建的 VI 程序当作子程序调用，以创建更复杂的程序，而这种调用的层次是没有限制的。举一个例子，我们把前面创建的温度计程序(Thermometer VI)作为一个子程序用在当前新建程序里，当前程序的前面板如下图所示，先前的温度计子程序用于采集数据，而当前的程序用于显示温度曲线，并在前面板上设定测量次数和每次测量间隔的延时。



当前程序的框图如下图所示，它把温度计子程序放置在一个 FOR 循环里，每次循环过程采集一次测量结果，当循环执行了设定的次数后，程序把采集的数据送到前面板的图表上显示。



LabVIEW这种创建和调用子程序的方法，使创建的程序结构模块化，更易于调试、理解和维护。

第二节 LabVIEW 的操作模板

LabVIEW 具有多个图形化的操作模板，用于创建和运行程序。这些操作模板可以随意在屏幕上移动，并可以放置在屏幕的任意位置。操纵模板共有三类，为工具（Tools）模板、控制（Controls）模板和功能（Functions）模板。

工具模板（Tools Palette）

工具模板为编程者提供了各种用于创建、修改和调试 VI 程序的工具。如果该模板没有出现，则可以在 Windows 菜单下选择 Show Tools Palette 命令以显示该模板。当从模板内选择了任一种工具后，鼠标箭头就会变成该工具相应的形状。当从 Windows 菜单下选择了 Show Help Window 功能后，把工具模板内选定的任一种工具光标放在框图程序的子程序（Sub VI）或图标上，就会显示相应的帮助信息。工具图标有如下几种：



由左而右由上而下依次为：

1. 操作工具：使用该工具来操作前面板的控制和显示。使用它向数字或字符串控制中键入值时，工具会变成标签工具的形状。
2. 选择工具：用于选择、移动或改变对象的大小。当它用于改变对象的连框大小时，会变成相应形状。
3. 标签工具：用于输入标签文本或者创建自由标签。当创建自由标签时它会变成相应形状。
4. 连线工具：用于在框图程序上连接对象。如果联机帮助的窗口被打开时，把该工具放在任一条连线上，就会显示相应的数据类型。
5. 对象弹出菜单工具：用左鼠标键可以弹出对象的弹出式菜单。
6. 漫游工具：使用该工具就可以不需要使用滚动条而在窗口中漫游。
7. 断点工具：使用该工具在 VI 的框图对象上设置断点。
8. 探针工具：可以在框图程序内的数据流线上设置探针。程序调试员可以通过控针窗口来观察该数据流线上的数据变化状况。
9. 颜色提取工具：使用该工具来提取颜色用于编辑其他的对象。
10. 颜色工具：用来给对象定义颜色。它也显示出对象的前景色和背景色。

与上述工具模板不同，控制和功能模板只显示顶层子模板的图标。在这些顶层子模板中包含许多不同的控制或功能子模板。通过这些控制或功能子模板可以找到创建程序所需的面板对象和框图对象。用鼠标点击顶层子模板图标就可以展开对应的控制或功能子模板，只需按下控制或功能子模板左上角的大头针就可以把对这个子模板变成浮动板留在屏幕上。

控制模板(Controls Palette)

用控制模板可以给前面板添加输入控制和输出显示。每个图标代表一个子模板。如果控制模板不显示，可以用 Windows 菜单的 Show Controls Palette 功能打开它，也可以在前面板的空白处，点击鼠标右键，以弹出控制模板。注：只有当打开前面板窗口时才能调用控制模板。

控制模板如左图所示，它包括如图所示的几个子模板。



由左而右由上而下依次为：

1. 数值子模板：包含数值的控制和显示。
2. 布尔值子模块：逻辑数值的控制和显示。
3. 字符串子模板：字符串和表格的控制和显示。
4. 列表和环（Ring）子模板：菜单环和列表栏的控制和显示。
5. 数组和群子模板：复合型数据类型的控制和显示。
6. 图形子模板：显示数据结果的趋势图和曲线图。
7. 路径和参考名(Refnum)子模板：文件路径和各种标识的控制和显示。
8. 控件容器库子模板：用于操作 OLE、ActiveX 等功能。
9. 对话框子模板：用于输入对话框的显示控制。
10. 修饰子模板：用于给前面板进行装饰的各种图形对象。
11. 用户自定义的控制和显示。
12. 调用存储在文件中的控制和显示的接口。

功能模板(Functions Palette)

功能模板是创建框图程序的工具。该模板上的每一个顶层图标都表示一个子模板。若功能模板不出现，则可以用 Windows 菜单下的 Show Functions Palette 功能打开它，也可以在框图程序窗口的空白处点击鼠标右键以弹出功能模板。

注：只有打开了框图程序窗口，才能出现功能模板。

功能模板如下图所示。



由左而右由上而下依次为：

1. 结构子模板：包括程序控制结构命令，例如循环控制等，以及全局变量和局部变量。
2. 数值运算符模板：包括各种常用的数值运算符，如+、-等；以及各种常见的数值运算式，如+1 运算；还包括数制转换、三角函数、对数、复数等运算，以及各种数值常数。
3. 布尔逻辑子模板：包括各种逻辑运算符以及布尔常数。
4. 字符串运算符模板：包含各种字符串操作函数、数值与字符串之间的转换函数，以及字符(串)常数等。
5. 数组子模板：包括数组运算函数、数组转换函数，以及常数数组等。
6. 群子模板。包括群的处理函数，以及群常数等。这里的群相当于 C 语言中的结构。
7. 比较子模板：包括各种比较运算函数，如大于、小于、等于。
8. 时间和对话框子模板：包括对话框窗口、时间和出错处理函数等。
9. 文件输入/输出子模板：包括处理文件输入/输出的程序和函数。
10. 仪器控制子模板：包括 GPIB(488、488.2)、串行、VXI 仪器控制的程序和函数，以及 VISA 的操作功能函数。
11. 仪器驱动程序库：用于装入各种仪器驱动程序。
12. 数据采集子模板：包括数据采集硬件的驱动程序，以及信号调理所需的各种功能模块。

13. 信号处理子模板：包括信号发生、时域及频域分析功能模块。
14. 数学模型子模板：包括统计、曲线拟合、公式框节点等功能模块，以及数值微分、积分等数值计算工具模块。
15. 图形与声音子模板：包括 3D、OpenGL、声音播放等功能模块。
16. 通讯子模板：包括 TCP、DDE、ActiveX 和 OLE 等功能模块。
17. 应用程序控制子模板：包括动态调用 VI、标准可执行程序的功能函数。
18. 底层接口子模板：包括调用动态连接库和 CIN 节点等功能模块。
19. 文档生成子模板。
20. 示教课程子模板：包括 LabVIEW 示教程序。
21. 用户自定义的子 VI 模板。

“选择…VI 子程序”子模板：包括一个对话框，可以选择一个 VI 程序作为子程序（SUB VI）插入当前程序中。

其它几个子模板是 LabVIEW 的附加 Toolkit 安装上去的。在 LabVIEW 完全版中不包括这些子模板。

第三节 创建一个 VI 程序

请记住，VI 程序具有三个要素：前面板、框图程序和图标/连接器。我们将在练习 1-1 中讨论图标/连接器。

1. 前面板

使用输入控制和输出显示来构成前面板。控制是用户输入数据到程序的接口。而显示是输出程序产生的数据接口。控制和显示有许多种类，可以从控制模板的各个子模板中选取。

两种最常用的前面板对象是数字控制和数字显示。若想要在数字控制中输入或修改数值，你只需要用操作工具（见工具模板）点击控制部件和增减按钮，或者用操作工具或标签工具双击数值栏进行输入数值修改。



2. 框图程序

框图程序是由节点、端点、图框和连线四种元素构成的。

节点类似于文本语言程序的语句、函数或者子程序。LabVIEW 有二种节点类型——函数节点和子 VI 节点。两者的区别在于：函数节点是 LabVIEW 以编译好了的机器代码供用户使用的，而子 VI 节点是以图形语言形式提供给用户的。用户可以访问和修改任一子 VI 节点的代码，但无法对函数节点进行修改。上面的框图程序所示的 VI 程序有两个功能函数节点，一个函数使两个数值相加，另一个函数使两数相减。

端点是只有一路输入/输出，且方向固定的节点。LabVIEW 有三类端点——前面板对象端点、全局与局部变量端点和常量端点。对象端点是数据在框图程序部分和前面板之间传输的接口。一般来说，一个 VI 的前面板上的对象（控制或显示）都在框图中有一个对象端点与之一一对应。当在前面板创建或删除面板对象时，可以自动创建或删除相应的对象端点。控制对象对应的端点在框图中是用粗框框住的，如例子中的 A 和 B 端点。它们只能在 VI 程序框图中作为数据流源点。显示对象对应的端点在框图中是用细框框住的。如例子中的 A+B 和 A-B 端点。它们只能在 VI 程序框图中作为数据流终点。常量端点永远只能在 VI 程序框图中作为数据流源点。

图框是 LabVIEW 实现程序结构控制命令的图形表示。如循环控制、条件分支控制和顺序控制等，编程人员可以使用它们控制 VI 程序的执行方式。代码接口节点（CIN）是框图程序与用户提供的 C 语言文本程序的接口。

连线是端口间的数据通道。它们类似于普通程序中的变量。数据是单向流动的，从源端口向一个或多个目的端口流动。不同的线型代表不同的数据类型。在彩显上，每种数据类型还以不同的颜色予以强调。

下面是一些常用数据类型所对应的线型和颜色：

	标量	一维数组	二维数组	
整形数	——	——	——	兰色
浮点数	——	——	——	橙色
逻辑量	绿色
字符串	~~~~~	~~~~~	~~~~~	粉色
文件路径	——	青色

当需要连接两个端点时，在第一个端点上点击连线工具（从工具模板栏调用），然后移动到另一个端点，再点击第二个端点。端点的先后次序不影响数据流动的方向。

当把连线工具放在端点上时，该端点区域将会闪烁，表示连线将会接通该端点。当把连线工具从一个端口接到另一个端口时，不需要按住鼠标键。当需要连线转弯时，点击一次鼠标键，即可以正交垂直方向地弯曲连线，按空格键可以改变转角的方向。

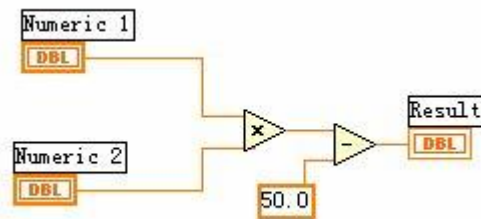
快速提示 接线头是为了帮助正确连接端口的连线。当把连线工具放到端口上，接线头就会弹出。接线头还有一个黄色小标识框，显示该端口的名字。

3. 从框图程序窗口创建前面板对象

用选择和连线工具，你都可以用鼠标右键点击任一节点和端点，然后从弹出菜单中选择“创建常数”，“创建控制”，或“创建显示”等命令。LabVIEW 会自动地在被创建的端点与所点击对象之间接好连线。

4. 数据流编程

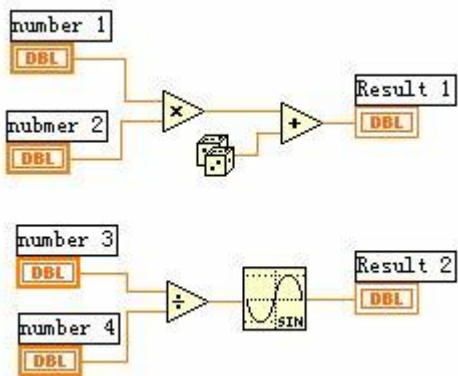
控制 VI 程序的运行方式叫做“数据流”。对一个节点而言，只有当它的所有输入端口上的数据都成为有效数据时，它才能被执行。当节点程序运行完毕后，它把结果数据送给所有的输出端口，使之成为有效数据。并且数据很快从源送到目的端口。



如左图所示，这个 VI 程序把两个输入数值相乘，再把乘积减去 50.0。这个程序中，框图程序从左往右执行，这个执行次序不是由于对象的摆放位置，而是由于相减运算函数的一个输入量是相乘函数的运算结果，它只有当相乘运算完成并把结果送到减运算的输入口后才能继续下去。请记住，一个节点（函数）只有当它所有的输入端的数据都成为有效数据后才能被执行，而且只有当它执行完成后，它的所有输出端口上的数据才成为有效。

再看另一个程序(见下图)，你认为哪一个节点函数将先执行 - 是乘法还是除法？在这个例子中，我们无法知道哪一个节点函数首先执行，因为所有输入量几乎同

时到达。对于这样一种相互独立的数据流程，如果又必须明确指定节点执行的先后次序，就必须使用顺序（Sequence）结构来明确执行次序。



第四节 程序调试技术

1. 找出语法错误

如果一个 VI 程序存在语法错误，则在面板工具条上的运行按钮将会变成一个折断的箭头，表示程序不能被执行。这时这个按钮被称作错误列表。点击它，则 LabVIEW 弹出错误清单窗口，点击其中任何一个所列出的错误，选用 Find 功能，则出错的对象或端口就会变成高亮。

2. 设置执行程序高亮

在 LabVIEW 的工具条上有一个画着灯泡的按钮，这个按钮叫做“高亮执行”按钮。点击这个按钮使该按钮图标变成高亮形式，再点击运行按钮，VI 程序就以较慢的速度运行，没有被执行的代码灰色显示，执行后的代码高亮显示，并显示数据流线上的数据值。这样，你就可以在根据数据的流动状态跟踪程序的执行。

3. 断点与单步执行

为了查找程序中的逻辑错误，你也许希望框图程序一个节点一个节点地执行。使用断点工具可以在程序的某一地点中止程序执行，用探针或者单步方式查看数据。使用断点工具时，点击你希望设置或者清除断点的地方。断点的显示对于节点或者图框表示为红框，对于连线表示为红点。当 VI 程序运行到断点被设置处，程序被暂停在将要执行的节点，以闪烁表示。按下单步执行按钮，闪烁的节点被执行，下一个将要执行的节点变为闪烁，指示它将被执行。你也可以点击暂停按钮，这样程序将连续执行直到下一个断点。

4. 探针

你可以用探针工具来查看当框图程序流经某一根连接线时的数据值。从 Tools 工具模板选择探针工具，再用鼠标左键点击你希望放置探针的连接线。这时显示器上会出现一个探针显示窗口。该窗口总是被显示在前面板窗口或框图窗口的上面。在框图中使用选择工具或连线工具，在连线上点击鼠标右键，在连线的弹出式菜单中选择“探针”命令，同样可以为该连线加上一个探针。

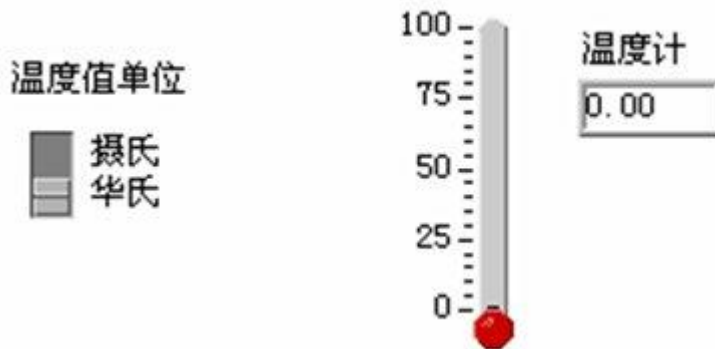
第五节 练习 1-1

- **目的：**创建一个 VI 程序，以便以后作为子 VI 程序使用。

创建一个 VI 程序模拟温度测量。假设传感器输出电压与温度成正比。例如，当温度为 70° F 时，传感器输出电压为 0.7V。本程序也可以用摄氏温度来代替华氏温度显示。

本程序用软件代替了 DAQ 数据采集卡。使用 Demo Read Voltage 子程序来仿真电压测量，然后把所测得的电压值转换成摄氏或华氏温度读数。

- **前面板：**



1. 用 File 菜单的 New 选项打开一个新的前面板窗口。
2. 把温度计指示部件放入前面板窗口。
 - a. 在前面板窗口的空白处点击鼠标键，然后从弹出的 Numeric 子模板中选择 Thermometer。
 - b. 在高亮的文本框中输入“温度计”，再点击鼠标键按钮。
3. 重新设定温度计的标尺范围为 0.0 到 100.0。使用标签工具 A，双击温度计标尺的 10.0，输入 100.0，再点击鼠标键或者工具栏中的 V 按钮。
4. 在前面板窗口中放入垂直开关控制。

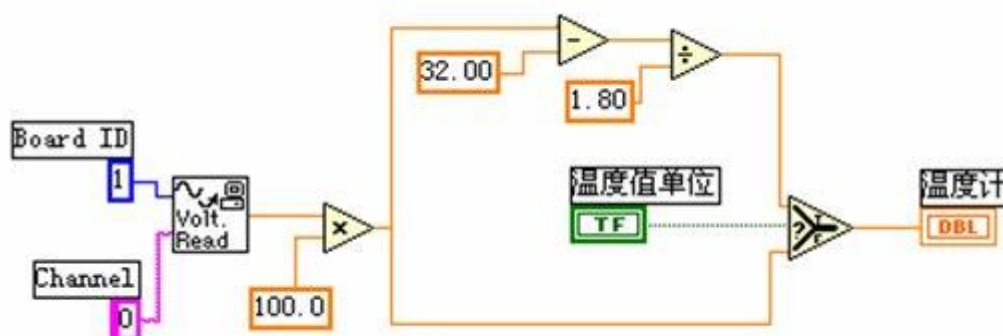
a. 在面板窗口的空白处点击鼠标键 然后弹出的 Boolean 子模板中选择 Vertical Switch,

在文本框中输入“温度值单位”,再点击鼠标键或者工具栏中的 V 按钮。

b. 使用标签工具 A, 在开关的“条件真”(true)位置旁边输入自由标签“摄氏”,

再在“条件假”(false)位置旁边输入自由标签“华氏”。

- 框图程序:



1. 从 Windows 菜单下选择 Show Diagram 功能打开框图程序窗口。

2. 点击框图程序窗口的空白处 弹出功能模板 从弹出的菜单中选择所需的对象。本程序用到下面的对象:

- Demo Read Voltage VI 程序 (Tutorial 子模板)。在本例中, 该程序模拟从 DAQ 卡的 0 通道读取电压值。
- Multiply (乘法) 功能 (Numeric 子模板)。在本例中, 将读取电压值乘以 100.00, 以获得华氏温度。
- Subtract (减法) 功能 (Numeric 子模板)。在本例中, 从华氏温度中减去 32.0, 以转换成摄氏温度。
- Divide (除法) 功能 (Numeric 子模板)。在本例中, 把相减的结果除以 1.8 以转换成摄氏温度。
- Select (选择) 功能 (Comparison 子模板)。取决于温标选择开关的值。该功能输出华氏温度 (当选择开关为 false) 或者摄氏温度 (选择开关为 True) 数值。
- 数值常数。用连线工具, 点击你希望连接一个数值常数的对象, 并选择 Create Constant 功能。若要修改常数值, 用标签工具双点数值, 再写入新的数值。
- 字符串常量。用连线工具, 点击你希望连接字符串常量的对象, 再选择 Create Constant 功能。要输入字符串, 用标签工具双击字符串, 再输入新的字符串。

3. 使用移位工具 (Positioning tool)，把图标移至图示的位置，再用连线工具连接起来。

请记住，如果要显示图标接线端口，则点击图标，再从弹出菜单中选择 Show Terminal s 功能。

也可以从 Help 菜单中选择 Show Help 功能以打开帮助信息窗口。

Demo Read Voltage VI 子程序模拟从数据采集卡的 0 通道读取电压，我们的程序再将读数乘以 100.0 转换成华氏温度读数，或者再把华氏温度转换成摄氏温度。

4. 选择前面板窗口，使之变成当前窗口，并运行 VI 程序。点击连续运行按钮，便程序运行于连续运行模式。

5. 再点击连续运行按钮，关闭连续运行模式。

6. 创建图标 Temp:

此图标可以将现程序作为子程序在其他程序中调用。创建方法如下:

- a. 在面板窗口的右上角的图标框中点击鼠标，从弹出菜单中选择 Edit Icon 功能。
- b. 双点选择工具，并按下 Delete 键，消除缺省的图标图案。
- c. 用画图工具画出温度计的图标。

注意，在用鼠标画线时按下 <shift> 键，则可以画出水平或垂直方向的连线。

使用文本工具写入文字，双击文本工具把字体换成 Small Font。

当图标创建完成后，点击 OK 以关闭图标编辑。生成的图标在面板窗口的右上角。

7. 创建连接器端口:

- a. 点击右上角的图标面板，从弹出菜单中选择 Show Connector 功能。
LabVIEW 将会根据控制和显示的数量选择一种连接器端口模式。在本例中，只有两个端口，一个是竖直开关，另一个是温度指示。
- b. 把连接器端口定义给开关和温度指示。
- c. 使用连线工具，在左边的连接器端口框内按鼠标键，则端口将会变黑。
再点击开关控制件，一个闪烁的虚线框将包围住该开关。
- d. 现在再点击右边的连接器端口框，使它变黑。再点击温度指示部件，一个闪烁的虚线框将包围住温度指示部件，这即表示着右边的连接器端口对应温度指示部件的数据输入。
- e. 如果再点击空白外，则虚线框将消失，而前面所选择的连接器端口将变暗，表示你已经将对象部件定义到各个连接器端口。

注意：LabVIEW 的惯例是前面板上控制的连接器端口放在图标的接线面板的左边，而显示的连接器端口放在图标的接线面板的右边。也就是说，图标的左边为输入端口而右边为输出端口。

以后我们会讲到如何使用 SubVI From Selection 功能来创建子程序 SubVI。

8. 确认当前文件的程序库路径为 Seminar.LLB，用文件菜单的 SAVE 功能保存上述文件，并将文件命名为 Thermometer.Vi。

现在，该程序已经编制完成了。它可以在其他程序中作为子程序来调用，在其它程序的框图窗口里，该温度计程序用前面创建的图标来表示。连接器端口的输入端用于选择温度单位，输出端用于输出温度值。

9. 关闭该程序

第六节 把一个 VI 程序作为子 VI 程序调用

在任意一个 VI 程序的框图窗口里，都可以把其他的 VI 程序作为子程序调用，只要被调用 VI 程序定义了图标和连接器端口即可。用户使用功能模板的 Select a VI 来完成。当使用该功能时，将弹出一个对话框，用户可以输入文件名。

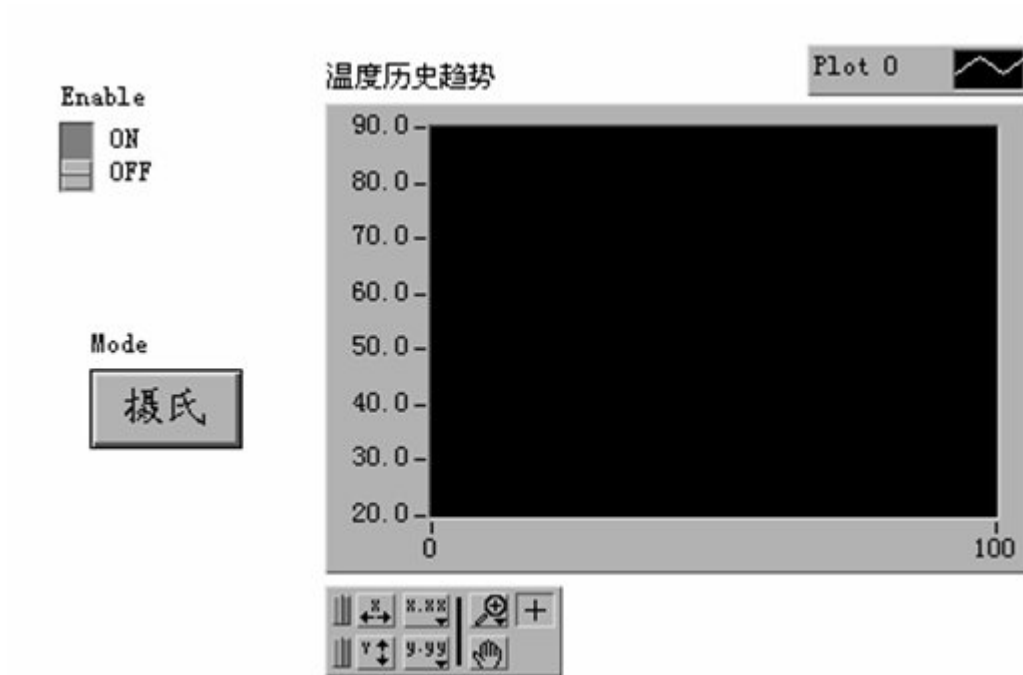
一个子 VI 程序，相当于普通程序的子程序。节点相当于子程序调用。子程序节点并不是子程序本身，就象一般程序的子程序调用语句并不是子程序本身一样。如果在一个框图程序中，有几个相同的子程序节点，它就象多次调用相同的子程序。请注意，该子程序的拷贝并不会在内存中存储多次。

第七节 练习 1-2

目的： 使用一个条件循环结构和一个被测波形图表实时地采集数据。

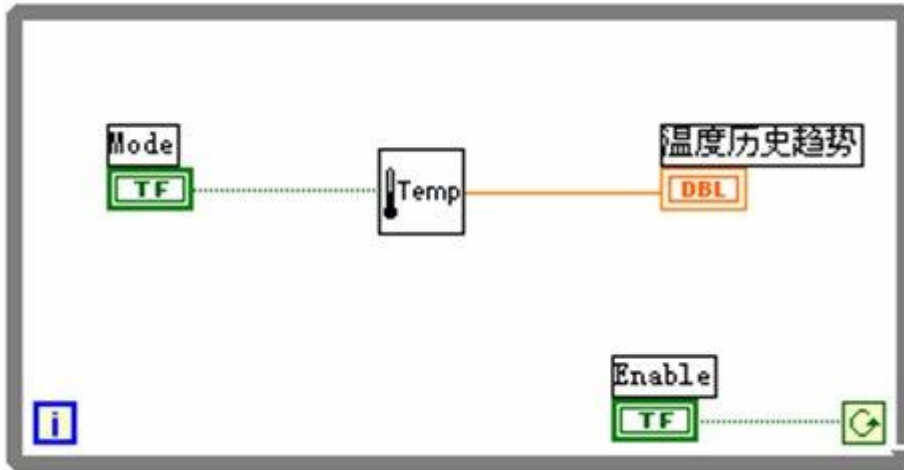
学员将创建一个 VI 程序，进行温度测量，并把结果在波形图表上显示。该 VI 程序使用我们前面创建的温度计程序（Thermometer VI）作为子程序。

前面板：



1. 打开一个新的前面板窗口，在里面放一个竖直开关（在 Boolean 逻辑部件子模板），给该开关标注为“Enable”。你可以用该开关来开始/停止数据采集。
2. 在前面板内再放置一个趋势图（Graph 子模板中的 Waveform Chart），标注为“温度历史趋势”。该图表将实时地显示温度值。
3. 由于趋势图将它的图标注解 plot 自动地标注为“plot 0”，你可以用标注工具将其重新标注为“Temp”。
4. 因为趋势图用于显示室内温度，需要对它的标尺进行重新定标。将 Y 轴的“10”改为“90”，而将“0.0”改为“20”。
5. 此时暂时不要创建模式转换开关，我们将尝试从框图程序窗口创建前面板的部件。

框图程序：



1. 打开框图程序窗口。
2. 从结构（Structures）工具模板选择条件循环结构“While Loop”放入框图程序窗口，调整该条件循环框的大小，把先前从前面板创建的两个节点放入循环框内。

条件循环结构是一种无限循环结构，只要条件满足，它就一直循环运行下去。在本例中，只要允许开关（Enable Switch）是 ON 状态，该 VI 程序就一直运行，采集温度测量值，并在图表上显示。
3. 放入其它的框图程序对象。Thermometer VI，这个 VI 程序是你在上个练习中创建的，从 Seminar.LLB 中调出（从 Select a VI…子模板）。
4. 按照上图的框图程序连好线。
5. 创建模式开关。把连线工具放在 Thermometer VI 的 Mode 输入端口上，按鼠标右键并选择 Create Control，这样就可以自动创建模式转换开关，并将它与 Thermometer VI 子程序相连线，再转换到前面板窗口，将模式转换开关的位置重新调整。
6. 在前面板窗口，使用标注工具，双击模式开关的“OFF”标签，并把它转换成“华氏”，再把“ON”标签转换成“摄氏”。要转换开关状态，使用操作工具（Operating Tool）。
7. 将模式开关设置为 ON 状态，运行该 VI 程序。
8. 要停止数据采集，点击 Enable 开关，使其状态变为 OFF，循环结束。
9. 修改 Enable 开关缺省设置，使你运行 VI 程序时不必每次打开该开关。
 - a. 若程序在运行状态，则关闭程序运行。
 - b. 把开关设置为 ON 状态。

c. 点击开关，从弹出菜单中选择 Data Operations>Make Current Value Default 选项，这将使 ON 状态变为缺省值。

d. 再点击开关，从弹出菜单中选择 Mechanical Action>Latch When Pressed 选项。

10. 运行该程序，把开关点击为 Stop 状态以停止数据采集。开关将变为 OFF 状态，但当条件循环结构再次读取其数值时，它又会变成 ON 状态。

增加定时器控制：

当你运行程序时，它将会尽可能快地运行。但是你可能希望以一定的时间间隔，例如一秒钟一次或者一分钟一次来采集数据。

你可以用 Wait Until Next ms Multiple 功能（在 Time & Dialog 子模板）来满足上述条件。该功能模块可以保证循环间隔时间不少于指定的毫秒数。

11. 如右图所示，使 VI 程序采样间隔为 500 毫秒。使用 Time & Dialog 子模板中的 Wait Until Next ms Multiple 功能，再加上时间常数 Numeric Constant，把它设置为 500。

12. 运行上述程序，试用不同的时间间隔值。

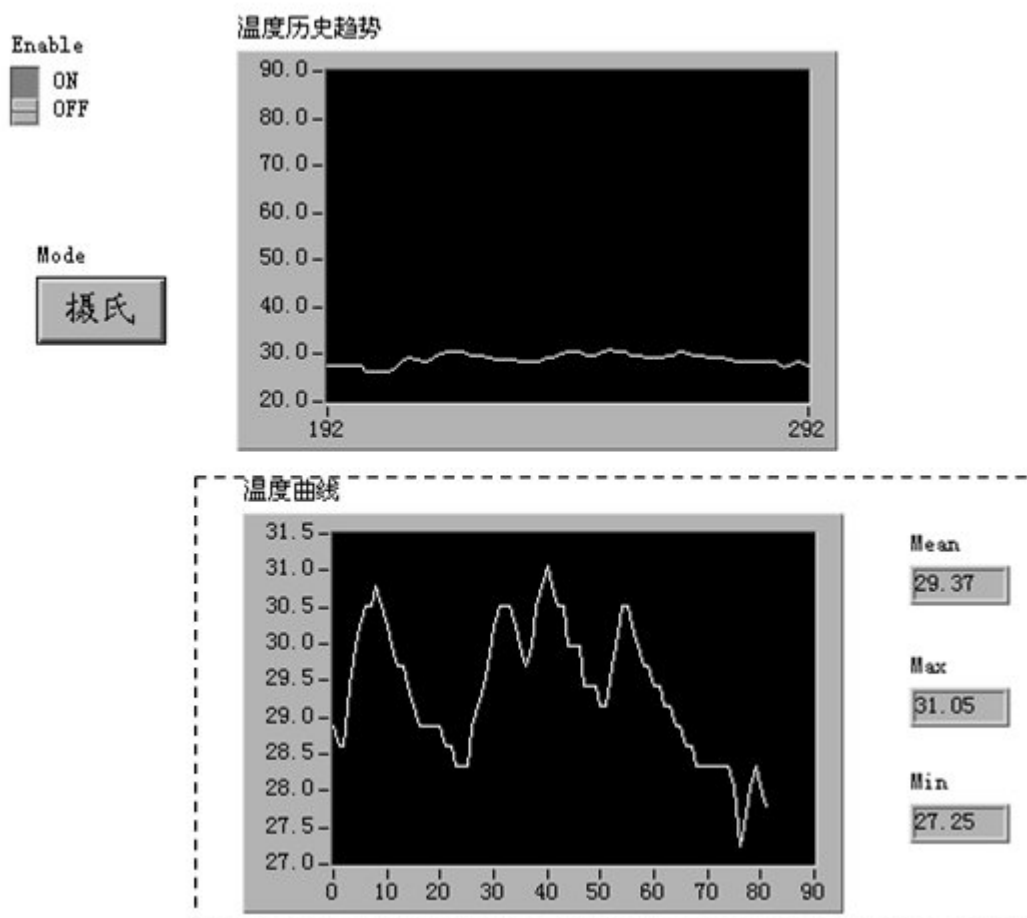
13. 关闭并保存上述程序，文件名为 Temperature Monitor.vi。

第八节 练习 1-3

目的：以图表方式显示数据并使用分析功能子程序。

利用练习 1-2 创建的 VI 程序，在数据采集过程中，实时地显示数据。当采集过程结束后，在图表上画出数据波形，并算出最大值、最小值和平均值。该练习只使用华氏温度单位。

前面板：

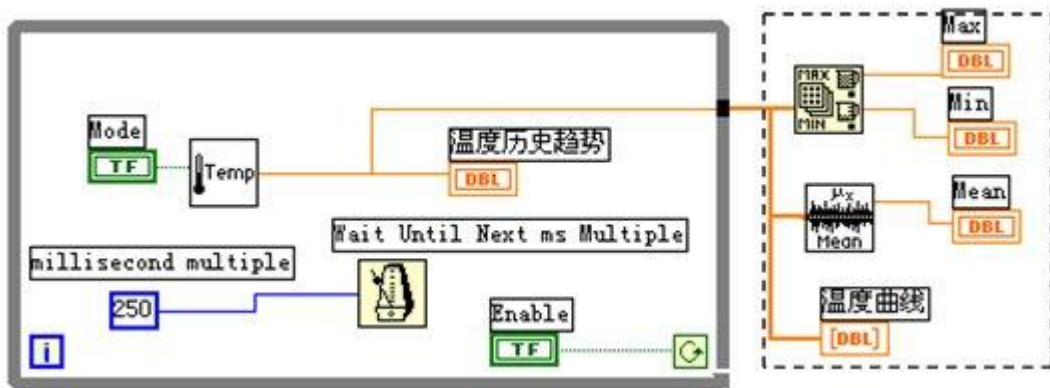


1. 打开练习 1-2 创建的 Temperature monitor.vi 程序。
2. 按照上图修改程序，其中被虚线框住的部分是新增加的。

趋势图“温度历史趋势”显示实时采集的数据。采集过程结束后，在 Temp Graph 中画出数据曲线，同时在 Mean、Max 和 Min 数字显示栏中显示出温度的平均值、最大值和最小值。

本例中只使用华氏温度单位。

框图程序：



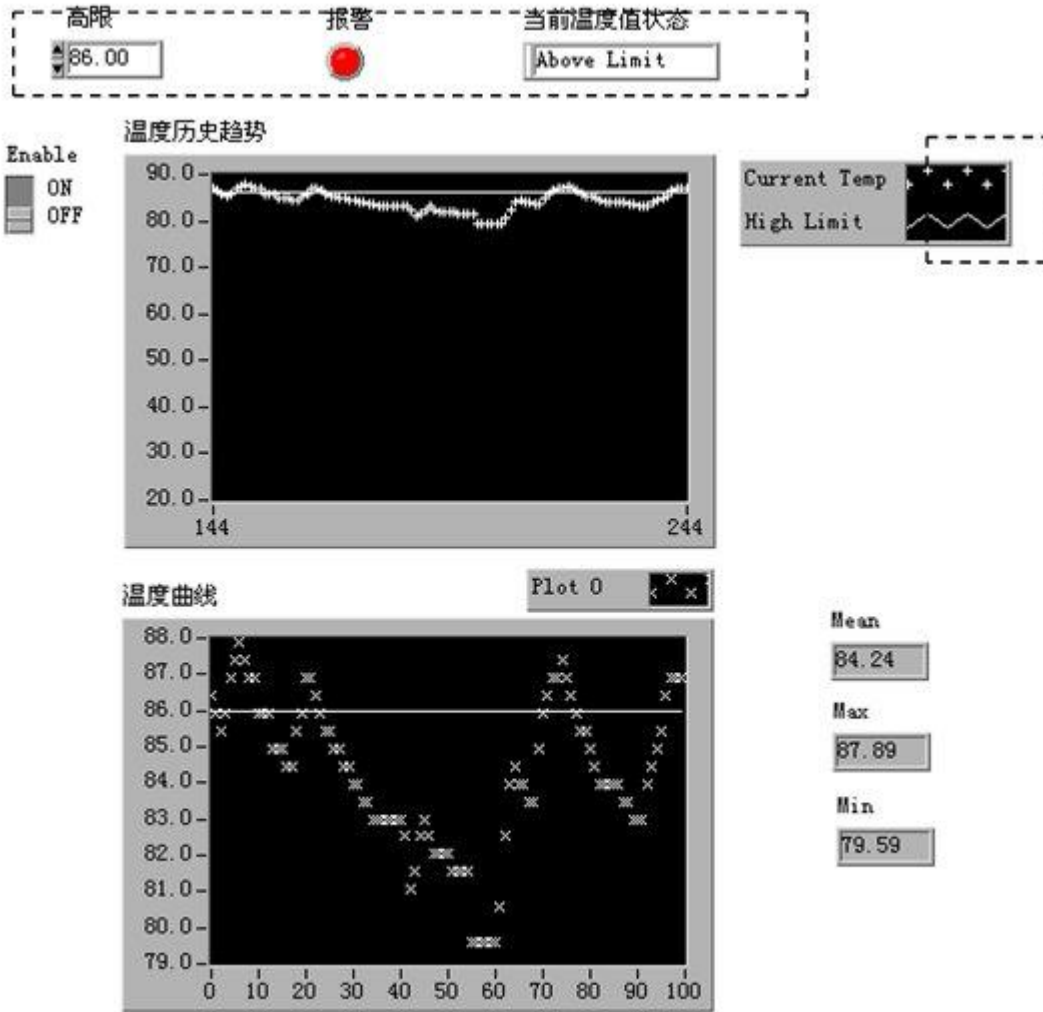
1. 完成上面的框图程序。被虚线框住的部分表示新增程序。条件循环框边线上的方块叫作通道（tunnel）。在本例中，通道是条件循环的数据通道口。若要建立数据索引，点击通道，并选择 Enable Indexing 选项，表示当条件循环执行时，把数据顺序放入一个数组中。循环结束后，通道输出该数组。否则，通道仅输出最后一次循环放入的数据值。
2. 返回前面板，并运行 VI 程序。
3. 当允许运行开关（Enable Switch）设置为 OFF 后，将显示温度数据曲线。
4. 面修改后的程序重命名为 Temperature Analysis.vi 并存盘。

第九节 练习 1-4

目的：学习使用 Case 结构。

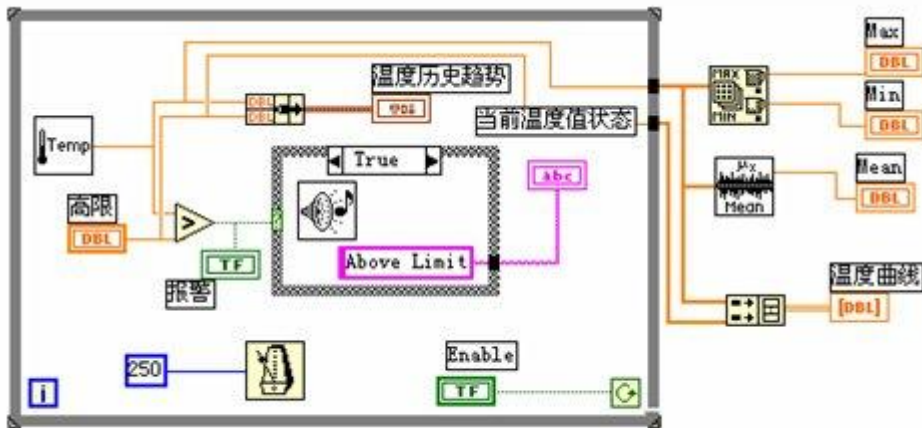
修改 Temperature Analysis.VI 程序以检测温度是否超出范围，当温度超出上限（High Limit）时，前面板上的 LED 将点亮，并且有一个蜂鸣器发声。

前面板：

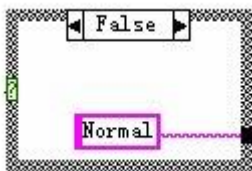


1. 打开练习 1-3 创建的 Temperature Analysis VI 程序。
2. 按照下图所示修改前面板。被虚线框住的部分表示增加的部件。High Limit 表示温度上限值。报警指示灯 (WARNING LED) 和当前温度状态 (Current Temperature State) 用来表示温度是否超限。点击趋势图, 并且选择 Show>Legend 和 Show>Digital Display 选项, 可以增加图注 (Legend) 和数字显示。

框图程序:



1. 按照上图编写框图程序。被虚线框住的部分为新增加的部件。下面的 FALSE Case 与图中的 TRUE Case 同属于一个 Case 结构。根据其输入端上的数值，来决定执行哪一个 Case 程序。如果 Thermometer Vi 子程序返回的温度值大于 High Limit 数值，将执行 True Case 程序，反之则执行 False Case 程序。



2. 返回前面板程序，在 High Limit 控制栏中输入 86。再运行 VI 程序。当温度超过 86 时，LED 将点亮，蜂鸣器也会发声。

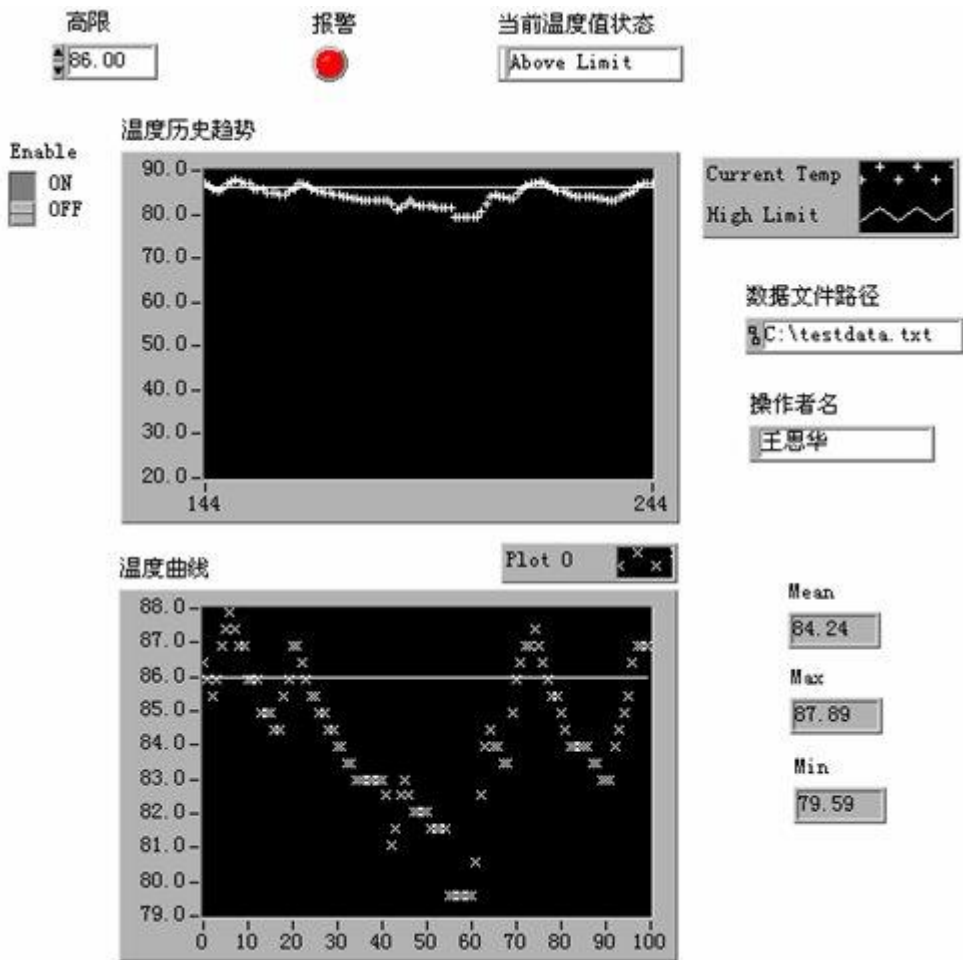
3. 将程序重新命名为 Temperature Control.vi，并保存起来。

第十节 练习 1-5

目的：学习使用顺序(Sequence)结构并且把数据传送到文件。

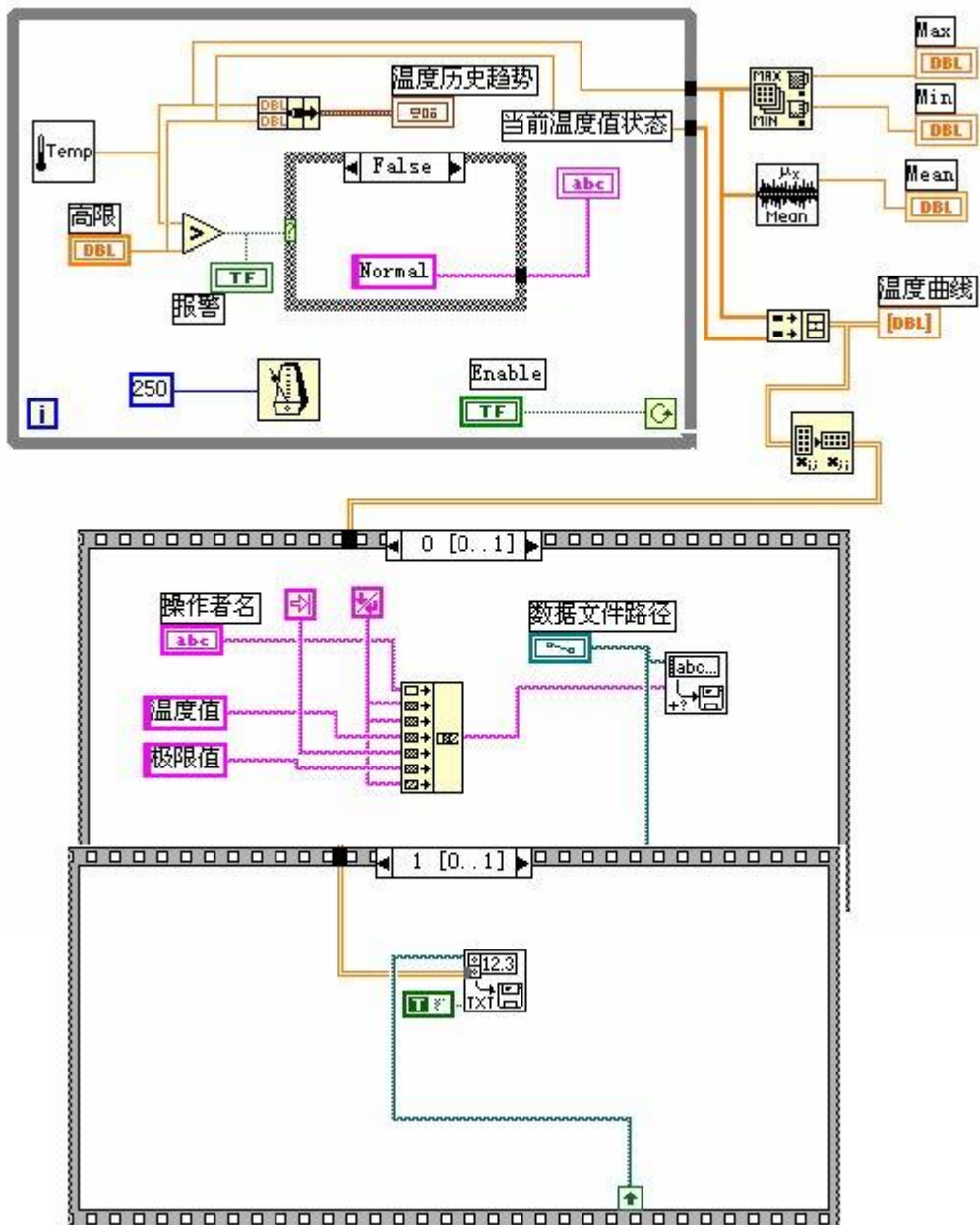
学员将修改 Temperature Control.vi 程序，以学习使用顺序结构和包括头文件的数据文件。当程序停止数据采集后，自动产生数据文件的头文件，它包括操作者名字和文件名，然后将采集的数据附在头文件后面。

前面板：



如上图所示。

框图程序:



1. 如图所示, 完成框图程序, 使用了如下的功能模块: Write Characters to File 模块 (在 File I/O 子模板)。该功能把一个字符串写入一个新的文件或者附加到一个已存在的文件中。它在写入前打开或者创建一个文件, 在完成时关闭该文件。在本例中, 它用来建立头文件格式。

Write to Spreadsheet File 模块 (在 File I/O 子模板)。该模块把一个二维或者一维单精度数组转换成字符串, 并把字符串写入一个新文件或者附回在一个已存在的文件后面。在本例中, 它将由温度采集数据和上限值组成的二维数组附加在一个已创建了头文件的数据文件后面。

Transpose 2D Array 模块 (在 Array 子模板)。在本例中, 它把二维数组转换成以列为分界的二维数组, 这样在写入数据文件时它就会以列的形式显示。

Concatenate String 模块（在 String 子模板）。在本例中，它用于创建头文件字符串。

顺序结构程序按指定的顺序执行各个程序步骤。在本例中，我们首先创建数据文件的头文件，然后再将数据写入文件中。

2. 返回前面板，在 High Limit 控制栏中输入 86，在 Operator Name 控制栏中输入你的名字，再输入数据文件名（例如 C:\testdata.txt）。运行该程序。当 Enable 开关设置为 OFF 状态后，将产生如下的 ASCII 文件。

王思华

```
TEMP LIMIT
86.426 86.000
85.937 86.000
85.449 86.000
85.937 86.000
86.912 86.000
87.402 86.000
87.891 86.000
87.402 86.000
86.914 86.000
86.914 86.000
85.937 86.000
85.937 86.000
```

3. 将文件重新命名为 Temperature Control Logger.vi，并存盘退出（用 Save As 选项）。

（练习 1 - 5 结束）。

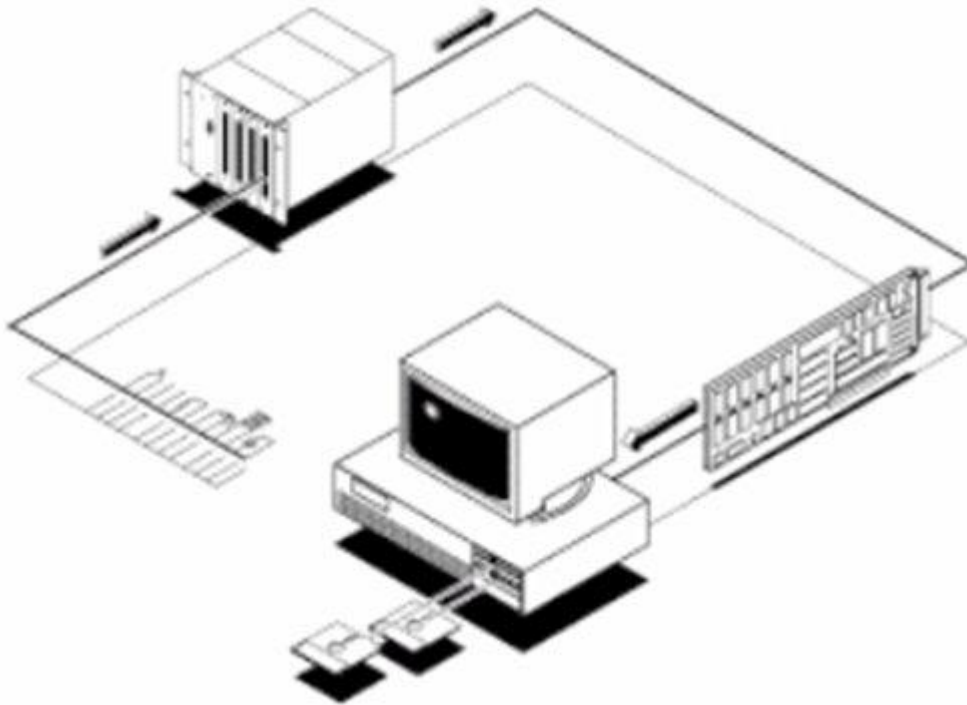
第二课 数据采集

第一节 概述

LabVIEW 的数据采集（Data Acquisition）程序库包括了许多 NI 公司数据采集（DAQ）卡的驱动控制程序。通常，一块卡可以完成多种功能 - 模/数转换，数/模转换，数字量输入/输出，以及计数器/定时器操作等。用户在使用之前必须对 DAQ 卡的硬件进行配置。这些控制程序用到了许多低层的 DAQ 驱动程序。本课程需要一块安装好的 DAQ 卡以及 LabVIEW 开发系统。

数据采集系统的组成：

DAQ 系统的基本任务是物理信号的产生或测量。但是要使计算机系统能够测量物理信号，必须要使用传感器把物理信号转换成电信号（电压或者电流信号）。有时不能把被测信号直接连接到 DAQ 卡，而必须使用信号调理辅助电路，先将信号进行一定的处理。总之，数据采集是借助软件来控制整个 DAQ 系统 - 包括采集原始数据、分析数据、给出结果等。



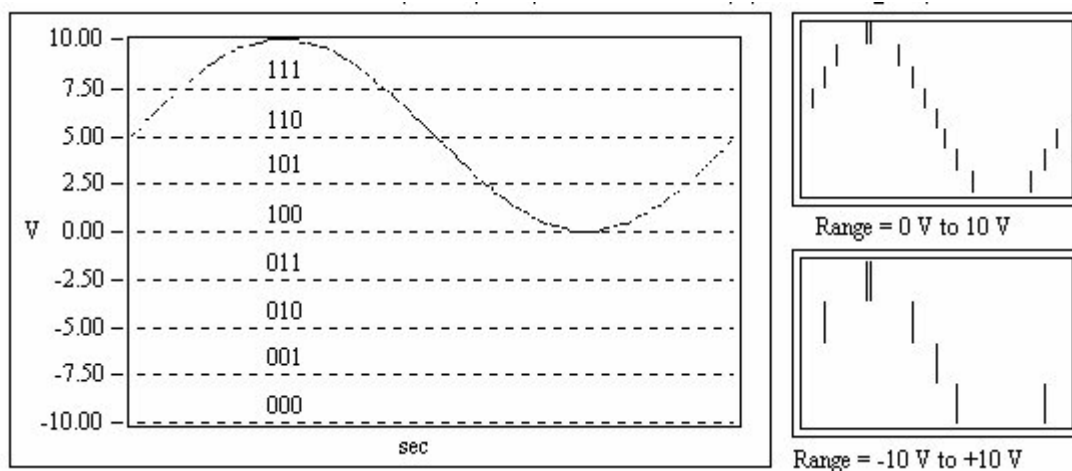
上图中描述了插入式 DAQ 卡。另一种方式是外接式 DAQ 系统。这样，就不需要在计算机内部插槽中插入板卡，这时，计算机与 DAQ 系统之间的通讯可以采用各种不同的总线，如 USB, 并行口或者 PCMCIA 等完成。这种结构适用于远程数据采集和控制系统。

模拟输入：

当采用 DAQ 卡测量模拟信号时，必须考虑下列因素：输入模式（单端输入或者差分输入）、分辨率、输入范围、采样速率，精度和噪声等。

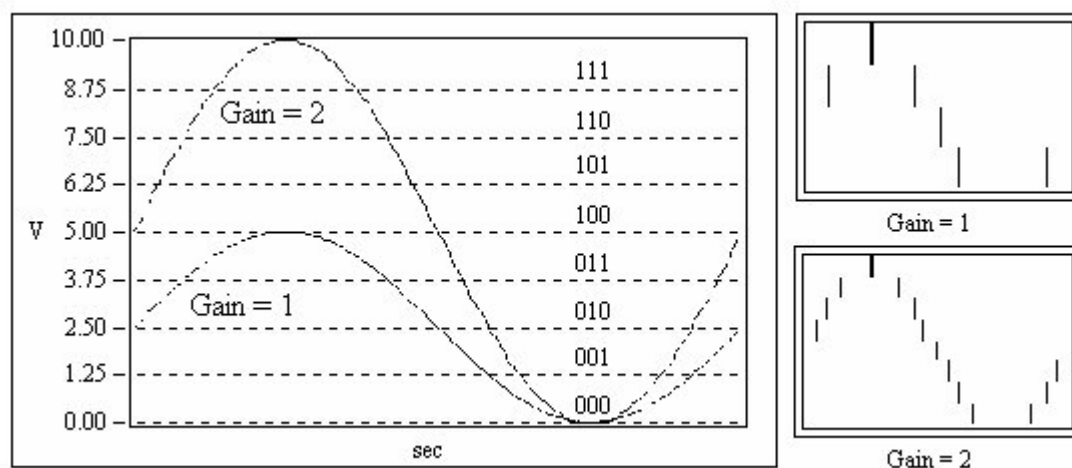
单端输入以一个共同接地点为参考点。这种方式适用于输入信号为高电平（大于一伏），信号源与采集端之间的距离较短（小于 15 英尺），并且所有输入信号有一个公共接地端。如果不能满足上述条件，则需要使用差分输入。差分输入方式下，每个输入可以有不同的接地参考点。并且，由于消除了共模噪声的误差，所以差分输入的精度较高。

输入范围是指 ADC 能够量化处理的最大、最小输入电压值。DAQ 卡提供了可选择的输入范围，它与分辨率、增益等配合，以获得最佳的测量精度。



分辨率是模/数转换所使用的数字位数。分辨率越高，输入信号的细分程度就越高，能够识别的信号变化量就越小。下图表示的是一个正弦波信号，以及用三位模/数转换所获得的数字结果。三位模/数转换把输入范围细分为 2³ 或者就 8 份。二进制数从 000 到 111 分别代表每一份。显然，此时数字信号不能很好地表示原始信号，因为分辨率不够高，许多变化在模/数转换过程中丢失了。然而，如果把分辨率增加为 16 位，模/数转换的细分数值就可以从 8 增加到 2¹⁶ 即 65536，它就可以相当准确地表示原始信号。

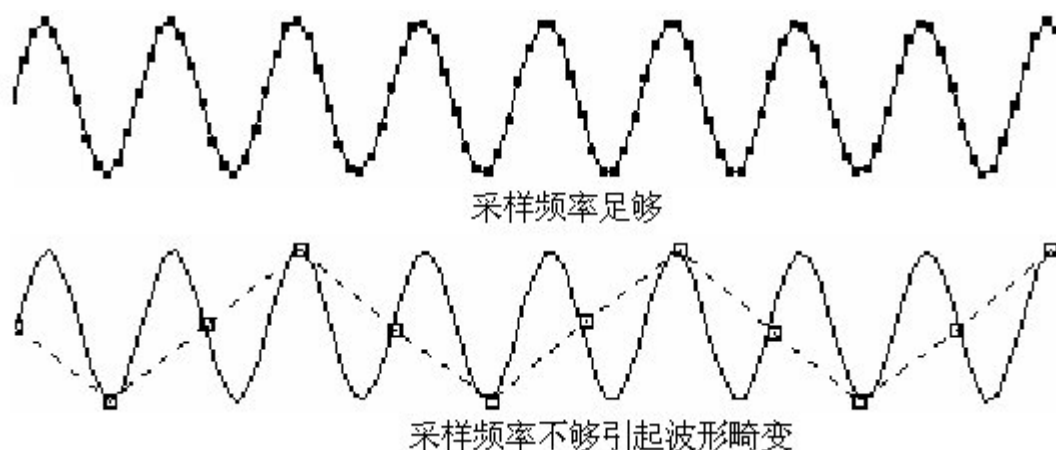
增益表示输入信号被处理前放大或缩小的倍数。给信号设置一个增益值，你就可以实际减小信号的输入范围，使模数转换能尽量地细分输入信号。例如，当使用一个 3 位模数转换，输入信号范围为 0 到 10 伏，上面的图显示了给信号设置增益值的效果。当增益=1 时，模/数转换只能在 5 伏范围内细分成 4 份，而当增益=2 时，就可以细分成 8 份，精度大大地提高了。但是必须注意，此时实际允许的输入信号范围为 0 到 5 伏。一旦超过 5 伏，当乘以增益 2 以后，输入到模/数转换的数值就会大于允许值 10 伏。



总之，输入范围，分辨率以及增益决定了输入信号可识别的最小模拟变化量。此最小模拟变化量对应于数字量的最小位上的 0, 1 变化，通常叫做转换宽度 (Code width)。其算式为：输入范围 / (增益 * 2^{分辨率})。

例如，一个 12 位的 DAQ 卡，输入范围为 0 到 10 伏，增益为 1，则可检测到 2.4mV 的电压变化。而当输入范围为 -10 到 10 伏 (20 伏)，可检测的电压变化量则为 4.8mV。

采样率决定了模/数变换的速率。采样率高，则在一定时间内采样点就多，对信号的数字表达就越精确。采样率必须保证一定的数值，如果太低，则精确度就很差。下面的图表示了采样率对精度的影响。



根据耐奎斯特采样理论，你的采样频率必须是信号最高频率的两倍。例如，音频信号的频率一般达到 20KHz，因此其采样频率一般需要 40KHz。

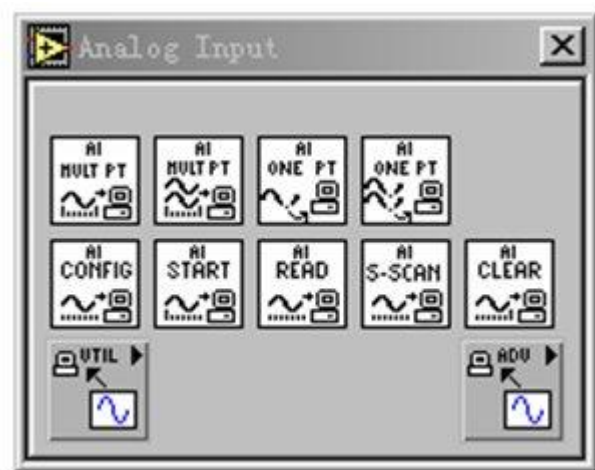
平均化。噪声将会引起输入信号畸变。噪声可以是计算机外部的或者内部的。要抑制外部噪声误差，可以使用适当的信号调理电路，也可以增加采样信号点数，再取这些信号的平均值以抑制噪声误差，这样误差可以减小到乘以下面的系数：

$$\frac{1}{\sqrt{\text{采样点数}}}$$

例如，如果以 100 个点来平均，则噪声误差将减小 1/10。

第二节 数据采集 VI 程序的调用方法

LabVIEW 的 DAQ 程序包括模拟输入、模拟输出、计数器操作、或者数字输入、输出等。你可以在框图程序窗口下选择 Data Acquisition。再在此子模板下，可以看到 6 个子模板：Analog Input, Analog Output, Digital I/O, Counter, Calibration and Configuration 和 Signal Conditioning。



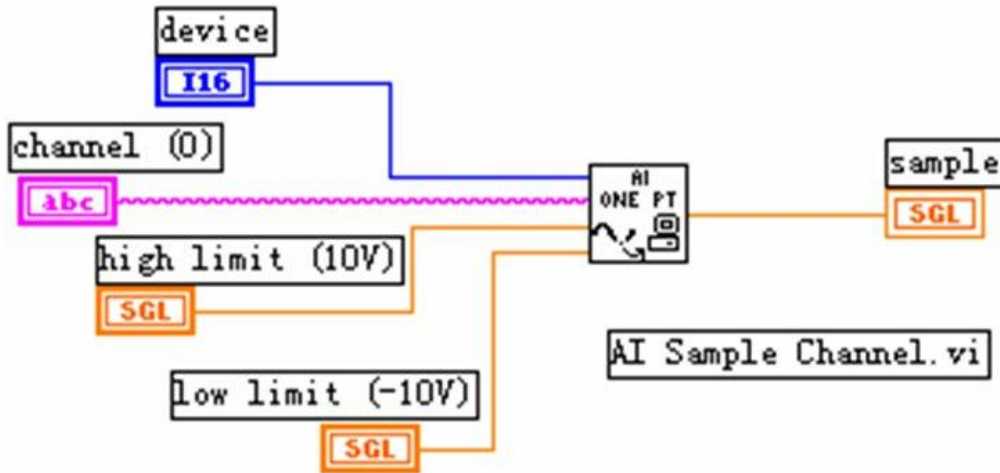
在上述 6 个子模板下，每个子模板又分成 Easy I/O Vis, Intermediate Vis, Utility Vis 和 Advanced Vis。下面我们以前面的 Analog Input 子模板为例进行解释。如图所示，在其顶层是简单模拟量输入 VI 程序（Easy AI），中间层是中级模拟量输入 VI 程序（Intermediate AI），然后下面是两个图标，一个用于调用模拟输入应用程序（Utility AI），一个调用高级模拟量输入程序（Advanced AI）。在本教材中，我们主要讲述简单和中级程序。高级程序涉及到与 DAQ 驱动程序的底层接口，而通用应用程序是中级程序的组合。

简单 I/O 程序仅仅只是表面层接口程序，它能执行基本的模拟量输入、模拟量输出、数字 I/O，以及计数器/定时器操作。它简单易用，并且包含一个简单的出错处理方法。当出错时，将弹出一个对话框，显示出错信息，而用户可以选择中止程序执行或者忽略错误。

相对于简单程序而言，中级程序具备更多的硬件设置功能，灵活性，并且可以更有效地开发实用程序。它具有许多简单程序所缺乏的功能，如外部时钟等。它对出错的处理也更加灵活，可以把出错状态信息传递给其它的程序，编程处理出错情况。

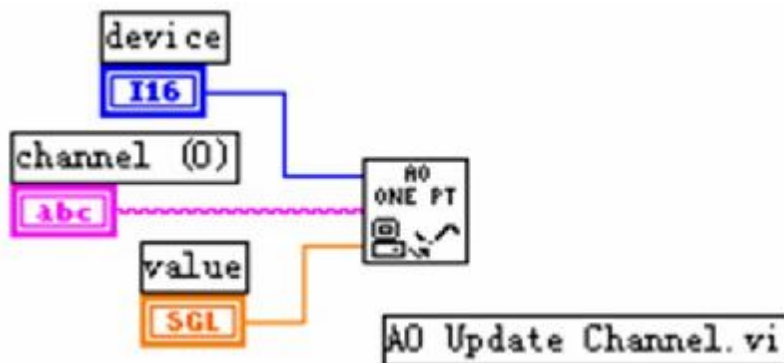
第三节 模拟输入与输出

在 Data Acquisition 子模板下的 Analog Input 子模板程序执行模/数转换操作。而 Analog Output 程序库则执行数/模转换或多重转换。



AI Sample Channel 程序测量指定通道上信号的一个采样点，并返回测量值。Device 是 DAQ 卡的设备编号，Channel 是描述模拟输入通道号的字符串，High Limit 和 Low Limit 指明输入信号的范围，缺省值为+10V 和-10V。

AO Update Channel 程序把一个给定电压值在一个模拟输出通道上输出。Device 是 DAQ 卡的设备编号，Channel 是输出通道号字符串 Voltage 是输出的电压值。



在上述程序执行时，如果发生错误，则弹出一个对话框，显示错误代码，你可以选择中止程序或者继续程序执行。

练习 2-1

目的： 输出一个模拟电压信号，并且用 DAQ 卡再次采集该信号。

首先编制一个 VI 程序，以 0.5 伏的间隔从 0 伏到 9.5 伏输出电压，再编制 VI 程序进行单点模拟输入电压测量，验证上述输出电压。最后，再修改程序，使之控制 Demo Box 上的 LED 指示灯。在演示盒 Demo Box 上，把模拟输出 CH0 与模拟输入 CH1 相连。

前面板:

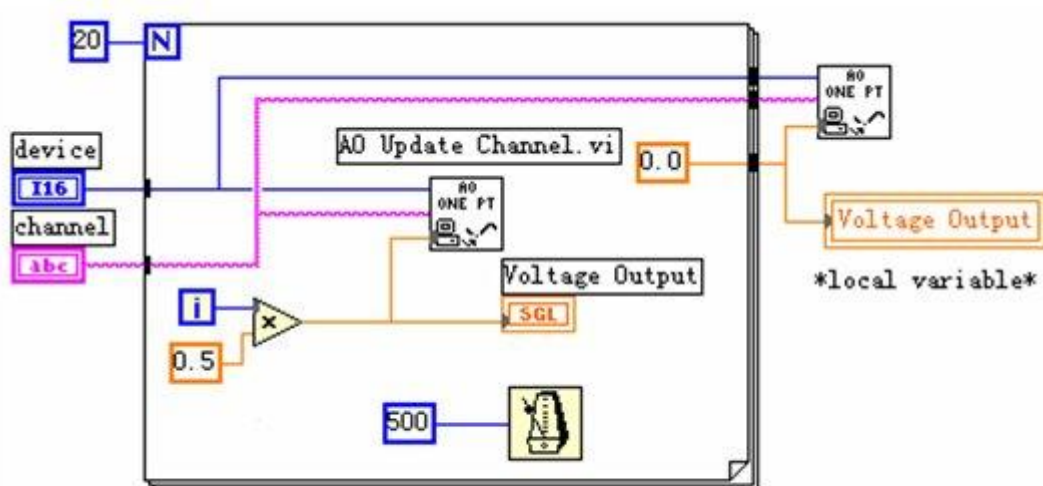


1. 打开电压输出 Voltage Output 程序窗口，用 Device 指定 DAQ 卡设备编号，Channel 字符串指定模拟输出通道，用 Voltage Output 输出显示来显示当前输出电压值。

2. 切换到框图程序。

框图程序

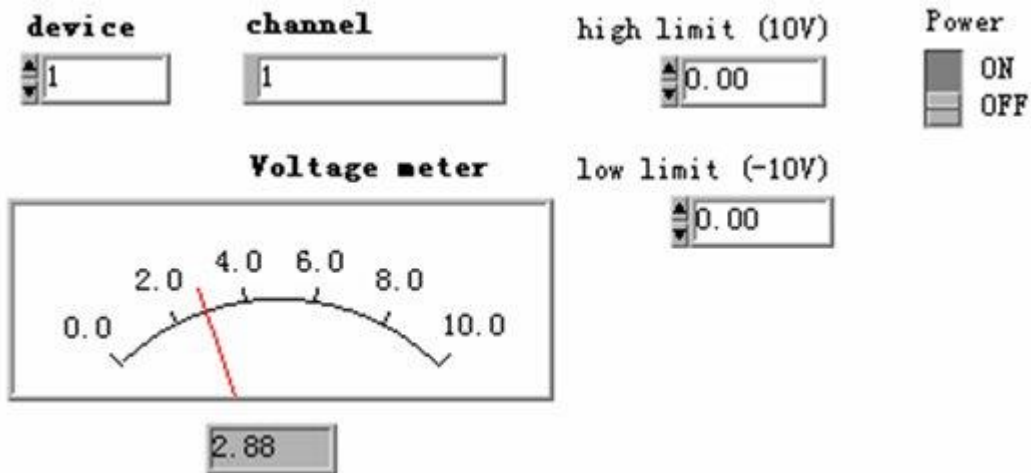
1. 按照下图建立框图程序。



对于循环 For Loop 要求每 500 毫秒执行一次，控制输出电压以 0.5 伏的间隔从 0 伏到 9.5 伏变化。当循环 For Loop 结束后，输出电压又复位为 0 伏。

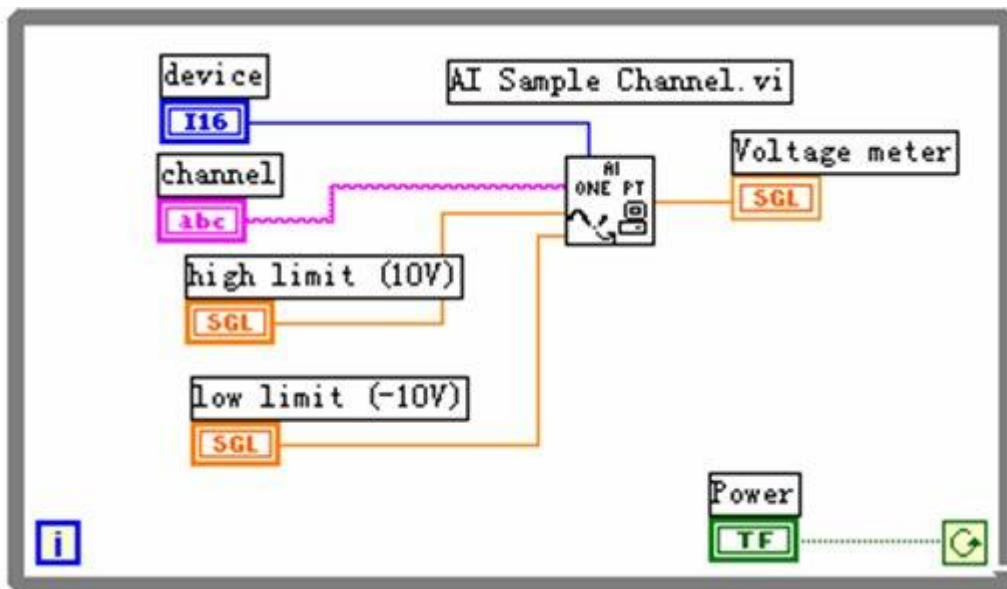
2. 关闭上述框图程序窗口，再打开前面板窗口，创建输入电压测量 Voltmeter 程序，以显示输入电压值。

前面板:



框图程序:

1. 按照下图编写程序。



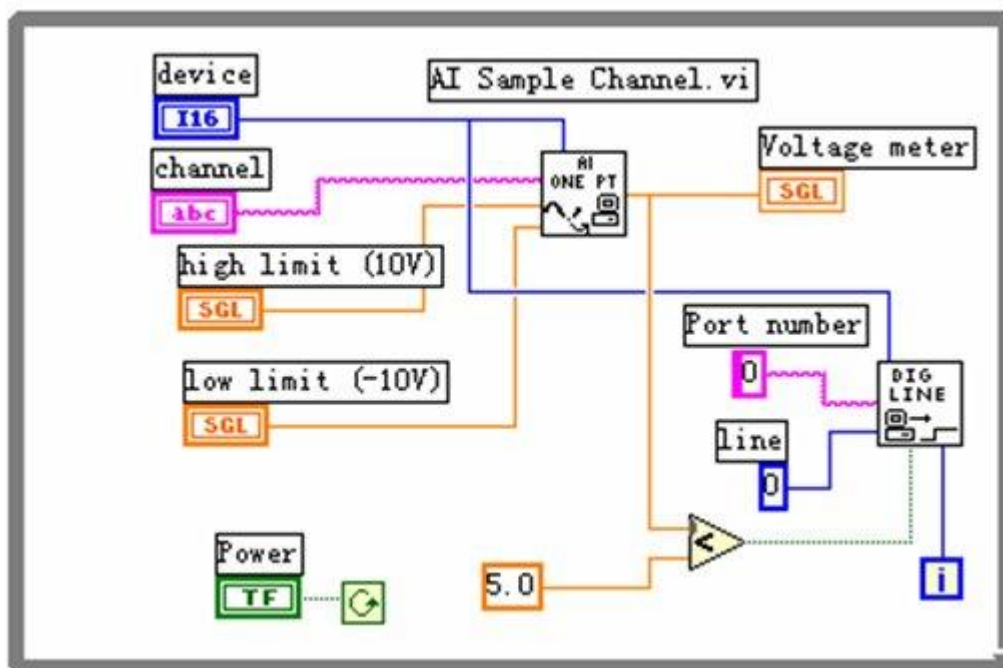
2. 在前面板的电压测量仪表盘，将其刻度定义为 0.0 至 10.0。

3. 在通道控制 Channel 输入值 1，上限控制 High Limit 输入值 10.0，下限控制 Low Limit 输入值 0.00，打开 Power 开关，运行该程序。

4. 如果运行过程中出现错误，将会弹出一个对话框，显示错误代码和解释。要模拟出错情况，在 Device 控制栏中输入 0 并运行程序，则将会显示出错情况。

5. 确认已在演示盒 Demo Box 中将模拟输出 CH0 接到模拟输入 CH1，运行电压输出 Voltage Output VI 程序，则在电压测量 Voltmeter 程序中将采集并显示电压值。

6. 停止程序运行，并按上图虚线框住的部分修改程序，当输入电压大于 5 伏时点亮 LED 灯。演示盒 Demo Box 上的 LED 灯是负逻辑，高电平时关闭 LED，低电平时打开 LED。



7. 再同时运行前面的电压输出程序和本电压输入程序，以验证运行效果。

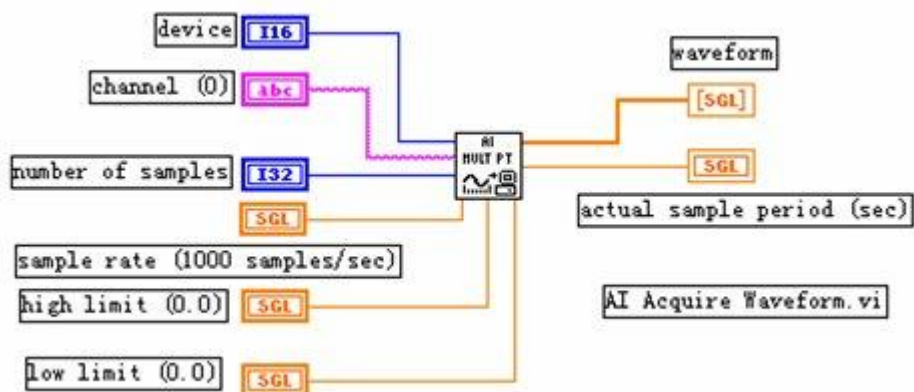
(练习 2-1 结束)。

第四节 波形的采集与产生

在许多应用场合，一次只采样一个数据点是不够的。另外，采样点之间的间隔很难保持恒定，因为它取决于很多因素，如循环的执行速度，子程序的调用时间等等。而使用 AI Acquire Waveform 和 AO Generate Waveform 程序，就可以以大于单点操作的速度进行多点的数据采集和波形产生，并且用户可以自定义采样速率。

波形采集：

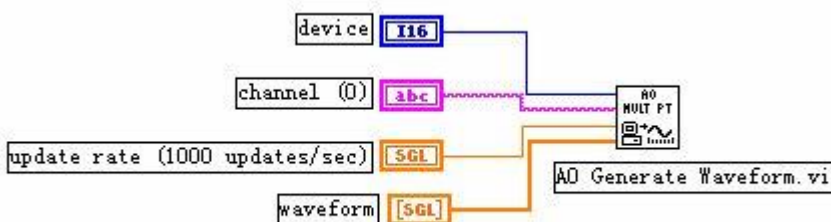
AI Acquire Waveform 程序从一个指定的输入通道按用户定义的采样率和采样点数采集数据并返回计算机。



上图中 Number of Samples 是采样点数，Sample rate 是采样率，以 Hz 为单位。Waveform 是模拟输入信号的一维采样数组，以伏特为单位。Actual sample period 是实际采样率的倒数，它可能与指定采样率有一些小偏差，偏差取决于计算机硬件速度。

波形产生：

AO Generate Waveform 程序在一个模拟输出通道上以用户定义的更新速率产生一个电压波形。Update rate 是每秒钟产生的电压数值更新点数。Waveform 是一个一维数组，它包含写到输出通道上的模拟电压值，以伏为单位。



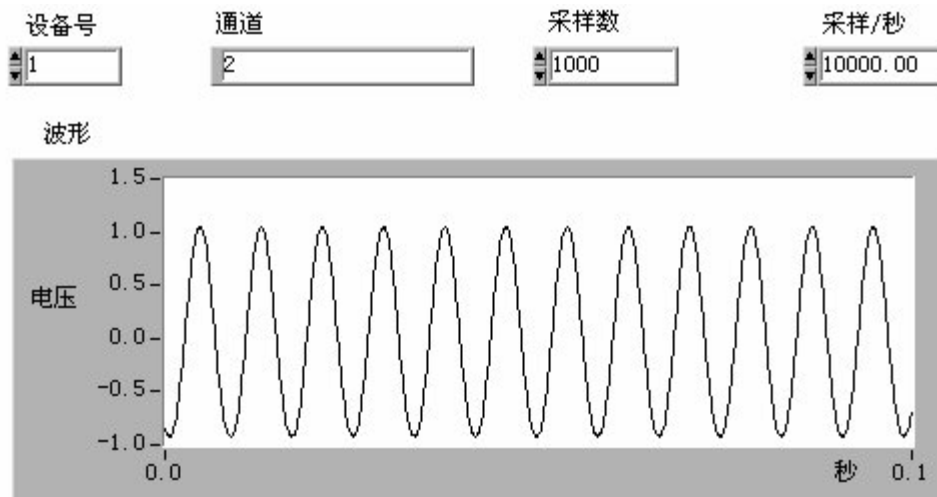
练习 2-2

目的： 采集并显示一个模拟信号波形。

学员将编写一个 VI 程序，它使用 DAQ 卡采集一个信号波形，并在图表上显示。在本例中，把演示盒上的模拟输入 CH1 接到函数发生器的正弦波输出端口上。

前面板：

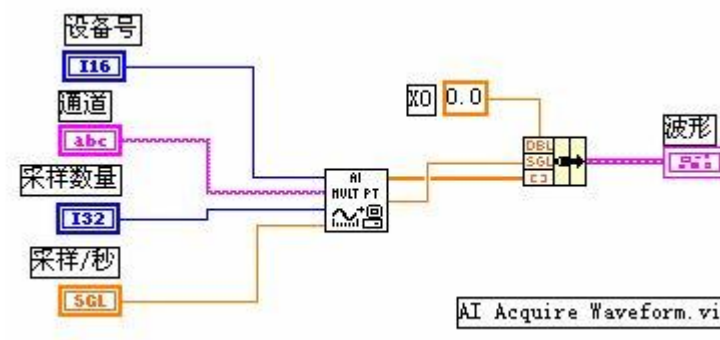
1. 打开一个新建前面板窗口，并照下图创建一个前面板程序。



“采样数”控制栏定义采样点数，而“采样/秒”控制栏定义采样率。

2. 切换到框图程序。

框图程序：



1. 按照上图创建框图程序。

2. 返回前面板窗口，输入各控制栏数值，并运行程序。图表窗口将绘出模拟信号波形。试用不同的采样率和采样点数，观察波形的差别。

(练习 2-2 结束)

第五节 扫描多个模拟输入通道

AI Acquire Waveforms 程序从多个输入通道以指定的采样率采集指定的采样点数，并将采样结果数据送回到计算机。Channels 控制栏指定要采样的多个通道的编号，各个通道号间以逗号隔开，例如 1, 2, 4。控制栏 Number of samples/ch 是每个通道要采集的采样点数。Scan rate 是每个通道每秒钟的采样点数即采样

率。Waveform 是一个二维数组，包含模拟输入信号电压数值，以伏为单位。Actual scan period 是实际采样率的倒数，由于计算机硬件的不同，实际采样率与指定的采样率可能有微小差异。

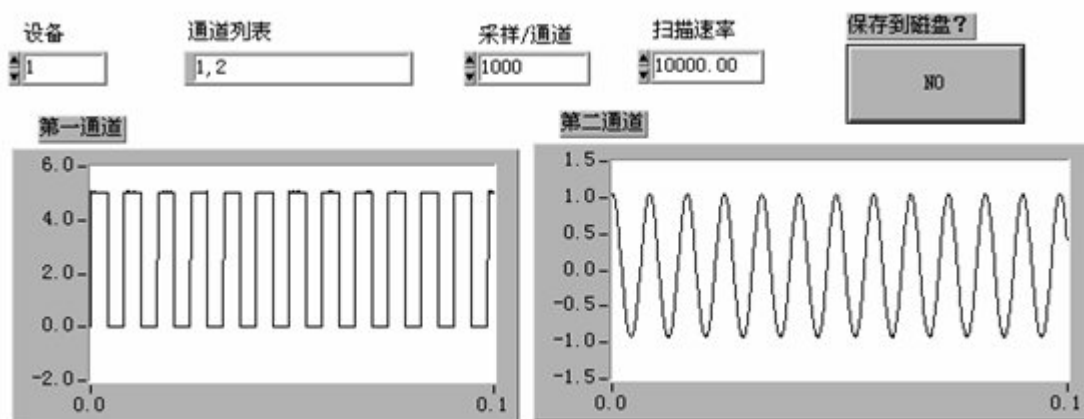
练习 2-3

目的：使用简单 I/O 程序来执行扫描多个通道的数据采集工作。

学员将创建一个 VI 程序，它扫描两个不同通道的模拟信号波形，并将它们分别在图表上绘出。在本例中，把 Demo Box 的模拟输入通道 CH1 接到正弦波/三角波输出口，而模拟输入通道 CH2 接到方波输出口。

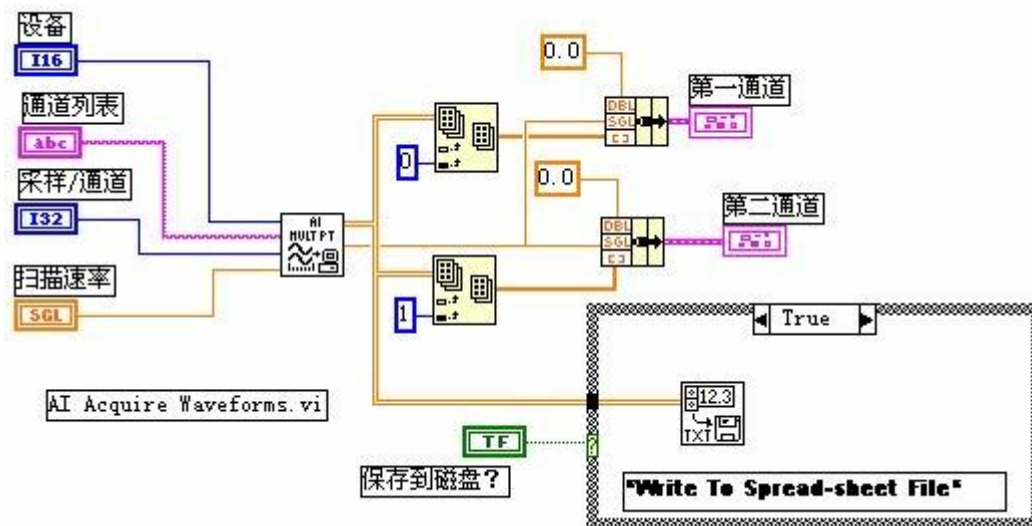
前面板：

1. 按照下图创建前面板程序，它有两个输入通道，可同时显示两个波形。



2. 切换到框图程序。

框图程序：



1. 创建如上图所示框图程序，它调用了如下的功能模块：

AI Acquire Waveforms 子程序（在 Data Acquisition>Analog Input 子模板）。在本例中，它在通道 1 和通道 2 以采样率 10000Hz 采样 1000 个点。

Index Array 子程序（在 Array 子模板），在本例中，把二维数组的第 0 列定义为通道 1 的采集数据，第 1 列定义为通道 2 的采集数据。

Write to Spreadsheet File 子程序（在 File I/O 子模板）。在本例中，把通道 1 和通道 2 的数据写入数据文件（spreadsheet file）。

2. 将演示盒上的模拟输入通道 1 与方波发生器连接，模拟通道 2 与正弦波发生器相连接。运行该程序，在图表上将显示输入的两个数据波形。

（练习 2-3 结束）。

第六节 连续数据采集

连续数据采集，或者说实时数据采集，是在不中断数据采集过程的情况下不断地向计算机返回采集数据。开始数据采集后，DAQ 卡不断地采集数据并将它们存贮在指定的缓冲区中（circular buffer），然后 LabVIEW 每隔一段时间将一批数据送入计算机进行处理。如果缓冲区放满了，DAQ 卡就会又重新从内存起始地址写入新数据，覆盖原来的数据。这个过程一直持续，直到采集到了指定数目的数据点，或者 LabVIEW 主动中止了采集过程，或者程序出现错误。这种工作方式对于需要把数据存入磁盘或者观察实时数据很有用。

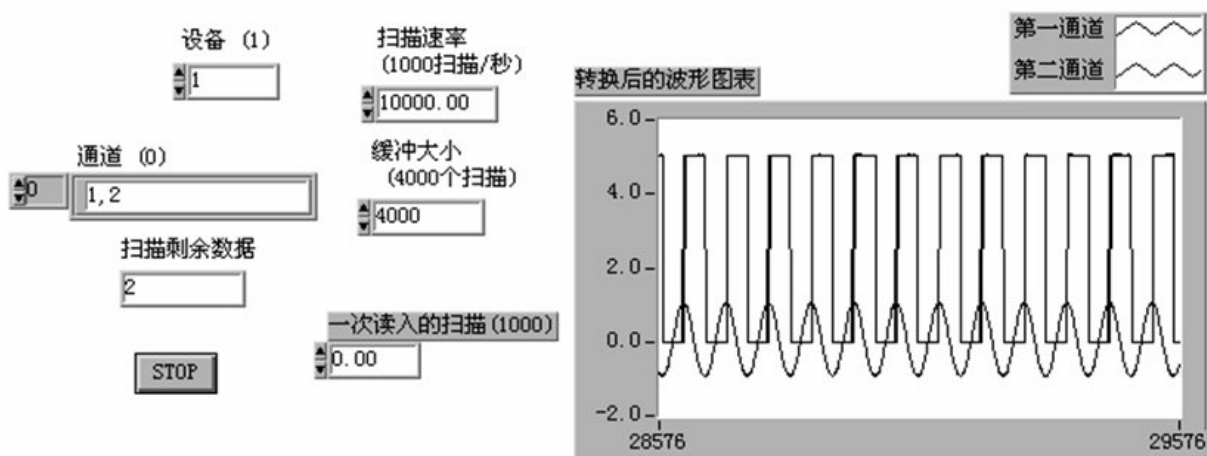
练习 2-4

目的：调用中级功能模块程序执行连续数据采集工作。

学员将验证一个连续数据采集程序，在本例中，把模拟输入通道 CH1 接到正弦/三角波输出口。

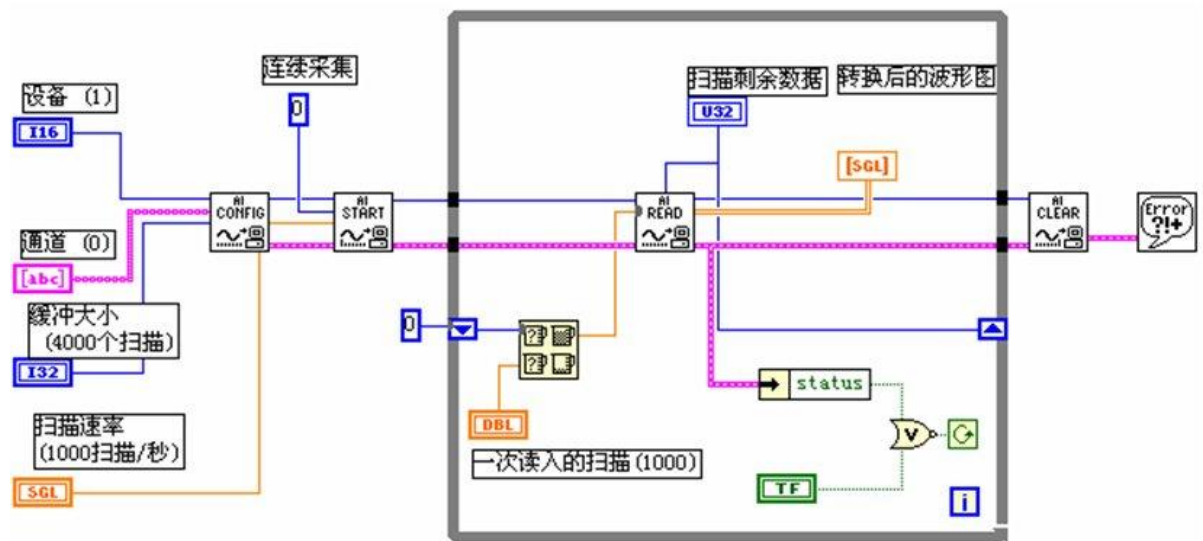
前面板：

1. 打开 Continuous Acquisition and Chart 程序。



2. 缓冲区的大小即数据存贮内存的大小用“缓冲大小”来指定，同时还用“一次读入的扫描”来指定每次从内存读取的数据量，用“扫描剩余数据”显示内存中尚未读取的采样数据个数。

框图程序：



1. 验证上图所示的框图程序。它调用了下面的功能模块：

AI Configure 程序（在 Data Acquisition>Analog Input 子模板）。设置设备编号。通道号和数据缓冲区大小。

AI Start 程序，用指定的采样率，以后台工作的方式开始连续数据采集过程。

AI Read 程序，在条件循环中连续采集指定数目的采样点，并返回数据到计算机。

AI Clear 程序，停止数据采集，释放所有资源。

2. 运行上述程序，观察结果。在本程序中，AI Read 子程序的每次读取的采样点数取决于设定值和内存中未读取的采样点数的比较结果，取它们中的较大者。

（练习 2-4 结束）。

第三课 仪器控制

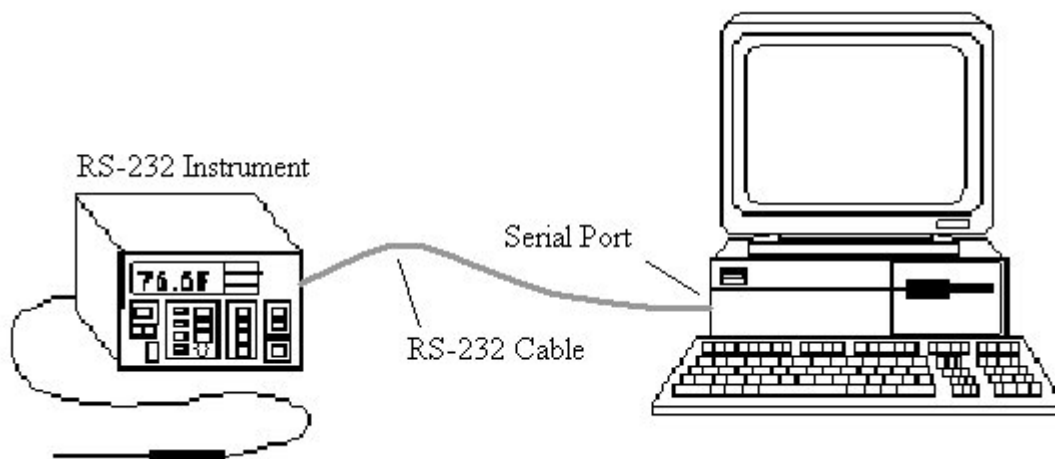
第一节 概述

本课程介绍使用 LabVIEW 来进行仪器控制的各种方法。要求学生学会串行 I/O、GPIB I/O 和 VISA I/O 的使用方法，同时也可以验证 LabVIEW 本身提供的仪器驱动程序。

本课程的实验设备要求一块已安装的 GPIB 卡，一台 GPIB 仪器以及 LabVIEW 开发系统。

第二节 串行通讯

串行通讯是一种常用的数据传输方法，它用于计算机与外设，例如一台可编程仪器，或者与另外一台计算机之间的通讯。串行通讯中发送方通过一条通讯线，一次一个字节，把数据传送到接收方。



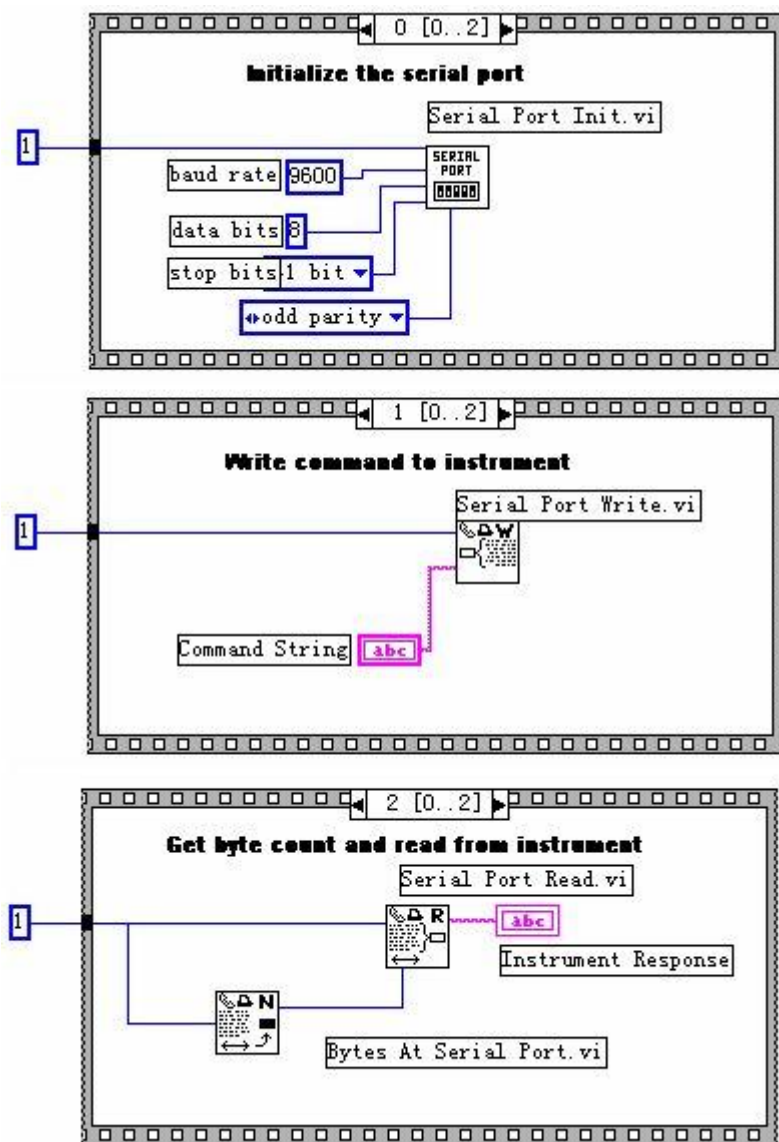
由于大多数电脑都有一至两个串行通讯接口，因此，串行通讯非常流行。许多 GPIB 仪器也都有串行接口。然而，串行通讯的缺陷是一个串行接口只能与一个设备进行通讯。

一些外设需要用特定字符来结束传送给它们的数据串。常用的结束字符是回车符、换行符或者分号。具体可以查阅设备使用手册以决定是否需要一个结束符。

在 LabVIEW 功能模板的 Instrument I/O>Serial 程序库中包含进行串行通讯操作的一些功能模块：

1. Serial Port Init VI 模块用于初始化所选择的串行口。Flow control 设置握手方式的参数。Buffer size 设置程序分配的输入/输出缓冲区的大小。Port number 决定通讯接口地址。Baud rate, data bits, stop bits 和 parity 等设置通讯参数。
2. Serial port write VI 模块把 String to write 中的数据写到 port number 指定的串行接口中。
3. Serial port read VI 模块从 Port number 指定的串行接口中读取 requested byte count 指定的字符个数。
4. Bytes at serial port VI 模块计算由 Port number 指定的串行接口的输入缓冲区中存放的字节个数，并将该数值存放于 Byte count 中。

在下面的实例中, 实现从一台串行仪器中读取测量值。首先, 用 Serial Port Init 模块初始化串行接口, 然后, 用 Serial Port Write 模块把命令参数发送给仪器, 接着用 Bytes at Serial Port 模块查明在串行输入缓冲区中已经读入的字节个数, 最后用 Serial Port Read 模块读取仪器数据。



第三节 IEEE 488 (GPIB) 概述

GPIB 程序库中包括 IEEE 488.2 应用程序和传统的 GPIB 应用程序。GPIB 488.2 应用程序中增加了 IEEE 488.2 兼容性, 具有 IEEE 488.2 的功能。然而, 在本课程中, 我们只讨论传统的 GPIB 应用程序。

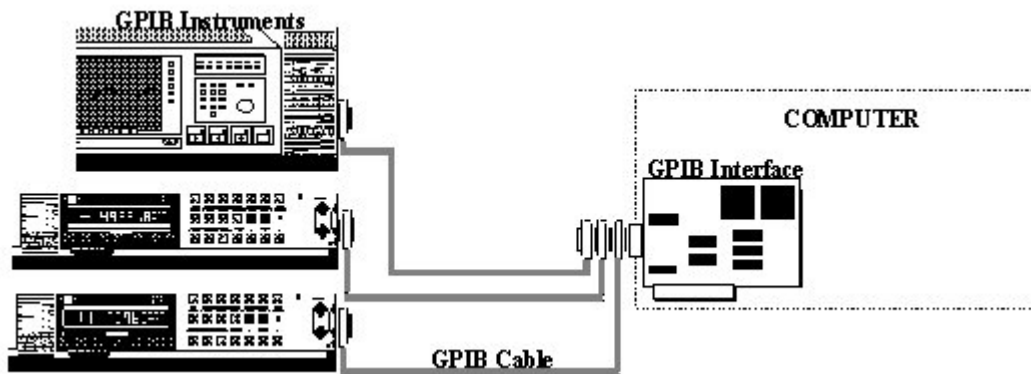
惠普公司在 60 年代末和 70 年代初开发了 GPIB 通用仪器控制接口总线标准。IEEE 国际组织在 1975 年对 GPIB 进行了标准化, 由此 GPIB 变成了 IEEE 488 标准。术语 GPIB, HP-IB 和 IEEE 488 都是同义词。GPIB 的原始目的是对测试仪器进行计

计算机控制。然而， GPIB 的用途十分广泛， 现在已广泛用于计算机与计算机之间的通讯， 以及对扫描仪和图像记录仪的控制。

GPIB 是一个数字化的 24 线并行总线。它包括 8 条数据线， 5 条控制线(ATN、 EOI、 IFC、 REN 和 SRQ)， 3 条握手线和 8 条地线。 GPIB 使用 8 位并行、 字节串行的异步通讯方式。也就是说， 所有字节都是通过总线顺序传送， 传送速度由最慢部分决定。由于 GPIB 的数据单位是字节（8 位）， 数据一般以 ASCII 码字符串方式传送。

有三种方式来标明传送数据结束。通常， GPIB 包括一根连接线（EOI）， 用来传送数据完毕信号。或者， 也可以在数据串结束处放入一个特定结束符（EOS）。有些仪器用 EOS 方法代替 EOI 信号线方法， 或者两种方法一起使用。还有一种方法， 听者（数据接收方）可以计数已传送的数据字节， 当达到限定的字节数时停止读取数据。只要 EOI， EOS 和限定字节数的逻辑“或”值为真， 数据传送就停止。一般字节计数法作为缺省的传送结束方法， 典型的字节数限定值等于或大于需要读取的数据值。

每个设备， 包括计算机接口卡， 必须有一个 0 到 30 之间的 GPIB 地址。一般 GPIB 接口板设置为地址 0， 仪器的 GPIB 地址从 1 到 30。 GPIB 有一个控者（你的电脑）来控制总线。在总线上传送仪器命令和数据， 控者寻址一个讲者， 一个或者多个听者。数据串在总线上从讲者向听者传送。 LabVIEW 的 GPIB 程序包自动处理寻址和大多数其他的总线管理功能。

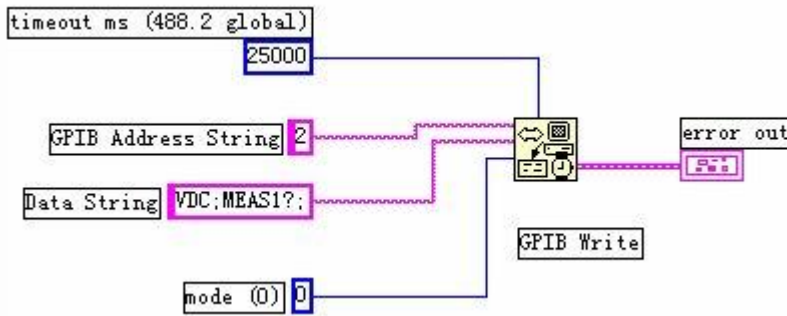


GPIB 软件包：

LabVIEW 在 Instrument I/O 功能模板的 GPIB 和 GPIB 488.2 子模板下有许多 GPIB 通讯功能子程序模块， 这些模块在工作平台上可以调用低层的 488.2 驱动程序。大多数的 GPIB 应用程序只需要从仪器读写数据串。下面讨论传统的 GPIB Write/Read 子程序模块。

GPIB Write 模块把 data string 中的数据写入 address string 指定的设备中。 Mode 指定如何结束 GPIB 写入过程， 如果在 timeout ms 指定的时间内操作未能完成， 则放弃此次操作。 Error In 和 Error Out 字符串与出错处理程序配合使

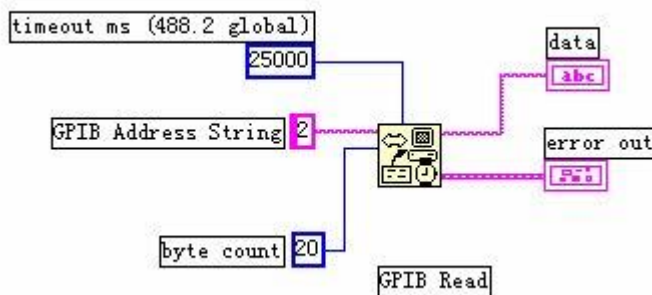
用，检测可能的出错情况。Status 是 16 位的布尔逻辑数组，每个元素代表 GPIB 控者的一种状态，详见本课程的稍后部分。



在上图程序中，GPIB WRITE 模块把“VDC; MEAS1? ;”字符串写入地址=2 的 GPIB 设备中，本实例采用缺省值 MODE=0，timeout ms=25000。GPIB READ 模块从 address string 指定地址的 GPIB 设备中读取由 byte count 指定的字节数，用户可以使用 Mode 参数指定结束读取的条件，与 byte count 一起使用。读取的数据由 data string 返回。

用户必须把读取的字符串转换成数值数据，才能进行数据处理，例如进行曲线显示。Error In 和 Error Out 是出错指示数簇。

GPIB READ 模块遇到下列情况之一则中止读取数据：（1）程序已经读取了所要求的字节数。（2）程序检测到一个错误。（3）程序操作超出时限。（4）程序检测到结束信息（由 EOI 发出）。（5）程序检测到结束字符 EOS。



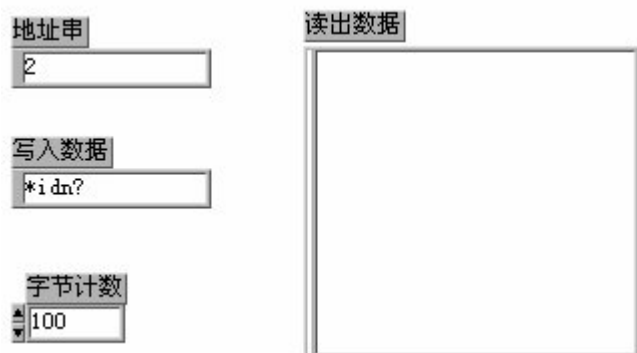
如图所示的程序中，GPIB Read 从地址=2 的设备中读取 20 个字节的数据。该程序使用了缺省值 MODE=0，TIMEOUT MS=25000。在本例中，如果读够了 20 个字节，或者检测到 EOI，或者超出 25000ms 时间，读取过程将结束。

练习 3-1

目的：使用 GPIB 子程序模块与 GPIB 设备通讯。

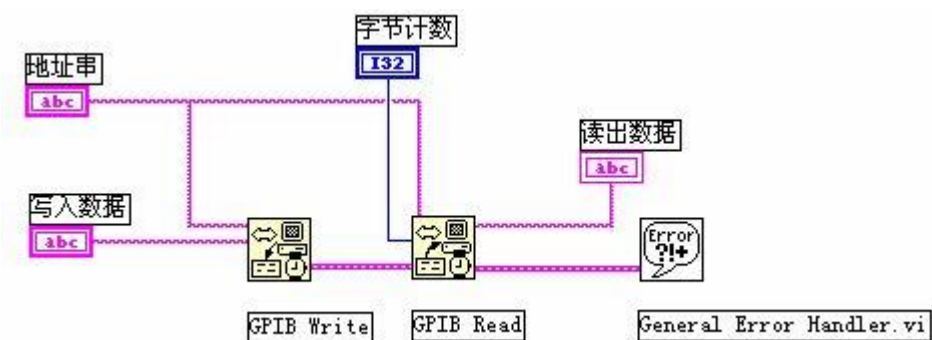
学员可以建立一个与任何 GPIB 仪器通讯的程序。本例采用传统的 GPIB 子程序与指定仪器进行 GPIB 读写操作。命令行参数 “*i dn?” 适用于大多数 IEEE 488.2 兼容仪器，它要求仪器返回其标识符。

前面板：



- 1、从 FILE 菜单中选择 NEW 打开一个新面板。
- 2、建立上图所示的控制和显示（请记住，从程序框图中，选择相应的功能模块，再选择 Create Control 或者 Create Indicator, 可以产生所有的控制和显示）。
- 3、从“写入数据”控制栏中输入“*i dn?”，在“字节计数”中输入数值 100。

程序框图：



- 1、打开框图编程窗口。
 - 2、创建如上图所示的框图子程序模块。
- GPIB Write 功能模块（在 Instrument I/O::GPIB 子模板）。该模块把字符串写入 GPIB 仪器。

GPIB Read 功能模块（在 Instrument I/O: : GPIB 子模板）。该模块从 GPIB 仪器中读取数据字符串。

General Error Handler 功能模块（在 TIME & DIALOG 子模板）。该模块检查出错报告字符串，如果发现错误，则显示一个对话框。

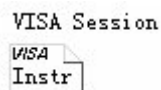
3、返回到前面板，并运行该程序。在“读出数据”显示栏中将显示仪器的标识字符串，如果无数据返回，而接收到一个 GPIB 出错信息，EABO，或者 error=6，这就表示你的仪器不能读命令参数“*i dn?”。请查阅仪器的使用手册以找出合适的相应命令。

4、把上述程序以 GPIB.vi 的名字保存，然后关闭。

（练习 3-1 结束）。

第四节 VISA 编程

VISA 是虚拟仪器软件结构体系（Virtual Instrument Software Architecture）的简称。VISA 是在所有 LabVIEW 工作平台上控制 VXI、GPIB、RS-232 以及其他种类仪器的单接口程序库。VISA 是由组成 VXI plug&play 系统联盟的 35 家最大的仪器仪表公司所统一采用的标准。采用了 VISA 标准，就可以不考虑时间及仪器 I/O 选择项，驱动软件可以相互相容使用。VISA 包含的功能模块在 Instrument I/O>VISA 子模板中。大多数的 VISA 功能模块使用了 VISA session 参数，该参数在 Control 模板的 Path and Refnum 子模板中。



VISA session 是每次程序操作过程的唯一逻辑标识符。它标识了与之通讯的设备名称以及进行 I/O 操作必需的配置信息。它由 VISA Open 功能模块产生，提供给 VISA 主功能模块使用。VISA Open 功能模块产生标识信息，然后把它传送给下一个 VISA 功能模块。这样简化了数据流编程。它相似于文件 I/O 模块的参考名功能。

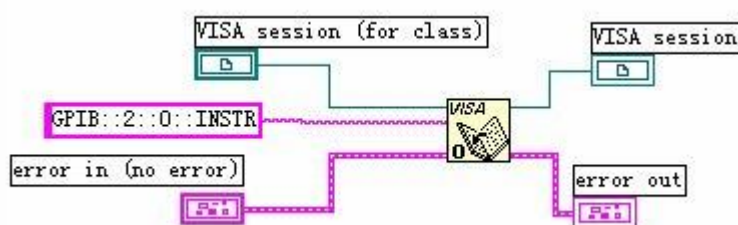
VISA session 的缺省值是 Instr。如果需要，可以打开 VISA session 选择如下值：Instr、GPIB Instr、Serial Instr 和 VISA/GPIB-VXI RBD Instr。下面介绍常用的 VISA 功能模块：VISA Open、VISA Write、VISA Read 和 VISA Close。

VISA Open 根据 Resource Name 和 VISA session 与指定的设备建立通讯。模块返回 VISA session 标识值，使用该标识值就可以调用此设备的任何其他的功能。Error in 和 Error out 字符串包含出错信息。

Resource Name 包含 I/O 接口类型以及设备地址等信息。其编程语法如下表所示：

接口	程序语句
SERIAL	ASRL[board][:: INSTR]
GPIB	GPIB[board]::primary address [::secondary address][:: INSTR]
VXI	VXI [board]::VXI logical address [:: INSTR]
GPIB-VXI	GPIB-VXI [board][::GPIB-VXI primary address]::VXI logical address [:: INSTR]

关键词 GPIB 用于同 GPIB 设备建立通讯，关键词 VXI 通过嵌入式或 MXI 总线控制器同 VXI 仪器建立通讯，关键词 GPIB-VXI 用于 GPIB-VXI 控制器，关键词 SERIAL 用于异步串行设备通讯。



在上例中，VISA open 模块通过仪器描述字符串“GPIB::2::0::INSTR”与主地址=2 的 GPIB 设备建立通讯。

VISA Write 模块把 write buffer 中的字符串写入 VISA session 指定的设备。dup VISA session 向下传送相同的 session 值。在 UNIX 工作平台上，数据同步写入；在其他工作平台上，数据异步写入。return count 返回实际传送的字节数。error in 和 error out 字符串包含出错状况。

VISA Read 读取由 VISA session 指定设备中的数据。byte count 指明读入 read buffer 中的字节数。dup VISA session 向下传送相同的 session 值。在 UNIX 工作平台上，数据同步读入；在其他工作平台上的数据异步读入。return count 返回实际传送的字节数。error in 和 error out 字符串包含出错状况。

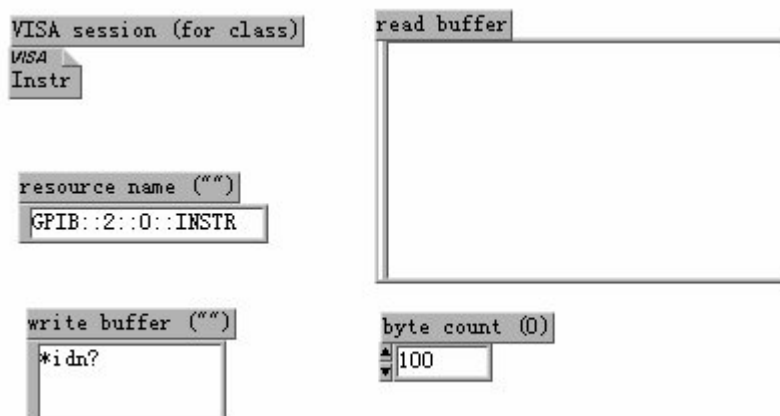
VISA Close 关闭由 VISA session 指定设备的通讯过程，释放系统资源，error in 和 error out 字符串包含出错状况。

练习 3-2

目的：用 VISA 模块与 GPIB 设备或者串行设备通讯。

本练习使用 VISA 功能模块向指定设备（GPIB 或者串行设备）读写数据。命令参数 “*idn?” 适用于大多数仪器，无论是 GPIB 通讯或者串行通讯。它返回仪器的标识字符串。本课中所有 VISA 练习都是根据 Fluke 45 万用表编程。

前面板：



1. 打开一个新的前面板，并且照上图建立控制和指示件。VISA session 控制件可以在 CONTROLS 模板中的 Path&Refnum 子模板中找到。也可以通过 VISA Open 功能模块创建。

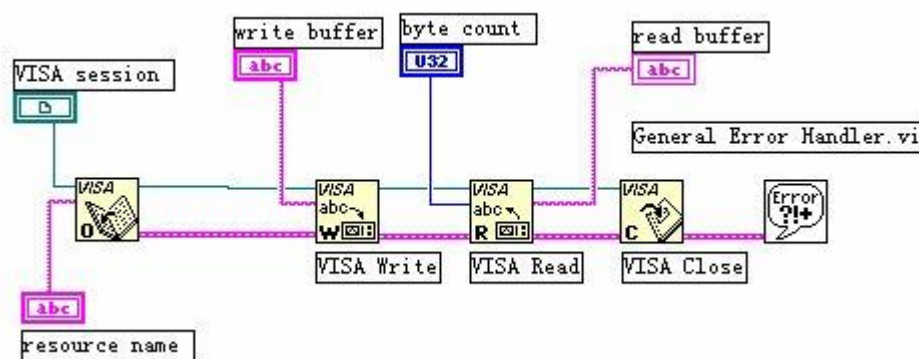
2. Resource Name 控制件应包含下列值：

*对于地址=2 的 GPIB 仪器：“GPIB::2::0::INSTR”

*对于 COM1 上的串行仪器：“ASRL 1::INSTR”

3. 在 Write Buffer 字符串控制栏中输入 “*idn?”，在 Byte Count 控制栏中输入 100。

程序框图：



1. 打开框图窗口。
2. 按照上图选择创建框图对象并连接线。调用了下面的模块：

VISA Open 功能模块（在 Instrument I/O::VISA 子模板）。此模块打开通讯过程，并产生 VISA session 参数。

VISA Write 功能模块（在 Instrument I/O::VISA 子模板）。此模块把数据串写入指定设备。

VISA Read 功能模块（在 Instrument I/O::VISA 子模板）。此模块从指定设备中读入数据。

VISA Close 功能模块（在 Instrument I/O::VISA 子模板）。此模块关闭 VISA Session。

3. 返回前面板并运行该程序。VISA session 控制件保持为设置值 INSTR。如果要修改此参数，可以打开 VISA session 控制件，选择 VISA Class 值。根据 Resource name 的设置值，可以选择与 GPIB 或者串行设备进行通讯。

4. 以 VISA.vi 文件名保存该程序，并关闭程序。

（练习 3-2 结束）

第五节 用 LabVIEW 编写仪器驱动程序

仪器的驱动软件是专门控制某种仪器的软件。LabVIEW 因为具有面板控制的概念，特别适合于创建仪器的驱动程序。软件的前面板部分可以模拟仪器的前面板操作。软件的框图部分可以传送前面板指定的命令参数到仪器以执行相应的操作。当建立了一个仪器的驱动程序后，就不必再记住仪器的控制命令，而只要从前面板输入简单数据即可。仅仅拥有控制单台仪器的软件，意义并不大。其真正意义在于可以把仪器驱动程序作为子程序调用，与其他子程序一道组成一个大控制程序，从而控制整个系统。

仪器驱动程序模板：

在 LabVIEW>EXAMPLES>INSTR>INSTTMPL.LLB 程序库中，有许多 VISA 仪器驱动程序模板程序。这些模板程序是适用于大多数仪器的驱动程序，并且是 LabVIEW 仪器驱动程序开发的基础。这些模板程序符合仪器驱动程序的标准，并且每个程序都有指导帮助指令以便修改程序以适应某种仪器。

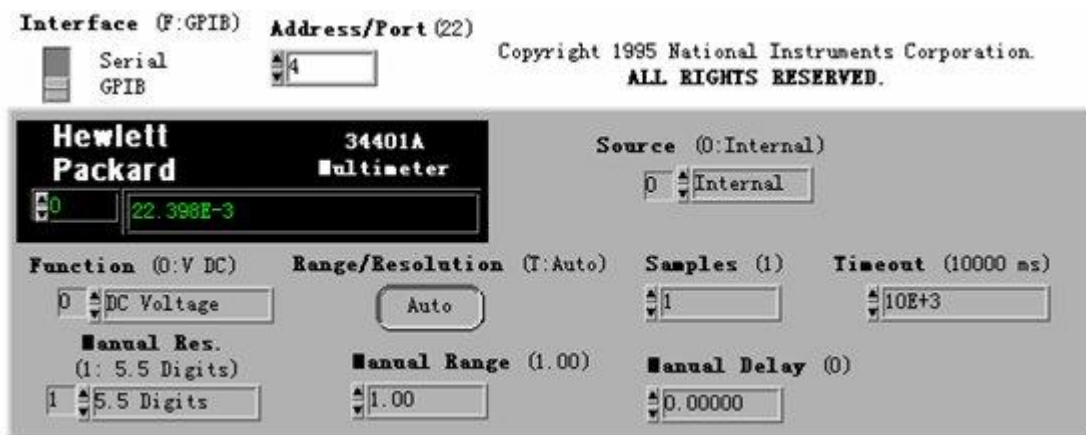
第六节 验证仪器驱动软件

下面验证 HP 34401A 万用表驱动软件。如果有一只 HP 34401A 万用表，就可以运行该程序。否则只能学习程序设计方法。

从 LabVIEW>EXAMPLES>INSTR>HP34401A.LLB 中打开 HP34401A Getting Started VI 程序。

该程序是一个仪表驱动应用程序。其前面板模仿仪表的前面板，从 Help 菜单中的 Show Help 可以打开帮助窗口，然后把光标指向程序前面板的各个控制件或指示件，就会出现前面板上鼠标所指对象的描述信息。当编写或使用这些仪器驱动程序时，档案资料是非常重要的，这些程序可以用来设置仪器参数和读取仪器数据。本程序是采用低层的仪器驱动模块子程序编写的。这些低层子程序用到了前面我们所学习的 VISA 功能子模块。

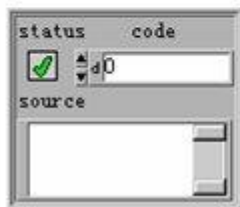
前面板：



WARNINGS:

- (1) You **MUST** have VISA Version 2.0 or higher installed to use this Instrument driver.
- (2) If using the Serial Interface, the pass/fail outputs - Pins 1 & 9 - **MUST NOT** be enabled on the serial connector, or instrument damage may result - See the Manual for details.

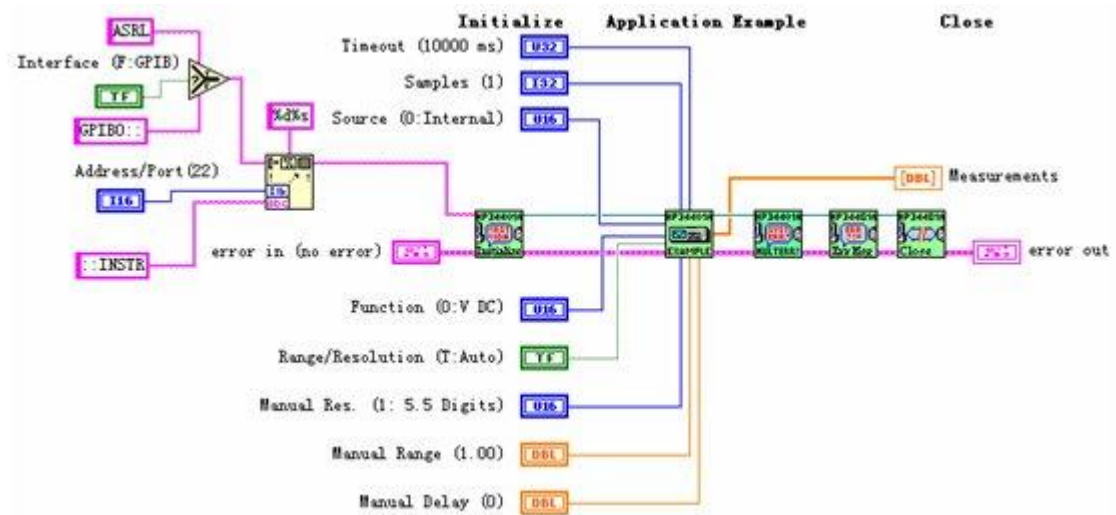
error in (no error)



error out



框图程序：



上述框图程序使用了如下的子程序：

HP34401A Initialize VI 子程序（LabVIEW>EXAMPLES>INSTR> HP34401A.LLB）。这个子程序用于与仪器建立通讯并产生一个 VISA session 标识字串。

HP34401A Application Example VI 子程序（LabVIEW>EXAMPLES>INSTR> HP34401A.LLB）。这个子程序关闭 VISA session 过程。

- 请注意上述程序中各个子程序之间的连线关系，以及 error in 和 error out 群的连接。通过这些连线，定义了数据流动的方式。

练习 3-3

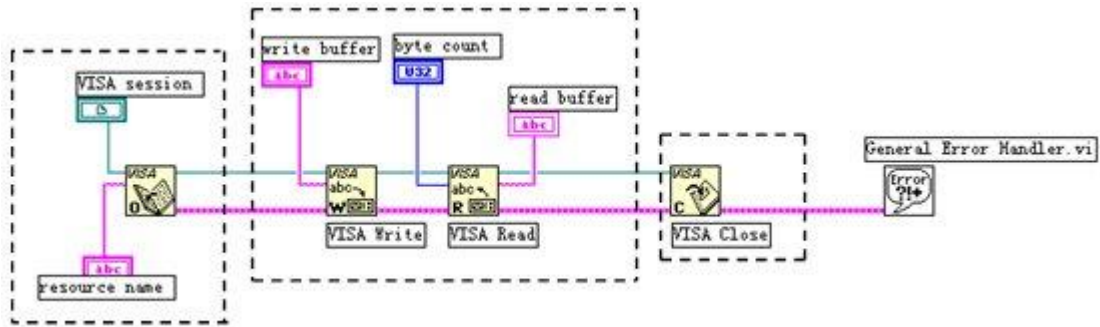
目的：在练习 3-2 的基础上创建一个简单的仪器驱动程序。

在本练习中，学员将学会创建仪器驱动程序的初始化、应用实例和关闭子程序模块。这些子程序已在前面讨论过。

前面板：

1. 从练习 3-2 中打开 VISA.VI 文件。
2. 本练习的前面板不需要修改。

框图程序：



1. 转换到框图程序窗口。

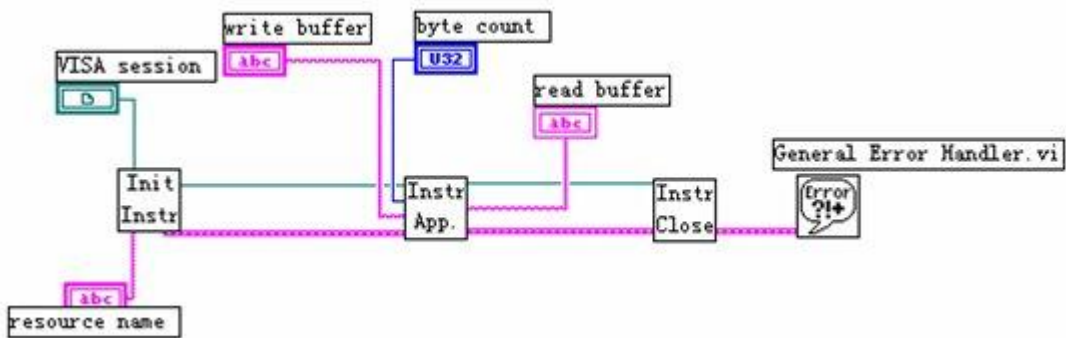
如图所示，选择第一个虚线方框部分，然后从 EDIT 菜单下调用 SubVI From Selecti on 功能。这个操作将根据所选择的部分创建一个子程序，有些程序对象排列可能会弄乱，所以需要重新整理框图。

再选择上图的第二个虚线方框部分，用 SubVI From Selecti on 功能创建一个子程序。

最后，再选择上图的第三个虚线方框部分，用 SubVI From Selecti on 功能创建一个子程序。

2. 当创建了上述所有仪器驱动子程序后，双点上述每个子程序图标，在 Seminar.IIb 目录下把它们分别保存为 Instrument Initialize.vi, Instrument Application.vi, 和 Instrument Close.vi 子程序。

- 注意：如果你想改变这些子程序的图标，则必须进入位于前面板窗口右上角的 Icon Edi tor 方式。当你要选择 Icon Edi tor 时，点击图标面板。



3. 完成了创建所有子程序后，把它们存入 Seminar.IIb, 并且修改缺省的图标形式。使框图程序如上图所示：

4. 返回前面板窗口，在 Resource Name 字符串中输入“GPIB::2::0:: INSTR”，运行此程序。如果你的仪器的地址不是 2，则把地址换成相应的数值。同样，要

仪器执行你给出的命令，只需在 Write Buffer 内输入相应的数值。同样，要仪器执行你给出的命令，只需在 Write Buffer 内输入相应的命令字串。

5. 把该程序以 Instrument Driver.vi 文件名保存，并关闭程序。

(练习 3-3 结束)

第四课 分析软件

第一节 概述

LabVIEW 的 Advanced Analysis 软件库包括数值分析、信号处理、曲线拟合以及其他软件分析功能。该软件库是建立虚拟仪器系统的重要工具，除了具有数学处理功能外，还具有专为仪器工业设计的独特的信号处理与测量功能。

除了 Advanced Analysis 软件库，NI 公司还提供一些附加的分析工具库，借助这些分析软件包，LabVIEW 可以具有更加强大的分析功能。这些分析工具库包括：

- 联合时频分析(Joint Time Frequency Analysis)工具箱：用于分析常规傅里叶变换不易处理的时-频特性。
- G Math 工具箱：提供了扩展的数学功能，如公式分析、求根植、画轮廓线等。
- 数字滤波器设计工具箱

详细的情况请查阅相关资料。

在本课程中，用户可以学习如何建立一个简单实用的频谱分析仪，如何使用数字滤波器，如何执行简单的曲线拟合功能等。本课程需要使用 LabVIEW 完全开发系统。

第二节 高级分析功能程序

在 LABVIEW 5.0 以前的版本，高级分析程序库被安排在函数模板的一个名为 Advanced Analysis 的子模板中。而现在流行的 LabVIEW 5.1 版则把它分成两个子模板。

1) Signal processing 子模板。其中包括：

- 信号发生。
- 时域信号。
- 频域信号。

- 测量。
- 数字滤波器。
- 窗函数。

2) Mathematics 子模板。其中包括:

- 公式框节点。
- 1D and 2D Evaluation。
- 微积分 (Calculus) 。
- 概率和统计。
- 曲线拟合。
- 线性代数。
- Array Operations。
- 优化处理。
- Zeroes。
- 数字函数。

第三节 信号产生

该软件库可以为网络分析或仿真产生多种常用信号。它可以和 NI 公司的 DAQ 卡一起使用，以产生模拟输出信号。

在模拟电路范围，信号频率以 Hz 或周期来测量，但在数字系统中我们使用数字频率，它是模拟频率与采样频率之比，如下所示：

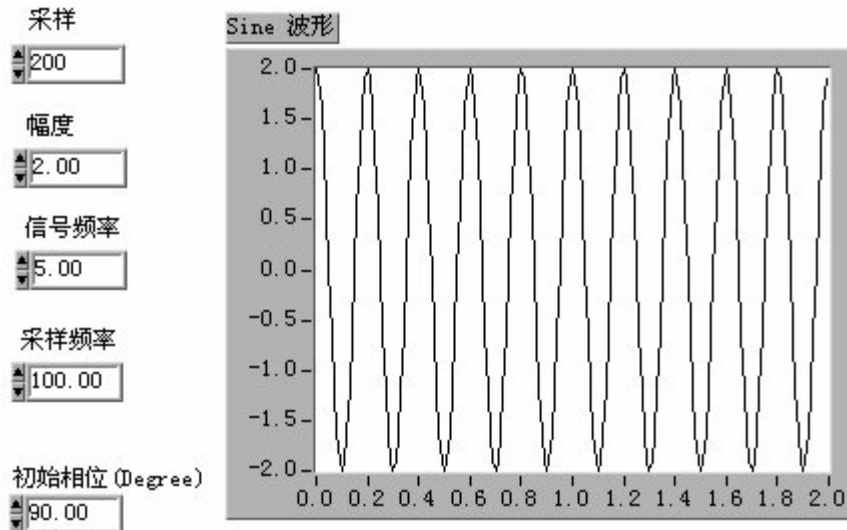
数字频率=模拟频率/采样频率

在许多信号产生子程序模块中，我们使用数字频率，因为难以确定采样频率。为了得到模拟频率输出，我们必须确定采样频率。如下面的练习所示：

练习 4-1

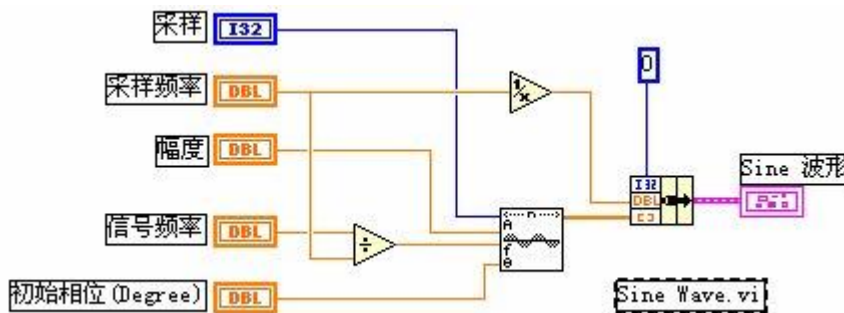
目的：产生一个指定频率的正弦波。

前面板：



1. 打开 Generate Sine.vi 程序。
2. 前面板包含下列控制栏：采样点数、幅值、信号频率、初始相位、采样频率。
3. 转入框图程序部分。

框图程序：



1. 验证上述框图程序，它使用了下面的子程序：

Sine Wave VI 子程序（在 Analysis>Signal Generation 子模板）。在本例中，它以采样频率 100Hz，产生有 200 个采样点的 5Hz 正弦波。

2. 注意查看框图，可以看到在连接到 Sine Wave.VI 程序之前，信号频率已经被采样频率相除。也就是说，Sine Wave 需要输入信号的数字频率。
3. 运行此程序，在前面板缺省值的情况下，将出现 5Hz 的正弦波。

采样与混频 (Aliasing)：

在前面板把信号频率改为 90Hz，再运行此程序。结果出现的信号频率却等于 10Hz。

这种现象叫混频偏差 (Aliasing)，只在数字频率范围出现。著名的耐奎斯特采样理论已经说明，最高信号稳定频率等于采样频率的一半。在本例中，采样频率等于 100Hz，所以最高信号频率为 50Hz，如果输入频率大于 50Hz，如本例中的 90Hz，它将会偏差到 $(N \times 50) - 90$ Hz > 0，即为 $(100 - 90) = 10$ Hz，也就是说，采样频率为 100Hz 的数字系统不能区分 10Hz 和 90Hz、20Hz 和 80Hz、51Hz 和 49Hz 等等。

因此，在设计数字频谱系统时，我们必须保证不要让大于 1/2 采样频率的信号进入系统。一旦进入了，就没有办法清除它们。为了防止偏差出现，我们一般采用低通滤波器。在本例中，我们可以使用抗混频模拟低通滤波器滤除任何大于 50Hz 的信号。加了滤波器以后，当采样频率为 100Hz 的系统内出现 10Hz 信号时，我们就可以肯定它是 10Hz 而非 90Hz。

(练习 4-1 结束)

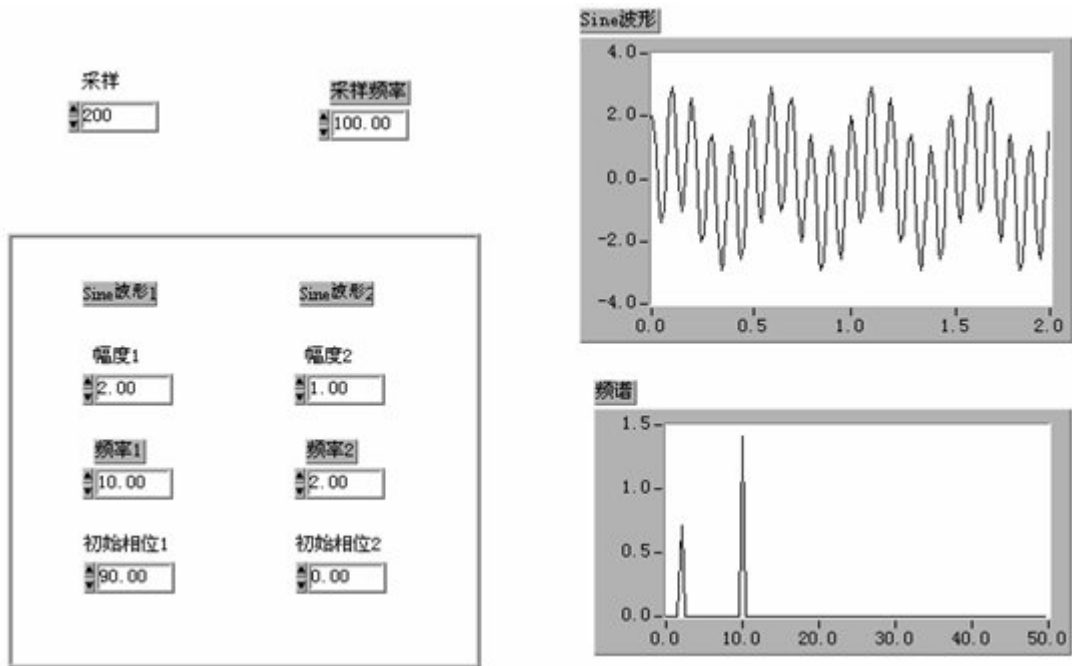
第四节 信号处理

在许多应用场合，需要计算信号的频谱，Advanced Analysis 程序库有许多这方面的功能模块程序。下面的例子讲述使用 Amplitude and Phase Spectrum VI 子程序来测量两个频率分量。

练习 4-2

目的：计算一个信号的频谱分量。

前面板：



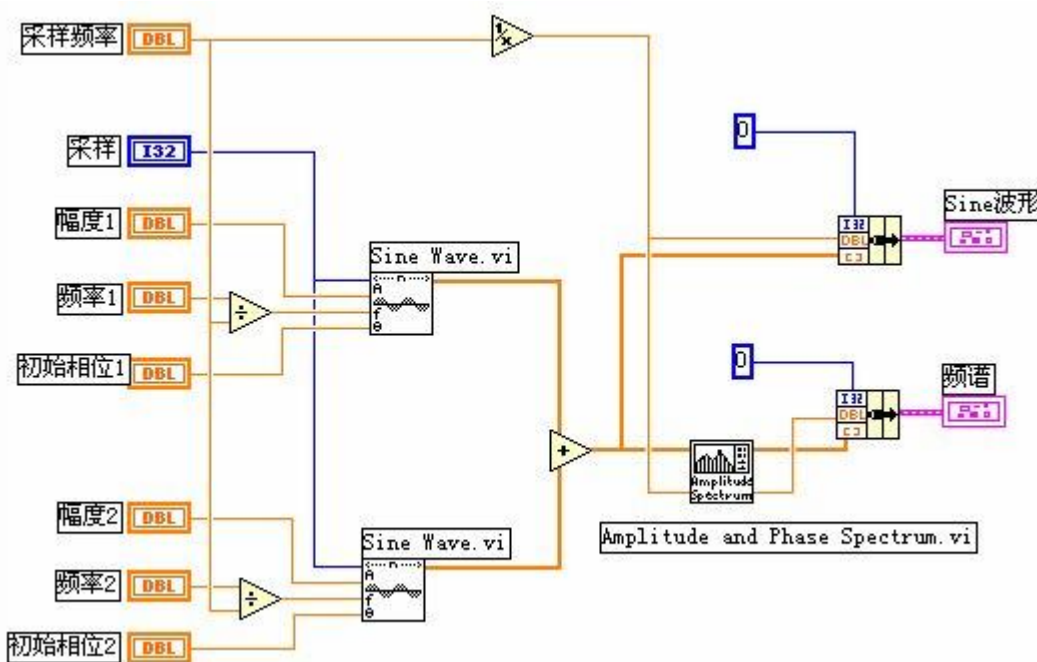
1. 打开 Compute Frequency Spectrum VI 程序。输入信号混合了两个正弦波，一个为 2Hz，另一个为 10Hz。采样频率为 100Hz，采样点为 200 个 10Hz 的信号幅值为 2 伏。2Hz 的信号幅值为 1 伏。

2. 切换到框图程序。

框图程序：

1. 框图程序如下次图所示，它使用了下面的子程序：

Amplitude and Phase Spectrum VI 子程序（在 Analysis>Measurement 子模板）。在本例中，它计算信号的幅度有效值（RMS）。



2. 运行程序。频谱图显示两个峰值，一个在 2Hz，另一个在 10Hz。2Hz 波形电压有效值为 0.717 伏，10Hz 波形电压有效值为 1.414 伏，分别对应其峰值 1 伏和 2 伏。

(练习 4-2 结束)

第五节 数字滤波器

数字滤波器用于改变或消除不需要的波形。它是应用最广泛的信号处理工具之一。两种数字滤波器分别是：FIR（有限脉冲响应）和 IIR（无限脉冲响应）滤波器。FIR 滤波器可以看成一般移动平均值，它也可以被设计成线性相位滤波器。IIR 滤波器有很好的幅值响应，但是非线性相位响应。

带通、带阻与过渡带宽

带通指的是滤波器的某一设定的频率范围，在这个频率范围的波形可以以最小的失真通过滤波器。通常，这个带通范围内的波形幅度既不增大也不缩小，我们称它为单位增益 (0dB)。**带阻**指的是滤波器使某一频率范围的波形不能通过。理想情况下，数字滤波器有单位增益的带通，完全不能通过的带阻，并且从带通到带阻的**过渡带宽**为零。在实际情况下，则不能满足上述条件。特别是从带通到带阻总有一个过渡过程，在一些情况下，使用者应精确说明过渡带宽。

带通纹波和带阻衰减

在有些应用场合，在带通范围内放大系数不等于单位增益是允许的。这种带通范围内的增益变化叫作带通纹波。另一方面，带阻衰减也不可能是无穷大，我们必须定义一个满意值。带通纹波和带阻衰减都是以分贝（dB）为单位，定义如下：

$$dB = 20 \times \log \left(\frac{A_o(f)}{A_i(f)} \right)$$

其中， $A_o(f)$ 和 $A_i(f)$ 是某个频率等于 f 的信号进出滤波器的幅度值。

例如，假设带通纹波为 -0.02dB ，则有：

$$-0.02 = 20 \times \log \left(\frac{A_o(f)}{A_i(f)} \right)$$

$$\frac{A_o(f)}{A_i(f)} = 10^{-0.001} = 0.9977$$

可以看到，输入/输出波形幅度是几乎相同的。

假设带阻衰减等于 -60dB ，则有：

$$-60 = 20 \times \log \left(\frac{A_o(f)}{A_i(f)} \right)$$

$$\frac{A_o(f)}{A_i(f)} = 10^{-3} = 0.001$$

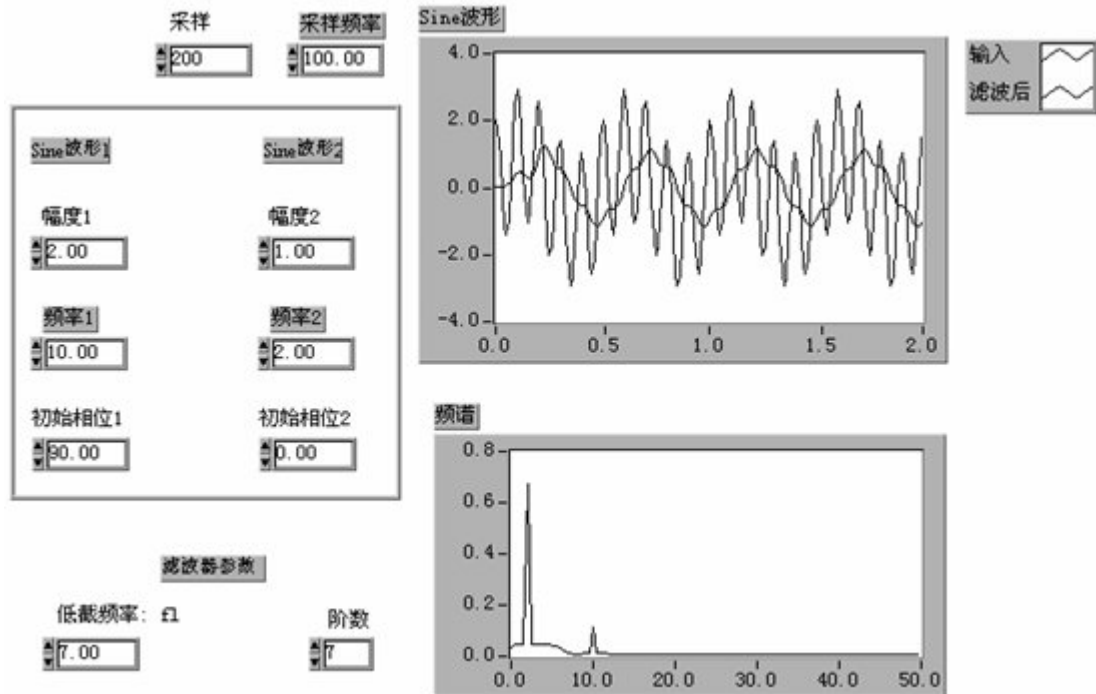
输出幅值仅是输入幅值的千分之一。

衰减值用分贝表示时经常不加负号，我们已经设定它为负值。

练习 4-3

目的：用数字滤波器消除不需要的频率分量

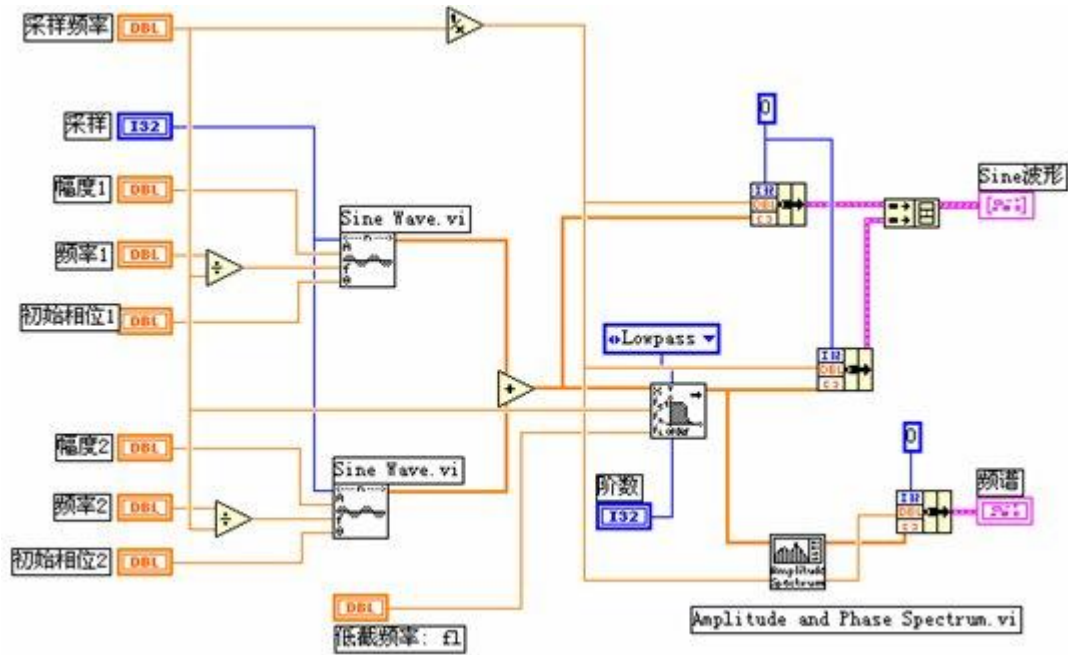
前面板：



1. 打开 Low Pass Filter.vi 程序。这个程序设计了一个低通巴特沃斯滤波器，从输入信号中滤除 10Hz 分量。

2. 转换到框图程序。

框图程序：



1. 验证主框图程序，它使用了下面的子程序： Butterworth Filter VI 子程序（在 Analysis>Filters 子目录）。在本例中，采用巴特沃斯滤波器，指定滤波器阶数与截止频率，滤除输入正弦波中的高频分量，把输出波形在屏幕上显示。

2. 运行该程序，改变滤波器阶数，观察滤波器的效果。

（练习 4-3 结束）

虽然在 LabVIEW 中使用数字滤波器很简便，但还是需要你对数字信号处理的理论有一个基本了解。为此，VI 公司提供了一个附加的工具软件，叫作数字滤波器设计工具箱(Digital Filter Design Toolkit)。该软件用 LabVIEW 开发，有很好的交互式图形用户界面。可以把所设计的滤波器集成到 LabVIEW、LabWindows/CVI 或者其他的编程环境。

第六节 曲线拟合

曲线拟合的目的是找出一系列的参数 a_0, a_1, \dots ，通过这些参数最好地模拟实验结果。下列是 LabVIEW 的各种曲线拟合类型：

- 线性拟合 - 把实验数据拟合为一条直线 $y[i]=a_0+a_1 \cdot X[i]$
- 指数拟合 - 把数据拟合为指数曲线 $y[i]=a_0 \cdot \exp(a_1 \cdot X[i])$
- 多项式拟合 - 把数据拟合为多项式函数：

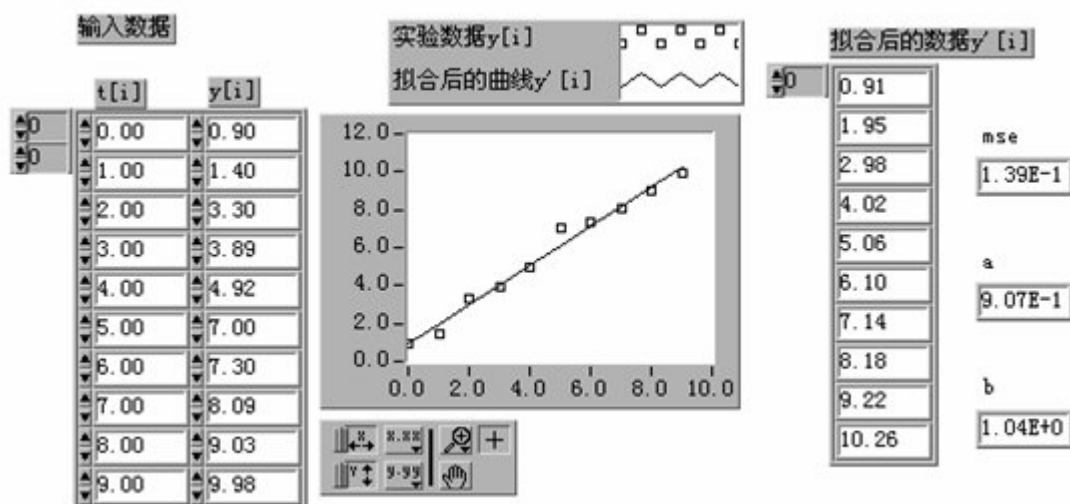
$$y[i]=a_0+a_1 \cdot X[i]+a_2 \cdot X[i]^2 \dots$$
- 通用多项式拟合 - 与多项式拟合相同，但可以选择不同的算法，以获得更好的精度和准确性。

- 通用线性拟合 - 公式为 $y[i]=a_0+a_1*f_1(X[i])+a_2*f_2(X[i])\dots$, 这里 $y[i]$ 是参数 $a_0, a_1, a_2\dots$ 等的线性组合。通用线性拟合也可以选择不同的算法来提高精度和准确度。例如: $y=a_0+a_1*\sin(X)$ 是一个线性拟合。因为 y 与参数 a_0, a_1 有着线性关系。同样道理, 多项式拟合也总是属于线性拟合, 但是它可以采用一些特殊算法以提高拟合处理的速度和精度。
- General Levenberg-Marquardt 拟合 - 把数据拟合为公式 $y[i]=f(X[i], a_0, a_1, a_2\dots)$ 。其中 $a_0, a_1, a_2\dots$ 是参数。这种方法是最通用的方法, 它不需要 y 与 $a_0, a_1, a_2\dots$ 有线性关系。它可用于线性或非线性拟合, 但一般用于非线性拟合, 因为对于线性曲线的处理采用通用线性拟合方法更加快捷。这种方法不能保证结果一定正确, 所以, 有必要验证拟合结果。

练习 4-4

目的: 对指数关系数据进行线性拟合

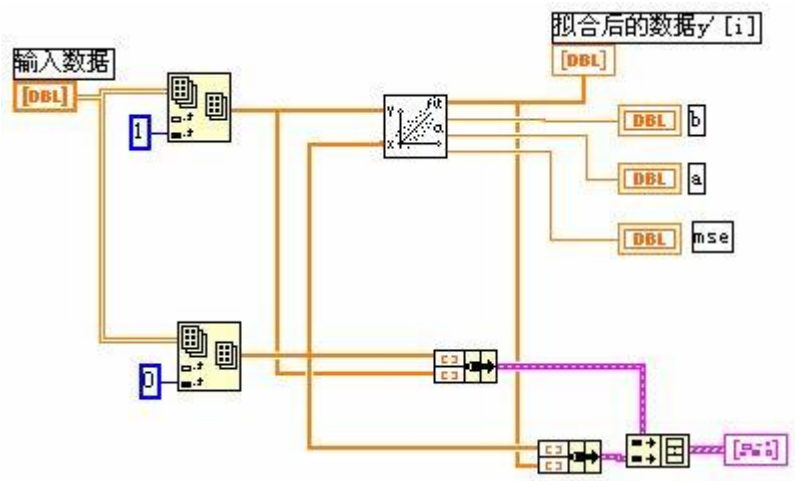
前面板:



1. 打开 Linear Curve Fit.vi 程序。这个例子假设我们收集了 10 对实验数据 t 和 y , 我们有理由相信它们之间有线性关系。

2. 切换到框图程序。

框图程序:



1. 验证框图程序，它使用了如下的子程序：

Linear Fit 子程序（在 Analysis>Curve Fitting 子模板）。在本例中，它把实验数据拟合为一条直线，求出系数 a 和 b ，以满足 $y[i]=a+b*t[i]$ ；以及实验结果和拟合结果之间的误差的均方根值。

2. 输入数据是一个两维数组，是采用 DAQ 卡采集数据时的通用格式。我们用 Index Array 子程序可以分解得到两个一维数组 $y[i]$ 与 $t[i]$ 。

3. “MSE” 表示误差均方根值，误差越小，拟合结果越好。

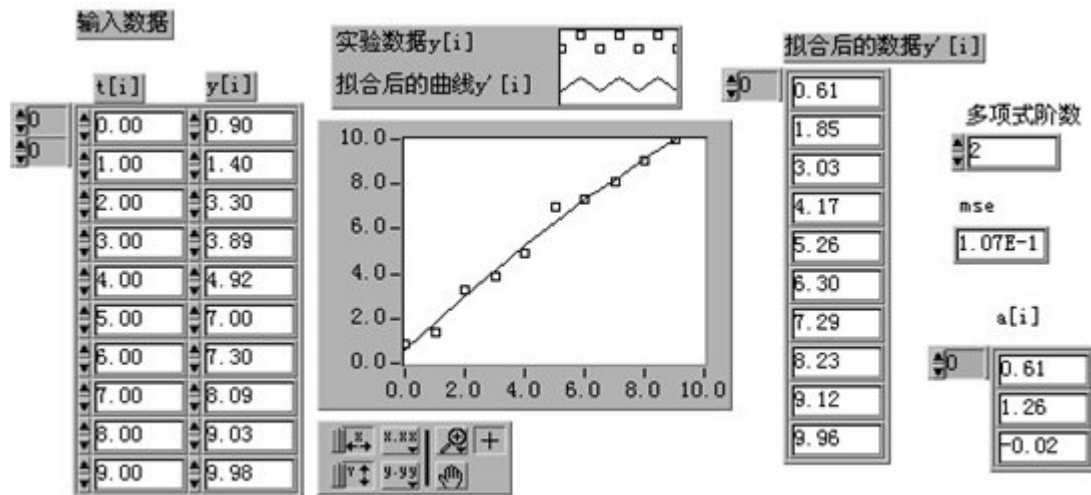
4. 运行该程序。曲线将显示实验数据和拟合结果。

（练习 4-4 结束）

练习 4-5

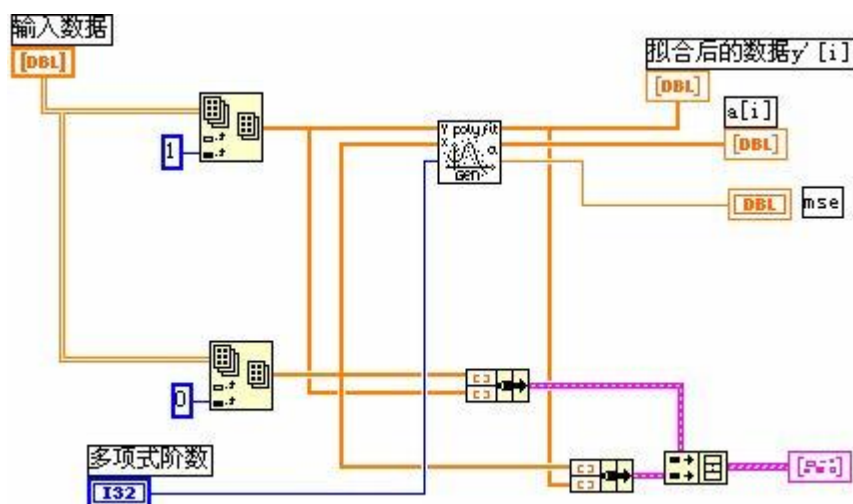
目的：对指数数据执行多项式拟合

前面板：



1. 打开 Polynomial Fit.vi 程序。这个例子假设实验数据之间为多项式关系： $y[i]=a_0+a_1*t[i]+a_2*t[i]^2\cdots$
2. 当多项式为一阶时，有两个参数，结果是一个线性拟合，与前一个例子相同。当阶数为二阶时，结果是一个二阶多项式带三个参数，多项式参数存放于数组 a[i]。
3. 切转到框图程序。

框图程序：



1. 验证框图程序，它使用了如下的子程序：

General Polynomial Fit.vi 子程序（在 Analysis>Curve Fitting 子目录）在本例中，把实验数据拟合为二阶多项式曲线，返回结果为拟合数据，多项式系数以及拟合误差的均方根值。

2. 本例使用多项式拟合，求出了多项式系数 a_0, a_1, a_2 等。一般情况下，我们总是尽可能使用最低阶的多项式。

3. 运行程序。将显示实验数据和拟合结果的数据。

（练习 4-4 结束）

第五课 实用工具软件包

第一节 概述

NI 公司致力于为用户的虚拟仪器系统提供一套完整的解决方法。用户可以获得的资源有：NI 公司以及第三方公司提供的软件工具箱和支持软件，用于扩展支持 LabVIEW；一个世界范围的技术支持网络；帮助用户开发的系统集成工程师和咨询人员。

第二节 常用软件工具箱

应用程序生成器(Application Builder)

这个软件工具箱是用于创建 LabVIEW 的可以单独运行的应用程序。用应用程序生成器创建了可单独运行的应用程序后，你可以运行该程序，但不能编辑它。把你的 VI 程序转换成可单独运用程序，就可保证别人不能查阅或修改这个程序，这样可以防止操作人员不小心修改了程序，也是将 VI 程序打包并派发的一个有效方法。

应用程序生成器提供两种运行模型供选择。你可以创建一个运行引擎（run-time engine），它可以用来运行任何的 VI 程序；或者你可以把运行引擎与一些指定的 VI 程序结合起来，创建一个可自运行的应用程序。

自动检测工具箱(Automated Test Toolkit for LabVIEW)

本工具箱将最强大的测试开发和管理工具融合在一起，可以满足建立大规模生产检测系统，或高复杂度自动确认系统时的各种需要。

自动检测工具箱包含下列程序包：

- **TestStand 即时可用的测试执行工具包:** 用于组织、控制和执行你的自动化框架、确认或生产检测系统。使用 TestStand, 你可以生成自己的操作界面、生成测试报告、以及按照要求顺序地执行测试过程。
- **IVI 驱动程序库:** 在这个库中, 集中了许多高性能的仪器驱动程序, 便于建立先进的测试系统。

企业连通工具箱(Enterprise Connectivity Toolkit)

本工具箱用于跟踪你的企业在生产、测试以及产品服务方面的研究或开发的进展。

企业连通工具箱包含下列程序包:

- **SQL 工具包:** 用于对多种使用 SQL 的当地或远程数据库的直接访问。这些程序使用高级语言编程的方法把通用数据库操作变成简单易用的操作。使用 SQL 工具包, 你可以用 30 多种数据库格式存储、读取或者更新数据记录。有经验的数据库用户可以在 LabVIEW 的工作环境下使用完整的 SQL 语句。你可以把自己的测试程序直接连接到一个数据库以存贮实验结果或者下载测试参数。
- **SPC 工具包:** LabVIEW 不仅可以监控过程, 而且还可以识别总是并且实质性地改善过程处理的质量。SPC 工具箱是过程控制中统计方法应用程序库。使用此工具包, 可以分析并跟踪过程处理程序。该工具包除了 SPC 计算子程序, 还包括许多应用实例。该工具包中包括所有 SPC 功能子程序, 共有三类 — 控制图表、过程统计和 pareto 分析方法。每一类程序都包括分析计算方法和显示绘图功能。

Internet 工具箱

Internet 开发工具用于把 VI 程序转换成可在 Internet 上执行的应用程序。使用 LabVIEW 再配合该工具, 就可以生成在任何 web 浏览器上均可以查阅到的你自己的虚拟仪器前面板的 http 服务器。采用用户查询或者服务器自动更新的工作方式, 前面板所表示的虚拟仪器会自动更新内容。由于程序内装了网络服务功能, 虚拟仪器程序可以对连接到它的多个客户要求作出响应, 为每个用户连续地更新显示。或者, 如果已经运行了网络服务程序, 你可以采用工具箱内的库函数把虚拟仪器程序转换成图像文件, 在 html 网页显示。你还可以在网络化程序内加入访问密码等级, 以限制其他用户对虚拟仪器面板和数据的访问级别。

此外, 该工具箱还包括用 G 图形编程方法建立通用网关接口程序。这样就可以根据用户在网页上的输入动态地决定连接方式。该工具箱还包含一些库函数用于在虚拟表程序内加入 e-mail 电子邮件和 ftp 文件传送功能。

PID 工具箱

PID 控制功能工具箱给 LabVIEW 程序加入复杂的控制算法。使用该软件包，你可以快速建立数据采集与控制系统。将 PID 控制工具箱与 LabVIEW 的算术与逻辑功能相结合，可以快速生成自动控制程序。

PID 程序包带有许多误差反馈及外部复位的 PID 算法。程序同时含有超前-滞后补偿和设置点斜率生成。控制参数包括多重循环，前馈，最小/最大偏差，和斜率/偏差。

PID 算法参数包括无缓冲自动/手动传送，不可处复位终结，正向/反向运行，手动输出调节，和运行/暂停开关。

使用 PID 控制工具箱，用户可以设计 PID 算法的控制策略，可以将输入/输出数据的单位从工程单位转换成百分比，可以设置 PID 算法的时间，最后，还可以针对闭环（最终增益）和开环（步进测试）方式调节过程参数。

Picture Control 工具箱

该工具箱是一个多功能的图形软件包，用于生成前面板显示。该工具箱在 LabVIEW 系统内加入图形控制和相应的程序库。用户可以应用这些程序建立一些框图程序，以动态地建立图像。使用这些工具软件，用户可以在前面板中加入一些新的图形显示。比如特殊的棒形图，饼形图表，和 Smith 图表。用户甚至可以加入一些动画功能，如机器人，测试仪器，UUT，或者二维实体显示等。在 LabVIEW 5.1 版的完全开发版本中已经包括了 this 工具箱。

大型应用程序的开发和有效性规则

当开发大型应用程序时，开发人员必须面对质量标准，源代码控制，复合测量等。LabVIEW 有一个软件工具箱可以查询这方面的问题。该工具箱针对包含多个开发人员的开发小组，或者开发大型应用程序的单个开发人员，或者需要遵循严格质量标准的 G 图形编程人员提供实用工具。

大型 LabVIEW 应用程序通常都必须遵守严格的质量标准，包括 ISO9000，FDA 标准，欧洲认证，以及其他国际标准。本程序包讨论了针对图形编程人员为了达到上述质量标准而必须遵循的步骤。

这一部分包含在 LabVIEW 的专业版本中。

第三节 分析工具软件

HIQ

HIQ 是一个交互式的工具环境，可以对数学，科学计算和工具问题的数据进行组织，可视化处理和组成数据文件。通过使用 HIQ，可以建立交互式 ActiveMath™ 应用程序和数据可视化应用程序。HIQ 集成了数学运算用户接口控制，数值分析，矩阵运算，以及在易于使用的交互式笔记本方式工具环境下的二维，三维和四维图形处理，在些工作环境下，问题和解答都以数学描述语言表达。

HIQ 有两类不同的方法用于数值分析和数据可视化处理——一种是交互式方法而另一种是编程方式。交互式方法可以处理常微分议程，数值积分，非线性方程等。交互式工具软件包括 Data Editor，二维或三维的 Graph Editor 和 Problem Solvers。用户可以用 Problem Solver Notebooks 交互式地处理问题。首先，在 Problem Solver Notebooks 的 GUI（图形用户接口）下输入功能函数和相关参数，然后，就可以在不需编程的情况下运行该函数，获得解答，该工作环境使用在 Data Editor 中定义的矢量和矩阵数据。此外，该工作环境也可以产生二维或三维图形并可以生成 HIQ 描述性语言程序。如果你需要解决一个特定应用总是或者有较多关联的问题，则可以采用 HIQ 描述性脚本编程语言。采用编程方法，可以联接数据、图形、文本、Problem Solver Notebooks 生成的描述性脚本语言程序，以及 600 多个分析和数据可视化功能函数。

使用 LabVIEW 和 HIQ，你可以先假设并建立数学模型，然后，建立控制循环程序采集实验数据并观察处理过程，再进行交互式的数据可视化处理并生成实验报告。LabVIEW 和 HIQ 可以共享上述数据结果，它们集成了科学处理和工程处理的必需工具。

信号处理套件(Signal Processing Suite)

该软件包提供给了用户信号处理的功能和高级的数字信号处理工具。

信号处理套件包含下列程序包：

- **数字滤波器设计工具箱 Digital Filter Design Toolkit**：用于交互式地设计数字滤波器。该工具箱的图形用户接口（GUI）是用于有限脉冲响应（FIR）和无限脉冲响应（IIR）滤波器的通用设计工具。滤波器的设计是通过交互式幅频响应图形或者零极点线图，滤波器的输出包括滤波器系数，传递函数，脉冲和阶跃响应。
- **三分之一倍频程分析工具箱(Third-Octave Analysis Toolkit)**：用于三分之一倍频程分析。该工具箱提供一个 GUI 用于三分之一倍频程分析和数据采集，并且遵守 ANSI S1.11-1986 标准。有了这个工具箱，就可以快速简单地测量声音，振动，和噪声信号。可以设置成一到四个输入通道，并且带有可编程的窗处理，权重，和平均功能。
- **联合时频分析工具箱(Joint Time Frequency Analysis (JTFA) Toolkit)**：用于分析时变信号。该工具箱增强了计算机处理时变信号的能力。它包括

Gabor 频谱图，针对时变频率的 JTFA 算法，以及其他常用算法，如 Wigner-Ville 和 Choi-Williams 算法等。

- **动态信号分析仪(Virtual Bench-DSA)**: 这是一个动态信号分析程序。该软件与 AT/EISA 或者 NuBUS 总线动态信号采集卡相结合，可以构成高质量、低成本的虚拟仪器系统。该软件有一个易于使用的 GUI，设置测量参数可以测出功率谱，幅度谱，相关性，暂态捕捉，交叉谱，和总谐波失真。此外，通过使用 DAQ 卡上的模拟输出功能，还可以进行网络和频率响应测量，也可以把该软件用作低频示波器同时在时域和频域观察信号。总之，Virtual Bench-DSA 是一个理想的低频信号测量与频谱分析工具，非常适合于声音信号，机械振动和噪音的分析。

G Math

G Math 工具箱是一个用于算术运算，数据分析和数据可视化的多用途软件包。该工具箱具有用图形编程方法来解决高级数学计算总是的功能。此外，该工具箱还可以在 LabVIEW 程序前面板中加入议程和数据可视化。G Math 工具箱包括 100 多个高级算术功能程序。例如常微分方程，求解根值，最优化，积分，微分，变换和函数等。该工具箱可以应用于许多场合，如过程控制模拟，生产制造与成本最优化，以及机械系统的模拟。

图像处理工具(Image Processing)

该图像处理软件可以提供图像处理功能和机器视觉功能，它包括 400 多个图像函数和交互式的图像处理窗口。图像可以是一维、二维或者三维。该软件与 NI 公司的 IMAQ 图像采集硬件和 DAQ 硬件卡一起使用。它们结合起来，可以给工厂计算机控制的机器提供视觉，对产品的位置、尺寸、标识符和质量做出精确判断。此外，由于该软件的柔软灵活性和可扩展性，它还可以应用于许多研究领域，从图像算法的开发一直到实验室自动化。

该软件有两种形式—基本版本和提高版本。基本版本具有图像的控制和处理功能，包含进行图像采集、显示、文件管理、区域选择及图像控制所必需的工具。而提高版本包含 400 多种高级图像处理功能，可进行高级图像处理、机器视觉化和分析图像。图像数据可进行编辑或转换，在进行物体形状检测时可以采用二进制算法、灰度等级、密度测量、边缘检测、立体开头检测、频率滤波，以及高级的形状处理函数，一个实物的形状分析需要计算 50 多个参数以确定物体的外形特征。