

从零开始学 LabVIEW  
之  
基于 LabVIEW8.5 的入门讲义

# 目录

前 言.....	4
<b>第一课 LABVIEW 概述.....</b>	<b>5</b>
第一节 虚拟仪器 (VI) 的概念 .....	5
第二节 LABVIEW 的操作模板 .....	7
工具模板 (Tools Palette) .....	7
控制模板(Controls Palette).....	8
函数模板(Functions Palette).....	9
第三节 创建一个 VI 程序 .....	11
1. 前面板.....	11
2. 框图程序.....	12
3. 从框图程序窗口 <b>创建</b> 前面板对象.....	13
4. 数据流编程.....	13
第四节 程序调试技术 .....	14
1. 找出语法错误.....	14
2. 设置执行程序 <b>高亮</b> .....	14
3. 断点与单步执行.....	15
4. <b>探针</b> .....	15
第五节 练习 1-1 .....	15
第六节 把一个 VI 程序作为子 VI 程序调用 .....	19
第七节 练习 1-2 .....	19
第八节 练习 1-3 .....	23
第九节 练习 1-4 .....	24
第十节 练习 1-5 .....	26
<b>第二课 数据采集和仪器控制.....</b>	<b>29</b>
第一节 概述 .....	29
第二节 使用采集卡进行数据采集和仪器控制 .....	32
第三节 LM35 温度传感器 .....	32
第四节 光敏电阻 .....	33
第五节 LED, 三基色 LED 七色跳变 .....	35
第六节 蜂鸣器.....	39
第七节 LED 颜色渐变.....	41
第八节 舵机控制 .....	43
第九节 步进电机 .....	49
<b>第三课 分析软件 (了解即可) .....</b>	<b>52</b>
第一节 概述 .....	52
第二节、高级分析功能程序.....	52

第三节 信号产生 .....	53
练习4-1 .....	53
第四节 信号处理 .....	55
练习4-2 .....	55
第五节 数字滤波器 .....	57
练习4-3 .....	58
第六节 曲线拟合 .....	59
练习4-4 .....	60
练习4-5 .....	61
<b>第四课 实用工具软件包（了解即可） .....</b>	<b>63</b>
第一节 概述 .....	63
第二节 常用软件工具箱 .....	63
第三节 分析工具软件 .....	65

# 前言

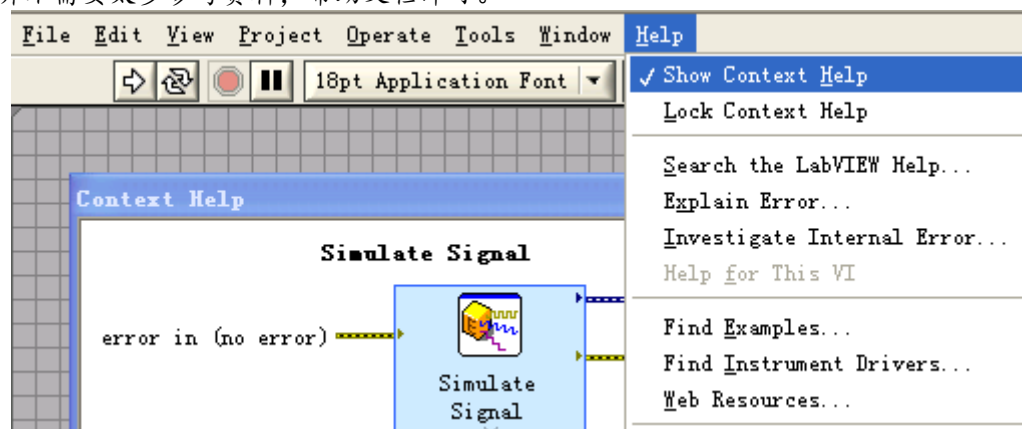
LabVIEW 是美国国家仪器公司 (NATIONAL INSTRUMENTS, 简称 NI) 开发的虚拟仪器集成控制软件, 其采用图形化编程语言 (G 语言) 编写程序, 根据标准协议从传感器或其他仪器串口读取、写入数据。




简言之, 程序中使用图标、图表及连线等创建了计算机屏幕上的“虚拟仪器”, 故程序以 .VI (Virtual Instrument) 为后缀。

虚拟仪器较传统仪器价格低廉, 容易修改更新, 在科研和工业中广泛应用。

LabVIEW 入门简单, 仅需记住以下几个**概念**即可轻松上手:

- 程序界面有**前后面板**之分, 前面板相当于实际物理仪器的操作旋钮板, 后面板相当于其内部连线。
- 优秀的软件总有强大的**帮助文档**支持, LabVIEW 也不例外。点击菜单栏“Help (帮助)”——“Show Context Help (显示即时帮助)”, 将鼠标悬停在函数等图标, 获得实时悬浮窗口的帮助提示。或者直接按 F1 键可以获得全面的帮助文档。有很多牛人学习 LabVIEW 并不需要太多参考资料, 帮助文档即可。



- LabVIEW 强调**数据流**的概念, 基于这一特点, 我们可以利用工具面板的“Probe data (探针)”工具  可以获得实时运行程序的每条连线上的数据情况, 利于我们观察调试。利用“HighLight Execution (高亮执行)”  功能可以查看数据流动情况以及出错节点的位置。
- 当不清楚一个函数节点的使用方式时, 右键单击接口, 选择“**Create (创建)**”——“常量 / 输入控件 / 显示控件”即可自动建立合适数据类型的控件, 方便易行。
- 控制面板和输入面板均有**搜索功能**  Search, 输入函数或模块名, 即可方便查找我们想要的控件。

原则上, 记住上面几条概念, 加上几个简单例子的熟悉就可以短时间内掌握 LabVIEW 的基本操作了。下面, 我们就通过例子中进一步熟悉和学习 LabVIEW。

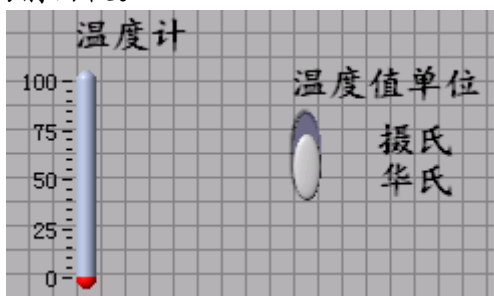
# 第一课 LabVIEW 概述

## 第一节 虚拟仪器 (VI) 的概念

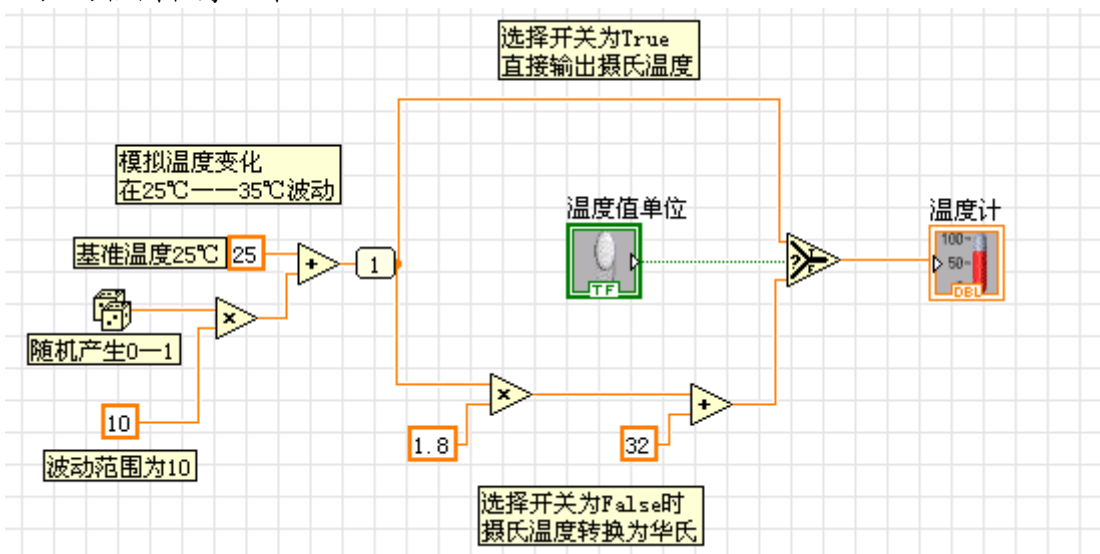
(本节只是举例展示，内容暂时不需要掌握！  
后面有详细介绍的章节)

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。

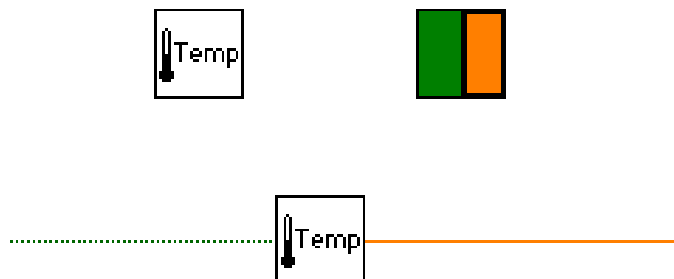


每一个程序前面板都对应着一段框图程序。框图程序用 LabVIEW 图形编程语言编写，可以把它理解成传统程序的源代码。框图程序由端口、节点、图框和连线构成。其中端口被用来同程序前面板的控制和显示传递数据，节点被用来实现函数和功能调用，图框被用来实现结构化程序控制命令，而连线代表程序执行过程中的数据流，定义了框图内的数据流动方向。上述温度计程序 (Thermometer VI) 的框图程序如下：



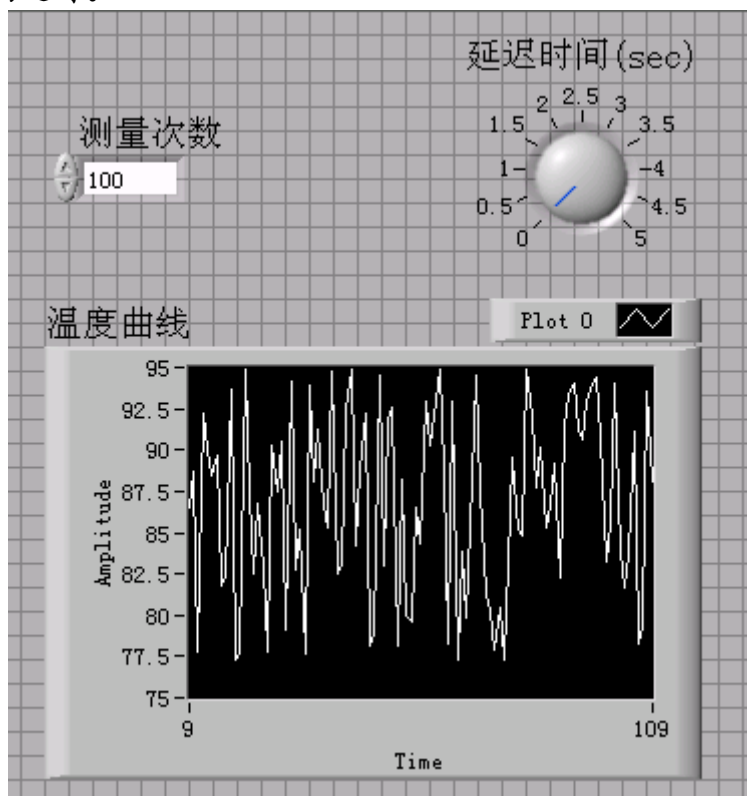
图标/连接器是子 VI 被其它 VI 调用的接口。图标是子 VI 在其他程序框图中

被调用的节点表现形式；而连接器则表示节点数据的输入/输出口，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（右键——快捷菜单——显示项——接线端）。

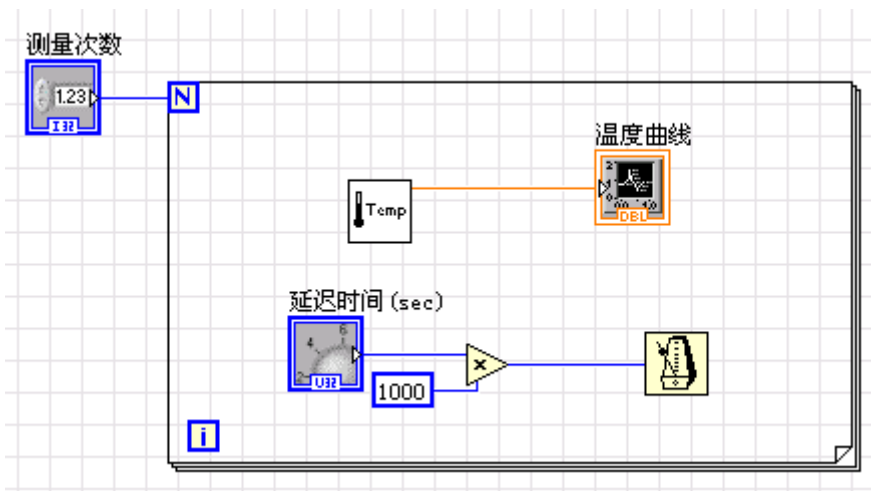


LabVIEW 的强大功能归因于它的层次化结构，用户可以把创建的 VI 程序当作子程序调用，以创建更复杂的程序，而这种调用的层次是没有限制的。

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

(不必记住所有图标，仅需了解常用功能，以及知晓大体所在位置及大致用途即可，本节请花几分钟时间快速浏览即可)

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：



自动选择工具：根据鼠标位置系统自动决定转换成下面几种图标功能的一种。











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。



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。



-  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。
-  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。
-  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。
-  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。
-  断点工具：使用该工具在 VI 的框图对象上设置断点。
-  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。
-  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。
-  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，控制和函数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。

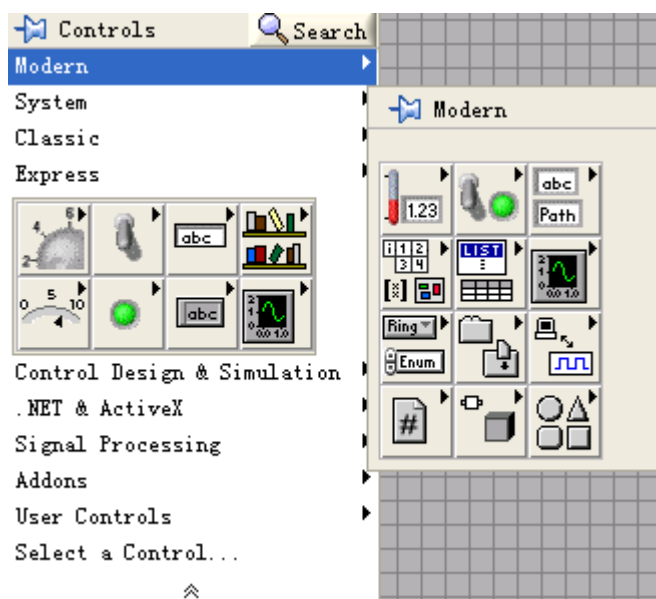
## 控制模板(Controls Palette)

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。

控制模板如下图所示，它包括如图所示的几个常用子模板。





数值子模板：包含数值的控制和显示。



布尔值子模板：逻辑数值的控制和显示。



字符串子模板：字符串和表格的控制和显示。



数组和群子模板：复合型数据类型的控制和显示。



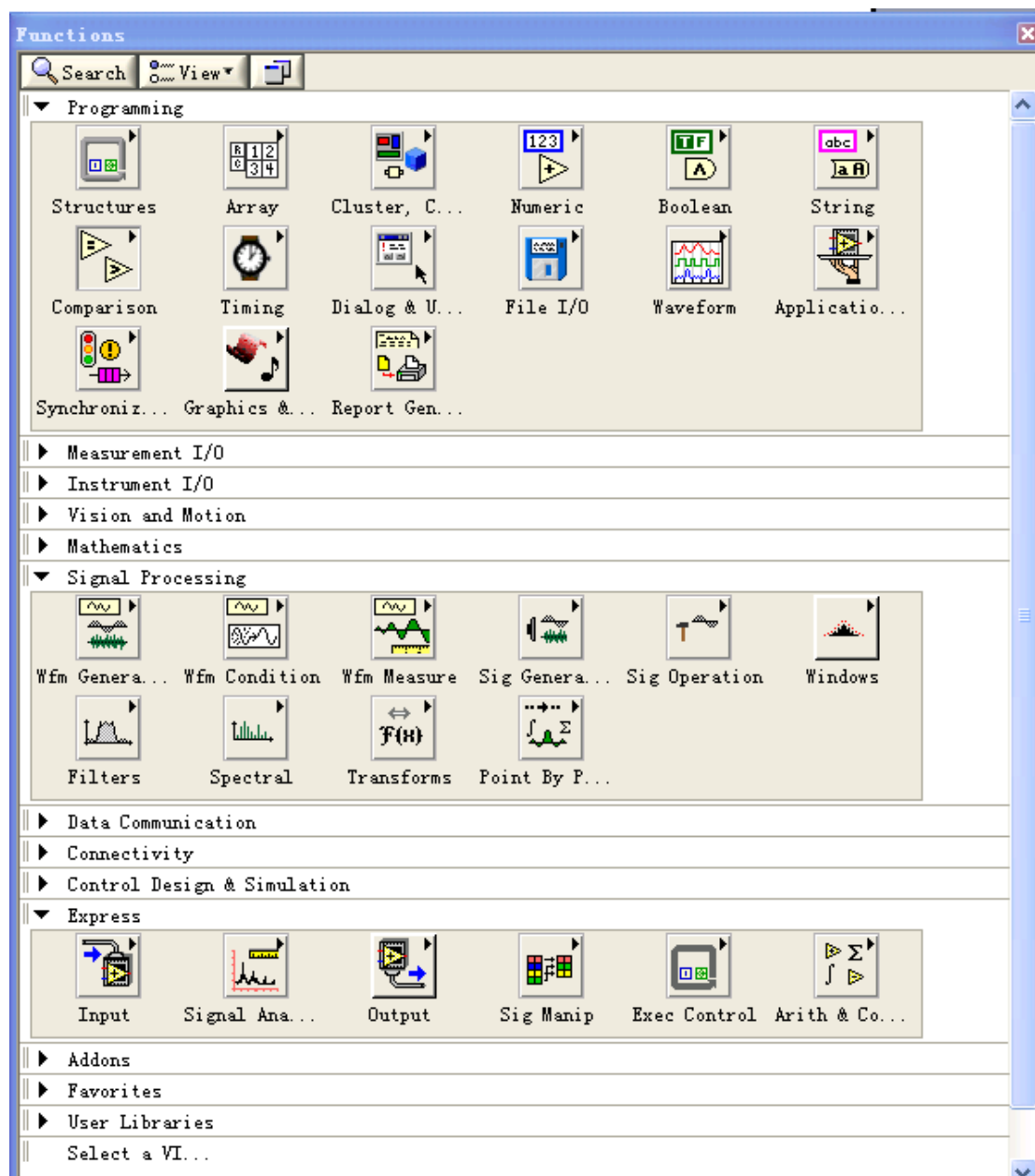
图形子模板：显示数据结果的趋势图和曲线图。

## 函数模板(Functions Palette)

函数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。若函数模板不出现，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函数模板。

注：只有打开了**框图程序（后面板）**窗口，才能出现函数模板。

函数模板如下图所示。



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。



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。



布尔逻辑子模板：包括各种逻辑运算符以及布尔常数。



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。



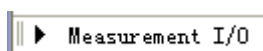
数组子模板：包括数组运算函数、数组转换函数，以及常数数组等。



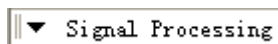
比较子模板：包括各种比较运算函数，如大于、小于、等于。



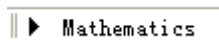
时间和对话框子模板：包括对话框窗口、时间和出错处理函数等。



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。



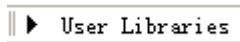
信号处理子模板：包括信号发生、时域及频域分析功能模块。



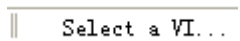
数学模型子模块：包括统计、曲线拟合、公式框节点等功能模块，以及数值微分、积分等数值计算工具模块。



图形与声音子模块：包括 3D、OpenGL、声音播放等功能模块。



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。

## 第三节 创建一个 VI 程序

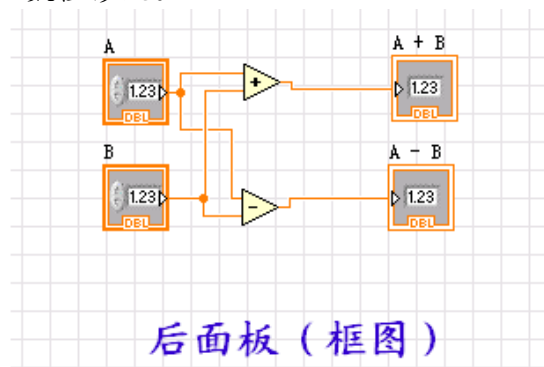
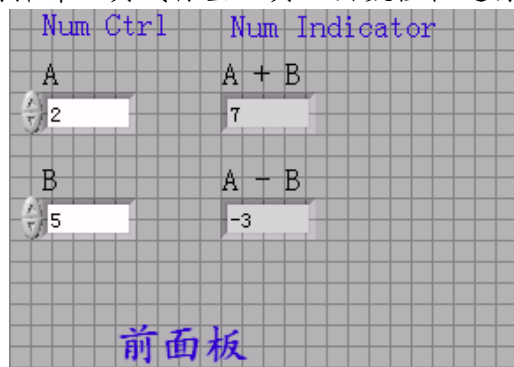
请记住，VI 程序具有三个要素：**前面板**、**框图程序(后面板)**和**图标/连接器**。我们将在练习 1-1 中讨论图标/连接器。

### 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（如，数值子模板）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

(下面几个概念暂时不清楚没有关系, 随着熟悉程度的增加, 过一段时间回过头看会发现一下子就清楚了)

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：**

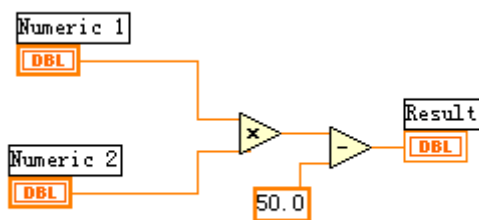
接线头是为了帮助正确连接端口的连线。当把连线工具放到端口上，接线头就会弹出。接线头还有一个黄色小标识框，显示该端口的名字。

### 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。

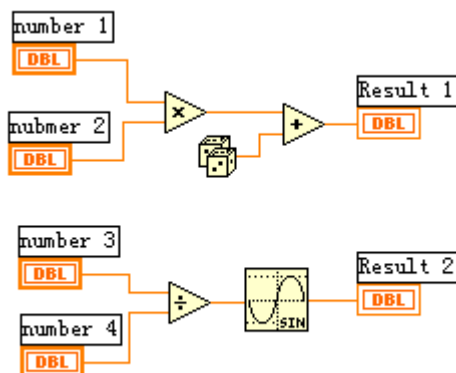
### 4. 数据流编程

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。




如上图所示，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(见右图)，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（后有介绍）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。

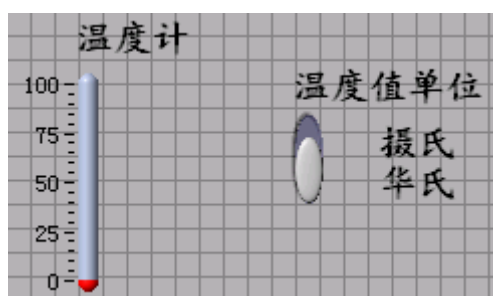
## 第五节 练习 1-1

**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  $70^{\circ} F$  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数据采集卡。使用随机生成函数（范围 0-1）来仿真电压测得值，然后把所测得的电压值转换成摄氏或华氏温度读数。

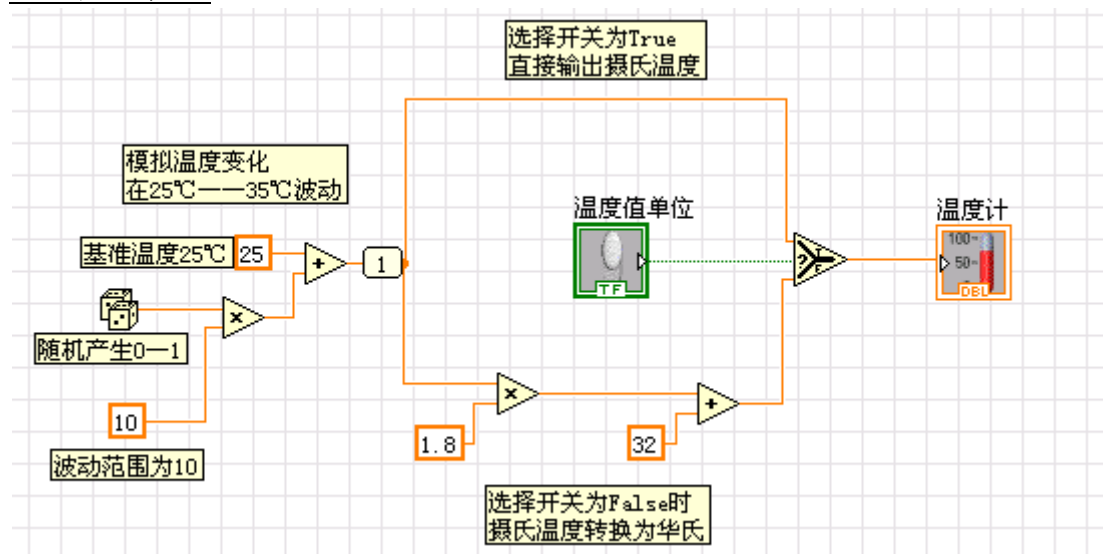
前面板：



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 Indicators 子模板中选择 Thermometer 图标。
  - b. 在高亮的文本框中输入“温度计”，再点击鼠标键按钮。
3. 如果要重新设定温度计的标尺范围，可以使用标签工具 A，双击温度计标尺的 100.0，可输入任意值，如 120.0，再点击鼠标键退出输入或者工具栏中的 A 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 Slide 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）位置旁边点击空白处输入自由标签“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 Block 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 Function (函数) 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：

Multiply (乘法) 功能 (Numeric 子模板)。在本例中，摄氏温度乘以 1.8 以便下一步转换为华氏温度。

Add (加法) 功能 (Numeric 子模板)。在本例中，将相乘得到的值加上 32.0，摄氏温度转换成了华氏温度。



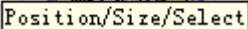
Random Number 程序 (Mathematics——Numeric 子模块, 若不好寻找该函数)。在本例中, 该程序模拟从采集卡的电压读取通道读取的真实物理量电压值。

我们假设温度围绕 25°C——35°C 波动, 因为 Random Number 为 0 到 1, 所以“模拟温度 = 25 + Random Num \* 10”

Select (选择) 功能 (Programming——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。或者在接口处, 右键调出快捷菜单, 选择“创建——常数”。若要修改常数值, 用标签工具双点数值, 再写入新的数值。

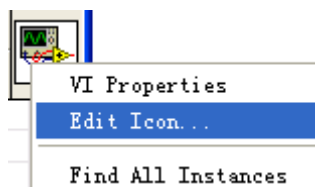


3. 使用移位工具 (Positioning tool) , 把图标移至图示的位置, 再用连线工具连接起来。

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 Terminals 功能。也可以从 Help 菜单中选择 Show Context Help 功能以打开即时帮助信息窗口。

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, 使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。

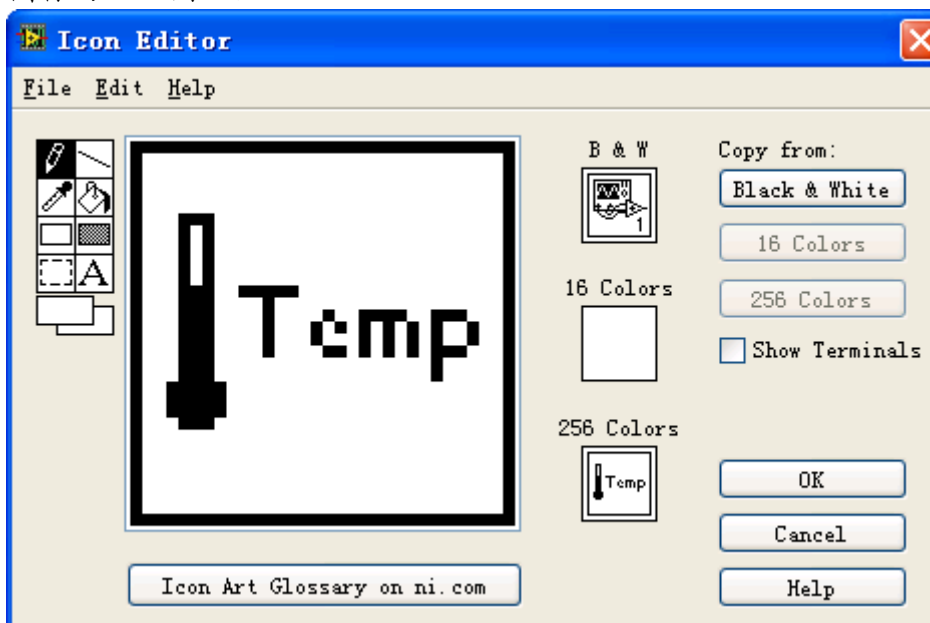


- b. 双点选择工具, 并按下 Delete 键, 消除缺省的图标图案。
- c. 用画图工具画出自定义的温度计的图标。

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。

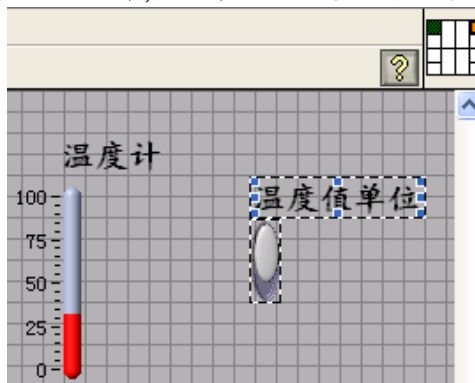
图标的画法例如：



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. 在前面板右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，这即表示着右边的联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，而前面所选择的联接器端口将变暗，表示你已经将对象部件定义到各个联接器端口。

注意：LabVIEW 的惯例是前面板上控制的联接器端口放在图标的接线面板的左边，而显示的联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。

8. 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。联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

### (练习 1-1 结束)

## 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 程序定义了图标和联接器端口即可。用户使用函数模板的“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。

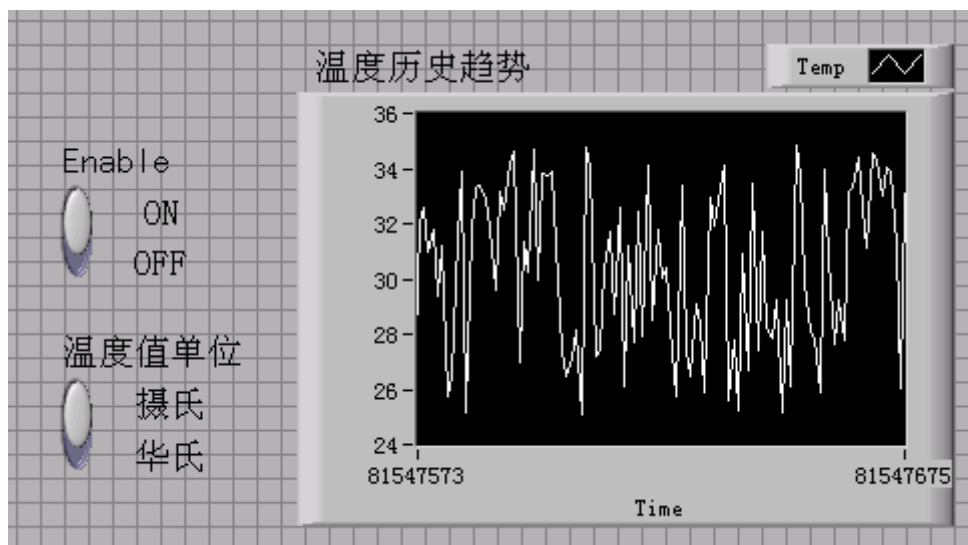
一个子 VI 程序，相当于普通程序的子程序。节点相当于子程序调用。子程序节点并不是子程序本身，就象一般程序的子程序调用语句并不是子程序本身一样。如果在一个框图程序中，有几个相同的子程序节点，它就象多次调用相同的子程序。请注意，该子程序的拷贝并不会在内存中存储多次。


## 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。

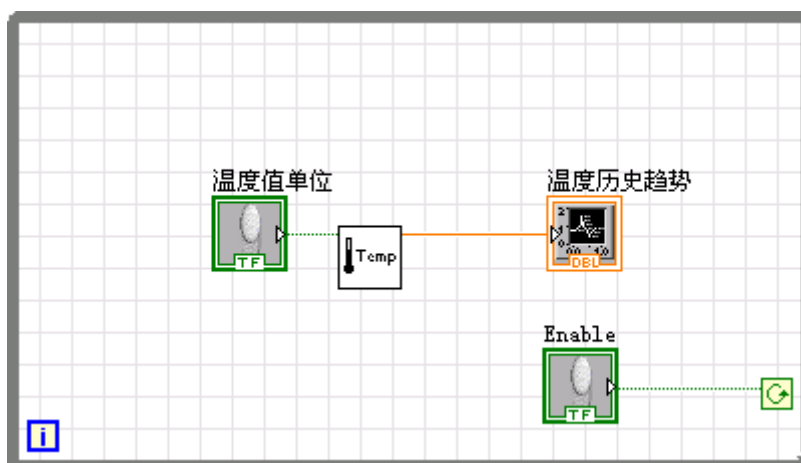
创建一个 VI 程序，进行温度测量，并把结果在波形图表上显示。该 VI 程序使用我们前面创建的温度计程序 (Thermometer VI) 作为子程序。



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（Express —— Graph Indicators 子模板 —— 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。键单击 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。删去系统自带的停止按钮，将 Enable 连接上去，并且将  (stop if ture) 单击改为  (continue if ture)。

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。

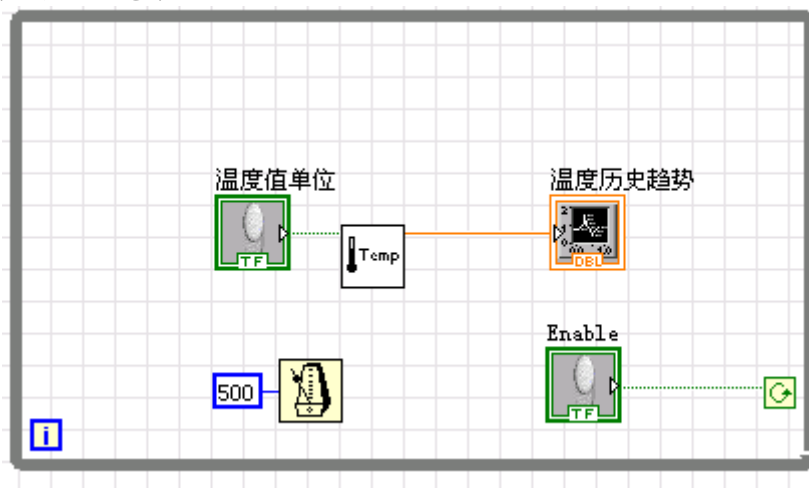
3. 放入其它的框图程序对象。Thermometer VI，这个 VI 程序是你在上个练习中创建的，从 Select a VI...子模板中调出。
4. 按照上图的框图程序连好线。
5. 创建模式开关。把连线工具放在 Thermometer VI 的 Mode 输入端口上，按鼠标右键并选择 Creat Control，这样就可以自动创建模式转换开关，并将它与 Thermometer VI 子程序相连线，再转换到前面板窗口，将模式转换开关的位置重新调整。
6. 在前面板窗口，使用标注工具，双击模式开关的 “OFF” 标签，并把它转换成 “华氏”，再把 “ON” 标签转换民 “摄氏”。要转换开关状态，使用操作工具 (Operating Tool)。
7. 将模式开关设置为 ON 状态，点击运行  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. 若修改时程序在运行状态，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。

增加定时器控制：

当你运行程序时，它将会尽可能快地运行。但是你也也许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 Programming——Timing 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。



10. 如上图所示，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（右键——创建——常量），把它设置为 500。
11. 运行上述程序，试用不同的时间间隔值。
12. 关闭并保存上述程序，文件名为 Temperature Monitor.vi。

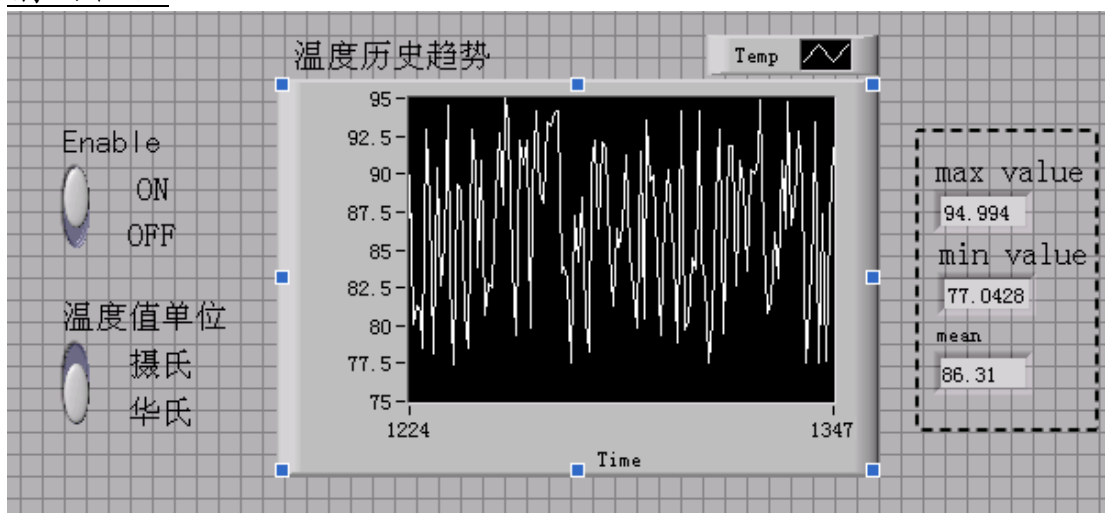
(练习 1-2 结束)。

## 第八节 练习 1-3

**目的：**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，算出最大值、最小值和平均值。

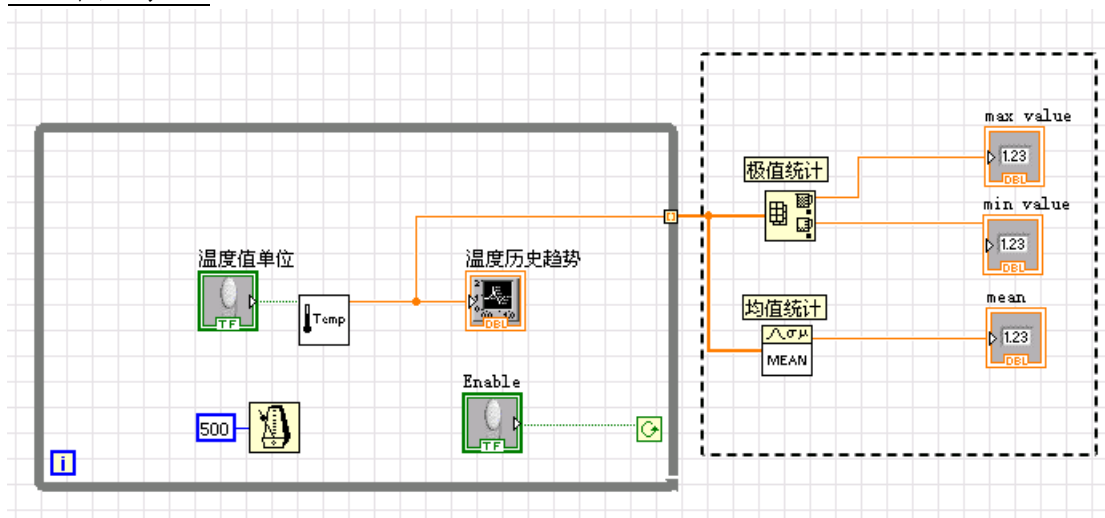
前面板：



1. 打开练习 1-2 创建的 Temperature monitor.vi 程序。
2. 被虚线框住的部分是新增加的。我们此次从后面板生成前面板的图标。

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 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。

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，此时连线是错误的。

2. 数组最小最大值函数的获取在后面板，函数模块 Programming —— Array —— Array Max & Min;  
平均值函数的获取：Programming —— Mathematics——Pro & Stat;  
建立完毕二者之后，在接口处右键单击“创建——显示控件”即可在前面板生成控件。
3. 返回前面板，并运行 VI 程序。
4. 面修改后的程序重命名为 Temperature Analysis.vi 并存盘。

### (练习 1 - 3 结束)

## 第九节 练习 1-4

**目的：**学习使用 Case 结构。

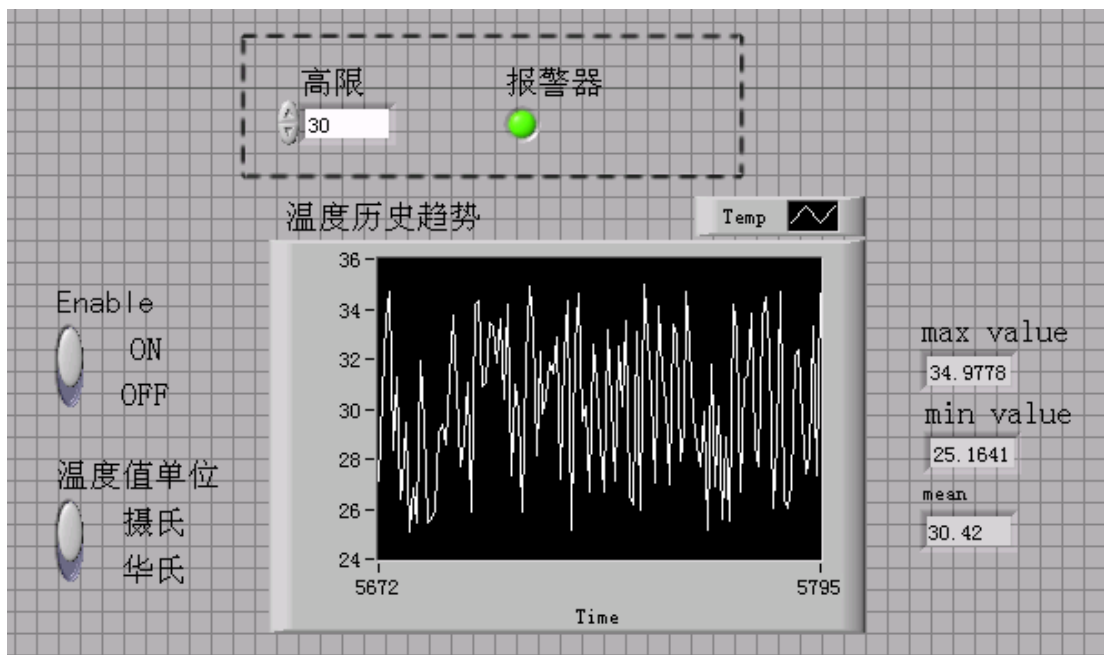
修改 Temperature Analysis.VI 程序以检测温度是否超出范围，当温度超出上限 (High Limit) 时，前面板上的 LED 将点亮，并且有一个蜂鸣器发声。

**前面板：**

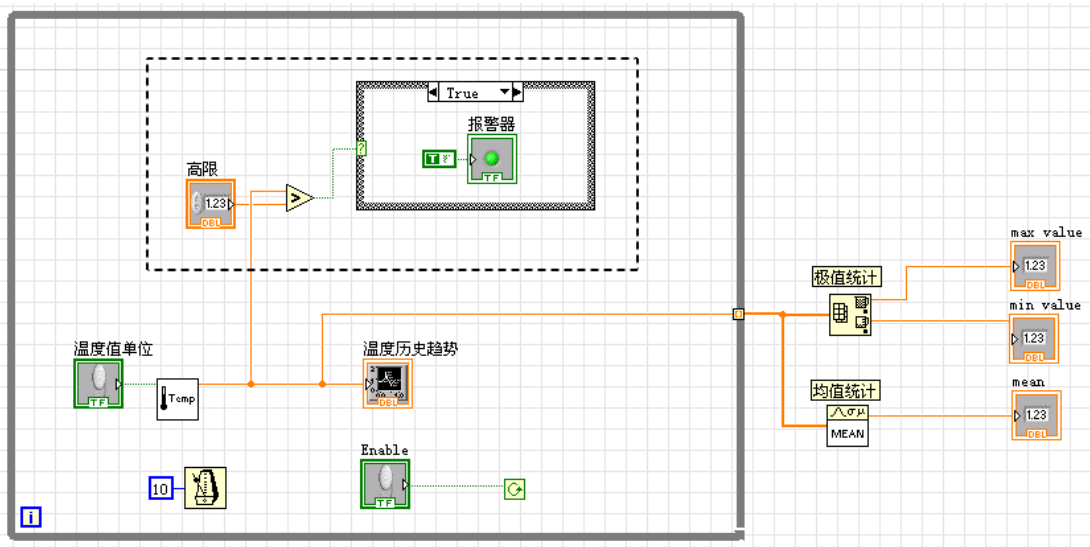
1. 打开练习 1-3 创建的 Temperature Analysis VI 程序。
2. 下图中被虚线框住的部分表示增加的部件，请对应添加，当然在后面板生成也是可以和方便的。“高限”表示温度上限值，为数字输入控件。“报警器”为圆形 LED 图标，用来表示温度是否超限。

它们分别位于前面板——Control (控制模块) ——Express——Num Ctrl 和 Express——LEDs——Round LEDs

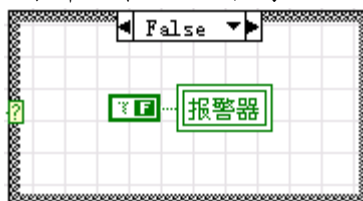




框图程序:



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“高限”输入数值，将执行 True Case 程序，反之则执行 False Case 程序。

注意到报警器是同一个，在 False 面板上，“报警器”是 True 面板的局部变量，代表为同一个指示量，创建方式为，框图视图下，右键 True

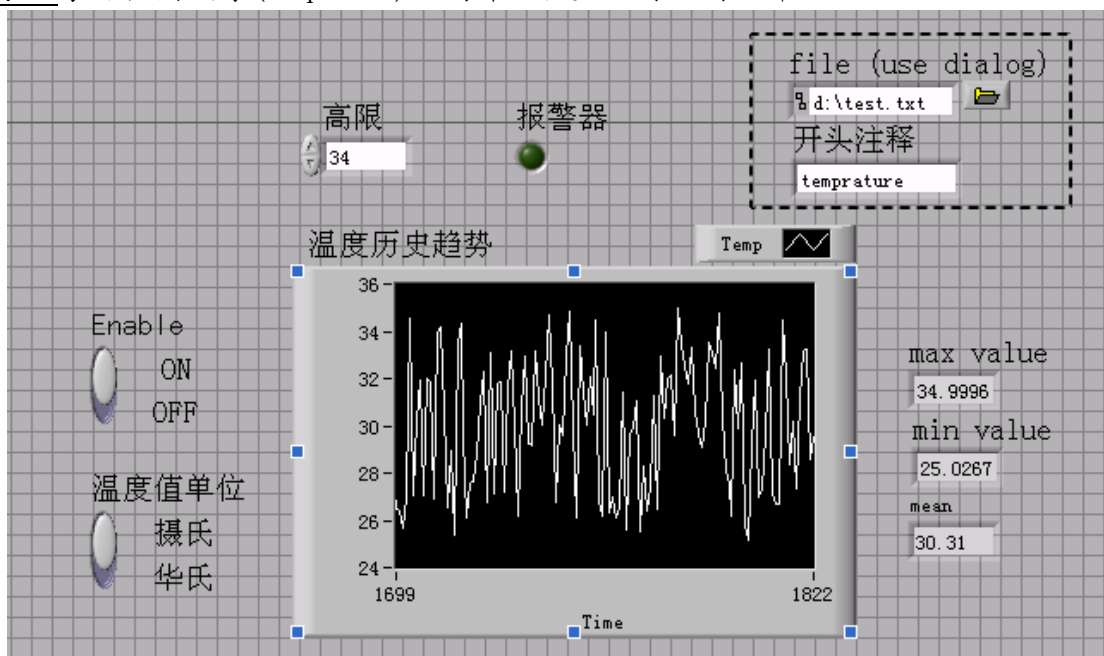
面板中“‘报警器’——创建——局部变量”。同时在 False 面板上，可以右键选择从该局部变量读取还是写入值。

2. 返回前面板程序，在“高限”控制栏中输入想要控制的温度值，例如，摄氏模式下的 34°C。再运行 VI 程序。当温度超过 34°C 时，LED 将点亮。
3. 将程序重新命名为 Temperature Control.vi，并保存起来。





### (练习 1 - 4 结束)

## 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。



虚框中为新增加内容，我们将从框图（后面）中生成添加之。

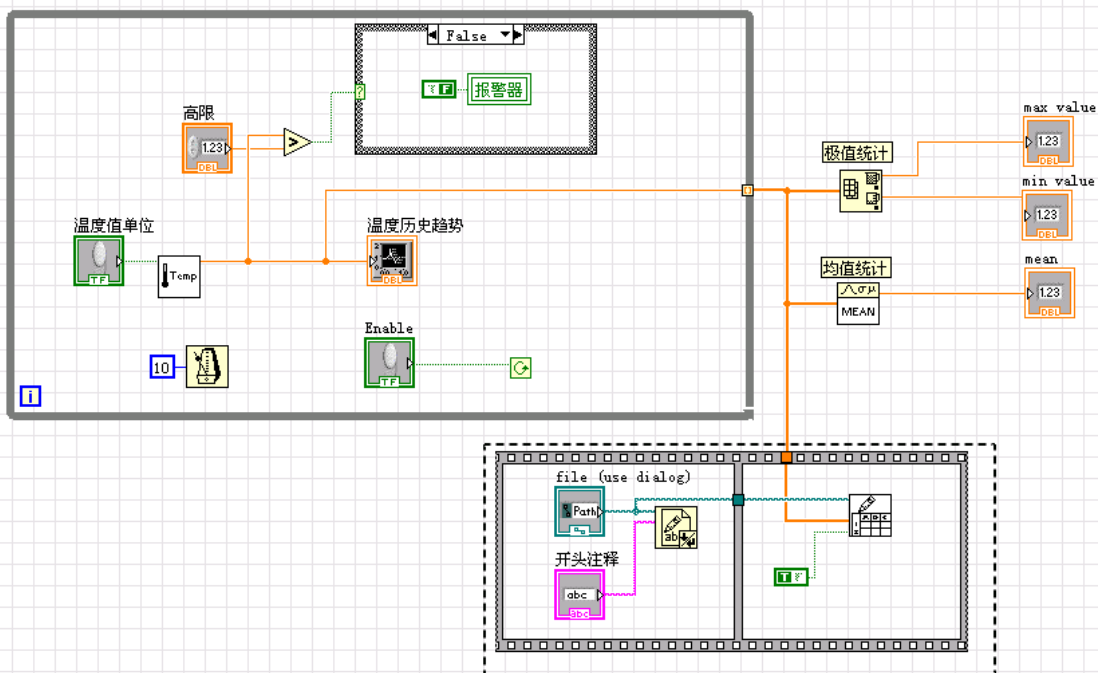
当程序停止数据采集后（此处必须使用 Enable 停止，不能使用     红色键，不然程序提前终止，不会有数据写入文件），开头注释为任意添加的字符串，采集的数据附在其后，共同写入指定的路径位置。

### 前面板：

如上图所示。

### 框图程序：

1. 框图程序如下，下面将逐步介绍：

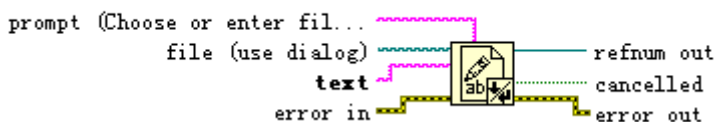


使用了平铺式顺序结构，顺序结构程序按指定的顺序执行各个程序步骤。在本例中，首先创建数据文件的开头注释文件，然后再将数据写入文件中。



第一部分使用了 Write To Text File Write Text... (在后面板, Functions—— Programming —— File I/O 子模块) 查看即时帮助以了解端口分布:

**Write to Text File**



在后面板 Write To Text File 图标的 file 端口右键单击“Create (创建) ——Control (控制) 元件”，即可建立文件路径元件，查看前面板，出现 file 端口，既可以在里面写入想要创建或打开的文件路径，也可以在运行时使用对话框选择路径。

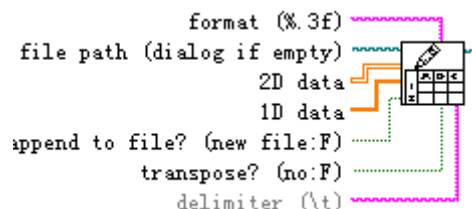
在 text 端口右键单击“Create (创建) ——Control (控制) 元件”，可以创建写入开头注释的元件。前面板生成的文本框可以填写任意的需要对后面加入数据说明的文字。




第二部分使用了 Write to Spreadsheet File 模块 Write Spre... (在后

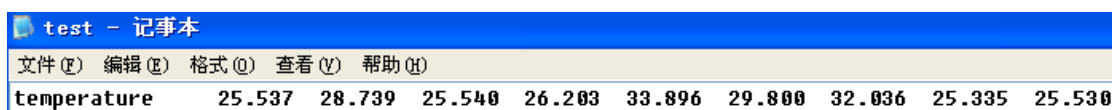
面板, Functions—— Programming —— File I/O 子模块)。该模块把一个二维或者一维单精度数组转换成字符串, 并把字符串写入一个新文件或者附回在一个已存在的文件后面。在本例中, 它将由温度采集数据组成的一维数组附加在一个已创建了开头注释的文件后面。

查看即时帮助:



File path 与前面一部分共享。因为是要附加输入, Append to file 输入布尔常量 “True”。

2. 返回前面板, 在“开头注释”控制栏中输入任意文字, 如“temperature”, 再输入数据文件名 (例如 D:\test.txt)。 **特别注意: 应当使用运行而非持续运行按钮来运行该程序。** 当 Enable 开关设置为 OFF 状态后 (只能使用该停止方式, 而不能通过  中断, 不然文件输入前程序即截止了), 打开 D 盘查看该文件, 将产生如下的 ASCII 文件。



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 数据采集和仪器控制

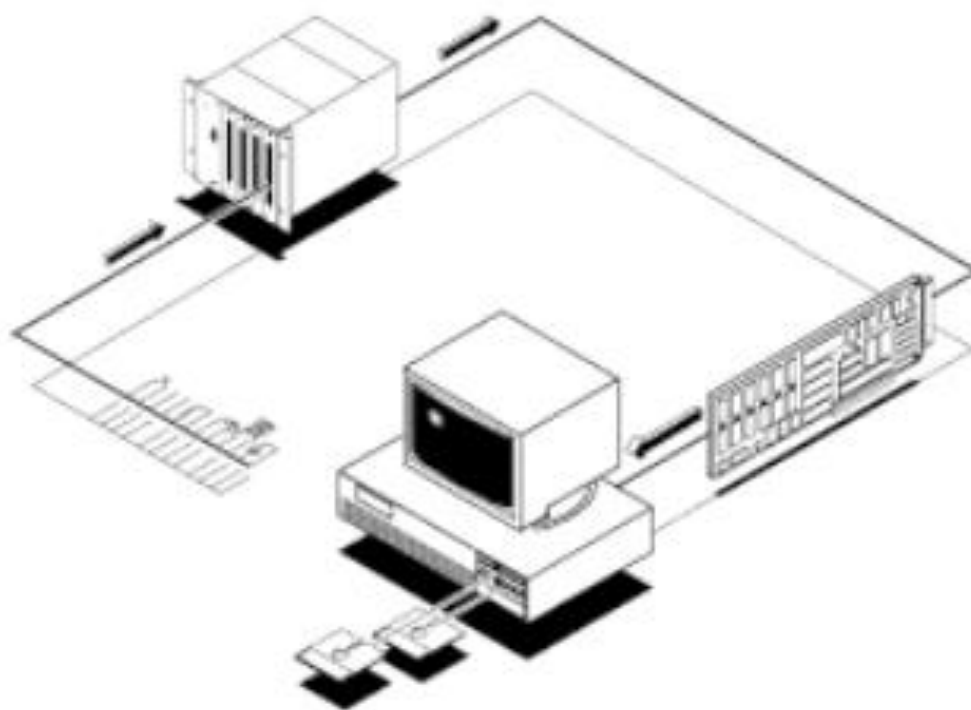
### 第一节 概述

labVIEW 程序普遍配合数据采集卡使用，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。用户在使用之前需要阅读板卡说明和下载安装相关硬件驱动。这些控制程序用到了许多低层的驱动程序。

本课程中以 MPS-010602 采集卡为例，讲解如何使用 labVIEW 编程处理从仪器中读取或写入信号。

**传感器**是电路中的重要组件，本课程将介绍几款常用类型的传感器，结合数据采集卡进行控制。

#### 数据采集系统的组成：



采集卡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采集卡，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采集卡系统 - 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采集卡卡。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，计算机与采集卡系统之间的通讯可以采用各种不同的总线，如并行口或者 PCMCIA 等完成。这种结构适用于远程数据采集和控制系统。

实验中采用的是外加板卡的方式。

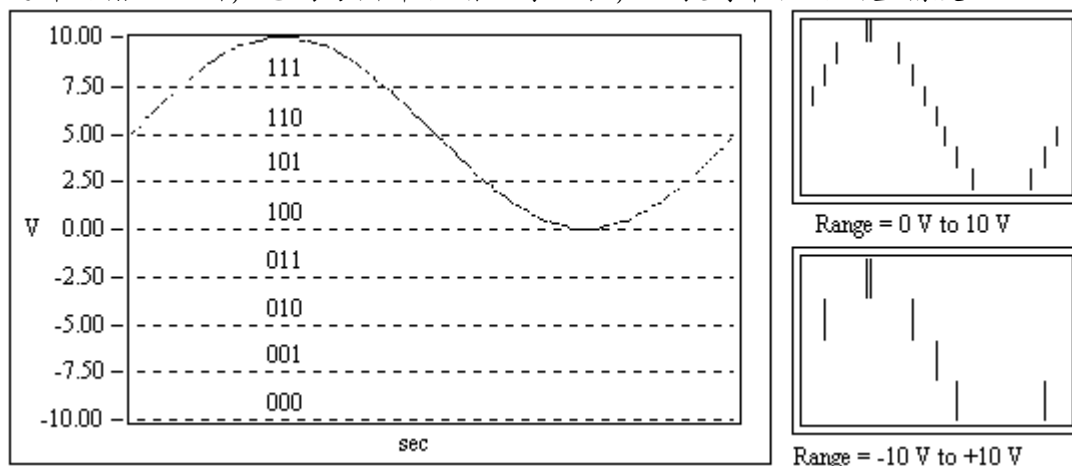
(下面部分暂时不懂没有关系，但当实际遇到相关问题时，记得回来找解决方案即可)

模拟输入：

当采用采集卡测量模拟信号时，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。

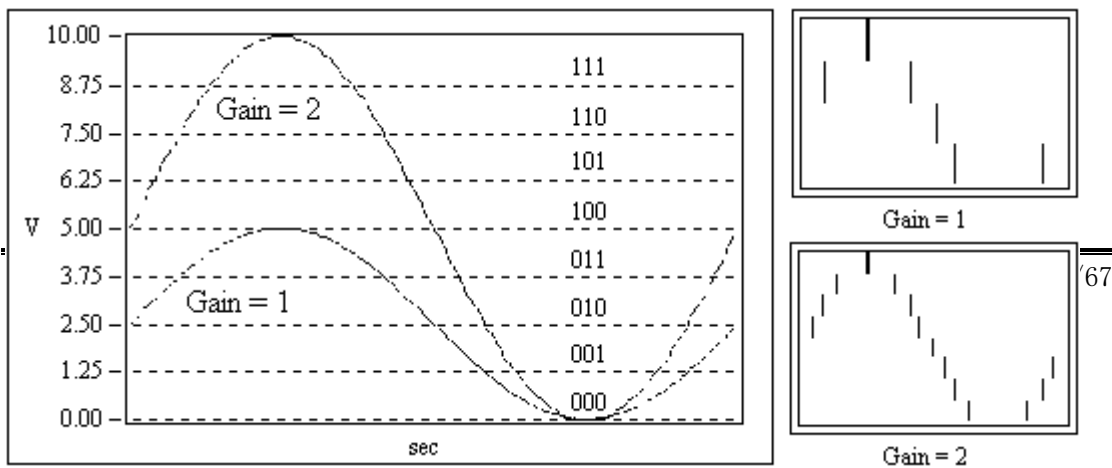
单端输入以一个共同接地点为参考点。这种方式适用于输入信号为高电平（大于一伏），信号源与采集端之间的距离较短（小于 15 英尺），并且所有输入信号有一个公共接地端。如果不能满足上述条件，则需要使用差分输入。差分输入方式下，每个输入可以有不同的接地参考点。并且，由于消除了共模噪声的误差，所以差分输入的精度较高。

输入范围是指 ADC 能够量化处理的最大、最小输入电压值。采集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  $2^3$  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  $2^{16}$  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

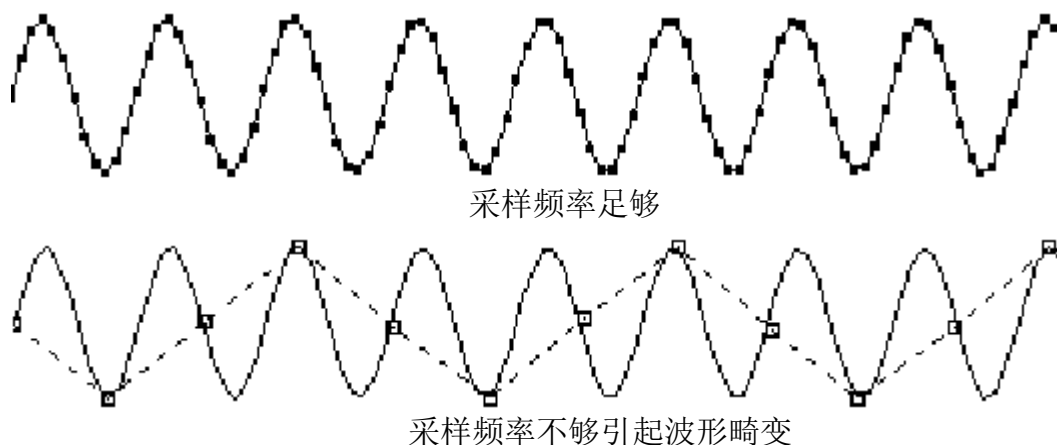


使用一个 3 位模数转换，输入信号范围为 0 到 10 伏，上面的图显示了给信号设置增益值的效果。当增益=1 时，模/数转换只能在 5 伏范围内细分成 4 份，而当增益=2 时，就可以细分成 8 份，精度大大地提高了。但是必须注意，此时实际允许的输入信号范围为 0 到 5 伏。一但超过 5 伏，当乘以增益 2 以后，输入到模/数转换的数值就会大于允许值 10 伏。

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<sup>分辨率</sup>)。

例如，一个 12 位的采集卡，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。

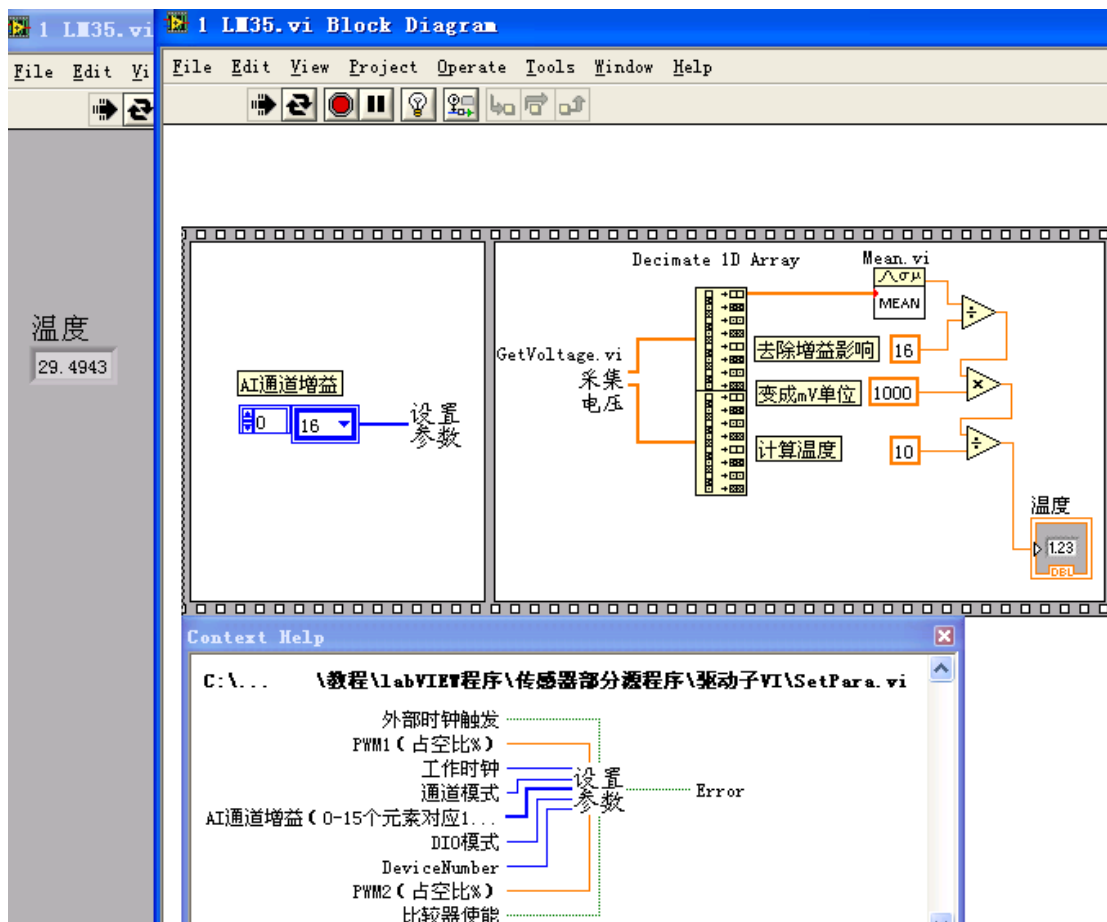
平均化。噪声将会引起输入信号畸变。噪声可以是计算机外部的或者内部的。要抑制外部噪声误差，可以使用适当的信号调理电路，也可以增加采样信号点数，再取这些信号的平均值以抑制噪声误差，这样误差可以减小到乘以下面的系数：

$$\frac{1}{\sqrt{\text{采样点数}}}$$

例如，如果以 100 个点来平均，则噪声误差将减小 1/10。







**注意：**

采集电压子VI是通过在程序框图右键,点击“选择VI...”加载 getvoltage.vi 得到的,这是已经封装的针对MPS本款数据采集卡的子VI(可理解为可调用函数)当希望了解自定义子VI或其他函数的功能和连线规则时,点击“帮助”——“显示即时帮助”,会得到实时的参数输入提醒(上图中下框),这在软件使用初期是非常有用的,务必用好。

另外,采集卡自带函数一般有VI说明,如MPS-010602说明书中第13页说明了采样数(int SampleNumber)等变量有需要为128的倍数的要求(默认128)。

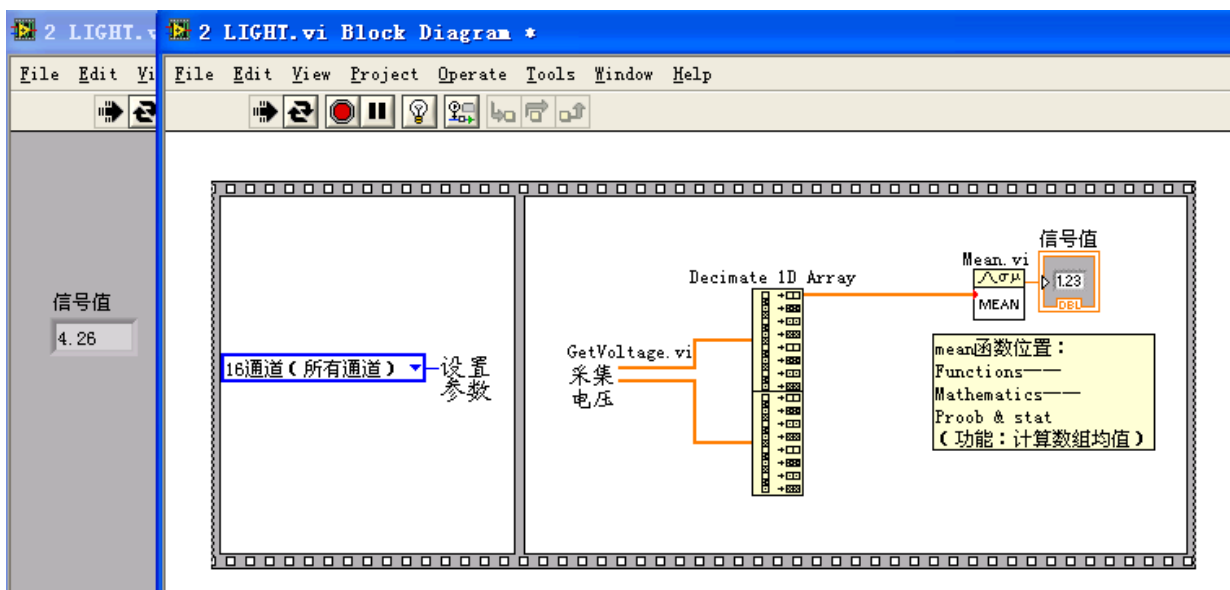
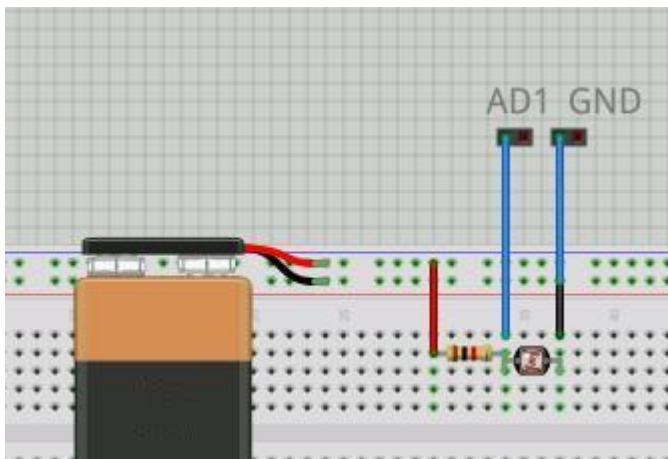
其中,“采集电压”后接数组函数“抽取一维数组”,其意义在于,电压采集的原始数据按照每个端口逐个顺序循环依次记录,而调用时需要单个端口数据,如此例中抽取了AD1端口数据。(关于多通道电压采集时数组记录方式详细说明,参见说明书第12页 float \* VoltageIn1) (关于数组函数,可参见《labVIEW宝典》,陈树学著,第47页)

### 第四节 光敏电阻

**光敏电阻特性：**阻值随光强增大而减小。

本实验将1K电阻与光敏电阻串联,根据分压原理,使用AD引脚检测光敏信号与LM35实验类似。

可以使用手机后置LED照耀光敏电阻,观察压值变化,进行效果验证。



其中“设置参数”的“通道模式”处右键——创建——常量，可以选择打开所有通道或其他。

## 第五节 LED, 三基色 LED 七色跳变

### ①单色 LED:

利用 Dx 数字信号输入输出接口, 我们可以实现 TTL 电平 (晶体管-晶体管-逻辑电平, +5V 等价于逻辑“1”, 0V 等价于逻辑“0”) / COMS 电平信号输出, 下面的例子中, LED 两端连接 D1、GND (接地), D1 输出高电平时 LED 点亮。

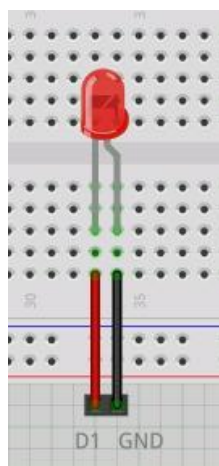
Dx 输入输出数据类型均为二进制, 参见 MPS-010602 使用说明书第 13、14 页 unsigned char \* DI、\* DO 项说明:

*DI 为一个一维数组, 其每个元素为 8 位 unsigned char 型数据, 8 个数据位分别代表同一时刻采样得到的 8 路数字端口电平状态。*

*如:  $DI[i] = 17$ , 即  $DI = 0b00010001$ , 表示 D1 和 D5 为高电平, 其余 6 个端口为低电平。*

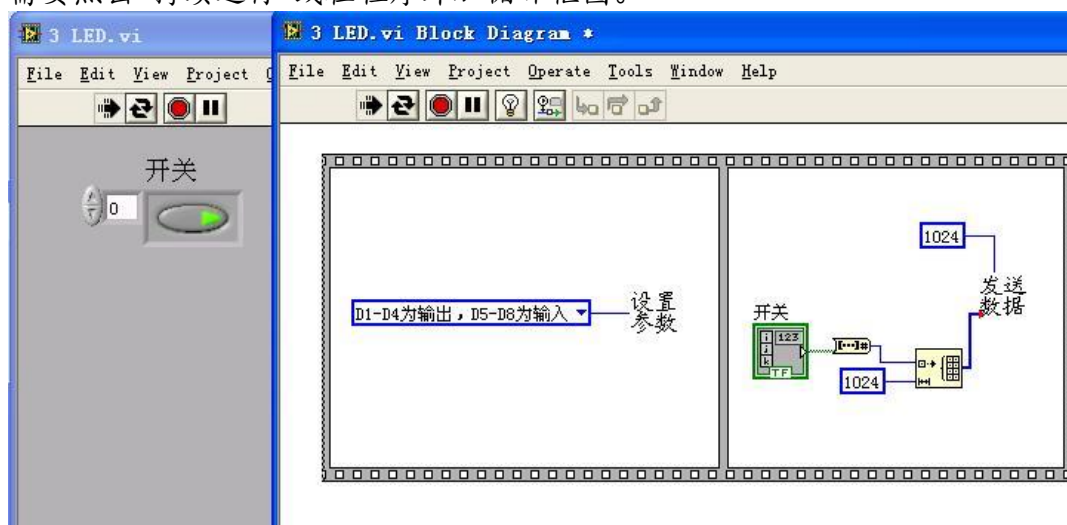
另外, 说明书第 12 页写明了端口置输入还是输出模式的转换规则 (int DIOModal)

连线图:



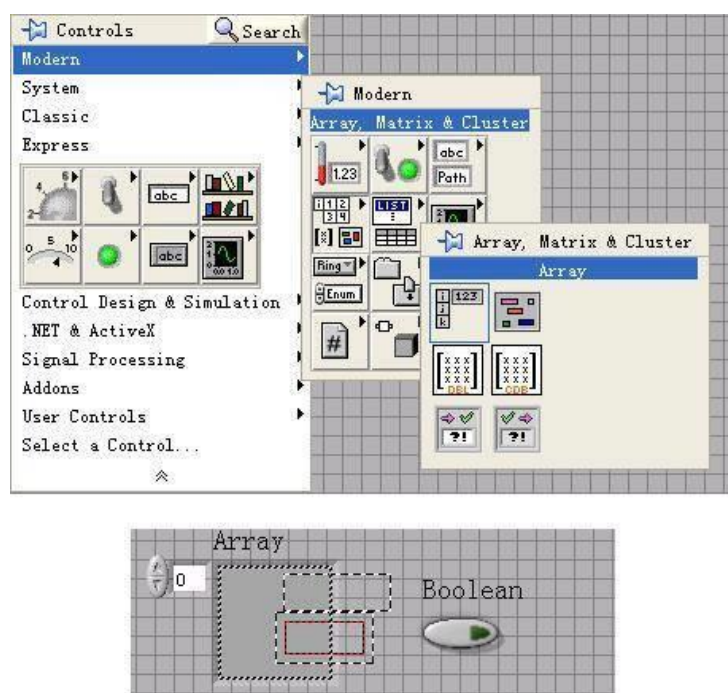
LabVIEW 程序:

需要点击“持续运行”或在程序外加循环框图。



说明:

- 设置参数和发送数据两个子 VI 可以使用已经封装的板卡自带驱动，通过“即时帮助”可了解参数输入方法，也可参照说明书自行构造子 VI
- 因为 Dx 接口既可以作为输入口也可以作为输出口，需要在参数设定.vi 中加以设置；发送数据.vi 中，传送的是表示电平高低的二进制布尔数组（此例为 1 维数组）对应的十进制数字
- 本程序试用了顺序结构，是为了保证参数设置的优先执行，避免错误，设置参数的 DIO 模式节点上点击右键，点击“创建”——“常量”即可创建选择菜单
- 使用了布尔量和数值量转换函数、数组生成函数，请参照即时帮助文档加以了解
- 图中前面板中为布尔开关数组，设置方法如下：



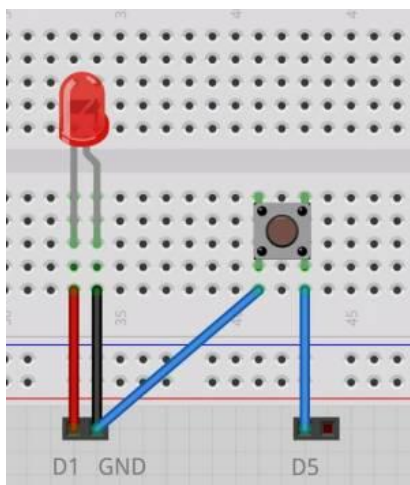
## ②带物理开关 LED:

Dx 接口还可以在输入（信号到计算机）模式工作，所以我们可以可以在电路中加入开关设置，通过开关通断影响输入电平高低，以读取开关量。

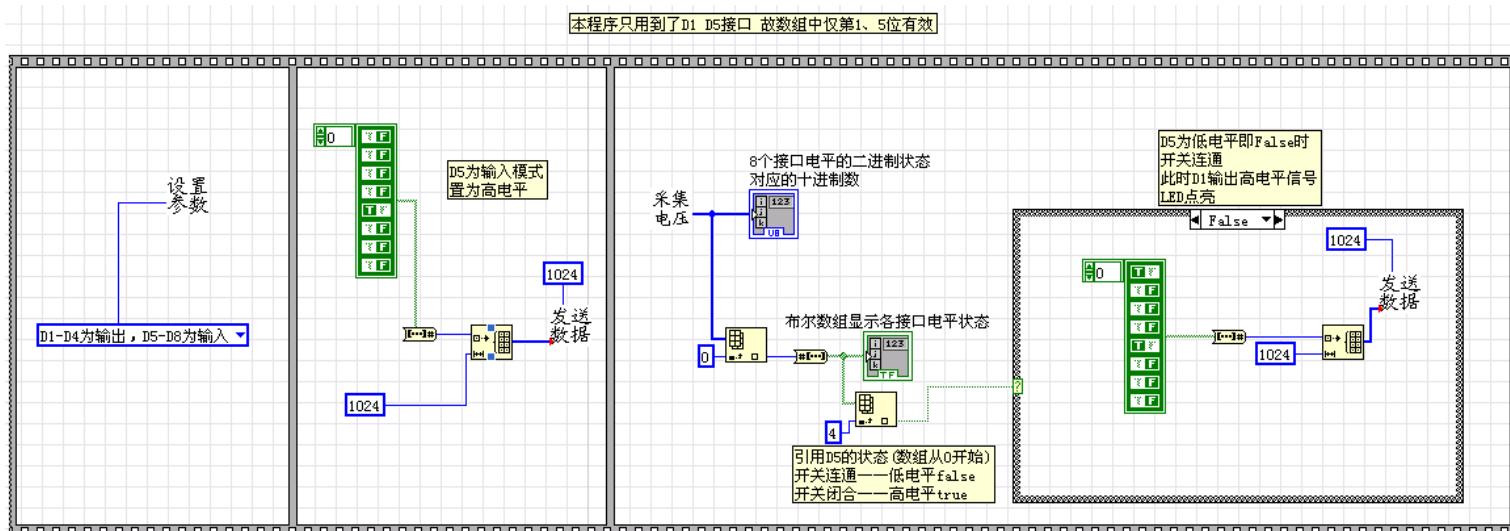
我们依然置 D1-D4 输出（指输出信号到元件），D5-D8 输入（指输入信号到计算机），实际使用的是 D1、D5 接口（见连线图）。

根据板卡说明书第 14 页 unsigned char \* D0 函数的介绍，我们可以了解到，配置 Dx 为输入模式的同时，应对 Dx 置为高电平。这样，初始 D5 置为高电平，开关连通后，配合 GND 接口，D5 变为低电平。也就是说，只要检测 D5 状态即可了解开关闭合情况（开关开：D5 高电平；闭：D5 低电平），进而与前例类似地控制 D1 的输出。

连线图：



LabVIEW 程序：



8个接口电平的二进制状态 对应的十进制数

0 16

布尔数组显示各接口电平状态

0 ● ● ● ● ● ● ● ●

1 2 3 4 5 6 7 8

前四个接口为输入，后四个输出

此时，前面板只是为了查看状态，不做控制功能  
可以发现，当开关接通时，D5变为低电平；闭合时为高电平

Context Help

**Number To Boolean Array**

number ——— Boolean array

Converts an integer or fixed-point number to a Boolean array. If you wire an integer to **number**, **Boolean array** returns an array of 8, 16, 32, or 64 elements, depending on the number of bits in the integer. If you wire a fixed-point number to **number**, the size of the array that **Boolean array** returns equals the word length of the fixed-point number. The 0th element of the array corresponds to the least significant bit of the two's complement of the binary representation of the integer.

[Detailed help](#)

顺序框图中，依次为：①初始参数设定：设置 D1 为输出模式，D5 为输入模

式；②设置输入模式 D5 为高电平；③检查 D5 电平并控制 D1 信号发射其中用到了数组索引函数、判断框图。

使用了数字显示和布尔数组显示两种方式了解各端口状态

### ③带物理开关 LED:

三原色原理:



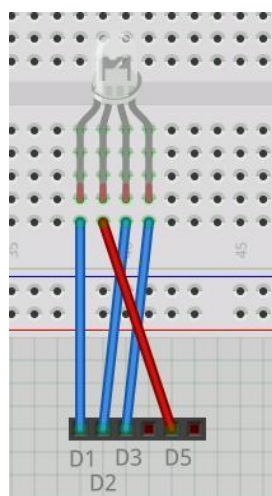
红色 + 绿色 = 黄色  
绿色 + 蓝色 = 青色  
红色 + 蓝色 = 紫色  
红色 + 绿色 + 蓝色 = 白色

三色 LED 元件有四个引脚，第二个接置高电平的 D5 接口（共阳极），事实上另外三个引脚（D1 D2 D3）与阳极组合即是三种颜色的独立 LED 灯

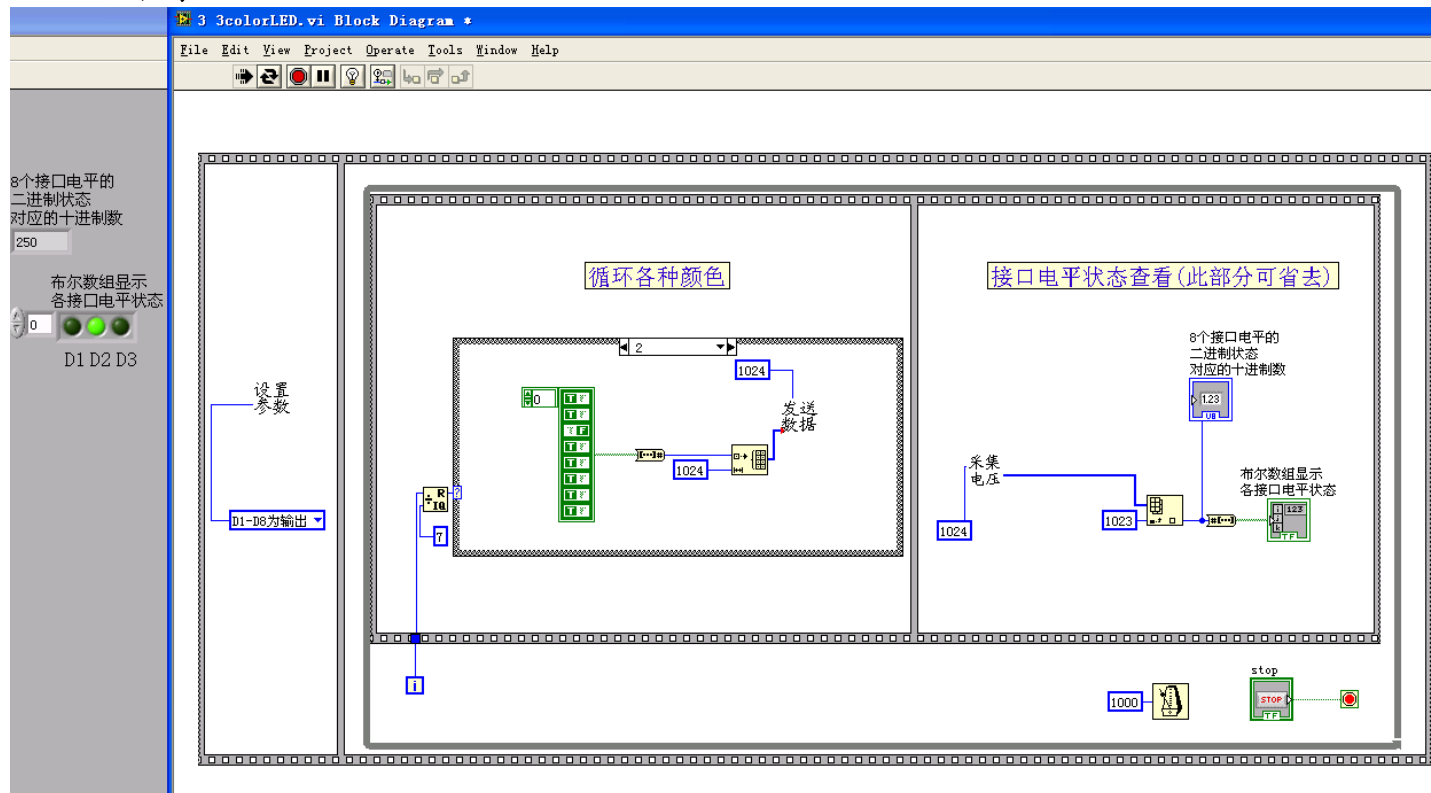
除 D5 接口外另三个引脚输出高电平时与阳极电位相当，无电流通过，该颜色 LED 不亮；设置为低电平时，配合高电位的阳极，有电流通过，该颜色 LED 点亮。若多种颜色点亮，则出现三原色原理所述彩色。

后面将学习 PWM 原理，利用之可以实现三色 LED 的颜色渐变，有兴趣的同学可以尝试一下。

连线图:



LabVIEW 程序：



说明：

- 1、使用了循环框图和选择框图，从而实现七种颜色（D1 D2 D3 各对应高低两个状态——即框图种 T、F 两种状态， $2^8-1=7$  种情况）的跳变  
使用了取余函数（Functions —— Mathematics —— Numeric —— Quotient&Remainder）
- 2、使用了时间控制（框图中钟的标志），每个颜色停留 1000ms
- 3、接口电平查看部分是可以略去的（不影响程序运行），设置的目的是了解串口状态以更好地看清原理。其中试用了“数组索引”，引用了第 1024 位（图中 1023），是因为程序要延时 1000ms，采集电压过程中可能有旧的数据累计，引用最新的（靠后位置的）电压数据保证了电平状态显示的及时性。

## 第六节 蜂鸣器

蜂鸣器有源，是指内部有无震荡发生源

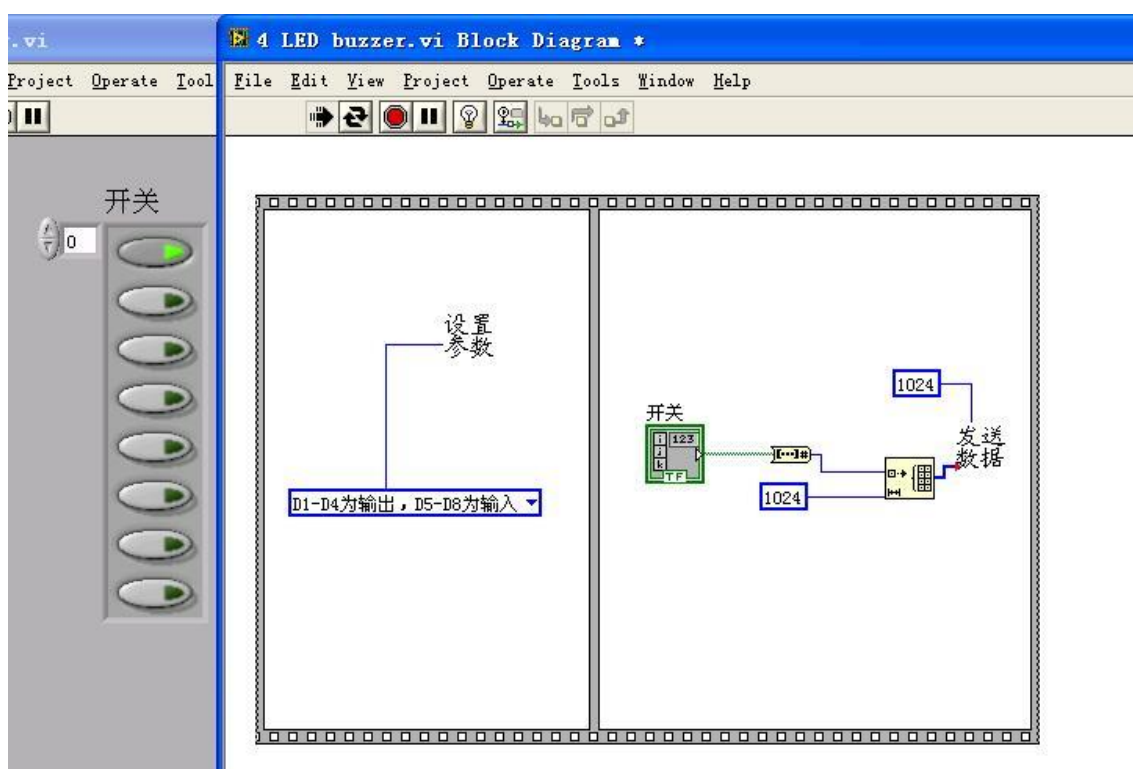
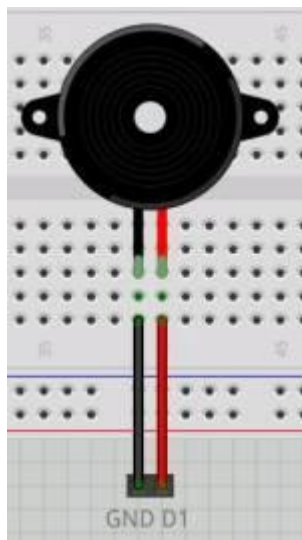
有源蜂鸣器内部置源，通电即工作；无源蜂鸣器需调制频率驱动

本次使用的传感器，有源蜂鸣器稍大一些，两引脚长短有差异，而无源的引脚长度相同

### ①有源蜂鸣器：

使用数字输入输出（DO）控制有源蜂鸣器通断，开关设置使用数字输入输出（DI / DO），与 LED 控制相同。

连线图及程序图：



## ②无源蜂鸣器：

接入 DA3 接口，使用该接口系统默认三角波输出，打开“多功能综合应用程序【MPS-010602】”调制“频道3”频率，驱动无源蜂鸣器工作，试着改变频率值，观察声音变化（1000Hz 左右时较清晰）。

有兴趣的同学还可以参照测试程序及产品手册，使用前面板中 Knob 旋钮图标更加方便的改变频率值，还可以接入电位器，旋转之改变频率从而使无源蜂鸣器发出不同频率的声音



## 第七节 LED 颜色渐变

### ①PWM 简介

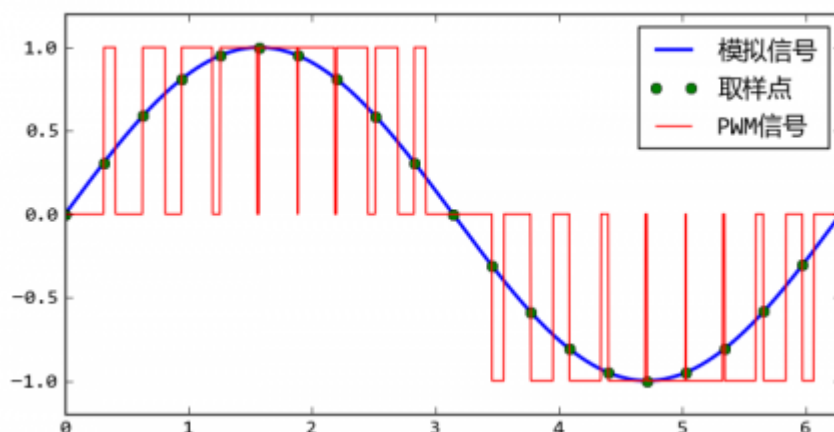
脉冲宽度调制 (Pulse Width Modulation): 对**模拟信号电平**进行数字编码。

事实上, 利用电流电压等模拟信号实现直接调制是直观和方便的, 但其成本和功耗大 (电热散耗), 对噪声敏感, PWM 的一个优点正是从处理器到被控系统信号都是**数字形式**的, 无需进行数模转换, 让信号保持为数字形式从而将噪声影响降到最小。

然而, 数字信号仅能输出 0 或 5V 电压值 (对应 0、1), 故通过高分辨率计数器, 利用脉冲方波信号中占空比实现信号编码

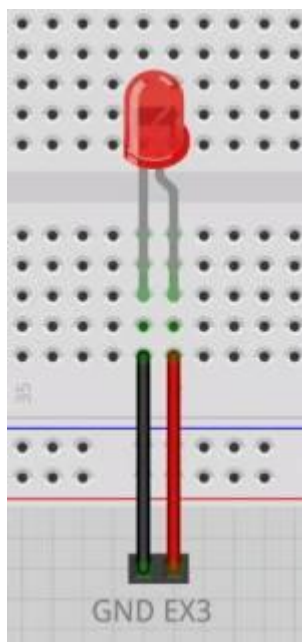
### ②PWM 直观解释

PWM 形式仍是数字的, 输出一个个等周期脉冲, 每个脉冲内占空比不一 (高低电位数比例不一) 可以认为, 一个脉冲等效要输出的模拟信号的一个点, 这个点的幅度计算公式为: **输出电压 = (高电位时间 / 脉冲周期) \* 最大电压**  
本例中, 也可以看做由于等效电流积累效应 (波形图面积), 产生了等效的电压效果。



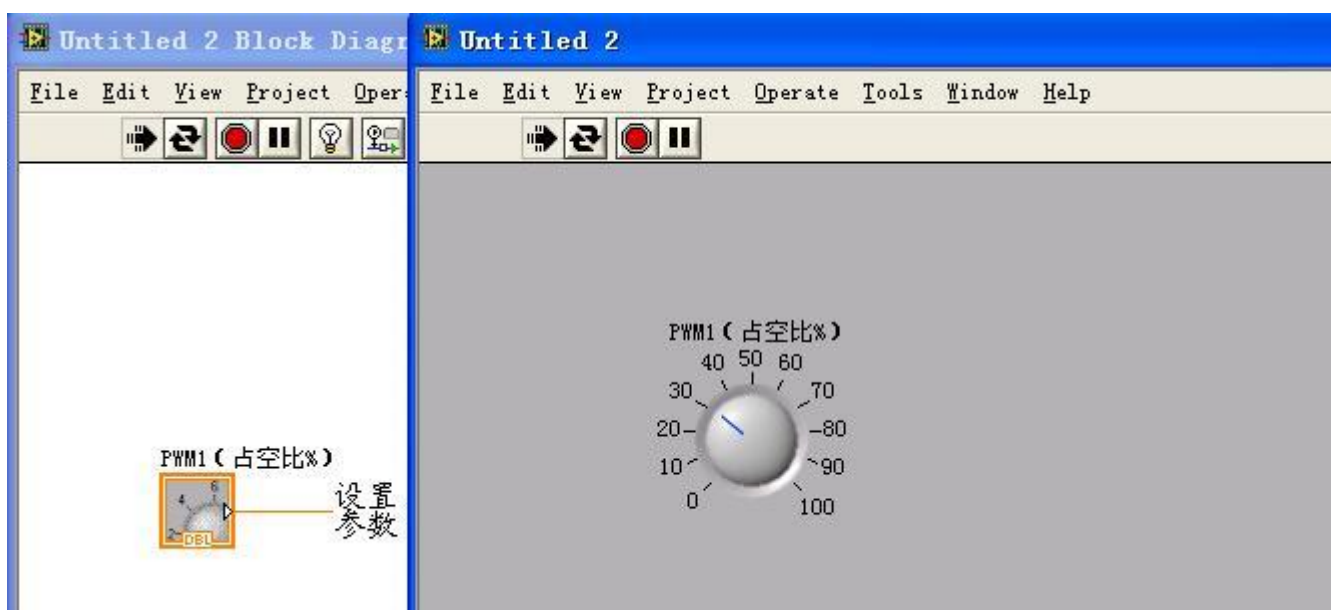
连线图:

(根据采集卡说明书, EX3 接口对应 PWM1 输出)



**程序图：**

调用采集卡自带封装驱动函数 SetPara.vi，即可设定占空比，从而调节 LED 亮度。



板卡上有蓝色 LED，其亮度表示了 PWM 占空比大小，调节过程中可以留意观察一下。

## 第八节 舵机控制

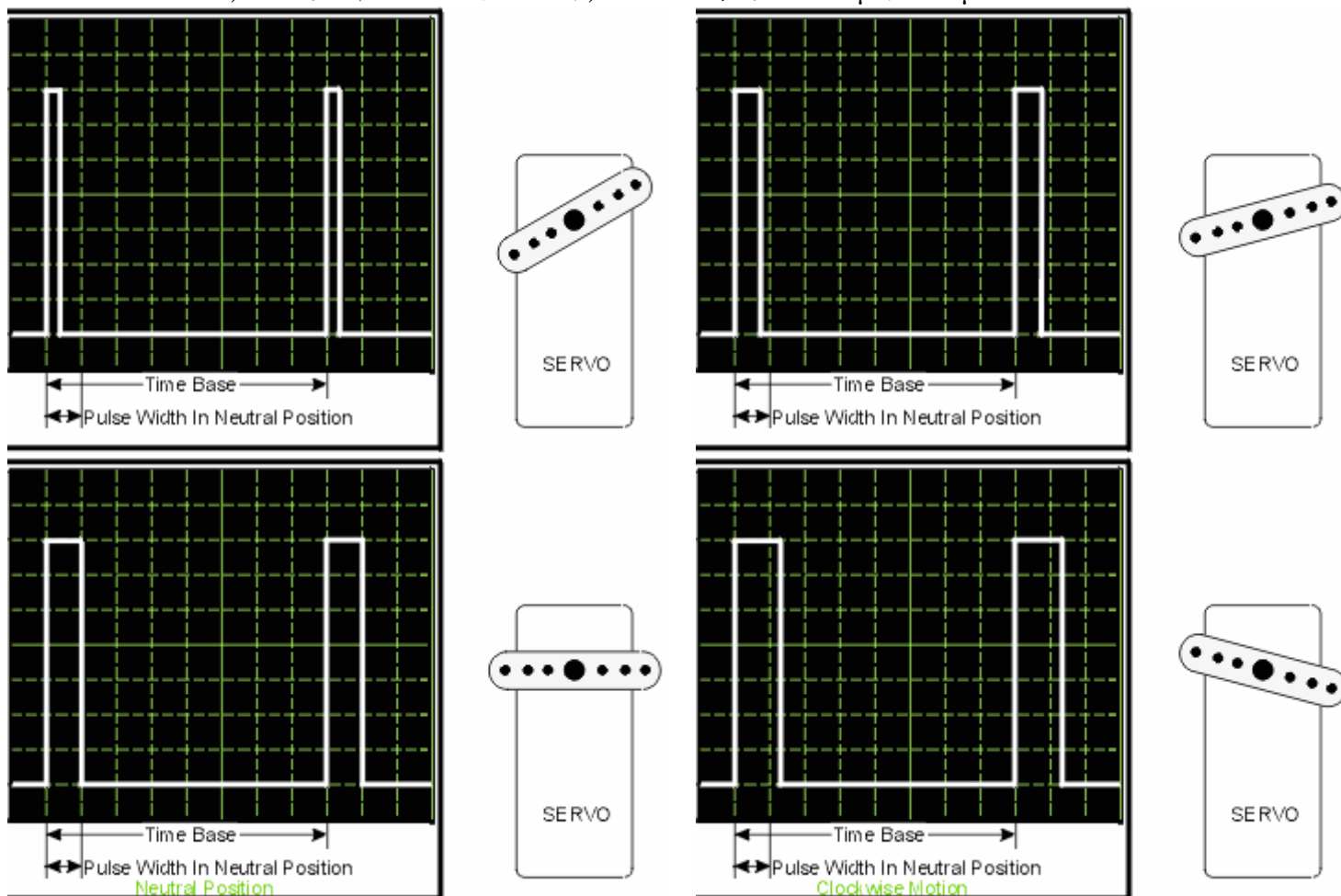
### ①基本原理

舵机是一种位置伺服的驱动器，广泛用于需要控制角度的系统中。

本实验采用 9g 舵机，其具有三个引脚：黄色——信号线；红色——电源；棕色——地线。

工作过程中，控制器发出 PWM 信号，舵机内部有基准信号可与控制器信号比对，驱动芯片通过压差大小及正负控制电机正负旋转。

本实验舵机采用比对用基准为标准 PWM 脉宽  $0.5\text{ms} \sim 2.5\text{ms}$ ，周期为  $20\text{ms}$ ，对应频率  $50\text{Hz}$ ，为了与  $0 \sim 180^\circ$  转角对应，我们取  $2.48\text{ms}$  为上限，得到整数间隔  $(2.48 - 0.5)\text{ms} / 180 = 11\mu\text{s}$ ，进而，脉宽相应改变  $11\mu\text{s}$ ，变化电机转动  $1^\circ$ 。反之，当我们希望电机转动  $1^\circ$  时，占空比需要变化  $11\mu\text{s}/1980\mu\text{s}$ 。



(本实验电压要求较高，故程序不能正常运行时不要过度伤心，先查看下电源情况，同时用示波器查看自己的 PWM 信号输出电压是否正常、周期是否正确，波形正确的情况下应当对自己的程序保持信心!)

(建议不要使用锂电池，因为其运转耗电量太大，锂电池容量不足，可以

使用类似 arduino 板卡的 5V 输出，这样的电压稳定持续)

## ②板卡自带 PWM 信号说明

MPS-010602 采集卡可以输出 PWM 信号，前面 LED 亮度渐变实验已经利用之，根据 MPS 说明书第 12 页，采样率不同 PWM 输出将对应两种频率

1: 时基为 24M, 16bit 输出 ——  $24 \times 10^6$  次计数为 1s, 以此为时间基准, 每次输出占  $2^{16} = 65536$  次, 故信号频率为  $24 \times 10^6 \div 65536 = 366 \text{ Hz}$

2: 时基为 2M, 8bit 输出, 信号频率为  $2 \times 10^6 \div 2^8 = 7.8 \text{ kHz}$

显然, 这两种频率与舵机使用的标准频率均不匹配, 我们先用自行构建 PWM 输出。同学们也可以将 GND 和 EX3 信号接出到示波器, 实际观察信号频率, 这在参数位置的情况下是便捷和直观的。

下面, 将介绍利用模拟信号输出、数字信号输出构造 PWM 信号的方法。

信号输出在科研实践和工业中是常用的, 例如, 光学实验室中可以使用 labVIEW 控制脉冲信号形状和输出, 达到精确控制光信号的目的。所以, 下面介绍的两种信号产生方式是常用和重要的。

## ③模拟信号构造 PWM

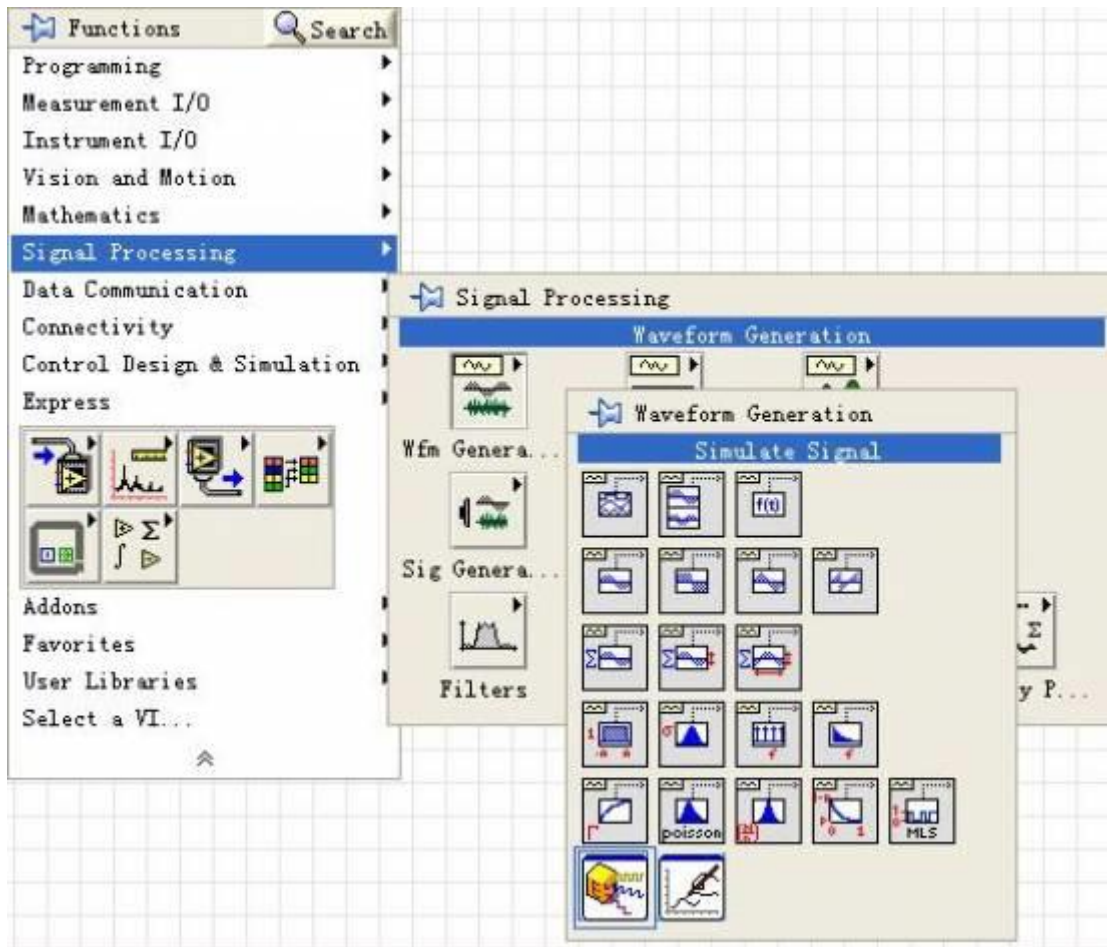
如下图, 选择信号生成函数, 选择模式为“方波”, 频率“50Hz”, 采样率 50000, 采样数 1024 (以上参数请严格设置)。

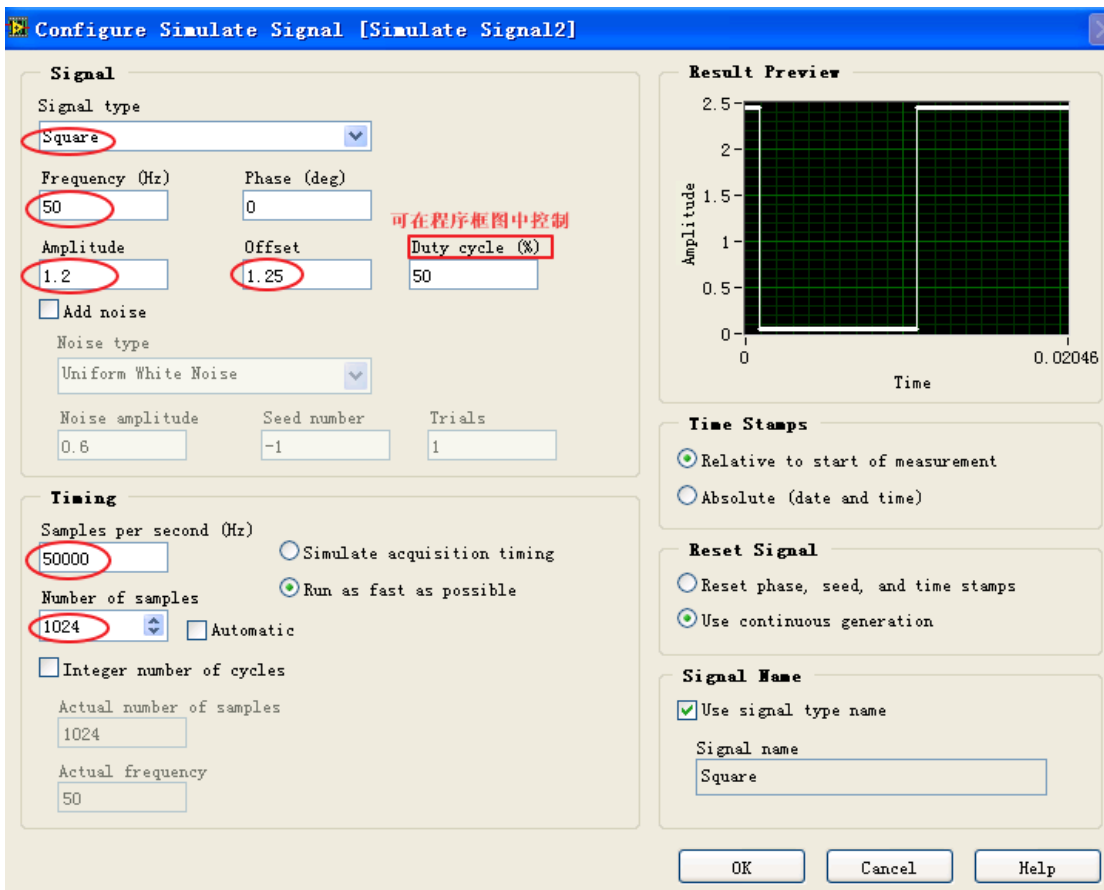
模拟输出电压范围: 0-2.5V, 置“偏置 offset”为 1.25V “幅度 Amplitude”为 1.2V, 则对应输出电压为  $1.25 \pm 1.2$  为 0.05-2.45V。

注意到有“占空比 Duty Cycle (%)”一栏, 这正是我们控制舵机的关键所在。

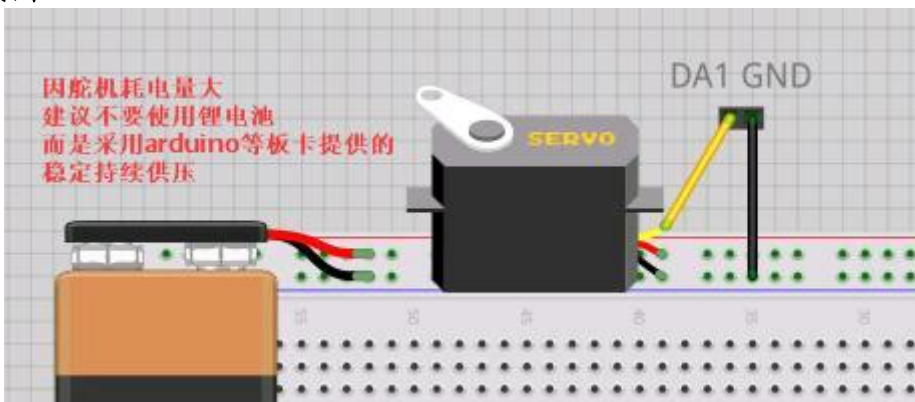
如前所述, 标准 PWM 脉宽为 0.5ms~2.5ms, 周期为 20ms, 我们只要控制占空比在 2.5%到 12.5%之间, 就可以对应 0-180°角度

对应关系为 占空比 =  $\left( \left( \text{angle} \div 180 \right) \times 2 \text{ ms} + 0.5 \text{ ms} \right) \div 20 \text{ ms} \times 100$ , 最后  $\times 100$  是因为输入格式要求为“XX%”

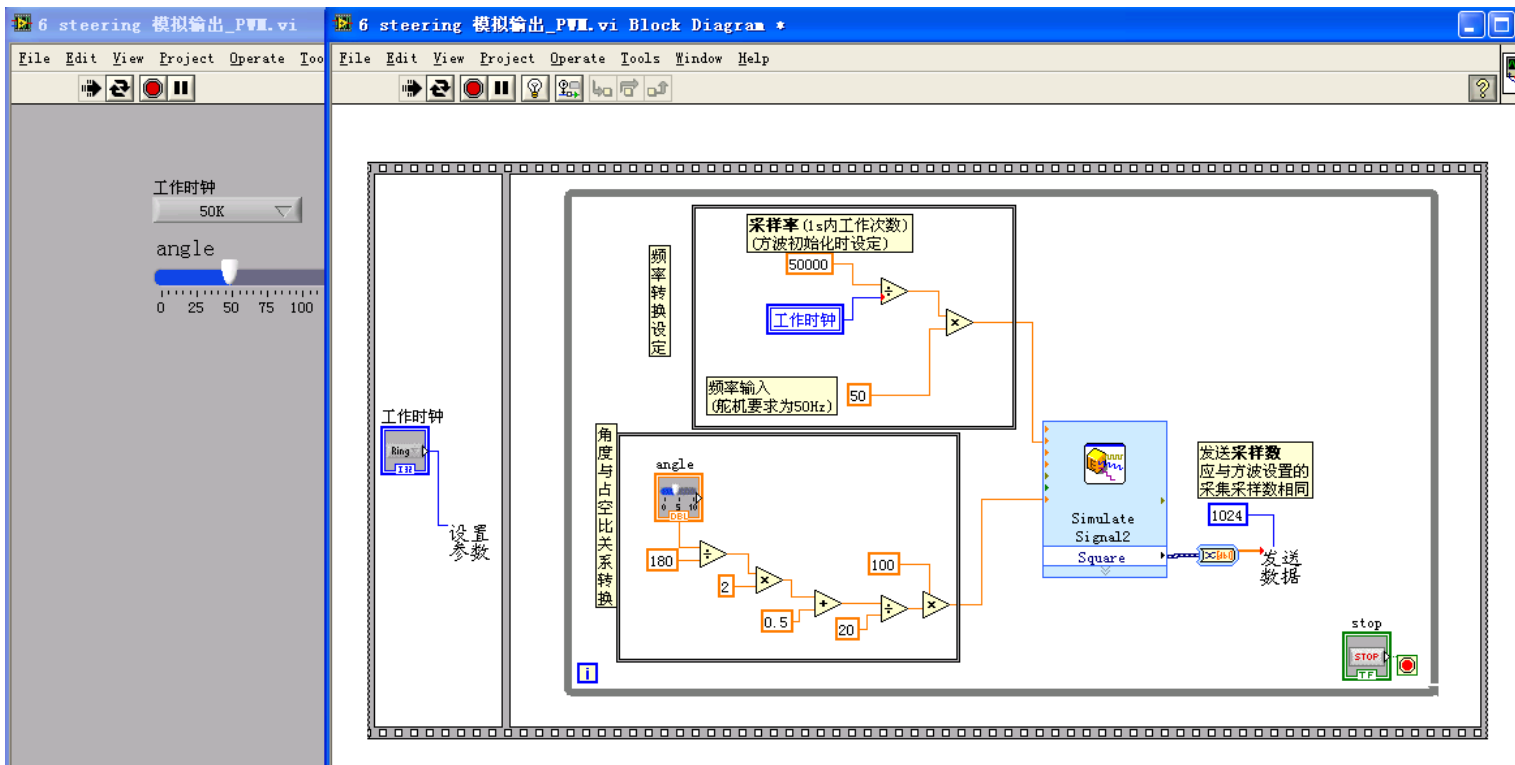




连线图:



程序图:



**说 明:**

当工作时钟和采样率一样时，“频率转换设定”是无用的；但二者不一时，框图中“频率转换设定”部分保证了波形的不变，有兴趣的同学可以思考原因，还可以外接示波器观察波形不进行转换时波形的变化(此时与舵机基准 PWM 信号不再匹配)。

框图中参数设置处“工作时钟”是在该接口“右键单击——新建——控制量 (Control)”设置的；“频率转换设定”的“工作时钟”是由“右键单击后面板设置参数处“工作时钟”图标——新建——局部变量”设置的，二者值同步变化，创建好之后右键设置为“从其读取值”。

发送数据处的采样数设置应当与方波初始设置相同。

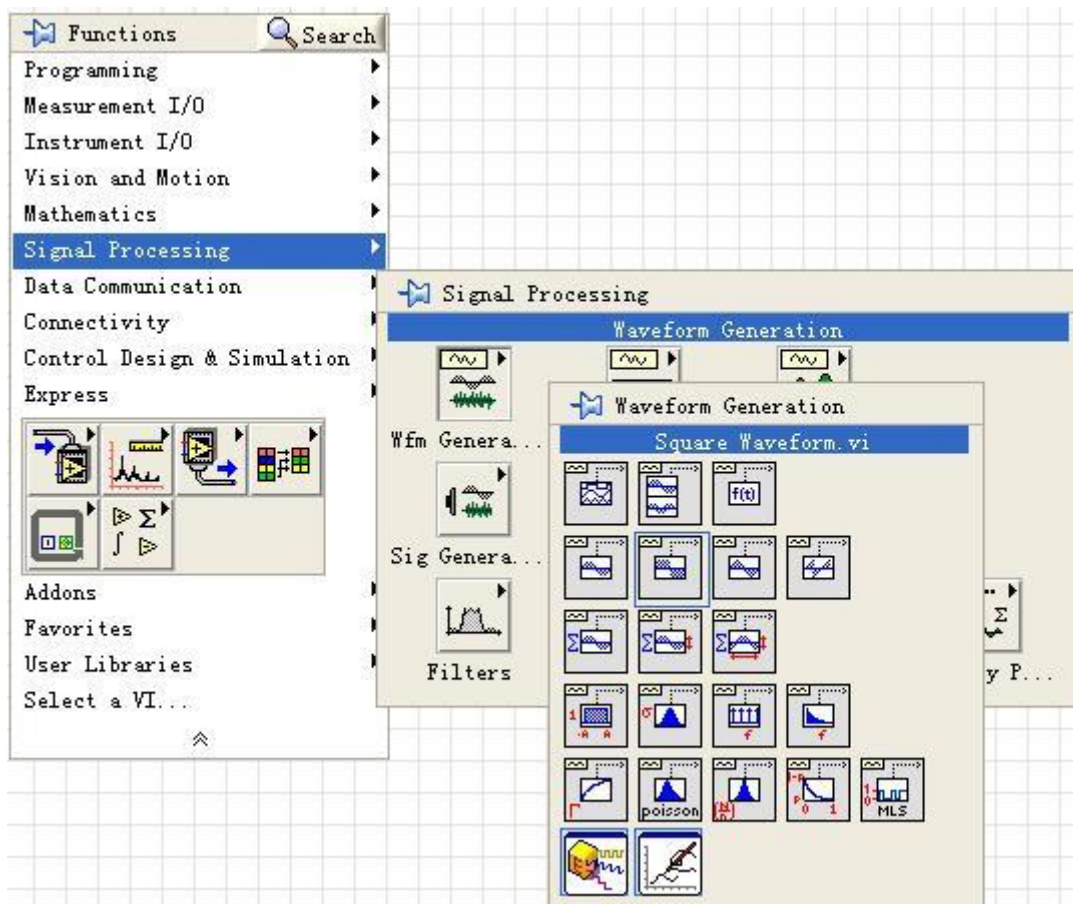
**③数字信号构造 PWM**

我们使用数字接口，输出高低电平，相对用模拟信号去构造 PWM，这更接近 PWM 的本意上面的例子已经成功运行了，接线相同，我们本次的目标是用数字接口输出和上例完全一样的波形信号！

甚至，还可以使用信号发生器直接调制方波信号，改变占空比模拟 PWM 来控制舵机！

**程序说明:**

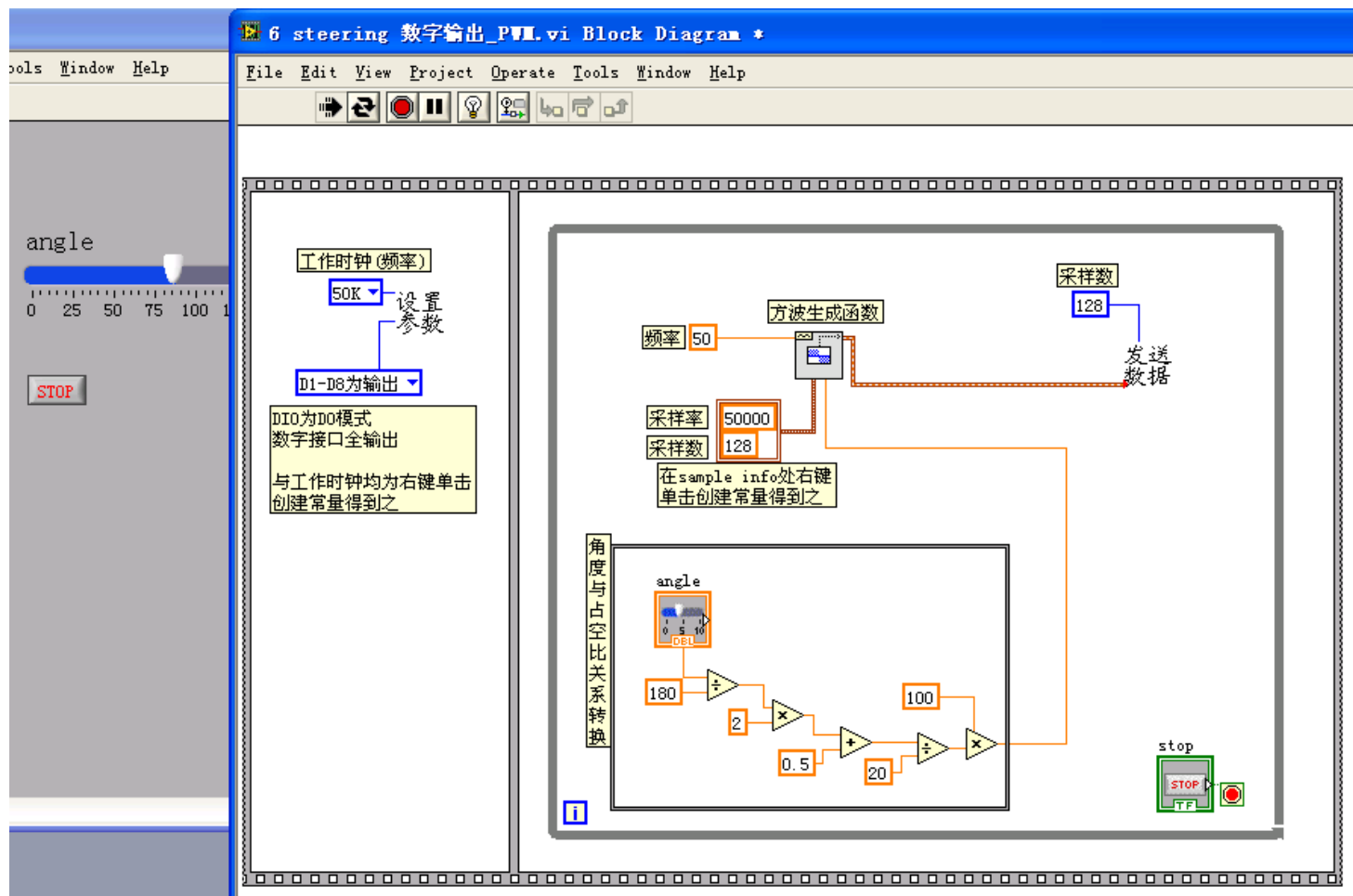
右键单击参数设置“DIO 模式”，“创建常量”，将数字接口均设为输出。与模拟信号调制 PWM 不同的是此次使用信号端口 D1 输出，且利用了“方波生成函数”，设置方法如下：



要注意的是，与前例模拟信号调制 PWM 一样，要特别当心采样率与工作时钟的匹配，以及生成信号采样数与发送信号采样数的匹配性。

另外，数字接口只能输出 3.3V 高电平和 0V 低电平。





效果验证:

连线图与前图基本一致, 只需要把 DA1 接口换做 D1 即可。

原则上, 只要本次输出控制信号波形与模拟信号方式完全相同就可以认为成功了, 用示波器连接两种输出端口 (D1、DA1), 同时显示在示波器上, 比较波形以验证效果。

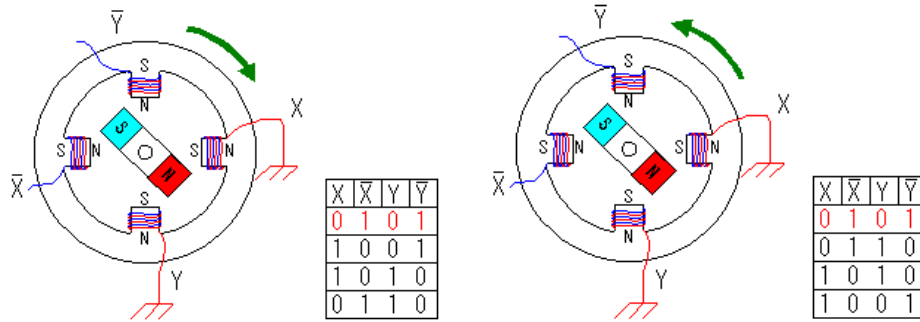
## 第九节 步进电机

### ①基本原理

步进电机是将电脉冲信号转变为角位移或线位移的开环控制无步进电机件。其过载性好, 以步为单位控制方便, 结构简单, 且可以疾始和骤停。

步进的含义是指, 每一个脉冲信号触发一个固定角度的旋转 (步进角)。利用脉冲个数来控制角位移量; 通过控制脉冲频率来控制电机转动的速度和加速度。

通常, 电机的转子为永磁体, 通过配置四个定子绕组的电流组合顺序, 磁场改变, 驱动转子转动与磁场相应方向。这也是为什么在步进电机不失步情况下, 误差不受之前每一步影响, 即不累计, 一圈中的误差仅受最后一步影响。



(图片和部分介绍引用自 [wiki.emsym](http://wiki.emsym), 左为正转, 右为逆转)

本实验采用 **28BYJ-48 型** 步进电机, 其中 28 表示电机直径 28 毫米, B 表示步进电机, Y 表示永磁, J 表示带减速箱, 48 表示可以四拍和八拍运行。

其他参数为: 5V 直流供电; 减速比为 1: 64, 步进角为 5.625/64 度, 即转动 1 圈, 需要  $360 \div (5.625 \div 64) = 4096$  个脉冲信号

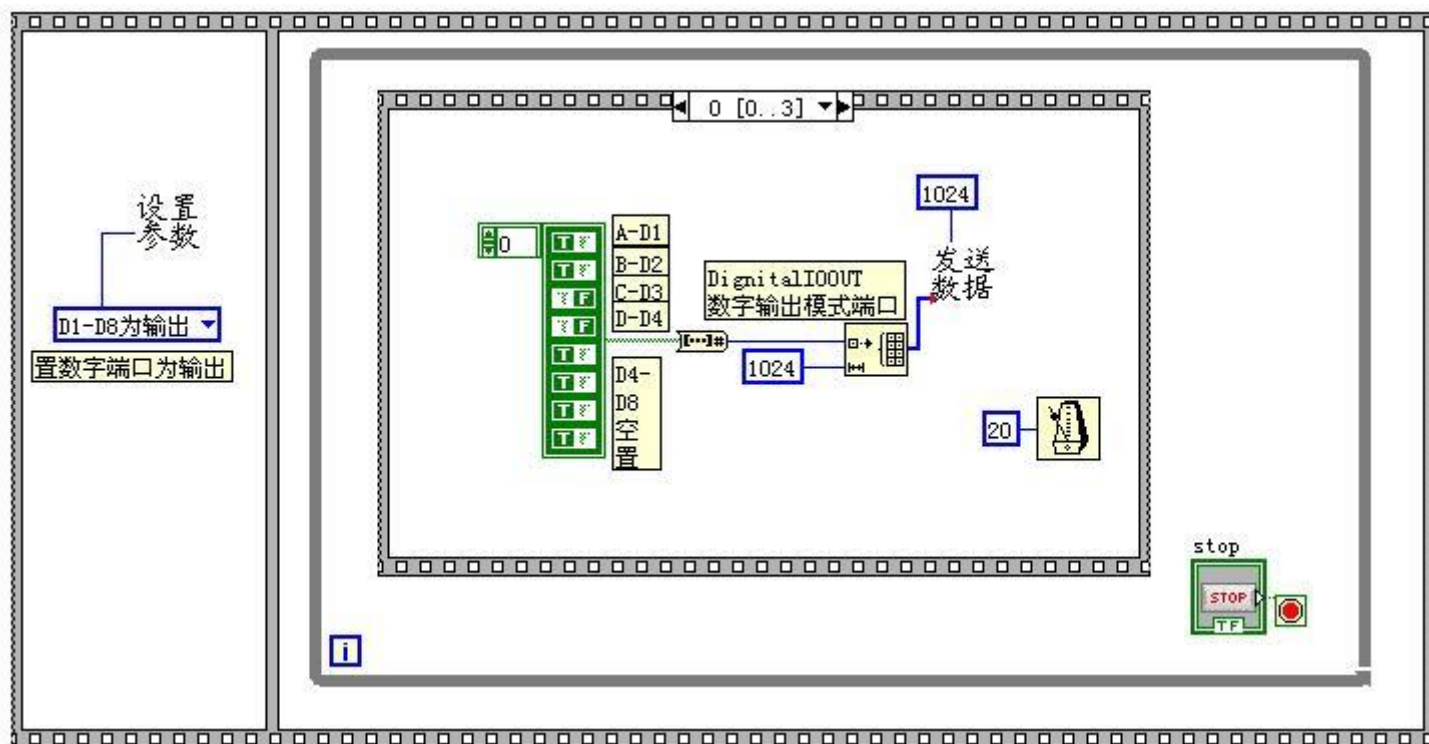
四相八拍驱动方式运行更平稳, 实验中采取该驱动方式, 与上面示意图一致, 而连线 A、B、C、D 对应图中 X', Y', X, Y, 即驱动方式为: 正向——AB-BC-CD-DA; 逆向——BA-AD-DC-CB。

## ②程序说明

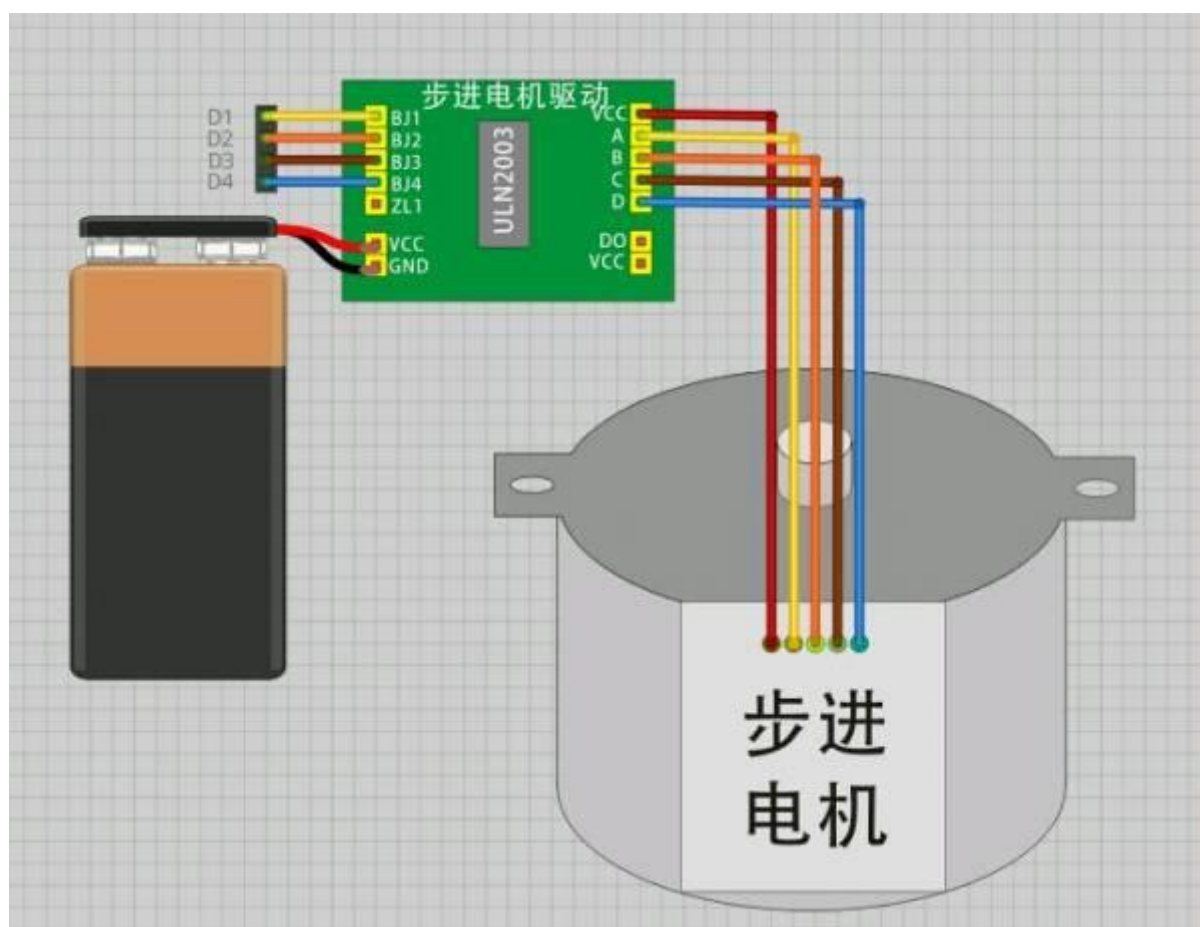
使用采集卡 D1-D4 四个数字端口的数字高低电平进行驱动。

程序中使用了层叠式顺序结构。未展示的其他三个结构相应地按上面说明置 A、B、C、D 通断。

此例为**正向匀速**转动, 类似地可以改变电位顺序以逆向转动, 或者可以延迟时间以改变转速, 但是该电机有 64 倍减速器, 转速较慢, 改变效果不明显。



连线图：电机为四相五线，五线中四个对应相位，另一个接阳极(VCC)。



## 第三课 分析软件

(其中涉及的实践中应用较广的信号输出的一些概念, 在 [wiki](#) 中[舵机控制](#)一节已有所简单介绍, 可以先将其了解清楚, 再查看本章介绍。

在课程范围内, 本章内容[了解即可](#)。)

### 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箱, 借助这些分析软件包, LabVIEW 可以具有更加强大的分析功能。这些分析工具箱包括:

- 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, 或者与我公司联系。

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：

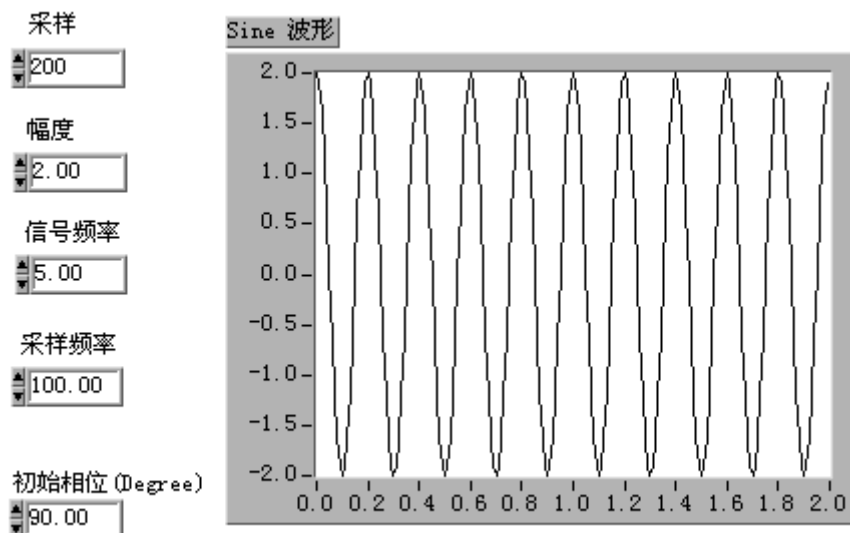
$$\text{数字频率} = \text{模拟频率} / \text{采样频率}$$

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

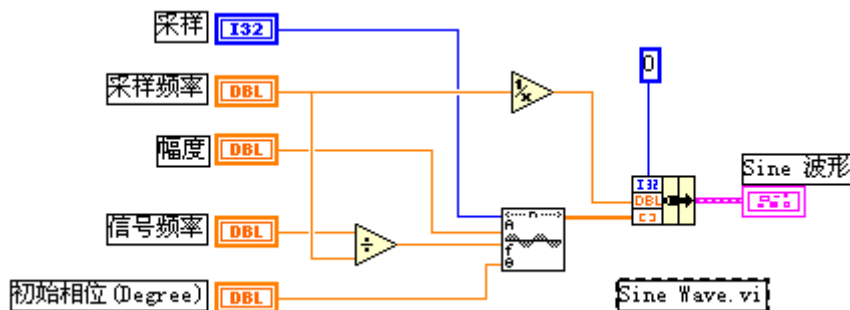
**目的：**产生一个指定频率的正弦波。

**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  $((N \times 50) - 90) \text{ Hz} > 0$ ，即为  $(100 - 90) = 10\text{Hz}$ ，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)

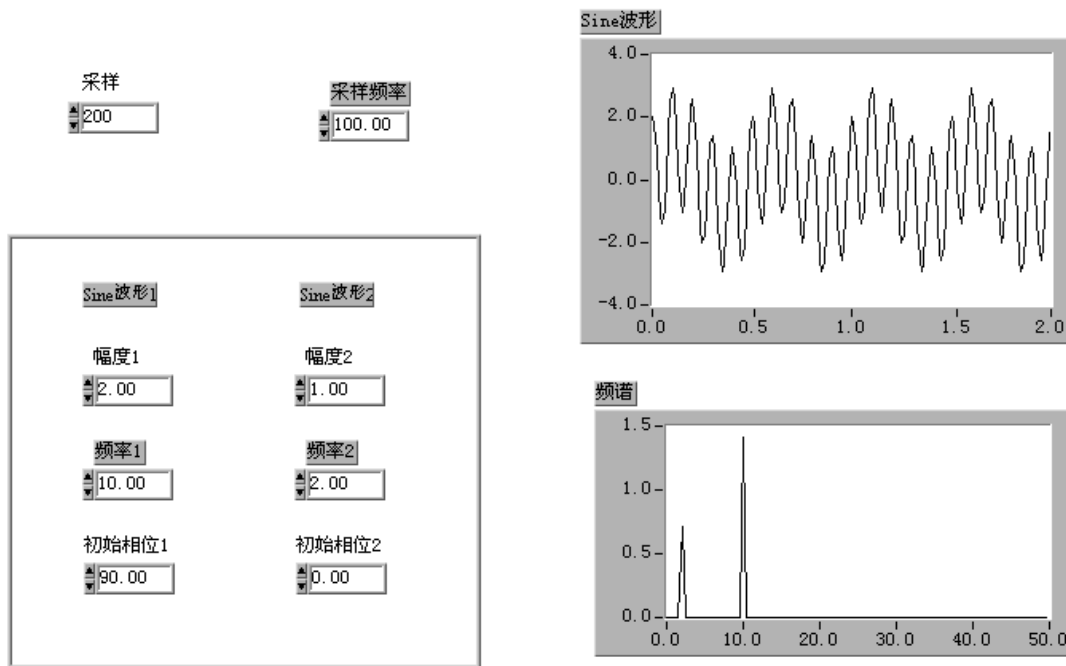
## 第四节 信号处理

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。

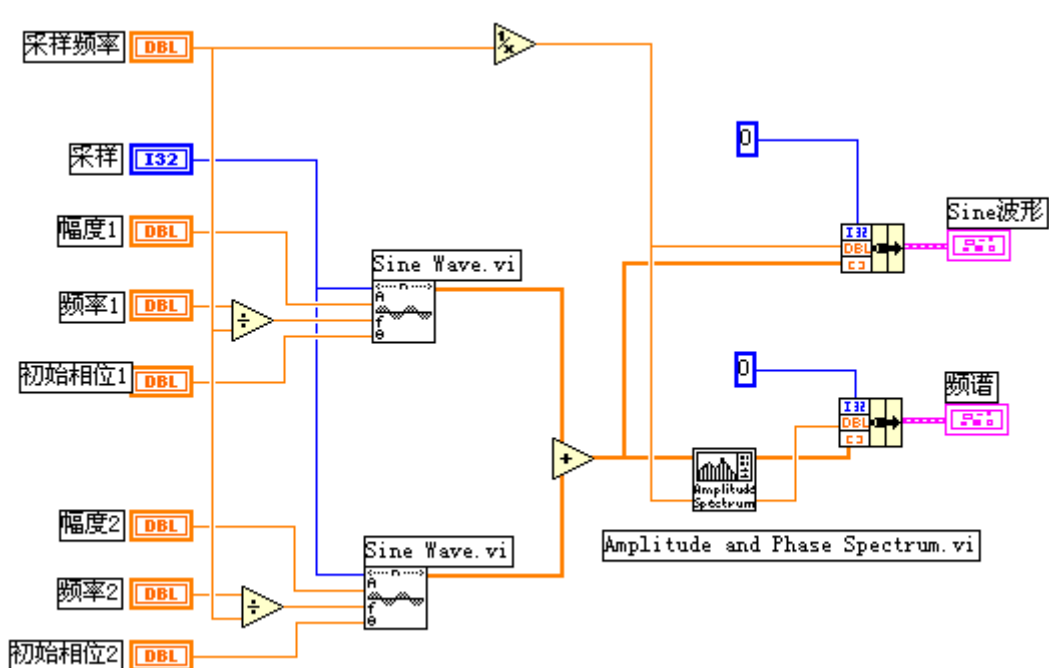
**前面板：**



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）。



波  
|  
对



## (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：

$$\text{dB} = 20 \times \log (A_o(f) / A_i(f))$$

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  $-0.02\text{dB}$ ，则有：

$$\begin{aligned} -0.02 &= 20 \times \log (A_o(f) / A_i(f)) \\ A_o(f) / A_i(f) &= 10^{\wedge} (-0.001) = 0.9977 \end{aligned}$$

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  $-60\text{dB}$ ，则有：

$$\begin{aligned} -60 &= 20 \times \log (A_o(f) / A_i(f)) \\ A_o(f) / A_i(f) &= 10^{\wedge} (-3) = 0.001 \end{aligned}$$

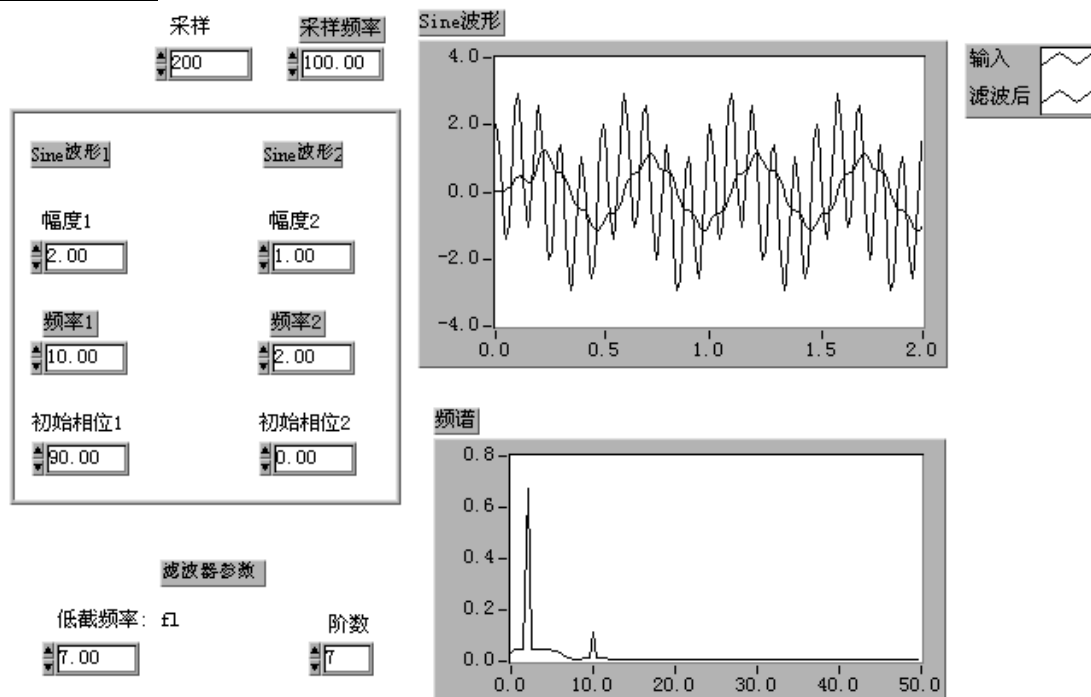
输出幅值仅是输入幅值的千分之一。

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

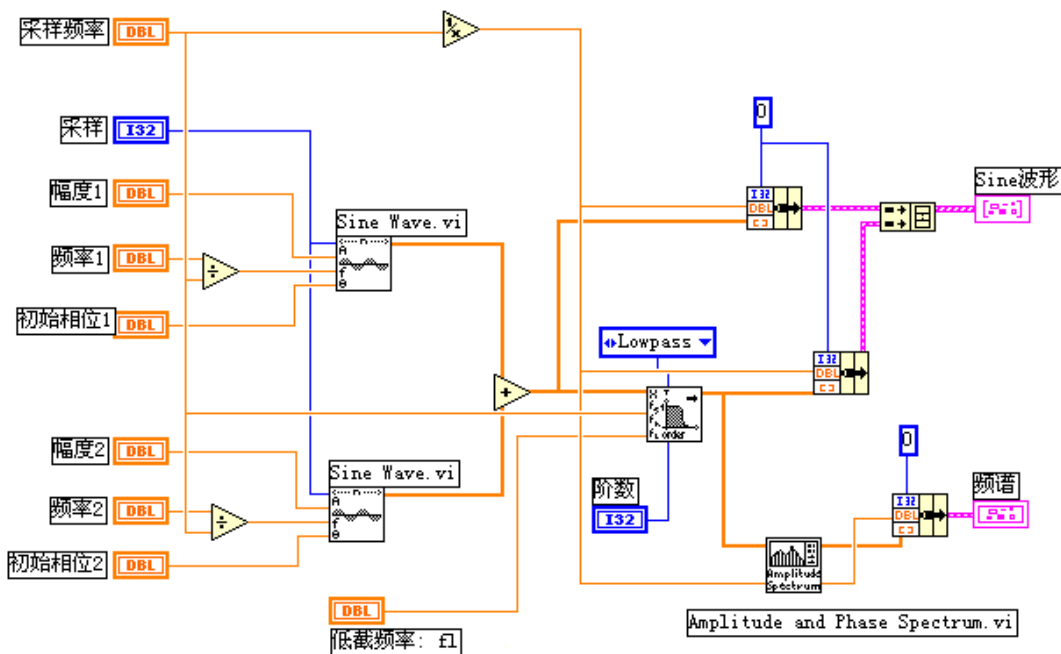
**前面板：**



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  $y[i]=a_0+a_1*X[i]$
- \* 指数拟合 – 把数据拟合为指数曲线  $y[i]=a_0*\exp(a_1*X[i])$
- \* 多项式拟合 – 把数据拟合为多项式函数:  

$$y[i]=a_0+a_1*X[i]+a_2*X[i]^2\dots$$
- \*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,

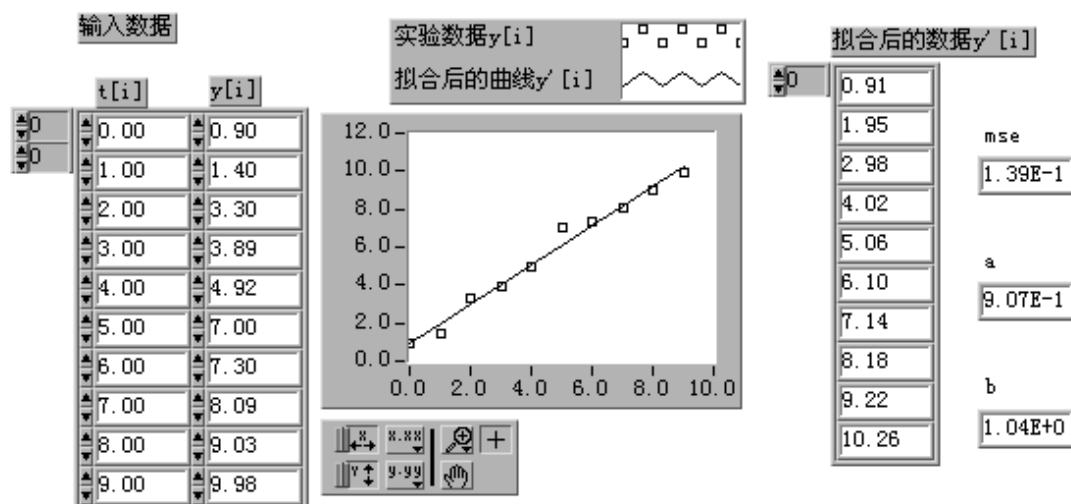
但是它可以采用一些特殊算法以提高拟合处理的速度和精度。

\*General Levenberg-Marquardt 拟合 – 把数据拟合为公式  $y[i]=f(X[i], a_0, a_1, a_2\cdots)$ 。其中  $a_0, a_1, a_2\cdots$  是参数。这种方法是最通用的方法，它不需要  $y$  与  $a_0, a_1, a_2\cdots$  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

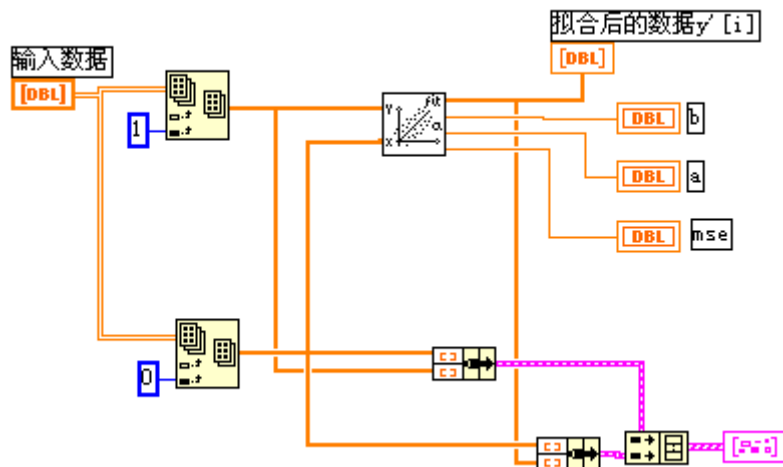
**目的：**对指数关系数据进行线性拟合

**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. 输入数据是一个二维数组，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。

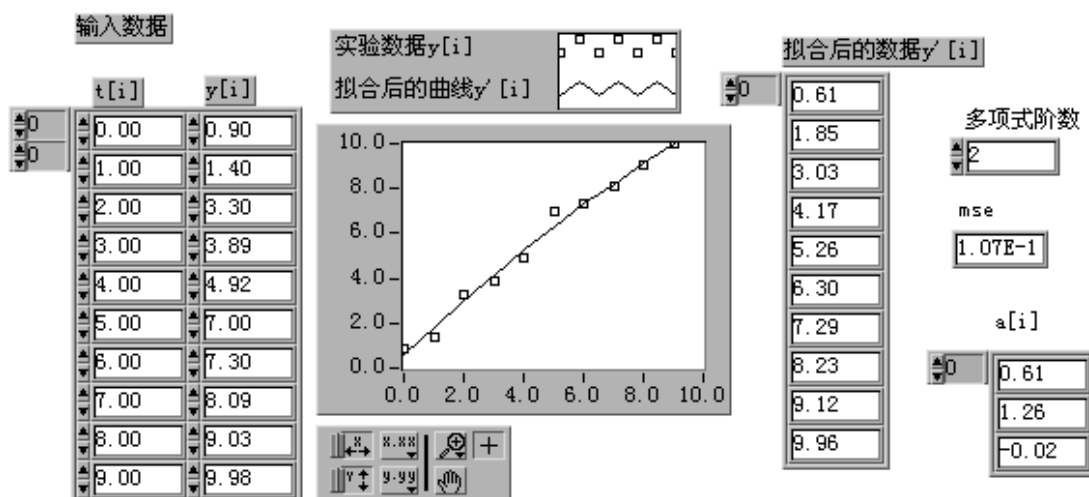
### (练习 4-4 结束)

### 练习 4-5

**目的：**对指数数据执行多项式拟合

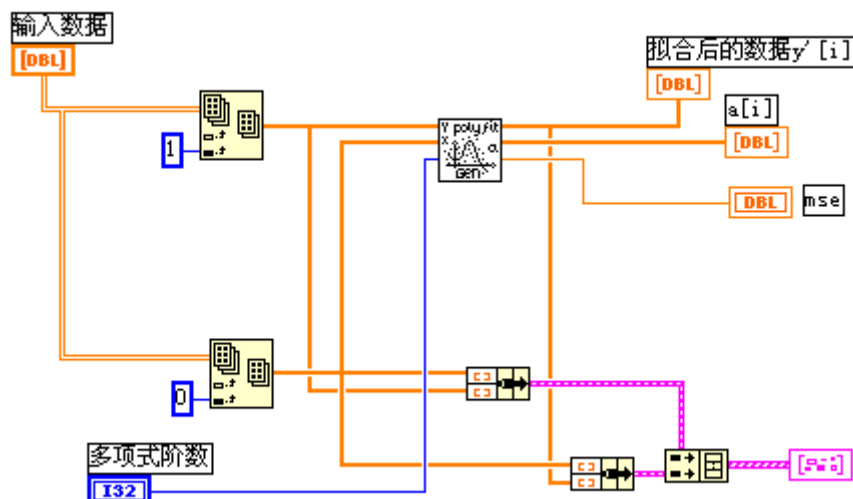
**前面板：**

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：  
 $y[i]=a_0+a_1*t[i]+a_2*t[i]^2\dots$
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 a0, a1, a2 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)

## 第四课 实用工具软件包 (了解即可)

### 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 的本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这个工具箱。

### **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，在这些工作环境下，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方程，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 (VirtualBench-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，VirtualBench-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。

本讲义内容和资料部分来自于北京中科泛华测控技术有限公司的教学资料《计算机虚拟仪器图形编程 LabVIEW 实验教材》第一、三、四章内容,对第一章使用的例子进行了全面的简化和优化,并在较新的 LabVIEW 版本 (8.5) 上进行了操作实践与截图展示,全部新版本源程序可在实验中心网站 (<http://phylab.fudan.edu.cn/doku.php?id=home:whyx:proj:labview0>) 获取;部分介绍性语言得到了**重构和新的解释**;位置有变化的**模块**进行了**更新**;添加了前言部分。

第二章参照单片机 Arduino 对传感器件和步进电机、舵机等简单仪器控制的相关教程进行编写,设计了使用面很广的几个实验案例,将其移植到使用**数据采集卡 MPS-010602**的数据采集与控制,全部实验内容已经经过全面测试,全部源程序和连线截图可在实验中心网站获取。

本讲义上的内容按下列许可协议发布:

[CC ttribution-Noncommercial-Share Alike 3.0 Unported](http://creativecommons.org/licenses/by-nc-sa/3.0/)

首次生成时间:2014 年 7 月 23 日星期三;

更新人:李楠 ([nanli12@fudan.edu.cn](mailto:nanli12@fudan.edu.cn)) 感谢**俞熹老师**思路指导和技术支持!

第一次更新:2014-10-28 更新第 28 页红字部分说明(李楠)

---