

源于实践 例例精彩

LabVIEW 虚拟仪器

基础与入门



110例

◎ 李江全 主编 ◎ 刘长征 王玉巍 党媚 副主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

LabVIEW 虚拟仪器 基础与入门 110 例

李江全 主 编
刘长征 王玉巍 党 媚 副主编

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书从实际应用出发,通过 110 个典型实例系统地介绍了虚拟仪器编程语言 LabVIEW 的程序设计方法及其应用技术。主要内容有 LabVIEW 程序设计基础、数值型数据、布尔型数据、字符串数据、数组数据与矩阵、簇数据、数据类型转换、程序流程控制、变量与节点、图形显示、文件 I/O、界面交互及子程序设计等。提供的实例由实例基础、设计任务和任务实现等部分组成,并有详细的操作步骤。

本书内容丰富,论述深入浅出,有较强的实用性和可操作性,是测控仪器、计算机应用、电子信息、机电一体化、自动化等专业学生和相关行业工程技术人员学习虚拟仪器技术的入门书籍。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

LabVIEW 虚拟仪器基础与入门 110 例 / 李江全主编. —北京:电子工业出版社, 2019.3

ISBN 978-7-121-35679-7

I. ①L… II. ①李… III. ①软件工具—程序设计 IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2018)第 280864 号

策划编辑:陈韦凯

责任编辑:陈韦凯 特约编辑:李 姣

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1 092 1/16 印张:14.75 字数:378 千字

版 次:2019 年 3 月第 1 版

印 次:2019 年 3 月第 1 次印刷

定 价:59.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:chenwk@phei.com.cn, (010) 88254441。

前 言

随着微电子技术和计算机技术的飞速发展，测试技术与计算机深层次的结合正引起测试仪器领域里一场新的革命，一种全新的仪器结构概念导致了新一代仪器——虚拟仪器的出现。它是现代计算机技术、通信技术和测量技术相结合的产物，是传统仪器观念的一次巨大变革，是产业发展的一个重要方向，它的出现使人类的测试技术进入了一个新的发展纪元。

虚拟仪器在实际应用中表现出传统仪器无法比拟的优势，可以说虚拟仪器技术是现代测控技术的关键组成部分。虚拟仪器由计算机和数据采集卡等相应硬件和专用软件构成，既有传统仪器的特征，又有一般仪器不具备的特殊功能，在现代测控应用中有着广泛的应用前景。

作为测试工程领域的强有力工具，近年来，虚拟仪器编程语言 LabVIEW 得到了业界的普遍认可，并在测控应用领域得到广泛应用。

本书从实际应用出发，通过 110 个典型实例系统地介绍了 LabVIEW 的程序设计方法及其应用技术，主要内容有 LabVIEW 程序设计基础、数值型数据、布尔型数据、字符串数据、数组数据与矩阵、簇数据、数据类型转换、程序结构、变量与节点、图形显示、文件 I/O、界面交互及子程序设计等。提供的实例由实例基础、设计任务和任务实现等部分组成，并有详细的操作步骤。

考虑到 LabVIEW 各版本向下兼容，且各版本编程环境及用法基本相同，因此为使更多读者能够使用本书程序，我们选用 LabVIEW 8.2 中文版作为主要设计平台，并将 LabVIEW 2015 中文版与其不同的地方予以指出。

本书内容丰富，论述深入浅出，有较强的实用性和可操作性，是测控仪器、计算机应用、电子信息、机电一体化、自动化等专业学生和相关行业工程技术人员学习虚拟仪器技术的入门书籍。

本书由石河子大学李江全编写第 0、第 1 章，刘长征编写第 2、第 3 章，吕琛编写第 5、第 8 章；新疆工程学院王玉巍编写第 4、第 9 章；西安航空职业技术学院党媚编写第 6、第 7 章；空军工程大学李丹阳编写第 10、第 11 章。

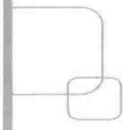
由于编者水平有限，书中难免存在不妥或错误之处，恳请广大读者批评指正。

编著者

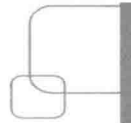
目 录

入门基础篇

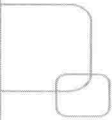
第 0 章 LabVIEW 程序设计基础	1
0.1 LabVIEW 的特点及应用	1
0.2 LabVIEW 中的基本概念	2
0.3 LabVIEW 的前面板设计	5
0.4 LabVIEW 的数据操作	10
0.5 体验 VI 程序设计	14
0.6 VI 的调试方法	19
第 1 章 数值型数据	24
实例基础 数值型数据概述	24
实例 1 数值输入与显示	26
实例 2 时间标识输入与显示	27
实例 3 滑动杆输出	28
实例 4 旋钮与转盘输出	30
实例 5 滚动条与刻度条	31
实例 6 数值数据基本运算	33
实例 7 数值常量的使用	34
第 2 章 布尔型数据	36
实例基础 布尔型数据概述	36
实例 8 开关控制指示灯	38
实例 9 数值比较与显示	39
实例 10 数值逻辑运算	40
实例 11 真常量与假常量	42
实例 12 确定按钮的使用	43
实例 13 停止按钮的使用	44
实例 14 单选按钮的使用	45
实例 15 按钮的快捷键设置	47



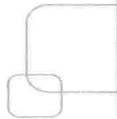
第 3 章 字符串数据	50
实例基础 字符串数据概述	50
实例 16 计算字符串的长度	53
实例 17 连接字符串	54
实例 18 截取字符串	55
实例 19 字符串大小写转换	57
实例 20 替换指定位置和长度的子字符串	58
实例 21 删除指定位置和长度的子字符串	59
实例 22 从指定位置插入子字符串	61
实例 23 搜索并替换子字符串	62
实例 24 搜索并删除子字符串	64
实例 25 格式化日期/时间字符串	65
实例 26 格式化写入字符串	66
实例 27 搜索并拆分字符串	67
实例 28 从指定位置拆分字符串	69
实例 29 选行并添加至字符串	71
实例 30 匹配字符串	72
实例 31 匹配真/假字符串	73
实例 32 组合框	75
第 4 章 数组数据与矩阵	77
实例基础 数组数据概述	77
实例 33 初始化数组	80
实例 34 创建一维数组	81
实例 35 创建二维数组	83
实例 36 计算数组大小	84
实例 37 求数组最大值与最小值	85
实例 38 删除数组元素	87
实例 39 数组索引	89
实例 40 替换数组子集	90
实例 41 提取子数组	92
实例 42 数组插入	93
实例 43 拆分一维数组	95
实例 44 一维数组排序	96
实例 45 搜索一维数组	98
实例 46 数组数据基本运算 1	99
实例 47 数组数据基本运算 2	101



实例 48	矩阵的基本运算	102
实例 49	求解线性代数方程	104
第 5 章 簇数据		106
实例基础	簇数据概述	106
实例 50	将基本数据捆绑成簇数据	108
实例 51	将簇数据解除捆绑	110
实例 52	按名称捆绑	111
实例 53	按名称解除捆绑	113
实例 54	将多个簇数据创建成簇数组	114
实例 55	索引与捆绑簇数组	116
第 6 章 数据类型转换		118
实例基础	数据类型转换概述	118
实例 56	字符串至路径转换	119
实例 57	路径至字符串转换	120
实例 58	数值至字符串转换	121
实例 59	字符串至数值转换	123
实例 60	字节数组至字符串转换	124
实例 61	字符串至字节数组转换	125
实例 62	数组至簇转换	127
实例 63	簇至数组转换	128
实例 64	布尔数组至数值转换	129
实例 65	数值至布尔数组转换	130
实例 66	布尔值至 (0,1) 转换	131
第 7 章 程序流程控制		133
实例基础	程序流程控制概述	133
实例 67	条件结构的使用 1	142
实例 68	条件结构的使用 2	144
实例 69	平铺式顺序结构的使用	145
实例 70	层叠式顺序结构的使用 1	146
实例 71	层叠式顺序结构的使用 2	147
实例 72	使用 For 循环结构产生随机数	149
实例 73	使用 For 循环结构计算 “n!”	150
实例 74	使用 For 循环结构计算累加值	151
实例 75	使用 While 循环结构产生随机数	152
实例 76	使用 While 循环结构计算 “n!”	154



实例 77	使用 While 循环结构计算累加值	155
实例 78	使用定时循环结构产生随机数	156
实例 79	使用定时循环结构计算“ $n!$ ”	158
实例 80	使用定时循环结构计算累加值	160
实例 81	定时顺序结构的使用	161
实例 82	事件结构的使用	163
实例 83	禁用结构的使用	165
第 8 章 变量与节点		167
实例基础 变量与节点概述		167
实例 84	局部变量的创建与使用	171
实例 85	全局变量的创建与使用	173
实例 86	使用公式节点进行数学运算	176
实例 87	使用反馈节点进行数值累加	178
实例 88	使用表达式节点进行数学运算	180
实例 89	使用属性节点控制控件的可见性	181
实例 90	使用属性节点控制控件的可用性	182
第 9 章 图形显示		184
实例基础 图形显示概述		184
实例 91	使用波形图表控件显示正弦波形	186
实例 92	使用波形图控件显示正弦波形	187
实例 93	波形图表与波形图控件的比较	188
实例 94	使用 XY 图控件显示一条曲线	190
实例 95	使用 XY 图控件显示两条曲线	192
实例 96	强度图与强度图表控件的比较	193
实例 97	使用三维曲面控件显示正弦波	195
实例 98	滤除信号噪声	197
第 10 章 文件 I/O		200
实例基础 文件 I/O 概述		200
实例 99	写入文本文件	201
实例 100	读取文本文件	204
实例 101	写入二进制文件	205
实例 102	读取二进制文件	206
实例 103	写入波形至文件	208
实例 104	从文件读取波形	210
实例 105	写入电子表格文件	211



实例 106 读取电子表格文件	213
第 11 章 界面交互及子程序设计	215
实例 107 创建登录对话框	215
实例 108 菜单的设计与使用	217
实例 109 子程序的创建	221
实例 110 子程序的调用	223
参考文献	225

第0章 LabVIEW 程序设计基础

本章作为 LabVIEW 的入门，介绍了 LabVIEW 的特点及应用，LabVIEW 中的基本概念，LabVIEW 的前面板设计，LabVIEW 的数据操作等基本知识。最后通过实例了解虚拟仪器编程语言 LabVIEW 的开发环境及其程序设计步骤，使读者对 LabVIEW 有一个初步认识。

0.1 LabVIEW 的特点及应用

0.1.1 LabVIEW 的特点

LabVIEW 包括控制与仿真、高级数字信号处理、统计过程控制、模糊控制和 PID 控制等众多附加软件包，运行于 Windows NT/XP、Linux、Macintosh 等多种平台的工业标准软件开发环境。

LabVIEW 在业界也称为虚拟仪器，它的表现形式和功能类似于实际的仪器，但 LabVIEW 程序很容易改变设置和功能。因此，LabVIEW 特别适用于实验室、多品种小批量的生产线等需要经常改变仪器参数和功能以及对信号进行分析、研究、传输等场合。

与传统的编程语言相较，LabVIEW 的图形编程方式能够节省程序开发时间，其运行速度却几乎不受影响，体现出了极高的效率。

由于采用了图形化编程语言，LabVIEW 产生的程序是框图的形式，易学易用，特别适合硬件工程师、实验室技术人员、生产线工艺技术人员的学习和使用，可以在很短的时间内掌握并应用到实际中去。

总之，由于 LabVIEW 能够为用户提供简明、直观、易用的图形编程方式，省时简便，深受用户青睐。

0.1.2 LabVIEW 的应用

LabVIEW 在航空、航天、通信、汽车、半导体和生物医学等世界范围的众多领域内得到了广泛应用，从简单的仪器控制、数据采集到尖端的测试和工业自动化，从大学实验室到工厂，从探索研究到技术集成，都有 LabVIEW 的应用。

1. LabVIEW 应用于测量与试验

LabVIEW 已成为测试与测量领域的工业标准，通过 GPIB、VXI、串行设备和插卡式数据采集板可以构成实际的数据采集系统。它提供了工业界最大的仪器驱动程序库以及众多的开发工具，使复杂的测量与试验任务变得简单易行。

2. LabVIEW 应用于过程控制与工业自动化

LabVIEW 强大的硬件驱动、图形显示能力和便捷的快速程序设计为过程控制和工业自动化应用提供了优秀的解决方案。

3. LabVIEW 应用于实验室研究与计算分析

LabVIEW 为科学家和工程师提供了功能强大的高级数学分析库,包括统计、估计、回归分析、线性代数、信号生成算法、时域和频域算法等众多科学领域,可满足各种计算和分析需要。

因此,许多工科大学已将 LabVIEW 作为课堂或实验室教学内容,作为工程师素质培养的一个方面。不同领域的科学家和工程师都借助这个易用的软件包来解决工作中的各种应用问题。

0.2 LabVIEW 中的基本概念

LabVIEW 是一个功能完整的程序设计语言,具有区别于其他程序设计语言的一些独特结构和语法规则。

应用 LabVIEW 编程的关键是掌握 LabVIEW 的基本概念和图形化编程的基本思想。

0.2.1 VI 与子 VI

用 LabVIEW 开发的应用程序称为 VI (Virtual Instrument 的英文缩写,即虚拟仪器)。

一个最基本的 VI 是由节点、端口以及连线组成的应用程序。

VI 运行采用数据流驱动,具有顺序、循环、条件等多种程序结构控制。

在 LabVIEW 中的子程序被称作子 VI (SubVI)。在程序中使用子 VI 有以下优点:

(1) 将一些代码封装成为一个子 VI (即一个图标或节点),可以使程序的结构变得更加清晰、明了。

(2) 将整个程序划分为若干模块,每个模块用一个或者几个子 VI 实现,易于程序的编写和维护。

(3) 将一些常用的功能编制成为一个子 VI,在需要的时候可以直接调用,不用重新编写这部分程序,因而子 VI 有利于代码复用。

正因为子 VI 的使用对编写 LabVIEW 程序有很多便利之处,所以在使用 LabVIEW 编写程序的时候经常会使用子 VI。

子 VI 由 3 部分组成,除前面板对象、框图程序外,还有图标的连接端口。连接端口的功能是与调用它的 VI 交换数据。

基于 LabVIEW 图形化编程语言的特点,在 LabVIEW 环境中,子 VI 也是以图标(节点)的形式出现的。在使用子 VI 时,需要定义其数据输入和输出的端口,然后就可以将其当作一个普通的 VI 来使用。

因此在使用 LabVIEW 编程时,应与其他编程语言一样,尽量采用模块化编程的思想,有效地利用子 VI,简化 VI 框图程序的结构,使其更加简洁,易于理解,可以提高 VI 的运行效率。

0.2.2 前面板

前面板就是图形化用户界面,用于设置输入数值和观察输出量,是人机交互的窗口。由于 VI 前面板是模拟真实仪器的前面板,所以输入量称为控制,输出量称为指示。

在前面板中,用户可以使用各种图标,如仪表、按钮、开关、波形图、实时趋势图等,这可使前面板的界面像真实的仪器面板一样。

图 0-1 是一个调压器程序的前面板。

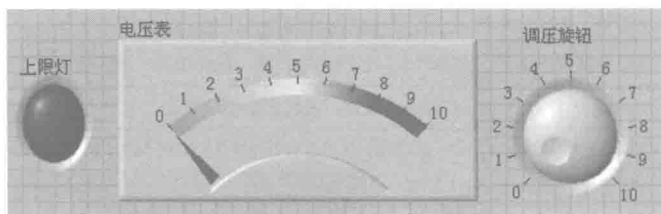


图 0-1 调压器程序的前面板

前面板对象按照功能可以分为控制、指示和修饰三种。控制是用户设置和修改 VI 程序中输入量的接口,如旋钮;指示则用于显示 VI 程序产生或输出的数据,如仪表。

如果将一个 VI 程序比作一台仪器的话,那么控制就是仪器的数据输入端口和控制开关,而指示则是仪器的显示窗口,用于显示测量结果。

在本书中,为方便起见,将前面板中的控制和指示统称为前面板对象或控件。

修饰的作用仅是将前面板点缀得更加美观,并不能作为 VI 的输入或输出来使用。在控制选板中专门有一个修饰子选板。

0.2.3 框图程序

每一个前面板都有一个框图程序与之对应。上述调压器的框图程序如图 0-2 所示,该框图程序的功能是通过调压旋钮产生数值,再通过电压表显示,当数值大于等于 8 时,上限灯改变颜色。

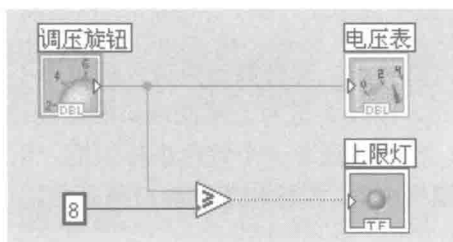


图 0-2 调压器的框图程序

框图程序用图形化编程语言编写，可以把它理解成传统编程语言程序中的源代码。用图形来进行编程，而不是用传统的代码来进行编程，这是 LabVIEW 最大的特色。

框图程序由节点、端口和连线组成。

1. 节点

节点是 VI 程序中的执行元素，类似于文本编程语言程序中的语句、函数或者子程序。上述调压器的框图程序中的数值常量、比较函数就是节点。

LabVIEW 共有 4 种类型的节点，如表 0-1 所示。

表 0-1 LabVIEW 节点类型

节点类型	节点功能
功能函数	LabVIEW 内置节点，提供基本的数据与对象操作，例如，数值计算、文件 I/O 操作、字符串运算、布尔运算、比较运算等
结构	用于控制程序执行方式的节点，包括顺序结构、条件结构、循环结构及公式节点等
代码接口节点	LabVIEW 与 C 语言文本程序的接口。通过代码接口节点，用户可以直接调用 C 语言编写的源程序
子 VI	将创建的 VI 以子 VI 的形式调用，相当于传统编程语言中子程序的调用。通过功能选板中的 Select VI 子选板可以添加一个子 VI 节点

2. 端口

节点之间、节点与前面板对象之间通过数据端口和数据连线来传递数据。

端口是数据在框图程序部分和前面板对象之间传输的通道接口，以及数据在框图程序的节点之间传输的接口。端口类似于文本程序中的参数和常数。

端口有两种类型：控制器/指示器端口和节点端口（即函数图标连线端口）。控制或指示端口用于前面板，当程序运行时，从控制器输入的数据就通过控制器端口传送到框图程序中。而当 VI 程序运行结束后，输出的数据就通过指示器端口从框图程序送回到前面板的指示器中。

当在前面板创建或删除控制器/指示器时，可以自动创建或删除相应的控制器/指示器端口。

一般情况下，LabVIEW 中的每个节点至少有一个端口，用于向其他图标传递数据。

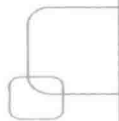
3. 连线

节点之间由数据连线按照一定的逻辑关系相互连接，以定义框图程序内的数据流动方向。连线是端口间的数据通道，类似于文本程序中的赋值语句。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，每种数据类型还可以用不同的颜色予以强调或区分。

连线点是连线的线头部分。接线头是为了帮助端口的连线位置正确。当把连线工具放到端口上时，接线头就会弹出。接线头还有一个黄色小标识框，用来显示该端口的名字。

连接端口通常是隐藏在图标中的。图标和连接端口都是由用户在编制 VI 时根据实际需要创建的。



0.2.4 数据流驱动

由于框图程序中的数据是沿数据连线按照程序中的逻辑关系流动的,因此,LabVIEW 编程又称为“数据流编程”。“数据流”控制 LabVIEW 程序的运行方式。

对一个节点而言,只有当它的输入端口上的所有数据都被提供以后,它才能够执行下去。当节点程序运行完毕以后,它会把结果数据送到其输出端口中,这些数据很快通过数据连线送至与之相连的目的端口。

“数据流”与常规编程语言中的“控制流”类似,相当于控制程序语句一步一步地执行。

例如,两数相加程序的前面板如图 0-3 所示,与之对应的框图程序如图 0-4 所示,这个 VI 程序控制 a 和 b 中的数值相加,然后再把相加之和乘以 100,结果送至指示 c 中显示。

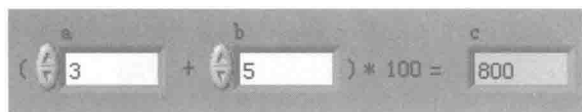


图 0-3 两数相加程序的前面板

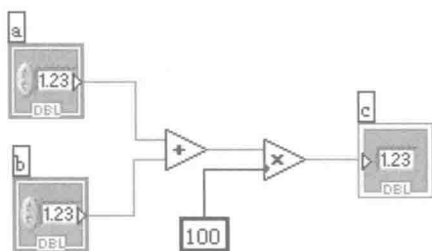


图 0-4 两数相加的框图程序

在这个程序中,框图程序从左向右执行,但这个执行次序不是由其对象的摆放位置来确定的,而是遵循相乘节点的一个输入量是相加节点的运算结果这一顺序。只有当相加运算完成并把结果送到相乘运算节点的输入端口后,相乘节点才能执行下去。

0.3 LabVIEW 的前面板设计

把 VI 应用程序界面称作前面板。前面板是 LabVIEW 的重要组成部分,是用 LabVIEW 编写的应用程序的界面。LabVIEW 提供非常丰富的界面控件对象,可以方便地设计出生动、直观、操作方便的用户界面。

LabVIEW 提供的专门用于前面板设计的输入和显示控件被分门别类地放置在控件选板中,当用户需要使用时,可以根据对象的类别从各个子选板中选取。前面板的对象按照其类型可以分为数值型、布尔型、字符串型、数组型、簇型、图形型等多种类型。

在用 LabVIEW 进行程序设计的过程中,对前面板的设计主要是编辑前面板控件和设置前面板控件的属性。

0.3.1 前面板对象的创建

设计应用程序界面所用到的前面板对象全部包含在控件选板中。

放置在前面板上的每一个控件都具有很多属性，其中多数与显示特征有关，在编程时就可以通过在控件上右击（即右键单击，以下同）更改其属性值。

设计前面板需要用到控件选板，用鼠标选择控件选板上的对象，然后在前面板上拖放即可。

以下举例说明前面板对象的创建过程。首先创建新的应用程序并保存为“创建对象.VI”。

切换到前面板窗口，在控件选板上单击“数值”控件子选板，选择“数值输入控件”，如图 0-5 所示，在前面板的适当位置单击，即可创建数值输入控件。修改数值控件的标签并输入“数字 1”。同样的方法可以创建数值型控件“垂直指针滑动杆”和“旋钮”，如图 0-6 所示。相应的在程序框图窗口中会产生代表控件的图标符号，如图 0-7 所示。



图 0-5 控件选板数值子选板

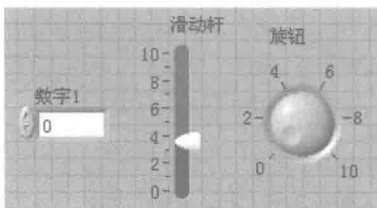


图 0-6 前面板窗口中对象的生成

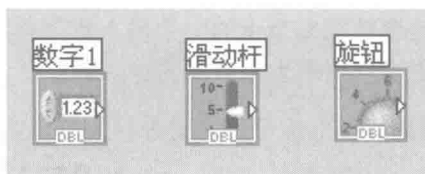


图 0-7 程序框图窗口中自动生成的图标

0.3.2 前面板对象的属性配置

此处介绍的前面板对象的配置方法适用于输入控件和显示控件。

右击前面板对象，如滑动杆控件，弹出快捷菜单，如图 0-8 所示。这里只介绍输入控件和显示控件共有的快捷菜单部分。

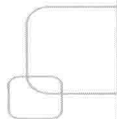
(1) 显示项：该菜单列表显示一个对象可以显示/隐藏的部分，如标签、标题等。

(2) 查找接线端：在代码窗口中高亮显示前面板对象。当代码窗口中对象太多时，对直接寻找控件对象是非常有效的。

(3) 转换为显示控件/转换为输入控件：将指定的对象改变为显示控件或输入控件。

任何一个前面板对象都有控制和指示两种属性，右击前面板对象，在弹出的快捷菜单中选择“转换为显示控件”或“转换为输入控件”，可以在控制和指示两种属性之间切换。

一般控件可以指定为显示量，也可以转化为输入量。比如右击滑动杆控件，在弹出的快



捷菜单中单击“转换为显示控件”，该控件已经变成了显示件。该变化也同时反映到程序框图窗口中的图标上。

(4) 创建：针对选择的对象创建局部变量、引用和属性节点等。

(5) 替换：选择其他的控件来代替当前的控件。

(6) 数据操作：包含一个编辑数据选项的子菜单。主要包括以下选项：重新初始化默认值和当前值设置为默认值。图 0-6 中，各个控件在设计时就已经有了默认的初始值，如果要改变这个初始值，则在设计时给控件输入指定的数值。在控件上右击，在弹出的快捷菜单中选择：数据操作/当前值设置为默认值，如图 0-9 所示。这样每次在程序打开时，控件就自动赋予了新的默认值。

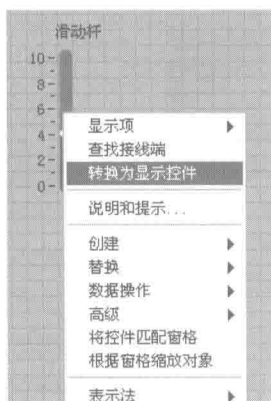


图 0-8 改变控件的属性

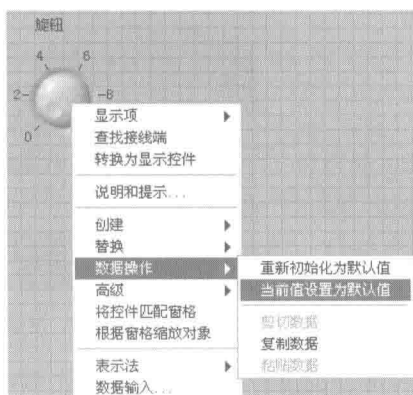


图 0-9 设置控件的默认值

(7) 高级：包含控件高级编辑选项的子菜单。主要包括以下选项：

快捷键：为控件分配快捷键，用户在没有鼠标的情况下仍然可以访问控件。

同步显示：控件将显示全部的更新数据，这种设置方法将影响 LabVIEW 的运行性能。

自定义：由用户定制控件，在控件编辑器中设计个性化的前面板对象。

隐藏输入控件/隐藏显示控件：在前面板中隐藏控件对象。要访问隐藏的对象，在代码窗口中右击控件对象，在弹出菜单中选择“显示输入控件”或“显示显示控件”。

0.3.3 前面板对象的修饰

作为一种基于图形模式的编程语言，LabVIEW 在图形界面的设计上有着得天独厚的优势，可以设计出漂亮、大方而且方便、易用的程序界面（即程序的前面板）。为了更好地进行前面板的设计，LabVIEW 提供了丰富的修饰前面板的方法以及专门用于装饰前面板的控件，下面介绍修饰前面板的方法和技巧。

1. 设置前面板对象的颜色

前景色和背景色是前面板对象的两个重要属性，合理地搭配对象的前景色和背景色会使用户的程序增色不少。一般情况下控件选板上的对象是以默认颜色被拖放到前面板中的，可以通过简单的操作进行修改。

对于前面板对象的颜色的编辑需要用到工具选板里的取色工具和颜色设置工具。此处创建新的 VI “设置颜色.vi”。在程序的前面板创建 1 个数值量控件“液罐”，颜色等均采用默认值。

颜色设置工具为，图标内有前后两个调色板，分别代表前景色和背景色。分别用鼠标单击两个调色板会出现颜色选择对话框，图 0-10 所示，以设置前景和背景的颜色。用鼠标单击颜色设置工具后，再在编辑对象的适当位置上单击鼠标，则被编辑对象就被分别设置成指定的前景色和背景色。

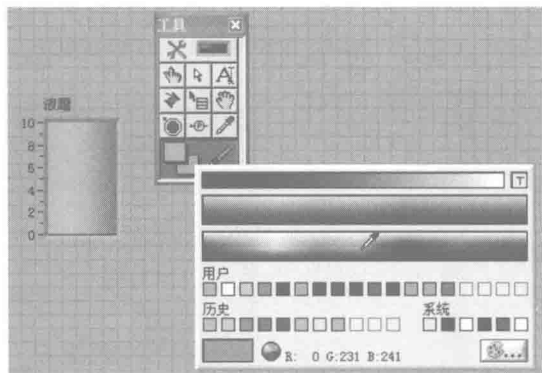


图 0-10 设置前景和背景颜色

另外一种简便的操作是，用鼠标单击颜色设置工具后，在被编辑对象的适当位置左右击，弹出颜色对话框并且动态地渲染被编辑的对象，选择合适的颜色后单击鼠标，完成颜色的设置。

2. 设置前面板对象的文字风格

在 LabVIEW 中，可以设置前面板文本对象的字体、颜色以及其他风格特征。这些可以通过 LabVIEW 工具栏中的字体按钮  12pt 应用程序字体 进行设置。单击该按钮，将弹出用于设置字体的下拉菜单，在菜单中，用户可以选择文字的字体、颜色、大小和风格。用户也可以在字体按钮的下拉菜单中选择字体对话框来设置字体的常用属性。字体设置对话框如图 0-11 所示，在这个对话框中几乎可以包括设置字体的所有属性。

3. 前面板对象的位置与排列

为了提高前面板外观设计的效率，LabVIEW 提供了前面板对象编辑控制的一些工具，尤其是在界面对象比较多时，这些工具就显得尤为重要。

在 LabVIEW 程序中，设置多个对象的相对位置关系是布置和修饰前面板过程中一件非常重要的工作。在 LabVIEW 中提供了专门用于调整多个对象位置关系的工具，它们位于 LabVIEW 的工具栏上。

LabVIEW 所提供的用于设置多个对象之间位置关系的工具，如图 0-12 所示，这两种工具分别用于调整多个对象的对齐关系以及调整对象之间的距离。

群组工具可以将一系列对象设置为组，以固定其相对位置关系，也可以锁定对象，以免在编辑过程中对象被移动。