

LabVIEW

程序设计与应用(第2版)

<http://www.phei.com.cn>



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

杨乐平 李海涛 杨磊 编著



附光盘

LabVIEW 程序设计与应用

(第2版)

杨乐平 李海涛 杨 磊 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以最新 LabVIEW 7 Express 版本为对象,系统介绍了 LabVIEW 程序设计的基本概念、关键技术和实际应用的专门知识。本书内容分为三大部分,第一部分介绍虚拟仪器的基本概念、图形化编程语言基本原理与特点、LabVIEW 编程环境;第二部分系统介绍 LabVIEW 程序设计的语法规则、程序结构和基本编程技巧;第三部分介绍 LabVIEW 在数据采集、仪器控制和通信等方面的应用。本书结构编排合理,运用大量实例阐述基本概念与编程难点,突出内容的系统性与实用性。为方便读者学习查阅,本书附带光盘按章节编排,提供了本书所有编程例子,并且列出了 LabVIEW 程序错误代码表,供读者参考。

本书可作为大、中专院校相关专业教材或教学参考书,也可供有关工程技术人员和软件工程师参考。

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 程序设计与应用 / 杨乐平, 李海涛, 杨磊编著. —2 版. —北京: 电子工业出版社, 2005.1

ISBN 7-121-00588-3

I. L... II. ①杨...②李...③杨... III. 软件工具, LabVIEW—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2004) 第 120494 号

责任编辑: 刘志红 特约编辑: 张莉

印 刷: 北京民族印刷厂

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

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 29.5 字数: 756 千字

印 次: 2005 年 1 月第 1 次印刷

印 数: 4000 册 定价: 42.00 元 (含光盘 1 张)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010) 68279077。质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

自 20 世纪 90 年代以来,随着计算机技术的迅猛发展,虚拟仪器技术在数据采集、自动测试和测量仪器领域得到广泛应用,促进和推动测试系统和测量仪器的设计方法与实现技术发生了深刻的变化。“软件就是仪器”已经成为测试与测量技术发展的重要标志。美国国家仪器公司(National Instruments,简称 NI)是虚拟仪器技术的主要倡导者和贡献者,其创新软件产品 LabVIEW(Laboratory Virtual Instrument Engineering Workbench,简称 LabVIEW)自 1986 年问世以来,已经成为虚拟仪器软件开发平台事实上的工业标准,在研究、制造和开发的众多领域得到广泛应用。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,人们都可以发现 LabVIEW 应用的成果和开发的产品。LabVIEW 所创新的图形化语言编程方法成为虚拟仪器技术应用与发展的基础,得到工业界和学术界的广泛关注。

最近几年,随着 LabVIEW 在国内普及和应用的推广,陆续出版了一批有关 LabVIEW 程序设计和应用开发方面的教材和专著,培养了一批 LabVIEW 的忠实用户和程序设计员。由本书编著者编写,电子工业出版社 2001 年 7 月出版的《LabVIEW 程序设计与应用》是国内第一本系统介绍 LabVIEW 程序设计的入门书。该书体系结构设计突出了图形化编程语言的概念、方法与特点,论述深入浅出,编排结构和编写风格采用寓教于例、图文结合的形式,得到了广大 LabVIEW 用户的认同和欢迎。该书已多次重印,国内多所高等院校已将该书选为教材。但是由于该书是以当时的 LabVIEW5.1 版本为蓝本组织内容,而目前 NI 公司已经推出的最新 LabVIEW 7 Express 版本,在开发环境、编程手段、应用管理等方面增加了许多先进功能,因此有必要对该书内容进行更新、完善和扩展,以适应技术发展的要求。

本书在保持原书风格、特色与体系基本不变的基础上,以 LabVIEW 7 Express 版本为对象,结合读者反馈意见和我们最新的研究成果,对原书内容进行了较大的增加、修订和调整。增加的内容主要包括 LabVIEW 7 Express 版本在编程环境、数据类型、模板设置等方面新增的功能,如 Express VI、动态数据、波形数据、事件结构、反馈节点、逐点数字信号处理等;也包括在数据采集、仪器控制和通信等方面的应用,增加或扩展了 VPP 和 IVI 仪器驱动器、DataSocket 编程、远程操作面板、PLC 通信等内容。在修订和调整时,主要对原书的一些论述和结构顺序进行了完善和优化,如将字符串运算从原第 8 章“字符串与文件 I/O”调整到第 4 章“数据操作”,原第 8 章则加大了文件操作与管理的论述与编程实例等。总之,本书力求在保持原书特色的基础上,内容更加系统完整,结构更加科学合理,应用更加全面深入,为广大 LabVIEW 用户提供一本易懂实用的入门教材。

参加本书编写工作的有国防科技大学杨乐平教授、李海涛博士和杨磊博士,全书由杨乐平教授统稿。本书可供相关专业高年级本科生和研究生作为教材使用,也可供从事测试计量、仪器设计、过程控制及数据处理方面工作的工程技术人员参考。

由于编著者水平有限,书中难免有疏漏和错误之处,恳请广大读者批评指正。

编著者

2004 年 10 月于国防科技大学

目 录

第 1 章 绪论	(1)
1.1 LabVIEW 概述	(1)
1.1.1 LabVIEW 起源	(1)
1.1.2 LabVIEW 概念创新	(2)
1.2 G 语言与虚拟仪器	(3)
1.3 LabVIEW 应用解决方案	(4)
1.4 LabVIEW 7 Express 新特性	(6)
第 2 章 LabVIEW 编程环境	(11)
2.1 LabVIEW 系统安装	(11)
2.2 LabVIEW 启动	(11)
2.3 LabVIEW 模板	(15)
2.4 VI 库	(21)
2.5 定制 LabVIEW 环境	(22)
2.5.1 环境参数设置	(22)
2.5.2 模板设置	(23)
第 3 章 LabVIEW 编程入门	(26)
3.1 基本概念	(26)
3.1.1 前面板	(27)
3.1.2 框图程序	(29)
3.1.3 使用数据连线	(33)
3.1.4 图标/连接端口	(41)
3.2 LabVIEW 术语	(42)
3.3 创建和编辑 VI	(43)
3.3.1 创建 VI	(44)
3.3.2 编辑 VI	(49)
3.4 运行和调试 VI	(60)
3.4.1 运行 VI	(60)
3.4.2 调试 VI	(61)
3.5 创建和调用 SubVI	(64)
3.5.1 创建 SubVI	(64)
3.5.2 调用 SubVI	(67)
3.6 Express VI	(69)

3.6.1	Express VI 的特点	(69)
3.6.2	Express VI 的使用方法	(72)
3.7	获取帮助	(74)
3.7.1	实时上下文帮助	(74)
3.7.2	VI 及功能模块帮助	(75)
3.7.3	LabVIEW 例程	(75)
3.7.4	LabVIEW 书架	(76)
3.7.5	LabVIEW 网络资源	(77)
第 4 章	数据操作	(78)
4.1	数据类型	(78)
4.1.1	数字型	(80)
4.1.2	布尔型	(85)
4.1.3	字符串型与路径	(86)
4.2	数学运算	(90)
4.2.1	数字常量	(91)
4.2.2	基本数学运算节点	(91)
4.2.3	类型转换节点	(93)
4.2.4	三角函数节点	(96)
4.2.5	对数节点	(97)
4.2.6	复数节点	(98)
4.2.7	附加常数节点	(99)
4.3	布尔运算	(100)
4.4	字符串运算	(102)
4.4.1	字符串常量	(102)
4.4.2	基本字符串运算	(103)
4.4.3	字符串/数字转换	(107)
4.4.4	字符串/数组/路径转换	(110)
4.4.5	附加字符串运算	(112)
4.5	比较运算	(115)
第 5 章	变量、数组、簇与波形数据	(118)
5.1	本地变量	(118)
5.1.1	本地变量的创建	(118)
5.1.2	本地变量的使用	(120)
5.1.3	本地变量的特点	(121)
5.2	全局变量	(122)
5.2.1	全局变量的创建	(122)
5.2.2	全局变量的使用	(123)
5.2.3	全局变量的特点	(124)

5.3	数组	(125)
5.3.1	数组的组成与创建	(125)
5.3.2	数组的使用	(127)
5.3.3	数组的特点	(149)
5.4	簇	(149)
5.4.1	簇的组成与创建	(149)
5.4.2	簇的使用	(151)
5.4.3	簇的特点	(161)
5.5	波形数据	(163)
5.5.1	波形数据的组成	(163)
5.5.2	波形数据的使用	(167)
5.5.3	波形数据的特点	(178)
第6章	结构与属性	(179)
6.1	For 循环	(179)
6.1.1	For 循环的组成	(180)
6.1.2	For 循环的使用	(182)
6.1.3	For 循环的特点	(186)
6.2	While 循环	(186)
6.2.1	While 循环的组成	(186)
6.2.2	While 循环的使用	(187)
6.2.3	While 循环的特点	(188)
6.3	顺序结构	(189)
6.3.1	顺序结构的组成	(190)
6.3.2	顺序结构的使用	(192)
6.3.3	顺序结构的特点	(194)
6.4	选择结构	(196)
6.4.1	选择结构的组成	(196)
6.4.2	选择结构的使用	(198)
6.4.3	选择结构的特点	(198)
6.5	事件结构	(199)
6.5.1	事件结构的组成	(200)
6.5.2	事件结构的使用	(201)
6.5.3	事件结构的特点	(203)
6.6	基本公式节点	(205)
6.6.1	基本公式节点的创建	(205)
6.6.2	基本公式节点的使用	(206)
6.6.3	基本公式节点的特点	(209)
6.7	属性节点	(209)

6.7.1	属性节点的创建	(209)
6.7.2	属性节点的使用	(211)
6.7.3	属性节点的特点	(214)
第 7 章	波形显示	(215)
7.1	事后记录波形图	(215)
7.1.1	事后记录波形图的组成	(216)
7.1.2	使用事后记录波形图	(217)
7.1.3	定制事后记录波形图的外观	(223)
7.2	实时趋势图	(231)
7.2.1	使用实时趋势图	(231)
7.2.2	定制实时趋势图的外观定制	(233)
7.3	XY 波形图	(235)
7.4	密度图形显示控件 (Intensity Graph)	(237)
7.4.1	使用密度图	(237)
7.4.2	定义密度图的颜色	(238)
7.4.3	设置密度图的外观	(239)
7.5	密度趋势图	(240)
7.6	三维曲面图	(241)
7.6.1	使用三维曲面图	(241)
7.6.2	设置三维曲面图的外观	(244)
7.7	三维参数曲面图	(247)
7.7.1	使用三维参数曲面图	(248)
7.7.2	设置三维参数曲面图的外观	(249)
7.8	三维曲线图	(249)
7.8.1	使用三维曲线图	(249)
7.8.2	设置三维曲线图的外观	(251)
7.9	极坐标图	(251)
7.9.1	使用极坐标图	(251)
7.9.2	设置极坐标图的外观	(252)
第 8 章	文件操作与管理	(253)
8.1	基本概念及术语	(253)
8.1.1	路径	(253)
8.1.2	标识号	(254)
8.1.3	文件 I/O 的出错管理	(256)
8.1.4	文件 I/O 操作流程控制	(256)
8.2	文件操作	(256)
8.2.1	文件定位与文件对话框	(257)
8.2.2	文件操作	(260)

8.3	文件管理	(267)
8.3.1	文件的删除、移动和复制	(268)
8.3.2	获取文件、目录的信息	(268)
8.3.3	路径、目录操作	(269)
8.4	数据存储与读取	(272)
8.4.1	LabVIEW 数据文件类型	(272)
8.4.2	数据文件存储与读取	(273)
第 9 章	数学分析与信号处理	(284)
9.1	数学分析	(285)
9.1.1	公式运算节点	(286)
9.1.2	函数计算与微积分	(289)
9.1.3	概率统计与曲线拟合	(295)
9.1.4	矩阵与数组运算	(300)
9.1.5	最优化与零点求解	(304)
9.1.6	数值函数	(309)
9.2	数字信号处理	(310)
9.3	波形测量	(328)
9.3.1	波形测量节点	(328)
9.3.2	波形测量应用实例	(331)
9.4	信号调理	(332)
9.4.1	信号调理节点	(332)
9.4.2	信号调理应用实例	(333)
9.5	波形监测	(335)
9.5.1	波形监测节点	(335)
9.5.2	波形监测应用实例	(336)
9.6	逐点信号分析	(338)
9.6.1	逐点信号分析的特点	(338)
9.6.2	逐点信号分析节点	(338)
9.6.3	逐点信号分析应用实例	(339)
第 10 章	LabVIEW 程序设计	(340)
10.1	人机交互界面	(340)
10.1.1	定制前面板对象	(340)
10.1.2	选单的编辑与响应	(341)
10.1.3	子面板的使用	(343)
10.1.4	界面装饰	(345)
10.2	定时与对话框	(346)
10.2.1	定时器	(347)
10.2.2	对话框	(347)

10.2.3	错误处理节点	(351)
10.3	LabVIEW 环境参数设置	(352)
10.3.1	新特性	(353)
10.3.2	路径与性能	(355)
10.3.3	编程界面	(356)
10.3.4	模板与调试	(359)
10.3.5	属性设置	(361)
10.3.6	VI Server 与 Web Server	(365)
10.4	VI 属性设置	(368)
10.4.1	一般设置	(369)
10.4.2	存储空间	(370)
10.4.3	帮助与编辑	(371)
10.4.4	版本历史与安全	(372)
10.4.5	窗口与运行	(374)
10.5	文件管理	(376)
10.6	创建应用程序	(378)
10.7	培养良好的编程风格	(381)
第 11 章	数据采集	(384)
11.1	数据采集基础	(384)
11.1.1	DAQ 功能	(384)
11.1.2	DAQ 节点的组织与结构	(385)
11.1.3	DAQ VIS 的组织结构	(386)
11.1.4	DAQ 节点常用参数简介	(388)
11.2	DAQ 设备的安装与配置	(391)
11.2.1	安装 PCI-1200 数据采集卡	(392)
11.2.2	配置 PCI-1200 数据采集卡	(392)
11.2.3	配置数据采集虚拟通道	(396)
11.3	DAQ 编程	(399)
11.3.1	简易 DAQ 编程	(399)
11.3.2	扩展 DAQ 编程	(402)
11.3.3	高级 DAQ 编程	(408)
第 12 章	仪器控制	(412)
12.1	仪器驱动器	(412)
12.1.1	VPP 仪器驱动器	(412)
12.1.2	IVI 仪器驱动器	(415)
12.2	VISA 标准	(416)
12.3	VISA 编程	(418)
12.3.1	VISA 节点	(419)

12.3.2	VISA 编程实例	(423)
12.4	VPP 驱动程序转换与编程	(424)
12.4.1	VT1432A 数字化仪简介	(425)
12.4.2	VPP 驱动程序转换	(425)
12.4.3	VPP 驱动程序编程实例	(428)
第 13 章	通信	(431)
13.1	串行通信	(431)
13.1.1	串口简介	(431)
13.1.2	串行通信节点	(434)
13.1.3	串行通信编程举例	(437)
13.2	网络通信	(439)
13.2.1	TCP 协议简介	(439)
13.2.2	TCP 节点	(440)
13.2.3	TCP 通信编程实例	(441)
13.3	DataSocket 通信	(443)
13.3.1	DataSocket 基本概念	(443)
13.3.2	DataSocket 节点	(446)
13.3.3	DataSocket 编程举例	(449)
13.4	远程面板	(452)
13.4.1	配置 LabVIEW Web Server	(453)
13.4.2	在 LabVIEW 环境中操作 Remote Panel	(454)
13.4.3	通过网页浏览器在网页中操作 Remote Panel	(456)
参考文献		(458)

第1章 绪 论

1.1 LabVIEW 概述

1.1.1 LabVIEW 起源

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开发集成环境。

数据采集、仪器控制、过程监控和自动测试是实验室研究和工业自动化领域广泛存在的实际任务。在 20 世纪 80 年代初个人计算机出现之前, 几乎所有拥有程控仪器的实验室都采用贵重的仪器控制器来控制测试系统, 这些功能单一、价格昂贵的仪器控制器通过一个集成通信口来控制 IEEE-488 总线仪器 (也称为 GPIB 程控仪器)。后来, 随着 PC 的出现, 工程师和科学家们找到了一种通过性能价格比高的通用 PC 控制台式仪器的方法, 各种基于 PC 的接口板卡产品迅速地打开了市场, NI 公司也应运而生。1983 年, NI 公司已经成为世界上 PC GPIB 接口卡最主要的供应商。

到 1983 年, GPIB 总线事实上已经成为连接仪器和计算机的通用标准接口。除了不同仪器制造商对 IEEE-488 标准的个别解释不同之外, 用户在物理上配置仪器和仪器系统基本上已没有问题。不过, 仪器控制软件的发展, 仍然存在许多问题。当时几乎所有的仪器控制程序都是由 BASIC 语言编写的。虽然与可读性差、编程专业性要求更高的机器语言和汇编语言相比, BASIC 语言已经具有许多优势 (如简单、可读性强的命令集和交互能力), 但与其他基于文本的高级语言一样, 它也存在一个根本问题, 即要求使用仪器的科学家、工程师和技术人员成为程序员。这些用户必须将他们关于仪器和应用的知识转化成一行的程序代码, 以形成测试程序。这个过程经常是费时费力的苦差事, 尤其是对当时那些很少编程或基本没有编程经验的测试工程师更是如此。

NI 公司有一支用 BASIC 语言开发仪器程控软件的程序员专门队伍, 因此, 它十分敏锐地感觉到程控仪器编程为工程师和科学家带来的负担, 清楚地意识到需要开发一个用于程控仪器编程的软件工具。NI 公司的创始人杰姆·特鲁查德博士、杰夫·柯德斯凯博士和他们的好友杰克·麦克里森组成了一个小组, 开始研究开发这个新的软件工具, 希望这个新的软件工具能够改变工程师和科学家从事测试开发的方式。他们首先想到的软件工具模型是电子表格软件, 电子表格软件解决了特鲁查德博士、柯德斯凯博士和麦克里森三人希望解决的共同问题: 使计算机更容易被非程序员的计算机用户使用。当然, 电子表格软件主要是为财务人员设计的, 而特鲁查德博士小组设想的软件工具是为从事测试和仪器控制的工程师和科学家服务的。

1984 年, 当时财政实力还相对较弱的 NI 公司决定投资启动该软件工程项目, 特鲁查

德博士负责研究工作, 并成立了一个基金会, 任命柯德斯凯为项目实施人。为了克服办公室日常事务干扰, 为创新和灵感创造一个安静环境, 柯德斯凯在得州首府奥斯汀得州大学附近安营扎寨, 这样一方面可以利用得州大学的图书馆资源, 另一方面也便于雇用学生程序员。测试与仪器领域一个重要事件就此拉开序幕。

1.1.2 LabVIEW 概念创新

LabVIEW 的概念雏形来源于特鲁查德和柯德斯凯两人 20 世纪 70 年代末期在 ARL (Applied Research Laboratory, 应用研究实验室) 完成的一个大型测试系统。该系统主要用于测试美国海军的声呐探测器, 研究人员也可用该系统开展水声学实验研究。这套测试系统的应用十分灵活, 因为为各层次用户提供了不同的交互接口。技术人员可以在某些预先确定的限制条件下, 操作测试系统完成指定的测试任务; 水声工程师可以进入低层设备设计测试过程; 而研究人员权限最大, 他们可以进入系统所有可编程硬件, 配置需要的测试系统。该测试系统也存在两个缺点: 一是系统需要超过 18 人一年; 二是用户必须理解选单上复杂的命令缩语。

通过几年的时间, 柯德斯凯把从该测试系统得到的启示发展到测试系统软件由多层虚拟仪器 (Virtual Instruments, 简称 VI) 构成的新概念。一个 VI 可以由更低层的多个 VI 组成, 就像真实仪器由印制电路板组成, 而印制电路板又由集成电路 (IC) 组成一样。底层 VI 代表了最基本的软件功能——计算与输入/输出 (I/O), 操作。柯德斯凯特别重视多层软件的互连与嵌套, 创造性地提出了各层 VI 都有相同结构形式的思想。在硬件领域, 将 IC 装配为电路板和将电路板装配为仪器的技术是完全不同的; 在软件领域, 由语句组成子程序和子程序组成程序也有差别, 更不用说由多个程序构成一个系统。所以柯德斯凯提出的所有层次 VI 一致性的结构和接口模型, 极大地简化了软件结构, 是对传统结构化软件设计思想的一个新发展。

虚拟仪器模型的另一个主要特征是每一个 VI 都有一个用户接口组件 (以下称 VI 前面板), 也就是与实际仪器面板相对应的软面板。在当时传统的编程语言里, 即使是简单的命令行用户接口, 在核心程序完成后也还要增加一系列复杂的输入输出语句, 何况是设计复杂的仪器操作面板, 但是 VI 前面板概念模型的提出改变了这一状况。VI 前面板接口已成为整个软件模型不可分割的一部分, 用户通过打开 VI 前面板, 就可以在系统的任何层次上与 VI 交互, 并且前面板对象的设计与修改不涉及程序结构和源代码的变化, 使得在开发过程中一步步调试软件模块和定位编程错误更为方便。

柯德斯凯是一个 UNIX 系统程序员, 当 1983 年 Apple 公司支持图形化操作界面的 Macintosh 个人计算机问世后, 柯德斯凯灵机一动, 他心目中可以模拟真实仪器前面板的 VI 图形化操作面板可以实现了。应该说, 图形化操作系统的出现, 为 LabVIEW 的实现奠定了技术基础, 然而, VI 仅仅有容易操作的图形化前面板还不够, 还必须在编程技术上有大的突破, 柯德斯凯再次想到了电子表格软件。电子表格软件采用财务人员最熟悉的数据表格和公式来实现财务软件编程, 广大工程师又是如何设计软件的呢? 最基本的办法是首先按照系统要求设计流程图, 再将流程图转化为具体的程序代码, 当然, 这个转化过程需要许多编程技巧。柯德斯凯设想构造设计一种基于框图的编程方法, 既便于用户概念设计, 其功能和灵活性又足以作为编程语言使用。

在分析比较了几种框图编程方法的优劣后,柯德斯凯决定采用数据流图作为编程工具。数据流图长期以来一直被认为是顶层软件设计的有效工具,但一般的数据流图并没有提供循环结构、顺序结构和条件结构等程序设计的基本要素。柯德斯凯扩展了数据流图功能,使它能够处理循环、顺序和条件等程序控制,并在此基础上提出了结构化数据流程图模型。1990年他的结构化数据流图和虚拟仪器面板获得了两项美国专利。

在VI模型、图形界面和结构化数据流图编程等核心技术确定后,编程实现相对容易多了。柯德斯凯用4个月时间组织了一支软件开发队伍,在Macintosh上开始编程工作。在开发过程中,麦克里森表现了卓越的项目组织管理能力,提出了整个软件设计的关键数据结构与模块关系,加快了整个软件开发进程。柯德斯凯领导的开发小组克服了程序溢出和内存不足等困难,于1986年5月推出LabVIEW Beta测试版,又经过几个月的反馈修改,于1986年10月正式发布了LabVIEW 1.0版。LabVIEW最初吸引的仅仅是没有任何编程语言经验的用户,这些用户相信采用LabVIEW就能实现有经验的程序员也难完成的应用程序。

有效的内存管理是使图形化编程语言优于普通解释语言的关键。由于数据流解释需要大量分配内存,因此,内存重用对数据流图编程效率至关重要,寻找内存重用的有效算法成为提高LabVIEW性能的关键。LabVIEW 1.1版解决了算法问题,随后改进的LabVIEW 1.2版是可靠性和鲁棒性很强的产品,但由于内在体系局限,其性能与C语言程序相比,仍然有较大差距。为了解决这个问题,1988年开始的LabVIEW 2.0采用了最新的面向对象编程(OOP)技术。当1990年1月LabVIEW 2.0发送给第一个热心用户时,LabVIEW程序在执行速度和灵活性等方面的改进令人惊叹。与普通编程语言采用编译、连接等步骤生成应用程序不同,LabVIEW 2.0编译器是整个软件集成和不可见的部分,编译速度极快。

LabVIEW 2.0以前的版本都是运行在Macintosh平台上的,在Windows 3.0操作系统出现,32位Windows应用程序设计成为可能后,LabVIEW才实现了从Macintosh平台到Windows平台的移植。1992年8月,跨平台的LabVIEW 2.5问世。1993年1月,增加了大量新特性的LabVIEW 3.0正式发行,这些新特性包括全局与局部变量、属性节点和执行动画。从LabVIEW 3.0版本开始,LabVIEW作为一个完整优异的图形化软件开发环境得到了工业界和学术界的认可,并开始迅速占领市场,赢得了广大用户的青睐。

1.2 G语言与虚拟仪器

从LabVIEW研制开发的过程可以看到,虽然LabVIEW本身是一个功能比较完整的软件开发环境,但它是为替代常规的BASIC或C语言而设计的,LabVIEW是编程语言而不仅仅是一个软件开发环境。作为编写应用程序的语言,除了编程方式不同外,LabVIEW具备语言的所有特性,因此又称之为G语言。

G语言是一种适合应用于任何编程任务,具有扩展函数库的通用编程语言。和BASIC或C语言一样,G语言定义了数据模型、结构类型和模块调用语法规则等编程语言的基本要素,在功能完整性和应用灵活性上不逊于任何高级语言,同时,G语言丰富的扩展函数库还为用户编程提供了极大的方便。这些扩展函数库主要面向数据采集、GPIB和串行仪器控制,以及数据分析、数据显示和数据存储。G语言还包括常用的程序调试工具,比如允许设置断点、单步调试、数据探针和动态显示执行程序流程等功能。G语言与传统高级编程语言最大的差别在于编程方式,一般高级语言采用文本编程,而G语言采用图形化编程方式。

G 语言编写的程序称为虚拟仪器 VI (Virtual Instruments), 因为它的界面和功能与真实仪器十分相像, 在 LabVIEW 环境下开发的应用程序都被冠以 .VI 后缀, 以表示虚拟仪器的含义。一个 VI 由前面板、数据流框图程序和图标连接端口组成, 各部分功能如下:

1. 前面板

前面板是 VI 的交互式用户接口, 与真实物理仪器面板相似。前面板可以包含旋钮、刻度盘、开关、图表和其他界面工具, 允许用户通过键盘或鼠标获取数据显示结果。

2. 数据流框图程序

VI 从数据流框图程序中接收指令, 框图程序是一种解决编程问题的图形化方法, 实际上是 VI 的程序代码。

3. 图标连接端口

VI 图标和连接端口的功能就像一个图形化参数列表, 可在 VI 与 SubVI 之间传递数据。一个 VI 既可以作为上层独立程序, 也可以作为其他程序 (或子程序) 的子程序。当一个 VI 作为子程序时, 称做 SubVI。

正是基于 VI 的上述特性, G 语言最佳地实现了模块化编程思想。用户可以将一个应用分解为一系列任务, 再将每个任务细分, 将一个复杂的应用分解为一系列简单的子任务, 为每个子任务建立一个 VI, 然后, 把这些 VI 组合在一起完成最终的应用程序。因为每个 SubVI 可以单独执行, 所以很容易调试。进一步而言, 许多低层 SubVI 可以完成一些常用功能, 因此, 用户可以开发特定的 SubVI 库, 以适用一般的应用程序。

G 语言是 LabVIEW 的核心, 熟练掌握 G 语言的编程要素和语法规则, 是开发高水平 LabVIEW 应用程序最重要的基础。换句话说, 要真正掌握 LabVIEW 开发工具, 必须把它作为一个编程语言, 而不仅仅是一个编程环境来学习, 这正是本书着力强调并贯穿于全过程的重点内容。

虚拟仪器概念是 LabVIEW 的精髓, 也是 G 语言区别于其他高级语言最显著的特征。正是由于 LabVIEW 的成功, 才使虚拟仪器的概念为学术界和工程界广泛接受; 反过来也正是因为虚拟仪器概念的延伸与扩展, 才使 LabVIEW 的应用更加广泛。

1.3 LabVIEW 应用解决方案

LabVIEW 自 1986 年正式推出至今不到 19 年的时间内, 已经从最初单一图形化编程功能、单一运行平台发展到目前以最新版本 LabVIEW 7 Express 为核心, 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/2000、Linux、Macintosh、Sun 和 HP-UX 等多种平台的工业标准软件开发环境。在美国, 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课题。

LabVIEW 在包括航空、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,都可以发现应用 LabVIEW 的成果和开发产品。

1. LabVIEW 应用于测试与测量

LabVIEW 已成为测试与测量领域的工业标准,通过 GPIB、VXI、PLC、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,同时还支持通过 Internet、ActiveX、DDE 和 SQL 等交互式通信方式实现数据共享,它提供的众多开发工具使复杂的测试与测量任务变得简单易行。

2. LabVIEW 应用于过程控制和工业自动化

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。对于更复杂更专业的工业自动化领域,在 LabVIEW 基础上发展起来的 BridgeVIEW 是更好的选择。

3. LabVIEW 应用于实验室研究与自动化

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。即使在联合时域分析、小波和数字滤波器设计等高级或特殊分析场合,LabVIEW 也为此提供了专门的附加软件包。

LabVIEW 是一个具有高度灵活性的开发系统,用户可以根据自己的应用领域和开发要求选择 LabVIEW 系统配置。NI 公司为不同层次用户提供了 3 种系统配置:

(1) LabVIEW 基本版

LabVIEW 基本版是用于开发数据采集和仪器控制系统的最小 LabVIEW 配置,包括 VISA、GPIB、RS-232、DAQ 和基本分析库,同时还包括支持 ActiveX、TCP/IP 和 DDE 等标准程序的接口。

(2) LabVIEW 完整版 (FDS)

除了基本版的功能外, FDS 还包括完整的高级分析库。

(3) LabVIEW 专业版 (PDS)

LabVIEW 专业版除了 FDS 的功能外,还具有专业程序员开发时所需要的全部工具,包括可执行文件生成工具、源代码控制、复杂矩阵分析、软件工程文档管理、质量控制标准文档、图形差异比较和大型软件项目管理文档工具等。

对一般用户而言,采购 LabVIEW 完整版,并根据实际应用选取专门的 LabVIEW 工具套件是最佳选择。表 1.3.1 列出了需单独购买的 LabVIEW 主要工具包。

除了表 1.3.1 列出的 LabVIEW 工具包以外,还有许多第三方软件开发商设计开发的大量定制 VI 供用户选择。实际上 LabVIEW 已经成为工业标准,形成了广泛的用户群体和专业开发人员,有力地促进了 LabVIEW 技术本身的发展与进步。从 LabVIEW 4.01 版本更新到 LabVIEW 7 版本,几乎一年更新一个版本就是 LabVIEW 技术发展和广泛应用最显著的标志。

表 1.3.1 需单独购买的主要 LabVIEW 工具包

LabVIEW 工具包名称	功 能
Signal Processing Toolkit	包括数字滤波器与联合时频分析工具包, 小波及滤波器组设计工具箱
Sound and Vibration Toolkit	声音和震动信号分析工具包
Order Analysis Toolkit	阶次分析工具包
Modulation Toolkit	调制工具包
PID Control Toolkit	提供 P、PI、PD 和 PID 控制算法并直接驱动硬件输出
Express VI Development Toolkit	Express VI 开发工具包
State Diagram Toolkit	状态图工具包, 支持从状态图直接生成代码
VI Analyzer Toolkit	VI 分析工具包, 提供自动测试工具以改进 VI 的设计
Motion Control Toolkit	运动控制工具包
Simulation Interface Toolkit	仿真与控制工具包
Database Connectivity Toolkit	数据库访问工具
Internet Toolkit	Internet 访问工具
SPC Toolkit	SPC 分析, 控制流程图, Pareto 分析
Vision and Image Processing Toolkit	高级图像分析工具
Report Generation Toolkit	报表生成工具

本书内容组织以 LabVIEW 7 Express 版本为基础。与之前的老版本相比, LabVIEW 7 Express 版本增加了许多先进功能和新特性, 下面首先对这些新功能、新特性作一概要性描述, 以便读者更好地理解本书增加的内容。

1.4 LabVIEW 7 Express 新特性

LabVIEW 7 Express 新增功能与特性概括起来可以归纳为: 进一步改进提高了编程功能与运行效率; 进一步简化了程序开发与管理; 进一步扩展了应用领域。总之, 这些新功能、新特性使得 LabVIEW 提供了创新的开发环境、全新的编程工具、交互式测量与运行方式选择, 同时具备强大的功能和方便的使用性能。这些新功能、新特性主要包括:

- 改进的 LabVIEW 启动界面, 如图 1.4.1 所示。单击 New 按钮将打开新建对话框, 该对话框包含一系列 VI 模板, 可创建基于特定模板的 VI。单击 Configure 按钮可以打开 Measurement & Automation Explorer (MAX)、创建 DAQmx 任务或创建 DAQmx 通道。

- Express VIs。又称快速 VI, 是针对常见的测试和测量应用而定制设计的 VI。一个 Express VI 通常综合了多个传统 VI 的功能于一身, 极大地简化和改进了用户的开发过程。Express VI 没有属性结点, 所有属性都通过对话框设置, 方便快捷。

- VI 模板。从主选单中选择 File→New, 打开新建对话框, 对话框列出了内建的模板, 可以选择并创建基于这些模板的 VI。