



LabVIEW LabVIEW

LabVIEW

程序设计与应用

杨乐平 李海涛 肖相生 等编著

LabVIEW LabVIEW
LabVIEW LabVIEW



(本书附光盘1张!)



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

LabVIEW 程序设计与应用

杨乐平 李海涛 肖相生 等编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以 LabVIEW(5.1 版)为对象,系统介绍虚拟仪器设计的主要思想、图形化语言编程原理与应用技术。

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、图形化编程语言的基本原理及其特点;第二部分系统介绍 LabVIEW 编程环境、语法规则、程序结构和编程技巧;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,运用大量实例阐述概念和编程难点,突出系统性和实用性。为方便学习,本书附录提供了 LabVIEW 常用技巧和各种错误代码。本书附带包含编程实例的光盘,内容丰富,供读者享用。

本书可作为大、中专院校相关专业教材或教学参考书,也可作为有关工程技术人员设计开发仪器或自动测试系统的技术参考书。

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,翻版必究。

图书在版编目(CIP)数据

LabVIEW 程序设计与应用/杨乐平等编著. - 北京:电子工业出版社,2001.7
ISBN 7-5053-6733-1

I .L … II .杨 … III .LabVIEW 语言 - 程序设计 IV .TP312

中国版本图书馆 CIP 数据核字(2001)第 029787 号

书 名: LabVIEW 程序设计与应用

编 著 者: 杨乐平 李海涛 肖相生 等

责任编辑: 段 颖

特约编辑: 印晓芬

排版制作: 电子工业出版社计算机排版室

印 刷 者: 北京兴华印刷厂

装 订 者: 三河市双峰装订厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

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: 19.5 字数: 471 千字

版 次: 2001 年 7 月第 1 版 2001 年 7 月第 1 次印刷

书 号: ISBN 7-5053-6733-1
TN·1454

印 数: 5000 册 定价: 34.00 元(含光盘)

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;
若书店售缺,请与本社发行部联系调换。电话 68279077

前 言

自 20 世纪 90 年代以来,在计算机技术的推动下,以虚拟仪器为标志的通用化、智能化和网络化测量仪器及测试系统得到了迅猛发展,使得测量仪器和数据采集系统的设计方法和实现技术产生了深刻的变化。所谓虚拟仪器技术,就是用户在通用计算机平台上,根据测试任务的需要来定义和设计仪器的测试功能,其实质是充分利用计算机来实现和扩展传统仪器功能。虚拟仪器技术综合运用了计算机技术、数字信号处理技术、标准总线技术和软件工程方法,代表了测量仪器与自动测试系统未来的发展方向。“软件就是仪器”反映了虚拟仪器技术的本质特征。

以图形化软件编程方法和集成开发环境为标志的虚拟仪器开发环境是虚拟仪器技术研究的重要内容,也是虚拟仪器技术应用与发展的技术基础。美国国家仪器公司(NATIONAL INSTRUMENTS™,简称 NI)的创新产品 LabVIEW 是目前最为成功、应用最为广泛的虚拟仪器软件开发环境,实际上,虚拟仪器的概念最初就是在开发 LabVIEW 时提出的。

本书编著者从 20 世纪 90 年代中期开始从事虚拟仪器技术的研究与开发,并且为有关专业本科生、研究生和继续教育学员开设了虚拟仪器技术和 LabVIEW 程序设计方面的课程,受到了广泛的欢迎。本书就是编著者在总结近几年教学和科研工作的基础上编写而成的。概括而言,本书编写有两个突出特点:一是本书立意不是仅仅介绍 LabVIEW 开发环境,而是把 LabVIEW 作为编程语言,将其基本概念和语法结构尽可能讲深讲透,使读者真正理解图形化编程语言与 C、BASIC 等文本编程语言的异同;二是本书结构编排和编写风格采用寓教于例、图文结合的形式,许多例子和编程技巧都是编著者多年研究和应用 LabVIEW 的经验总结,无论对初学者还是对已经具备 LabVIEW 编程能力的用户都十分有益。

参加本书编写的有国防科技大学杨乐平教授、李海涛博士、肖相生高工和钟涛博士。本书可供相关专业高年级本科生以及研究生作为教材使用,也可供从事测试计量、数据采集和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

2001 年 4 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 概念与创新	(2)
1.2 G 语言与虚拟仪器	(3)
1.3 LabVIEW 应用解决方案	(4)
第 2 章 LabVIEW 编程环境	(7)
2.1 LabVIEW 系统安装	(7)
2.2 LabVIEW 启动	(8)
2.3 LabVIEW 模板	(10)
2.4 VI 库	(12)
2.5 定制 LabVIEW 环境	(13)
第 3 章 LabVIEW 编程入门	(15)
3.1 基本概念与术语	(15)
3.1.1 基本概念	(15)
3.1.2 常用术语	(18)
3.2 创建和编辑 VI	(19)
3.2.1 创建 VI	(20)
3.2.2 编辑 VI	(26)
3.3 运行和调试 VI	(33)
3.3.1 运行 VI	(33)
3.3.2 调试 VI	(34)
3.4 创建和调用 SubVI	(36)
3.4.1 创建 SubVI	(36)
3.4.2 调用 SubVI	(38)
第 4 章 数据操作	(41)
4.1 数据类型	(41)
4.1.1 数字型	(41)
4.1.2 布尔型	(44)
4.2 数学运算	(45)
4.2.1 基本数学运算节点	(46)
4.2.2 类型转换节点	(48)
4.2.3 三角函数节点	(51)
4.2.4 对数节点	(52)
4.2.5 复数节点	(53)

4.2.6 附加常数节点	(53)
4.3 布尔运算	(55)
4.4 比较运算	(56)
第5章 变量、数组与簇	(59)
5.1 本地变量	(59)
5.1.1 本地变量的创建	(59)
5.1.2 本地变量的使用	(61)
5.1.3 本地变量的特点	(62)
5.2 全局变量	(63)
5.2.1 全局变量的创建	(63)
5.2.2 全局变量的使用	(64)
5.2.3 全局变量的特点	(65)
5.3 数组	(66)
5.3.1 数组的组成与创建	(66)
5.3.2 数组的使用	(67)
5.3.3 数组的特点	(83)
5.4 簇	(83)
5.4.1 簇的组成与创建	(83)
5.4.2 簇的使用	(85)
5.4.3 簇的特点	(94)
第6章 结构与属性控制	(97)
6.1 For 循环	(97)
6.1.1 For 循环的组成	(98)
6.1.2 For 循环的使用	(99)
6.1.3 For 循环的特点	(101)
6.2 While 循环	(103)
6.2.1 While 循环的组成	(104)
6.2.2 While 循环的使用	(104)
6.2.3 While 循环的特点	(106)
6.3 顺序结构	(107)
6.3.1 顺序结构的组成	(108)
6.3.2 顺序结构的使用	(109)
6.3.3 顺序结构的特点	(110)
6.4 选择结构	(111)
6.4.1 选择结构的组成	(112)
6.4.2 选择结构的使用	(114)
6.4.3 选择结构的特点	(115)
6.5 公式节点	(115)
6.5.1 公式节点的创建	(116)

6.5.2	公式节点的使用	(117)
6.5.3	公式节点的特点	(118)
6.6	属性节点	(118)
6.6.1	属性节点的创建	(119)
6.6.2	属性节点的使用	(120)
6.6.3	属性节点的特点	(123)
第7章	波形显示控件	(125)
7.1	事后记录波形控件(Waveform Graph)	(126)
7.1.1	控件面板结构	(126)
7.1.2	控件使用方法	(127)
7.1.3	波形显示控件外观定制	(131)
7.2	实时趋势图控件(Waveform Chart)	(137)
7.2.1	实时趋势图控件使用方法	(137)
7.2.2	实时趋势图控件的外观定制	(139)
7.3	XY 波形记录控件(XY Graph)	(141)
7.4	密度图形显示控件(Intensity Graph)	(142)
7.4.1	密度显示控件的使用方法	(143)
7.4.2	密度图形显示控件的颜色定义	(144)
7.4.3	密度图形显示控件外观设置	(147)
7.5	密度趋势图控件(Intensity Chart)	(147)
第8章	字符串与文件 I/O	(149)
8.1	字符串控件与节点	(149)
8.1.1	字符串控件	(149)
8.1.2	表格控件(Table)	(153)
8.1.3	字符串节点	(155)
8.2	文件 I/O	(165)
8.2.1	基本概念及术语	(165)
8.2.2	文件 I/O 节点	(169)
第9章	数学分析与信号处理	(183)
9.1	数学分析(Mathematics)	(183)
9.1.1	公式计算的简捷输入(Formula)	(183)
9.1.2	线性代数(Linear Algebra)	(186)
9.1.3	数学运算(Calculus)	(187)
9.1.4	概率与统计(Probability and Statistics)	(190)
9.1.5	曲线拟合(Curve Fitting)	(191)
9.2	信号处理(Signal Processing)	(193)
9.2.1	信号发生(Signal Generation)节点	(193)
9.2.2	数字信号处理节点	(195)
9.2.3	测试用节点	(197)

9.2.4 数字滤波器	(198)
9.2.5 窗函数(windows)节点	(198)
第 10 章 LabVIEW 程序设计	(201)
10.1 人机交互界面	(201)
10.1.1 程序属性设置	(201)
10.1.2 控件定制	(205)
10.1.3 菜单的编辑与响应	(206)
10.1.4 界面装饰	(208)
10.2 定时与对话框	(208)
10.2.1 定时器	(209)
10.2.2 对话框	(209)
10.2.3 错误处理节点	(210)
10.3 文件管理	(210)
10.4 创建应用程序	(213)
10.5 建立程序的 Web 文档	(216)
10.6 培养良好的编程风格	(217)
第 11 章 数据采集	(221)
11.1 数据采集的基本概念	(221)
11.1.1 DAQ 基本概念	(221)
11.1.2 DAQ VIs 的位置	(222)
11.1.3 DAQ VIs 的组织结构	(223)
11.1.4 DAQ VIs 常用参数简介	(224)
11.2 DAQ 的安装与配置	(228)
11.2.1 安装 PCI-1200 数据采集卡	(229)
11.2.2 PCI-1200 卡 I/O 配置	(231)
11.2.3 PCI-1200 卡通道配置	(234)
11.3 DAQ 应用	(239)
11.3.1 虚拟示波器结构与组成	(239)
11.3.2 仪器功能	(240)
11.3.3 软件设计与实现	(240)
第 12 章 仪器控制	(247)
12.1 仪器驱动器	(247)
12.2 VISA 标准	(250)
12.3 简单的 VISA 编程	(252)
12.3.1 VISA 节点的位置	(252)
12.3.2 基本 VISA 节点的用法	(253)
12.3.3 几个简单的 VISA 节点	(258)
第 13 章 通信	(261)
13.1 串行通信	(261)

13.1.1 串行通信节点的位置	(261)
13.1.2 串行通信节点的使用方法	(262)
13.1.3 串行通信应用举例	(264)
13.2 局域网通信	(266)
13.2.1 TCP/IP 节点的位置	(266)
13.2.2 TCP/IP 节点的使用方法	(267)
13.2.3 网络通信应用举例	(268)
附录 A LabVIEW 常用操作技巧	(271)
附录 B VISA 错误代码表	(273)
附录 C 数据采集 (DAQ) 错误代码表	(277)
附录 D TCP 与 UDP 错误代码表	(289)
附录 E 串行通信错误代码表	(290)
附录 F 公式节点错误表	(291)
附录 G 数学分析 (Mathematics) 错误代码表	(292)
附录 H 信号处理 (Signal Processing) 错误代码表	(294)
附录 I 仪器驱动器 (Instrument Driver) 错误代码表	(297)
附录 J DDE 错误代码表	(298)
附录 K GPIB 错误代码表	(299)
参考文献	(300)

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讯口来控制 IEEE-488 总线仪器（也称为 GPIB 程控仪器）。后来，随着 PC 机的出现，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 机控制台式仪器的方法，各种基于 PC 机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 机 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），但与所有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，尤其是对那些很少编程或基本没有编程经验的用户更是如此。

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, 柯德斯凯在德州首府奥斯汀德州大学附近安营扎寨, 这样一方面可以利用德州大学的图书馆资源, 另一方面也便于雇佣学生程序员。

1.1.2 LabVIEW 概念与创新

LabVIEW 的概念雏形来源于特鲁查德和柯德斯凯两人于 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纳探测器, 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, 因为它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: 一是超过 18 万人的开发费用; 二是用户必须理解菜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 可以由更底层的多个 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, 核心程序完成后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。而 VI 模型的前面板接口是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,麦克里森表现出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很难完成的应用程序。

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 年推出的 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,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 由交互式用户接口、数据流框图和图标连接端口组成,各部分功能如下。

① VI 的交互式用户接口因为与真实物理仪器面板相似,又称作前面板。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并显示结果。

② VI 从数据流框图接收指令。框图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 VI 模块化特性。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,称作 SubVI。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。

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,经过短短不到 15 年的时间,已经发展到以最新版本 LabVIEW 6i 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/98、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,包括独立代码生成工具、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 6i 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
PID Control Toolkit	提供 P、PI、PD 和 PID 控制算法并直接驱动硬件输出
Fuzzy logic Control Toolkit	提供实现模糊控制的 VI
Control and Simulation Toolkit	线性与非线性仿真与控制,允许使用控制流程图设计控制系统
SQL Toolkit	提供完整的 SQL 操作功能
Signal Processing Toolset	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
SPC Toolkit	SPC 分析,控制流图, Pareto 分析
Gmath Toolkit	高级数学分析工具,包括差分方程、最优化、拉普拉斯变换和积分/微分等
IMAQ Vision	高级图像分析工具

第2章 LabVIEW 编程环境

2.1 LabVIEW 系统安装

考虑到目前国内用户的实际情况和本书提供例子程序为用户直接应用的能力, 本书内容组织以 1999 年推出的 LabVIEW 5.1 版本为基础。最新的 LabVIEW 6i 版本在功能布局 and 开发环境设计上有一些新的改进, 除了例子程序应用可能需要修改外, 本书介绍的其他内容也完全适用于 LabVIEW 6i 的用户。

LabVIEW 5.1 for Windows 98/NT 的安装十分简单, 只需运行 Install Disk 或 CD 中的 Setup 程序, 按照屏幕提示, 一步步选择必要的安装选项即可完成。整个系统安装时间取决于硬件平台和选择的安装选项, 一般在 20 分钟之内即可完成。LabVIEW 所有文件约占 110 MB 左右的硬盘容量。安装完成后, LabVIEW 系统目录包含的主要文件和文件夹如下:

LabVIEW.exe LabVIEW 启动程序;

LabVIEW.ini LabVIEW 配置选项文件;

Vi.lib 目录 存放 LabVIEW 自带 VI 库文件, 包括 GPIB、数据采集和数学分析等 VI, 此目录中的多数 VI 都可以从 LabVIEW 功能模板中调用;

Examples 目录 存放大量 LabVIEW 编程实例程序, 其中 readme.vi 文件是关于例子的使用指南;

Resource 目录 存放 LabVIEW 应用程序需要的数据文件、动态连接库文件和 ActiveX 服务文件等资源;

Cintools 目录 存放连接 C 代码与 LabVIEW VI 必需的文件;

Project 目录 存放 LabVIEW Project 菜单项目的支持文件;

Menus 目录 存放控制模板和功能模板的结构配置文件;

Instr.lib 目录 存放 VXI 仪器、GPIB 仪器和串行接口仪器驱动程序;

Help 目录 存放完整的在线文档和例子帮助文件;

Tutorial 目录 存放运行在线教学软件必需的文件;

Activity 目录 存放在用户手册中提供的编程实例 VI 文件;

User.lib 目录 存放用户自己创建的 VI, 此目录的 VI 可以从 LabVIEW 功能模板中的 user Libraries 子模板调用。

为了控制 VXI、GPIB 和 DAQ 设备, 在 LabVIEW 系统安装完成后, 还必须运行专门仪器驱动器和 VISA 库函数的安装程序。一般随 LabVIEW 系统光盘, NI 公司还提供一张专门的仪器驱动器光盘供用户选择安装。

2.2 LabVIEW 启动

运行 LabVIEW 执行程序或双击 LabVIEW 图标后, LabVIEW 5.1 的启动画面如图 2.2.1 所示。

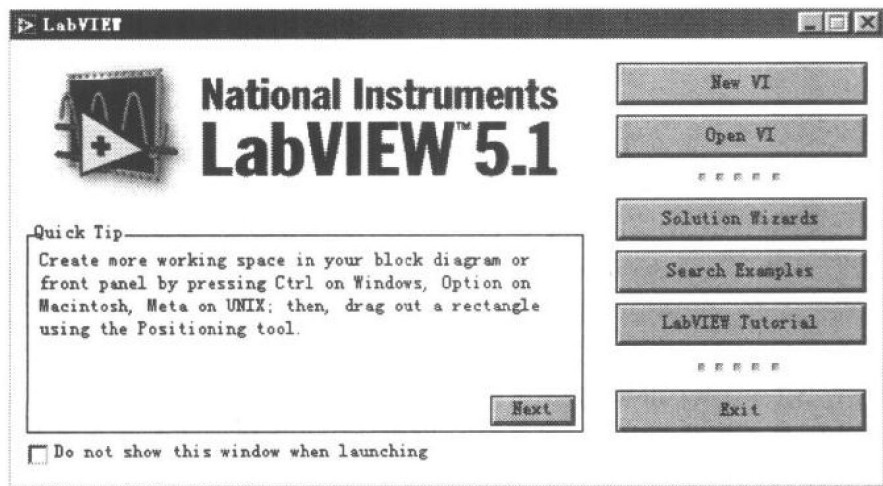


图 2.2.1 LabVIEW 5.1 的启动画面

各按钮功能如下:

New VI	创建一个新的 VI。
Open VI	打开一个 VI。
Solution Wizards	启动一个交互式实用程序, 允许用户配置和定制数据采集系统。
Search Examples	启动一个实用程序, 列出并打开可供用户选择的 LabVIEW 例子。
LabVIEW Tutorial	执行一个交互式在线教学多媒体软件, 演示一个完整的教学过程大约需要 15 分钟。
Exit	退出 LabVIEW 应用程序。

像许多 Windows 应用程序一样, LabVIEW 启动画面有一栏显示快速提示, 通过单击 NEXT 按钮, 可以浏览更多的 LabVIEW 编程技巧提示。屏幕左下端的开关允许用户选择简化的启动窗口, 简化窗口只列出创建 VI、打开 VI 和退出 LabVIEW 三个按钮。

当用户单击 New VI 按钮进入 LabVIEW 编程环境后, 首先出现在用户面前的是两个无标题 (untitled) 窗口。一个是前面板窗口, 用于编辑和显示 VI 前面板对象; 另一个是框图程序窗口, 用于编辑和显示流程图 (程序代码)。两个窗口菜单条设计完全一样, 除了框图程序窗口增加了 4 个用于程序调试的工具按钮以外, 两个窗口工具条的设计也是一致的。下面以框图程序窗口为例, 说明窗口工具条和窗口弹出式菜单的布局与功能。

1. 窗口工具条

窗口工具条中各图标的功能如表 2.2.1 所示。