



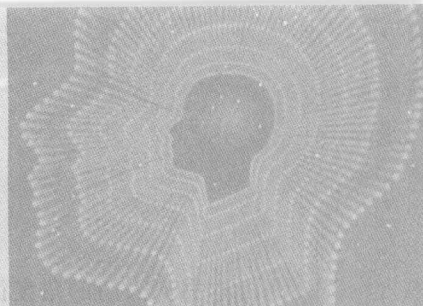
LabVIEW 8.5

快速入门与提高

孙秋野 柳 昂 王云爽 编著



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS



LabVIEW 8.5

快速入门与提高

孙秋野 柳 昂 王云爽 编著



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

内容简介

本书从入门的角度循序渐进地讲解了 LabVIEW 8.5 的基本应用技术,通过理论与实例相结合的方式,深入浅出地介绍了 LabVIEW 8.5 的使用方法和技巧。大量详实的例子都给出了具体的创建过程和程序具体的运行过程,使读者在学习感到轻松易懂。

全书共分 12 章,每个章节都配有实例,读者可以快捷地掌握 LabVIEW 8.5 的基础知识和编程方法。本书适用于各个领域的工程技术人员。

图书在版编目(CIP)数据

Labview 8.5 快速入门与提高/孙秋野,柳昂,王云爽编著. —西安:西安交通大学出版社,2009.5
ISBN 978-7-5605-3046-8

I. L… II. ①孙…②柳…③王… III. 软件工具,Labview 8.5-程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字(2009)第 011514 号

书 名 LabVIEW 8.5 快速入门与提高
编 著 孙秋野 柳昂 王云爽
责任编辑 屈晓燕 贺峰涛

出版发行 西安交通大学出版社
(西安市兴庆南路 10 号 邮政编码 710049)

网 址 <http://www.xjtpress.com>
电 话 (029)82668357 82667874(发行中心)
(029)82668315 82669096(总编办)

传 真 (029)82668280
印 刷 西安市新城区兴庆印刷厂

开 本 787mm×1 092mm 1/16 印张 19.75 字数 479 千字
版次印次 2009 年 5 月第 1 版 2009 年 5 月第 1 次印刷
书 号 ISBN 978-7-5605-3046-8/TP·517
定 价 32.00 元

读者购书、书店添货、如发现印装质量问题,请与本社发行中心联系、调换。

订购热线:(029)82665248 (029)82665249

投稿热线:(029)82664954

读者信箱:jdlgy@yahoo.cn

版权所有 侵权必究

前 言

虚拟仪器技术是由计算机技术、测量技术和微电子技术高速发展而孕育出的一项革命性技术,是用户在通用计算机平台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。

虚拟仪器是虚拟技术的一个重要组成部分。虚拟仪器的实质是利用计算机显示器的显示功能来模拟传统仪器的控制面板,以多种形式表达输出检测结果;利用计算机强大的软件功能实现信号数据的运算、分析和处理;利用 I/O 接口设备完成信号的采集、测量与调试,从而完成各种测试功能的一种计算机仪器系统。使用者用鼠标或键盘操作虚拟面板,就如同使用一台专用测量仪器一样。早期的虚拟仪器技术主要用于军事、航空、航天等领域和科研院所,现在已经越来越多地出现在了工厂及其他的民用场合。

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, NI) 推出的一种基于“图形”方式的集成化程序开发环境,是目前国际上惟一的图形化编程语言。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。由于 LabVIEW 是利用流程图来完成测试任务,基本上不用写程序代码,这大大降低了对 LabVIEW 的使用者编程能力的要求,同时也在很大程度上拓宽了 LabVIEW 使用者的范围,各个领域的工程技术人员都可以使用它来进行开发。

LabVIEW 8.5 是目前 NI 公司推出的 LabVIEW 的最新版本,它的发布大大缩小了软件的易用性和强大功能之间的差距,更使其应用于测量和从测量中获取有用信息的过程变得比以往任何时候都更容易、更准确、更迅速。

本书从入门的角度循序渐进地讲解了 LabVIEW 8.5 的基本应用技术,通过理论与实例相结合的方式,深入浅出地介绍了 LabVIEW 8.5 的使用方法和技巧。大量详实的例子都给出了具体的创建过程和程序具体的运行过程,使读者在学习中感到轻松易懂。

全书共分为 12 章,每个章节中都配有实例,读者可以快捷地掌握 LabVIEW 8.5 的基本知识和编程方法。

第 1 章主要介绍了虚拟技术的有关概念,虚拟仪器和虚拟技术的关系,G 语言的特点和 LabVIEW 的概念及其应用。

第 2 章介绍了 LabVIEW 的安装所需的硬件配置、安装过程、LabVIEW 编辑界面及系统菜单、最基本的选板工具和 LabVIEW 帮助系统,为以后的学习打下最基本的基础。

第 3 章主要讲解了创建、编辑及修改 VI 对象的方法和技巧。

第 4 章结合简单的实例介绍了 LabVIEW 中基本数据类型数值型、布尔型和字符型,以及程序设计的基本结构。

第 5 章在实例中说明了子 VI 在 LabVIEW 中是如何实现的。使用子 VI 可以使程序框图更加简化,通过应用子 VI 把程序模块化,也增加了程序的易读性。

第 6 章介绍了前面板的设计。包括 VI 属性的设置、用户菜单的设计、属性节点的使用和

控件的自定义。在 VI 属性窗口中可以查看 VI 的属性并对属性进行自定义。通过使用用户菜单编辑器或菜单函数可以自定义用户菜单。LabVIEW 中的控件、VI 等都有预定义的数据和动作,通过属性节点的使用可以激活这些属性,实现很多高级的控制功能。在 LabVIEW 中可以使用控件编辑器改变原有控件的外观属性,定制符合用户需求的控件图标。

第 7 章主要讲述了字符串控件和表格与列表控件的使用方法,其次介绍了字符串函数的功能和操作方法,使用这些函数可以方便地实现字符串的处理。此外,使用字符串函数中的丰富类型转换函数,可以实现字符串与数值、数组和路径等数据类型的转换。

第 8 章介绍了 9 种文件存储格式,并分析了各种存储方式的使用范围和优、缺点。

第 9 章主要介绍了常用的图形显示控件及其使用方法。通过本章的学习读者可以学会根据需要选择不同的显示控件。通过使用属性节点用户可以方便地改变图形控件的属性,使图形显示更具有可操作性和多样性。

第 10 章主要介绍了数据库和 SQL 语言的一些基本知识,同时重点介绍了三种数据库访问和操作方法。

第 11 章主要介绍了数据采集的相关知识和概念。

第 12 章主要介绍了总线和通信的相关知识和概念。

在本书的编写过程中,得到了黎明副教授、王志强博士、张铁岩教授、刘国威高级工程师的大力支持,同时葛辉、刘鑫蕊、罗艳红、梁绵鑫、赵琰、马铁东、魏庆来、张锐等朋友也给予了很多帮助,还要感谢陶洪生、姚志强和王健在本书的编写过程中给予的无私帮助与大力支持,帮助我们解决了很多具体问题,在终稿的排版及校阅过程中盛科、辛红、孙羽、付振宇、孙艳、曹建赟、朱莱等也付出了很多劳动,在此一并表示感谢。

由于编者水平有限,书中难免有不妥和错误之处,敬请读者批评指正。

本书在编写过程中,参考了国内外许多文献,在此一并表示感谢。

编者 2007 年 10 月

本书在编写过程中,参考了国内外许多文献,在此一并表示感谢。

本书在编写过程中,参考了国内外许多文献,在此一并表示感谢。

本书在编写过程中,参考了国内外许多文献,在此一并表示感谢。

本书在编写过程中,参考了国内外许多文献,在此一并表示感谢。

本书在编写过程中,参考了国内外许多文献,在此一并表示感谢。

本书在编写过程中,参考了国内外许多文献,在此一并表示感谢。

本书在编写过程中,参考了国内外许多文献,在此一并表示感谢。

目 录

第 1 章 绪论	(1)	3.3.10 为对象上色	(46)
1.1 G 语言与虚拟仪器概述	(1)	3.4 调试 VI	(47)
1.1.1 G 语言的概念	(1)	3.4.1 错误信息	(47)
1.1.2 虚拟仪器的概念	(2)	3.4.2 加亮执行	(48)
1.1.3 虚拟仪器的构成及其分类	(2)	3.4.3 单步执行	(48)
1.2 LabVIEW 概述及应用	(3)	3.4.4 断点与探针	(49)
1.2.1 LabVIEW 概述	(3)	本章小结	(49)
1.2.2 LabVIEW 的应用	(4)	第 4 章 数据类型与程序控制	(50)
1.2.3 LabVIEW 8.5 新特性	(5)	4.1 数据类型及其操作	(50)
本章小结	(7)	4.1.1 数值型	(50)
第 2 章 LabVIEW 8.5 开发环境	(8)	4.1.2 布尔型	(59)
2.1 LabVIEW 系统安装	(8)	4.1.3 字符串与路径	(61)
2.2 LabVIEW 启动	(11)	4.2 LabVIEW 的数据结构	(63)
2.3 LabVIEW 编辑界面及系统菜单	(12)	4.2.1 数组	(63)
2.3.1 LabVIEW 的编辑界面	(12)	4.2.2 簇	(72)
2.3.2 LabVIEW 的系统菜单	(13)	4.3 程序控制	(74)
2.4 LabVIEW 选板	(17)	4.3.1 循环控制	(74)
2.4.1 工具选板	(17)	4.3.2 顺序结构	(77)
2.4.2 控件选板	(19)	4.3.3 条件结构	(79)
2.4.3 函数选板	(21)	4.3.4 事件结构	(82)
2.5 LabVIEW 帮助系统	(23)	4.3.5 移位寄存器、隧道和反馈节点	(88)
本章小结	(26)	4.3.6 局部变量与全局变量	(94)
第 3 章 基本操作——创建、编辑与调试	(27)	4.3.7 公式节点与脚本	(99)
3.1 LabVIEW 的基本概念	(27)	本章小结	(104)
3.2 创建 VI	(27)	第 5 章 子 VI 的实现	(105)
3.2.1 创建方法	(27)	5.1 子 VI 的概念	(105)
3.2.2 创建实例	(29)	5.2 连接器和图标	(106)
3.3 编辑 VI	(32)	5.2.1 设置连接器端口	(106)
3.3.1 选择对象	(32)	5.2.2 定制子 VI 个性图标	(109)
3.3.2 移动对象	(33)	5.3 创建子 VI	(110)
3.3.3 删除和复制对象	(34)	5.3.1 用现有 VI 创建子 VI	(110)
3.3.4 调整对象大小	(35)	5.3.2 选定内容创建成子 VI	(112)
3.3.5 标签对象	(36)	5.4 添加子 VI 至用户库	(115)
3.3.6 字体、字形和文本	(36)	5.5 调用子 VI	(116)
3.3.7 编辑连线	(38)	5.6 层次窗口	(117)
3.3.8 对象的大小、对齐和分布	(42)	本章小结	(119)
3.3.9 组合和锁定对象	(45)		

第 6 章 前面板的设计	(120)	8.3.9 图形文件	(182)
6.1 VI 属性的设置	(120)	8.3.10 文件 I/O Express VI	(183)
6.1.1 常规属性页	(121)	本章小结	(184)
6.1.2 内存属性页	(121)	第 9 章 图形显示控件的配置与使用	(186)
6.1.3 说明信息属性页	(121)	9.1 图形控件的分类	(186)
6.1.4 修订历史属性页	(122)	9.2 波形图表	(187)
6.1.5 编辑器选项属性页	(123)	9.2.1 波形图表外观与属性的配置	(187)
6.1.6 保护属性页	(123)	9.2.2 数据输入类型及实例分析	(194)
6.1.7 窗口外观属性页	(124)	9.3 波形图	(197)
6.1.8 窗口大小属性页	(124)	9.3.1 波形图表的数据输入类型	(197)
6.1.9 窗口运行时位置属性页	(125)	9.3.2 游标图例的创建与使用	(199)
6.1.10 执行属性页	(125)	9.4 XY 图	(202)
6.1.11 打印选项属性页	(126)	9.4.1 XY 图数据输入类型	(202)
6.2 用户菜单的设计	(127)	9.4.2 Express XY 图的使用	(204)
6.2.1 菜单编辑器	(127)	9.4.3 XY 图表实例分析:蝴蝶效应图 的实现	(205)
6.2.2 菜单函数模板	(128)	9.5 强度图	(208)
6.2.3 用户菜单编程实例	(131)	9.6 数字波形图	(210)
6.3 属性节点	(134)	9.7 三维数据显示图	(212)
6.3.1 直接创建属性节点	(135)	9.7.1 三维曲面图	(213)
6.3.2 编程方法创建属性节点	(139)	9.7.2 三维参数曲面图	(217)
6.3.3 属性节点使用实例	(140)	9.7.3 三维曲线图	(218)
6.4 控件定制	(142)	本章小结	(219)
本章小结	(147)	第 10 章 数据库的访问	(220)
第 7 章 字符串的实现	(148)	10.1 LabVIEW 访问数据库的途径	(220)
7.1 字符串	(148)	10.2 关系数据库标准语言 SQL	(221)
7.1.1 字符串控件	(148)	10.2.1 SQL 语言的基本概念	(221)
7.1.2 列表与表格	(151)	10.2.2 常见 SQL 语句的使用	(222)
7.2 字符串函数的使用	(153)	10.3 Microsoft ADO 编程基础	(225)
本章小结	(159)	10.3.1 ADO 简介	(225)
第 8 章 文件 I/O 的实现	(160)	10.3.2 ADO 对象模型	(226)
8.1 文件的类型	(160)	10.4 使用 LabSQL 工具包访问数据库	(234)
8.2 基本文件 I/O 函数	(162)	10.4.1 LabSQL 工具包介绍	(234)
8.3 常用文件类型的使用	(165)	10.4.2 数据源的创建	(240)
8.3.1 电子表格文件	(165)	10.4.3 数据源的连接	(242)
8.3.2 文本文件	(166)	10.4.4 LabSQL 工具包应用实例分析	(248)
8.3.3 二进制文件	(168)	10.4.5 密码登陆及用户管理界面的 设计	(251)
8.3.4 波形文件	(170)	10.4.6 调用 ADO 控件的方法访问数	
8.3.5 数据存储文件和 TDM 流文件	(173)		
8.3.6 数据记录文件	(178)		
8.3.7 配置文件	(180)		
8.3.8 XML 文件	(181)		

数据库	(258)	本章小结	(283)
10.5 使用 Database Connectivity 工具包		第 12 章 总线与通信	(284)
访问数据库	(260)	12.1 仪器总线概述	(284)
10.5.1 Database Connectivity 工具包		12.1.1 GPIB	(284)
介绍	(260)	12.1.2 VXI	(286)
10.5.2 Database Connectivity 工具包		12.1.3 PXI	(289)
应用实例分析	(263)	12.1.4 PCI	(293)
本章小结	(266)	12.1.5 总线平台的比较	(295)
第 11 章 数据采集	(267)	12.2 通信	(296)
11.1 DAQ 系统概述	(267)	12.2.1 TCP/IP 与 UDP 协议	(297)
11.1.1 DAQ 系统的构成	(267)	12.2.2 DataSocket 通信	(303)
11.1.2 DAQ 系统的功能	(268)	本章小结	(306)
11.2 设备驱动及 PCI-6221 的安装 ...	(272)	参考文献	(307)
11.3 DAQ VI 的组织结构	(277)		
11.4 使用 DAQ Assistant	(279)		

第1章

绪论

LabVIEW 是一款功能强大而又复杂的开发软件。熟练的掌握 LabVIEW 将在很大程度上减轻了很多领域的工程技术人员的编程压力。本章作为入门章节,将从虚拟仪器的概念、G 语言的优势以及 LabVIEW 的发展历程、应用领域等方面开始,让读者对 LabVIEW 有一个初步了解。

1.1 G 语言与虚拟仪器概述

1.1.1 G 语言的概念

G 语言是 LabVIEW 采用的图形化编程的语言。LabVIEW 是一个功能比较完整的软件开发环境,它是为替代常规的 BASIC 或 C 语言而设计的。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。

LabVIEW 的动态连续跟踪方式可以连续、动态地观察程序中的数据及其变化情况。但是与现有的计算机高级语言不同的是,LabVIEW 采用图形化编程语言——G 语言,产生块状的程序。用 LabVIEW 编程的过程就像设计电路图一样,因此,LabVIEW 比其他语言的开发环境更方便、更有效。

G 语言是一种适合于任何编程任务,具有扩展函数库的通用编程语言。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 Instrument),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。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 语言有丰富的扩展函数库。这些扩展函数库主要面向数据采集、GPIB 和串行仪器控制、数据分析、数据显示与数据存储。G 语言还包括常用的程序调试工具,比如单步调试、允许设置断点、数据探针和动态显示执行程序流程

等功能。

1.1.2 虚拟仪器的概念

所谓虚拟仪器,就是在以计算机为核心的硬件平台上,其功能由用户设计和定义,具有虚拟面板,其测试功能由测试软件实现的一种计算机仪器系统。虚拟仪器的实质是利用计算机显示器的显示功能来模拟传统仪器的控制面板,以多种形式表达输出检测结果;利用计算机强大的软件功能实现信号数据的运算、分析和处理;利用 I/O 接口设备完成信号的采集、测量与调试,从而完成各种测试功能的一种计算机仪器系统。使用者用鼠标或键盘操作虚拟面板,就如同使用一台专用测量仪器一样。因此,虚拟仪器的出现,使测量仪器与计算机的界限模糊了。

虚拟仪器(Virtual Instrument,简称 VI)是虚拟技术的一个重要组成部分。虚拟仪器技术是由计算机技术、测量技术和微电子技术高速发展而孕育出的一项革命性技术,是用户在通用计算机平台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。虚拟仪器的特点表现在:各种仪器的差异主要在软件的方面,尽可能采用通用的硬件;有强大的数据处理功能,可以充分发挥计算机的能力;用户可以根据自己的需要来定义出各种仪器。

虚拟仪器是现代计算机技术和仪器技术深层次结合的产物,是当今计算机辅助测试(CAT)领域的一项重要技术。虚拟仪器是计算机硬件资源、仪器与测控系统硬件资源和虚拟仪器软件资源三者的有效结合。

虚拟仪器的“虚拟”两字主要包含以下两方面的含义:(1)虚拟仪器的面板是虚拟的;(2)虚拟仪器测量功能是通过图形化软件流程图的编程来实现的。

虚拟仪器具有性能高、扩展性强、开发时间少、无缝集成四大优势。与传统仪器相比它有以下功能特点:功能由用户自己定义;可方便的与网络外设及多种仪器连接;界面图形化,计算机直接读取数据并分析处理;数据可编辑、存储、打印;软件是关键部分;价格低廉,仅是传统仪器的 10%~20%;基于计算机技术开放的功能模块可构成多种仪器;技术更新快;基于软件体系的结构可大大节省开发费用。

1.1.3 虚拟仪器的构成及其分类

虚拟仪器由通用仪器硬件平台(简称硬件平台)和应用软件两大部分构成,如图 1-1 所示。

1. 虚拟仪器的硬件平台

构成虚拟仪器的硬件平台有两部分。

(1) 计算机

它一般为一台 PC 机或者工作站,是硬件平台的核心。

(2) I/O 接口设备

I/O 接口设备要完成被测输入信号的采集、放大、模/数转换。不同的总线有其相应 I/O 接口硬件设备,虚拟仪器的构成方式主要有 5 种类型:PC-DAQ 系统、GPIB 系统、VXI 系统、PXI 系统和串口系统。

PC-DAQ 系统是以数据采集板、信号调理电路及计算机为仪器硬件平台组成的插卡式

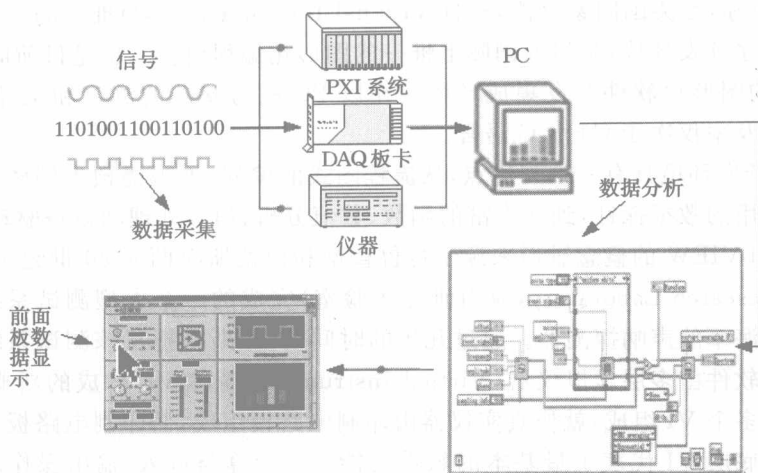


图 1-1 虚拟仪器的构成

虚拟仪器系统。这种系统采用 PCI 或计算机本身的 ISA 总线,将数采卡/板(DAQ)插入计算机的空槽中即可。

GPIB 系统是以 GPIB 标准总线仪器与计算机为仪器硬件平台组成的虚拟仪器测试系统。

VXI 系统是以 VXI 标准总线仪器模块与计算机为仪器硬件平台组成的虚拟仪器测试系统。

PXI 系统是以 PXI 标准总线仪器模块与计算机为仪器硬件平台组成的虚拟仪器测试系统。

串口系统是以 Serial 标准总线仪器测试系统与计算机为仪器硬件平台组成的虚拟仪器测试系统。

无论上述哪种 VI 系统,都通过应用软件将仪器硬件与计算机相结合。

2. 虚拟仪器的软件

开发虚拟仪器必须有合适的软件工具,目前的虚拟仪器软件开发工具只有如下两类。

文本式编程语言:如 Visual C++, Visual Basic, LabWindows/CVI 等。

图形化编程语言:如 LabVIEW, HPVEE 等。

这些软件开发工具为用户设计虚拟仪器应用软件提供了最大限度的方便条件与良好的开发环境。

虚拟仪器软件由两部分构成,即应用程序和 I/O 接口仪器驱动程序。虚拟仪器的应用程序包含两方面功能的程序:

- (1) 实现虚拟面板功能的软件程序;
- (2) 定义测试功能的流程图软件程序。

1.2 LabVIEW 概述及应用

1.2.1 LabVIEW 概述

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, NI)推出的一种基于“图形”方式的集成化程序开发环境,是目前国际上惟一的图形化编程语言。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。在以 PC 机为基础的测量和工控软件中,LabVIEW 的市场普及率仅次于 C++/C 语言。

LabVIEW 开发环境具有一系列优点,从流程图式的编程,不需要预先编译就存在语法检测和调试过程使用的数据探针,到其丰富的函数、数值分析、信号处理和设备驱动等功能,都是十分优越的。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。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 Instrument, 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/输出操作。柯德斯凯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。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。

LabVIEW 被公认为是标准的数据采集和仪器控制软件,LabVIEW 是一种面向最终用户的开发工具,可为实现仪器编程和数据采集系统提供便捷途径。

1.2.2 LabVIEW 的应用

LabVIEW 在测试与测量、过程控制和工业自动化和实验室研究与自动化等方面都得到了广泛的应用。世界范围内,汽车、通信、航空、半导体、电子设计生产、过程控制和生物医学等各领域均通过 LabVIEW 提高了应用开发的效率,涵盖了从研发、测试、生产到服务的产品开发所有阶段。使用 LabVIEW 可以实现和完成在任何平台上进行数据采集、仪器控制和连接、机器视觉、运动控制、模块化仪器、工业监控、开关(Switching)方案等众多方面的功能和任务。

LabVIEW 将广泛的数据采集、分析与显示功能集中在了同一个环境中,可以在自己的平台上无缝地集成一套完整的应用方案。LabVIEW 是一个开放式的开发环境,用户可以将其与任何测量硬件轻松连接。LabVIEW 的交互式测量助手、自动代码生成以及与成千上万个设备的简易连接功能,能够轻而易举地完成数据采集。LabVIEW 简化了与数百家仪器厂商的数千种仪器设备的连接和通信。使用 LabVIEW 中的仪器驱动程序、交互式仪器 I/O 助手(Instrument I/O Assistant)和内置仪器 I/O 函数库,可以从 GPIB、串口、以太网、PXI、USB 接口仪器及 VXI 仪器中快速采集数据。

LabVIEW 带有超过 450 个内置函数,专门用于从采集到的数据中挖掘有用的信息,用于分析测量数据及处理信号。LabVIEW 提供一系列工具用于数据显示、用户界面设计、Web 信

息发布、报告生成、数据管理及软件连接。只需简单地从控件选板中拖放内置的控制件和显示件,然后点击鼠标即可利用交互式的属性页面轻松自定义它的功效和外观。

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、TCO/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 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、原代码控制、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。

1.2.3 LabVIEW 8.5 新特性

LabVIEW 软件自 1986 年第一个版本问世以来,就以其图形化的编程理念在工程业界引起了广泛的关注。20 年来,NI 的研发团队不断带来功能上的改进和扩展,使这一软件平台一直保持着创新的发展历程。LabVIEW 8.5 构建在 NI 已投资 10 年的多线程编程技术之上,通过直觉的并行数据流语言,简化了多内核与 FPGA 架构的应用开发。随着处理器制造商寻求并行多内核架构提升性能,运行在新的处理器上的 LabVIEW 8.5 能提供更快的测试吞吐量、更有效率的处理和分析能力以及更为可靠的实时系统。LabVIEW 8.5 在以图形化语言集成了包括数据流、C 代码、计算公式、仿真、状态表等高级设计模式的同时,也通过新的状态图设计模块(Statechart Design Module),使得 LabVIEW 平台进一步增强大型事件驱动项目的开发效率。而且在支持台式机平台、实时系统、FPGA 以及微处理器等领域都进行了升级支持。

LabVIEW 8.5 的推出意味着 NI 产品不断的与时俱进。

最新推出的 LabVIEW 8.5 三个关键的新特点在于支持多内核平台、进行多线程并行计算、增强可编程能力。

1. 多内核与多线程并行计算提高处理能力

LabVIEW 将自动或手动将每个应用分解到多执行线程中,并对处理进行优化。LabVIEW 8.5 的新特性之一在于支持多内核处理器和多线程处理。通过 LabVIEW 并行数据流语言,用户可轻松通过并行编程构架来编写多线程程序,并映射至多内核架构。LabVIEW 8.5 还可以根据可用的内核数量,调整线程总数(total number of threads),并提供增强的线程安全驱动程序与链接库,从而提升处理性能,减少使用多核处理器的复杂性。

LabVIEW 8.5 也可通 LabVIEW Real-Time 环境,提供对称多线程处理功能,让嵌入式与工业级系统的工程师不需牺牲特定系统性能,即可跨多重内核自动下载平衡任务。利用最新

版本的 LabVIEW, 用户可手动指派部份程序代码至特定处理器内核, 以微调实时系统或将时间重要的程序代码隔离到专用内核。工程师与科学家现在可以使用新的 NI Real-Time Execution Trace Toolkit 2.0, 以真实地显示程序代码区段之间的时序关系以及正在执行程序代码的个别线程与处理内核, 满足了多内核开发时日渐增多的除错与最优化程序代码的挑战。

2. FPGA 增加仪器配置的灵活性

LabVIEW 的固有并行机制也使之成为开发 FPGA 应用的理想平台。LabVIEW 8.5 通过增强的 PFPGA 模块, 持续简化 FPGA 的编程设计任务。该模块可自动进行 I/O 配置、IP 开发、通用 I/O 的整体设定、计数器/定时器和编码器的应用。利用 FPGA 模块, 可自动产生复杂的高速 DMA 数据传输程序代码。此外, LabVIEW 8.5 提供多信道滤波功能与机器自动化中广泛需要的 PID 控制函数, 从而大幅减少高信道数应用的 FPGA 资源。目前, NI 和其合作伙伴可提供超过 60 个可应用于 LabVIEW FPGA 的 IP 内核和实例。

这种利用简洁的 LabVIEW 图形界面直接进行 FPGA 配置, 并实现不同 I/O 功能的创新对于使用 LabVIEW 的用户来说无疑会带来很多益处, 最重要的就是领域专家可通过 LabVIEW 平台利用 FPGA 技术而无需花费时间去掌握硬件设计知识或 VHDL 编程语言。在这个模块中, NI 考虑到了用户在使用过程中的特定应用编程需求和有代表性应用的高效率 IP, 在易用性和器件利用率上都有所考虑和折衷。

对 FPGA 的重视加上 LabVIEW 的图形化特性, 将使 NI 产品的适用性和灵活性得以进一步加强。

3. 增强大型事件驱动项目的开发效率

LabVIEW 8.5 增加了新的状态图模块, 利用基于统一模式语言 (UML) 标准的类似的高阶状态图符号, 可协助工程师设计并仿真这些基于事件的系统。

因为 LabVIEW Statechart Module 是以 LabVIEW 图形化编程语言为基础, 工程师可用单一平台迅速地设计、原型制作及发布他们的系统, 以运行在实时或基于 FPGA 架构系统上的实际 I/O 来组合相似的状态图符号。

4. 兼容不同 I/O, 支持不同平台

LabVIEW 8.5 新增多种 I/O 阵列、测量与显示增强功能, 适于建立基于 PAC 的工业级系统, 包含可为 LabVIEW 用户扩充工业级连接的 OPC 驱动程序, 几乎把可兼容的 PLC 与工业级设备的数量翻了一番。

LabVIEW 8.5 还为工业用计算机监控系统新增了振动与顺序追踪测量及机器视觉算法。对于高信道数的系统, 新的多重变量编辑软件通过简单的数据表接口, 可让用户轻松并快速地配置或编辑数百种的 I/O 标签。此外, 最新版本的 LabVIEW 具有新的弹性管线显示工具, 以简化现实工业级用户接口的创建程序; 而其互动的拖曳方式可直接将 I/O 标签直接固定于用户接口显示器, 这种接口显示器运行在基于 Windows CE 的工业级触控面板及手持式 PDA 上。

LabVIEW 8.5 新功能具体体现在很多方面, 比如管理用户的大型 LabVIEW 应用程序, 使用 LabVIEW 8.5, 用户可以更轻松地开发并部署大型应用; 应用分布式和实时系统, 不论是将应用发布于同一台机器的不同进程, 或是将应用发布于各远程设备的不同架构 (如 FPGA 和 DSP) 上, LabVIEW 8.5 的分布式智能总能助您轻松开发多处理器系统; 全新仪器控制技

术,LabVIEW 8.5 有业界领先的仪器控制软件包,LabVIEW 8.5 以其全新的技术令仪器控制更进一步;全新 GUI 控制,利用 LabVIEW 8.5 全新的用户界面对象和功能,用户能开发出专业化的可完全自定义的前面板;信号处理分析和数学运算,LabVIEW 8.5 对数学、信号处理和分析进行了重大的补充和完善,LabVIEW 最新发布的版本中,信号处理分析和数学运算具有更为全面和强大的库,其中包括 500 多个函数;全新编码结构,LabVIEW 8.5 引入了新的编码结构,以增强 LabVIEW 的应用开发能力并提高代码运行的效率。

本章小结

在这一章中主要介绍了 LabVIEW 的相关的背景知识、G 语言的概念和虚拟仪器的强大功能及应用,最后简述了 LabVIEW 8.5 的新特性,让读者对最新的 LabVIEW 版本有更深刻的认识。现在就开始我们的 LabVIEW 之旅。

第 2 章

LabVIEW 8.5 开发环境

2.1 LabVIEW 系统安装

安装 LabVIEW 8.5 软件之前,用户需要对个人计算机的软硬件配置作一定的了解。安装 LabVIEW 8.5 最低配置如下。

CPU:安装 Pentium III/Celeron 866MHz 及以上处理器。

内存:最小内存为 256MB,最佳配置为 512MB 及以上。

显示器支持的分辨率:1024×768 像素。

硬盘空间:1.2 GB

操作系统:安装在 Windows 2000/XP/Vista、Mac OS 和 Linux 等不同的操作系统上。

LabVIEW 的安装十分简单,只要运行安装光盘中的 Setup 程序,按照屏幕提示,一步一步选择必要的安装选项即可。

在启动 LabVIEW 8.5 的安装程序前,退出其他所有正在运行的程序,这是由于一些程序如杀毒软件等,会使安装时间变长甚至失败。

双击 LabVIEW 8.5 的安装程序后,屏幕上将会出现安装初始化界面,如图 2-1 所示。

初始化 LabVIEW 8.5 的安装程序运行完毕后,系统会提示用户输入产品序列号,如图 2-2 所示。安装 LabVIEW 8.5 的试用版不需要输入序列号,其使用期限为 30 天,在程序编译完成后,不能打包生成独立可执行应用程序(EXE)和安装程序(Installer)。在使用正版软件输入正确序列号后,单击下一步,进入如图 2-3 所示的安装目录界面。用户可以根据需要选择安装目录,建议按照默认目录安装。

在目录选择完成后,单击下一步,进入如图 2-4 所示的模块安装界面。

当安装成功后,将会出现如图 2-5 所示的界面,此时单击完成按钮,即完成程序的安装,就可以使用 LabVIEW 软件了。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,关于这些驱动的安装方法将在第 8 章予以介绍。



图 2-1 初始化 LabVIEW 8.5 的安装程序

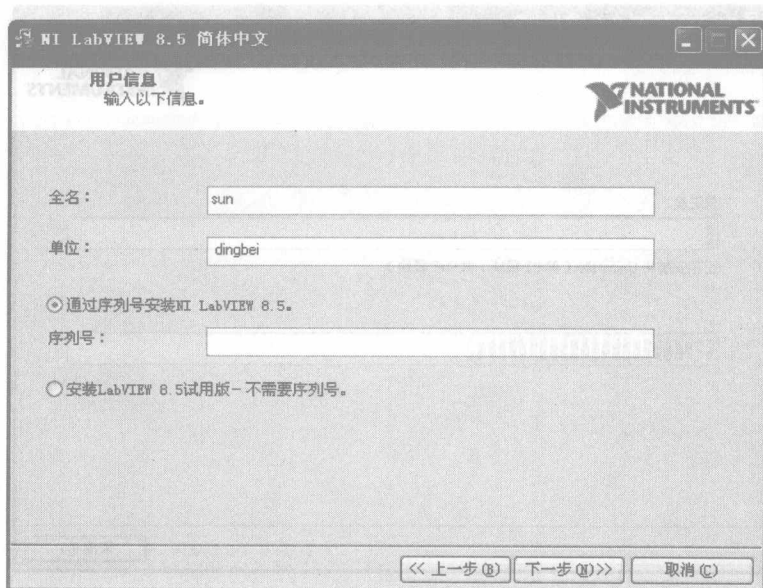


图 2-2 输入 LabVIEW 8.5 的产品序列号和用户信息的界面