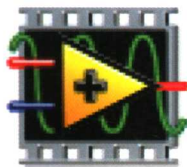




精通

LabVIEW 8.X

王磊 陶梅 编著



LabVIEWTM



- 从入门的角度讲解LabVIEW 的基本应用技术
- 每个章节都配有丰富、翔实的例程
- 帮助读者深入了解并精通LabVIEW

TP312/2854D

2008

精通 LabVIEW 8.X

王磊 陶梅 编著

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书是畅销书升级版,以 LabVIEW 为对象,通过理论与实例相结合的方式,深入浅出地介绍其使用方法和技巧,目的在于让读者快速掌握这门功能强大的图形化编程语言。全书从 LabVIEW 的背景介绍、安装启动讲起,详细介绍了 LabVIEW 编程的基本操作,创建、编辑、修改前面板的方法,创建和调试后面板代码的技巧,数组、簇、字符串,以及波形数据等特殊数据类型的使用方法,数据的表达与显示方法、文件的输入、输出方法,LabVIEW 与其他应用程序的接口等主要内容。

本书在编写过程中,紧密结合作者本人与其他开发者的心得体会,编制了大量实例,具有内容紧凑、语言通俗、实用性强的特点。本书可供高等院校学生使用,也可供从事测试技术的工程技术人员自学和参考。

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

精通 LabVIEW 8.X / 王磊, 陶梅编著. —北京: 电子工业出版社, 2008.5
ISBN 978-7-121-06225-4

I. 精… II. ①王… ②陶… III. 软件工具, LabVIEW 8.X—程序设计 IV. TP311.56

中国版本图书馆 CIP 数据核字(2008)第 036268 号

责任编辑: 李 冰

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

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

开 本: 787×1092 1/16 印张: 24 字数: 468 千字

印 次: 2008 年 5 月第 1 次印刷

印 数: 5000 册 定价: 45.00 元(含光盘 1 张)

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。

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。

服务热线: (010) 88258888。

前 言

LabVIEW 是美国国家仪器公司 (National Instrument, 简称 NI 公司) 推出的一门图形化编程语言, 同时也是著名的虚拟仪器开发平台。作为一门图形化编程语言 (G 语言), LabVIEW 秉承了其简单易用的一贯作风, 使用户能够快速编写出功能强大的应用程序; 作为 NI 公司主推的虚拟仪器开发平台, LabVIEW 担当了“软件即仪器”这一虚拟仪器关键理念中的主角。

LabVIEW 是 Laboratory Virtual Instrument Engineering Workbench 的英文缩写, 这是一个工业标准的图形化开发环境, 它结合了图形化编程方式的高性能与灵活性, 以及专为测试测量与自动化控制应用设计的高端性能与配置功能, 能为数据采集、仪器控制、测量分析与数据显示等各种应用提供必要的开发工具。

LabVIEW 8.0 的发布大大缩短了软件易用性和强大功能之间的差距, 为工程师提供了效率与性能俱佳的真正出色的开发环境。这不但适用于各种测量和自动化领域, 而且无论工程师是否有丰富的开发经验, 都能顺利应用。进行测量和从测量结果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。

本书从入门的角度讲解了 LabVIEW 8.0 的基本应用技术。全书循序渐进, 通过理论与实例相结合的方式, 深入浅出地介绍 LabVIEW 的使用方法和技巧。

本书导读

全书共分 12 章, 由浅入深地介绍了 LabVIEW 8.0 的使用方法, 在每个章节中都配有实例, 目的在于让读者结合实例更加快捷地掌握 LabVIEW 的编程方法。

第 1 章介绍 LabVIEW 这种图形化编程语言的特点, LabVIEW 与“虚拟仪器技术”的关系, 以及 LabVIEW 中的基本概念。

第 2 章主要介绍 LabVIEW 8.0 的安装、启动及其编程环境, 并且指导读者用 LabVIEW 编写第一个应用程序。

第 3 章讲解创建、编辑, 以及修改前面板对象的方法和技巧。

第 4 章主要介绍用 LabVIEW 编写后面板结构框图程序的方法, 以及调试 LabVIEW 程序的技巧。

第 5 章介绍数组、簇、字符串, 以及波形数据这几种 LabVIEW 中的特殊数据类型的使用。

第 6 章以对比结构化和面向对象技术进行程序设计的理念, 介绍 LabVIEW 中控制程

序运行流程的方法。

第 7 章讲解 LabVIEW 中布尔型、数值型、字符型等类型数据的表达方法，以及数据的图形显示。

第 8 章介绍 LabVIEW 中文件的输入和输出。

第 9 章介绍用 LabVIEW 进行信号发生、分析和处理的方法。

第 10 章介绍 LabVIEW 与其他应用程序之间的接口。

第 11 章和第 12 章通过较为复杂的实例分析来进一步介绍用 LabVIEW 编写虚拟仪器程序的方法和技巧。

本书特色

- 深入浅出、图文并茂，以大量的实例作为讲解 LabVIEW 用法的辅助手段，力求让读者快速上手，掌握语言。
- 语言严谨、逻辑清晰，讲解透彻，让读者准确理解书中讲解的内容，更加全面地把握全书的脉络。

本书主要面向 LabVIEW 的初、中级用户，介绍 LabVIEW 编程的基础知识和基本技巧，适合作为其入门和提高的教材和参考书。

由于时间仓促，在写作方式和内容上难免存在缺点和不足，请读者批评指正。

致谢

作者是 2003 年在清华大学精密仪器与机械学系邓焱老师实验室进行毕业设计的时候开始接触并学习虚拟仪器技术和 LabVIEW 的，是邓老师将作者带入虚拟仪器技术这一美妙绝伦的技术领域，使作者能窥其一斑，至今都受益匪浅。在此，作为学生，作者对邓老师致以最为诚挚的谢意。

作者在近 5 年使用 LabVIEW 进行实际工程应用的过程中，美国国家仪器公司(National Instrument)的工程师们，如董东、熊翔等曾经给予作者极其专业和充满热忱的技术支持；北京中科泛华测控技术有限公司的工作人员也在作者遇到困难时施以援手。在此，作者表示衷心地感谢。

在书稿的写作和出版过程中，电子工业出版社博文视点的李冰编辑给予作者大量帮助，其专业的工作素养给作者留下了非常深刻的印象。在此，作者表示由衷的感谢。

作 者
2008 年 5 月

目 录

第 1 篇 入门篇

第 1 章 LabVIEW 概述	2	2.3.1 编辑工具——工具模板	32
1.1 LabVIEW 与 G 语言	2	2.3.2 前面板设计工具——控件模板	33
1.2 LabVIEW 与虚拟仪器技术	4	2.3.3 框图程序设计工具——函数模板	35
1.2.1 虚拟仪器的概念及其相对于传统仪器的优势	4	2.4 LabVIEW 8.0 的新功能与新特性	38
1.2.2 虚拟仪器的特点	6	2.5 用 LabVIEW 编写第一个应用程序	41
1.2.3 LabVIEW 是著名的虚拟仪器开发平台	8	2.6 使用 LabVIEW 8.0 的帮助系统	46
1.3 LabVIEW 中的基本概念	8	2.6.1 使用上下文实时帮助	46
1.3.1 VI 的概念	9	2.6.2 使用目录和索引查找在线帮助	48
1.3.2 图标、连线和框图	10	2.6.3 查找 LabVIEW 例程	49
1.3.3 子 VI 与子程序	12	2.6.4 使用网络资源	50
1.3.4 LabVIEW 8.0 的新特色	12	2.7 本章小结	51
1.4 本章小结	13	第 3 章 创建程序的前面板	52
第 2 章 认识 LabVIEW 8.0	14	3.1 编辑前面板对象	52
2.1 安装和启动 LabVIEW 8.0	14	3.1.1 编辑数值型控件	53
2.1.1 安装 LabVIEW 8.0	14	3.1.2 编辑布尔型控件	56
2.1.2 启动 LabVIEW 8.0	19	3.1.3 编辑图形型控件	59
2.2 LabVIEW 8.0 的菜单简介	23	3.2 设置前面板对象的属性	63
2.2.1 文件菜单	24	3.2.1 设置数值型控件的属性	63
2.2.2 编辑菜单	25	3.2.2 设置文本型控件的属性	66
2.2.3 视图菜单	26	3.2.3 设置布尔型控件的属性	72
2.2.4 工程菜单	27	3.2.4 设置图形显示控件的属性	73
2.2.5 操作菜单	28	3.3 前面板的修饰	77
2.2.6 工具菜单	29	3.3.1 设置前面板对象的颜色以及文字的风格	77
2.2.7 窗口菜单	30	3.3.2 设置多个对象的位置关系和大小	78
2.2.8 帮助菜单	31	3.3.3 用修饰控件装饰	79
2.3 LabVIEW 8.0 的操作模板	32		

3.4	实例——创建数字示波器·····	83
3.5	本章小结·····	86
第4章	编写和调试程序代码·····	87
4.1	LabVIEW 8.0 中的基本数据类型 及其相互转换·····	87
4.1.1	LabVIEW 8.0 中的基本 数据类型·····	87
4.1.2	数据类型间的转换·····	90
4.2	用函数、VIs 和 Express VIs 进行 程序设计·····	94
4.2.1	构建 LabVIEW 程序的基本 元素·····	94
4.2.2	创建 LabVIEW 应用程序·····	95
4.3	使用 Express VIs 进行程序设计·····	99
4.4	使用子程序·····	103
4.5	菜单设计·····	108
4.6	局部变量和全局变量·····	112
4.6.1	局部变量·····	113
4.6.2	全局变量·····	116
4.7	LabVIEW 8.0 的容错机制和调试 技巧·····	120
4.7.1	设置断点调试·····	122
4.7.2	设置探针·····	124
4.7.3	单步执行和循环运行·····	125
4.7.4	高亮显示程序的运行·····	127
4.8	实例——编写并调试“数字滤 波器”程序·····	128
4.9	本章小结·····	130
第5章	LabVIEW 中的特殊数据类型 及其使用·····	131
5.1	数组数据·····	131
5.1.1	创建数组数据·····	131
5.1.2	处理数组数据·····	135
5.2	簇数据·····	141
5.2.1	创建簇数据·····	142
5.2.2	处理簇数据·····	143
5.3	字符串数据·····	150
5.3.1	创建字符串数据·····	150
5.3.2	处理字符串数据·····	152
5.4	波形数据·····	156
5.4.1	创建波形数据·····	157
5.4.2	处理波形数据·····	163
5.5	本章小结·····	169
第6章	结构化与面向对象的程序设计·····	170
6.1	循环结构·····	171
6.1.1	For 循环·····	171
6.1.2	While 循环·····	177
6.2	分支结构·····	179
6.3	顺序结构·····	183
6.4	公式节点·····	185
6.5	反馈节点·····	188
6.6	时序循环·····	190
6.7	事件结构与面向对象的程序设计·····	193
6.8	本章小结·····	196
第2篇 精通篇		
第7章	数据的表达与显示·····	198
7.1	布尔型数据的表达与显示·····	198
7.2	数值型数据的表达与显示·····	201
7.3	字符串型数据的表达与显示·····	205
7.4	用二维图形显示和表达数据·····	208
7.4.1	使用 Waveform Graph 显示 和表达数据·····	209
7.4.2	使用 Waveform Chart 显示 和表达数据·····	214
7.4.3	使用 Express XY Graph 显示和表达数据·····	217
7.5	用三维图形显示和表达数据·····	220

第 7 章 数据的表达与显示	198
7.1 布尔型数据的表达与显示	198
7.2 数值型数据的表达与显示	201
7.3 字符串型数据的表达与显示	205
7.4 用二维图形显示和表达数据	208
7.4.1 使用 Waveform Graph 显示和表达数据	209
7.4.2 使用 Waveform Chart 显示和表达数据	214
7.4.3 使用 Express XY Graph 显示和表达数据	217
7.5 用三维图形显示和表达数据	220

7.5.1 使用 Intensity Chart 显示 和表达数据	220	8.8 使用基本文件输入、输出函数 进行文件操作	256
7.5.2 使用 Intensity Graph 显示 和表达数据	221	8.8.1 利用基本文件输入、输出 函数存储电子表格文件	257
7.5.3 使用 3D Surface Graph 显示和表达数据	223	8.8.2 利用基本文件输入、输出 函数存储文本文件	258
7.5.4 使用 3D Curve Graph 显示和表达数据	227	8.8.3 利用基本文件输入、输出 函数存储二进制文件	259
7.6 用其他方式显示和表达数据	229	第 9 章 利用 LabVIEW 进行信号的发生、 分析和处理	260
7.7 本章小结	231	9.1 信号的发生	260
第 8 章 文件的输入与输出	232	9.1.1 基本函数发生器	262
8.1 LabVIEW 支持的基本文件类型 及其操作函数简介	232	9.1.2 基本多频信号发生器	264
8.2 电子表格文件的输入、输出	237	9.1.3 带有幅值信息的基本 多频信号发生器	265
8.2.1 电子表格文件的输入	237	9.1.4 多频波形信号发生器	266
8.2.2 电子表格文件的输出	239	9.1.5 白噪声信号发生器	267
8.3 文本文件的输入、输出	240	9.1.6 高斯白噪声信号发生器	268
8.3.1 文本文件的输入	240	9.1.7 周期随机噪声信号发生器	268
8.3.2 文本文件的输出	242	9.1.8 Simulate Signal Express VI	269
8.4 二进制文件的输入、输出	243	9.2 信号的时域分析	271
8.4.1 二进制文件的输入	244	9.2.1 基本平均值与均方差 VI	271
8.4.2 二进制文件的输出	245	9.2.2 平均值与均方差 VI	272
8.5 数据记录文件的输入、输出	246	9.2.3 周期平均值与均方差值 VI	273
8.5.1 数据记录文件的输入	247	9.2.4 过渡态测量 VI	275
8.5.2 数据记录文件的输出	248	9.2.5 脉冲测量 VI	275
8.6 波形文件的输入、输出	250	9.2.6 幅值以及极大值、极 小值 VI	276
8.6.1 波形文件的输入	250	9.2.7 提取信号单频率信息 VI	277
8.6.2 波形文件的输出	251	9.2.8 频率测量 Express VI	278
8.7 LabVIEW Measurement 文件的 输入、输出	252	9.2.9 时域和过渡态测量 Express VI	279
8.7.1 LabVIEW Measurement 文件的输入	252	9.2.10 幅值和极大值、极小值 测量 Express VI	280
8.7.2 LabVIEW Measurement 文件的输出	254		

9.3 信号的频域分析	281	第 11 章 实例分析——用 LabVIEW 构建简单的虚拟仪器系统	313
9.3.1 快速傅里叶变换功率谱 VI	282	11.1 用 LabVIEW 8.0 构建多功能信号发生器	313
9.3.2 快速傅里叶变换功率谱密度 VI	283	11.2 用 LabVIEW 8.0 构建带有频谱分析功能的高性能示波器	318
9.3.3 快速傅里叶变换幅值—相位谱 VI	283	11.3 用 LabVIEW 8.0 构建简单的信号分析与处理系统	320
9.3.4 快速傅里叶变换实部—虚部谱 VI	284	11.4 用 LabVIEW 8.0 构建复杂的信号发生系统	323
9.3.5 频率响应函数的幅值—相位谱 VI	285	11.5 用 LabVIEW 8.0 构建复杂的信号分析与处理系统	330
9.3.6 频率响应函数的实部—虚部谱 VI	286	11.6 用 LabVIEW 8.0 构建信号选择系统	338
9.3.7 互谱（幅值—相位谱）VI	287	11.7 本章小结	343
9.3.8 互谱（实部—虚部谱）VI	288	第 12 章 实例分析——用 LabVIEW 构建数据采集系统	344
9.4 信号的滤波处理	289	12.1 用 LabVIEW 构建数据采集系统的前期准备	344
9.4.1 巴特沃斯滤波器	290	12.2 LabVIEW 8.0 提供的用于数据采集的 VIs	345
9.4.2 切比雪夫滤波器	291	12.3 构建单通道数据采集系统	347
9.4.3 反-切比雪夫滤波器	292	12.4 将数据采集系统由单通道扩展为多通道	350
9.4.4 贝赛尔滤波器	293	12.5 本章小结	353
9.4.5 Filter Express VI	294	附录 A 习题	354
9.5 本章小结	295	附录 B 习题答案	361
第 10 章 LabVIEW 与其他应用程序的接口	296		
10.1 调用 C 语言代码	296		
10.2 调用 Windows API	303		
10.3 调用 ActiveX 控件	308		
10.4 本章小结	312		

第1篇

入门篇

第 1 章

LabVIEW 概述

本章主要介绍 LabVIEW 这种图形化编程语言的基本特点, 以及用 LabVIEW 编写程序时所涉及的基本概念。

本章主要包括:

- LabVIEW 与 G 语言
- LabVIEW 与虚拟仪器技术
- LabVIEW 中的基本概念

1.1 LabVIEW 与 G 语言

LabVIEW (Laboratory Virtual Instrument Engineering), 即实验室虚拟仪器集成环境, 是一种图形化的编程语言——G 语言。和 Visual Basic、Visual C++、Delphi、Perl 等基于文本型程序代码的编程语言不同, LabVIEW 采用图形模式的结构框图构建程序代码, 因而, 在使用这种语言编程时, 基本上不写程序代码, 取而代之的是用图标、连线构成的流程图。它尽可能地利用了技术人员、科学家、工程师所熟悉的术语、图标和概念, 因此, LabVIEW 是一个面向最终用户的工具。它可以增强用户构建自己的科学和工程系统的能力, 提供了实现仪器编程和数据采集系统的便捷途径。使用它进行原理研究、设计、测试并实现仪器系统时, 可以大大提高工作效率。

LabVIEW 是一个工业标准的图形化开发环境, 它结合了图形化编程方式的高性能与灵活性以及专为测试、测量与自动化控制应用设计的高端性能与配置功能, 能为数据采集、仪器控制、测量分析与数据显示等各种应用提供必要的开发工具, 因此, LabVIEW 通过降低应用系统开发时间与项目筹建成本帮助科学家与工程师们提高工作效率。

LabVIEW 被广泛应用于各种行业中,包括汽车、半导体、航空航天、交通运输、高校实验室、电信、生物医药与电子等。无论在哪个行业,工程师与科学家们都可以使用 LabVIEW 创建功能强大的测试、测量与自动化控制系统,在产品开发中进行快速原型创建与仿真工作。在产品生产过程中,工程师们也可以利用 LabVIEW 进行生产测试,监控各个产品的生产过程。总之,LabVIEW 可用于各行各业产品开发的阶段。

LabVIEW 的功能非常强大,它是带有可扩展函数库和子程序库的通用程序设计系统,不仅可以用于一般的 Windows 桌面应用程序设计,而且还提供了用于 GPIB 设备控制、VXI 总线控制、串行口设备控制,以及数据分析、显示和存储等应用程序模块,其强大的专用函数库使得它非常适合编写用于测试、测量以及工业控制的应用程序。LabVIEW 可方便地调用 Windows 动态链接库和用户自定义的动态链接库中的函数,还提供了 CIN (Code Interface Node) 节点使得用户可以使用由 C 或 C++ 语言,如 ANSI C 等编译的程序模块,使得 LabVIEW 成为一个开放的开发平台。LabVIEW 还直接支持动态数据交换 (DDE)、结构化查询语言 (SQL)、TCP 和 UDP 网络协议等。此外,LabVIEW 还提供了专门用于程序开发的工具箱,使得用户能够很方便地设置断点,动态地执行程序来非常直观形象地观察数据的传输过程,而且可以方便地进行调试。

当我们困惑于基于文本模式的编程语言,陷入函数、数组、指针、表达式乃至对象、封装、继承等枯燥的概念和代码中时,我们迫切需要一种代码直观、层次清晰、简单易用却不失功能强大的语言,G 语言就是这样一种语言,而 LabVIEW 则是 G 语言的杰出代表。LabVIEW 基于 G 语言的基本特征——用图标和框图产生块状程序,这对于熟悉仪器结构和硬件电路的硬件工程师、现场工程技术人员及测试技术人员来说,编程就像是设计电路图一样;因此,硬件工程师、现场工程技术人员及测试技术人员们学习 LabVIEW 可以驾轻就熟,在很短的时间内就能够学会并应用 LabVIEW。

从运行机制上看,LabVIEW——这种语言的运行机制就宏观上讲已经不再是传统的冯·诺伊曼计算机体系结构的执行方式了。传统的计算机语言(如 C 语言)中的顺序执行结构在 LabVIEW 中被并行机制所代替;从本质上讲,它是一种带有图形控制流结构的数据流模式 (Data Flow Mode),这种方式确保了程序中的函数节点 (Function Node),只有在获得它的全部数据后才能够被执行。也就是说,在这种数据流程序的概念中,程序的执行是数据驱动的,它不受操作系统、计算机等因素的影响。

LabVIEW 的程序是数据流驱动的。数据流程序设计规定,一个目标只有当它的所有输入有效时才能执行;而目标的输出,只有当它的功能完全时才是有效的。这样,LabVIEW 中被连接的方框图之间的数据流控制着程序的执行次序,而不像文本程序受到行顺序执行的约束。因而,我们可以通过相互连接功能方框图快速简捷地开发应用程序,甚至还可以有多个数据通道同步运行。

1.2 LabVIEW 与虚拟仪器技术

提到 LabVIEW 就不得不提“虚拟仪器技术”，可以说 LabVIEW 是依托“虚拟仪器技术”的发展需要而诞生的，是“虚拟仪器技术”的精髓理念——“软件就是仪器”中的软件部分，或者说是“虚拟仪器技术”实现的核心环节。正因为 LabVIEW 与“虚拟仪器技术”有着千丝万缕的联系，要想了解乃至用好 LabVIEW 就需要先了解“虚拟仪器技术”。通过这一节的学习，在了解 LabVIEW 与“虚拟仪器技术”关系的同时，读者将会体会到 LabVIEW 的强大功能和学习 LabVIEW 这种图形化编程语言的重要意义。

1.2.1 虚拟仪器的概念及其相对于传统仪器的优势

传统仪器大多由以下三大功能模块组成：对被测信号的采集与控制模块、分析与处理模块，以及测量结果的表达与输出模块。传统仪器的这些功能都是以硬件（或固化的软件）的形式存在的。将这些功能移植到计算机上完成，在计算机上插上数据采集卡，然后利用软件在屏幕上生成仪器面板，并且用软件来进行信号的分析处理，这就构成了一台虚拟仪器。

虚拟仪器是一种全新的仪器概念，它是利用计算机的硬件资源（CPU、存储器、显示器、键盘、鼠标）、标准数字电路（GPIB、RS-232 接口总线、新型的 VXI 接口总线、信号调理和转换电路、图像采集电路、现场总线等）以及计算机软件资源（数据分析与表达、过程通信、图像用户界面等），经过有针对性的开发测试，使之成为一套相当于使用者自己专门设计的传统仪器。

简单地说，虚拟仪器技术就是用户自定义的基于 PC 技术的测试和测量解决方案。相对于传统仪器，它有四大优势：性能高、扩展性强、开发时间少、完美的集成功能。

- 性能高

虚拟仪器是在 PC 技术的基础上发展起来的，所以完全“继承”了以现成的 PC 技术为主导的最新商业技术的优点，包括功能卓越的处理器和文件 I/O，使用户在数据导入磁盘的同时就能实时地进行复杂的分析。随着数据传输到硬盘功能的不断加强，以及与 PC 总线的结合，高速数据记录已经较少依赖大容量的本地内存。以一台 60G 的示波器为例，在采用虚拟仪器技术的情况下，构建这样一台示波器是相当简单的，只要将一台基于 PC 的数字转换器放置在 PC 机中，就能以高达每秒 100MB 的速度将数据导入磁盘。

虚拟仪器技术的另一突出优势就是不断提高的网络带宽。因特网和越来越快的计算机网络的数据分享进入了一个全新的阶段，将因特网和 NI（National Instrument，美国国家仪器公司）的软硬件产品相结合，用户就能够轻松地与地球另一端的同事共享测量结果，分享“天涯若比邻”的便捷。

图 1.1 很好地说明了虚拟仪器的性能, 随着计算机存储器、图形处理器、I/O 单元, 以及 CPU 速度与性能的提高, 虚拟仪器的性能会随之提高, 而这都有赖于虚拟仪器“软件就是仪器”的概念, 换言之, 这种性能的提高有赖于软件, 而 LabVIEW 就是这种“虚拟仪器软件”最好的开发平台和开发环境之一。

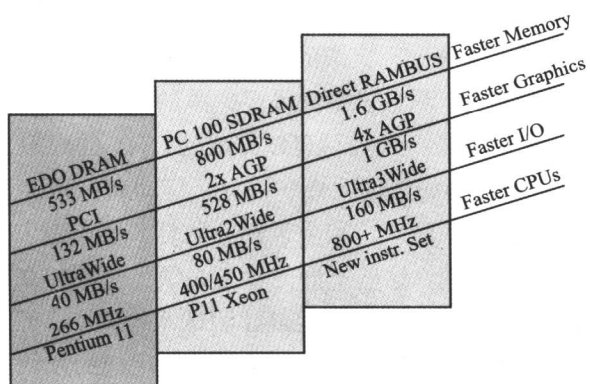


图 1.1 虚拟仪器的性能有赖于计算机的性能

• 扩展性强

虚拟仪器的软、硬件工具使得工程师和科学家们不再囿于当前的技术中。得益于虚拟仪器软件的灵活性, 只需更新用户的计算机或测量硬件, 就能以最少的硬件投资和极少的、甚至无须软件的升级即可改进用户的整个系统。在利用最新科技的时候, 用户可以把它们集成到现有的测量设备中来, 最终以较少的成本加速产品上市的时间。

图 1.2 诠释了虚拟仪器技术扩展性强的特点。应用虚拟仪器开发平台 (如 LabVIEW) 开发出的虚拟仪器, 可以将商业技术应用于运动控制、工业自动化、测试与测量等领域, 并快速将其转化为产品, 而这一转化过程的关键是用 LabVIEW 开发出来的软件。

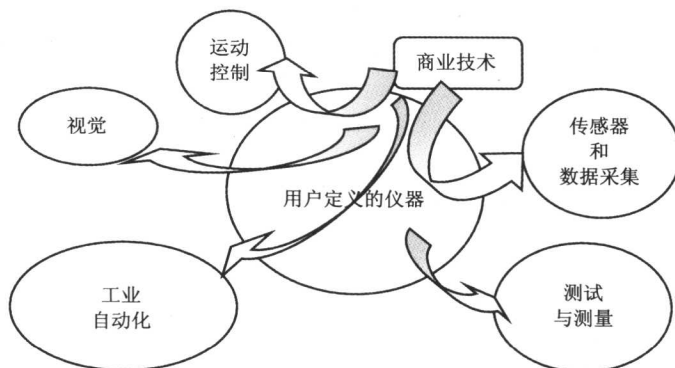


图 1.2 虚拟仪器具有很强的扩展性

- 开发时间少

在驱动和应用两个层面上，虚拟仪器技术高效的软件构架能与计算机、仪器仪表和通信方面的最新技术结合在一起。NI 设计这一软件构架的初衷就是为了方便用户的操作，同时还提供了灵活性和强大的功能，使用户轻松地配置、创建、部署、维护和修改高性能、低成本的测量和控制解决方案。

- 完美的集成

虚拟仪器技术从本质上说是一个集成的软硬件概念。随着产品在功能上不断地趋于复杂，工程师们通常需要集成多个测量设备来满足完整的测试需求，但是这些不同设备间的连接和集成总是耗费大量时间，不是轻易可以完成的。NI 的虚拟仪器软件平台为所有的 I/O 设备提供了标准的接口，例如数据采集、视觉、运动和分布式 I/O 等，帮助用户轻松地将多个测量设备集成到单个系统中，减少了任务的复杂性。为了获得最高的性能、简单的开发过程和系统层面上的协调，这些不同的设备必须保持其独立性，同时还要紧密地集成在一起。NI 的结构可以使开发者们快速地创建测试系统，并随着要求的改变轻松地完成对系统的修改。得益于这一集成式的构架带来的好处，用户的系统可以更具竞争性，因为用户可以更高效地设计和测试高质量的产品，并将它们更快速地投入市场。

虚拟仪器与传统仪器特点的对比如表 1.1 所示。

表 1.1 虚拟仪器与传统仪器特点的对比

虚拟仪器	传统仪器
开放、灵活，可与计算机技术保持同步发展	封闭、仪器间相互配合较差
关键是软件，系统性能升级方便，通过网络下载升级程序即可	关键是硬件，升级成本较高，且升级必须上门服务
价格低廉，仪器间的资源可重复利用	价格昂贵，仪器间一般无法相互利用
用户可定义仪器功能	只有厂家能定义仪器功能
可以与网络及周边设备方便连接	功能单一，只能连接有限的独立设备
开发与维护费用降至最低	开发与维护开销高
技术更新周期短（1~2 年）	技术更新周期长（5~10 年）

1.2.2 虚拟仪器的特点

虚拟仪器是计算机技术和仪器测量技术相结合的产物，它充分利用了计算机强大的运算处理功能，突破了传统仪器在数据处理、显示、传输、存储等方面的限制。目前，基于 PC 的 A/D 及 D/A 转换，开关量输入、输出，定时计数的硬件模块，在技术指标及可靠性等方面已相当成熟，而且价格低廉。常用传感器及相应的调理模块也趋向模块化、标准化，因而减少了硬件的重复开发，这使得使用者可以方便地对其进行硬件维护、功能扩展和软件升级。

虚拟仪器的特点如下:

- 具有可变性、多层性、自助性的面板

虚拟仪器的面板可以做到与传统仪器一样,可以有显示器显示波形,有 LED 指示数字,有指针式的表头指示刻度,有旋钮、滑动条、开关按钮,有报警指示灯和声音等。而虚拟仪器的优越之处在于传统仪器面板上的元器件是硬件,是由厂商设计确定的,不可改变地安装在专用的面板上。而虚拟仪器的面板由计算机的显示器构成,面板上的各种显示控制元件是软件图库中的各种功能图形,由用户设计面板,调用图形块,用户可以不受“标准件”和“加工工艺”限制,随意增、删、移动元器件,变化尺寸、色彩等。还可以制作多层下拉面板、帮助文件等,做出远远超过传统仪器的全汉化、生动美观、界面友好的面板。

- 强大的信号处理能力

用适当的硬件接口电路,对信号进行采集、放大、滤波、隔离、A/D 转换后,虚拟仪器就可以灵活、充分地利用通用计算机的大量实用软件工具,对信号进行各种计算、分析、判断、处理、图形或数字显示,经过 D/A 转换后控制执行器件的动作。

- 虚拟仪器的功能、性能、指标可由用户定义

一方面可以根据用户的不同要求对同一仪器的功能、性能、指标进行修改或增删,彻底打破了传统仪器一经设计、制造完成后,其功能、性能、指标不可改变的封闭性、单一性。另一方面也可以将多种仪器的功能、性能、指标等以软件的形式集成在一个“功能软件库”——虚拟仪器库内,通过它们的不同组合以及和各种不同类型的硬件接口搭配,使得在一台个人计算机就可实现各种仪器的不同功能,大大提高了仪器功能的灵活性,甚至可以进行非常复杂的测试工作。

- 具有标准的、功能强大的接口总线、板卡及相应软件

GPIB 通用接口总线 (General Purpose Interface Bus) 又称 IEEE488 国际标准接口总线,三十年来广泛应用于仪器领域。但是只适用于消息级器件的互操作,不适用于寄存器级器件。VXI 总线 1987 年被首次推出,迅速成为 IEEE1155 国际标准。VXI 硬件的通用性,使任意厂家、各种类型仪器接口不会发生电气和机械方面的冲突。VXI 总线的开放性,保证任何系统一旦建立,将来仍能得到很好的应用。VXI 能保持每个仪器之间精确定时和同步,具有 40 M Bytes/s 的高数据传输率。VXI 模块化仪器被认为是虚拟仪器最理想的硬件平台,是仪器硬件的发展方向。此外,还有 VISA、PCI 等标准 I/O 卡及其相应的驱动程序库,这些都为虚拟仪器的数据采集和控制提供了强大支持。

- 虚拟仪器具有开发周期短、成本低、维护方便、易于应用的特点

这些特点决定了应用“虚拟仪器技术”可以快速、低成本地开发出具有相当大柔性的仪器,并且这种仪器便于维护和升级。

1.2.3 LabVIEW 是著名的虚拟仪器开发平台

上面两节分别介绍了虚拟仪器的概念、虚拟仪器相对于传统仪器的优势，以及虚拟仪器的特点。从虚拟仪器的概念出发，不难发现，软件——虚拟仪器的开发平台是虚拟仪器的精髓，而 LabVIEW 正是一款优秀的虚拟仪器软件开发平台。

作为美国国家仪器公司（National Instrument，简称 NI 公司）推出的虚拟仪器开发平台，LabVIEW 以其直观、简便的编程方式，众多的源码级的设备驱动程序，多种多样的对分析和表达功能的支持，为用户快捷地构建自己在实际生产中所需要的仪器系统创造了基础条件。

由于采用了图形化编程语言——G 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践中去。因此，硬件工程师、现场工程技术人员及测试技术人员们学习 LabVIEW 驾轻就熟，在很短的时间内就能够学会并应用 LabVIEW，也不必去记忆那眼花缭乱的文本式程序代码。

LabVIEW 程序又称为虚拟仪器，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和设备的参数和功能的场合，以及对信号进行分析、研究、传输等场合。

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，能够将烦琐复杂的语言编程简化成为以菜单提示方式选择功能，并且用线条将各种功能连接起来，十分省时简便，深受用户青睐。与传统的编程语言比较，LabVIEW 图形编程方式能够节省 85% 以上的程序开发时间，其运行速度却几乎不受影响，体现出了极高的效率。使用虚拟仪器产品，用户可以根据实际生产需要重新构筑新的仪器系统。例如，用户可以将原有的带有 RS232 接口的仪器、VXI 总线仪器，以及 GPIB 仪器通过计算机连接在一起，组成各种各样新的仪器系统，由计算机进行统一管理和操作。

可以预见，由于 LabVIEW 这些其他语言无法比拟的优势，已经成为该领域的一朵奇葩，最终将引发传统的仪器产业一场新的革命。

1.3 LabVIEW 中的基本概念

为了让读者更快地熟悉 LabVIEW 的编程模式和编程方法，了解其中的基本概念是很有必要的，这一节主要结合 LabVIEW 8.0 的编程环境介绍 LabVIEW 中的基本概念。