

一样的软件，不一样的学习方法

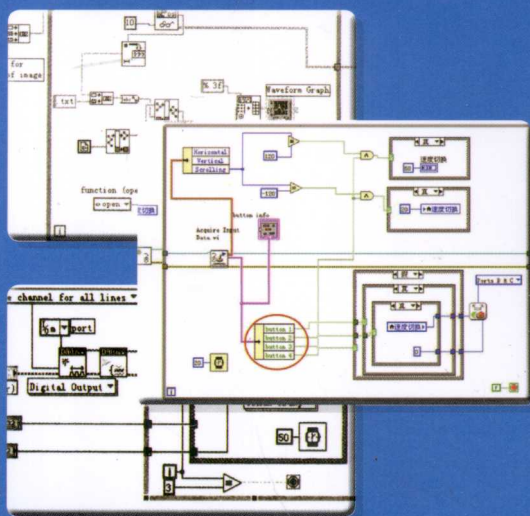
精通

LabVIEW

虚拟仪器程序设计 与案例实现

章佳荣 王璨 赵国宇 编著

- 全面介绍了 LabVIEW 编程的一些基本操作，如数据类型及相互的转换、程序结构、数据表达与显示、文件 I/O 等。
- 详细介绍了 LabVIEW 在工程领域的典型应用示例及调试技巧，如信号处理的应用、与外部程序的接口、多线程技术、数据库应用、数据采集与仪器控制的应用、在网络、串口通信中的应用等。
- 解惑答疑。针对学习过程中容易遇到的问题，将零星点滴的经验、技巧、难点一一列出并加以分析，最大限度地贴近和满足读者的需要。



PPT + 源程序 + 视频讲解

人民邮电出版社
POSTS & TELECOM PRESS

013031771

TP311.56
1138

精通

LabVIEW

虚拟仪器程序设计 与案例实现

■ 章佳荣 王璨 赵国宇 编著



北航

C1636574

人民邮电出版社
北京

TP311.56
1138

177180810

图书在版编目 (CIP) 数据

精通LabVIEW虚拟仪器程序设计与案例实现 / 章佳荣, 王璨, 赵国宇编著. — 北京: 人民邮电出版社, 2013. 4

ISBN 978-7-115-29769-3

I. ①精… II. ①章… ②王… ③赵… 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(2012)第250230号

内 容 提 要

全书分4篇共18章,详细地介绍了LabVIEW的基本概念、基本操作以及在工程领域的应用。

第一篇“入门篇”,介绍了LabVIEW的一些相关概念、特点及安装步骤,并用一个简单的“Hello World”应用让读者对LabVIEW编程有一个整体的了解。第二篇“基础篇”,详细介绍了LabVIEW编程的一些基本操作。如数据类型及相互的转换、LabVIEW的程序结构、LabVIEW的数据表达与显示、文件I/O,通过这些技术的介绍和学习,读者就能熟练掌握LabVIEW的基本操作。第三篇“应用篇”,详细介绍了LabVIEW在工程领域的一些应用、相关实例、调试技巧以及应用程序发布等。如LabVIEW在数学分析中的应用、LabVIEW在信号处理中的应用、LabVIEW与外部程序的接口、LabVIEW的多线程技术、LabVIEW的数据库应用、LabVIEW在数据采集与仪器控制中的应用、LabVIEW在网络和串口等通信中的应用,以及LabVIEW的界面设计与美化 and 代码优化方法等。第四篇疑难问题解答,帮助读者尽快融入实战角色。

精通 LabVIEW 虚拟仪器程序设计与案例实现

- ◆ 编 著 章佳荣 王 璨 赵国宇
责任编辑 张 涛
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 25.75
字数: 661千字 2013年4月第1版
印数: 1-3500册 2013年4月北京第1次印刷

ISBN 978-7-115-29769-3

定价: 59.00元(附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

LabVIEW 是一款图形化的编程工具——G 语言，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，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。

目前，市面上关于 LabVIEW 的书籍很多，但大多数只是简单地介绍与 LabVIEW 开发环境相关的一些基础知识，缺乏比较详细的工程应用实例，对于有一定 LabVIEW 开发经验的人来说，就显示有点过于简单了。而本书，进行了由简入难地详细讲解，并且在书中加入了许多在工程应用当中积累起来的宝贵经验及实用实例。对于初学者，可以借鉴这些经验少走弯路；对于有一定编程基础的熟手，可以利用书中的一些例子，缩短软件的开发周期。

全书分 4 篇共 18 章，由简入难地详细介绍了 LabVIEW 的基本概念、基本操作以及在工程领域的应用。

第一篇“入门篇”（1~2 章），简单介绍了 LabVIEW 的一些相关概念、特点及安装步骤，并用一个简单的“Hello World”让读者对 LabVIEW 编程有一个整体了解。

第二篇“基础篇”（3~6 章），详细介绍了 LabVIEW 编程的一些基本操作。第 3 章介绍 LabVIEW 的数据类型及相互转换，第 4 章介绍 LabVIEW 的程序结构，第 5 章介绍 LabVIEW 的数据表达与显示，第 6 章介绍 LabVIEW 的文件 I/O。经过这几章的介绍，读者就能熟练掌握 LabVIEW 的基本操作了。

第三篇“应用篇”（7~17 章），详细介绍了 LabVIEW 在工程应用领域的一些应用、相关实例、调试技巧以及应用程序发布等。第 7 章介绍 LabVIEW 在数学分析中的应用，第 8 章介绍 LabVIEW 在信号处理中的应用，第 9 章介绍 LabVIEW 与外部程序的接口，第 10 章介绍 LabVIEW 的多线程技术，第 11 章介绍 LabVIEW 在数据采集与仪器控制中的应用，第 12 章介绍 LabVIEW 在网络、串口等通信中的应用，第 13 章介绍 LabVIEW 的界面设计与美化，第 14 章介绍代码优化方法，第 15 章介绍 LabVIEW 的项目管理与报表生成，第 16 章介绍 LabVIEW 应用程序发布方法，第 17 章介绍工程应用实例。

第四篇疑难问题解答（18 章），主要包括一些疑难问题的解答，帮助读者解决实战中的一些问题。

目 录

入 门 篇

第 1 章 LabVIEW 概述..... 2	2.4.3 函数选板..... 20
1.1 LabVIEW 与 G 语言..... 2	2.5 LabVIEW 菜单栏及工具栏.....20
1.2 LabVIEW 与虚拟仪器..... 3	2.5.1 LabVIEW 菜单栏..... 20
1.3 LabVIEW 的发展历程..... 5	2.5.2 LabVIEW 工具栏..... 22
1.4 LabVIEW2011 新特性..... 5	2.6 VI 的创建与编辑.....22
1.5 思考与练习..... 7	2.6.1 VI 的创建..... 22
第 2 章 开始 LabVIEW 编程..... 8	2.6.2 VI 的编辑..... 23
2.1 LabVIEW 安装..... 8	2.7 VI 的运行与调试.....25
2.2 启动 LabVIEW..... 12	2.8 LabVIEW 常用快捷键.....25
2.3 基本概念..... 14	2.9 LabVIEW 帮助.....26
2.3.1 VI..... 14	2.9.1 即时上下文帮助..... 26
2.3.2 前面板..... 14	2.9.2 查找范例..... 26
2.3.3 程序框图..... 16	2.9.3 在线帮助..... 27
2.4 LabVIEW 操作面板..... 17	2.10 综合实例: 创建一个
2.4.1 工具选板..... 17	“Hello World” 程序.....27
2.4.2 控件选板..... 18	2.11 思考与练习.....29

基 础 篇

第 3 章 LabVIEW 中的数据类型..... 32	3.2.4 布尔值/数字值转换..... 44
3.1 基本数据类型..... 32	3.2.5 数字与时间标识的转换..... 45
3.1.1 数字型..... 34	3.3 数据运算与操作.....45
3.1.2 布尔型..... 35	3.3.1 基本数学运算与操作..... 45
3.1.3 字符串型与路径..... 36	3.3.2 字符串运算与操作..... 47
3.1.4 枚举型..... 37	3.3.3 布尔运算与操作..... 48
3.1.5 簇..... 38	3.3.4 比较运算..... 49
3.1.6 数组..... 38	3.3.5 数组与矩阵操作..... 51
3.1.7 波形数据..... 39	3.3.6 簇操作..... 55
3.1.8 时间标识..... 39	3.3.7 波形数据操作..... 57
3.1.9 变体..... 39	3.4 综合实例: 模拟温度监测.....57
3.2 数据类型之间的转换..... 40	3.5 思考与练习.....58
3.2.1 不同数字类型之间的转换..... 40	第 4 章 LabVIEW 中的程序结构..... 59
3.2.2 数字/字符串转换..... 41	4.1 循环结构.....59
3.2.3 字符串/数组/路径转换..... 43	

4.1.1 For 循环	60	5.5 用三维图形表达与显示	
4.1.2 While 循环	63	数据	109
4.2 顺序结构	64	5.5.1 三维曲面图	110
4.2.1 平铺式顺序结构	64	5.5.2 三维参数图	111
4.2.2 层叠式顺序结构	66	5.5.3 三维曲线图	112
4.3 定时结构	67	5.6 其他图形显示控件	112
4.3.1 定时循环	67	5.6.1 极坐标图	113
4.3.2 定时顺序	69	5.6.2 最小—最大曲线显示	
4.3.3 定时 VI	69	控件	113
4.4 条件结构	70	5.7 综合实例: 绘制李萨如图形	114
4.5 事件结构	72	5.8 思考与练习	115
4.6 公式节点	76	第 6 章 LabVIEW 中的文件 I/O	116
4.7 反馈节点	78	6.1 文件 I/O 基本概念介绍	116
4.8 变量	78	6.2 常用文件	117
4.8.1 局部变量	78	6.2.1 二进制文件 (.dat)	117
4.8.2 全局变量	80	6.2.2 文本文件 (.txt)	120
4.8.3 共享变量	82	6.2.3 电子表格文件 (.xls)	123
4.9 禁用结构	83	6.3 LabVIEW 的特殊文件	125
4.9.1 程序框图禁用结构	83	6.3.1 波形文件 (Waveform	
4.9.2 条件禁用结构	83	Files)	125
4.10 综合实例	84	6.3.2 XML 文件	126
4.10.1 综合实例 1: 数组求和	84	6.3.3 数据存储文件 (TDM)	127
4.10.2 综合实例 2: 动态窗口		6.3.4 高速数据流文件	
管理	84	(TDMS)	128
4.11 思考与练习	87	6.3.5 测量文件 (LVM)	130
第 5 章 LabVIEW 中的数据表达与		6.4 其他文件	131
显示	88	6.4.1 音频文件 (.wav)	131
5.1 数值型数据的表达与显示	88	6.4.2 压缩文件 (.zip)	132
5.1.1 数值输入与显示控件	89	6.4.3 配置文件 (.ini)	133
5.1.2 滑动杆控件	91	6.5 文件工具	133
5.1.3 进度条控件	92	6.5.1 路径、目录操作	133
5.1.4 旋钮类控件	92	6.5.2 获取文件、目录的信息	134
5.1.5 其他个性控件	92	6.5.3 文件位置与大小设置	135
5.2 布尔型数据的表达与显示	93	6.5.4 文件操作	136
5.3 字符型数据的表达与显示	95	6.6 综合实例	137
5.4 用二维图形表达与显示数据	97	6.6.1 综合实例 1: 文件循环	
5.4.1 波形图表与波形图	98	写入	137
5.4.2 XY 图和 ExpressXY 图	106	6.6.2 综合实例 2: 文件循环	
5.4.3 强度图表与强度图	107	保存	137
5.4.4 数字波形图和混合波		6.6.3 综合实例 3: 读取 Excel	
形图	107	文件	138
		6.7 思考与练习	139

应用篇

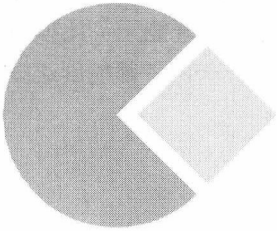
第 7 章 数学分析	142	8.5.3 触发与门限.....	185
7.1 初等与特殊函数.....	142	8.6 逐点分析.....	188
7.2 线性代数.....	143	8.7 综合实例：声音回波特性 分析.....	189
7.3 微积分.....	145	8.8 思考与练习.....	190
7.4 概率与统计.....	147	第 9 章 外部程序接口	191
7.5 最优化.....	148	9.1 DLL 与 API 调用.....	191
7.6 微分方程.....	149	9.1.1 DLL 调用.....	191
7.7 几何.....	150	9.1.2 参数类型的配置.....	193
7.8 多项式.....	151	9.1.3 Windows API 调用.....	194
7.9 曲线拟合.....	152	9.2 可执行文件的调用.....	197
7.10 插值.....	153	9.3 ActiveX 调用.....	198
7.11 脚本与公式.....	155	9.3.1 ActiveX 自动化.....	198
7.12 综合实例：水箱问题.....	156	9.3.2 ActiveX 容器.....	200
7.13 思考与练习.....	157	9.3.3 ActiveX 事件.....	201
第 8 章 信号处理	158	9.3.4 LabVIEW 作为服务器端.....	203
8.1 信号发生.....	158	9.4 LabVIEW 与 MATLAB 混合 编程.....	205
8.1.1 基本函数发生器.....	159	9.5 综合实例：用 LabVIEW 与 MATLAB 混合编程实现小波 去噪.....	205
8.1.2 多频信号发生器.....	161	9.6 思考与练习.....	207
8.1.3 噪声信号发生器.....	162	第 10 章 多线程技术	208
8.1.4 用公式节点产生信号.....	163	10.1 多线程基本概念.....	208
8.1.5 用 Express VI 产生信号.....	163	10.2 多线程带来的问题.....	210
8.2 时域分析.....	164	10.3 LabVIEW 中的多线程.....	211
8.2.1 基本平均值与均方差 测量.....	165	10.3.1 LabVIEW 的执行系统.....	211
8.2.2 过渡态测量.....	166	10.3.2 运行队列.....	213
8.2.3 提取信号单频信息.....	169	10.3.3 LabVIEW 多线程中的 DLL.....	214
8.2.4 相关.....	170	10.3.4 定制线程配置.....	215
8.2.5 谐波失真分析.....	171	10.3.5 LabVIEW 对多核 CPU 的 支持.....	218
8.3 频域分析.....	172	10.4 生产者/消费者模式.....	219
8.3.1 傅里叶变换.....	173	10.4.1 生产者/消费者的优势.....	219
8.3.2 拉普拉斯变换.....	174	10.4.2 生产者/消费者基本组成 结构.....	219
8.3.3 功率谱分析.....	175	10.4.3 多消费者循环.....	223
8.4 信号调理.....	176	10.4.4 基于队列状态机的.....	
8.4.1 滤波器.....	176		
8.4.2 窗函数.....	178		
8.4.3 波形调理.....	179		
8.5 波形监测.....	183		
8.5.1 边界检测.....	183		
8.5.2 波峰波谷检测.....	184		

生产者/消费者结构	224	12.6 思考与练习	280
10.5 综合实例: 多线程计时器	226	第 13 章 界面设计与美化	281
10.6 思考与练习	226	13.1 界面设计的一般原则与 常用界面风格	281
第 11 章 数据采集与仪器控制	227	13.1.1 界面设计的一般原则	281
11.1 数据采集	227	13.1.2 常用界面风格	282
11.1.1 数据采集系统基本组成	227	13.2 菜单设计	284
11.1.2 NI 数据采集硬件产品 及其应用领域	229	13.3 属性节点与调用节点	285
11.1.3 硬件选型重要参数	230	13.4 子 VI 调用与重载	287
11.1.4 配置管理软件 MAX	231	13.4.1 子 VI 的创建与调用	287
11.1.5 NI-DAQ 应用举例	235	13.4.2 多面板程序设计	289
11.2 仪器控制	239	13.4.3 动态载入界面	290
11.2.1 常用总线介绍	239	13.5 界面美化	293
11.2.2 仪器驱动程序	242	13.5.1 使用布局工具排列对象	293
11.2.3 LabVIEW 仪器控制	242	13.5.2 添加背景图片	293
11.2.4 LabVIEW 与第三方 硬件的连接	249	13.5.3 自定义控件	294
11.3 综合实例: “NI PCI-MIO- 16E-1” 采集卡使用	249	13.5.4 动画	296
11.4 思考与练习	253	13.5.5 利用控件选板与工具 选板	297
第 12 章 通信	254	13.5.6 VI 属性设置	299
12.1 串行通信	254	13.6 综合实例: 用属性节点 控制 Tab 控件	300
12.1.1 串口介绍	254	13.7 思考与练习	302
12.1.2 串口接线定义与连接 方式	255	第 14 章 代码优化	303
12.1.3 LabVIEW 中的串口 编程	258	14.1 代码优化的必要性	303
12.2 网络通信	262	14.2 VI 性能分析工具	303
12.2.1 TCP 协议通信	262	14.2.1 性能和内存分析工具	304
12.2.2 UDP 协议通信	267	14.2.2 显示缓冲区分配工具	306
12.3 DataSocket 通信	269	14.2.3 VI 统计工具	307
12.3.1 DataSocket 技术	269	14.2.4 查找可并行的循环工具	308
12.3.2 DataSocket 逻辑构成	270	14.3 VI 内存使用	309
12.3.3 DataSocket 编程	271	14.3.1 虚拟内存	309
12.4 远程面板	273	14.3.2 VI 组件内存管理	310
12.4.1 配置 LabVIEW Web Server	274	14.3.3 数据流编程和数据 缓冲区	311
12.4.2 在 LabVIEW 环境中 操作 Remote Panels	276	14.3.4 条件输入控件和数据 缓冲区	312
12.5 综合实例: 基于 TCP 协议的 DSP 调试软件	277	14.3.5 前面板的内存问题	312
		14.4 高效使用内存的规则	312
		14.5 常用代码优化方法与技巧	313
		14.5.1 及时释放内存	313

14.5.2	输出重复使用输入缓冲区	314	第 16 章	应用程序发布	337
14.5.3	使用一致的数据类型	314	16.1	LabVIEW 程序生成规范	337
14.5.4	避免频繁调整数组大小	315	16.2	发布应用程序前的准备	338
14.5.5	开发高效的数据结构	317	16.3	创建源代码发布	339
14.5.6	使用元素同址操作结构	319	16.4	创建独立应用程序 (EXE)	344
14.5.7	防止内存泄漏	321	16.5	创建安装程序 (SETUP)	351
14.5.8	动态调用子 VI	321	16.6	创建共享库 (DLL)	357
14.6	思考与练习	322	16.7	思考与练习	360
第 15 章	项目管理与报表生成	323	第 17 章	工程应用实例介绍	362
15.1	项目管理	323	17.1	基于声卡的信号采集与分析软件	362
15.1.1	项目浏览器	323	17.2	DSO25216 驱动开发	368
15.1.2	源代码管理工具	324	17.3	管道漏点检测与定位	374
15.1.3	LLB 管理器	325	17.4	多通道数据采集软件	382
15.2	报表生成	326	17.5	基于串口通信的上位机控制软件	385
15.2.1	报表生成 VI 介绍	326			
15.2.2	简易报表生成	328			
15.2.3	高级报表生成	330			
15.2.4	报表生成工具包	331			
15.3	综合实例: 报表生成	335			
15.4	思考与练习	336			

疑难问题解答

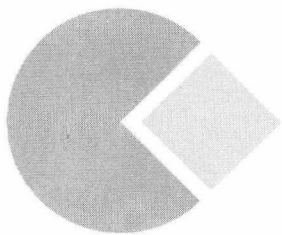
第 18 章	LabVIEW 应用技巧集萃及常见疑难解答	390	问题	397	
18.1	LabVIEW 应用技巧集萃	390	18.2.3	界面与显示方面的问题	398
18.2	常见疑难问题及解答	396	18.2.4	程序运行与应用程序发布方面的问题	400
18.2.1	数据读取与文件操作方面的问题	396	18.2.5	其他问题	401
18.2.2	仪器控制与驱动方面的问题				



入门篇

第 1 章 LabVIEW 概述

第 2 章 开始 LabVIEW 编程



第1章 LabVIEW 概述

LabVIEW 是一款图形化的编程语言，工程师和科学家可使用 NI LabVIEW 图形化编程环境，借由直观的图标和连线，开发复杂且类似流程图的测量、测试和控制系统。它既能无可比拟地集成数千款硬件设备，也能通过上百个内置库实现高级分析和数据的可视化——两者都能创建虚拟仪器。

【本章内容提要】

- ◇ LabVIEW 与 G 语言
- ◇ LabVIEW 与虚拟仪器
- ◇ LabVIEW 的发展历程
- ◇ LabVIEW 2011 的新特性

1.1 LabVIEW 与 G 语言

1. 什么是 G 语言？

语言是一种符号系统（文字和语音），是用来表达我们的思想及进行相互交流和沟通的一种手段或者说是一种工具。当计算机出现后，人类又创造出能够与计算机进行沟通的语言，被称为：程序语言。程序语言是用来指挥“冷冰冰的计算机”表达我们的想法（意念）并能够被计算机执行的程序代码。

计算机语言的演变和发展，完全是为了更好地指挥计算机为人类工作。计算机程序语言历经了下面几代的发展演变。

- ◇ 机器语言（Machine language）
- ◇ 汇编语言（Assembly language）
- ◇ 高级语言
- ◇ 图形化编程语言（G 语言）

图形化的程序语言，又被称为“G”语言。使用这种语言编程时，基本上不写程序代码，取而代之的是结构框图或流程图。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。它可以增强你构建自己的科学和工程系统的能力，提供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。

2. 什么是 LabVIEW？

LabVIEW（Laboratory Virtual Instrument Engineering Workbench），即实验室虚拟仪器集

成环境，是一种典型的图形化编程语言（G 语言），是一个工业标准的图形化开发环境。它结合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，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了必要的开发工具。

LabVIEW 被广泛应用于各种行业，包括汽车、半导体、航空航天、交通运输、电信、生物医药等。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。

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、串行设备控制，以及数据分析、显示和存储等的应用程序模块。其强大的专用函数库使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 Window 动态链接库和用户自定义的动态链接库中的函数，这使 LabVIEW 成为一个开放的开发平台。LabVIEW 还支持动态数据交换（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，使用户能够很方便地设置断点，动态执行程序以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。

3. LabVIEW 的运行机理

从运行机制上看，LabVIEW 这款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的方框图快速简洁地开发应用程序，甚至还可以有多个数据通道同步运行。

12 LabVIEW 与虚拟仪器

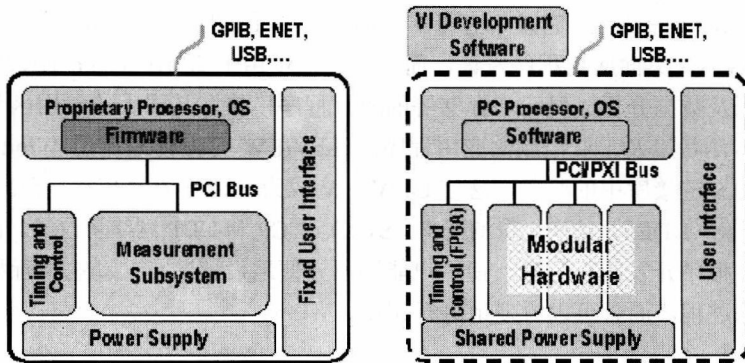
1. 什么是虚拟仪器？

虚拟仪器（virtual instrument）是基于计算机的仪器。计算机和仪器的密切结合是目前仪器发展的一个重要方向。粗略地说，这种结合有两种方式，一种是将计算机装入仪器，其典型例子就是所谓智能化的仪器。随着计算机功能的日益强大以及其体积的日趋缩小，这类仪器功能也越来越强大，目前已经出现了含嵌入式系统的仪器。另一种方式是将仪器装入计算机。以通用的计算机硬件及操作系统为依托，实现各种仪器功能。虚拟仪器主要是指后一种方式。

虚拟仪器实际上是一个按照仪器需求组织的数据采集系统。在虚拟仪器的研究中涉及的基础理论主要有计算机数据采集和数字信号处理。在这一领域内，目前使用较为广泛的计算机语言是美国 NI 公司的 LabVIEW。

2. 虚拟仪器与传统仪器有什么不同？

虚拟仪器由用户定义，而传统仪器则功能固定且由厂商定义。如图 1-1 所示，传统仪器（左图）与虚拟仪器（右图）有许多相同的组件结构，但在体系结构原理上完全不同。



▲图 1-1 传统仪器（左图）与虚拟仪器（右图）的区别

每一台虚拟仪器系统都由两部分组成：软件和硬件。不使用厂商定义的、预封装好的软件和硬件，工程师和科学家获得了最大的用户定义的灵活性。传统仪器把所有软件和测量电路封装在一起，利用仪器前面板为用户提供一组有限的功能。而虚拟仪器系统提供的则是完成测量或控制任务所需的所有软件和硬件设备，功能完全由用户自定义。此外，利用虚拟仪器技术，工程师和科学家们还可以使用高效且功能强大的软件来自定义采集、分析、存储、共享和显示功能。

3. LabVIEW 与虚拟仪器

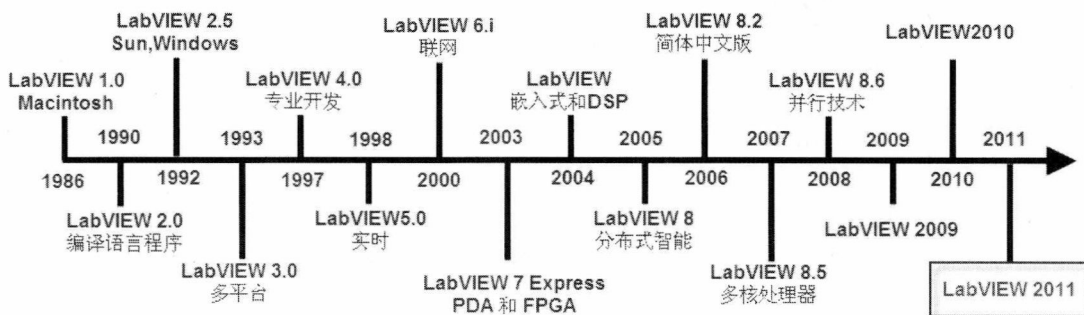
虚拟仪器的起源可以追溯到 20 世纪 70 年代，那时计算机测控系统在国防、航天等领域已经有了相当发展。PC 出现以后，仪器级的计算机化成为可能，甚至在 Microsoft 公司的 Windows 诞生之前，NI 公司已经在 Macintosh 计算机上推出了 LabVIEW2.0 以前的版本。对虚拟仪器和 LabVIEW 长期、系统、有效的研究开发使得该公司成为业界公认的权威。目前 LabVIEW 的最新版本为 LabVIEW2011。使用 LabVIEW 软件，用户可以借助于它提供的软件环境。该环境由于其数据流编程特性、LabVIEW Real-Time 工具对嵌入式平台开发的多核支持，以及自上而下的为多核而设计的软件层次，是进行并行编程的首选。

LabVIEW 作为一种图形化的编程语言，广泛地被工业界、学术界和研究实验室所接受，被视为一个标准的数据采集和仪器控制软件。LabVIEW 集成了与满足 GPIB、VXI、RS-232 和 RS-485 协议的硬件及数据采集卡通信的全部功能。它还内置了便于应用 TCP/IP、ActiveX 等软件标准的库函数。这是一款功能强大且灵活的软件。利用它可以方便地建立自己的虚拟仪器，其图形化的界面使得编程及使用过程都十分生动有趣。

利用 LabVIEW，可产生独立运行的可执行文件，它是一个真正的 32 位编译器。像许多重要的软件一样，LabVIEW 提供了 Windows、UNIX、Linux、Macintosh 的多种版本。

1.3 LabVIEW 的发展历程

LabVIEW 是实验室虚拟仪器开发环境 (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。25 年来 LabVIEW 的发展历程和各期的版本如图 1-2 所示。



▲图 1-2 LabVIEW 的发展历程

LabVIEW 的概念雏形来源于特鲁查德和柯德斯凯两人在 20 世纪 70 年代在 ARL (Applied Research Laboratory, 应用研究实验室) 完成的一个大型测试系统。1986 年 5 月, LabVIEW Beta 版面世, 经过几个月的修改, 1986 年 10 月, LabVIEW 1.0 正式发布。最初 LabVIEW 吸引的仅仅是没有任何编程经验的用户。这些用户相信采用 LabVIEW 就能实现有经验的程序员也难以完成的任务。

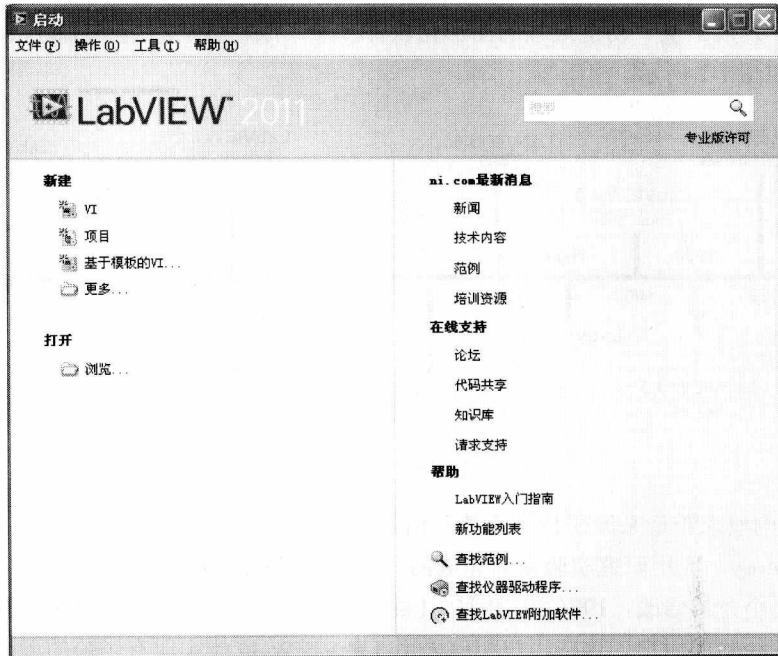
有效的内存管理是使用图形化编程语言优于普通解释语言的关键。由于数据流解释需要大量内存分配, 因此, 内存重用对数据流图编程效率至关重要, 寻找内存重用的有效算法成为提高 LabVIEW 性能的关键。LabVIEW 1.1 解决了算法问题, 随后改进的 LabVIEW 1.2 是可靠性和鲁棒性很强的产品, 但由于内在体系局限, 其性能与 C 语言程序相比, 仍然有较大的差距。为了解决这个问题, 1988 年开始的 LabVIEW 2.0 采用了最新的面向对象编程 (OOP) 技术。当 1990 年 1 月 LabVIEW 2.0 被发送给第一个热心用户使用, LabVIEW 程序在执行速度和灵活性方面的改进令人惊叹。

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 程序设计成为可能后, LabVIEW 才实现了从 Macintosh 到 Windows 平台的移植。1992 年, 跨平台的 LabVIEW 2.5 问世。1993 年的 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。

1.4 LabVIEW 2011 新特性

LabVIEW 2011 是 LabVIEW 系列发布 25 周年时推出的领先系统设计软件, 其界面如图 1-3 所示。LabVIEW 2011 通过新的工程实例库及其对大量硬件设备和部署目标的交互支持极大地增进

了效率。这其中包含新的多核 NI CompactRIO 控制器及当今业界性能最强大的射频向量信号分析仪之一的 NIPXIe-5665。LabVIEW2011 还支持内置在最新的 Microsoft.NET 框架的组件，并且基于用户的反馈新增了多项新特性。综合以上优势，LabVIEW2011 能够帮助工程师将零散的系统部件集成为一个统一并可重配置的平台，从而让工程应用更高效、更出色、成本也更低。



▲图 1-3 LabVIEW2011

“25 年前，我们创建了 LabVIEW 来帮助工程师从繁杂的编程及系统集成中解脱出来，将精力专注于应用和创新。而今天，LabVIEW 已经变成了针对测试测量和控制领先的图形化系统设计软件。每一次软件版本的更新，不论是无缝集成最新的硬件，还是添加新的函数库和 API，以及新增基于用户反馈的新特性，我们的主要目的依然是在所有工程环境中提高应用效率。”——JeffKodosky (LabVIEW 的发明者兼公司创始人之一)。

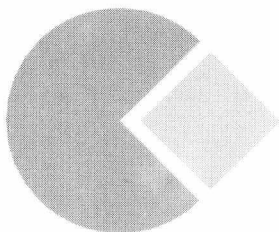
LabVIEW2011 使工程师在多种任务下极大地提高了效率。它主要包括以下高效功能。

- ✧ 快速的程序界面视觉化效果，其时尚的用户界面包括新的控件及显示控件银色面板。
- ✧ 可重用的代码，包括对最新的 .NET 程序组件、.m 结构体的支持，以及包含新的 XilinxIP 库的 LabVIEW FPGA 模块。
- ✧ 最快可达原先 5 倍的 FPGA 代码载入、连线、编辑及编译。
- ✧ 程序的可编程建立及部署于目标的可执行文件。
- ✧ 新的通信 API 可大量生成异步线程以创建更快的多线程应用。
- ✧ LabVIEW2011 以其良好的关键任务应用稳定性和对众多工业界领先硬件的快捷集成能力，赋予了测量和控制系统设计者们强大的信心，使他们可以在已有的设施条件下高效地创新。

结合硬件模块后的 LabVIEW2011 是实现 NI 图形化系统设计方法的核心，为设计、原型和发布提供了统一的高效的平台。从基本测量应用到最复杂先进的研究项目，工程师和科学家在几乎每个工业领域都在使用图形化系统设计。

15 思考与练习

- (1) 计算机语言的发展经历了哪几个阶段?
- (2) 什么叫 G 语言? 有什么特点与优势?
- (3) LabVIEW 的发展经历了哪几个阶段?
- (4) 什么叫虚拟仪器。
- (5) LabVIEW2011 有哪些新特性?



第2章 开始 LabVIEW 编程

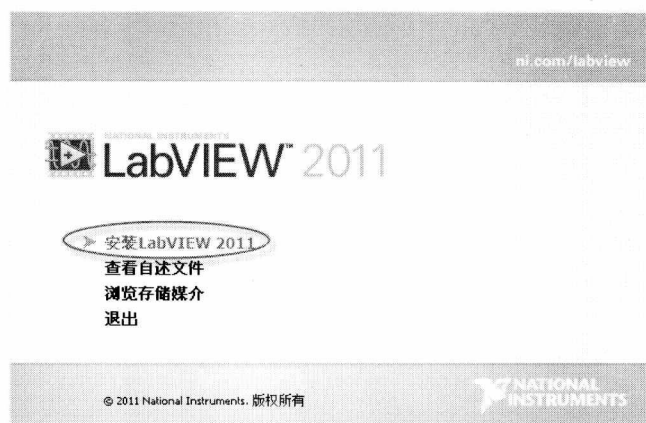
LabVIEW 作为图形化的编程语言，有非常友好的用户界面，其安装和使用都非常方便。本章主要介绍 LabVIEW 的安装、开发环境、基本概念等。

【本章内容提要】

- ◇ LabVIEW 的安装
- ◇ LabVIEW 的基本概念介绍
- ◇ LabVIEW 的操作面板与工具栏设置
- ◇ VI 的创建与编辑
- ◇ VI 的运行与调试
- ◇ LabVIEW 常用快捷键介绍
- ◇ 如何获取 LabVIEW 帮助文档

2.1 LabVIEW 安装

LabVIEW 的安装简单明了，启动安装程序后，单击图 2-1 所示界面中的“安装 LabVIEW2011”，进入图 2-2 所示的初始化界面。



▲图 2-1 LabVIEW2011 安装启动界面

安装程序的初始化大约需要 1 分钟左右的时间。初始化完成后进入欢迎界面，如图 2-3 所示。单击“下一步”，进入“用户信息”设置界面，如图 2-4 所示，在这里用户可以根据自己需要，设置相关信息。