

一样的软件，不一样的学习方法

LabVIEW

虚拟仪器程序设计 从入门到精通

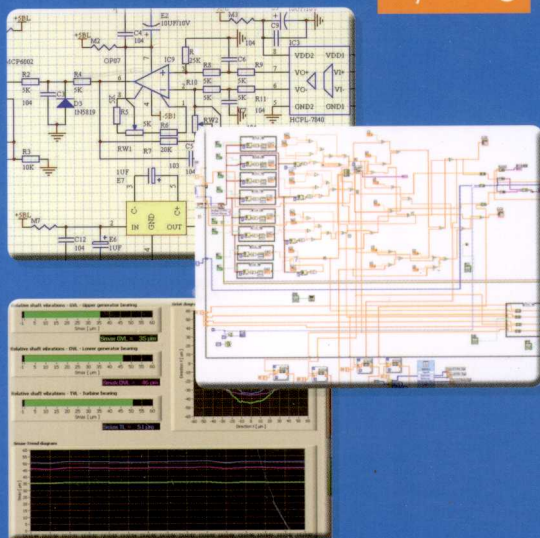
林静 林振宇 郑福仁 编著

第一版

■ 一线资深 LabVIEW CLD 认证工程师浓缩多年实践经验倾力打造。

易学易用。“知识点 + 针对每个知识点的小实例 + 综合实例”的写作技巧，可以帮助读者快速地学习掌握 LabVIEW，并解决工程实践中的问题。

■ 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



DVD-ROM

人民邮电出版社
POSTS & TELECOM PRESS

013031732

TP311.56
1026-2

LabVIEW

虚拟仪器程序设计

从入门到精通

林静 林振宇 郑福仁 编著

第一版



北航

C1636613

人民邮电出版社
北京

P

TP311.56

1026-2

957180810

图书在版编目 (CIP) 数据

LabVIEW虚拟仪器程序设计从入门到精通 / 林静, 林振宇, 郑福仁编著. -- 2版. -- 北京: 人民邮电出版社, 2013.4

ISBN 978-7-115-29724-2

I. ①L… II. ①林… ②林… ③郑… 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244082号

内 容 提 要

本书共 5 篇, 25 章, 全面、细致地讲述了 LabVIEW 的新特性及操作方法、关键细节技巧和工程应用实践经验。

第一篇 LabVIEW 技术基础 (第 1 章~第 12 章), 主要讲述了 LabVIEW 编程环境、LabVIEW 前面板设计、LabVIEW 程序流程和结构、波形显示、程序动态控制、文件输入与输出、生成安装包和可执行文件、程序调试与程序性能等; 第二篇 LabVIEW 与其他应用程序的链接 (第 13 章~第 16 章), 主要讲解 ActiveX 技术应用、LabVIEW 与编程语言的链接、C 代码调用、动态链接库函数调用、LabVIEW 与数据库的链接等; 第三篇 LabVIEW 应用实战技术 (第 17 章~第 20 章), 主要讲解了测试系统与虚拟仪器、LabVIEW 中的数据采集、总线技术、网络通信等; 第四篇工程应用综合案例 (第 21 章~第 23 章), 通过 3 个完整的案例: 风机自动控制系统、风速采集、听力计控制软件开发, 讲解了如何用 LabVIEW 进行项目实战的知识和技能, 使读者可以学以致用地掌握 LabVIEW 的实践应用; 第五篇常见疑难解答与经验技巧集萃 (第 24 章~第 25 章), 将常见问题和解决技巧以疑难解答的方式集中在本篇讲述, 帮助读者掌握学习捷径, 切实提高工作效率。

本书是在第一版的基础上升级的, 本书是一本全面、系统讲述 LabVIEW 基础操作、应用与程序开发的教学用书或参考书, 通过本书的学习, 读者能够快速、深入地学习和掌握该软件的强大功能, 并积累较为丰富的应用实践经验。可作为 LabVIEW 的初、中级读者的入门或进阶教程和从事 LabVIEW 的广大工程技术人员的参考书, 也可作为院校本科生、研究生的 LabVIEW 课程教材或自学教程。

LabVIEW 虚拟仪器程序设计从入门到精通 (第二版)

◆ 编 著 林 静 林振宇 郑福仁

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>

三河市海波印务有限公司印刷

◆ 开本: 787×1092 1/16

印张: 26

字数: 798 千字

印数: 4 501—8 000 册

2013 年 4 月第 2 版

2013 年 4 月河北第 1 次印刷

ISBN 978-7-115-29724-2

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

虚拟仪器是当前测控领域的技术热点，它代表了未来仪器技术的发展方向，LabVIEW 是一款优秀的虚拟仪器软件开发平台。目前，LabVIEW 在中国的测试技术及教育领域内得到很大的应用。

对于大多数用户来说，LabVIEW 软件的入门较慢、较难，主要原因是，这方面优秀的、实用性强的参考书很少，有些书籍采用软件版本陈旧，跟不上软件升级的需求。

本书在强调实用性强的基础上追求了新颖性和灵活性，以最新软件版本为实践平台，内容讲述细致、深入，贴近开发测试系统人员需求，并将近年来 LabVIEW 应用于典型行业的工程实例、实践技巧与经验呈献给读者。书中针对知识点列举了大量实例，使读者学习和理解更加容易、透彻。另外将作者多年积累的经验、技巧以疑难解答的方式集中在本书最后部分讲述，这使得不同层次的读者容易学习和掌握软件的应用，帮助读者掌握学习捷径，提高工作效率。

本书内容

书中的实例程序都是作者近年来从事虚拟仪器工作的工程实践成果，书中关于工具包章节的学习，需要读者安装相应的工具包后进行实践，而有些涉及数据采集的程序需要安装相应的硬件才能运转。全书共 5 篇分为 25 章。第一篇为 LabVIEW 技术基础，包括第 1~12 章；第二篇为 LabVIEW 与其他应用程序的链接，包括第 13~16 章；第三篇为 LabVIEW 应用实战技术，包括第 17~20 章；第四篇为工程应用综合案例，包括第 21~23 章；第五篇为常见疑难解答与经验技巧集萃，包括第 24~25 章。

本书特色

本书是一本全面、系统讲述 LabVIEW 基础操作、应用与程序开发的教材，通过本书的学习，读者能够快速、深入学习和掌握该软件的强大功能，并积累较为丰富的应用实践经验。

- 软件版本采用新的 LabVIEW。
- 一线资深 LabVIEW CLD 认证工程师浓缩多年实践经验倾力打造。
- 内容系统、全面，由浅入深、循序渐进。本书全面、细致地讲述了 LabVIEW 软件操作方法、关键细节技巧和工程应用实践经验，适合读者的学习规律和需要，明确学习的出发点和落脚点。
- 知识点+针对每个知识点的小实例+综合实例的讲述方式，可以使读者快速地学习掌握 LabVIEW 软件操作及应用该知识点解决工程实践中的问题。综合实例部分，深入细致剖析工程应用的流程、细节、难点、技巧，可以起到融会贯通的作用。
- 实例来源于作者设计的大型工程项目，最大程度地贴近实际应用需要。
- 常见疑难解答与技巧集萃。对于常见的、典型的、相对零散的问题、技巧也是读者非常渴望得到的知识，本教程以灵活的方式在本书最后部分集中讲述。
- 本书附带光盘，收录实例操作视频文件。

- 本书附带光盘，收录实例操作视频文件。

本书由林静、林振宇、郑福仁主编，参与编写的还有郝旭宁、李建鹏、赵伟茗、刘钦、于志伟、张永岗、周世宾、姚志伟、曹文平、张应迁、张洪才、邱洪钢、张青莲、陆绍强、汪海波。

读者对象

本书可作为 LabVIEW 的初、中级读者的入门或进阶教程，从事 LabVIEW 的广大工程技术人员的参考书，也可作为本科生、研究生的 LabVIEW 课程教材或自学教程。

由于时间仓促，再加水平有限，书中难免会有一些不足之处，希望广大读者批评指正，联系方式 zhangtao@ptpress.com.cn。

编 者

目 录

第一篇 LabVIEW 技术基础

第 1 章 LabVIEW 概述	2	第 3 章 LabVIEW 的编辑环境	22
1.1 LabVIEW 的基本知识	2	3.1 构建完整的 LabVIEW 工程	22
1.1.1 什么是 LabVIEW	2	3.1.1 前面板	22
1.1.2 数据流的概念	3	3.1.2 程序框图	23
1.2 LabVIEW 的工作环境和编程示例	3	3.1.3 LabVIEW 工程	25
1.2.1 LabVIEW 的工作环境	3	3.2 LabVIEW 的菜单设置	33
1.2.2 LabVIEW 自带编程示例	4	3.2.1 执行工具条	33
1.3 小结	5	3.2.2 LabVIEW 面板设置	34
第 2 章 LabVIEW 前面板设计	6	3.2.3 即时帮助	35
2.1 LabVIEW 前面板控件概述	6	3.3 浮动选板	36
2.1.1 LabVIEW 控件类型	6	3.3.1 工具选板	36
2.1.2 LabVIEW 控件选板	6	3.3.2 控件选板	38
2.2 LabVIEW 控件选板详细分类	7	3.3.3 定制选板	40
2.2.1 数值控件	7	3.3.4 菜单快捷键	41
2.2.2 布尔控件	8	3.3.5 帮助	44
2.2.3 字符串与路径控件	8	3.4 小结	45
2.2.4 数组、矩阵、簇控件	8	第 4 章 数据表达	47
2.2.5 列表与表格控件	8	4.1 数值	47
2.2.6 图形控件	8	4.1.1 数值	47
2.2.7 下拉列表与枚举控件	9	4.1.2 数值函数	49
2.2.8 容器控件	9	4.2 布尔量	53
2.2.9 I/O 控件	9	4.2.1 布尔数据	53
2.2.10 引用句柄控件	9	4.2.2 机械动作	54
2.2.11 变体与类控件	10	4.3 数组	54
2.3 控件设置	10	4.3.1 创建数组	55
2.3.1 快捷菜单	10	4.3.2 数组函数	57
2.3.2 属性对话框	10	4.4 簇	60
2.4 前面板对象的操作	15	4.4.1 创建簇	60
2.4.1 焦点	15	4.4.2 簇函数	61
2.4.2 控件的布置	16	4.4.3 错误簇	63
2.5 定制控件	17	4.5 字符串	64
2.6 小结	21	4.5.1 字符串控件	65
		4.5.2 组合框控件	65
		4.5.3 字符串函数	66

4.6	局部变量、全局变量和共享变量	77	6.6	混合信号图	131
4.6.1	局部变量	77	6.6.1	混合信号图的特点	131
4.6.2	全局变量	79	6.6.2	混合信号图的应用	132
4.6.3	共享变量	81	6.7	三维图形表示	134
4.7	小结	86	6.7.1	三维图形	134
第 5 章	程序流程和结构	87	6.7.2	三维图片控件	134
5.1	循环结构	87	6.8	控件	134
5.1.1	For 循环	87	6.9	小结	138
5.1.2	While 循环	88	第 7 章	同步数据传递	139
5.1.3	移位寄存器	89	7.1	通知器操作	139
5.1.4	反馈节点	91	7.1.1	通知器概念	139
5.2	条件结构	92	7.1.2	通知器函数	139
5.2.1	创建条件结构	92	7.2	通知器操作典型实例	143
5.2.2	条件结构的隧道	93	7.3	队列操作	145
5.2.3	条件结构的应用实例	94	7.3.1	队列概念	145
5.3	顺序结构	99	7.3.2	队列函数	145
5.3.1	平铺式顺序结构	99	7.4	队列操作应用及实例	147
5.3.2	层叠式顺序结构	99	7.4.1	队列操作典型实例	147
5.3.3	顺序结构的数据传递	100	7.4.2	队列消息处理器	147
5.4	事件结构	101	7.4.3	生产者/消费者模式	148
5.4.1	事件结构	101	7.4.4	技巧点拨	149
5.4.2	通知事件与过滤事件	105	7.5	信号量操作	150
5.4.3	动态事件与用户事件	105	7.5.1	信号量概念	150
5.4.4	事件结构的状态机模式	107	7.5.2	信号量函数	150
5.5	定时结构	109	7.5.3	信号量操作典型实例	151
5.5.1	定时循环	110	7.5.4	信号量技巧提示	153
5.5.2	定时顺序结构	111	7.6	集合点操作	153
5.5.3	定时源函数	112	7.6.1	集合点函数	153
5.6	公式节点	113	7.6.2	集合点操作典型实例	153
5.7	禁用结构	114	7.7	事件发生	155
5.7.1	程序框图禁用结构	114	7.7.1	事件发生概念	155
5.7.2	条件禁用结构	115	7.7.2	事件发生函数	155
5.8	小结	116	7.7.3	事件发生典型实例	156
第 6 章	波形显示	117	7.8	首次调用	156
6.1	波形图表	117	7.9	小结	157
6.1.1	波形图表的主要特点	117	第 8 章	程序动态控制	158
6.1.2	波形图表的主要设置	117	8.1	VI 服务器技术	158
6.1.3	波形图表的应用	123	8.1.1	VI 服务器简介	158
6.2	波形图	124	8.1.2	VI 服务器函数介绍	158
6.2.1	波形图的主要特点	124	8.2	VI 服务器的应用实例	164
6.2.2	波形图显示	125	8.2.1	动态控制 VI 运行	164
6.3	XY 图	128	8.2.2	前面板对象属性控制	166
6.4	强度图和强度图表	129	8.3	引用	169
6.5	数字波形图	130	8.3.1	属性和方法的引用	169

8.3.2	应用程序引用	170	9.7.2	数据记录	204
8.3.3	VI 引用	170	9.8	小结	207
8.3.4	控件引用	171	第 10 章	XControl 的制作	208
8.3.5	应用实例	171	10.1	XControl 的特点介绍	208
8.3.6	VI 服务器类层次结构	172	10.2	XControl 的制作	208
8.4	应用程序控制其他函数	175	第 11 章	生成安装包和可执行文件	213
8.5	小结	178	11.1	生成可执行文件	213
第 9 章	文件输入/输出	179	11.2	生成安装包	217
9.1	基本文件输入输出	179	11.3	注意事项	220
9.1.1	选择文件格式	179	第 12 章	程序调试与程序性能	222
9.1.2	文件常量	180	12.1	修复程序创建中的错误	222
9.1.3	读写电子表格文件	181	12.1.1	错误列表	222
9.1.4	读写测量文件	182	12.1.2	常见的错误	222
9.2	高级文件输入/输出操作	186	12.2	程序调试工具	223
9.2.1	文件输入/输出的 基本操作	186	12.2.1	工具条上的程序 调试工具	223
9.2.2	文本文件的输入/输出	188	12.2.2	程序调试选项设置	224
9.2.3	二进制文件的输入/ 输出操作	190	12.2.3	工具选板上的调试工具	225
9.3	配置文件的操作	192	12.3	程序常见的调试方法	226
9.4	TDMS 文件操作	195	12.4	LabVIEW 的多线程程序	227
9.5	存储	198	12.4.1	基本概念	228
9.5.1	基本存储函数	198	12.4.2	LabVIEW 的多线程 实现方法	228
9.5.2	高级存储函数	200	12.5	LabVIEW 程序性能	232
9.6	ZIP 函数	200	12.5.1	程序性能监测	233
9.7	高级文件函数	201	12.5.2	优化 LabVIEW 程序	236
9.7.1	高级文件函数的 基本操作	201	12.6	小结	240

第二篇 LabVIEW 与其他应用程序的链接

第 13 章	ActiveX 技术应用	242	14.1.2	报表函数——Word	255
13.1	ActiveX 技术简介	242	14.1.3	报表函数——Excel	258
13.2	在 LabVIEW 中使用 ActiveX 控件	242	14.2	报表函数的应用	264
13.3	ActiveX 自动化	244	14.2.1	报表基本函数的应用	264
13.3.1	ActiveX 自动化函数	244	14.2.2	Word 报表应用	264
13.3.2	ActiveX 自动化函数 的应用	245	14.2.3	Excel 报表应用	265
13.4	LabVIEW 作为 ActiveX 服务器	247	14.3	生成可执行文件和安装包	268
13.5	生成可执行文件和安装程序	248	第 15 章	动态链接库函数调用	269
第 14 章	LabVIEW 与 Microsoft 的链接	249	15.1	动态链接库函数调用实例	269
14.1	报表生成函数介绍	249	15.2	调用库函数节点详细解析	271
14.1.1	报表基本函数介绍	249	15.2.1	调用库函数页面	271
			15.2.2	参数页面	271
			15.2.3	回调页面	272

15.2.4 错误检查页面	272	16.2.1 在 Access 中建立一个数据库	276
15.3 在 LabVIEW 中执行系统命令	272	16.2.2 数据库基本操作	278
15.4 小结	273	16.2.3 SQL 语句的操作	281
第 16 章 LabVIEW 与数据库的链接	274	16.3 生成可执行文件和安装文件	283
16.1 Database Connectivity 工具包介绍	274	16.3.1 生成可执行文件	283
16.2 Database Connectivity 应用典型实例	276	16.3.2 生成安装文件	284
		16.4 小结	284

第三篇 LabVIEW 应用实战技术

第 17 章 测试系统与虚拟仪器	286	18.3.3 从 MAX DAQmx 任务生成代码	309
17.1 测试系统的概念	286	18.3.4 在 LabVIEW 中使用 NI-DAQmx 任务	309
17.1.1 传感器	286	18.4 高级数据采集	312
17.1.2 信号调理	287	18.4.1 高级数据采集函数	312
17.2 数据采集	288	18.4.2 使用任务触发的数据采集	313
17.2.1 数据采集基本原理	288	18.4.3 连续采集	314
17.2.2 数据采集设备	290	18.4.4 数字脉冲计数	314
17.3 测试信号	292	18.5 小结	315
17.3.1 直流电压测量	292	第 19 章 总线技术	316
17.3.2 高电压测量和隔离	294	19.1 LabVIEW 支持的总线	316
17.4 选择并配置 DAQ 测量硬件	295	19.1.1 仪器的发展	316
17.4.1 选择硬件	295	19.1.2 仪器控制介绍	316
17.4.2 安装 DAQ 设备驱动程序	296	19.2 GPIB 总线	323
17.4.3 测量与自动化管理器	297	19.2.1 GPIB 总线定义	323
17.4.4 NI-DAQmx	297	19.2.2 GPIB 总线在 LabVIEW 中的实现	324
17.4.5 在 MAX 中配置 NI-DAQmx 设备	297	19.3 串口总线	326
17.4.6 配置数据采集	300	19.3.1 串口总线定义	326
17.5 小结	301	19.3.2 串口总线在 LabVIEW 中的实现	328
第 18 章 LabVIEW 中的数据采集	302	19.4 串口总线采集应用实例	331
18.1 测量 I/O	302	19.4.1 串口函数应用实例	331
18.1.1 DAQ 助手	302	19.4.2 串口通信中常见问题	332
18.1.2 模拟 I/O	303	19.4.3 串口通信程序调试几点建议	333
18.1.3 数字 I/O	304	19.4.4 GPIB 仪器通信和 RS-232 仪器通信的区别	334
18.2 DAQ 助手实例	304	19.5 USB 总线仪器控制	334
18.2.1 缓冲方式模拟输入	304	19.5.1 USB 总线仪器特点	334
18.2.2 读取数字输入	305		
18.3 NI-DAQmx 任务	306		
18.3.1 创建 NI-DAQmx 任务	306		
18.3.2 在 LabVIEW 中引用 NI-DAQmx 任务	308		

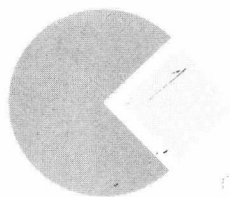
19.5.2	通过配置 NI-VISA, 控制用户的 USB 设备	336	20.3.2	向 DataSocket 写数据	348
19.6	小结	341	20.3.3	从 DataSocket 读数据	348
第 20 章	网络通信	342	20.3.4	DataSocket 传递变体 数据	349
20.1	计算机网络概述	342	20.3.5	DataSocket 使用缓冲方式 传输数据	350
20.1.1	计算机网络的 功能与发展	342	20.3.6	统一资源定位符	350
20.1.2	计算机网络结构	342	20.4	TCP 通信	351
20.1.3	计算机网络模型	343	20.4.1	TCP 函数介绍	351
20.1.4	计算机网络协议	344	20.4.2	TCP 通信应用	352
20.2	DataSocket 技术	344	20.5	在 Web 上发布程序	353
20.2.1	DataSocket 的特点	344	20.5.1	服务器端 Web 发布设置	353
20.2.2	DataSocket 的介绍	344	21.5.2	在 Web 上发布 HTML 文件	354
20.3	DataSocket Server 应用	347	20.5.3	远程面板连接管理器	357
20.3.1	读 DataSocket Server 预定义数据项初始值	347	20.6	小结	358

第四篇 工程应用综合案例

第 21 章	风机自动控制系统	360	第 23 章	听力计控制软件	374
21.1	项目的工程背景	360	23.1	软件背景	374
21.2	系统硬件结构	360	23.2	系统的总体架构	374
21.3	系统软件结构	360	23.2.1	听力计软件事件状态机 结构介绍	374
21.3.1	系统软件介绍	361	23.2.2	事件状态机难点分析	375
21.3.2	变频器通信	362	23.3	串口通信	377
21.4	小结	368	23.3.1	利用 VISA 进行通信 结构开发	377
第 22 章	风速采集、保存与分析	369	23.3.2	判断数据	378
22.1	项目的工程背景	369	23.3.3	延时设置	378
22.2	测试系统原理	369	23.4	实时数据库连接	378
22.2.1	数据采集卡	369	23.4.1	数据库操作的结构	378
22.2.2	数据保存	371	23.4.2	数据库操作的具体实现	379
22.2.3	程序的修饰	371	23.5	小结	380
22.3	小结	373			

第五篇 常见疑难解答与经验技巧集萃

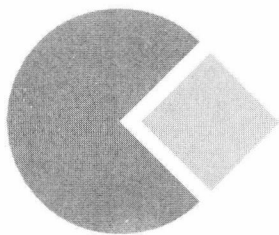
第 24 章	LabVIEW 常见疑难解答与 经验技巧集萃	382	24.2	LabVIEW 与其他应用程序 链接的基础问题	397
24.1	LabVIEW 技术基础问题	382	第 25 章	LabVIEW 工程应用常见 疑难解答与经验技巧集萃	399



第一篇

LabVIEW 技术基础

- ▶ 第 1 章 LabVIEW 概述
- ▶ 第 2 章 LabVIEW 前面板设计
- ▶ 第 3 章 LabVIEW 的编辑环境
- ▶ 第 4 章 数据表达
- ▶ 第 5 章 程序流程和结构
- ▶ 第 6 章 波形显示
- ▶ 第 7 章 同步数据传递
- ▶ 第 8 章 程序动态控制
- ▶ 第 9 章 文件输入/输出
- ▶ 第 10 章 XControl 的制作
- ▶ 第 11 章 生成安装包和可执行文件
- ▶ 第 12 章 程序调试与程序性能



第1章 LabVIEW 概述

1.1 LabVIEW 的基本知识

1.1.1 什么是 LabVIEW

LabVIEW 是 Laboratory Virtual Instrument Engineering Workbench 的缩写，是一款使用图形符号来编写程序的编程环境。在这方面，它不同于传统的编程语言，如 Visual Basic、C、C++ 或 Java，这些语言使用文本方式编程。而 LabVIEW 不仅是一款编程语言，它还是为科学家和工程师等设计的一种编程开发环境和运行系统，编程只是这些人员工作的一部分。LabVIEW 开发环境可以运行在 Windows、Mac 或 Linux 系统的计算机上，用 LabVIEW 编写的应用程序可以运行在上述系统上，还可以运行在 Microsoft Pocket PC、Microsoft Windows CE、Palm OS 以及多种嵌入式平台上，包括 FPGA、DSP、ARM 等微处理器上。

使用 LabVIEW 功能强大的图形编程语言能够提高编程的效率，这种语言被称做 G 语言，即图形编程语言。使用传统的编程语言需要花费几周甚至几个月才能编写的程序，用 LabVIEW 只需几个小时就能完成。因为 LabVIEW 是专为测量、数据分析并提交结果而设计的，且 LabVIEW 拥有如此功能众多的图形用户界面又易于编程，使得它对于仿真、结果显示、通用编程，甚至学习基本编程概念也同样是很理想的语言。

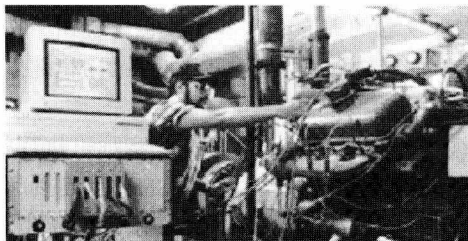
与标准的实验室仪器相比，LabVIEW 提供了更大的灵活性，因为它是基于软件的。LabVIEW 用户无法作为仪器的生产者，但是可以定义仪器的功能。使用计算机、插入式硬件和 LabVIEW 能共同组成一个可完全配置的虚拟仪器，用以完成用户的任务。使用 LabVIEW，用户可以根据需要创建所需的任何类型的虚拟仪器，而其成本仅仅是传统仪器的一小部分。当需求变化时，还可以在瞬间修改虚拟仪器。

LabVIEW 拥有庞大的函数和子程序库，这些库可以帮助用户完成编程中的大部分任务，使得 LabVIEW 用户免于被传统编程语言中指针、内存分配以及其他莫名的编程问题所困扰。LabVIEW 也包含特定的应用程序库代码，如数据采集 (DAQ)、通用功能接口总线 (GPIB)、串行接口仪器控制、数据分析、数据显示、数据存储、Internet 通信等。分析库包含了大量实用的函数，如信号产生、信号处理、滤波器、窗口、统计、回归、线性代数、矩阵运算等。

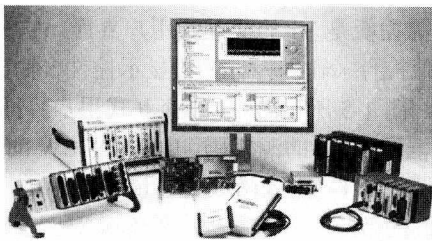
LabVIEW 图形化的本质使它理所当然地可以作为一个数据显示软件包，可以任何我们想要的方式显示输出结果，图表、图形和用户自定义图形仅是可选输出方式中的一部分。

图 1-1 所示为自动润滑检测的应用。

图 1-2 所示为运用 LabVIEW 进行编程的工作环境。



▲图 1-1 自动润滑检测的应用



▲图 1-2 运用 LabVIEW 进行编程的工作环境

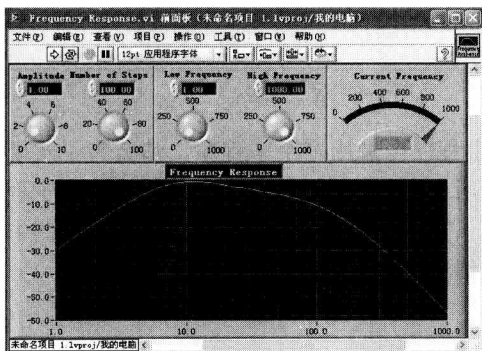
1.1.2 数据流的概念

LabVIEW 编程开发环境与标准 C 或 Java 开发系统的一个重要区别是：标准语言编程系统采用基于文本的代码行编程；而 LabVIEW 则使用图形编程语言，通常被称做 G 语言，在称为框图的图形框架内编程。

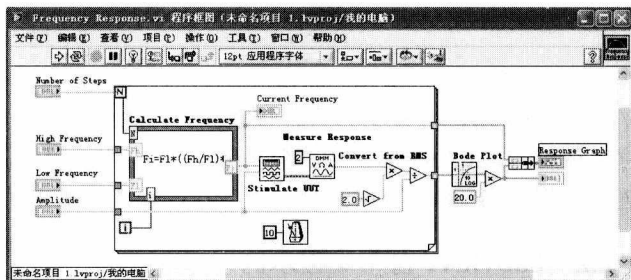
图形编程消除了文本编程中设计的许多语法细节，如 (；) 和 ({}) 的放置经常会让人煞费脑筋。在 LabVIEW 中就不用担心这样的问题，即使是某个地方出现了连接错误，LabVIEW 也会明确地指出错误，而不是像文本编程语言那样模棱两可。

图形编程语言中最需要关注的是应用程序中的数据流，因为其简单的语法使程序变得更加简洁易懂。图 1-3 和图 1-4 所示为用 LabVIEW 编写的程序前面板和程序框图。

LabVIEW 使用科学家和工程师们熟悉的术语、图标和概念，依靠图形符号而不是文本语言来定义程序的功能。LabVIEW 的运行基于数据流的原理，一个函数只有收到必要的数据后才可以运行。基于这些特点，对于没有编程经验的人来说也是可以学会 LabVIEW 的。



▲图 1-3 用 LabVIEW 编写的程序前面板



▲图 1-4 用 LabVIEW 编写的程序框图

12 LabVIEW 的工作环境和编程示例

1.2.1 LabVIEW 的工作环境

一款 LabVIEW 程序由一个或多个虚拟仪器组成，虚拟仪器在后面简称为 VI。而之所以称做虚拟仪器，是因为它们的外观和操作通常是模拟了实际的物理仪器。然而，在这些面板之后，它们有着类似于流行的编程语言，如 C 和 B 语言中的主程序、函数、子程序等。

一般每个 VI 都由 3 个主要部分组成：前面板、框图和图标。

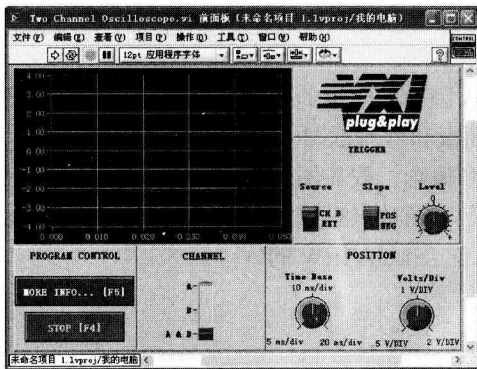


注

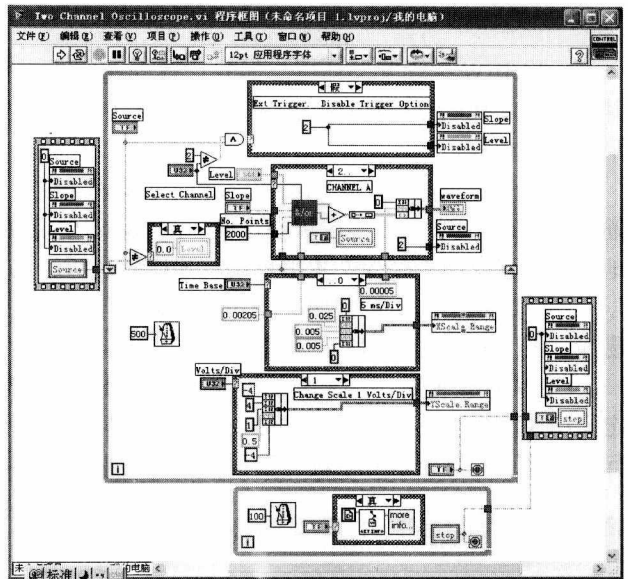
全局变量是一种特殊的 VI，只有前面板，没有框图，对图标可以编辑，但是很少有人编辑。

- **前面板**是 VI 的交互式用户界面，它模拟了物理仪器的前面板。前面板包含旋钮、按钮、图形及其他控件（输入控件）和显示控件（输出控件），而且通过编程，可以使用鼠标和键盘作为输入设备。图 1-5 所示为 VI 的交互式用户界面。

- **框图**是 VI 的源代码，由 LabVIEW 的图形化编程即 G 语言构成。框图是实际可执行的程序。框图由低级 VI、内置函数、常量和程序执行控制结构等构成，用连线将合适的对象连接起来定义它们之间的数据流。前面板上的对象对应于框图上的终端，这样数据就可以从用户传递到程序，再回传给用户。图 1-6 所示为前面板所对应的程序框图。



▲图 1-5 交互式用户界面



▲图 1-6 程序框图

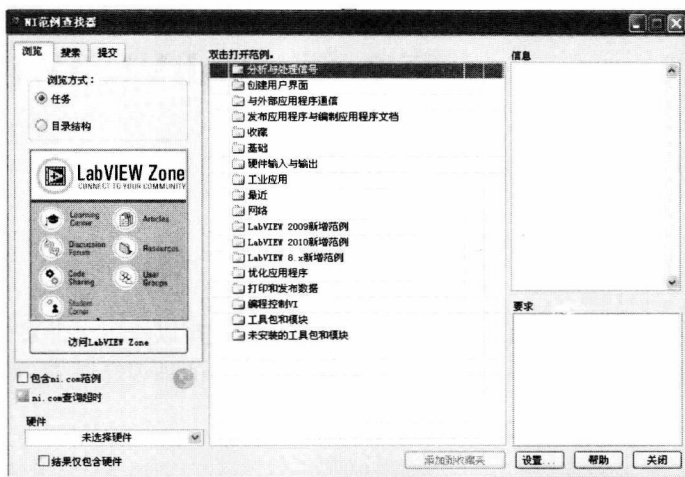
- 在编写 LabVIEW 应用程序时，往往需要在一个主程序中调用多个子程序，那么为了实现 VI 之间的调用，VI 就必须有连接器图标。被另外一个 VI 所使用的 VI 称为子 VI，也可以称为子程序。图标是 VI 的图形表示，会在另外的 VI 框图中作为一个对象使用，连接器用于从其他框图中连线数据到当前 VI。连接器定义了 VI 的输入和输出，类似于子程序的参数。

1.2.2 LabVIEW 自带编程示例

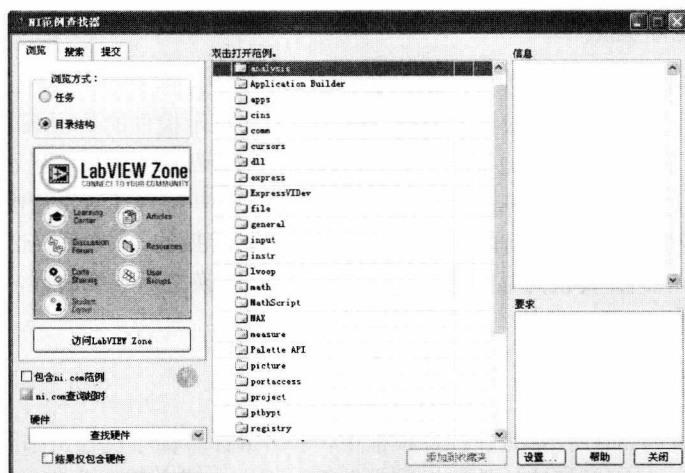
与其他编程语言不同，LabVIEW 拥有庞大的自带编程示例，这有助于学习一般的编程技术并了解完成通用硬件输入/输出和数据处理任务的应用程序。通过 NI 范例查找器可以找到相关的范例。有几个途径可以进入 NI 范例查找器，一为在启动页面的右下角单击“查找范例…”；二为在启动 LabVIEW 编程界面后选择“帮助”菜单，从中选择“查找范例”菜单项。NI 范例查找器打开后如图 1-7 所示。

浏览方式可以设置为任务或者目录结构。按任务浏览时，目录结构即按照任务编辑，可以很明了地根据自己的项目任务来选择相近的实例。按目录结构浏览时如图 1-8 所示，用户可以根据自己在编程时遇到的难点来查找相应的实例。

使用 NI 范例查找器，还可以根据关键字直接搜索相关的示例，如 Serial、Excel 等，NI 范例查找器将直接定位到相关的示例。



▲图 1-7 NI 范例查找器

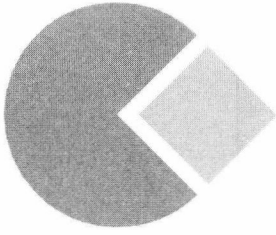


▲图 1-8 根据目录结构浏览范例

1.3 小结

LabVIEW 是功能强大而又灵活的仪器应用和分析软件，它使用图形化编程语言，有时也称做 G 语言，创建成为虚拟仪器或 VI 的程序。用户与程序通过前面板进行交互。每个前面板有一个与之对应的框图，也就是 VI 的源代码。LabVIEW 有很多内置函数，便于进行编程。在框图中，将部件连接起来显示数据流向。

使用 NI 范例查找器可以查找所学主题相关的示例，可以通过任务或目录结构来浏览示例。



第2章 LabVIEW 前面板设计

前面板是程序与用户交流的窗口，一个设计良好的前面板可以给用户带来一个友好的感觉。前面板主要是由输入和输出控件构成，本章主要介绍前面板控件的外观等细节，对于该控件的程序应用将在第4章讲解。

2.1 LabVIEW 前面板控件概述

2.1.1 LabVIEW 控件类型

前面板控件有些是用户用来向程序中输入数据的，这些控件称做输入控件；另一些则是程序向用户输出运行结果的，这些控件称做显示控件。输入控件和显示控件的数据流方向刚好相反。输入控件的端口边框比显示控件的端口边框粗，而且输入控件的接线端在右侧，显示空间的接线端在左侧。有些控件在模板上有输入控件和显示控件两种类型。而有些控件比较常用哪种类型就会给出哪种类型。但是控制件和显示件放在前面板上以后是可以互相转换的。转换的方法很简单，在控件或它的端口的弹出菜单中选择转换为显示/输入控件即可。若转换之后连线断开，或者无法运行，则说明该控件只能用做输入/显示控件。

2.1.2 LabVIEW 控件选板

设计前面板所用的全部控件都是在控件模板中，单击鼠标右键可以弹出临时的控件选板，在控件选板上按下固定销，就可以把该控件选板固定在前面板上；或者单击菜单栏的查看>>控件选板，可以打开固定的控件选板。控件选板的几种显示方式如图2-1所示。

控件的种类有数值输入控件（如滑动杆和旋钮）、数值显示控件（如仪表和量表、图表）、布尔控件（如按钮和开关）、字符串、路径、数组、簇、列表框、树形控件、表格、下拉列表控件、枚举控件和容器控件等。



▲图2-1 控件选板的几种显示方式

前面板控件有新式、经典和系统等3种样式。

选择**文件>>VI 属性**，从**类别**列表中选择**编辑器选项**改变控件的样式。然后右键单击接线端，从快捷菜单中选择**创建>>输入控件**或**创建>>显示控件**，创建出的控件的样式就会产生相应的改变。选择**工具>>选项**，从**类别**列表中选择**前面板**，可以改变控件样式。此后，如通过右键单击接线端，再从快捷菜单中选择**创建>>输入控件**或**创建>>显示控件**而创建控件时，新建控件的样式就会产生相应的改变。

- 新式及经典控件

许多前面板对象具有高彩外观。为了获取对象的最佳外观，显示器最低应设置为16色位。

位于新式面板上的控件也有相应的低彩对象。经典选板上的控件适于创建在256色和16色显示器上显示的VI。

- 系统控件

位于系统选板上的系统控件可用在用户创建的对话框中。系统控件专为在对话框中使用而特别设计，包括下拉列表和旋转控件、数值滑动杆、进度条、滚动条、列表框、表格、字符串和路径控件、选项卡控件、树形控件、按钮、复选框、单选按钮和自动匹配父对象背景色的不透明标签等。这些控件仅在外观上与前面板控件不同，这些控件的颜色与为系统设置的颜色相同。

在不同的VI运行平台上，系统控件的外观也不同。在不同的平台上运行VI时，系统控件将改变颜色和外观，与该平台的标准对话框控件匹配。

2.2 LabVIEW 控件选板详细分类

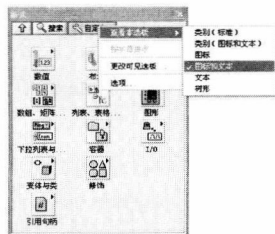
单击控件选板的自定义按钮，在弹出的菜单上选择查看本选板，再选择类别（图标和文本），控件选板每个类别图标下面将会添加上文本标识，如图2-2所示。

2.2.1 数值控件

数值控件是输入和显示数值数据的最简单方式。对这些前面板对象可在水平方向上调整大小，以显示更多位数。

为数值控件输入一个新的数值时，工具栏上会出现**确定输入**按钮，提醒用户只有按下回车键，或在数字显示框外单击鼠标，或单击**确定输入**按钮时，新数值才会替换旧数值。VI运行时，LabVIEW将一直处于等待状态，直到用户进行上述某一操作从而确认新数值。例如，将数字显示框中的数值改为135时，VI不会接收1或13，而是接收完整的135。

默认状态下，LabVIEW的数字显示和存储与计算器类似。数值控件一般最多显示6位数字，超过6位则自动转换为以科学计数法表示。右键单击数值对象，从快捷菜单中选择**显示格式**，打开**数值属性**对话框的**显示格式**选项卡，从中可配置LabVIEW在切换到科学计数法之前所显示的数字位数。图2-3所示为数值控件子选板。



▲图2-2 图标和文本方式显示控件选板



▲图2-3 数值控件子选板