

 电子信息与电气学科规划教材

LabVIEW

程序设计教程

江建军 刘继光 编著

- ◎结合多年教学与应用程序开发经验
- ◎提取 LabVIEW 程序设计的通用思想
- ◎涵盖 LabVIEW 基础知识和高级设计
- ◎提供丰富的实例、习题、建议和课件



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY <http://www.phei.com.cn>

TP312/2798

2008

电子信息与电气学科规划教材

LabVIEW 程序设计教程

LabVIEW Program Design Methodology

江建军 刘继光 编著

LabVIEW 程序设计教程 / 江建军, 刘继光编著. — 北京: 电子工业出版社, 2008.3

(清华大学出版社教材)

ISBN 978-7-121-05932-3

I. ①L... II. ①江... ②刘... III. 软件工具 IV. TP311.56

中国版本图书馆CIP数据核字(2008)第012118号

责任编辑: 刘继光

封面设计: 刘继光

印刷: 深圳市南山区印刷厂

发行: 深圳市南山区印刷厂

出版发行: 电子工业出版社

北京市西城区百万庄大街24号

开: 本: 787×1092 1/16

印: 次: 2008年3月第1次印刷

印: 数: 3000册 定价: 30.00元

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

服务热线: (010)88258888

内 容 简 介

本书从程序设计的角度全面介绍了 LabVIEW 的编程技术和工程应用。全书分为 11 章, 包括: LabVIEW 编程基础、LabVIEW 编程环境、LabVIEW 程序设计模式、LabVIEW 与多线程、LabVIEW 程序调试与异常处理、LabVIEW 面向对象程序设计、LabVIEW 程序动态运行、LabVIEW 调用操作系统功能、LabVIEW 应用程序生成规范、LabVIEW 在测试测量中的应用和在线实验系统设计。

本书各章设有提示与建议, 并配有习题, 以帮助读者巩固学习重点。本书的教学课件和源代码可从华信教育资源网 (www.huaxin.edu.cn) 免费下载。

本书可以作为普通高等学校虚拟仪器技术与 LabVIEW 程序设计相关课程的教材, 也可作为工业界或科技界使用 LabVIEW 作为测试测量系统开发工具的工程师或科研人员的参考书。

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, 侵权必究。

图书在版编目 (CIP) 数据

LabVIEW 程序设计教程 / 江建军, 刘继光编著. —北京: 电子工业出版社, 2008.3

(电子信息与电气学科规划教材)

ISBN 978-7-121-05935-3

I. L… II. ①江…②刘… III. 软件工具, LabVIEW—程序设计—高等学校—教材 IV. TP311.56

中国版本图书馆 CIP 数据核字 (2008) 第 015118 号

策划编辑: 张 濮

责任编辑: 刘 凡

印 刷: 涿州市京南印刷厂

装 订: 涿州市桃园装订有限公司

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: 19.25 字数: 493 千字

印 次: 2008 年 3 月第 1 次印刷

印 数: 3000 册 定价: 30.00 元

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 zltts@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

PREFACE

● 产业应用背景

伴随着以计算机和网络为代表的信息技术的快速发展,基于计算机软件平台的测量控制系统被广泛地应用于各行各业中,“软件就是仪器”的思想得到了广泛的认同和实践。同时,美国国家仪器公司(National Instruments, NI)提供的虚拟仪器程序设计语言 LabVIEW,已经成为开发测量控制系统的重要工具,在高等院校和工业部门等不同的领域中都得到了普遍应用。自 1986 年第一版发布以来,LabVIEW 已经走过了二十多年。2006 年,NI 公司发布了 20 年的纪念版 LabVIEW 8.20 版。随着版本的不断升级,LabVIEW 在用户开发界面的友好性、系统开发的模块化和数据分析的功能化等方面都得到了不同程度的改进,更适合于不同基础的工程技术人员使用。近年来,随着国内虚拟仪器技术应用的不断发展,接触和从事 LabVIEW 程序设计的工程技术人员数量也不断扩大,并且出现了一批以 LabVIEW 作为主要开发平台的程序员。

自发布以来,LabVIEW 设计思想一直着眼于为那些未接受过计算机程序设计方法培训的专业工程技术人员提供一种测量控制系统的程序开发工具。因此,LabVIEW 具有简单直观、易于理解等特点。因为使用了图形化的编程方法,LabVIEW 简化了测量控制系统的开发过程,缩短了系统开发和调试的周期,但同时也可能造成系统开发过程中的一些问题。主要表现为,在开发大型或复杂的测量控制系统时,不具有良好设计风格的 LabVIEW 程序将很难进行修改、调试和复用。即使是具有传统程序设计经验和良好编程习惯的程序设计人员,如果对 LabVIEW 程序设计的原理、方法和特点没有深入的理解和掌握,而仍延用传统语言的程序设计模式,那么所开发出来的系统可能依然存在一些隐含错误。

● 写作特色

为了解决 LabVIEW 工程应用中的设计问题,我们结合多年创新教学的实践经验编写了本书。首先,本书并不以 LabVIEW 的某个特定版本作为基础来介绍这种图形化语言的使用过程,而是从一门程序设计语言的角度,介绍并分析了 LabVIEW 外部程序设计风格、内部实现原理、代码编译特点等重要内容。其次,本书从程序设计原理和方法的角度,将 LabVIEW 与传统的文本式编程语言在各个方面进行了对比和分析,力求帮助不同基础的学习者理解掌握 LabVIEW 程序设计的特点。最后,本书抽取出了 LabVIEW 程序设计的通用思想和设计模式,丰富了 LabVIEW 程序设计理论。

本书在写作上突出理论联系实际的特点,一方面通过理论讲解突出计算机程序设计理论与 LabVIEW 程序设计的联系,另一方面则通过实例说明 LabVIEW 程序设计方法与具体开发过程的联系。我们觉得,写作本书并不应仅仅满足于告诉读者怎么做,更重要的是启发读者自己去思考,力争将一种优秀的程序设计思想传达给读者。这样,即使以后遇到书中没有涉及的问题,读者也会有能力根据这种思想去解决问题。

● 内容结构

本书共分 11 章,包括 LabVIEW 编程基础、LabVIEW 编程环境、LabVIEW 程序设计模式、LabVIEW 与多线程、LabVIEW 程序调试与异常处理、LabVIEW 面向对象程序设计、LabVIEW 程序动态运行、LabVIEW 调用操作系统功能、LabVIEW 应用程序生成规范、LabVIEW 在测试测量中的应用和在线实验系统设计。其中,前两章为 LabVIEW 程序设计的基础知识,使初学者对 LabVIEW 编程有一定的了解;第 3~9 章为 LabVIEW 的高级程序设计技术,帮助有一定基础的读者提高 LabVIEW 程序设计能力和效率;第 10 章介绍 LabVIEW 在测试测量中的应用,详细讲解如何使用 LabVIEW 控制 DAQ、控制仪器和创建典型的测量应用;第 11 章为 LabVIEW 程序设计的应用,通过讲解使用 LabVIEW 编写的 VI-1508 在线实验系统,使读者进一步掌握如何使用 LabVIEW 设计较大规模的应用程序。

全书内容从 LabVIEW 编程基础到高级编程,再到应用开发,从浅到深,一脉相承。书中内容尽量考虑了各个不同层次读者的需要,涵盖了与 LabVIEW 程序设计有关的诸多内容,重点突出了 LabVIEW 程序设计的模式、LabVIEW 面向对象编程、LabVIEW 异常处理等内容。

本书可以作为普通高等学校中虚拟仪器技术与 LabVIEW 程序设计相关课程的教材,也适合作为工业界或科技界以 LabVIEW 作为测试测量系统开发工具的工程师或科研人员的进阶参考书。对于初学者,学习时应该按照章节顺序阅读本书;对于有一定经验的读者,可以选择相关的章节阅读,书后的附录可作为参考资源。

本书主要由江建军教授和刘继光博士编著。参加本书相应部分编写工作的人员有华中科技大学电子科学与技术系的刘浩(第 3 章、第 9 章和第 10 章)、黄建喜(第 4 章和第 6 章)、王一楠(第 2 章和第 7 章)、刘成(第 5 章和第 8 章)、袁秋(第 1 章和第 11 章),张秀成教授参与了部分章节的审阅工作,邹时磊、梅雨、赵文峰和徐燕子承担了文稿整理和部分绘图工作,全书由江建军教授与刘继光博士统稿。

● 学习支持

本书中的所有实例,都在 Windows XP Professional 操作系统平台上测试过。每章末尾都有总结性质的“提示与建议”,帮助读者掌握 LabVIEW 程序设计的关键点和小技巧,然后还配有习题,帮助读者巩固所学的知识。这些内容都是从教学过程中提炼出来的,希望能对读者的学习有一定的帮助。

书中实例源程序和教学课件等教学资源可以在华信教育资源网(www.huaxin.edu.cn)上免费注册下载。如果读者在学习过程中遇到问题,也可以通过 E-mail: jiangjj@mail.hust.edu.cn 作者联系交流。

● 致谢

感谢 CCMS 实验室的研究生为本书的撰写奠定了基础,尤其要感谢范少春博士、明繁华硕士、杨彪硕士、周毅硕士等为在线试验系统的开发做出的很大贡献。

感谢 NI 公司的技术人员在撰写本书过程中提供的帮助,特别要感谢 NI 公司大中华区总经理陈大庞博士、高校市场部经理陈庆全先生和上海聚星仪器有限公司邵晖博士对我们的鼎力支持。

另外,在此还要感谢电子工业出版社编辑们的辛勤劳动,感谢所有为本教材进行审阅并提出宝贵意见,以及在编写出版过程中给予热情帮助和支持的同志们。

由于作者水平有限,书中难免有疏漏和错误之处,恳请广大读者批评指正。



2008 年 1 月于华中科技大学喻园

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目 录

CONTENTS

第 1 章 LabVIEW 编程基础	1
1.1 LabVIEW 概述	2
1.1.1 虚拟仪器技术	2
1.1.2 LabVIEW 开发系统	3
1.1.3 LabVIEW 帮助选项	4
1.1.4 LabVIEW 开发环境	5
1.2 创建前面板	10
1.2.1 数据控件	10
1.2.2 窗口控件	17
1.2.3 特殊控件	22
1.3 创建背面板	23
1.3.1 程序结构	23
1.3.2 数据运算	25
1.3.3 高级工具	26
1.3.4 仪器 I/O	26
1.3.5 仪器驱动	27
1.3.6 信号处理	28
1.4 创建子程序	30
1.4.1 图标与连线板	30
1.4.2 创建与调用函数	32
1.5 LabVIEW 编程风格	34
1.5.1 前面板规范	34
1.5.2 背面板风格	36
1.5.3 文件目录组织	38
1.6 LabVIEW 8.5 新特性	38
1.7 提示与建议	39
1.8 习题	41

第 2 章 LabVIEW 编程环境 43

2.1 软件环境	44
2.1.1 系统环境	44
2.1.2 编程环境	47

2.1.3 VI 服务器	52
2.1.4 Web 服务器	53
2.2 VI 属性设置	55
2.2.1 基本属性	55
2.2.2 窗口属性	55
2.2.3 前面板设置	56
2.2.4 运行菜单设置	56
2.2.5 运行属性	57
2.2.6 自定义控件	58
2.3 其他工具	58
2.3.1 数据日志	58
2.3.2 源代码管理	59
2.3.3 LLB 管理	61
2.3.4 MathScript	61
2.4 提示与建议	62
2.5 习题	63

第 3 章 LabVIEW 程序设计模式 65

3.1 面向对象设计模式	66
3.1.1 作用	66
3.1.2 要素	66
3.2 LabVIEW 程序设计模式	67
3.2.1 应用程序的基本框架	67
3.2.2 图形化数据流编程	68
3.2.3 设计模式分类	69
3.3 状态机模式 (State Machines)	69
3.3.1 状态机	69
3.3.2 枚举类型	70
3.3.3 顺序型状态机模式 (Queued State Machine)	71
3.3.4 改进的顺序型状态机模式	73
3.3.5 测试流程型状态机	73

3.4 消息队列处理模式	74	4.3.3 可重入技术	111
3.4.1 消息队列	75	4.4 LabVIEW 线程通信与同步	112
3.4.2 队列方式	75	4.4.1 局部变量	112
3.4.3 使用数组处理消息队列	76	4.4.2 全局变量	113
3.4.4 使用队列函数处理消息队列	79	4.4.3 事件发生	114
3.5 用户界面事件模式	79	4.4.4 通知	116
3.5.1 事件框架	80	4.4.5 队列	118
3.5.2 用户界面事件	80	4.4.6 信号量	119
3.5.3 用户自定义事件	81	4.4.7 集合点	121
3.5.4 事件注册	82	4.5 提示与建议	122
3.5.5 用户界面事件示例	84	4.6 习题	122
3.6 状态机—用户界面事件混合模式	86	第 5 章 LabVIEW 程序调试与异常处理 125	
3.6.1 状态机与用户界面事件的结合	86	5.1 概述	126
3.6.2 状态机—用户界面事件示例	87	5.1.1 程序调试	126
3.7 其他模式	88	5.1.2 异常处理	126
3.7.1 主从线程模式	88	5.2 LabVIEW 错误信息	127
3.7.2 生产消费模式	88	5.2.1 I/O 错误	127
3.7.3 后台服务模式	89	5.2.2 逻辑错误	128
3.7.4 应用程序启动模式	90	5.2.3 错误簇	128
3.7.5 代理模式	90	5.2.4 错误代码	130
3.8 提示与建议	92	5.3 内置异常处理工具	131
3.9 习题	93	5.3.1 简易异常处理器	131
第 4 章 LabVIEW 与多线程 95		5.3.2 通用异常处理器	132
4.1 多线程技术	96	5.3.3 查找第一个错误	132
4.1.1 基本概念	96	5.4 LabVIEW 异常处理过程	132
4.1.2 线程调度	97	5.4.1 异常处理时机	132
4.1.3 线程通信与同步	98	5.4.2 自定义错误	133
4.2 LabVIEW 多线程环境	99	5.4.3 一般处理模式	134
4.2.1 执行系统	99	5.4.4 While 循环处理模式	135
4.2.2 线程数量设置	101	5.4.5 条件结构处理模式	135
4.2.3 VI 优先级	102	5.4.6 状态机处理模式	136
4.2.4 优先级问题	104	5.4.7 错误日志	136
4.2.5 subroutine 优先级	104	5.4.8 错误退出	137
4.2.6 多线程程序的优势	105	5.4.9 外部错误管理	137
4.3 LabVIEW 多线程程序设计	106	5.4.10 实例	139
4.3.1 线程优化	106	5.5 LabVIEW 程序调试技术	140
4.3.2 超线程程序	109	5.5.1 错误列表	141
		5.5.2 高亮执行	141
		5.5.3 单步执行	142

5.5.4 探针工具	142
5.5.5 保存连线值工具	144
5.5.6 断点工具	144
5.5.7 悬挂执行	144
5.5.8 使能部分程序框图	145
5.6 提示与建议	147
5.7 习题	148

第6章 LabVIEW 面向对象程序设计 149

6.1 面向对象	150
6.1.1 对象与类	150
6.1.2 属性与方法	150
6.1.3 三要素	151
6.2 LabVIEW 与面向对象程序 设计	152
6.2.1 术语	152
6.2.2 继承	153
6.2.3 封装	153
6.2.4 多态	153
6.3 LabVIEW 对象基本实现	154
6.3.1 创建类	154
6.3.2 设置继承	154
6.3.3 创建方法	156
6.3.4 使用类	158
6.4 LabVIEW 对象高级技巧	159
6.4.1 构造函数与析构函数	159
6.4.2 内存分布	159
6.4.3 动态分配	160
6.4.4 XControl	161
6.4.5 多态技术	163
6.4.6 锁定与探针	164
6.4.7 未实现的技术	167
6.5 实例介绍	168
6.5.1 测试要求	168
6.5.2 解决方案	169
6.5.3 面向对象的优势	170
6.5.4 总结	171
6.6 提示与建议	172
6.7 习题	172

第7章 LabVIEW 程序动态运行 173

7.1 动态运行基础	174
7.1.1 VI Server 技术	174
7.1.2 Refnum 控件	174
7.2 动态运行功能	175
7.2.1 相关函数	175
7.2.2 编程步骤	177
7.2.3 属性控制	178
7.2.4 方法调用	179
7.2.5 属性特征	179
7.2.6 动态调用	180
7.2.7 远程访问	180
7.3 动态注册事件	181
7.3.1 相关函数	181
7.3.2 功能扩展	181
7.3.3 编程步骤	182
7.4 动态运行编程实例	183
7.4.1 属性设置和方法调用	183
7.4.2 动态调用	184
7.4.3 类型操作	187
7.4.4 远程调用	189
7.4.5 拖放应用	190
7.4.6 代码优化	192
7.5 提示与建议	194
7.6 习题	194

第8章 LabVIEW 调用操作系统功能 195

8.1 读/写电子表格文件	196
8.1.1 写操作	196
8.1.2 读操作	197
8.2 读/写 XML 文件	197
8.2.1 写操作	198
8.2.2 读操作	199
8.3 读/写配置文件	200
8.3.1 写操作	200
8.3.2 读操作	202
8.4 读/写系统注册表	203
8.4.1 写操作	204
8.4.2 读操作	205
8.4.3 删除操作	205

8.4.4 自动配置 ODBC 数据源	206	10.2.4 测量电阻	252
8.5 输入设备控制	206	10.2.5 产生电压信号	253
8.6 调用动态链接库(DLL)	207	10.3 使用 LabVIEW 控制仪器	254
8.6.1 动态链接库简介	207	10.3.1 仪器驱动的结构	254
8.6.2 DLL 与 API	208	10.3.2 仪器驱动的类型	255
8.6.3 调用 DLL	209	10.3.3 LabVIEW 中的 VISA 模块	256
8.6.4 参数配置	210	10.4 使用 LabVIEW 创建典型的测量应用	258
8.6.5 调用 Windows API	212	10.4.1 使用 LabVIEW 创建典型的 DAQ 应用	258
8.7 ActiveX	213	10.4.2 使用 LabVIEW 创建典型的 VISA 应用	259
8.7.1 ActiveX 自动化(Automation)	214	10.5 提示与建议	260
8.7.2 ActiveX 容器(Container)	215	10.6 习题	260
8.7.3 ActiveX 事件(Events)	215	第 11 章 在线实验系统设计	261
8.7.4 应用实例	216	11.1 在线实验系统	262
8.8 执行系统命令	219	11.1.1 网络虚拟实验室仿真实验平台	262
8.9 提示与建议	221	11.1.2 VI-1508 实验系统	263
8.10 习题	221	11.1.3 网络通信技术	263
第 9 章 LabVIEW 应用程序生成规范	223	11.2 LabVIEW 软件工程	268
9.1 概述	224	11.2.1 软件生命周期	269
9.2 使用程序生成规范	224	11.2.2 软件开发模型	269
9.2.1 创建源代码发布(Source Distribution)	225	11.3 LabVIEW 项目管理	273
9.2.2 创建独立应用程序(EXE)	227	11.4 在线实验系统程序设计实例	274
9.2.3 创建共享库(DLL)	232	11.4.1 第一步 需求分析	275
9.2.4 创建 Zip 压缩文件	233	11.4.2 第二步 系统详细设计	276
9.2.5 创建 Windows 安装程序(Installer)	235	11.4.3 第三步 编写程序	281
9.3 提示与建议	239	11.4.4 第四步 测试程序	281
9.4 习题	240	11.4.5 第五步 程序发布	281
第 10 章 LabVIEW 在测试测量中的应用	243	11.4.6 第六步 文档	281
10.1 测试测量基础	244	11.5 提示与建议	282
10.1.1 信号源	244	11.6 习题	282
10.1.2 信号调理	244	附录 A 快捷操作	283
10.1.3 信号采样	247	附录 B 错误代码表	285
10.1.4 触发	249	附录 C LabVIEW 术语表	287
10.1.5 信号分析	249	参考文献	291
10.2 使用 LabVIEW 控制 DAQ	250	后记	293
10.2.1 测量直流电压	251		
10.2.2 测量交流电压	252		
10.2.3 测量电流	252		

第 1 章

LabVIEW 编程基础

CHAPTER 1

LabVIEW (Laboratory Virtual Instrument Engineering Workbench)

是美国国家仪器 (National Instruments, NI) 公司开发的应用于工业测试测量的集成开发环境, 也是工业上广泛使用的一种功能强大的图形化编程软件。本章将概括性地介绍 LabVIEW 的基本功能、新特性及 LabVIEW 的编程风格。

- ▶ LabVIEW 概述
- ▶ 创建前面板
- ▶ 创建背面板
- ▶ 创建子程序
- ▶ LabVIEW 编程风格
- ▶ LabVIEW 8.5 新特性

1.1 LabVIEW 概述

本书中涉及的 LabVIEW 技术是笔者使用 LabVIEW 以来的一些切身感受，并不针对固定的某个 LabVIEW 开发版本，但本书截图实例来自于 LabVIEW 8.20，它是 LabVIEW 诞生 20 周年的纪念版，同时也是第一个支持简体中文的 LabVIEW 版本，图 1-1 所示为 LabVIEW 8.20 的启动界面。

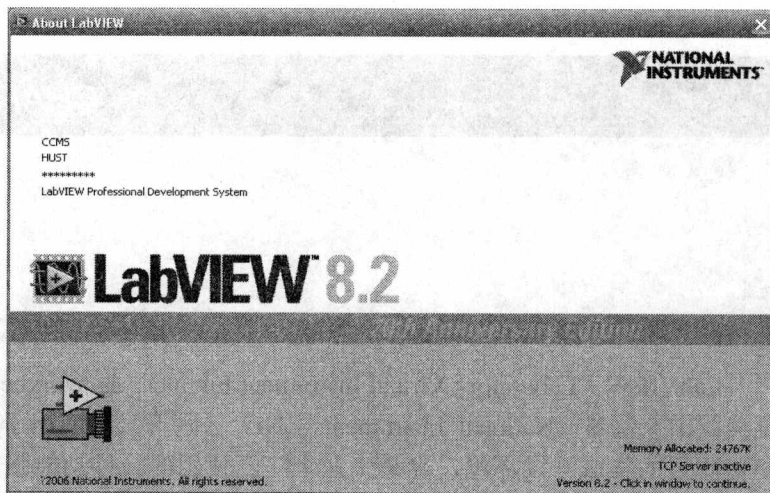


图 1-1 LabVIEW 8.20 启动界面

1.1.1 虚拟仪器技术

虚拟仪器技术首先由 NI (National Instruments) 公司提出，它是以计算机软、硬件技术为核心，以自动控制技术、传感器技术、现代信号处理技术、现代网络技术、数值分析技术为支撑，以各专业学科为应用背景的现代测试技术。它利用高性能的模块化集成概念和方法，结合软件设计平台高效、简便的程序编译功能，依据用户各类特殊需求创建出人机对话界面，实现并取代各类特殊、昂贵的测试仪器的功能，目前已成为测试理论和应用实验研究的重要支撑。

20 世纪 70 年代，因为个人电脑技术的出现，人们开始考虑用电脑来处理传统仪器测试的数据，同时 GPIB 技术也发展起来，促进了 IEEE 488.2 标准的诞生；20 世纪 80 年代，随着计算机技术进一步发展，计算机主板上有了多个扩展槽，并出现了插在计算机里的数据采集卡，这样的系统已经可以进行一些简单的数据采集工作，将采集到的数据直接由计算机软件进行处理，这就是虚拟仪器技术的雏形；20 世纪 90 年代，计算机总线速度进一步提高，PCI 总线的数据传输速率达到了 132 Mbps，1996 年底，NI 公司在 PCI 数据总线的基础上提出了第一代 PXI (PCI eXtensions for Instrumentation) 系统的技术规范。

到 21 世纪初，全球已有超过 25 000 用户在使用虚拟仪器技术，其中不乏国际知名的大公司，像 Nokia、Siemens、Tektronix 等。在世界财富 500 强中的制造业厂商，有 95% 都采用了虚拟仪器技术。据专家预测，到 2010 年我国将有 50% 的仪器为虚拟仪器。虚拟仪器将在航空、航天、通信、医疗、电力、石油、铁路等行业普及应用。表 1-1 列出了虚拟仪器和传统仪器各自的特点。

表 1-1 虚拟仪器与传统仪器的比较

虚 拟 仪 器	传 统 仪 器
用户自定义	生产厂家定义
开发和维护费用低	开发和维护费用高
技术更新周期短（0.5~1 年）	技术更新周期长（5~10 年）
软件是关键	硬件是关键
价格低廉	价格昂贵
开放灵活、与计算机同步，可重复用和重新配置	固定
可利用网络与各远程仪器设备进行通信	只可连接有限设备
自动、智能化、远距离传输	功能单一，操作不便

1.1.2 LabVIEW 开发系统

图 1-2 所示为 LabVIEW 8.20 的导航窗口，相对于 8.0 以前的版本有了较大的改动，呈现出更多的新特性。

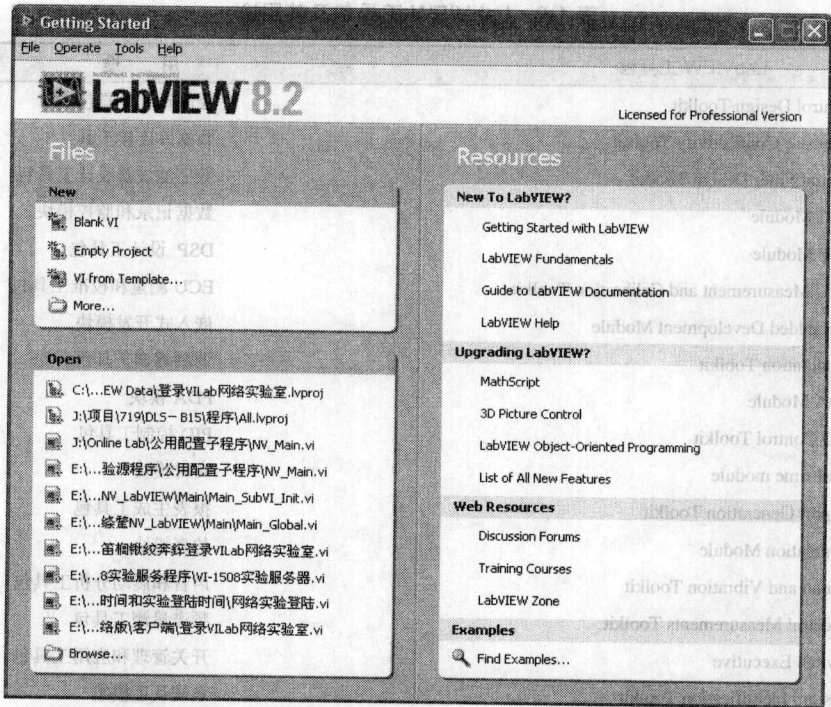


图 1-2 LabVIEW 8.20 的导航窗口

窗口的右边栏目中已经给出了 LabVIEW 8.20 所有的新特性：包括与 The MathWorks 公司 MATLAB 软件的兼容性、用于构建工业控制器的基于 FPGA 的自定义硬件设计向导、用于设计和维护大型测试系统使用的面向对象的工具、基于 OpenGL 的 3D 显示技术（OpenGL-based 3-D visualization）、创建高级用户界面的全新向导工具，以及为 .NET、Web 和 DLL 自动生成界面等新功能。

自 1986 年 NI 公司正式发布 LabVIEW 1.0 以来的 20 年时间里，LabVIEW 不断完善，已

经从最初简单的图形编程、支持单一平台的开发软件，发展到现在以 LabVIEW 为核心，包括数字信号处理、过程控制、模糊控制等众多开发软件包，运行于主流平台的工业软件开发环境。

LabVIEW 以其完善的架构，将工业通用的 GPIB, PXI, VXI 等总线的仪器集合到一起，为用户提供了统一的开发标准，使用户在开发过程中不再需要考虑具体的硬件平台。NI 公司还提供了业界最大的仪器驱动库，同时支持 TCP/IP 协议和 ActiveX, SQL 等技术，通过网络实现实时数据共享，使整个测试测量不再局限于特定的时间和地点。

在以 LabVIEW 为核心的软件架构上，NI 公司同时提供了运动控制、机器视觉、信号处理、嵌入式开发、模拟仿真等开发包和全面而强大的硬件，为过程控制和工业自动化提供了简单易用的解决方案。基于“软件就是仪器”的理念，LabVIEW 为科研提供了强大的支持。

随着软件和硬件功能的不断完善，NI 公司提供了整套的产品研发解决方案，大大缩短了产品原型研发时间。LabVIEW 不仅仅只是一个软件平台，学习 LabVIEW 软件开发技术的同时，也需要开发人员不断熟悉测控行业的相关技术。以 LabVIEW 为核心，NI 公司针对不同的应用提供了各种软件包，构成了强大的软件开发平台，表 1-2 所示为 NI 提供的主要软件包及其用途介绍。

表 1-2 LabVIEW 工具包及其用途

LabVIEW 工具包	用 途
Control Design Toolkit	控制设计工具包
Database Connectivity Toolset	数据库连接工具
Digital Filter Design Toolkit	数字滤波器设计工具包
DSC Module	数据记录和监控模块
DSP Module	DSP 设计工具包
ECU Measurement and Calibration Toolkit	ECU 测量和校准工具包
Embedded Development Module	嵌入式开发模块
Modulation Toolkit	调制解调工具包
PDA Module	PDA 模块
PID Control Toolkit	PID 控制工具包
Real-time module	实时模块
Report Generation Toolkit	报表生成工具包
Simulation Module	仿真模块
Sound and Vibration Toolkit	声音和震动分析工具包
Spectral Measurements Toolkit	频谱量测工具包
Switch Executive	开关管理和应用工具包
System Identification Toolkit	系统认证模组
Touch Panel Module	触控面板模块

1.1.3 LabVIEW 帮助选项

LabVIEW 在拥有强大功能的同时，也提供了同样强大的帮助系统。对于开发环境中的每个函数和节点，LabVIEW 都提供了在线帮助，打开 LabVIEW 的“Help→Show Context Help”菜单项，当鼠标指向函数和节点时在帮助窗口中将显示出该函数和节点的简短说明，如图 1-3 所示。单击

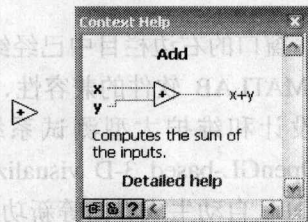


图 1-3 LabVIEW 的实时帮助窗口

“Detailed Help”，可以看到更多关于 Add 节点的帮助，从帮助文档中可以了解到当前节点所有的功能及用法。

LabVIEW 也提供了大量的实例，可以让初学者尽快了解整个开发系统。在不断添加 LabVIEW 工具包的同时，这些实例也会不断增加。如果计算机连通网络，还可以通过 NI 的网站获取更多例子。图 1-4 所示为 LabVIEW 的范例查找器。

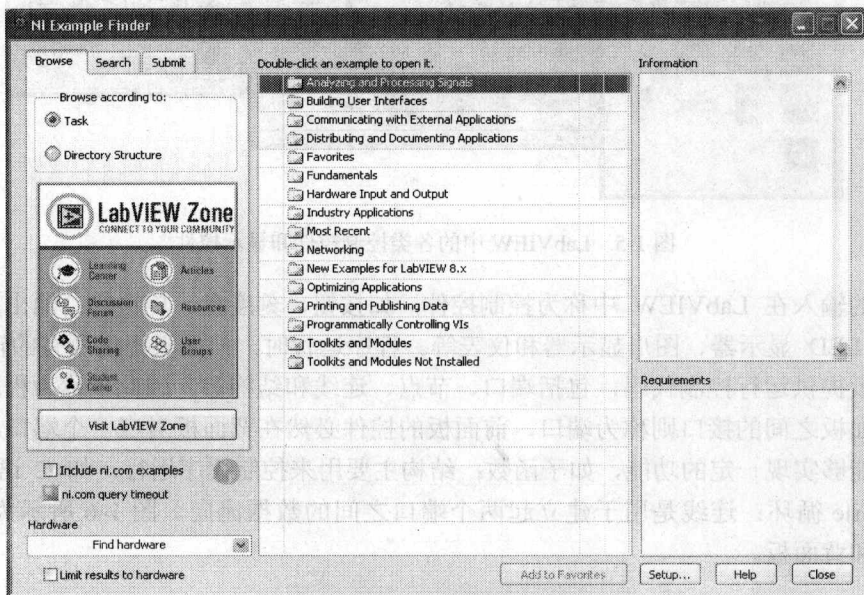


图 1-4 LabVIEW 的范例查找器

在学习 LabVIEW 的过程中，很多人往往忽视了 LabVIEW 帮助功能，其实这些帮助文档是非常重要的，从中可以获取到很多有用的信息，希望读者能够重视。

1.1.4 LabVIEW 开发环境

LabVIEW 程序被称为 VI (Virtual Instrument)，并以 .vi 作为扩展名。LabVIEW 程序通常由前面板和背面板组成：前面板为程序界面放置控制控件和显示控件，背面板用来编写程序框图。

1. 控制控件/显示控件

控制控件为程序提供输入端口，包括数值型、布尔型、文本、枚举等控制控件。对应于各种控制控件，显示控件则完成了各类数据的输出和显示。图 1-5 所示为 LabVIEW 中的各类控制控件和显示控件。

相比于文本编程语言，LabVIEW 提供了基于图形的开发环境。这种开发环境是基于数据流的，程序中一个模块运行与否取决于数据是否到达该模块，即模块所有的接口都得到数据时模块才会运行；运行结束时输出口发送数据，传递给其他模块。基于数据流方式的程序运行模式决定了程序本身就是并行运行的，而普通的文本开发环境中代码运行顺序取决于代码的先后，实现并行运行的程序较为复杂。

同其他开发环境一样，当一个程序调用其他的程序时，被调用的程序可以称为子程序。LabVIEW 程序就是一个模块，这意味着 LabVIEW 程序或子程序可以单独运行。

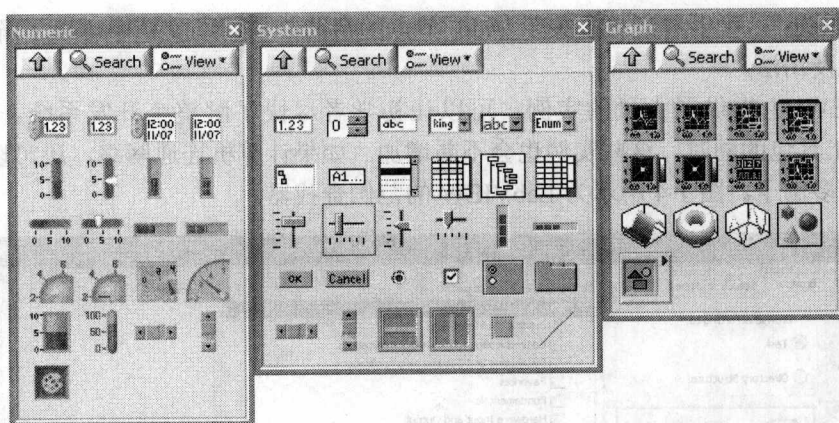


图 1-5 LabVIEW 中的各类控制控件和显示控件

程序的输入在 LabVIEW 中称为控制控件，如按钮、刻度盘、旋钮等；输出则称为显示控件，如 LED 显示器、图形显示器和仪表等。程序运行时，前面板提供所有的输入/输出控件，背面板提供运行控制代码，包括端口、节点、连线和结构等，因此也称为程序框图。前面板和背面板之间的接口则称为端口，前面板的控件必然在背面板对应一个端口；节点带有端口，并能够实现一定的功能，如子函数；结构主要用来控制程序运行，如 C 语言中的 for 循环、While 循环；连线是用于建立起两个端口之间的数据流向。图 1-6 所示为 LabVIEW 的前面板和背面板。

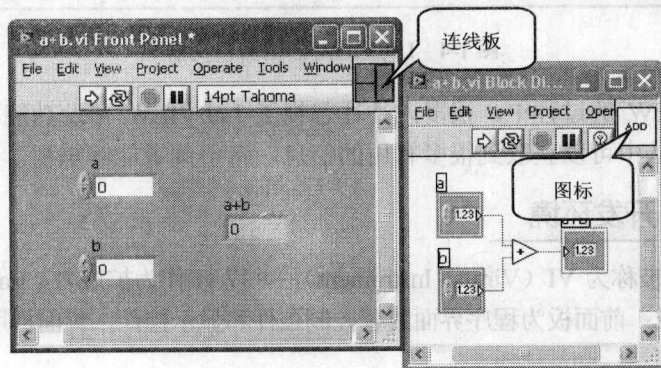


图 1-6 LabVIEW 程序的前面板和背面板

在图形操作系统中，一个代表应用程序的图片称为图标。在 LabVIEW 程序的前面板和背面板的右上角同样可以看到这样的小图标，它也具有同样的功能。当程序被用做子程序时，背面板上将显示子程序图标。连接端口定义了子程序的输入/输出端口，C 语言中也可以见到类似的定义，函数 `int sub(int Input1, int Input2)` 中的 `Input1` 称为参数，也就是函数的接口。图 1-6 中前面板右上角的方框即为连线板。

右击 LabVIEW 前面板空白处，即可打开控件选板，如图 1-7 所示。它提供了前面板上所有的控制控件和显示控件，但控件选板只能在前面板显示。

控制选板包含了表 1-3 所示的子选板。