



普通高等教育电子信息类“十三五”规划教材

# LabVIEW

2016 程序设计教程——

# 从入门到精通

主 编 何小群

副主编 何新军



西安电子科技大学出版社  
<http://www.xduph.com>

普通高等教育电子信息类“十三五”规划教材

# LabVIEW 2016 程序设计教程

## ——从入门到精通

主 编 何小群

副主编 何新军

西安电子科技大学出版社

## 内 容 简 介

本书按照软件开发的设计思想编写,首先介绍了虚拟仪器的基本概念和 LabVIEW 2016 应用软件开发的基础知识,之后详细讲解了 LabVIEW 2016 安装和编程环境、数据类型与基本操作、程序编辑与调试、程序结构设计、图形显示、快速 VI 技术(Express VI)、文件操作、人机界面设计、面向对象编程、数据库操作、网络通信与编程、多线程技术、串口开发与应用、项目管理和应用程序发布以及综合项目实例等内容。本书采用实例讲解的方法介绍每一个知识点,以让读者达到融会贯通的效果。本书结合大量经典案例,深入浅出地讲解 LabVIEW 2016 程序设计思想的重点和难点,使初学者快速具备使用 LabVIEW 集成开发环境设计测量系统的能力。本书具有内容全面翔实、结构合理紧凑、语言浅显易懂以及实用性强等特点。

本书适合 LabVIEW 入门级读者以及相关专业的工程项目开发人员阅读,更适合高等院校物联网工程、自动化、仪器仪表、测量控制等相关专业的应用型本科和高职学生使用。

## 图书在版编目(CIP)数据

LabVIEW 2016 程序设计教程:从入门到精通 / 何小群主编. — 西安:西安电子科技大学出版社,2018.10  
ISBN 978-7-5606-5067-8

I. ① L… II. ① 何… III. ① 软件工具—程序设计—教材 IV. ① TP311.56

中国版本图书馆 CIP 数据核字(2018)第 222657 号

策划编辑 刘玉芳

责任编辑 黄 菡 阎 彬

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2018 年 10 月第 1 版 2018 年 10 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 24.5

字 数 581 千字

印 数 1~3000 册

定 价 56.00 元

ISBN 978-7-5606-5067-8 / TP

**XDUP 5369001-1**

\*\*\*\*\* 如有印装问题可调换 \*\*\*\*\*



# 前 言

LabVIEW 是一种业界领先的工业标准图形化编程工具，可用来开发测试测量、控制系统，是解决工业现场等场合的快速开发原型问题的理想选择。

本书以软件开发设计思想为主线，以由易到难、深入浅出为原则，按照条理清晰、内容全面、实例经典、实用性强的要求，对 LabVIEW 2016 编程进行了全面详细的介绍，尤其对 LabVIEW 编程人员经常讨论的热点问题进行了重点介绍。此外，本书所有的知识点都给出了恰当的实例，读者通过学习这些实例，可以快速掌握很多非常实用的编程技巧，例如图表自动图例、多面板程序设计、人机界面设计等。

在本书的编写过程中，编者几乎参考了 LabVIEW 帮助文档的所有内容以及大部分现有的 LabVIEW 书籍，搜索了 NI 网站中的大量网络资源，并且总结了编者多年的 LabVIEW 编程知识和编程技巧。通过本书，读者可以从入门开始，逐步深入地对 LabVIEW 进行学习，直到成为真正精通 LabVIEW 的编程高手。

本书具有以下特点：

(1) 知识讲解扎实。本书全面详细地介绍了 LabVIEW 的基本概念以及程序开发的基础知识，内容讲解翔实，实例贴切，特别适合从事 LabVIEW 软件设计的初学者使用。

(2) 知识实用性强。本书的全部实例均利用实验室内的环境进行设计与开发，且程序全部经过调试与验证。

(3) 知识覆盖面广。本书精选了若干个典型实例，内容新颖，反映了当前虚拟仪器的发展及时代的需求。

为了使初学者快速具备使用 LabVIEW 设计测试测量系统的能力，全书从实用角度出发，将内容分为 18 个章节进行介绍。

第 1 章 LabVIEW 概述：包括虚拟仪器的结构、特点介绍和 LabVIEW 简介。

第 2 章 LabVIEW 2016 安装：介绍了 LabVIEW 2016 集成开发环境的安装和配置。

第 3 章 LabVIEW 2016 编程环境：详细介绍了 LabVIEW 2016 的编程界面、菜单栏、工具栏、选板和帮助等。

第 4 章 数据类型与基本操作：详细介绍了 LabVIEW 2016 的数据类型和基本运算。

第 5 章 程序编辑与调试：详细介绍了 LabVIEW 2016 VI 的创建、编辑、运行、调试和错误处理等。

第 6 章 程序结构设计：重点介绍了顺序结构、条件结构、循环结构、事件结构、使能结构、定时结构及公式节点和变量等基本知识。

第 7 章 图形显示：重点介绍波形显示、XY 图与强度图形、数字波形图和三维图形的显示等。

第 8 章 快速 VI 技术 (Express VI)：详细介绍了 Express VI 的创建以及示例等。

第 9 章 子 VI 和属性节点：重点介绍了子 VI 的创建和属性节点的应用等。

第 10 章 文件操作：重点介绍了文本文件、电子表格文件、二进制文件、波形文件、配置文件和 XML 文件等基本文件的读写操作技巧。

第 11 章 人机界面设计：重点介绍了下拉列表、列表框、对话框、菜单、选项卡、多面板等高级控件的应用和人机界面设计的基本技巧等。

第 12 章 面向对象编程：重点介绍了对象的创建以及继承、多态的应用等内容。

第 13 章 数据库操作：介绍了通过 LabVIEW 2016 操作 MySQL 数据库，实现数据库的增、删、改、查等基本操作。

第 14 章 网络通信与编程：重点介绍了 TCP 通信、UDP 通信、DataSocket 技术通信和远程访问技术等。

第 15 章 多线程技术：重点介绍多线程的概念、VI 的优先级设置、生产者/消费者结构。

第 16 章 串口开发与应用：重点介绍串口的参数设置、串口通信软件开发等。

第 17 章 项目管理和应用程序发布：重点介绍如何进行大型项目的管理和应用程序的发布等。

第 18 章 综合项目实例：以双通道频谱滤波器设计为例，按照软件工程的思想，对项目设计、开发到发布的整个过程进行详细讲解。

本书在内容安排上循序渐进、深入浅出，力求重点突出，面向实际应用，提高读者的编程能力和解决实际问题的能力。

本书由何小群（重庆工程学院）任主编、何新军（中冶赛迪技术研究中心有限公司）任副主编，其中何新军高级工程师负责第 16 章、第 17 章的编写，其余章节由何小群老师编写。本书的编写得到了重庆工程学院电子与物联网学院全体老师的鼓励和支持，在此向他们表示衷心感谢！

由于编者水平有限，时间仓促，书中不当之处在所难免，敬请读者批评指正。

编者

2018 年 6 月

# 目 录

第 1 章 LabVIEW 概述.....	1	3.8 LabVIEW 应用开发实例.....	34
1.1 虚拟仪器概述.....	1	思考与练习.....	36
1.1.1 虚拟仪器的结构.....	1	第 4 章 数据类型与基本操作.....	37
1.1.2 虚拟仪器的特点.....	2	4.1 基本数据类型.....	37
1.1.3 虚拟仪器的硬件.....	3	4.1.1 数值型.....	37
1.1.4 虚拟仪器的软件.....	3	4.1.2 布尔型.....	43
1.2 LabVIEW 基本介绍和发展历史.....	5	4.1.3 枚举类型.....	46
1.2.1 什么是 LabVIEW.....	5	4.1.4 时间类型.....	48
1.2.2 LabVIEW 的应用领域.....	6	4.1.5 变体类型.....	50
1.2.3 LabVIEW 的优势.....	7	4.2 数据运算.....	51
1.2.4 LabVIEW 的起源与发展历程.....	8	4.2.1 数值函数选板.....	51
1.2.5 LabVIEW 2016 的新增特性.....	8	4.2.2 布尔函数选板.....	53
思考与练习.....	10	4.2.3 比较函数选板.....	54
第 2 章 LabVIEW 2016 安装.....	11	4.2.4 表达式节点.....	55
2.1 计算机性能要求.....	11	4.3 数组.....	56
2.2 LabVIEW 2016 的安装.....	11	4.3.1 创建一维数组.....	56
第 3 章 LabVIEW 2016 编程环境.....	18	4.3.2 创建多维数组.....	57
3.1 LabVIEW 2016 启动.....	18	4.3.3 循环结构创建数组.....	58
3.2 LabVIEW 2016 编程界面.....	19	4.3.4 数组函数.....	59
3.3 LabVIEW 2016 菜单栏.....	21	4.4 字符串.....	64
3.4 LabVIEW 2016 工具栏.....	25	4.4.1 字符串与路径.....	64
3.5 LabVIEW 2016 选板.....	27	4.4.2 字符串数组控件.....	67
3.5.1 “控件”选板.....	27	4.4.3 字符串函数.....	68
3.5.2 “函数”选板.....	28	4.5 簇.....	76
3.5.3 “工具”选板.....	29	4.5.1 簇的创建.....	77
3.6 LabVIEW 2016 帮助.....	31	4.5.2 簇函数.....	78
3.6.1 显示“即时帮助”.....	31	4.6 矩阵.....	81
3.6.2 LabVIEW 帮助.....	31	4.6.1 矩阵概念.....	82
3.6.3 查找范例.....	32	4.6.2 矩阵转置.....	83
3.6.4 网络资源.....	33	4.6.3 矩阵的基本运算.....	84
3.7 自定义编程环境.....	33	4.7 波形.....	85
3.7.1 定制控件和函数选板.....	33	4.7.1 波形数据.....	85
3.7.2 环境参数设置.....	34	4.7.2 波形操作函数.....	86

思考与练习 .....	88	6.3.2 While 循环 .....	124
<b>第 5 章 程序编辑与调试</b> .....	<b>90</b>	6.3.3 循环结构数据通道与自动索引 .....	125
5.1 VI 的创建 .....	90	6.3.4 移位寄存器 .....	126
5.1.1 前面板的创建 .....	90	6.3.5 反馈节点 .....	127
5.1.2 程序框图的创建 .....	91	6.3.6 循环结构应用实例 .....	128
5.1.3 图标的创建 .....	92	6.4 公式节点 .....	130
5.2 VI 的编辑 .....	93	6.4.1 公式节点变量 .....	130
5.2.1 选择对象 .....	93	6.4.2 公式节点运算符和函数 .....	131
5.2.2 移动对象 .....	94	6.5 事件结构 .....	132
5.2.3 复制和删除对象 .....	94	6.5.1 事件选择器标签 .....	133
5.2.4 对齐和分布对象 .....	95	6.5.2 事件数据节点 .....	134
5.2.5 调整对象大小 .....	95	6.5.3 事件结构的设置 .....	135
5.2.6 重新排序 .....	96	6.5.4 在 LabVIEW 中使用事件 .....	138
5.2.7 修改对象外观 .....	97	6.6 使能结构 .....	140
5.2.8 连线 .....	98	6.6.1 程序框图禁用结构 .....	140
5.3 VI 的运行和调试 .....	99	6.6.2 条件禁用结构 .....	141
5.3.1 运行调试工具 .....	100	6.7 变量 .....	143
5.3.2 高亮显示执行过程 .....	100	6.7.1 局部变量 .....	144
5.3.3 探针与断点诊断 .....	102	6.7.2 全局变量 .....	147
5.3.4 程序调试技巧 .....	105	6.8 定时结构 .....	150
5.4 错误处理 .....	106	6.8.1 定时循环 .....	151
5.4.1 错误簇 .....	107	6.8.2 定时顺序 .....	153
5.4.2 错误捕获 .....	109	6.8.3 定时 VI .....	153
5.4.3 错误报告 .....	111	思考与练习 .....	154
5.4.4 错误处理 .....	112	<b>第 7 章 图形显示</b> .....	<b>156</b>
思考与练习 .....	114	7.1 波形显示 .....	156
<b>第 6 章 程序结构设计</b> .....	<b>115</b>	7.1.1 波形图 .....	156
6.1 顺序结构 .....	115	7.1.2 波形图显示实例 .....	162
6.1.1 平铺式顺序结构 .....	116	7.1.3 波形图表 .....	165
6.1.2 层叠式顺序结构 .....	117	7.1.4 波形图表实例 .....	168
6.1.3 顺序结构之间的转换 .....	117	7.2 XY 图与 Express XY 图 .....	169
6.2 条件结构 .....	119	7.2.1 XY 图 .....	170
6.2.1 条件选择器标签 .....	119	7.2.2 绘制简单利萨如图 .....	170
6.2.2 分支选择器 .....	121	7.2.3 XY 图实例 .....	171
6.2.3 分支子程序框 .....	122	7.2.4 Express XY 图 .....	173
6.3 循环结构 .....	123	7.3 强度图形 .....	175
6.3.1 For 循环 .....	123	7.3.1 强度图 .....	175

7.3.2 强度图表.....	176	10.7 测量文件.....	225
7.4 数字波形图.....	177	10.8 配置文件.....	227
7.5 三维图形.....	179	10.9 XML 文件.....	229
7.5.1 三维曲面图.....	180	10.10 TDMS 文件.....	231
7.5.2 三维参数图.....	181	思考与练习.....	235
7.5.3 三维曲线图.....	183	<b>第 11 章 人机界面设计.....</b>	<b>236</b>
7.6 图形控件.....	184	11.1 下拉列表控件和枚举控件.....	236
7.6.1 极坐标图.....	185	11.2 列表框控件.....	238
7.6.2 最小-最大曲线显示控件.....	186	11.2.1 单列列表框.....	238
思考与练习.....	188	11.2.2 多列列表框.....	241
<b>第 8 章 快速 VI 技术(Express VI).....</b>	<b>189</b>	11.3 表格与树形控件.....	242
8.1 Express VI 简介.....	189	11.3.1 表格.....	243
8.2 Express VI 的创建.....	190	11.3.2 树形控件.....	244
8.3 波形分析示例.....	193	11.4 VI 属性设置.....	247
8.4 声音录制播放 Express VI 示例.....	194	11.5 对话框.....	252
8.5 弹出信息录入框 Express VI 示例.....	195	11.5.1 普通对话框.....	252
思考与练习.....	196	11.5.2 自定义对话框.....	254
<b>第 9 章 子 VI 和属性节点.....</b>	<b>197</b>	11.6 菜单.....	255
9.1 子 VI.....	197	11.6.1 菜单函数.....	255
9.1.1 创建子 VI.....	197	11.6.2 运行主菜单.....	256
9.1.2 定义子 VI 属性.....	199	11.6.3 右键快捷菜单.....	261
9.2 属性节点.....	202	11.7 选项卡.....	263
9.3 调用节点和引用句柄.....	205	11.8 多面板.....	264
9.4 属性节点应用实例.....	207	11.9 光标.....	266
9.4.1 进度条.....	207	11.10 自定义控件和数据类型.....	268
9.4.2 单选按钮.....	208	11.10.1 自定义控件.....	268
思考与练习.....	209	11.10.2 自定义数据类型.....	269
<b>第 10 章 文件操作.....</b>	<b>211</b>	11.10.3 自定义控件选板和函数选板.....	270
10.1 文件操作基本术语.....	211	11.11 用户界面设计.....	271
10.1.1 文件的基本概念.....	211	11.11.1 修饰静态界面.....	271
10.1.2 文件的基本类型.....	212	11.11.2 动态交互界面.....	272
10.1.3 文件操作基本函数.....	213	11.12 VI 程序设计规则.....	272
10.2 文本文件.....	215	11.12.1 关于前面板的设计.....	273
10.3 电子表格文件.....	217	11.12.2 关于程序框图的设计.....	275
10.4 二进制文件.....	219	11.12.3 关于 VI.....	275
10.5 波形文件.....	221	思考与练习.....	276
10.6 数据记录文件.....	223	<b>第 12 章 面向对象编程.....</b>	<b>278</b>



12.1 面向对象的基本概念 .....	278	14.1.5 蓝牙技术.....	321
12.2 类的创建 .....	279	14.2 DataSocket 技术通信.....	323
12.3 对象的创建 .....	282	14.2.1 DataSocket 服务管理器.....	323
12.4 继承 .....	283	14.2.2 DataSocket 服务器 .....	325
12.5 多态 .....	285	14.2.3 DataSocket 应用程序接口.....	325
12.6 动态方法 .....	287	14.3 远程访问 .....	328
12.7 LabVIEW 面向对象其他特点 .....	289	14.3.1 配置服务器.....	328
12.7.1 构造函数与析构函数.....	289	14.3.2 远程面板控制.....	329
12.7.2 对象数据文件操作.....	290	14.3.3 浏览器访问.....	330
12.8 两种编程方法的比较 .....	290	<b>第 15 章 多线程技术 .....</b>	<b>332</b>
12.8.1 测试目的.....	291	15.1 基本概念 .....	332
12.8.2 面向过程的方法.....	291	15.2 VI 的优先级设置 .....	332
12.8.3 面向对象的方法.....	292	15.3 生产者/消费者结构 .....	333
12.8.4 两种方法的比较.....	293	15.3.1 队列函数.....	333
<b>第 13 章 数据库操作 .....</b>	<b>294</b>	15.3.2 事件型生产者/消费者结构.....	335
13.1 LabSQL .....	294	15.3.3 数据型生产者/消费者结构.....	336
13.1.1 LabSQL 简介 .....	294	<b>第 16 章 串口开发与应用 .....</b>	<b>337</b>
13.1.2 LabSQL 的安装.....	295	16.1 基本概念 .....	337
13.2 数据源配置 .....	296	16.2 串口的参数设置 .....	337
13.2.1 ODBC 简介.....	296	16.3 串口通信软件开发 .....	338
13.2.2 ADO 简介 .....	297	16.3.1 串口通信函数.....	338
13.2.3 建立数据源.....	298	16.3.2 多线程串口通信软件设计 .....	339
13.3 MySQL 基础 .....	299	<b>第 17 章 项目管理和应用程序发布.....</b>	<b>343</b>
13.3.1 MySQL 下载与安装.....	300	17.1 创建项目 .....	343
13.3.2 启动 MySQL 服务.....	303	17.2 对项目中的项进行排序.....	344
13.3.3 MySQL 常用图形管理工具.....	304	17.3 项目库 .....	345
13.4 数据库操作 .....	305	17.4 发布应用程序 .....	345
13.4.1 查询记录.....	305	17.4.1 生成应用程序(EXE).....	346
13.4.2 添加记录.....	306	17.4.2 生成安装程序.....	349
13.4.3 删除记录.....	307	<b>第 18 章 综合项目实例 .....</b>	<b>351</b>
13.4.4 修改记录.....	308	18.1 软件工程基础 .....	351
<b>第 14 章 网络通信与编程 .....</b>	<b>310</b>	18.1.1 软件工程概念.....	351
14.1 网络协议通信 .....	310	18.1.2 软件生命周期.....	352
14.1.1 TCP 通信 .....	311	18.1.3 软件开发模型.....	353
14.1.2 UDP 通信.....	316	18.1.4 软件需求分析.....	357
14.1.3 SMTP Email.....	318	18.1.5 软件设计.....	359
14.1.4 IrDA 技术 .....	320	18.1.6 程序编码.....	360

18.1.7	软件测试.....	362	18.4.5	各 VI 控件之间数据流的连线 .....	371
18.1.8	软件实施和维护.....	363	18.4.6	各 VI 信号的图形显示控件 .....	372
18.2	项目设计目的 .....	363	18.4.7	创建上下截止频率调节控件 .....	372
18.3	项目功能模块介绍 .....	364	18.4.8	信号掩区和边界测试输出 信号的显示设置.....	373
18.4	项目功能设计实现 .....	366	18.4.9	创建截止频率变化子 VI.....	376
18.4.1	滤波信号的产生.....	366	18.5	发布应用程序 .....	377
18.4.2	添加滤波器 VI 控件.....	367	18.5.1	独立可执行程序(EXE).....	377
18.4.3	双通道频谱测量.....	368	18.5.2	安装程序.....	380
18.4.4	检测信号.....	369			

# 第1章 LabVIEW 概述

本章首先从虚拟仪器引出 LabVIEW 软件，然后介绍 LabVIEW 的应用、优势、起源与发展历程，最后介绍 LabVIEW 2016 的新增特性。

## 1.1 虚拟仪器概述

虚拟仪器是美国国家仪器公司 (National Instruments, NI)1986 年提出的虚拟测量仪器 (VI)概念，是现代计算机技术和仪器技术深层次结合的产物，是计算机辅助测试领域的一项重要技术。虚拟仪器引发了传统仪器领域的一场重大变革，使得计算机和网络技术得以长驱直入仪器领域。虚拟仪器和仪器技术结合起来，开创了“软件即是仪器”的先河。

### 1.1.1 虚拟仪器的结构

虚拟仪器是基于计算机的仪器，是计算机与仪器密切结合的最终产物。粗略地说，这种结合有两种方式：一种是将计算机装入仪器，其典型的例子就是智能化仪器；另一种方式是将仪器装入计算机，以通用的计算机硬件和操作系统为依托，实现各种仪器功能。虚拟仪器主要指第二种方式。从实质上讲，虚拟仪器就是利用硬件系统完成信号的采集、测量与调理，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，利用计算机的显示器模拟传统仪器的控制面板，以多种形式输出检测结果，从而完成所选的各种测试功能的仪器。

虚拟仪器可以通过多种接口(GPIB、VXI、PXI 等)或具有这些接口的仪器来连接被测对象和计算机。虚拟仪器的机构如图 1.1 所示。



图 1.1 虚拟仪器机构图

虚拟仪器系统包括仪器硬件和应用软件两大部分。仪器硬件是计算机的外围电路，与计算机一起构成了虚拟仪器系统的硬件环境，是应用软件的基础；应用软件则是虚拟仪器的核心，在基本硬件确定后，软件通过不同功能模块即软件模块的组合构成多种仪器，赋予系统特有的功能，以实现不同的测量功能。

虚拟仪器的“虚拟”有以下两个层面的含义：

(1) 虚拟的控制面板。传统仪器通过设置在面板上的各种“控件”来完成操作和功能，如各种开关、按键、滑动调节杆、显示器等，实现仪器电源的“通”或“断”，被测信号的“输入通道”、“放大倍数”、“滤波特性”等参数设置，测量结果的“数值显示”、“波形显示”等。

传统仪器面板上的“控件”都是实物，而且是用手动或触摸等方式进行操作的，而虚拟仪器面板上的各种“控件”都是虚构的，它们的外形是与传统仪器控件的实物外形相像的图标，实际功能是通过相应的软件程序来实现的。

(2) 虚拟的测量、测试与分析。传统的仪器通过设计具体的模拟或数字电路来实现仪器的测量、测试及分析功能，而虚拟仪器则利用软件程序实现这些功能。

可见，虚拟仪器是由计算机硬件资源、模块化仪器硬件以及用于数据分析、过程通信和图形用户界面的软件组成的测控系统，是一种由计算机操纵的模块化仪器系统。

### 1.1.2 虚拟仪器的特点

虚拟仪器技术利用高性能的模块化硬件，结合高效灵活的软件来完成各种测试测量和自动化的应用。灵活高效的软件能创建完全自定义的用户界面，模块化的硬件能方便地提供全方位的系统集成，标准的软/硬件平台能满足对同步和定时应用的需求。与传统仪器相比，虚拟仪器具有以下四个特点。

#### 1. 性能高

虚拟仪器技术是在 PC 技术的基础上发展起来的，因此完全继承了以现成即用的 PC 技术为主导的最新商业技术的优点，包括功能超卓的处理器和文件 I/O，使用户在数据高速导入系统的同时就能实时地进行复杂的分析。此外，不断发展的因特网和越来越快的计算机网络使得虚拟仪器技术展现出更强大的优势。

#### 2. 扩展性强

虚拟仪器应用软件灵活的扩展性，使得工程师和科学家们在应用日益发展的软硬件工具过程中，不再局限于当前的技术，只需要在已有的应用工程中更新计算机或测量硬件，就能以最少的硬件投资和极少甚至无需软件上的升级改进整个系统。

#### 3. 开发时间少

虚拟仪器在驱动和应用两个层面上，直接把目前高效的软件架构与计算机、仪器仪表和通信方面的最新技术通过软件的形式结合在一起，大大降低了驱动和应用两个层面的开发调试过程，同时还具有强大的灵活性，使得用户能轻松地配置、创建、分布、维护和修改高性能及低成本的测量和控制解决方案。

#### 4. 无缝集成

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,而连接和集成这些不同设备总是要耗费大量的时间。虚拟仪器软件平台为所有的 I/O 设备提供了标准的接口,帮助用户轻松地将多个测量设备集成到单个系统,减少了任务的复杂性。

### 1.1.3 虚拟仪器的硬件

随着测试测量应用的日益复杂,目前市场上提供的模块化硬件产品也非常丰富,比如总线类型支持 PCI、PXI、PCMCIA、USB 和 1394 总线等,产品种类从数据采集、信号调理、声音和振动测量、视觉、运动、仪器控制、分布式 I/O 到 CAN 接口工业通信等。

按照硬件接口的不同,虚拟仪器可分为基于 PC 总线、GPIB、VXI 总线和 PXI 总线的四种标准体系结构。

#### 1. 基于 PC 总线的虚拟仪器

由于个人计算机的用户量大、通用性强,基于 PC 总线的虚拟仪器成为人们的首选。这种硬件一般采用基于 PC 总线的通用 DAQ(数据采集卡),主要的 PC 总线有 ISA、PCI、PC/104 等。这类虚拟仪器充分利用了计算机的资源,大大增加了测试系统的灵活性和扩展性。利用通用型 DAQ 可方便快捷地组建基于计算机的仪器,易于实现一机多型和一机多用。随着 A/D 转换技术、精密放大技术、滤波技术与数字信号调制技术等迅速发展,DAQ 的采样速率已达到 2 Gb/s,精度高达 24 位,通道数高达 64 个,并能任意组合数字 I/O、模拟 I/O、计数器/定时器等通道,大大扩展了仪器的功能。

#### 2. 基于 GPIB 的虚拟仪器

已有的专业仪器多数配有 GPIB(General Purpose Interface Bus,通用接口总线),所以利用此类仪器构建基于计算机的虚拟仪器一般利用 GPIB 实现。基于 GPIB 的虚拟仪器充分利用了现有条件,实现测量、检测等功能;但其数据传输速率一般低于 500 kb/s,不适合对系统速度要求较高的应用。

#### 3. 基于 VXI 总线的虚拟仪器

VXI 系统最多可包含 256 个装置,主要由主机箱、控制器、具有多种功能的模块仪器和驱动软件、系统应用软件等组成,具有即插即用的特性,所以系统中各功能模块可随意更换构成新系统。基于 VXI 总线的虚拟仪器具有模块化、系统化、通用化以及 VXI 仪器的互换性和互操作性的特征。VXI 的价格相对较高,适合于尖端的测试领域。

#### 4. 基于 PXI 总线的虚拟仪器

PXI(PCI eXtension for Instrumentation)总线整合了台式计算机的高速 PCI 总线的 80 Mb/s 优势,借鉴于 VXI 总线中先进的仪器技术,如同步触发、板间总线、星形触发总线、板载时钟等特性,兼容 CompactPCI 机械规范,并增加了主动冷却、环境测试(温度、湿度、振动和冲击试验)等要求。

### 1.1.4 虚拟仪器的软件

虚拟仪器框架从底层到顶层,由 VISA(Virtual Instrumentation Software Architecture)库、

仪器驱动程序、应用软件三部分组成。

### 1. VISA 库

VISA 库即虚拟仪器软件体系结构库,实质上就是标准 I/O 函数库及相关规范的总称,一般将该 I/O 函数库称为 VISA 库。VISA 库驻留于计算机系统中,执行仪器总线的特殊功能,起着连接计算机与仪器的作用,以实现对该仪器的程序控制。

#### 1) VISA 库的作用

对于虚拟仪器驱动程序的开发编程者来说,VISA 库是一个可调用的操作函数集。作为标准化的 I/O 接口软件规范,VISA 库的作用有以下四点:

- (1) 为所有使用者提供统一的软件编程基础,对驱动程序、应用程序不必考虑接口,均可使用。
- (2) 仅规定为用户提供标准函数,不对具体实现作任何说明。
- (3) 用于编写符合 VPP 规范(VXI Plug & Play, VXI 总线即插即用型驱动器)的仪器驱动程序,完成计算机与仪器之间的命令和数据传输,实现对仪器的控制。
- (4) VISA 库作为底层 I/O 接口软件,运用于计算机系统中。

#### 2) VISA 库的特点

VISA 库的主要特点有以下四点:

- (1) 适用于各类仪器,如 VXI、PXI、GPIB、RS-232 和 USB 仪器等。
- (2) 与硬件接口无关。
- (3) 既适用于单处理器结构,又适合于多处理器或分布式结构。
- (4) 适用于各种网络机制。

VISA 只解决了仪器接口的可互换性(即改变接口或总线方式不必修改测试程序),但并没有解决更高层次的针对不同仪器的可互换性。

### 2. 仪器驱动程序

所谓仪器驱动程序,是指能实现某一仪器系统控制与通信的软件程序集,是应用程序实现仪器控制的桥梁。仪器的驱动程序由仪器生产商以源码形式提供给用户使用,每个仪器模块都有自己的仪器驱动程序。

常用的虚拟仪器设计软件集成了大量常用仪器的驱动程序,以方便编程者使用。经常使用的测量仪器有几十大类、上万种型号,各种仪器的驱动程序都不相同,为使同类功能的仪器可以互换而不修改测试软件,即实现仪器的可互换性,世界各大仪器公司都在为仪器驱动程序研究和制定统一的标准及规范而努力。

仪器驱动程序又称为驱动器。目前广泛使用的驱动器规范有 VPP 规范和 IVI(Interchangeable Virtual Instruments, 互换型驱动器)规范两种。

### 3. 应用软件

应用软件是直接面向操作用户的程序。这类软件建立在仪器驱动程序之上,通过提供的测控操作界面、丰富的数据分析与处理功能等完成自动测试任务;尤其是通用数字处理软件,集中体现了虚拟仪器的优点。

虚拟仪器应用软件的开发工具包括通用编程软件和专业图形化编程软件两类。

(1) 通用编程软件。这类软件主要有 Microsoft 公司的 Visual Basic 与 Visual C++、Borland 公司的 Delphi、Sybase 公司的 PowerBuilder 等。这类软件功能强大,但不是专门为虚拟仪器而设计的,因此利用通用编程软件开发虚拟仪器,需要开发者具备较高的软件编程技术,同时对虚拟仪器技术也相当了解。

(2) 专业图形化编程软件。这类软件主要有 HP 公司的 VEE、NI 公司的 LabVIEW 和 Lab Windows/CVI 等。这类软件专门用于虚拟仪器的开发,对开发者的要求较低,只要了解软件的总体功能以及所要设计的虚拟仪器的功能就可快捷方便地进行开发。

## 1.2 LabVIEW 基本介绍和发展历史

LabVIEW 是一种程序开发环境,由美国国家仪器(NI)公司研制开发,类似于 C 和 BASIC 开发环境,但是 LabVIEW 与其他计算机语言的显著区别是:其他计算机语言都是采用基于文本的语言产生代码,而 LabVIEW 使用图形化编辑语言 G 编写程序,产生的程序是框图的形式。LabVIEW 软件是 NI 设计平台的核心,也是开发测量或控制系统的理想选择。LabVIEW 开发环境集成了工程师和科学家快速构建各种应用所需的所有工具,旨在帮助工程师和科学家解决问题,提高生产力和不断创新。

### 1.2.1 什么是 LabVIEW

LabVIEW(Laboratory Virtual Instrument Engineering Workbench)是一种图形化的编程语言的开发环境。传统文本编程语言根据语句和指令的先后顺序决定程序执行顺序,而 LabVIEW 则采用数据流编程方式,程序框图中节点之间的数据流向决定了程序的执行顺序。它用图标表示函数,用连线表示数据流向。

与 C 和 BASIC 一样,LabVIEW 也是通用的编程系统,有一个可以完成任何编程任务的庞大函数库。LabVIEW 函数库包括数据采集、GPIB、串口控制、数据分析、数据显示及数据存储等函数。LabVIEW 也有传统的程序调试工具,如设置断点、以动画方式显示数据及其子程序(子 VI)的结果、单步执行等,便于程序的调试。

LabVIEW 提供很多外观与传统仪器(如示波器、万用表)类似的控件,可用来方便地创建用户界面。用户界面在 LabVIEW 中被称为前面板,使用图标和连线,可以通过编程对前面板上的对象进行控制。LabVIEW 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,它集成了满足 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。这是一个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生动有趣。

图形化的程序语言,又称为 G 语言。使用这种语言编程时,基本上不写程序代码,取而代之的是流程图或框图。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。

## 1.2.2 LabVIEW 的应用领域

LabVIEW 有很多优点，比较适合电子类工程师进行快速的程序设计和开发，尤其是在某些特殊领域其特点尤其突出。

### 1. 测试测量

LabVIEW 最初就是为测试测量而设计的，因而测试测量也就是现在 LabVIEW 应用最广泛的领域。经过多年的发展，LabVIEW 在测试测量领域获得了广泛的认可。至今，大多数主流的测试仪器、数据采集设备都拥有专门的 LabVIEW 驱动程序，使用 LabVIEW 可以非常便捷地控制这些硬件设备。同时，用户也可以十分方便地找到各种适用于测试测量领域的 LabVIEW 工具包，这些工具包几乎覆盖了用户所需的所有功能，用户在这些工具包的基础上再开发程序就容易多了，有时甚至只需简单地调用几个工具包中的函数，就可以组成一个完整的测试测量应用程序。

### 2. 控制

控制与测试是两个相关度非常高的领域，从测试领域起家的 LabVIEW 自然而然地首先拓展至控制领域。LabVIEW 拥有专门用于控制领域的模块——LabVIEW DSC，除此之外，工业控制领域常用的设备、数据线等通常也都带有相应的 LabVIEW 驱动程序。使用 LabVIEW 可以非常方便地编制各种控制程序。

### 3. 仿真

LabVIEW 包含了多种多样的数学运算函数，特别适合进行模拟、仿真、原型设计等工作。在设计机电设备之前，可以先在计算机上用 LabVIEW 搭建仿真原型，验证设计的合理性，找到潜在的问题。在高等教育领域，有时使用 LabVIEW 进行软件模拟，可以使学生实践所学的知识技能。

### 4. 儿童教育

由于图形外观漂亮且容易吸引儿童的注意力，同时图形比文本更容易被儿童接受和理解，所以 LabVIEW 非常受少年儿童的欢迎。对于没有任何计算机知识的儿童而言，可以把 LabVIEW 理解成是一种特殊的“积木”：把不同的元件搭在一起，就可以实现自己所需的功能。著名的可编程玩具乐高积木使用的就是 LabVIEW 编程语言，儿童经过短暂的指导就可以利用乐高积木提供的元件搭建各种车辆模型、机器人等，再使用 LabVIEW 编写控制其运动和行为的程序。除了应用于玩具，LabVIEW 还有专门用于中小学生的版本。

### 5. 快速开发

完成一个功能类似的大型应用软件，熟练的 LabVIEW 程序员所需的开发时间大概只是熟练的 C 程序员所需时间的 1/5 左右；所以，如果项目开发时间紧张，应该优先考虑使用 LabVIEW，以缩短开发时间。

### 6. 跨平台

如果同一个程序需要运行于多个硬件设备之上，也可以优先考虑使用 LabVIEW。LabVIEW 具有良好的平台一致性，LabVIEW 的代码不需任何修改就可以运行在常见的三



大台式机操作系统(Windows、Mac OS 及 Linux)上。除此之外, LabVIEW 还支持各种实时操作系统和嵌入式设备, 比如常见的 PDA、FPGA 以及运行 VxWorks 和 PharLap 系统的 RT 设备。

### 1.2.3 LabVIEW 的优势

LabVIEW 是专为工程师和科研人员设计的集成式开发环境。LabVIEW 的本质是一种图形化编程语言(G), 采用的是数据流模型, 而不是顺序文本代码行, 使用户能够根据自己的思路以可视化的布局编写功能代码, 这意味着用户可以减少花在语句和语法上的时间, 而将更多的时间花在解决重要的问题上。通常, 使用 LabVIEW 开发应用系统的速度比使用其他编程语言快 4~10 倍, 这一惊人速度背后的原因在于 LabVIEW 易学易用, 它所提供的工具使创建测试和测量应用变得更为轻松。

LabVIEW 的优势主要体现在以下几个方面:

- (1) 简化开发。使用直观的图形化编程语言, 按照工程师脑中所想编写代码。
- (2) 无可比拟的硬件集成。可以采集任意总线上的任意测量硬件数据。
- (3) 自定义用户界面。使用易于拖放的控件快速开发用户界面, 可视化开发程序。
- (4) 广泛的分析和信号处理 IP 开发数据分析和高级控制算法。
- (5) 为解决方案的各个组成部分选择合适的方法。在单个开发环境中集成图形化编程、基于文本的编程以及其他编程方法, 帮助用户高效构建自定义软件解决方案。
- (6) 部署软件至正确的硬件。无论是桌面 PC、工业计算机, 还是嵌入式设备, 均可将 LabVIEW 代码部署至正确的硬件, 无需根据不同的终端重新编写代码。
- (7) 强大的多线程执行。通过固有的并行软件自动利用多核处理器提供的性能优势。
- (8) 记录和共享测量数据。以任意文件类型或报表格式保存、显示和共享测量结果。
- (9) 与 Microsoft 的 Excel 和 Word 交互。能够将报表直接发送到 Microsoft 应用程序 (Excel 和 Word)。可采用 ActiveX 或用于 Microsoft Office 的 NI LabVIEW 报告生成工具包, 通过编程加以实现。报告生成工具包抽象化与 Excel 和 Word 交互的复杂性而且让我们能着力设计实际报告元素。使用这些 VI, 能轻松地将标题、表格和图形添加至 Microsoft 文档。还有, 能在可接受 LabVIEW 调用的 Word 和 Excel 中创建模板, 实现更加自动化和标准化的报告功能。
- (10) 关注数据, 而非文件。LabVIEW 存储、管理与报告工具的设计便于抽象化细节以及文件 I/O 与报告的生成, 从而帮助用户关注数据的采集。借助针对工程数据的 TDMS 文件格式、针对传统文件的 DataPlugins、用于搜索的 NI DataFinder 和强大的报告工具, 用户不必根据存储和报告的局限限制采集。当硬件速度加快而且存储更廉价时, LabVIEW 继续提供工具来帮助用户从收集的全部数据中获得最大利益。
- (11) 充分利用团队的专业性。LabVIEW 系统设计软件为用户提供了重要的 I/O 访问、有用的 IP 和易用的图形化编程。它还可与第三方软件程序和函数库进行互操作, 帮助用户重复利用工程团队已经开发好的代码以及使用其擅长的软件语言。LabVIEW 可集成多种语言, 简化与本地运行或在网络上运行的其他软件的通信, 因而可帮助工程师利用所有可用的工具成功地进行开发。