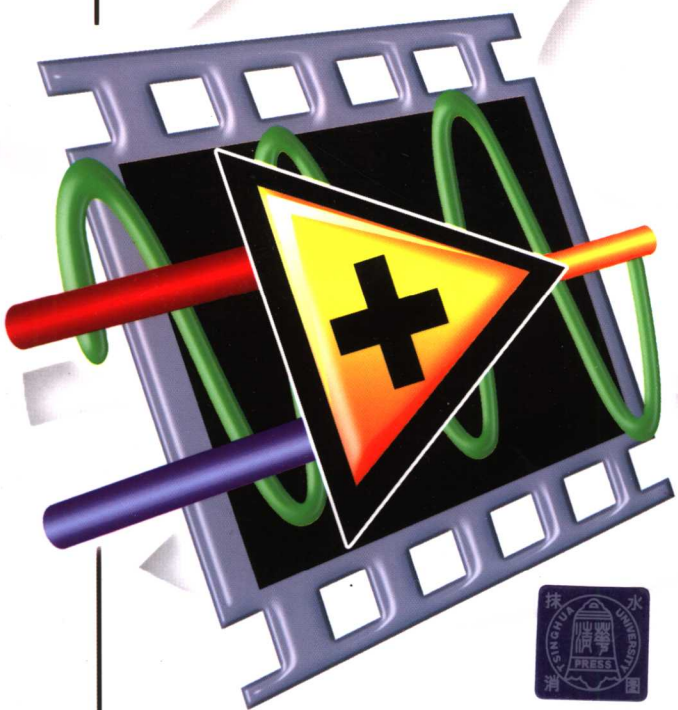
LabVIEW 8.20 程序设计从入门到精通

陈锡辉　张银鸿　编著



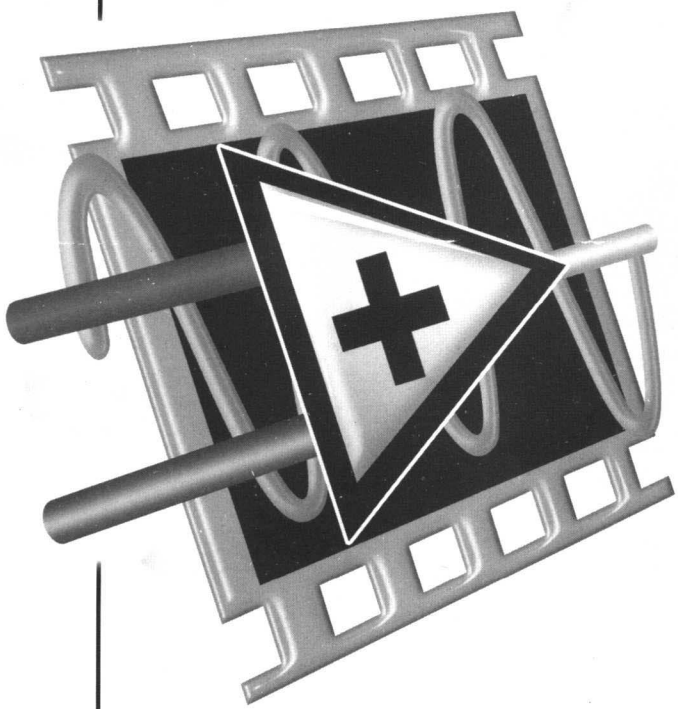
附光盘

本书实例代码

清华大学出版社

LabVIEW 8.20 程序设计从入门到精通

陈锡辉　张银鸿　编著



清华大学出版社
北京

内 容 简 介

本书全面介绍了 LabVIEW 8.20 专业开发版中的各种编程知识与技巧。基础篇循序渐进地介绍了 LabVIEW 程序设计所需的基础知识，包括 LabVIEW 编程环境、数据操作、程序结构、复杂数据类型、图表图形、Express VI、文件 I/O、子 VI、属性节点与方法节点、人机界面交互设计、数学分析与信号处理、数据采集与仪器控制。高级篇针对 LabVIEW 高级编程人员深入浅出地介绍了各种 LabVIEW 高级编程知识与技巧，包括管理与开发 LabVIEW 大型项目、LabVIEW 中的面向对象编程、动态程序控制技术、LabVIEW 外部程序接口、访问数据库、网络编程、定时结构与同步技术、LabVIEW 程序设计优化、发布应用程序。本书语言生动精炼、内容详尽，并且包含了大量实用的技巧实例。

本书可作为高等院校虚拟仪器及相关课程的教材或教学参考书，也可供从事相关工作的工程师或科研人员学习或参考。

图书在版编目（CIP）数据

LabVIEW 8.20 程序设计从入门到精通 / 陈锡辉，张银鸿编著. —北京：清华大学出版社，2007.7
ISBN 978-7-302-15230-9

Ⅰ. L…　Ⅱ. ①陈… ②张…　Ⅲ. 软件工具，LabVIEW 8.0－程序设计　Ⅳ. TP311.56

中国版本图书馆 CIP 数据核字（2007）第 071272 号

责任编辑：夏兆彦　王冰飞
责任校对：张　剑
责任印制：孟凡玉
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　**邮　　编**：100084
c－service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 刷 者：北京鑫丰华彩印有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：203×260　**印　张**：25.25　**字　数**：662 千字
含光盘 1 张
版　　次：2007 年 7 月第 1 版　　**印　　次**：2007 年 9 月第2 次印刷
印　　数：4001～7000
定　　价：49.00 元

陈锡辉　中国科学院高能物理研究所博士。目前从事北京正负电子对撞机二期改造工程中慢控制系统的总体框架与软件设计工作。对数据采集与仪器控制相关的软件编程工作特别偏爱。有四年 LabVIEW 实际开发经验，编程功底深厚，成功开发过多个基于 LabVIEW 的大中型项目。

FOREWORD

前言

LabVIEW 是一种业界领先的工业标准图形化编程工具，主要用于开发测试、测量与控制系统。它是专门为工程师和科学家而设计的直观图形化编程语言。它将软件和各种不同的测量仪器硬件及计算机集成在一起，建立虚拟仪器系统，以形成用户自定义的解决方案。经过 20 年的演变和改进，在基于 PC 的测量自动化领域，LabVIEW 确立了其主导地位，并成为业界的事实标准，同时也给传统的教学研究带来了巨大的变化。一个基于计算机的自动化实验室能大大提高研究人员的工作效率并改进学生的学习方式。利用 NI 的虚拟仪器技术，让以往复杂的数据采集工作变得异常简单，老师和同学都可以集中时间和精力用于实验的执行、数据的分析及结论的总结上，而不用将大量的时间花费在实验系统设备的搭建中。在国外，虚拟仪器技术已经是很多大学院校独立开设的一门课程，相信在未来几年内虚拟仪器技术也必将广泛流行于全国各大学校园理工科院系。

2006 年是 LabVIEW 图形化软件开发平台正式推出 20 周年，为了庆祝和纪念这一具有历史意义的时间，NI 将最新发布的 LabVIEW 20 周年纪念版命名为 8.20 版本。相对于 LabVIEW 7.x，LabVIEW 8.0 与 LabVIEW 8.20 的更新力度超过了以前任何一个版本。NI 在 LabVIEW 8 上的研发投入超过了 LabVIEW 7 的两倍。

本书特色

本书针对 LabVIEW 8.20 专业开发版，以由简到难、逐步深入的原则对 LabVIEW 编程进行了全面详细的介绍，尤其是对 LabVIEW 编程人员经常讨论的热点问题进行了重点介绍，例如如何通过 LabVIEW 开发大型项目、如何优化 LabVIEW 程序人机界面与性能、如何通过 LabVIEW 调用 DLL 或 ActiveX、如何通过 LabVIEW 实现网络编程以及与数据库连接等。此外，本书几乎所有的知识点都配合了恰当的实例，所有这些实例都在本书附带的光盘中，读者可以在“附录 A 本书技巧实例索引”中快速检索其中主要的技巧实例。通过这些技巧实例，读者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、基于状态机的温度控制系统、启动界面设计等。

本书编写过程中，几乎参考了 LabVIEW 联机帮助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 编程经验与心得，因此本书几乎包含了所有常用的 LabVIEW 编程知识与编程技巧。如果您是对 LabVIEW 一无所知的新手，通过本书，您可以从入门开始，并逐步深入地对 LabVIEW 进行学习，直到成为真正精通 LabVIEW 的编程高手。如果您已经使用过 LabVIEW 多年，相信本书也会为您提供有益的帮助，并成为您快速定位所需 LabVIEW 编程知识的必备参考。

由于 LabVIEW 8.20 完全向下兼容，因此即使您使用的是 LabVIEW 7.x 或 8.0 版本，本书的绝大部分内容对您仍然适用。

本书结构

本书分为基础篇和高级篇两部分。基础篇介绍了 LabVIEW 编程所需的基础知识，掌握了这些基础知识，您就可以编写从硬件连接、数据采集到数据分析处理、图形显示、存储和查询等功能丰富的小型自动化测试测量程序了。高级篇介绍了编写 LabVIEW 大型或高级应用程序所需的高级知识，通过这些高级知识，您可以编写融合多种高级编程技术的大型分布式应用，并且在各方面提高您所编写的 LabVIEW 程序的质量，例如扩展性、可读性和稳定性等。

本书由陈锡辉主编。张银鸿编写了第 3 章和第 4 章，其余章节均由陈锡辉编写。

本书在编写过程中得到了 NI 工程师以及广大网友的热心帮助；我的多位学弟作为本书初稿的体验读者，提出了宝贵的建议；在此对他们表示衷心的感谢！最后，将爱和感谢献给我美丽的未婚妻范艳丽，她不仅校对了本书部分章节而且提出了很多宝贵意见；最重要的是，她一如既往的鼓励与关心使得我在半年多的时间里能够耐心尽力地写好本书的每一个细节。

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，不吝赐教。本书作者开设了自己的 LabVIEW 博客讨论区：http://labviewstudy.blog.edu.cn，无论您有任何建议或问题，都可以在这里给作者留言。您也可以发送 E-mail 至 chenxh@ihep.ac.cn，作者将尽心尽力地为您服务！

陈锡辉

2007 年 3 月于中国科学院高能物理所

CONTENTS

目录

基 础 篇

高 级 篇

基 础 篇

第1章 揭开LabVIEW的面纱

“最初只存在机器语言，计算机的世界里一片黑暗。可是不久，汇编语言问世了，给计算机的世界投下了一缕曙光。后来，Fortran 的出现带来了光明。”

这是一段关于计算机语言的“圣经”，它表达了长久以来人们对计算机高级语言发展的渴望。计算机发明的初衷就是服务人类，帮助人们去实现他们所关注的事情。然而复杂、枯涩的编程语言把大部分人都拒之门外，人们不得不花费大量的时间与精力去学习与他们所关注的事情毫不相关的语法、命令和编译器等诸如此类的东西。

LabVIEW 图形化编程语言的出现终于把人们——尤其是工程师和科学家们从繁杂的编程工作中解放出来，使他们能够真正专心于自己所关注的事情。通过 LabVIEW 图形化编程环境，编程者可以像搭积木一样“搭建”所见即所得的程序界面，而程序的执行内容则由一个个表示函数的图标和图标之间的数据流连线构成。这不仅使得编程者不再需要记忆纷繁复杂的语法和函数原型，更使编写程序的过程与工程师们的思维习惯相符合，从而使编写程序的过程也变得生动起来。

1.1 LabVIEW 简介

1.1.1 什么是 LabVIEW

LabVIEW（Laboratory Virtual Instrument Engineering Workbench）是一种用图标代替文本行创建应用程序的图形化编程语言。传统文本编程语言根据语句和指令的先后顺序决定程序的执行顺序，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。

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。这就是图形化源代码，又称 G（Graphics）代码。LabVIEW 的图形化源代码在某种程度上类似于数据流流程图，因此又被称作**程序框图代码**。前面板上的每一个控件对应于程序框图中的一个对象，当数据“流向”该控件时，控件就会根据自己的特性以一定的方式显示数据，例如开关、数字或图形。图 1.1 就是一个 LabVIEW 程序实例的前面板与程序框图，该例模拟了一个温度监测系统。

LabVIEW 程序被称为 VI（Virtual Instrument），即虚拟仪器，这是因为它的很多界面控件与操作都模拟了现实世界中的仪器，例如示波器与万用表等。LabVIEW 的核心概念就是“软件即是仪器”，即虚拟仪器的概念。LabVIEW 还包含了大量的工具与函数用于数据采集、分析、显示与

存储等。这些工具都是向导式的工具，用户只需要一步步按照提示就可以实现与仪器的连接和参数的设置。而程序员也不用去记忆这些大量的函数，因为这些函数都以图标与名称的形式存在于一个小小的函数面板上，当需要用到某个函数时把它从函数面板上拖放到程序框图中就可以了。这一切都是图形化带来的好处。

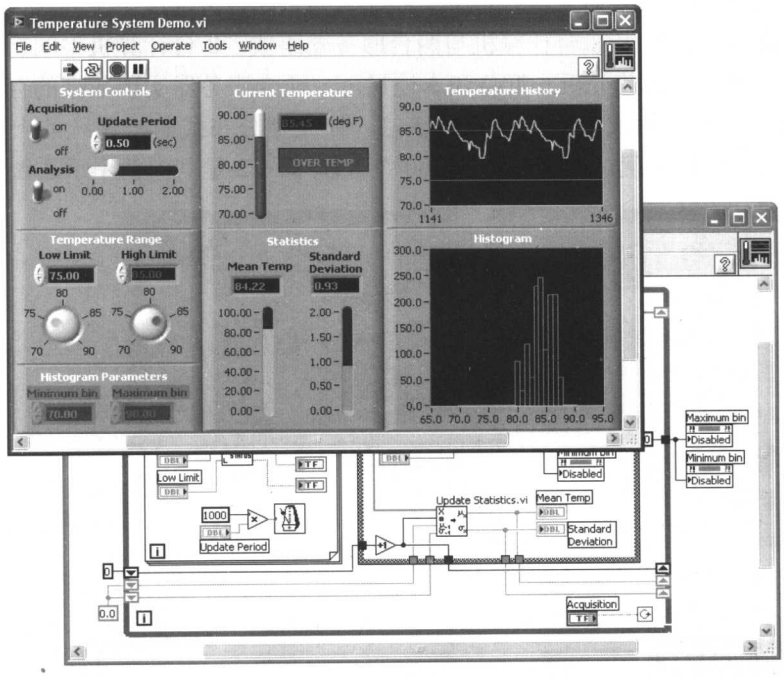


图 1.1　LabVIEW 程序的前面板与程序框图举例

1.1.2　LabVIEW 的作用

由于 LabVIEW 可以用来创建通用的应用程序，因此被称为一种通用的编程语言。但是它在测试、测量和自动化等领域具有更大的优势，因为 LabVIEW 提供了大量的工具与函数用于数据采集、分析、显示和存储。同时它还提供了大量常用于自动化测试测量领域的图形控件。这使得用户可以在数分钟内完成一套完整的从仪器连接、数据采集到分析、显示和存储的自动化测试测量系统。因此它被广泛地应用于汽车、通信、航空、半导体、电子设计生产、过程控制和生物医学等各个领域，涵盖了从研发、测试、生产到服务的产品开发所有阶段。NI 网站上拥有上千个应用案例供读者参考：http://www.ni.com/solutions/。今天欧美的许多高校非计算机专业的学生选修 G 语言并用它开发应用软件的人数已经超过 C 等文本语言。近年来我国高校 G 语言教学实践正在迅速展开。

LabVIEW 不仅可以用来快速搭建小型自动化测试测量系统，还可以用来开发大型的分布式数据采集与控制系统。

在美国 Lawrence Livermore 国家实验室，一个花费 2000 万美金的极为复杂的飞秒激光切割系统就是基于 LabVIEW 开发的。该系统中，4 台 Windows NT 工作站用网络连接起来，LabVIEW 用来给激光提供测量、控制和自动定序，同时作为半熟练操作者的高层用户界面。几乎安装了所

有类型的 I/O 硬件：DAQ、GPIB、串行、远程控制 SCXI、VME/VXI 以及 IMAQ 成像。由于这个项目的极端重要性，因此本项目采取了正式的软件质量保证过程。软件开发总共用了 4 个年度，创建了约 600 个 VI。

在作者参与的北京正负电子对撞机二期工程北京谱仪慢控制系统中，大约有 30 种物理量共 7000 多点的现场数据点需要实时采集控制和分析记录等。作者负责该系统的软件体系结构设计与大部分代码的实现。该系统由 8 台计算机与两台服务器组成。8 台计算机不间断地采集来自于十几种硬件设备的数据，并将其分析、汇总和本地显示。两台服务器实现数据的存储和网络发布，以供科学家们随时随地获得或控制探测器的状态。该大型分布式监控系统的上层软件完全基于 LabVIEW 及其 DSC 模块实现，共创建了约 300 个 VI。图 1.2 是其中两个子系统的主界面。

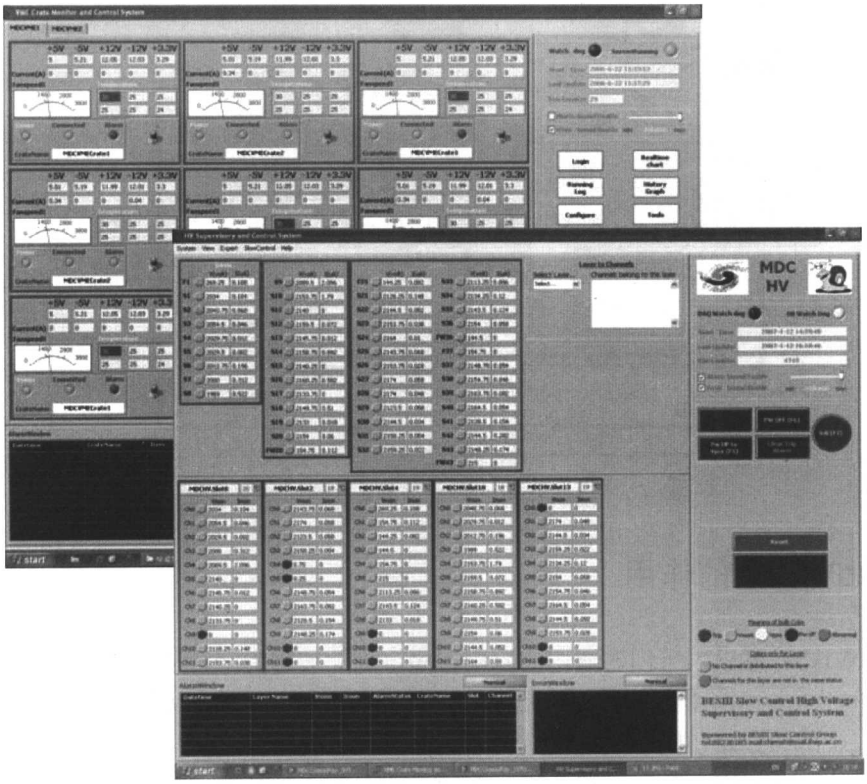


图 1.2　北京谱仪慢控制系统中的两个子系统的主界面

基于 LabVIEW 实现的最大的系统应该是 Honeywell-Measurex 公司由 Dirk Demol 领导的小组开发的 MxProline。它是一流的分布式过程控制系统，其 95%的代码都是用 LabVIEW 编写的。该系统使用了 5000 个以上的 VI，可以处理超过 10 万个变量（包括物理 I/O 和计算值）。

1.1.3　选择 LabVIEW 的原因

选择 LabVIEW 开发测试和测量应用程序的一大决定性因素是其开发速度。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 易用易学，它所提供的工具使创建测试和测量应用变得更为轻松。

LabVIEW 的具体优势主要体现在以下几个方面。

（1）提供了丰富的图形控件，并采用图形化的编程方法，彻底把工程师们从复杂枯涩的文本编程工作中解放出来。

（2）内建的编译器在用户编写程序的同时就在后台自动完成了编译。因此用户在编写程序的过程中如果有语法错误，它会被立即显示出来。

（3）由于采用数据流模型，它实现了自动的多线程，从而能充分利用处理器尤其是多处理器的处理能力。

（4）通过 DLL、CIN 节点、ActiveX、.NET 或 MATLAB 脚本节点等技术，可以轻松实现 LabVIEW 与其他编程语言混和编程。

（5）通过应用程序生成器可以轻松地发布 EXE、动态链接库或安装包。

（6）LabVIEW 提供了大量的驱动与专用工具，几乎能与任何接口的硬件轻松连接。

（7）LabVIEW 内建了 600 多个分析函数，用于数据分析和信号处理。

（8）NI 同时提供了丰富的附加模块，用于扩展 LabVIEW 在不同领域中的应用，例如实时模块、PDA 模块、FPGA 模块、数据记录与监控（DSC）模块、机器视觉模块与触摸屏模块等。

1.2 LabVIEW 的起源与发展历程

1.2.1 起源

早在 20 世纪 80 年代初引入个人计算机之前，几乎所有使用可编程仪器的实验室都通过专门的仪器控制器用于控制他们的测试系统。这些价格昂贵而且功能单一的控制器通过一个必备的端口控制使用 IEEE-488 总线（即 GPIB 总线）的仪器。到了 1983 年，随着个人计算机的出现，National Instruments 公司成为个人计算机的 GPIB 硬件接口的主要供货商。然而当时用于控制仪器的软件表现却不太好，当时几乎 100%的仪器控制程序都是用 BASIC 语言开发的。虽然 BASIC 有一定的优势，例如简单、可读性强的指令集以及可交互功能等，但是它存在一个根本性的问题：像其他文本编程语言一样，如果要通过计算机控制仪器，无论是科学家、工程师还是技术人员都必须懂得编程。他们必须把他们的应用软件和仪器使用的知识转化为文本行，而这种过程多半是繁重而单调乏味的，尤其是对那些从来没有编程经验的人来说。

National Instruments 公司有自己的编程团队，其任务是致力于开发用于控制仪器的 BASIC 程序。他们敏感地注意到了仪器编程工作压在工程师和科学家身上的负担，那就是开发出一种用于开发仪器软件程序的新工具。但是这种工具将采用什么形式呢？两位 NI 公司的创始人 Jim Truchard 和 Jeff Kodosky 博士，连同 Jack MacCrisken（后来成为一名顾问），开始着手开发这种软件工具。Truchard 主要研究能够显著地改进科学工作者和工程师们进行测试开发方式的工具。他想到的软件产品模型是电子数据表格。电子数据表格解决了 Truchard、Kodosky 和 MacCrisken 都同样面临的问题，即如何使非编程的计算机用户能够使用计算机。只不过电子数据表格处理的是财务计划制定者的问题，而这个三人组想的是如何帮助工程师和科学工作者。这三个人的口号是：发明出一种软件工具，它对工程师和科学工作者的影响力要和电子数据表格对财务界的影响力一