

<<LabVIEW宝典>>

图书基本信息

书名：<<LabVIEW宝典>>

13位ISBN编号：9787121129612

10位ISBN编号：7121129612

出版时间：2011-3

出版时间：电子工业

作者：陈树学//刘萱

页数：603

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

计算机的出现，彻底改变了人们的工作和生活方式。

如今计算机已经无处不在，进入了每个人的生活之中。

在工程技术人员看来，计算机不仅仅是人们常见的PC，也包括各种微处理器。

从这个角度看，我们无时无刻不在使用计算机，例如，电视、洗衣机、自动提款机等都依赖计算机来实现各种便捷的功能。

相同的计算机可以完成不同的工作，在于它们使用不同的程序，而程序是由计算机编程语言创建的。

短短几十年中，出现了众多的编程语言，这些编程语言的共同特点是采用文本方式创建程序。

文本方式编程对编程人员要求很高，这使得计算机编程只能是少数人才可以从事的职业。

美国国家仪器公司（National Instruments，简称NI）的创新软件产品LabVIEW，允许用图形方式编程，摒弃了晦涩难懂的文本代码，使得计算机编程不再是少数人的专利。

LabVIEW的最早版本诞生于1986年，几乎和Windows的最早版本同步，这注定了LabVIEW是多平台的编程语言，适合于不同的操作系统。

20世纪80年代初，NI公司是GPIB总线设备的主要供货商，丰富的硬件经验和强大的软件开发需求，促使NI公司的工程师们决心寻找一种代替传统编程语言的开发工具，这导致了1986年LabVIEW的横空出世。

LabVIEW是由测试工程师开发的专用编程语言，因此，LabVIEW具有鲜明的行业特点，最早主要用于测试测量领域。

NI公司独创了虚拟仪器的理念，提出了“软件就是仪器”的理念，并逐步成为业界的标准。

随着LabVIEW的不断发展，几乎每隔一两年，都要推出新的版本。

LabVIEW的应用范围已经覆盖了工业自动化、测试测量、嵌入式应用、运动控制、图像处理、计算机仿真、FPGA等众多领域。

以LabVIEW为核心，采用不同的专用工具包，统一的图形编程方式，可以实现不同技术领域的需求。

《LabVIEW宝典》要点 《LabVIEW宝典》作者多年使用LabVIEW作为主要的编程语言，因此特别关注LabVIEW在工业领域的具体运用以及LabVIEW的实用编程技术。

本书共17章，分为入门篇、高级篇、工程应用篇。

第1~5章为入门篇，介绍了LabVIEW的基本概念、基本函数的用法和常用的运行结构，详尽地分析了LabVIEW的基本数据结构和文件存储方式。

第6~11章为高级篇，介绍了应用程序、VI和控件的引用、属性和方法，以及各类高级控件的运用方法。

第8章介绍了LabVIEW的文本编程方式以及DLL、C语言接口，第9章详细介绍了基于Matlab语法的MathScript编程技术，第10章介绍了LabVIEW基于组件的编程方法。

第12~17章为工程应用篇，具体而细致地讲解在做实际工程开发时所用到的LabVIEW编程技术。本篇结合LabVIEW的常用工具包，具体分析了计算机串口、并口、网络通信方面的内容，以及数据记录和监控工具包、数据库连接工具包、报表生成工具包、状态图工具包、FPGA工具包等等。

第13、第14章详细介绍了数据采集的基本原理和常用编程方法，其中重点介绍了LabVIEW在实时系统下的运用。

第15章讲解LabVIEW实时系统的开发案例。

第16章讲解了LabVIEW数据采集系统开发案例。

第17章讲解了FPGA的开发案例，利用LabVIEW图形编程方式，提高开发效率。

《LabVIEW宝典》读者 《LabVIEW宝典》可作为高等院校通信、测量技术、自动控制等相关课程的教材和教学参考书，也可作为相关工程技术人员设计开发仪器或自动测试系统的技术手册。

《LabVIEW宝典》特色 《LabVIEW宝典》内容非常丰富，在每个章节都采用了大量的实例，针对具体编程实践中遇到的问题，提出了多种解决方法。

在兼顾基础知识介绍的前提下，深入讨论了LabVIEW的高级编程方法和编程技巧。

《LabVIEW宝典》的宗旨是作为实用工具书，侧重于来自工程实践的一线案例。

笔者在写作过程中，与众多的LabVIEW爱好者进行了充分的沟通与交流，总结编程过程中经常遇到的问题，并作为本书的重要内容。

《LabVIEW宝典》使用了大量篇幅，重点讲解NI公司各种专用工具包的运用，具有一定的深度和实用价值，特别适合于具有一定基础的编程人员使用。

在基础篇中，虽然也介绍了许多LabVIEW的基础知识，但还是侧重于具体应用，其中大量的例程可以直接在具体项目中使用。

在高级篇中，重点介绍了LabVIEW常用的编程模式，以及状态图工具包的运用，同时介绍了新增的面向对象的编程方法。

内容概要

本书详尽讲解了labview常用的编程方法、编程技巧和工程应用技术。全书共分为3篇，其中：入门篇归纳总结了labview编程人员必须掌握的基础知识，包括labview的基本概念、基本函数的用法和常用的运行结构，以及labview的基本数据结构和文件存储方式；高级篇细致地讲解了引用、属性、方法以及各类高级控件的运用，labview的文本方式编程以及dll、c语言接口，基于matlab语法的mathscript编程技术，labview基于组件的高级编程方法和编程模式；工程应用篇介绍了串口、并口和网络通信的常用方法，数据采集?基本原理和方法，labview实时系统的构建和编程，以及各种常用专业工具包的使用方法，包括数据库连接工具包、数据监控与记录工具包、报表生成工具包、状态图工具包等等。

本书可作为高等院校通信、测量技术、自动控制等相关课程的教材和教学参考书，也可作为相关工程技术人员设计开发仪器或自动测试系统的技术手册。

作者简介

陈树学，1989年毕业于哈尔滨工业大学动力工程系，毕业后进入一汽集团公司，负责专用机床、加工自动线和测试设备的软件开发工作。

初期使用C语言，从LabVIEW 6.0版本开始，采用LabVIEW作为首选编程语言。

刘萱，2005年毕业于天津大学计算机科学与技术专业。

毕业后进入天津久荣车轮技术有限公司主持车轮轮胎检测设备系统软件的研发工作。

2006年首次接触LabVIEW 7.2并在实际项目中使用，几年来积累了利用LabVIEW开发工程项目的丰富经验。

其中独立使用LabVIEW进行开发的设备机型有：ROT-1在线式车轮摆差检测线、ROT-2 / 3手动车轮摆差机、RRPC，丁B轮胎滚动阻力试验机、CFT-3 / 5不转式车轮弯曲疲劳试验机等。

书籍目录

第1部分 入门篇

第1章 打开labview编程之门

1.1 从vi开始

1.1.1 创建vi

1.1.2 控件属性设置与快捷菜单

1.1.3 创建控件、常量、局部变量、引用、属性节点和方法节点

1.1.4 创建自定义控件

1.2 编辑前面板和程序框图

1.2.1 选择、移动和删除对象

1.2.2 使用布局工具

1.3 vi及其属性对话框

1.3.1 vi的层次结构

1.3.2 调用子vi

1.3.3 vi的属性设置

1.4 基本控件及其使用方法

1.4.1 基本数值控件

1.4.2 基本布尔控件

1.4.3 控?的通用编辑方法

1.4.4 字符串和路径控件

1.4.5 下拉列表与枚举控件

1.4.6 数组控件及其属性设置

1.4.7 簇控件

1.4.8 时间标识控件与波形数据控件

1.5 小结

第2章 labview基本函数

2.1 必须了解的一些基本算术运算函数

2.1.1 基本运算函数

2.1.2 标量之间的基本运算

2.1.3 标量与数组的运算

2.1.4 数组与数组的运算

2.1.5 数组的函数

2.1.6 标量与簇的基本运算

2.1.7 簇与簇的运算

2.1.8 簇的函数

2.2 必须了解的位运算函数和逻辑运算函数

2.2.1 常用逻辑运算函数

2.2.2 位运算

2.2.3 深入理解复合运算函数

2.3 必须了解的关系运算函数和比较函数

2.3.1 比较模式

2.3.2 通用关系运算函数

2.3.3 “比较0”关系运算函数

2.3.4 复杂关系运算函数

2.3.5 字符关系运算函数

2.3.6 表达式节点与公式快速vi

<<LabVIEW宝典>>

2.4 小结

第3章 labview的程序运行结构

3.1 两种不同的循环结构

3.1.1 for循环的组成和特点

3.1.2 for循环与数组

3.1.3 for循环与移位寄存器

3.1.4 for循环中的continue和break

3.1.5 while循环，不仅仅是循环

3.1.6 while循环与定时

3.1.7 反馈节点

3.2 定时结构

3.2.1 定时循环的基本组成要素和配置对话框

3.2.2 定时顺序结构

3.3 独特的条件结构

3.3.1 条件结构的基本结构

3.3.2 布尔型输入

3.3.3 错误簇输入

3.3.4 数值型输入

3.3.5 枚举型输入

3.3.6 下拉列表输入

3.3.7 字符串和组合框输入

3.3.8 输入、输出隧道

3.3.9 多重if-else的处理方法

3.4 不和谐的顺序结构

3.4.1 多线程运行次序

3.4.2 两种不同的顺序结构

3.4.3 隧道与顺序局部变量

3.4.4 顺序结构的替代

3.4.5 顺序结构的典型应用

3.5 程序框图禁用结构

3.6 局部变量、内置全局变量和功能（lv2型）全局变量

3.6.1 局部变量

3.6.2 内置全局变量

3.6.3 功能（lv2型）全局变量

3.7 事件结构

3.7.1 事件结构的基本构成和创建方法

3.7.2 事件的分类及其特点

3.7.3 事件结构之间的数据传送与共享

3.7.4 事件发生的次序、过滤和转发

3.7.5 正确地使用事件结构

3.8 小结

第4章 labview的数据结构及内存优化

4.1 常用数据类型转换函数

4.1.1 “强制类型转换”函数

4.1.2 “平化至字符串”与“字符串还原”函数

4.1.3 变体数据

4.2 整数的类型转换及内存映射

<<LabVIEW宝典>>

- 4.2.1 布尔型数据与字符串和数值的相互转换
 - 4.2.2 u8类型数据与字符串之间的相互转换
 - 4.2.3 整数与整数类型的相互转换
 - 4.3 其他标量数据类型的类型转换及内存映射
 - 4.3.1 定点数和浮点数的类型转换与内存映射
 - 4.3.2 复数的类型转换及内存映射
 - 4.3.3 时间标识的类型转换与内存映射
 - 4.4 复合数据类型
 - 4.4.1 标量数组的内存映射
 - 4.4.2 字符串、路径和字符串数组的内存映射
 - 4.4.3 labview使用的编码
 - 4.5 簇的内存映射
 - 4.5.1 标量组成的簇
 - 4.5.2 包含数组和字符串的簇
 - 4.6 类型描述符
 - 4.6.1 类型描述符的基本构成要素
 - 4.6.2 常用类型描述符列表
 - 4.6.3 常见数据类型的类型描述符结构
 - 4.7 openg中有关类型描述符的函数
 - 4.7.1 类型描述符函数
 - 4.7.2 利用类型描述符处理枚举型数据
 - 4.7.3 利用类型描述符处理簇
 - 4.8 几种常用的内存分析工具和方法
 - 4.8.1 内存的重要性
 - 4.8.2 内存和性能查看工具
 - 4.8.3 vi使用的内存
 - 4.8.4 优化内存的一般注意事项
 - 4.8.5 数组与内存优化
 - 4.8.6 在循环中避免不必要的计算、读/写控件或者变量
 - 4.9 影响vi运行速度的因素
 - 4.9.1 硬件输入/输出
 - 4.9.2 屏幕显示
 - 4.10 小结
- 第5章 字符串与文件存储
- 5.1 字符串
 - 5.1.1 几种常用的字符串常量
 - 5.1.2 几种常用的字符串函数
 - 5.1.3 “匹配模式”和“匹配正则表达式”函数
 - 5.1.4 字符串与数值的相互转换
 - 5.1.5 功能强大的“格式化字符串”和“扫描字符串”函数
 - 5.1.6 数组与电子表格字符串
 - 5.1.7 附加字符串函数
 - 5.2 文件存储
 - 5.2.1 文本文件与二进制文件的区别
 - 5.2.2 文件常量和通用目录、文件函数
 - 5.2.3 构造路径的方法
 - 5.2.4 文本文件的读写

<<LabVIEW宝典>>

5.2.5 数据记录文件的读写

5.2.6 读/写二进制文件

5.2.7 ini文件的读写

5.2.8 注册表的读写

5.2.9 tdm文件

5.2.10 tdms文件

5.3 小结

第2部分 高级篇

第6章 labview对象的解析

6.1 labview控?对象的层次继承结构

6.1.1 布尔控件的层次继承结构

6.1.2 通用类的属性

6.1.3 图形对象类

6.2 图形对象类的子类

6.2.1 前面板类

6.2.2 窗格类和分隔栏类

6.2.3 labview的坐标映射

6.2.4 修饰类

6.3 控件类

6.3.1 控件类的常用属性

6.3.2 控件类的常用方法

6.3.3 数值控件类

6.4 常用控件的专用属?

6.4.1 布尔控件的专用属性

6.4.2 枚举和下拉列表控件的专用属性

6.4.3 字符串、路径和组合框控件的专用属性

6.4.4 数组的属性和方法

6.4.5 簇的属性及方法

6.5 引用句柄

6.6 vi的属性

6.6.1 获取vi的引用

6.6.2 常用vi属性

6.6.3 vi前面板属性

6.7 常用vi方法

6.7.1 获取前面板、程序框图和vi图标的图像

6.7.2 前面板的运行位置控制方法

6.7.3 打印控制

6.7.4 默认值方法

6.8 动态调用vi

6.8.1 静态调用和动态调用的比较

6.8.2 通过“引用节点调用”函数动态调用vi

6.8.3 一般类型vi的动态调用

6.8.4 创建闪屏

6.8.5 创建后台运行程序

6.8.6 创建向导程序

6.8.7 动态调用vi之间的数据交换

6.9 应用程序的属性和方法

<<LabVIEW宝典>>

6.9.1 获取应用程序引用句柄

6.9.2 常用应用程序的属性

6.10 小结

第7章 高级控件的运用

7.1 列表框

7.1.1 列表框的创建及显示风格

7.1.2 列表框的常用属性、方法与事件

7.1.3 列表框的应用举例

7.2 多列列表框

7.2.1 显示多列项目并排序

7.2.2 多列列表框的特效制作

7.3 表格

7.3.1 表格的常用属性和方法

7.3.2 表格的应用举例

7.4 树形控件

7.4.1 树形控件的创建与静态编辑

7.4.2 树形控件的常用属性、方法和事件

7.4.3 树形控件高级应用举例

7.5 波形图表

7.5.1 波形图表的组成要件

7.5.2 波形图表的输入类型

7.5.3 波形图表专用属性

7.5.4 波形图表应用举例

7.6 波形图

7.6.1 波形图控件的创建和组成要件

7.6.2 波形图控件的输入类型

7.6.3 波形图控件的专用属性

7.6.4 波形图控件的高级应用举例

7.7 xy图

7.7.1 xy图的输入数据类型

7.7.2 xy图的高级应用

7.8 强度图表和强度图

7.9 数字数据、数字波形数据与数字波形图

7.9.1 数字数据

7.9.2 数字波形数据和数字波形图

7.10 图片控件

7.10.1 利用图片控件显示图片

7.10.2 常用?图操作函数

7.10.3 图片控件的高级应用

7.11 小结

第8章 文本式编程与外部程序接口

8.1 公式节点

8.1.1 公式节点的数据类型、语法与控制结构

8.1.2 公式节点的应用举例

8.2 调用库函数

8.2.1 dll与api函数

8.2.2 如何调用dll函数

<<LabVIEW宝典>>

- 8.2.3 常用api函数的调用
- 8.2.4 labview调用dll的局限性
- 8.3 cin
 - 8.3.1 cin创建的一般过程
 - 8.3.2 cin的数据类型和常用函数
 - 8.3.3 cin与内存管理器
 - 8.3.4 cin的运行过程和数据共享
- 8.4 系统命令
- 8.5 剪切板
- 8.6 dde库
 - 8.6.1 dde概述
 - 8.6.2 labview中dde的常用方法
- 8.7 activex控件与activex文档
 - 8.7.1 activex的基本概念
 - 8.7.2 activex控件的调用过程
 - 8.7.3 activex应用实例
 - 8.7.4 activex自动化服务器
- 8.8 .net技术
 - 8.8.1 .net控件
 - 8.8.2 .net服务
 - 8.8.3 利用.net创建托盘程序
- 8.9 小结
- 第9章 mathscript
 - 9.1 如何使用mathscript
 - 9.1.1 使用mathscript节点
 - 9.1.2 使用mathscript交互窗口
 - 9.2 mathscript常用命令
 - 9.3 mathscript基础知识
 - 9.3.1 创建向量和矩阵的基本方法
 - 9.3.2 矩阵的基本运算
 - 9.3.3 标准矩阵
 - 9.3.4 矩阵元素的插入、替换、删除和提取
 - 9.3.5 矩阵元素的排序和搜索特征值
 - 9.3.6 常用的矩阵变换函数
 - 9.3.7 矩阵中元素的数据类型及其转换
 - 9.3.8 关系运算、逻辑运算和位操作
 - 9.3.9 集合函数
 - 9.3.10 时间、日期和计时函数
 - 9.4 程序控制结构与函数
 - 9.4.1 for循环和while循环
 - 9.4.2 if条件结构和switch分支条件结构
 - 9.4.3 函数和脚本文件
 - 9.5 数据统计和数据插值拟合
 - 9.5.1 常用数据统计函数
 - 9.5.2 数据插值
 - 9.6 多项式、积分和微分
 - 9.6.1 多项式

<<LabVIEW宝典>>

- 9.6.2 极值与零点
- 9.6.3 积分和微分
- 9.7 数据的图形显示
 - 9.7.1 窗口类属性与常用窗口操作函数
 - 9.7.2 绘图区域属性
 - 9.7.3 线对象和文本对象的属性及常用函数
 - 9.7.4 基本绘图函数
- 9.8 小结
- 第10章 面向对象编程与运行控制技术
 - 10.1 数据的封装与隔离
 - 10.1.1 合理地使用数据流
 - 10.1.2 lv2型全局变量
 - 10.1.3 值变化与上升下降沿
 - 10.1.4 定时触发与计数器
 - 10.2 动作机 (action engine)
 - 10.2.1 准备建立动作机
 - 10.2.2 建立动作机的步骤
 - 10.3 用户事件与动态注册事件
 - 10.3.1 用户事件
 - 10.3.2 动态注册事件
 - 10.4 堆栈与数据缓冲区
 - 10.4.1 堆栈的实现
 - 10.4.2 数据缓冲区
 - 10.5 同步控制技术
 - 10.5.1 队列
 - 10.5.2 通知器
 - 10.5.3 信号量与集合点
 - 10.6 项目管理器
 - 10.6.1 项目管理器的结构
 - 10.6.2 虚拟文件夹与项目库
 - 10.7 面向对象编程
 - 10.7.1 面向对象编程的基本概念
 - 10.7.2 类的封装特性
 - 10.7.3 类的继承特性
 - 10.7.4 类的多态特性
 - 10.7.5 类变量
 - 10.7.6 动态加载类与引用转换
 - 10.8 小结
- 第11章 人机交互与编程风格
 - 11.1 对话框
 - 11.1.1 内置对话框
 - 11.1.2 用户输入和显示对话框
 - 11.1.3 定制对话框
 - 11.2 菜单
 - 11.2.1 创建静态菜单
 - 11.2.2 菜单相关函数
 - 11.2.3 动态创建菜单

<<LabVIEW宝典>>

- 11.2.4 调用多个静态菜单
 - 11.2.5 存储动态建立的菜单
 - 11.2.6 自动触发预定义菜单项
 - 11.3 光标工具
 - 11.3.1 “ 设置为忙碌状态 ” vi与“ 取消设置为忙碌状态 ” vi
 - 11.3.2 使用光标文件
 - 11.4 选项卡、子面板与分隔栏
 - 11.4.1 选项卡控件
 - 11.4.2 分隔栏控件
 - 11.4.3 子面板控件
 - 11.5 xcontrol
 - 11.5.1 hover按钮
 - 11.5.2 新建xcontrol
 - 11.5.3 修改数据控件和状态控件
 - 11.5.4 修改“ 外观 ” vi
 - 11.5.5 创建属性和方法
 - 11.5.6 调试xcontrol
 - 11.5.7 自定义属性对话框与快捷菜单
 - 11.6 错误处理
 - 11.6.1 错误簇
 - 11.6.2 常用错误处理函数
 - 11.7 labview的编程风格
 - 11.7.1 编程风格的内涵
 - 11.7.2 前面板设计应该遵循的原则
 - 11.7.3 程序框图设计应该遵循的原则
 - 11.7.4 连接板设计应该遵循的原则
 - 11.7.5 图标设计应该遵循的原则
 - 11.7.6 数据结构应该遵循的原则
 - 11.7.7 错误处理应该遵循的原则
 - 11.8 小结
- 第3部分 工程应用篇
- 第12章 labview设计模式与状态图工具
- 12.1 程序的基本单元vi
 - 12.1.1 vi的可重入属性
 - 12.1.2 vi模板与代码重用
 - 12.1.3 vi的调试
 - 12.1.4 vi的重构
 - 12.2 labview标准设计模式
 - 12.2.1 使用事件的顶层应用程序
 - 12.2.2 使用事件的对话框
 - 12.2.3 标准状态机设计模式
 - 12.2.4 队列消息处理器设计模式
 - 12.2.5 用户界面事件处理器设计模式
 - 12.2.6 生产者/消费者设计模式（事件）
 - 12.2.7 生产者/消费者设计模式（数据）
 - 12.2.8 主从设计模式（master/slave）
 - 12.3 简单设计模式

<<LabVIEW宝典>>

- 12.3.1 顺序结构简单设计模式
- 12.3.2 事件结构与定时结构简单设计模式
- 12.4 古典型状态机
 - 12.4.1 早期界面处理状态机
 - 12.4.2 顺序结构的缺陷
 - 12.4.3 顺序状态机
 - 12.4.4 处理公共状态
 - 12.4.5 事件状态机
- 12.5 状态机工具包 (state diagram)
 - 12.5.1 调用状态机工具
 - 12.5.2 使用状态图编辑器
 - 12.5.3 添加转换条件和?态代码
 - 12.5.4 选择独立运行或者子vi方式
- 12.6 消息队列状态机
 - 12.6.1 消息队列状态机的基本结构
 - 12.6.2 消息数据队列状态机
 - 12.6.3 进入、运行和离开状态的处理
 - 12.6.4 事件驱动方式消息队列状态机
- 12.7 状态图工具包 (statechart)
 - 12.7.1 状态图工具包简介
 - 12.7.2 同步和异步方式
 - 12.7.3 创建状态图
 - 12.7.4 同步型状态图
 - 12.7.5 状态图的调用和调试
 - 12.7.6 异步型状态图
 - 12.7.7 区域、超级状态和子状态
 - 12.7.8 多区域并发、连接、分叉与子图
 - 12.7.9 高级应用函数
- 12.8 小结
- 第13章 labview通信与dsc
 - 13.1 串口通信
 - 13.1.1 串口通信的基本概念
 - 13.1.2 串口通信的准备工作
 - 13.1.3 串口通信函数
 - 13.1.4 串口通信典型应用举例
 - 13.2 并?通信
 - 13.2.1 设置并口通信模式
 - 13.2.2 传送字节型数据
 - 13.2.3 传送epp模式数据
 - 13.3 共享变量
 - 13.3.1 共享变量与共享变量引擎
 - 13.3.2 创建与监视共享变量
 - 13.3.3 共享变量的内部缓冲机制
 - 13.3.4 共享变量的批量创建、部署与引用
 - 13.4 datasocket
 - 13.4.1 datasocket支持的协议与url
 - 13.4.2 datasocket服务器与服务管理器

<<LabVIEW宝典>>

- 13.4.3 datasocket api
- 13.4.4 datasocket api应用举例
- 13.4.5 datasocket控件绑定
- 13.5 tcp与udp网络通信
 - 13.5.1 tcp通信
 - 13.5.2 tcp stm库
 - 13.5.3 udp通信
- 13.6 dsc工具包
 - 13.6.1 opc与dsc的基本概念
 - 13.6.2 dsc强大的图形化能力
 - 13.6.3 opc配置与i/o变量
 - 13.6.4 modbus
 - 13.6.5 共享变量的属性
 - 13.6.6 共享变量引擎sve函数
 - 13.6.7 预警与事件
 - 13.6.8 数据记录
 - 13.6.9 安全与权限管理
- 13.7 小结
- 第14章 数据库与报表工具包
 - 14.1 准备使用数据库工具包
 - 14.1.1 创建数据库
 - 14.1.2 建立数据源
 - 14.1.3 数据库工具包支持的数据类型
 - 14.1.4 ado模型
 - 14.2 数据库基本操作
 - 14.2.1 建立连接
 - 14.2.2 表操作
 - 14.2.3 插入数据
 - 14.2.4 读取数据
 - 14.2.5 记录集与数据浏览
 - 14.2.6 事务与提交
 - 14.2.7 使用命令对象和sql语句
 - 14.3 报表与报表工具包
 - 14.3.1 labview中的报表vi
 - 14.3.2 vi说明信息与html报表
 - 14.3.3 报表布局与高级报表vi
 - 14.3.4 利用word和excel模板创建报表
 - 14.4 利用报表工具包操作excel
 - 14.4.1 常用的简单excel vi
 - 14.4.2 单元格格式
 - 14.4.3 图表与图片vi
 - 14.4.4 excel通用vi和高级vi
 - 14.5 利用报表工具操作word
 - 14.5.1 word简单vi
 - 14.5.2 word通用vi
 - 14.5.3 word表格与图表vi
 - 14.6 小结

第15章 labview与实时操作系统

15.1 实时操作系统

15.1.1 实时操作系统的特点与实现

15.1.2 操作系统的有关名词解释

?15.1.3 labview中的实时开发软件

15.1.4 labview支持的实时操作系统

15.1.5 labview实时平台概述

15.2 实时控制器软件安装及其配置

15.2.1 配置实时系统bios (pxi)

15.2.2 max下安装pxi实时软件

15.2.3 识别远程设备

15.2.4 建立实时项目

15.3 实时应用软件高级编程及技巧

15.3.1 实时操作系统下labview不支持的特性

15.3.2 实时操作系统下的多线程

15.3.3 实时系统中时间确定性的实现

15.3.4 实时系统中的线程间通信

15.3.5 实时控制系统的网络通信

15.3.6 实时控制系统的软件架构

15.4 小结

第16章 labview实现数据采集

16.1 数据采集的基本概念

16.1.1 信号

16.1.2 传感器

16.1.3 信号处理

16.2 数据采集卡

16.2.1 数据采集卡的定义及分类

16.2.2 多功能数据采集卡原理图

16.2.3 数据采集的关键参数和概念

16.2.4 数据采集卡与信号接地

16.3 采样定理

16.4 降低系统噪声和提高精度

16.5 如何选购采集卡

16.6 数据采集软件基础

16.6.1 采集系统的安装

16.6.2 ni采集卡的常用函数

16.6.3 研华常用采集函数

16.7 基于ni-daqmx的高级编程

16.7.1 触发信号 (trigger)

16.7.2 采集系统时钟

16.7.3 多板卡之间的同步采集

16.7.4 数据传输机制

16.7.5 完整波形输出

16.7.6 并行结构进行采集

16.7.7 通过硬件信号触发定时循环运行

16.7.8 用ni-daqmx的事件编写面向事件驱动的程序

16.7.9 选择合适的读取策略

16.7.10 使用ni-daqmx控制任务安全中止采集

16.7.11 计数器/定时器及其应用

16.8 小结

第17章 fpga开发

17.1 fpga的基本概念与crio的组成

17.1.1 fpga的基本概念

17.1.2 crio的构成

17.1.3 构建fpga项目

17.2 fpga编程

17.2.1 fpga基本i/o之模拟量输入/输出

17.2.2 fpga基本i/o之数字量输入/输出

17.2.3 fpga定时、时钟与分频

17.2.4 fpga计数器应用

17.2.5 触发与外部时钟循环

17.2.6 fpga常用函数

17.2.7 fpga多线程与线程之间的数据交换

17.2.8 fpga ip core

17.3 fpga与rt程序之间的数据交换

17.3.1 读写控件方式

17.3.2 中断

17.3.3 fifo

17.3.4 扫描方式

17.3.5 专用c模块

17.3.6 fpga程序的优化

17.4 spartan-3e开发板

17.4.1 spartan-3e简介

17.4.2 建立spartan-3e fpga项目

17.4.3 编译fpga程序

17.5 小结

章节摘录

版权页：插图：Windows操作系统中，检测事件是否发生是由操作系统负责的。它在内部维护一个消息队列，对每一个发生的事件，以消息的形式置于队列之中。因此，消息的处理是按顺序进行的，当然如果有必要，紧迫的消息可以提前调出队列。消息的种类非常多，但是对于LabVIEW的编程来说，只要了解一些基本常识就可以了。假如我们没有使用事件结构，操作系统是否会检测消息或者事件呢？操作系统当然会检测而且处理消息，我们的事件结构不过是对需要响应的事件注册了回调函数。如果某个事件没有响应，操作系统采用默认方式处理。因此，我们可以决定是否响应事件，并根据外部情况，决定事件是否继续或者终止。我们知道，LabVIEW的事件结构既可以处理窗格事件（也就是前面板客户区中发生的事件），也可以处理前面板上控件发生的事件。如果在两个事件分支中，分别响应窗格的鼠标按下事件和控件按下事件，此时我们单击控件，到底响应哪个事件？如果同时响应两个事件，它们的先后次序如何？一般是，先产生前面板窗格事件，然后产生控件事件。鼠标事件包括鼠标进入、鼠标离开、鼠标按下、鼠标释放、鼠标移动等。除了鼠标移动是连续不断发生的持续事件，其他鼠标事件都是一次性事件。持续事件耗费资源比较多，若非必要，不宜使用。我们可以通过动态注册事件的方法，在必要的时候，允许鼠标移动事件，然后暂时停止鼠标移动事件。

媒体关注与评论

首先得感谢作者，为我们虚拟仪器行业的朋友们带来了一部关于LabVIEW的如此精彩的好书！俗话说“一分耕耘一分收获”，两位作者历时两年，付出了大量的心血，对自己多年的LabVIEW开发经验进行梳理和总结，才完成此书。

能为此书写书评，本人深感荣幸！

本书讲述的内容全面、细致而且准确，图文并茂，值得LabVIEW开发者认真品味和学习。

本书从初级到较深层次的LabVIEW技术都有较详尽的介绍，对于初步接触LabVIEW的朋友来说，这是一本很实用的书籍。

当然，本书不仅适合于初学者，对于已经熟悉了LabVIEW的朋友，也能从中学习到大量自己曾经忽略或者没有接触过的知识。

书里面讲述的一些知识点，我以前也没有接触过，读完此书获益匪浅。

好书当如良师益友，为你指点迷津，与你切磋技艺。

《LabVIEW宝典》就是这样一本值得大家品读的好书！

——CPUBBS论坛创始人，版主，赖卫国

编辑推荐

《LabVIEW宝典》：配套PPT电子课件LabVIEW工程师的10年经验总结全程配套700多个VI案例文件3大完整工程实践一线项目

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>