

高等院校计算机应用系列教材

LabVIEW 虚拟仪器 程序设计教程

张 峤 主 编

朱世宇 孙令翠 陆 鹏 副主编

清华大学出版社

北 京

内 容 简 介

本书主要介绍了利用 LabVIEW 2018 进行虚拟仪器程序设计的方法和技巧,详细讲解了虚拟仪器的概况、LabVIEW 2018 编程环境及其基本操作、数据类型与运算、程序结构、图形显示、子 VI 属性节点和人机界面设计、文件 I/O、网络与通信编程、LabVIEW 数据库编程、数据采集以及信号分析与处理。本书每个章节都配有大量的编程实例,可以让读者更加快捷地掌握相应的编程方法,并能熟练运用。

本书针对高等院校学生的特点,以软件开发设计思想为主线,按照“循序渐进、逐步深入、重在实践”的原则来编写,适合作为高等院校电子电路相关专业的教材,也可供虚拟仪器程序设计人员作为参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。举报:010-62782989, beiqinquan@tup.tsinghua.edu.cn

图书在版编目(CIP)数据

LabVIEW 虚拟仪器程序设计教程 / 张峤主编. —北京:清华大学出版社, 2021.7

高等院校计算机应用系列教材

ISBN 978-7-302-57320-3

I. ①L… II. ①张… III. ①软件工具—程序设计—高等学校—教材 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2021)第 012332 号

责任编辑:刘金喜

封面设计:高娟妮

版式设计:思创景点

责任校对:成凤进

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:三河市铭诚印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:21

字 数:538 千字

版 次:2021 年 8 月第 1 版

印 次:2021 年 8 月第 1 次印刷

定 价:65.00 元

产品编号:087030-01

前 言

现代科学技术、生产和国防的重要特点之一，就是进行大量的测试和统计。现代工业大生产中，用在测试上的工时和费用已占整个生产成本的 20%~30%，因此提高测试水平、降低测试成本、减少测试误差、提高测试效率，对国民经济的各个领域都是至关重要的。测试手段的现代化，已被公认为科学技术和生产现代化的重要条件和明显标志。

在这样的大环境下，虚拟仪器(Virtual Instrumentation, VI)应运而生，它突破了以往传统仪器的特点，充分利用不断发展和完善的计算机技术，以通用计算机和标准总线技术为平台，利用计算机的硬件资源，并辅以软件作为虚拟仪器的开发平台。用户利用面向测量仪器的控制和管理的视窗软件平台 LabVIEW、一台普通的计算机、若干软件包和基本的硬件电路(如数据采集电路、GPIB 仪表、VXI 仪表等)就可以构建一套完整的测试系统，并具备数据处理的功能和友好的人机界面(通常称为虚拟面板)。现如今，基于 LabVIEW 的虚拟仪器已经成为一种业界领先的工业标准化、图形化编程工具，主要用来开发测试测量、控制系统。

美国国家仪器公司(National Instruments, NI)作为虚拟仪器技术的主要倡导者，无论是在硬件还是软件上都做出了突出的贡献，其推出的图形化编程语言——LabVIEW 是目前国际上最成功的图形化集成开发环境，并在众多领域得到了广泛应用。LabVIEW 自 1986 年问世以来，经过不断改进和版本升级，已经从最初简单的数据采集和仪器控制的工具发展成为科技人员用来设计、发布虚拟仪器软件的图形化平台，并具有强大的功能和易用性。

本书针对高等院校学生的特点，以软件开发设计思想为主线，按照“循序渐进、逐步深入、重在实践”的原则，加入大量示例来帮助读者学习。此外，本书通过理论与实例相结合的方式，介绍了利用 LabVIEW 2018 进行虚拟仪器程序设计的方法和技巧。

全书共分 11 章，主要内容如下。

第 1 章介绍仪器的发展、虚拟仪器相关知识、虚拟仪器的分类、虚拟仪器的开发环境。

第 2 章介绍 LabVIEW 基础、LabVIEW 2018 的安装、LabVIEW 2018 的编程环境和 LabVIEW 2018 的基本操作。

第 3 章介绍 LabVIEW 的数据类型和基本操作，包括基本数据类型、数据操作、数组、字符串与路径、簇和矩阵等。

第 4 章介绍 LabVIEW 中的程序结构，重点介绍了 LabVIEW 中的顺序结构、循环结构、条件结构、事件结构、禁用结构、公式节点、定时结构等。

第 5 章介绍 LabVIEW 的图形显示，包括波形显示、XY 图与 Express XY 图、强度图与强度图表、数字波形图、三维图形等。

第 6 章介绍 LabVIEW 中子 VI、属性节点和人机界面设计，主要介绍了子 VI 的创建和属性节点的应用，并重点介绍了下拉列表、对话框、菜单等高级控件的应用和人机界面设计的基本技巧。

第7章介绍 LabVIEW 中的文件 I/O 操作, 主要包括文件操作基本函数、文本文件、电子表格文件、二进制文件、波形文件、测量文件、配置文件、XML 文件及 TDMS 文件的写入与读取等操作。

第8章介绍网络与通信编程, 包括 TCP 通信、UDP 通信、串行通信及其他通信技术。

第9章介绍数据编程的相关知识, 包括 LabVIEW 数据库基础、LabSQL 数据库访问、ADO 数据库访问和 LabVIEW SQL Toolkit 数据库访问。

第10章介绍数据采集的相关知识, 包括数据采集的基础知识、DAQ 设备的安装与测试、NI-DAQmx 基础及 DAQmx 数据采集应用编程实例。

第11章介绍 LabVIEW 中信号的分析与处理, 包括信号的发生、波形调理与波形测量、信号的时域与频域分析、滤波器、窗函数及逐点分析等内容。

本书 PPT 教学课件和案例源文件可通过 <http://www.tupwk.com.cn/downpage> 下载。服务邮箱: 476371891@qq.com。

本书主要由张峤、朱世宇、孙令翠、陆鹏编写, 由谢箭主审。参加编写、校对的还有曾凡琳、张书欣, 他们在本书的编写过程中, 都做了大量的工作, 在此表示感谢。

由于编者水平有限, 书中疏漏之处在所难免, 敬请读者指正。

目 录

第	
1 章 虚拟仪器基础	1
1.1 仪器的发展	1
1.2 虚拟仪器概述	2
1.2.1 虚拟仪器的发展	2
1.2.2 虚拟仪器的概念	2
1.2.3 虚拟仪器的构成	3
1.2.4 虚拟仪器的特点	4
1.2.5 虚拟仪器的优势	4
1.3 虚拟仪器的分类	5
1.4 虚拟仪器的应用领域	6
1.5 虚拟仪器的开发环境	7
1.5.1 虚拟仪器开发软件	7
1.5.2 G语言的概念	7
习题	8
第 2 章 LabVIEW 编程环境及基本操作	9
2.1 LabVIEW 概述	9
2.1.1 什么是 LabVIEW	9
2.1.2 LabVIEW 发展史	9
2.1.3 LabVIEW 的优势	10
2.2 LabVIEW 2018 的安装	11
2.2.1 计算机环境要求	11
2.2.2 LabVIEW 2018 的安装过程	11
2.3 LabVIEW 2018 编程环境	15
2.3.1 LabVIEW 2018 的启动	15
2.3.2 LabVIEW 2018 菜单栏和	
工具栏	17
2.3.3 LabVIEW 2018 选板	27
2.3.4 LabVIEW 2018 帮助系统	30
2.4 LabVIEW 2018 的基本操作	32
2.4.1 VI 的创建	32
2.4.2 数据流的编程机制	35
2.4.3 VI 的运行与调试	36
2.4.4 LabVIEW 的初步操作	38
习题	45
第 3 章 数据类型与运算	46
3.1 基本数据类型	46
3.1.1 数值型	46
3.1.2 布尔型	51
3.1.3 枚举	53
3.1.4 时间类型	56
3.1.5 路径	58
3.2 数据运算	59
3.2.1 “数值”函数选板	59
3.2.2 “布尔”函数选板	62
3.2.3 “比较”函数选板	64
3.3 数组	67
3.3.1 数组的创建	68
3.3.2 数组的算术运算	72
3.3.3 “数组”函数及操作	72
3.4 字符串与路径	88
3.4.1 字符串控件	89

3.4.2 字符串运算	92	4.6 禁用结构	142
3.5 簇	95	4.6.1 程序框图禁用结构	142
3.5.1 簇的创建	95	4.6.2 条件禁用结构	143
3.5.2 簇函数及操作	98	4.7 公式节点	145
3.5.3 错误输入及错误输出簇	104	4.8 定时结构	147
3.6 矩阵	105	4.8.1 定时循环	147
习题	107	4.8.2 定时顺序	149
第4章 LabVIEW 的程序结构	108	习题	150
4.1 顺序结构	108	第5章 图形显示	152
4.1.1 LabVIEW程序数据流编程	108	5.1 波形显示	152
4.1.2 顺序结构的组成	109	5.1.1 波形图	153
4.1.3 顺序结构中的数据传递	112	5.1.2 波形图表	160
4.1.4 顺序结构应用举例	113	5.2 XY图与Express XY图	163
4.2 循环结构	114	5.2.1 XY图	164
4.2.1 For循环	114	5.2.2 XY图实例	165
4.2.2 While循环	119	5.2.3 Express XY图	167
4.2.3 循环结构外部与内部数据		5.3 强度图与强度图表	168
交换	120	5.3.1 强度图	168
4.2.4 自动索引	121	5.3.2 强度图表	169
4.2.5 移位寄存器及反馈节点	124	5.4 数字波形图	170
4.3 条件结构	127	5.4.1 数字数据	171
4.3.1 条件结构的组成	127	5.4.2 数字波形图	171
4.3.2 条件结构的配置及操作	128	5.5 三维图形	172
4.3.3 条件结构内部与外部的数据		5.5.1 三维曲面图	173
交换	130	5.5.2 三维参数图	174
4.3.4 条件结构应用举例	130	5.5.3 三维曲线图	175
4.4 事件结构	131	5.6 混合信号图	177
4.4.1 事件驱动概念	131	5.7 其他图形控件	178
4.4.2 事件结构的组成	132	习题	180
4.4.3 事件结构的配置与操作	133	第6章 子VI属性节点和人机界面	
4.4.4 事件结构的应用举例	135	设计	181
4.5 变量	136	6.1 子VI	181
4.5.1 局部变量	136	6.1.1 创建子VI	181
4.5.2 全局变量	138	6.1.2 定义子VI属性	183
4.5.3 使用局部变量和全局变量的		6.2 属性节点	186
注意事项	141	6.3 调用节点和引用句柄	189

6.4 人机界面设计	191	8.1.3 实例	249
6.4.1 下拉列表控件和枚举控件	191	8.2 UDP通信	251
6.4.2 列表框控件	193	8.2.1 UDP简介	251
6.4.3 表格与树形控件	197	8.2.2 UDP函数	251
6.4.4 树形控件	199	8.2.3 实例	253
6.4.5 对话框控件	202	8.3 串行通信	255
6.4.6 菜单控件	204	8.3.1 串行通信简介	255
6.4.7 选项卡控件	212	8.3.2 串行通信函数	256
6.4.8 多面板设置	213	8.3.3 实例	258
6.4.9 设置光标目标	215	8.4 LabVIEW中的其他通信	
6.5 生成.exe文件和安装文件	216	技术	259
6.5.1 生成.exe文件	216	8.4.1 共享变量	259
6.5.2 生成安装文件	219	8.4.2 LabVIEW的Web服务器	260
习题	221	8.4.3 SMTP E-mail VI	260
第7章 文件 I/O	222	8.4.4 IrDA函数	260
7.1 文件操作	222	8.4.5 蓝牙VI和函数	260
7.1.1 文件的相关概念	222	习题	260
7.1.2 文件的基本类型	223	第9章 LabVIEW 数据库编程	261
7.2 文件操作基本函数	224	9.1 LabVIEW数据库基础	261
7.2.1 打开/创建/替换文件函数	225	9.1.1 LabVIEW数据库访问方法	261
7.2.2 关闭文件函数	225	9.1.2 开放数据库互连基础	262
7.2.3 格式化写入文件函数	225	9.1.3 ADO数据访问技术	265
7.2.4 扫描文件函数	226	9.2 LabSQL数据库访问	266
7.3 文本文件	226	9.2.1 安装LabSQL	266
7.4 电子表格文件	228	9.2.2 LabSQL工具VI简介	266
7.5 二进制文件	230	9.2.3 LabSQL应用举例	268
7.6 波形文件	232	9.3 ADO数据库访问	270
7.7 测量文件	234	9.3.1 LabVIEW中对ADO的调用	270
7.8 配置文件	236	9.3.2 ADO数据库访问应用举例	272
7.9 XML文件	238	9.4 LabVIEW SQL Toolkit数据库	
7.10 TDMS文件	240	访问	274
习题	244	习题	275
第8章 网络与通信编程	246	第10章 数据采集	276
8.1 TCP通信	246	10.1 数据采集的基础	276
8.1.1 TCP简介	246	10.1.1 奈奎斯特采样定理	276
8.1.2 TCP函数	246	10.1.2 输入信号类型	277

10.1.3	信号接地与测量系统	278	11.1	信号发生	299
10.1.4	数据采集系统构成	279	11.1.1	波形生成	299
10.2	DAQ设备的安装与测试	281	11.1.2	生成信号	305
10.2.1	数据采集卡的安装	281	11.2	波形调理和波形测量	308
10.2.2	数据采集卡的测试及配置	282	11.2.1	波形调理	308
10.3	NI-DAQmx简介	285	11.2.2	波形测量	310
10.3.1	传统的NI-DAQ与NI-DAQmx	285	11.3	信号时域与频域分析	312
10.3.2	NI-DAQmx数据采集控件	286	11.3.1	信号的时域分析	312
10.3.3	NI-DAQmx数据采集VI	286	11.3.2	信号的频域分析	314
10.3.4	DAQ助手的使用	288	11.4	滤波器	319
10.4	DAQmx数据采集应用编程实例	291	11.4.1	Butterworth滤波器	320
10.4.1	模拟信号输入	291	11.4.2	贝塞尔滤波器	321
10.4.2	模拟信号输出	294	11.5	窗函数	322
10.4.3	数字I/O	295	11.5.1	信号加窗前后频谱对比实例	322
10.4.4	计数器	296	11.5.2	利用窗函数分辨小幅度信号	323
习题		298	11.6	逐点分析	324
第 11 章	信号分析与处理	299	习题		325
			参考文献		326

虚拟仪器基础

随着数字电子技术、计算机技术和数字信号处理技术的飞速发展，以及这些技术在测量领域的广泛应用，仪器技术领域发生了巨大变化。从最初的机械式仪器、模拟仪器到现在的数字化仪器、嵌入式系统仪器和智能仪器，新的测试理论、测试方法不断应用于实践，仪器技术领域的各种创新和积累都使现代测量仪器的性能发生着质的飞跃。

1.1 仪器的发展

仪器是伴随着人类测试活动的发展而发展的，从古代的曹冲称象发展到今天的虚拟仪器，大概可以分为四个阶段。

第一阶段为机械式的测量仪器，它们以物理和电磁技术为基础，例如摆钟、机械钟表、手表(上发条的表)、秒表、温度计、体温计、天平、体重秤、水表、气表、电表等。

第二阶段为模拟式电子仪器，出现于20世纪初，它是伴随着电子技术的发展而产生的，其读数由人工完成，人为误差较大，且效率低，功能单一。如石英手表(用电池供电)、石英钟、指针万用表、指针毫伏表、模拟示波器、模拟式频谱分析仪、高频信号发生器等。

第三阶段为数字式电子测量仪器，出现于20世纪50年代，伴随着数字电路技术、传感器技术以及模/数转换技术的发展而产生。数字测量仪器测量精度、测量速度有了很大的提高。例如电子表、电子秒表、电子钟、数字温度计、红外数字温度计、电子天平、电子秤、台式空气尘埃粒子计数器、血液分析仪、台式数字万用表、数字存储示波器、数字频谱分析仪、网络分析仪等，这些仪器在结构上更加复杂，功能也更加多样。

第四阶段为智能仪器，出现于20世纪60年代，是伴随着计算机技术的发展而产生的。智能仪器从单一功能的仪器发展为集控制、测量、分析、计算、显示与存储为一体的自动测试系统，将微处理器嵌入到仪器中，仪器可以独立运行程序，且具有数据存储、数据运算、逻辑判断和自动化操作等功能，具有一定的智能性，如数字电压表、数字示波器。

在智能仪器时代，虽然仪器的发展已经有了一定的智能性，但由于各仪器不能同步触发，仪器之间无法直接通信、无法传递模拟信号，在PC内仪器容易受较强的电磁干扰，PC无法满足重载仪器对电流和散热的要求等现实问题的制约，促使人们思考如何改变仪器的生产技术，如将很多个仪器集成在一起，又不受电磁干扰，且能够使它们同时工作，继而提出了一种全新的仪器概念——虚拟仪器(Virtual Instrumentation, VI)。

1.2 虚拟仪器概述

1.2.1 虚拟仪器的发展

世界上最早开发和应用虚拟仪器的公司是美国 National Instruments(NI, 国家仪器)公司。20 世纪 70 年代, 杰姆特·鲁查德和杰夫·柯德斯凯两人为美国海军研制了一种声呐测试仪。它是基于计算机的测试仪, 能为用户提供多种数据、多层次的交互式接口, 在计算机的控制下完成指定的测试工作。它还可以对系统中的可编程控制器进行编程, 配置不同的测试系统。

由于是第一次开发虚拟仪器, 成本相当高且开发的周期相当长, 用户在操作过程中要学习很多的指令, 难以在短时间内掌握。后来杰姆特·鲁查德和杰夫·柯德斯凯两人在多次研发后, 总结经验, 将一些功能进行了模块化处理, 大大简化了程序结构和操作的复杂性。这是一次里程碑式的改进, 为以后虚拟仪器的发展提供了良好的基础。

1986 年 5 月 NI 公司推出了 LabVIEW(Laboratory Virtual Instrument Engineering Workbench, 实验室虚拟仪器集成环境)的 Beta 版本作为虚拟仪器的开发环境, 同年 10 月推出了 LabVIEW 1.0 正式版。在 Windows 3.0 操作系统出现之前, LabVIEW 全是运行在 Macintosh 平台上的。当 Windows 3.0 操作系统出现后, 其良好的操作性能得到了广大用户的肯定, 并为虚拟仪器的快速发展奠定了基础。随后通过 NI 公司的研发, 虚拟仪器的功能越来越强大, 系统的性能越来越好, 一直发展到现在的 LabVIEW 2018。30 多年来 LabVIEW 从未停止过创新, 不断改进、更新和扩展, 使 LabVIEW 牢牢占据了自动化测试、测量领域的领先地位, 更改变了测试测量和控制应用系统的格局。

由于虚拟仪器具有先进的性能和广泛的应用前景, 在 NI 公司之后还有一些国际知名厂商也加入到虚拟仪器的研发中。例如, HP 公司、PC 仪器公司、Racal 公司等先后研发了一些仪器, 但 NI 公司仍然处于领先地位。

时至今日, 虚拟仪器行业发展已相当成熟, 相信它在未来有更广阔的应用前景。

1.2.2 虚拟仪器的概念

虚拟仪器的核心思想为“软件即是仪器”。传统仪器把所有软件和测量电路封装在一起, 利用仪器前面板为用户提供一组有限的功能。而 NI 公司提出, 用一台计算机, 通过软件编程得到一个类似仪器的面板, 这个软件通过 I/O 接口连接仪器硬件, 这样就形成了一个虚拟仪器系统。虚拟仪器系统展现给用户的是软件面板, 因此其功能可以完全由用户用软件自定义编写。从这一思想出发, 基于计算机、软件和 I/O 部件来构建虚拟仪器, 引起了仪器概念的巨大转变。

虚拟仪器, 就是在以计算机为核心的硬件平台上, 根据用户对仪器的设计定义, 用软件实现虚拟控制面板设计和测试功能的一种计算机仪器系统。虚拟仪器的实质是利用计算机显示器的显示功能来模拟传统仪器的控制面板, 以多种形式表达和输出检测结果; 利用计算机强大的软件功能实现信号的运算、分析、处理; 利用 I/O 接口设备完成信号的采集与调理, 从而完成各种测试功能的计算机测试系统。用户通过鼠标、键盘或触摸屏来操作虚拟面板, 就如同使用一台专用测量仪器一样, 实现所需要的测量目的。因此, 虚拟仪器的出现, 使测量仪器与计算机的界限变得模糊了。

可见,虚拟仪器是将现有的计算机技术、软件设计技术和高性能模块化硬件结合在一起而建立起来的功能强大而又灵活易变的仪器。在虚拟仪器中,硬件仅仅是为了解决信号的输入、输出和调理,软件才是整个仪器系统的关键。在不改变硬件的情况下,用户可以通过修改软件,方便地改变、增减仪器系统的功能和规模,因此在虚拟仪器中可以说“软件就是仪器”。

虚拟仪器的“虚拟”两字主要包含以下两个方面的含义。

(1) 虚拟仪器的面板是虚拟的。传统仪器通过设置在面板上的各种“开关”“旋钮”等来完成一些操作和功能,这些“开关”“旋钮”等都是实物,而且是用手动或触摸来进行操作的;而虚拟仪器面板上的“开关”“旋钮”等,它们的外形是与实物和传统仪器的“开关”“旋钮”等相像的图标,其操作通过计算机的鼠标和键盘来实现,实际功能通过相应的软件程序来实现。

(2) 虚拟仪器的测量功能是通过图形化软件流程图的编程来实现的。传统的仪器是通过设计具体的电子电路来实现仪器的测量测试及分析功能,而虚拟仪器是在以计算机为核心组成的硬件平台支持下,通过软件编程来实现仪器功能的,这也充分体现了测试技术与计算机深层次的结合。

1.2.3 虚拟仪器的构成

虚拟仪器通过应用程序将通用计算机与功能化硬件结合起来,完成对被测量数据的采集、分析、处理、显示、存储、打印等功能。因此,与传统仪器一样,虚拟仪器从功能上可划分为数据采集、数据分析、数据显示三大模块,其内部功能框图如图 1-1 所示。

其中,数据采集模块主要完成信号的调理采集;数据分析模块主要对数据进行各种分析处理;数据显示模块则将采集到的数据和分析后的结果表达出来。

从实现方法上讲,虚拟仪器由通用仪器硬件平台(简称硬件平台)和应用软件两大部分构成。

1. 虚拟仪器的硬件平台

虚拟仪器硬件的作用是获取测试对象的被测信号,由计算机和 I/O 接口设备组成。

(1) 计算机是虚拟仪器硬件平台的核心,一般为个人计算机或者工作站。

(2) I/O 接口设备是为计算机配置的电子测量仪器硬件模块,主要包括各种传感器、信号调理器、模拟/数字转换器(ADC)、数字/模拟转换器(DAC)、数据采集器(DAQ)等。

计算机及其配置的电子测量仪器硬件模块组成了虚拟仪器测试硬件平台的基础。

2. 虚拟仪器的软件

虚拟仪器软件实现数据采集、分析、处理、显示等功能,并将其集成为仪器操作与运行的命令环境。虚拟仪器软件包括接口软件、仪器驱动软件 and 应用程序。图 1-2 所示为虚拟仪器软件层次结构。

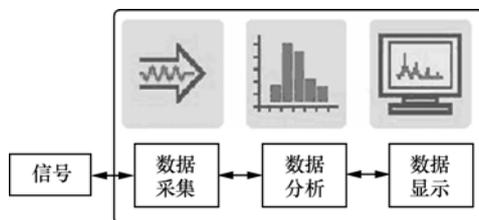


图 1-1 虚拟仪器构成方式

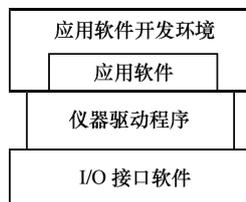


图 1-2 虚拟仪器软件层次结构

(1) 接口软件是为虚拟仪器驱动层提供信息传递的底层软件,是实现开放、灵活的虚拟仪器的基

础。接口软件的功能是直接对仪器进行控制，完成数据读/写。由于仪器硬件的种类繁多，为了保证硬件的“即插即用”，接口软件需要提供独立于硬件的 I/O 接口。

(2) 仪器驱动程序是连接虚拟仪器应用软件与接口软件的纽带和桥梁，其功能是为虚拟仪器应用软件层提供抽象的仪器操作集。对于虚拟仪器应用软件来说，对仪器的操作是通过调用虚拟仪器驱动提供的单一接口来实现的；而虚拟仪器驱动又是通过调用接口软件所提供的单一接口来实现的。

(3) 虚拟仪器应用软件直接面对操作用户，提供了快捷、友好的测控操作界面，以及图形、图表等数据显示方式。它只对虚拟仪器驱动进行调用，本身不进行任何数据处理。对于虚拟仪器应用软件的开发者来说，在不了解仪器内部操作与实现的情况下，也可以进行虚拟仪器应用软件的设计和开发。

1.2.4 虚拟仪器的特点

与传统仪器相比，虚拟仪器具有以下 3 个特点。

1. 不强调物理上的实现形式

虚拟仪器通过软件功能来实现数据采集与控制、数据处理与分析及数据显示这三部分的物理功能。它充分利用计算机系统强大的数据处理能力，在基本硬件的支持下，利用软件完成数据的采集、控制、分析和处理以及测试结果的显示等，通过软硬件的配合来实现传统仪器的各种功能。

2. 在系统内实现软硬件资源共享

虚拟仪器的最大特点是将计算机资源与仪器硬件、数字信号处理技术相结合，在系统内共享软硬件资源。它打破了以往由厂家定义仪器功能的模式，而变成了由用户自己定义仪器功能。使用相同的硬件系统，通过不同的软件编程，就可实现功能完全不同的测量仪器。

3. 图形化的软件面板

虚拟仪器没有常规仪器的控制面板，而是利用计算机强大的图形环境，采用可视化的图形编程语言和平台，在计算机屏幕上建立图形化的软面板来替代常规的传统仪器面板。软面板上具有与实际仪器相似的旋钮、开关、指示灯及其他控制部件。在操作时，用户通过鼠标或键盘操作软面板来检验仪器的通信和操作。

1.2.5 虚拟仪器的优势

除上述特点之外，与传统仪器相比，虚拟仪器还有如下 6 个方面的优势。

1. 性能高

虚拟仪器是建立在计算机平台基础之上的。随着计算机运行速度的快速发展、数据传输能力的不断加强及与 PC 总线的结合，在将数据高速导入磁盘的同时就能适时地对其进行复杂的分析计算并保存。同时越来越快的计算机网络使得虚拟仪器技术展现出其更强大的优势，使数据分享进入了一个全新的阶段。将因特网和虚拟仪器技术相结合，就能够轻松地发布测量结果。

2. 扩展性强、灵活性好

虚拟仪器的提出是针对传统仪器而言的，它们之间的最大区别是虚拟仪器完成的测量或控制任务中所需的软件和硬件设备是由用户定义的。虚拟仪器具有开放的模块化设计，用户可按自己的要求对其开发使用。其软件适应性强，只需要修改程序和部分硬件就能开发出不同的测试系统。

3. 智能化程度高

虚拟仪器相关的软件应用简单，功能强大，集成了大量常用的工具。它具有强大的数据分析、计算和图形显示等功能。虚拟仪器软件平台为所有的 I/O 设备提供了标准的接口，如数据采集、视觉、运动和分布式 I/O 等，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。

4. 界面友好

虚拟仪器采用图形化界面，在屏幕上可以直接构建出仪器面板。这些操作过程简单、快捷，仪器的功能选择、参数设置和显示等都可以通过人机对话来实现。

5. 开发时间少

高效的软件构架与计算机、仪器仪表和通信方面的最新技术结合在一起，使开发过程相当快。它提供了充分发挥个人才能和想象空间的平台，用户可编写适合自己的仪器仪表。虚拟仪器软件构架的初衷就是方便用户的灵活操作，同时提供强大的功能，让用户能轻松地配置、创建、发布和维护系统。

6. 兼容性强

虚拟仪器和传统仪器会并存一段时间，有一些场合必然要将两者结合使用，它们之间的兼容性问题已成为关注的焦点。现在的虚拟仪器可与传统仪器兼容，也就是说，虚拟仪器可以和传统仪器搭配工作。虚拟仪器提供了与常用传统仪器连接的总线，例如 USB、GPIB、串行总线和以太网等，同时也提供了大量相互连接的函数库，方便它们之间的连接。

1.3 虚拟仪器的分类

根据所使用的仪器硬件不同，虚拟仪器硬件系统可以分为 PC-DAQ 系统、GPIB 系统、VXI/PXI/LXI 系统、串口系统、现场总线系统等。

1. PC-DAQ 系统

PC-DAQ(Data Acquisition, 数据采集)系统是以数据采集板、信号调理电路和计算机为仪器硬件平台组成的插卡式虚拟仪器系统。它采用 PCI 或 ISA 计算机本身的总线，故将数据采集卡/板(DAQ)插入计算机的空槽中即可。

2. GPIB 系统

GPIB(General-Purpose Interface Bus, 通用接口总线)系统是以 GPIB 标准总线仪器和计算机为仪器硬件平台组成的虚拟仪器测试系统。典型的 GPIB 测试系统由一台计算机、一块 GPIB 接口板和几台 GPIB 仪器组成。GPIB 接口板插入计算机的插槽中，建立起计算机与具有 GPIB 接口的仪器设备之间的通信桥梁。

3. VXI/PXI/LXI 系统

VXI/PXI/LXI 系统是一类模块化的仪器系统，其硬件结构与工控机类似。每种仪器都是一个计算机插件，每种仪器都没有硬件构成的仪器面板，而由计算机显示屏幕替代。

VXI(VMEbus eXtensions for Instrumentation)总线技术出现于 20 世纪 80 年代。VXI 总线的出现

将高级测量与测试设备带入模块化领域，目前这类系统已逐渐退出市场。

PXI(PCI eXtensions for Instrumentation)总线技术出现于 20 世纪 90 年代。该总线技术是在 PCI 总线内核技术上增加了成熟的技术规范和要求而形成的，具有高度的可扩展性(有 8 个扩展槽，一般的台式 PCI 只有 3 个或 4 个扩展槽)和传输速率高的特点。PXI 系统是目前使用较多的一类模块化虚拟仪器系统。

LXI(LAN eXtensions for Instrumentation)总线技术出现于 2004 年，是继 GPIB 技术、VXI/PXI 技术之后的新一代基于以太网(LAN)的自动测试系统模块化构架平台标准。以太网的错误检测、故障定位、长距离互联、树状拓扑结构以及网络传输速率等都比现有的总线技术优越。因此，LXI 系统可能成为虚拟仪器系统发展的主流方向。

4. 串口系统

串口系统是以 Serial(串口)标准总线仪器和计算机为仪器硬件平台组成的虚拟仪器测试系统。

5. 现场总线系统

现场总线系统以 Field Bus(现场总线)标准总线仪器及计算机为仪器硬件平台，具有可靠性高、稳定性好、抗干扰能力强、通信速率快、造价低及维护成本低等优点。

无论上述哪种虚拟仪器系统，都是通过应用软件将仪器硬件与通用计算机相结合的。其中，PC-DAQ 测量系统是构成虚拟仪器的最基本的方式，也是较为廉价的方式。

1.4 虚拟仪器的应用领域

虚拟仪器目前已在航天航空、教学、核工业、军工、通信测试、铁道、汽车、电子产品等各行各业得到了广泛应用，其中在以下几个方面尤为突出。

(1) 测试测量：LabVIEW 最初就是为测试测量而设计的，因而测试测量是现在 LabVIEW 最广泛的应用领域。经过多年的发展，LabVIEW 在测试测量领域获得了广泛的承认。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 WDSC。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，如果使用 LabVIEW 进行软件模拟，就可以达到同样的效果，使学生不致失去实践的机会。

(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”：把不同的原件搭在一起，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搭建成各种车辆模型、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学习使用的版本。

(5) 快速开发：根据笔者参与的一些项目统计，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，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，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 的代码无须任何修改就可以在常见的三大台式机操作系统(Windows、Mac 及 Linux)上运行。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(Route Target, 路由目标)设备。

同时，虚拟仪器在高校某些较难理解的专业课教学中也得到了较好的应用。这些课程在传统教学中由于实验室设备和条件的限制，很难培养学生的实际动手能力和创新能力，而虚拟仪器代替了传统实验仪器，用软件模拟实验过程，使学生有更多的时间理解原理和掌握设计，无须花太多时间在前期准备和编程上，达到了事半功倍的效果。

1.5 虚拟仪器的开发环境

1.5.1 虚拟仪器开发软件

虚拟仪器应用软件开发环境是设计虚拟仪器所必需的软件工具。应用软件开发环境的选择，以开发人员的喜好不同而不同，但最终都必须提供给用户一个界面友好、功能强大的应用程序。软件在虚拟仪器中处于重要的地位，它担负着对数据进行分析处理的任务，如数字滤波、频谱变换等。在很大程度上，虚拟仪器能否成功运行，就取决于软件。因此，美国 NI 公司提出了“软件就是仪器”的口号。

目前已有多种虚拟仪器的软件开发工具，主要分为以下两类。

(1) 传统的文本式编程方法，如 C、Visual C++、Visual Basic、LabWindows/CVI 等。

(2) 图形化编程方法，如 NI 公司的 LabVIEW 软件、HP 公司的 VEE 等。使用图形化软件编程的优势是软件开发周期短、编程容易，特别适合于不具有专业编程水平的工程技术人员。

1.5.2 G 语言的概念

“G”语言中的“G”为“Graph”，其编程理念是基本上不写程序代码，取而代之的是流程图，产生的程序都是框图的形式。它尽可能利用了技术人员、科研人员、工程师所熟悉的术语、图标和概念，是一种适合于任何编程任务，具有扩展函数库的通用编程语言。G 语言与传统高级编程语言最大的差别在于编程方式，一般高级语言采用文本编程，而 G 语言采用图形化编程方式。G 语言定义了数据模型、结构类型和模块调用语法规则等编程语言的基本要素，在功能完整性和应用灵活性方面不逊于任何高级编程语言。同时，G 语言有丰富的扩展函数库，这些扩展函数库主要面向数据

采集、GPIB 和串行仪器控制、数据分析、数据显示与数据存储。G 语言还包括常用的程序调试工具，比如单步调试、允许设置断点、数据探针和动态显示程序执行流程等功能。

使用 G 语言编程方法的 LabVIEW 是一种图形化的程序语言，是一个面向最终用户的工具。它可以增强构建自己的学科和工程系统的能力，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。

利用 LabVIEW 的动态连续跟踪方式可以连续、动态地观察程序中的数据及其变化情况，用 LabVIEW 编程的过程就像设计电路图一样，因此，LabVIEW 比其他语言的开发环境更方便、更有效。使用 LabVIEW 编写的程序称为虚拟仪器，因为它的界面和功能与真实仪器十分相像，在 LabVIEW 环境下开发的应用程序都被冠以“.vi”的扩展名，以表示虚拟仪器的含义。

现今绝大多数虚拟仪器的开发环境使用 LabVIEW，因此从第 2 章开始将介绍 LabVIEW 的相关知识。

习题

1. 简述仪器和虚拟仪器的相同点和不同点。
2. 简述虚拟仪器中虚拟的含义和特点。
3. 简述虚拟仪器的分类。
4. 虚拟仪器的应用领域有哪些？
5. 简述虚拟仪器的软件开发环境。