



# LabVIEW 2018

## 虚拟仪器程序设计

毛琼 王敏 等编著

第②版



- 全书所有实例教学视频
- 视频总时长超过400分钟
- 仿真程序源代码文件



机械工业出版社  
CHINA MACHINE PRESS

EDA 工程与应用丛书

# LabVIEW 2018 虚拟仪器程序设计

第2版

毛 琼 王 敏 等编著



机械工业出版社

本书中心明确，结构紧凑，思路清晰，通过理论与实例相结合的方式，深入浅出地介绍了 LabVIEW 2018 的使用方法和使用技巧。

全书共 9 章，内容包括 LabVIEW 概述、图形编辑环境、前面板的设计、程序框图设计基础、程序结构、数据函数、文件操作、数据分析以及数学计算。本书每章都配有必要的实例，便于读者结合实例更加快捷地掌握 LabVIEW 的编程方法。

本书面向 LabVIEW 初、中级用户编写，旨在帮助读者用较短的时间快速熟练地掌握 LabVIEW 虚拟仪器设计的技巧和方法，提高读者的实践能力，达到所学即所用、一学即会的目的。

## 图书在版编目 (CIP) 数据

LabVIEW 2018 虚拟仪器程序设计 / 毛琼等编著. —2 版. —北京: 机械工业出版社, 2018.10

(EDA 工程与应用丛书)

ISBN 978-7-111-61164-6

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

中国版本图书馆 CIP 数据核字 (2018) 第 238557 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 尚 晨 责任校对: 张艳霞

责任印制: 张 博

三河市宏达印刷有限公司印刷

2018 年 11 月第 2 版·第 1 次印刷

184mm×260mm·25.25 印张·616 千字

0001—3000 册

标准书号: ISBN 978-7-111-61164-6

定价: 79.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

服务咨询热线: (010) 88361066

机工官网: [www.cmpbook.com](http://www.cmpbook.com)

读者购书热线: (010) 68326294

机工官博: [weibo.com/cmp1952](http://weibo.com/cmp1952)

(010) 88379203

教育服务网: [www.cmpedu.com](http://www.cmpedu.com)

封面无防伪标均为盗版

金书网: [www.golden-book.com](http://www.golden-book.com)

# 前 言

虚拟仪器实际上是一个按照仪器需求组织起来的数据采集系统。虚拟仪器的研究中涉及的基础理论主要有计算机数据采集和数字信号处理。目前在这一领域内,使用较为广泛的计算机语言是美国 NI 公司的 LabVIEW 软件。

虚拟仪器的起源可以追溯到 20 世纪 70 年代,当时计算机测控系统在国防、航天等领域已经有了相当不错的发展。PC 的出现使仪器级的计算机化成为可能,甚至在 Microsoft 公司的 Windows 诞生之前,NI 公司已经在 Macintosh 计算机上推出了 LabVIEW 早期版本。

对虚拟仪器和 LabVIEW 进行的长期、系统、有效的研发使 NI 公司成为业界公认的权威。LabVIEW 是图形化开发环境语言,又称 G 语言,它结合了图形化编程方式的高性能与灵活性,以及专为测试测量与自动化控制应用设计的高性能模块及功能,能为数据采集、仪器控制、测量分析与数据显示等各种应用提供必要的开发工具。

LabVIEW 2018 简体中文版是 NI 公司发布的最新中文版本。它的发布大大缓解了软件易用性和强大功能之间的矛盾,为工程师提供了效率与性能俱佳的开发平台。它适用于各种测量和自动化领域,而且,无论工程师是否有丰富的开发经验,都能顺利应用。

本书在编写过程中详细介绍了学习 LabVIEW 所要注意的问题,使读者更加深刻地理解各种函数与 VI,以“知识点——实例——知识点——实例”的形式介绍全书内容,以理论构建主干,以实例填补枝蔓,内容丰富全面,并充满实战性,有利于读者全面地掌握本书所介绍的内容,锻炼实际操作能力。

本书主要面向 LabVIEW 的初、中级用户,可作为大、中专院校相关专业的教学和参考用书,也可供有关工程技术人员和软件工程师参考。

本书由解放军陆军工程大学石家庄校区的毛琼老师和石家庄三维书屋文化传播有限公司的王敏老师编著,其中毛琼编写了第 1~8 章,王敏编写了第 9 章。张辉、赵志超、徐声杰、朱玉莲、赵黎黎、王正军、宫鹏涵、李兵、吴秋彦、闫聪聪、解江坤、张亭和井晓翠等也参加了部分章节的编写工作。

由于时间仓促,加上编者水平有限,书中存在不足之处在所难免,欢迎读者加入学习交流 QQ 群(654532572),登录网站 [www.sjzswsw.com](http://www.sjzswsw.com) 或者联系 [win760520@126.com](mailto:win760520@126.com) 沟通交流,欢迎批评指正,编者将不胜感激。

编 者

# 目 录

## 前言

|                         |    |
|-------------------------|----|
| 第 1 章 LabVIEW 概述        | 1  |
| 1.1 虚拟仪器                | 1  |
| 1.1.1 概念                | 1  |
| 1.1.2 开发环境              | 2  |
| 1.1.3 组成                | 2  |
| 1.2 LabVIEW 简介          | 4  |
| 1.2.1 LabVIEW 概述        | 4  |
| 1.2.2 LabVIEW 2018 的新功能 | 5  |
| 1.2.3 LabVIEW 的使用       | 8  |
| 1.3 LabVIEW 的安装         | 8  |
| 1.4 LabVIEW 应用程序        | 14 |
| 1.4.1 前面板               | 14 |
| 1.4.2 程序框图              | 15 |
| 1.4.3 图标/连接器            | 16 |
| 第 2 章 图形编辑环境            | 17 |
| 2.1 文件管理                | 17 |
| 2.1.1 新建 VI             | 17 |
| 2.1.2 编辑 VI 图标          | 18 |
| 2.1.3 保存 VI             | 19 |
| 2.1.4 新建文件              | 19 |
| 2.1.5 创建项目              | 20 |
| 2.2 LabVIEW 操作选板        | 21 |
| 2.2.1 控件选板              | 21 |
| 2.2.2 工具选板              | 21 |
| 2.2.3 函数选板              | 22 |
| 2.2.4 选板可见性设置           | 22 |
| 2.3 项目浏览器               | 23 |
| 2.4 菜单设计                | 25 |
| 2.4.1 菜单编辑器             | 25 |
| 2.4.2 实例——自定义菜单系统       | 27 |
| 2.4.3 “菜单”函数            | 29 |
| 2.5 控件                  | 30 |

|              |                   |           |
|--------------|-------------------|-----------|
| 2.5.1        | 新式控件              | 31        |
| 2.5.2        | NXG 风格控件          | 36        |
| 2.5.3        | 经典控件              | 38        |
| 2.5.4        | 银色控件              | 41        |
| 2.5.5        | 系统控件              | 43        |
| 2.5.6        | Express 控件        | 45        |
| 2.5.7        | .NET 与 ActiveX 控件 | 46        |
| 2.5.8        | 用户控件              | 47        |
| <b>第 3 章</b> | <b>前面板的设计</b>     | <b>48</b> |
| 3.1          | 前面板组成             | 48        |
| 3.1.1        | 数值、布尔、字符串与路径      | 48        |
| 3.1.2        | 实例——气温测试系统        | 51        |
| 3.1.3        | 数组                | 52        |
| 3.1.4        | 簇                 | 53        |
| 3.1.5        | 实例——簇数组筛选         | 55        |
| 3.1.6        | 图形                | 56        |
| 3.1.7        | 三维图形              | 64        |
| 3.1.8        | 极坐标图              | 72        |
| 3.1.9        | 实例——信号生成系统        | 73        |
| 3.2          | 对象的选择与删除          | 74        |
| 3.2.1        | 选择对象              | 74        |
| 3.2.2        | 删除对象              | 76        |
| 3.2.3        | 变更对象位置            | 76        |
| 3.3          | 对象属性编辑            | 77        |
| 3.3.1        | 设置数值型控件的属性        | 77        |
| 3.3.2        | 设置文本型控件的属性        | 80        |
| 3.3.3        | 设置布尔型控件的属性        | 81        |
| 3.3.4        | 设置图形显示控件的属性       | 82        |
| 3.3.5        | 实例——波形比较          | 85        |
| 3.4          | 设置前面板的外观          | 87        |
| 3.4.1        | 改变对象的大小           | 87        |
| 3.4.2        | 改变对象颜色            | 90        |
| 3.4.3        | 设置对象的字体           | 90        |
| 3.4.4        | 在窗口中添加标签          | 91        |
| 3.4.5        | 对象编辑窗口            | 91        |
| 3.4.6        | 实例——设计计算机控件       | 93        |
| 3.5          | 设置对象的位置关系         | 95        |
| 3.5.1        | 对齐关系              | 95        |
| 3.5.2        | 分布对象              | 96        |

|              |                       |            |
|--------------|-----------------------|------------|
| 3.5.3        | 改变对象在窗口中的前后次序 .....   | 98         |
| 3.5.4        | 组合与锁定对象 .....         | 99         |
| 3.5.5        | 网格排布 .....            | 99         |
| 3.6          | 综合实例——车速实时记录系统 .....  | 100        |
| <b>第 4 章</b> | <b>程序框图设计基础 .....</b> | <b>105</b> |
| 4.1          | 程序框图结构 .....          | 105        |
| 4.2          | 工具选板 .....            | 107        |
| 4.2.1        | 使用断点 .....            | 107        |
| 4.2.2        | 使用探针 .....            | 108        |
| 4.3          | 数学函数与 VI .....        | 108        |
| 4.3.1        | 数值函数 .....            | 108        |
| 4.3.2        | 初等与特殊函数 .....         | 111        |
| 4.3.3        | 函数快捷命令 .....          | 112        |
| 4.3.4        | 实例——颜色数值转换系统 .....    | 113        |
| 4.4          | 结构 VI 和函数 .....       | 115        |
| 4.4.1        | 分类 .....              | 115        |
| 4.4.2        | 多态性 .....             | 115        |
| 4.5          | VI 的设计 .....          | 116        |
| 4.5.1        | 创建 VI 前面板 .....       | 116        |
| 4.5.2        | 创建程序框图 .....          | 117        |
| 4.5.3        | 对象连接 .....            | 118        |
| 4.5.4        | 运行 VI .....           | 120        |
| 4.5.5        | 设置图标 .....            | 120        |
| 4.5.6        | 实例——日历 .....          | 123        |
| 4.6          | 调试 VI .....           | 127        |
| 4.6.1        | 纠正 VI 的错误 .....       | 127        |
| 4.6.2        | 高亮显示程序执行过程 .....      | 128        |
| 4.6.3        | 单步通过 VI 及其子 VI .....  | 128        |
| 4.7          | 子 VI .....            | 130        |
| 4.7.1        | 创建子 VI .....          | 130        |
| 4.7.2        | 连线端口 .....            | 131        |
| 4.7.3        | 调用子 VI .....          | 133        |
| 4.7.4        | 实例——数字遥控灯系统 .....     | 135        |
| 4.8          | 性能和内存信息 .....         | 139        |
| 4.9          | 提高 VI 的执行速度 .....     | 140        |
| 4.10         | 减少 VI 内存的使用 .....     | 144        |
| 4.11         | 搜索控件、VI 和函数 .....     | 154        |
| 4.12         | 属性节点 .....            | 155        |
| 4.13         | 综合实例——血压测试系统 .....    | 157        |

|       |                    |     |
|-------|--------------------|-----|
| 第 5 章 | 程序结构               | 161 |
| 5.1   | 循环结构               | 161 |
| 5.1.1 | For 循环             | 161 |
| 5.1.2 | While 循环           | 163 |
| 5.1.3 | 实例——公务卡管理系统        | 164 |
| 5.2   | 结构函数传递数据           | 168 |
| 5.2.1 | 反馈节点               | 168 |
| 5.2.2 | 移位寄存器              | 169 |
| 5.2.3 | 实例——延迟波形           | 172 |
| 5.3   | 层次结构               | 175 |
| 5.3.1 | 条件结构               | 175 |
| 5.3.2 | 实例——LED 控制         | 177 |
| 5.3.3 | 顺序结构               | 181 |
| 5.3.4 | 事件结构               | 183 |
| 5.3.5 | 程序框图禁用结构           | 186 |
| 5.3.6 | 条件禁用结构             | 187 |
| 5.4   | 定时循环               | 187 |
| 5.4.1 | 定时循环和定时顺序结构        | 187 |
| 5.4.2 | 配置定时循环和定时顺序结构      | 189 |
| 5.4.3 | 同步开始定时结构和中止定时结构的执行 | 193 |
| 5.5   | 公式节点               | 194 |
| 5.6   | 变量                 | 196 |
| 5.6.1 | 共享变量               | 196 |
| 5.6.2 | 局部变量               | 199 |
| 5.6.3 | 全局变量               | 201 |
| 5.7   | 综合实例——全局变量的使用      | 202 |
| 第 6 章 | 数据函数               | 205 |
| 6.1   | 数组函数               | 205 |
| 6.1.1 | 数组大小               | 206 |
| 6.1.2 | 创建数组               | 206 |
| 6.1.3 | 一维数组排序             | 208 |
| 6.1.4 | 索引数组               | 209 |
| 6.1.5 | 初始化数组              | 210 |
| 6.1.6 | 替换数组子集             | 211 |
| 6.1.7 | 删除数组元素             | 211 |
| 6.1.8 | 实例——仿真显示           | 212 |
| 6.1.9 | 实例——选项卡数组          | 214 |
| 6.2   | 簇函数                | 217 |
| 6.2.1 | 解除捆绑和按名称解除捆绑       | 218 |

|       |                          |     |
|-------|--------------------------|-----|
| 6.2.2 | 捆绑函数                     | 219 |
| 6.2.3 | 按名称捆绑                    | 220 |
| 6.2.4 | 创建簇数组                    | 221 |
| 6.2.5 | 簇至数组转换和数组至簇转换            | 222 |
| 6.2.6 | 变体函数                     | 223 |
| 6.2.7 | 实例——矩形的绘制                | 226 |
| 6.3   | 基本波形函数                   | 231 |
| 6.3.1 | 获取波形成分                   | 232 |
| 6.3.2 | 创建波形                     | 233 |
| 6.3.3 | 设置波形函数和获取波形函数            | 233 |
| 6.3.4 | 索引波形数组函数                 | 234 |
| 6.3.5 | 获取波形子集函数                 | 234 |
| 6.3.6 | 实例——不同数据创建波形             | 234 |
| 6.3.7 | Express 函数               | 237 |
| 6.4   | 综合实例——使用 Express VI 生成曲线 | 243 |
| 第 7 章 | 文件操作                     | 249 |
| 7.1   | 文件                       | 249 |
| 7.1.1 | 文件的类型                    | 249 |
| 7.1.2 | 路径                       | 251 |
| 7.1.3 | 实例——打开文件                 | 252 |
| 7.2   | 文件操作的 VI 和函数             | 253 |
| 7.2.1 | 用于常用文件 I/O 操作的 VI 和函数    | 254 |
| 7.2.2 | 文件常量                     | 263 |
| 7.2.3 | 配置文件 VI                  | 263 |
| 7.2.4 | TDM 流                    | 265 |
| 7.2.5 | 存储/数据插件                  | 273 |
| 7.2.6 | Zip                      | 274 |
| 7.2.7 | XML                      | 275 |
| 7.2.8 | 波形文件 I/O 函数              | 279 |
| 7.2.9 | 高级文件 I/O 函数              | 280 |
| 7.3   | 文件的输入与输出                 | 282 |
| 7.3.1 | 文本文件的写入与读取               | 282 |
| 7.3.2 | 电子表格文件的写入与读取             | 284 |
| 7.3.3 | 二进制文件的写入与读取              | 288 |
| 7.3.4 | 数据记录文件的创建和读取             | 289 |
| 7.3.5 | 测量文件的写入与读取               | 290 |
| 7.3.6 | 配置文件的创建与读取               | 292 |
| 7.3.7 | 记录前面板数据                  | 295 |
| 7.3.8 | 数据与 XML 格式间的相互转换         | 296 |

|              |                |            |
|--------------|----------------|------------|
| 7.4          | 综合实例——编辑选中文件   | 298        |
| <b>第 8 章</b> | <b>数据分析</b>    | <b>302</b> |
| 8.1          | 数据采集基础         | 302        |
| 8.1.1        | DAQ 功能概述       | 302        |
| 8.1.2        | NI-DAQ 安装      | 304        |
| 8.1.3        | 安装设备和接口        | 308        |
| 8.2          | 数据采集节点介绍       | 310        |
| 8.2.1        | DAQ 节点常用的参数简介  | 310        |
| 8.2.2        | DAQmx 节点       | 312        |
| 8.3          | 波形分析           | 320        |
| 8.3.1        | 波形生成           | 321        |
| 8.3.2        | 波形调理           | 327        |
| 8.3.3        | 波形测量           | 336        |
| 8.4          | 信号分析           | 344        |
| 8.4.1        | 信号生成           | 344        |
| 8.4.2        | 信号运算           | 346        |
| 8.5          | 其余 VI          | 349        |
| 8.5.1        | 窗              | 349        |
| 8.5.2        | 滤波器            | 349        |
| 8.5.3        | 谱分析            | 352        |
| 8.5.4        | 变换             | 352        |
| 8.5.5        | 逐点             | 353        |
| 8.6          | 综合实例——火车故障检测系统 | 353        |
| <b>第 9 章</b> | <b>数学计算</b>    | <b>361</b> |
| 9.1          | 数学函数           | 361        |
| 9.2          | 初等与特殊函数和 VI    | 361        |
| 9.2.1        | 三角函数           | 361        |
| 9.2.2        | 指数函数           | 364        |
| 9.2.3        | 双曲函数           | 366        |
| 9.2.4        | 离散数学           | 367        |
| 9.2.5        | 贝塞尔曲线          | 367        |
| 9.2.6        | Gamma 函数       | 368        |
| 9.2.7        | 超几何函数          | 368        |
| 9.2.8        | 椭圆积分函数         | 369        |
| 9.2.9        | 指数积分函数         | 369        |
| 9.2.10       | 误差函数           | 369        |
| 9.2.11       | 椭圆与抛物函数        | 370        |
| 9.3          | 线性代数 VI        | 371        |
| 9.3.1        | 矩阵             | 371        |

9.3.2 矩阵范数.....373

9.4 拟合 VI .....373

9.5 内插与外推 VI.....375

9.6 概率与统计 VI.....378

9.7 最优化 VI .....380

9.8 微分方程 VI .....381

9.9 多项式 VI .....381

9.10 综合实例——预测成本.....382

附录 LabVIEW 快捷键汇总 .....385

参考文献 .....391

# 第1章 LabVIEW 概述

LabVIEW 是搭建虚拟仪器系统的软件，若想连接虚拟仪器，必须学习 LabVIEW 基本的知识。

本章首先介绍了虚拟仪器系统的基本概念、组成与特点，然后介绍 LabVIEW 虚拟仪器软件的简介与安装，最后介绍虚拟仪器系统 LabVIEW 软件的应用程序环境。



## 学习要点

- 虚拟仪器
- LabVIEW 简介
- LabVIEW 的安装
- LabVIEW 应用程序

## 1.1 虚拟仪器

随着计算机技术、大规模集成电路技术和通信技术的飞速发展，仪器技术领域发生了巨大的变化。从最初的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；新的测试领域随着学科门类的交叉发展不断涌现；仪器结构也随着设计思想的更新不断发展。仪器技术领域的各种创新积累使现代测量仪器的性能发生了质的飞跃，导致了仪器的概念和形式发生了突破性的变化，出现了一种全新的仪器概念——虚拟仪器（Virtual Instrument）。

虚拟仪器把计算机技术、电子技术、传感器技术、信号处理技术和软件技术结合起来，除继承传统仪器的功能外，还增加了许多传统仪器所不能及的先进功能。虚拟仪器的最大特点是其灵活性，用户在使用过程中可以根据需要添加或删除仪器功能，以满足各种需求和各种环境，并且能充分利用计算机丰富的软硬件资源，突破传统仪器在数据处理、表达、传送以及存储方面的限制。

### 1.1.1 概念

虚拟仪器是指通过应用程序将计算机与功能化模块结合起来，用户可以通过直观、友好的图形界面来操作计算机，就像在操作自己定义、自己设计的仪器一样，从而完成对被测量的采集、分析、处理、显示、存储和打印。

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、分析和处理；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。使用者用鼠标或键盘操作虚拟面板，就如同使用一台专用的测量仪器一样。因此，虚拟仪器的出现，使测

量仪器与计算机的界限模糊了。

虚拟仪器的“虚拟”两字主要包含以下两方面的含义。

1) 虚拟仪器面板上的各种“图标”与传统仪器面板上的各种“器件”所完成的功能是相同的：由各种开关、按钮、显示器等图标实现仪器电源的“通”“断”，实现被测信号的“输入通道”“放大倍数”等参数的设置，以及实现测量结果的“数值显示”“波形显示”等。

传统仪器面板上的器件都是实物，而且是由手动和触摸进行操作的；虚拟仪器前面板是外形与实物相像的“图标”，每个图标的“通”“断”“放大”等动作通过用户操作计算机鼠标或键盘来完成。因此，设计虚拟仪器前面板就是在前面板设计窗口中摆放所需要的图标，然后对图标的属性进行设置。

2) 虚拟仪器测量功能是通过图形化软件流程图的编程来实现的，虚拟仪器是在以 PC 为核心组成的硬件平台支持下，通过软件编程来实现仪器的功能。因为可以通过不同测试功能软件模块的组合来实现多种测试功能，所以在硬件平台确定后，就有“软件就是仪器”的说法。这也体现了测试技术与计算机深层次的结合。

### 1.1.2 开发环境

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，因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，它肩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，虚拟仪器能否成功的运行，都取决于软件。因此，美国 NI 公司提出了“软件就是仪器”的口号。

通常在编制虚拟仪器软件时有两种方法。一种是传统的编程方法，采用高级语言，如 VC++、VB、Delphi 等；另一种是采用流行的图形化编程方法，如采用 NI 公司的 LabVIEW、LabWindows/CVI 软件以及 HP 公司的 VEE 等软件进行编程。使用图形化软件编程的优势是软件开发周期短，编程容易，适用于不具有专业编程水平的工程技术人员。

虚拟仪器系统的软件主要包括仪器驱动程序、应用程序和软面板程序。仪器驱动程序主要用来初始化虚拟仪器，设定特定的参数和工作方式，使虚拟仪器保持正常的工作状态。应用程序主要对采集来的数据信号进行分析处理，用户可以通过编制应用程序来定义虚拟仪器的功能。软面板程序用来提供用户与虚拟仪器的接口，它可以在计算机屏幕上生成一个和传统仪器面板相似的图形界面，用于显示测量和处理的结果，另一方面，用户也可以通过控制软面板上的开关和按钮，模拟传统仪器的操作，通过键盘和鼠标，实现对虚拟仪器系统的控制。

### 1.1.3 组成

从功能上来说，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，完成对被测量的采集、分析、处理、显示、存储和打印等功能，因此，与传统仪器一样，虚拟仪器同样划分为数据采集、数据分析处理、结果表达三大功能模块。图 1-1 所示为其内部功能框图。虚拟仪器以透明的方式把计算机资源和仪器硬件的测试能力结合起来，实现了仪器的功能。

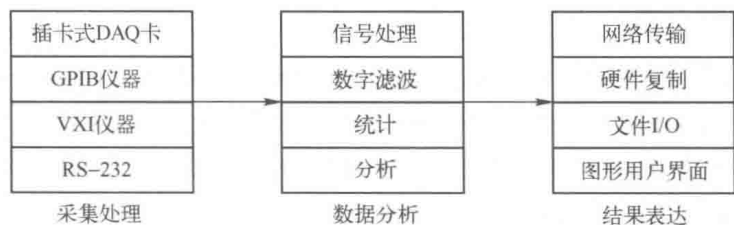


图 1-1 虚拟仪器构成方式

图 1-1 中采集处理模块主要完成数据的调理采集；数据分析模块对数据进行各种分析处理；结果表达模块则将采集到的数据和分析后的结果表达出来。

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。其结构框图如图 1-2 所示。

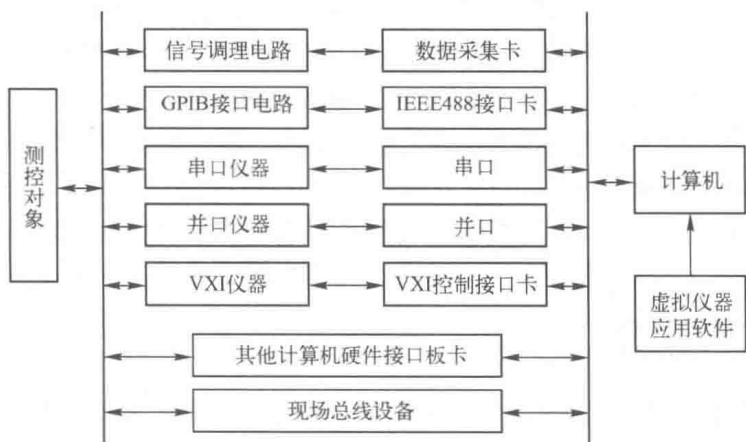


图 1-2 虚拟仪器结构框图

## 1. 硬件平台

虚拟仪器的硬件平台由计算机和 I/O 接口设备组成。

1) 计算机是硬件平台的核心，一般为一台 PC 或者工作站。

2) I/O 接口设备主要完成被测输入信号的放大、调理、模数转换和数据采集。可根据实际情况采用不同的 I/O 接口硬件设备，如数据采集卡（DAQ）、GPIB 总线仪器、VXI 总线仪器和串口仪器等。虚拟仪器构成方式有五种类型，如图 1-3 所示。无论哪种 VI 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。

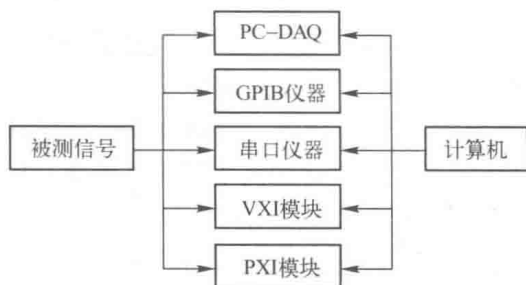


图 1-3 虚拟仪器构成方式

## 2. 软件平台

虚拟仪器软件将可选硬件（如 DAQ、GPIB、RS232、VXI 和 PXI）和可以重复使用源代码库函数的软件结合起来，实现模块间的通信、定时与触发，源代码库函数为用户构造自己的虚拟仪器系统提供了基本的软件模块。当用户的测试要求变化时，可以方便地由用户自己来增减软件模块，或重新配置现有系统以满足其测试要求。

虚拟仪器软件包括应用程序和 I/O 接口设备驱动程序。

### （1）应用程序

1) 实现虚拟仪器前面板功能的软件程序，即测试管理层，是用户与仪器之间交流信息的纽带。虚拟仪器在工作时利用软面板去控制系统。与传统仪器前面板相比，虚拟仪器软面板的最大特点是软面板由用户自己定义。因此，不同用户可以根据自己的需要组成灵活多样的虚拟仪器控制面板。

2) 定义测试功能的流程图软件程序，利用计算机强大的计算能力和虚拟仪器开发软件功能强大的函数库，极大提高了虚拟仪器的数据分析处理能力。如 HP-VEE 可提供 200 种以上的数学运算和分析功能，从基本的数学运算到微积分、数字信号处理和回归分析。LabVIEW 的内置分析能力能对采集到的信号进行平滑、数字滤波、频域转换等分析处理。

### （2）I/O 接口设备驱动程序

I/O 接口设备驱动程序用来完成特定外部硬件设备的扩展、驱动与通信。

## 1.2 LabVIEW 简介

本节主要介绍了图形化编程语言 LabVIEW，并对当前最新版本 LabVIEW 2018 简体中文版的新功能和新特性进行了介绍。

### 1.2.1 LabVIEW 概述

LabVIEW 是实验室虚拟仪器集成环境（Laboratory Virtual Instrument Engineering Workbench）的简称，是美国国家仪器公司（NATIONAL INSTRUMENTS, NI）的创新软件产品，也是目前应用最广、发展最快、功能最强的图形化软件开发集成环境之一，又称为 G 语言。和 Visual Basic、Visual C++、Delphi、Perl 等基于文本型程序代码的编程语言不同，LabVIEW 采用图形模式的结构框图构建程序代码，因而，在使用这种语言编程时，基本上不需要写程序代码，取而代之的是用图标、连线构成的流程图。它尽可能地利用了开发人员、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。

LabVIEW 是一个标准的图形化开发环境，它结合了图形化编程方式的高性能与灵活性以及专为测试、测量与自动化控制应用设计的高端性能与配置功能，能为数据采集、仪器控制、测量分析与数据显示等各种应用提供必要的开发工具，因此，LabVIEW 通过降低应用系统开发与项目筹建成本帮助科学家与工程师们提高工作效率。

LabVIEW 被广泛应用于各种行业中，包括汽车、半导体、航空航天、交通运输、科学实验、电信、生物医药与电子等。无论在哪个行业，工程师与科学家们都可以使用

LabVIEW 创建功能强大的测试、测量与自动化控制系统，在产品开发中进行快速原型创建与仿真工作。在产品的生产过程中，工程师们也可以利用 LabVIEW 进行生产测试，监控各个产品的生产过程。总之，LabVIEW 可用于各行各业产品开发的阶段。

LabVIEW 的功能非常强大，它是可扩展函数库和子程序库的通用程序设计系统，不仅可以用于一般的 Windows 桌面应用程序设计，而且还提供了用于 GPIB 设备控制、VXI 总线控制、串行口设备控制，以及数据分析、显示和存储等应用程序模块，其强大的专用函数库使得它非常适合编写用于测试、测量以及工业控制的应用程序。LabVIEW 可方便地调用 Windows 动态链接库和用户自定义的动态链接库中的函数，还提供了 CIN (Code Interface Node) 节点使得用户可以使用由 C 或 C++ 语言，如 ANSI C 等编译的程序模块，使得 LabVIEW 成为一个开放的开发平台。LabVIEW 还直接支持动态数据交换 (DDE)、结构化查询语言 (SQL)、TCP 和 UDP 网络协议等。此外，LabVIEW 还提供了专门用于程序开发的工具箱，使得用户可以很方便地设置断点，动态的执行程序可以非常直观形象地观察数据的传输过程，而且可以方便地进行调试。

当人们困惑基于文本模式的编程语言，陷入函数、数组、指针、表达式乃至对象、封装、继承等枯燥的概念和代码中时，迫切需要一种代码直观、层次清晰、简单易用且功能强大的语言。G 语言就是这样一种语言，而 LabVIEW 则是 G 语言应用的杰出代表。LabVIEW 基于 G 语言的基本特征——用图标和框图产生块状程序，这对于熟悉仪器结构和硬件电路的硬件工程师、现场工程技术人员及测试技术人员来说，编程就像是设计电路图一样。因此，硬件工程师、现场技术人员及测试技术人员学习 LabVIEW 可以驾轻就熟，在很短的时间内就能够学会并应用 LabVIEW。

从运行机制上看，LabVIEW——这种语言的运行机制就宏观上讲已经不再是传统的冯·诺伊曼计算机体系结构的执行方式了。传统的计算机语言（如 C 语言）中的顺序执行结构在 LabVIEW 中被并行机制所代替；从本质上讲，它是一种带有图形控制流结构的数据流模式 (Data Flow Mode)，这种方式确保了程序中的函数节点 (Function Node)，只有在获得它的全部数据后才能够被执行。也就是说，在这种数据流程序的概念中，程序的执行是数据驱动的，它不受操作系统、计算机等因素的影响。

LabVIEW 的程序是数据流驱动的。数据流程序设计规定，一个目标只有当它的所有输入都有效时才能执行；而目标的输出，只有当它的功能完全时才是有效的。这样，LabVIEW 中被连接的方框图之间的数据流控制程序的执行次序，而不像文本程序受到行顺序执行的约束。因而，可以通过相互连接功能方框图快速简洁地开发应用程序，甚至还可以有多个数据通道同步运行。

## 1.2.2 LabVIEW 2018 的新功能

LabVIEW 2018 是 NI 公司推出的 LabVIEW 软件的最新版本，是目前功能最为强大的 LabVIEW 系列软件。

与原来的版本相比，新版本的 LabVIEW 有以下一些主要的新功能和更改。

### 1. 针对不同数据类型自定义自适应 VI

- 比较选板新增检查类型子选板。

- 使用“检查类型”VI 和函数可强制让自适应 VI (.vim) 只接受满足特定要求的数据类型。
- 使用类型专用结构可为指定数据类型自定义自适应 VI (.vim) 中的代码段。

## 2. 使用 LabVIEW 的命令行接口运行操作

LabVIEW 2018 允许使用 LabVIEW 的命令行接口 (CLI) 执行命令在 LabVIEW 中运行操作。用于 LabVIEW 的 CLI 支持以下操作：

- MassCompile: 批量编译指定目录中的文件。
- ExecuteBuildSpec: 使用指定生成规范中的设定生成应用程序、库或比特文件，并返回输出文件的路径。
- RunVI: 使用预定义连线板接口运行 VI，并返回输出或错误信息。
- CloseLabVIEW: 关闭 LabVIEW，无提示。
- (VI Analyzer 工具包) RunVIANalyzer: LabVIEW VI Analyzer 工具包中运行指定的 VI 分析器任务，并将测试报告保存到指定位置。
- (Unit Test Framework 工具包) RunUnitTests: 在 LabVIEW Unit Test Framework 工具包中对指定文件运行测试，并将 JUnit 文件保存到指定位置。

## 3. 从 LabVIEW 调用 Python 代码

互连接口选板新增 Python 子选板，从 LabVIEW 代码中可以调用 Python 代码。Python 选板包含以下函数：

- 打开 Python 会话：用特定版本的 Python 打开 Python 会话。
- Python 节点：直接调用 Python 函数。
- 关闭 Python 会话：关闭 Python 会话。

## 4. 应用程序生成器的改进

LabVIEW 2018 对 LabVIEW 应用程序生成器和程序生成规范进行了下列改进。

- 1) 在 Windows 和 Linux Real-Time 终端上创建程序包。
- 2) LabVIEW 生成的 .NET 程序集的向后兼容性支持。

## 5. 环境改进

LabVIEW 2018 包含以下对 LabVIEW 环境的改进：

### (1) 创建自定义类型的功能改进

LabVIEW 2018 提供更多创建自定义类型的方式，可将自定义控件的所有实例链接到已保存的自定义控件文件。

### (2) 用于格式化文本的键盘快捷键

在 LabVIEW 环境中编辑文本时，使用以下键盘快捷键来格式化字体样式：

- <Ctrl-B>：加粗文本。
- <Ctrl-I>：斜体文本。
- <Ctrl-U>：下划线文本。

## 6. 程序框图的改进

LabVIEW 2018 对程序框图和相关功能进行了以下改进：

### (1) 并行 For 循环中错误处理的改进

LabVIEW 2018 新增了错误寄存器以简化启用了并行循环的 For 循环的错误处理。错误