第 3 章　虚拟仪器软件开发平台 LabVIEW

构造一台虚拟仪器，基本硬件确定以后，就可以通过不同的软件实现不同的功能。软件是虚拟仪器的关键。目前流行的虚拟仪器软件开发工具有两类：文本式编程语言，有 C、C++、VB、VC、LabWindows/CVI 等；图形化编程语言，有 LabVIEW、Agilent VEE 等。其中，LabVIEW 是目前应用最广、发展最快、功能最强的图形化软件之一。

本章主要介绍图形化编程语言 LabVIEW 的概念和特点，以及 LabVIEW 2019 的编程环境与操作方法，并通过一个具体示例来说明 LabVIEW 2019 创建虚拟仪器的一般步骤。

3.1　LabVIEW 概述

3.1.1　LabVIEW 的含义

LabVIEW（Laboratory Virtual Instrument Engineering Workbench，实验室虚拟仪器集成环境）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（又称 G 语言），是由 NI 公司推出的虚拟仪器开发平台。

LabVIEW 作为一种强大的虚拟仪器开发平台，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。LabVIEW 集成了 GPIB、VXI、PXI、RS-232C、USB 的硬件和数据采集卡通信的全部功能，并且它还内置了便于应用 TCP/IP、Active X 等软件标准的库函数。因此，LabVIEW 是一个功能强大且灵活的软件，利用它可以方便地组建自己的虚拟仪器。

使用 LabVIEW 开发平台编制的程序称为 VI（Virtual Instrument），包括前面板、程序框图及图标/连线板 3 部分。LabVIEW 简化了虚拟仪器的开发过程，缩短了仪器开发和调试周期，它让用户从烦琐的计算机代码编写中解脱出来，把大部分精力投入仪器设计和分析中，而不再拘泥于程序的细节。

3.1.2　LabVIEW 的特点

LabVIEW 是一种图形化编程语言，使用这种语言编程时，基本上不用写程序代码，取而代之的是程序框图。LabVIEW 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工程系统的能力，提供了实现仪器编程和数据采集的便捷途径，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 通过图形符号来描述程序的行为，消除了令人烦恼的语法规则，减轻了用户编程的负担，提高了效率。总而言之，LabVIEW 的特点如下。

（1）图形化的编程环境

LabVIEW 的基本编程单元是图标，不同的图标表示不同的功能模块。用 LabVIEW 编写

程序的过程也就是将多个图标连接起来的过程，连线表示功能模块之间存在数据的传递。被连接的对象之间的数据流控制着执行顺序，并允许有多个数据通路同步运行。其编程过程近似人的思维过程，直观易学，编程效率高，无须编写任何文本格式的代码，易为多数工程技术人员接受。

（2）开发功能高效、通用

LabVIEW 是一个带有扩展功能库和子程序库的通用程序设计系统，提供了数百种功能模块（类似其他计算机语言的子程序或函数），包括信号采集、信号分析与处理、信号输出、数据存取、数据通信等，涵盖了仪器的各个环节，用户通过拖放及简单的连线，就可以在极短的时间内设计好一个高效的仪器程序。

（3）支持多种仪器和数据采集硬件的驱动

LabVIEW 提供了数百种仪器的源代码驱动程序，包括 DAQ、GPIB、VXI、PXI、RS-232C 等，根据需要还可以在 LabVIEW 中自行开发各种硬件驱动程序，也可以通过动态链接库（DLL）利用其他语言开发驱动函数库，从而进一步扩展其功能。

（4）查错、调试能力强大

LabVIEW 的查错、调试功能非常强大。程序查错无须先编译，只要有语法错误，LabVIEW 就会自动显示并给出错误类型、原因及准确的位置。进行程序调试时，既有传统的程序调试手段，如设置断点、单步运行等，又有独到的高亮执行工具，就像电影中的慢镜头一样，使程序动画式执行，利于设计者观察程序的运行细节。同时可以在任何位置插入任意多的数据探针，程序在调试状态下运行时，LabVIEW 会给出各种探针的具体数值，通过观察数据流的变化情况、程序运行的逻辑状态，就可以寻找错误、判断原因，从而大大缩短程序调试时间。

（5）网络功能强大

LabVIEW 支持常用网络协议，如 TCP/IP、UDP、DataSocket 等，方便远程测控系统的开发。

（6）开放性强

LabVIEW 具有很强的开放性，是一个开放的开发环境，能和第三方软件连接。通过 LabVIEW 可以把现有的应用程序和.NET 组件、ActiveX、DLL 等相连，可以和 MATLAB 混合编程，也可以在 LabVIEW 中创建能在其他软件环境中调用的独立执行程序或动态链接库。

3.1.3 LabVIEW 的发展

1986 年 10 月，NI 公司基于 Macintosh 平台正式发布了 LabVIEW 1.0，随后对编辑器、图形显示及其他细节进行重大改进，在 1990 年 1 月发布了 LabVIEW 2.0。1992 年 LabVIEW 实现了从 Macintosh 平台到 Windows 平台的移植，1993 年 1 月 LabVIEW 3.0 正式发行。此时 LabVIEW 已经成为包含了几千个 VI 的大型应用软件系统，作为一个比较完整的软件开发环境得到认可，并迅速占领市场。

1996 年 4 月 LabVIEW 4.0 问世，实现了应用程序生成器（LabVIEW Application Builder）的单独执行，并向数据采集 DAQ 通道方向进行了延伸。1998 年 2 月发布的 LabVIEW 5.0 对以前版本全面修改，对编辑器和执行系统进行了重写，尽管增加了复杂性，但也大大增强了 LabVIEW 的可靠性。1999 年 6 月，NI 公司发布了用于实时应用程序的分支——LabVIEW RT 版。

2000 年 6 月 LabVIEW 6.0 发布，LabVIEW 6.0 拥有新的用户界面特征（如 3D 显示）、扩展功能及各层内存优化，另外还具有一项重要的功能——强大 VI 服务器。2003 年 5 月发布的 LabVIEW 7 Express 引入了波形数据类型和一些交互性更强、基于配置的函数，使用户应用开发更简便，在很大程度上简化了测量和自动化应用任务的开发，并对 LabVIEW 使用范围进行扩充，实现了对 PDA 和 FPGA 等硬件的支持。2005 年发布了 LabVIEW 8.0，为分布在不同计算目标上的各种应用程序的开发和发布提供支持。

2006 年，NI 公司为庆祝和纪念 LabVIEW 正式推出 20 周年，在当年 10 月发布了 LabVIEW 的 20 周年纪念版——LabVIEW 8.2。该版本增加了仿真框图和 MathScript 节点两大功能，提升了 LabVIEW 在设计市场的地位，同时第一次推出了简体中文版，方便了中国科技人员的学习和使用。

2007 年 8 月 LabVIEW 8.5 发布。LabVIEW 8.5 凭借其本质上的并行数据流特性，简化了多核及 FPGA 应用的开发。2008 年 8 月发布了 LabVIEW 8.6，通过采用多核处理器技术提高测试及控制系统的吞吐量，在基于 FPGA 的高级控制及嵌入式原型应用中缩短开发时间，更便捷地创建分布式测量系统，采集远程数据。2009 年 8 月发布的 LabVIEW 2009，通过对软件工程过程（包括对关键测试软件的开发、发布和维护）流水线化，有效简化了复杂测试系统开发的挑战。同时，它提供了如虚拟化技术的并行编程特性，提升了基于多核测试应用的工作性能，并改进了 FPGA 编译器性能，帮助简化 FPGA 的可重复配置 I/O（RIO）的开发。此外，LabVIEW 2009 针对在统一的硬件平台上测试 WLAN、WiMAX、GPS 和 MIMO 等多种无线标准提供了新的解决方案。2010 年 8 月发布的 LabVIEW 2010，新增了即时编译技术，可将执行代码的效率提高 20%。

2011 年 8 月，备受赞誉的 LabVIEW 软件迎来了 25 周年，NI 公司发布了 LabVIEW 2011。该版本通过新的工程实例库及其对大量硬件设备和部署目标的交互支持，极大地增进了效率。LabVIEW 2011 还支持内置在最新的 Microsoft .NET 框架的组件，并且基于用户的反馈新增了多项新特性。LabVIEW 2011 能够帮助工程师将零散的系统部件集成为一个统一并可重配置的平台，从而让工程应用更高效、成本也更低。正如 LabVIEW 的发明者兼公司创始人之一 Jeff Kodosky 所说：“25 年前，我们创建了 LabVIEW 来帮助工程师从繁杂的编程及系统集成中解脱出来，将精力专注于应用和创新。而今天，LabVIEW 已经变成了针对测试测量和控制领先的图形化系统设计软件。每一次软件版本的更新，不论是无缝集成最新的硬件，还是添加新的函数库和 API，以及新增基于用户反馈的新特性，我们的主要目的依然是在所有工程环境中提高应用效率。”

2012 年 8 月，NI 公司发布了 LabVIEW 2012。2013 年 8 月发布的 LabVIEW 2013，不仅支持 NI Linux 实时操作系统，方便开发人员访问动态、社区数据库，还是全新 cRIO-9068 软件定制的控制器的基础。2014 年 8 月发布的 LabVIEW 2014，通过跨系统复用相同的代码和工程流程来标准化用户与硬件交互的方式，这一方式也使得工程师能够根据未来需求调整应用程序。2015 年 8 月发布的 LabVIEW 2015，提供了快速便捷的开发方式和调试工具，帮助开发人员更高效地与所创建的系统进行交互。2016 年 8 月发布的 LabVIEW 2016，新增了通道连线功能等，可简化并行代码之间的复杂通信，并且可以用到桌面和实时系统，有助于提高代码可读性及减少开发时间。2017 年 5 月发布的 LabVIEW 2017，增加了多个 VI 服务器对象，增加了多个 VI 脚本对象，增加了 LabVIEW 第三方许可和激活工具包。2018 年 5 月发布

的 LabVIEW 2018，进一步优化了性能，改进了生成优化机器代码的后台编译器，使代码执行速度提高了很多，启动速度比以前版本更快。

2019 年 5 月，NI 公司发布了 LabVIEW 2019。LabVIEW 2019 简化了分布式测试和控制系统的设计，可以帮助用户缩短产品上市时间。此外，NI 提供的现成硬件不仅备受肯定，而且可定制，这些硬件与 LabVIEW 2019 相结合，可以帮助用户轻松开发和部署大型工业生产系统。

LabVIEW 2019 包括以下主要功能：

- 本地软件包构建，便于代码分发；
- Run-Time 引擎可向后兼容，便于使用现有的二进制文件；
- 本地 Python 节点，便于在 LabVIEW 中调用 Python 脚本；
- 64 位的 LabVIEW FPGA 模块；
- 支持 Vivado 2017.2 FPGA 编译工具；
- LabVIEW Azure 云工具包；
- 兼容所有 NI 硬件。

3.1.4 LabVIEW 的应用

LabVIEW 在包括航空、航天、通信、汽车、交通运输、半导体、生物医学与电子等众多领域内得到了广泛应用。从简单的仪器控制、数据采集到尖端的测试和工业自动化，从大学实验室到工厂企业，从探索研究到技术集成，都有 LabVIEW 应用的成果。

1．LabVIEW 应用于测试和测量

LabVIEW 已成为测试和测量领域的工业标准，通过 GPIB、VXI、PXI、串行设备和插卡式数据采集板可以构成实际的数据采集系统。LabVIEW 提供了工业界最大的仪器驱动程序库及众多的开发工具，使复杂的测试和测量任务变得简单易行。

2．LabVIEW 应用于过程控制和工业自动化

LabVIEW 强大的硬件驱动、图形显示能力和便捷的快速程序设计为过程控制和工业自动化应用提供了优秀的解决方案。

3．LabVIEW 应用于实验室研究与计算分析

LabVIEW 为科学家和工程师提供了功能强大的高级数学分析库，包括统计、估计、回归分析、线性代数、信号生成算法、时域和频域算法等，可满足各种计算和分析需要。因此，许多大学已将 LabVIEW 作为课堂或实验室教学内容，作为工程师素质培养的一个方面。不同领域的科学家和工程师都可借助这个易用的软件包来解决工作中的各种应用问题。

3.1.5 LabVIEW 的安装和启动

LabVIEW 2019 可以安装在 Windows 10、Windows 8.1、Windows 7 SP1、Windows Server 2012 R2、Windows Server 2008 R2 等不同的操作系统上。在安装 LabVIEW 2019 时，不同的操作系统对系统的配置要求不同，用户在安装 LabVIEW 2019 前需对计算机的软/硬件环境有一定的了解。

LabVIEW 2019 的下载、安装过程十分简单，进入 NI 公司的网站，注册、登录后，在 https://www.ni.com/zh-cn/support/downloads.html 软件和驱动程序下载页面，选择 LabVIEW

2019 下载，下载 NI Package Manager，然后使用 NI Package Manager 安装 LabVIEW 2019、工具包等。

NI Package Manager 安装 LabVIEW 2019 的界面如图 3.1 所示。

图 3.1　NI Package Manager 安装 LabVIEW 2019 的界面

整个系统的安装时间取决于硬件平台和选择的安装选项。LabVIEW 2019 开发环境约占 5GB 的硬盘空间（包括 NI 设备驱动程序等）。

当 LabVIEW 2019 成功安装到计算机后，在 Windows 的开始菜单中便会自动生成启动 LabVIEW 2019 的快捷方式。例如，若选择安装了 LabVIEW 2019 SP1（32 位）中文版支持的模块和工具包，则在开始菜单中就会出现 LabVIEW 应用程序的快捷方式——NI LabVIEW 2019 SP1（32 位），单击这个快捷方式就可启动 LabVIEW 2019，启动界面如图 3.2 所示。

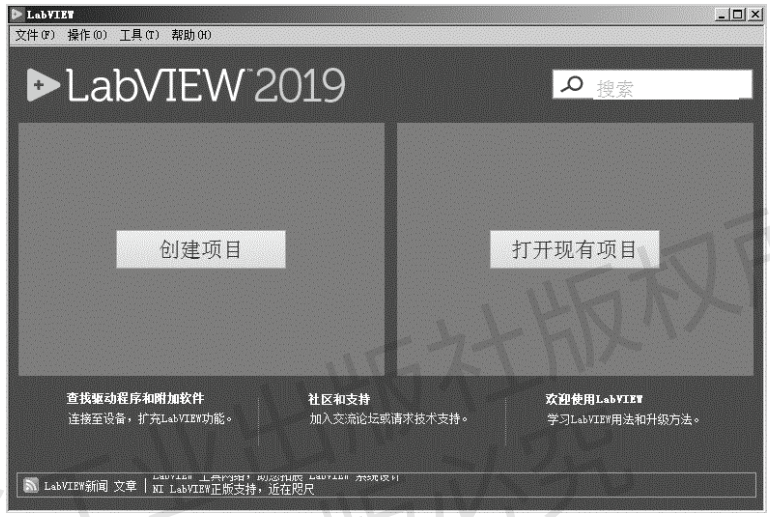


图 3.2　LabVIEW 2019 的启动界面

在 LabVIEW 2019 的启动界面中可创建新 VI（LabVIEW的程序）文件或打开现有的 LabVIEW 文件，可以基于模板或范例创建新项目，也可以通过启动界面访问 LabVIEW 的扩展资源和教程。

3.2 LabVIEW 2019 的编程环境

LabVIEW 2019 程序开发环境采用图形化的编程方式，无须编写任何代码，它不仅包含丰富的数据采集、分析及存储的库函数，还提供了 GPIB、VXI、PXI、RS-232C、USB 等通信总线标准的功能函数，可以驱动不同总线接口的设备和仪器。LabVIEW 2019 具有强大的网络功能，支持常用的网络协议，可以方便地设计开发网络测控仪器，并有多种程序调试手段，如断点设置、单步调试等。

3.2.1 LabVIEW 程序的基本构成

使用 LabVIEW 开发环境编制的程序称为 VI。VI 由以下 3 部分构成。

- 前面板：即用户界面。
- 程序框图：包含用于定义 VI 功能的图形化源代码。
- 图标和连线板：用以识别 VI 的接口，以便在创建 VI 时调用另一个 VI。当一个 VI 应用在其他 VI 中时，则称该 VI 为子 VI。子 VI 相当于文本编程语言中的子程序。

1．前面板

前面板是 VI 的用户界面。创建 VI 时，通常应先设计前面板，然后设计程序框图，执行在前面板上创建的输入/输出任务。前面板示例如图 3.3 所示。

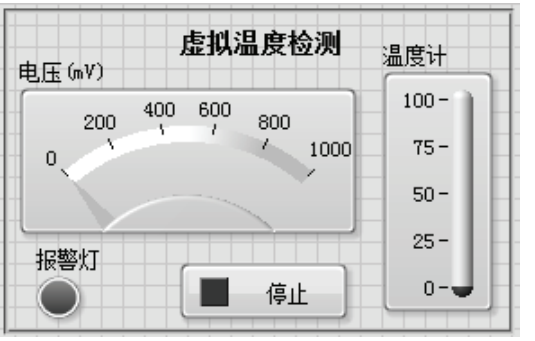


图 3.3　前面板示例

前面板上有输入控件和显示控件两类对象，用于模拟真实仪表的前面板。输入控件和显示控件用各种各样的图形形式出现在前面板上，具体表现为旋钮、按钮、图形、指示灯及其他的控制和显示对象等，这使得用户界面更加直观易懂。

2．程序框图

前面板创建完毕后，便可使用图形化的函数添加源代码来控制前面板上的对象。程序框图是图形化源代码的集合，图形化源代码又称 G 代码或程序框图代码。含有接线端、函数和连线等的程序框图示例如图 3.4 所示。

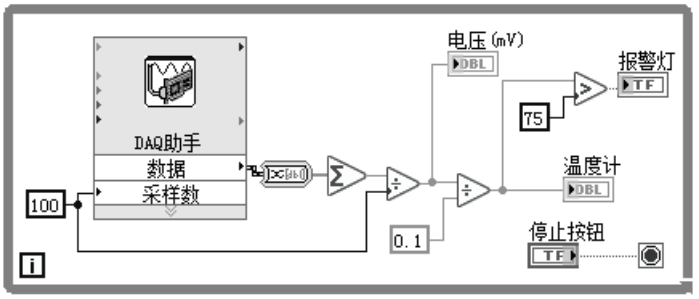


图 3.4　程序框图示例

程序框图对象包括接线端和节点，将各个对象用连线连接便创建了程序框图。接线端的颜色和符号表明了相应输入控件或显示控件的数据类型。程序框图由接线端、节点、连线和结构等构成，功能简介如下。

（1）接线端

接线端用来表示输入控件和显示控件的数据类型。在程序框图中，可将前面板的输入控件和显示控件显示为图标或数据类型接线端（如图 3.4 中的温度计、报警灯等）。接线端是在前面板和程序框图之间交换信息的输入/输出接口。

（2）节点

节点是程序框图上的对象，具有输入/输出接口，在 VI 运行时进行运算。节点相当于文本编程语言中的语句、运算符、函数和子程序。如图 3.4 中的“÷”“>”函数就是节点。

（3）连线

程序框图中对象的数据传输通过连线实现。每根连线都只有一个数据源，但可以与多个读取该数据的 VI 和函数连接。不同数据类型的连线有不同的颜色、粗细和样式。

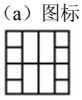
（4）结构

结构是文本编程语言中的循环和条件语句的图形化表示。使用程序框图中的结构，可以对代码块进行重复操作，如是按条件执行代码或是按特定顺序执行代码。

3．图标和连线板

创建 VI 的前面板和程序框图后，创建图标和连线板，以便将该 VI 作为子 VI 调用。图标和连线板相当于文本编程语言中的函数原型。每个 VI 都显示为一个图标，位于前面板和程序框图窗口的右上角，如图 3.5（a）所示。

（a）图标



（b）连线板

图 3.5　图标和连线板

图标是 VI 的图形化表示，可包含文字、图形或图文组合。如果将一个 VI 当作子 VI 使用，程序框图上将显示代表该子 VI 的图标，可双击图标进行修改或编辑。

若需将 VI 当作子 VI 使用，还需创建连线板，如图 3.5（b）所示。连线板用于显示 VI 中所有输入控件和显示控件的接线端，类似于文本编程语言中调用函数时使用的参数列表。连线板标明了可与该 VI 连接的输入端和输出端，以便将该 VI 作为子 VI 调用。连线板在其输入端接收数据，然后通过前面板的输入控件传输至程序框图的代码中，并从前面板的显示控件中接收运算结果并传输至其输出端。

3.2.2 LabVIEW 2019 的操作选板

设计一个 LabVIEW 应用程序，主要是利用 LabVIEW 提供的 3 个操作选板来完成。这 3 个操作选板是：工具选板、控件选板和函数选板，这些选板集中反映了 LabVIEW 的功能与特征。下面分别介绍工具选板、控件选板和函数选板。

1．工具选板

在前面板和程序框图中都可以看到工具选板，它提供了用于创建、编辑、修改前面板和程序框图中的对象，以及调试 VI 的各种工具。

图 3.6　LabVIEW 2019 的工具选板

LabVIEW 2019 的工具选板如图 3.6 所示。如果自动选择工具已打开，当光标移到前面板或程序框图的对象上时，LabVIEW 将自动从工具选板中选择相应的工具。如果打开的 VI 没有出现工具选板，在 LabVIEW 2019 的菜单中选择【查看】→【工具选板】，即可打开工具选板。当从工具选板中选择了任意一种工具后，光标就会变成该工具相应的形状。

工具选板中各工具的图标、名称及功能见表 3.1。

表 3.1　工具选板中各工具的图标、名称及功能

图 标	名 称	功　能
	自动选择工具	按下自动选择工具后，当光标在前面板或程序对象图标上移动时，系统自动从工具选板上选择相应工具，方便用户操作
	操作值工具	用于操作前面板的控制和显示
	定位/调整大小/选择工具	用于选择、移动或改变对象的位置和大小
	编辑文本工具	用于输入标签文本或创建标签
	连线工具	用于在程序框图中连接两个对象的数据端口。当使用连线工具接近对象时，会自动显示出其数据端口以供连线之用
	对象快捷菜单工具	使用该工具单击窗口任意位置，均可以弹出对象的快捷菜单
	滚动窗口工具	使用该工具可以不需要使用滚动条就可以自由滚动整个图形
	设置/清除断点工具	在调试程序过程中设置/清除断点
	探针数据工具	可以在程序框图内的数据流线上设置探针，通过探针窗口来观察该数据流线上的数据变化
	获取颜色工具	提提取对象的颜色，来编辑其他对象的颜色
	设置颜色工具	用于设置窗口中对象的前景色和背景色

2．控件选板

控件选板仅位于前面板，它包含了用于创建前面板对象所需的各种输入控件和显示控件。输入控件是指按钮、旋钮、转盘等输入装置，用来模拟仪器的输入，为 VI 的程序框图提供数据；显示控件是指图表、指示灯等显示装置，用来模拟仪器的输出，显示程序框图获取或生成的数据。

如果打开的 VI 没有出现控件选板，在 LabVIEW 2019 的菜单中选择【查看】→【控件选板】，或在前面板活动窗口单击鼠标右键，即可弹出控件选板。LabVIEW 将记住控件选板的位置和大小，因此当 LabVIEW 重启时，控件选板的位置和大小保持不变。

LabVIEW 2019 的控件选板如图 3.7 所示。

图 3.7　LabVIEW 2019 的控件选板

控件选板中控件的种类有：数值（如滑动杆和旋钮），布尔（如按钮和开关），字符串与路径，数组、矩阵与簇，列表、表格和树，图形，下拉列表与枚举，容器，I/O，变体与类，修饰，引用句柄等。控件样式有新式、NXG 风格、银色、系统和经典等。LabVIEW 2019 的新式控件子选板中的图标、名称及功能见表 3.2。

表 3.2　LabVIEW 2019 的新式控件子选板中的图标、名称及功能

图 标	名 称	功　能
	数值	提供各种数值输入和显示控件，如滑动杆、滚动条、旋钮、转盘和数值显示框等
	布尔	提供各种布尔型的输入和显示控件，包含各种开关、按钮、指示灯等

续表

图 标	名 称	功 能
	字符串与路径	提供字符串输入、显示控件和文件路径控件
	数组、矩阵与簇	用来创建数组、矩阵和簇类型的输入及显示控件
	列表、表格和树	用来创建列表、表格和树形式数据的输入及显示控件
	图形	提供以图形和图表的方式显示数值结果的控件，如波形图、波形图表、XY 图、强度图、数字波形图、三维图等控件
	下拉列表与枚举	用来创建文本下拉列表、菜单下拉列表、枚举、图片下拉列表等类型的控件
	容器	用来创建水平与垂直分隔栏、容器、子面板等控件
	I/O	用来对配置的 DAQ 通道名称、VISA 资源名称和 IVI 逻辑名称等进行传递
	变体与类	用于变体与类的数据进行交互
	修饰	用于前面板的设计和装饰，如用于装饰界面的框和线条等
	引用句柄	包含各类引用句柄控件，用于传递文件、目录、设备和网络连接等被操作对象的标识信息

3．函数选板

函数选板仅位于程序框图，它包含了编写程序过程中用到的 VI 和函数，主要用于构建程序框图中的节点，对 VI 程序框图进行设计。

LabVIEW 2019 的函数选板如图 3.8 所示。

如同控件选板一样，函数选板中所有 VI 和函数被分门别类地存放在一系列子选板中，如编程、测量 I/O、仪器 I/O、数学、信号处理、数据通信等函数子选板。表 3.3 介绍了 LabVIEW 2019 函数选板中最常用的【编程】函数子选板的图标、名称及功能。

图 3.8　LabVIEW 2019 的函数选板

表 3.3　LabVIEW 2019 函数选板中【编程】函数子选板的图标、名称及功能

图 标	名 称	功 能
	结构	用于程序的流程控制，如循环、顺序、分支等
	数组	用于数组的创建和操作，包括数组运算函数、数组转换函数，以及常数数组等
	簇、类与变体	用于簇、类与变体相关的各种函数，如簇的捆绑、解除及簇与变体和其他数据类型之间的类型转换
	数值	常用的数值计算、各种数值型数据类型之间的互相转换、复数计算和常用数学常量
	布尔	用来对单个布尔值或布尔数组进行逻辑操作
	字符串	用于对字符串型数据的操作，如搜索和替换字符串、扫描字符串等
	比较	用于对布尔值、字符串、数值、数组和簇的比较

续表

图 标	名 称	功 能
	定时	用于控制程序执行速度，包含各种定时、等待、时间类型转换函数
	对话框与用户界面	用于创建各种按钮对话框、提示对话框、显示对话框及建立菜单、帮助、事件等
	文件 I/O	用于创建、打开、读取及写入文件等，包括各种文件操作函数、路径操作函数
	波形	用于进行和波形有关的操作，如获取波形成分、创建波形等
	应用程序控制	用于打开与关闭应用程序，包括程序的停止、退出等程序控制函数
	同步	用于同步并行执行的任务并在并行任务间传递数据
	图形与声音	用于创建图形、从图形文件获取数据、对声音信息的处理等，包含各种图形图像显示、声音播放等函数
	报表生成	用于创建和控制应用程序报表，如新建报表、打印报表等

3.2.3 LabVIEW 2019 的菜单和工具栏

菜单和工具栏用于操作和修改前面板及程序框图上的对象。LabVIEW 2019 为创建的 VI 同时打开两个窗口：前面板窗口和程序框图窗口。这两个窗口具有相同的菜单和工具栏（区别在于调试功能按钮只出现在程序框图窗口中）。

1．菜单

VI 窗口顶部的菜单为通用菜单。LabVIEW 2019 的菜单包括文件、编辑、查看、项目、操作、工具、窗口、帮助 8 大项。

（1）文件菜单

主要完成 VI 文件和项目文件的新建、打开、关闭、保存，以及打印、属性设置和退出程序等操作。

（2）编辑菜单

主要完成操作撤销、选定对象的剪切、复制、粘贴、删除，从文件中导入图片，删除断线，对齐所选项，查找和替换等操作。

（3）查看菜单

用于弹出控件选板、函数选板、工具选板、错误列表、VI 层次结构、LabVIEW 类层次结构、浏览关系、书签管理、类浏览器操作及弹出启动窗口、导航窗口等操作。

（4）项目菜单

主要完成项目的创建、打开、关闭、保存、显示项目路径等操作。

（5）操作菜单

主要完成运行、停止、断点、单步、结束时打印、结束时记录、数据记录、改变运行模式和连接远程前面板等操作。

（6）工具菜单

主要完成仪器驱动程序的更新和导入、VI 性能分析、用户名设定、生成应用程序、管理远程面板的连接、网络发布工具、多种选项设定等操作。

（7）窗口菜单

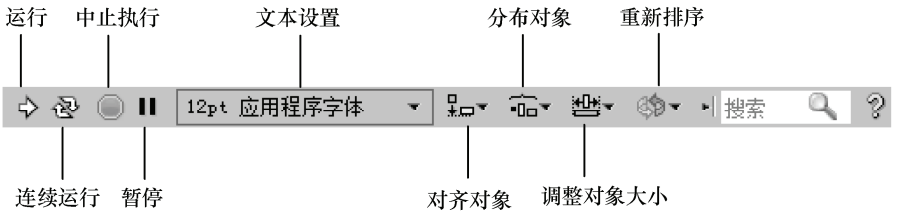
主要完成前面板/程序框图窗口切换、左右两栏显示窗口、上下两栏显示窗口和全屏显示窗口等操作。

（8）帮助菜单

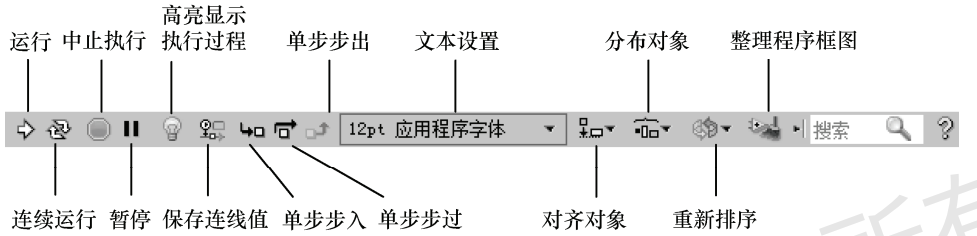
LabVIEW 2019 提供了功能强大的帮助功能，主要有及时帮助、LabVIEW 帮助、解释错误、查找范例、查找仪器驱动、检查更新等功能。

2．工具栏

工具栏按钮用于运行、中断、终止、调试 VI、修改字体、对齐、组合、分布对象等。LabVIEW 2019 的前面板窗口和程序框图窗口都有各自的工具栏，两者大多数的工具栏按钮相同，区别在于程序框图工具栏增加了几个调试功能按钮。前面板窗口和程序框图窗口工具栏如图 3.9 所示。



(a) 前面板窗口工具栏

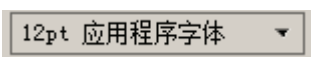


(b) 程序框图窗口工具栏

图 3.9　LabVIEW 2019 的工具栏

工具栏按钮功能见表 3.4。

表 3.4　工具栏按钮功能一览表

图标	名称	功能说明
	运行	单击此按钮，可运行当前 VI
	连续运行	单击此按钮，可重复连续运行当前 VI
	中止执行	单击此按钮，可终止当前 VI 运行
	暂停	单击此按钮，可暂停当前 VI 运行，再次单击此按钮，VI 又继续执行
	高亮显示执行过程	单击此按钮，可动态显示 VI 执行时数据的流动（仅程序框图窗口有）
	保存连线值	先单击此按钮，可使 VI 运行后为各条连线上的数据保留值（仅程序框图窗口有）
	单步步入	单击单步步入按钮时，将执行第一个操作，然后在子 VI 或结构的下一个操作前暂停（仅程序框图窗口有）
	单步步过	执行节点并在下一个节点前暂停（仅程序框图窗口有）
	单步步出	结束当前节点的操作并暂停。VI 结束操作时，单步步出按钮将变为灰色（仅程序框图窗口有）
	文本设置	用于设置文本的字体、大小和样式等
	对齐对象	可将选定的对象按某一规则对齐，对齐方式有竖直对齐、上边对齐、左边对齐等
	分布对象	用于改变界面上对象的分布方式
	调整对象大小	用于将前面板上的对象调整为相同大小（仅前面板窗口有）
	重新排序	为选定对象重新设定在窗口中的前后叠放顺序
	整理程序框图	自动将程序框图上的对象重新连线及重新安排位置（仅程序框图窗口有）

3.2.4　LabVIEW 2019 的数据类型

LabVIEW 作为一种完整的编程语言，与其他文本编程语言一样，它的数据操作是最基本的操作，同时还拥有特殊的一些数据类型。LabVIEW 主要的数据类型包括基本数据类型，如数值型、字符型和布尔型，还包括结构类型（包括一个以上的元素），如数组和簇。

图形化编程语言与 C 语言有着显著不同，C 语言的数据是放置在已声明的“变量”中，LabVIEW 的数据（常数除外）不是放置在变量中，而是放置在前面板的对象——控件（包括输入控件和显示控件）中，同时控件本身还确定了数据的数据类型。

LabVIEW 以浮点数、定点数、整数、无符号整数及复数表示数值数据。数据类型的差别在于用于存储数据的位数和表示数字的范围不同，双精度和单精度及复数数据在 LabVIEW

中以橙色表示，蓝色则代表所有整数的数值数据。LabVIEW 中数值数据类型见表 3.5。

表 3.5　数值数据类型表

数据类型	标记	简要说明
定点	FXP	定点数，最大位数为 64 位
单精度	SGL	单精度浮点数，存储位数为 32 位
双精度	DBL	双精度浮点数，存储位数为 64 位
扩展精度	EXT	扩展精度浮点数，存储位数为 128 位
单精度复数	CSG	复数单精度浮点数，实部和虚部存储位数均为 32 位
双精度复数	CDB	复数双精度浮点数，实部和虚部存储位数均为 64 位
扩展精度复数	CXT	复数扩展精度浮点数，实部和虚部存储位数均为 128 位
单字节整型	I8	有符号整数，存储位数为 8 位，取值范围-128～127
双字节整型	I16	有符号整数，存储位数为 16 位，取值范围-32 768～32 767
长整型	I32	有符号整数，存储位数为 32 位，取值范围-2 147 483 648～2 147 483 647
64 位整型	I64	有符号整数，存储位数为 64 位，取值范围-18 446 744 073 709 551 616～18 446 744 073 709 551 615
无符号单字节整型	U8	无符号整数，取值范围 0～255
无符号双字节整型	U16	无符号整数，取值范围 0～65 535
无符号长整型	U32	无符号整数，取值范围 0～4 294 967 295
无符号 64 位整型	U64	无符号整数，取值范围 0～1 844 674 407 309 551 615

布尔型的值为 1 或 0，即真（True）或假（False）。通常情况下，布尔型即为逻辑型，LabVIEW 用 8 位二进制数保存布尔数据，如 8 位的值均为 0，布尔值为 False，所有非 0 值都表示 True。在 LabVIEW 中，用绿色代表布尔型数据。

字符串是可显示或不可显示的 ASCII 字符序列。字符串可以提供与平台无关的信息和数据格式，是 LabVIEW 中一种基本的数据类型，LabVIEW 中的字符串以粉色表示。

路径是一种特殊的字符串，专门用于对文件路径的处理。路径控件用于输入或返回文件或目录的地址，路径控件与字符串控件的工作原理类似，但 LabVIEW 会根据用户使用操作平台的标准句法将路径按一定格式处理。

不同的数据类型在 LabVIEW 中用不同的线型和颜色表示，见表 3.6。

表 3.6　不同数据类型对应的线型和颜色

数据类型	标　量	一维数组	二维数组	颜　色
整型数				蓝色
浮点数				橙色
逻辑量				绿色
字符串				粉色
文件路径				青色

3.3　LabVIEW 2019 的初步操作

一个完整的 VI 由前面板、程序框图和图标/连线板组成。本节将介绍 LabVIEW 2019 程序设计的基本方法。

3.3.1　创建和编辑 VI

在 LabVIEW 2019 的启动界面单击【文件】→【新建 VI】，就会出现如图 3.10 所示的前面板和程序框图窗口。

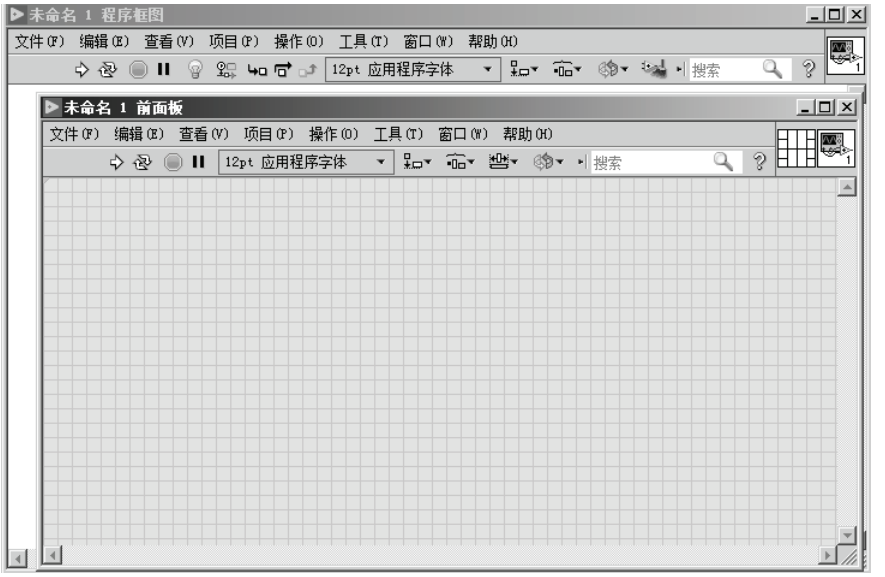


图 3.10　新建 VI 窗口

1．前面板设计

用工具选板中相应的工具去取用控件选板中的程序所需的相关控件，排列到前面板窗口中的合适位置，打开控件的属性设计窗口进行参数设置，并加上各种文字说明或标签，也可以加入一些装饰用的控件。一般情况下，前面板窗口中创建的控件会自动在程序框图窗口创建相应的接线端。一个简单的两数相加与两数相减 VI 的前面板如图 3.11 所示。

2．程序框图设计

每个前面板都有一个程序框图与之对应。程序框图用图形化编程语言编写，可以把它理解成传统编程语言程序中的源代码。

用工具选板中相应的工具去取用函数选板中的程序所需的相关控件，排列到程序框图窗口中的合适位置，这些控件即是程序框图中的节点或结构。图 3.11 所示的前面板所对应的程序框图如图 3.12 所示。

3．数据流编程

数据流编程就是连线操作。程序框图中对象的数据传输通过连线实现。在图 3.12 中，输入控件和显示控件的接线端口通过连线实现加、减运算。连线也是程序设计中较为复杂的问

题。程序框图上的每个对象都带有自己的接线端口，连线将构成对象之间的数据通道。由于这不是几何意义上的连线，因此并非任意两个端口间都可连线。连线类似于高级文本语言程序中的变量数据单向流动，从源端口向一个或多个目的端口流动。不同数据类型的连线有不同的颜色、粗细和样式。

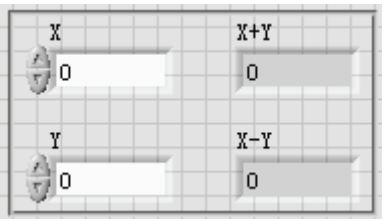


图 3.11　两数相加与两数相减 VI 的前面板

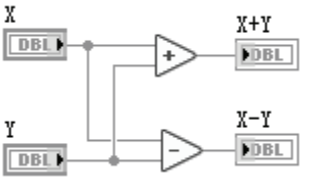


图 3.12　两数相加与两数相减 VI 的程序框图

当需要连接两个端口时，在第一个端口上单击连线工具（从工具选板中调用），然后移动到另一个端口，再单击第二个端口。端口的先后次序不影响数据流动的方向。

当把连线工具放在端口上时，该端口区域将会闪烁，表示连线将会接通该端口。当把连线工具从一个端口接到另一个端口时，不需要按住鼠标。当需要连线转弯时，单击即可以正交垂直方向弯曲连线，按空格键可以改变转角的方向。

接线头是为了帮助正确连接端口的连线。当把连线工具放到端口上，接线头就会弹出。接线头还有一个黄色小标识框，以显示该端口的名字。

线形为波折形的连线表示坏线。出现坏线的原因有很多，例如连接了两个控制对象，源端口和终端口的数据类型不匹配等。可以通过使用定位工具单击坏线，再按 Delete 键来删除坏线。

4．编辑 VI 图标

为了唯一标识创建的 VI，可为该 VI 创建一个图标。双击前面板窗口或程序框图窗口右上角的图标，或用鼠标右键单击图标，在弹出的快捷菜单中单击【编辑图标】，将弹出图标编辑器对话框，如图 3.13 所示。该对话框包括模板、图标文本、符号、图层几部分，用户可以根据自己的设计需要进行选择。

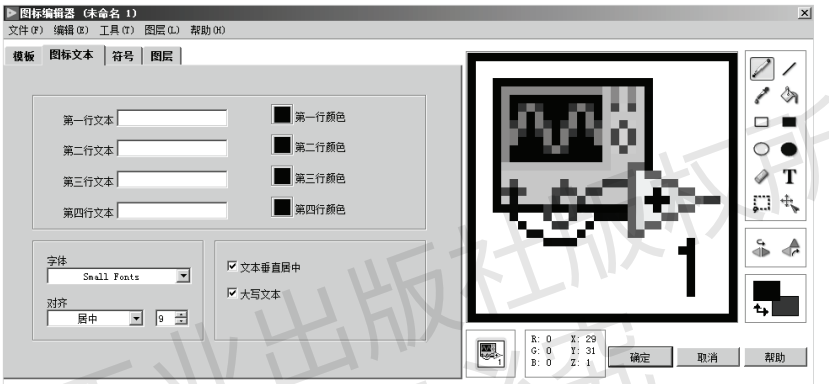


图 3.13　图标编辑器对话框

在图标编辑器中可以创建用户自己的图标。图标编辑器的用法与 Windows 操作系统中的画图工具软件类似，使用图标编辑器右边的图标文本、符号和编辑工具可编辑 VI 图标。

5．保存 VI

在前面板窗口或程序框图窗口中选择菜单中的【文件】→【保存】命令，然后在弹出的保存文件对话框中选择适当的路径和文件名保存 VI。

3.3.2 运行和调试 VI

运行和调试程序是在用任何一种编程语言编程的过程中最重要的一步。通过这一步，编程者可以查找出程序中存在的错误，根据这些错误和运行结果修改、优化程序，使编写的程序达到预期的效果。LabVIEW 提供了有效的编程调试环境，可帮助用户完成程序的调试。

1．运行 VI

在 LabVIEW 中可以通过两种方式来运行 VI，即运行和连续运行。

（1）运行 VI

在前面板窗口或程序框图窗口的工具栏中单击【运行】按钮，可以运行 VI。使用这种方式运行 VI，VI 只运行一次，当 VI 正在运行时，【运行】按钮会变成状态。

（2）连续运行 VI

在工具栏中单击【连续运行】按钮，可以连续运行 VI。连续运行的意思是指一次 VI 运行结束后，继续重新运行 VI。当 VI 正在连续运行时，【连续运行】按钮会变成状态。单击按钮，可停止 VI 的连续运行。

当 VI 处于运行状态时，单击工具栏中的按钮，可强行终止 VI 的运行；单击按钮，可暂停 VI 的运行。

2．查找 VI 中的错误

LabVIEW 在编辑过程中有一个自动编译的效果，即正在创建或编辑的 VI 没有错误时，运行按钮为正常状态，若运行按钮呈断开的形状，则说明程序中存在错误。单击断开的按钮或选择菜单中的【查看】→【错误列表】，可查找 VI 断开的原因。错误列表列出了所有的错误。单击【帮助】按钮，可显示 LabVIEW 帮助中对错误的详细描述和纠正错误的相关主题。

VI 断开的常见原因如下：

① 数据类型不匹配或存在未连接的接线端。

② 必须连接的程序框图接线端没有连线。

③ 子 VI 处于断开状态或在程序框图上放置子 VI 图标后编辑了该子 VI 的连线板。

3．高亮显示执行过程

单击程序框图窗口工具栏中的【高亮显示执行过程】按钮，可查看程序框图的动态执行过程。

高亮显示执行过程通过沿连线移动的圆点显示数据在程序框图上从一个节点移动到另一个节点的过程。使用高亮显示执行过程的同时，结合单步执行，可查看 VI 中的数据从一个节点移动到另一个节点的全过程。

按照下列步骤，使用高亮显示执行过程：

① 打开任意 VI 的程序框图。

② 单击程序框图窗口工具栏上的【高亮显示执行过程】按钮，启用执行过程高亮显示。

③ 运行 VI，并查看 VI 运行时的程序框图。

高亮显示执行过程时，注意连线上的圆点。圆点的移动显示了数据从一个节点传递至另一个节点的过程，每个接线端的值也将同时显示。

④ 停止 VI。

⑤ 再次单击【高亮显示执行过程】按钮，即可立即停止高亮显示执行过程。

4．单步执行

单步执行 VI 时可查看运行时程序框图上的每个执行步骤。所有单步执行按钮仅在单步执行模式下影响 VI 或子 VI 的运行。单步执行一个 VI 时，该 VI 的各个子 VI 既可单步执行，也可正常运行。

按照下列步骤单步执行 VI：

① 单击程序框图窗口工具栏上的【开始单步执行】按钮。

② 根据需要选择并单击以下按钮：

【单步步入】，打开一个节点并暂停。再次单击【单步步入】按钮时，将执行第一个操作，然后在子 VI 或结构的下一个操作前暂停。

【单步步过】，执行一个节点并在达到下一个节点时暂停。

【单步步出】，结束当前节点的操作并暂停。VI 结束操作时，单步步出按钮将变为灰色。

在单步执行 VI 时，若某些节点发生闪烁，表示这些节点已准备就绪，可以执行。

将光标移动到【单步步过】、【单步步入】或【单步步出】按钮时，可看到一个提示框，该提示框描述了单击该按钮后的下一步执行情况。若单步执行 VI 并高亮显示执行过程，执行符号将出现在当前执行的子 VI 的图标上。

③ 当整个程序框图被一个闪烁的边框包围时，单击【单步步出】按钮可结束单步执行。不处于单步执行状态时，相关的单步执行按钮将变为灰色。

5．设置探针

工具选板中的探针工具用于检查 VI 运行时连线上的值。若程序框图较复杂且包含一系列每步执行都可能返回错误值的操作，则可使用探针工具。利用探针并结合高亮显示执行过程、单步执行和断点，可确认数据是否有误并找出错误数据。当 VI 运行时，若有数据流过探针查看的数据连线，探针对话框会自动显示这些流过的数据。当执行过程由于单步执行或断点而在某一节点处暂停，可用探针探测刚才执行的连线，查看流经该连线的数值。

6．设置断点

使用断点工具在 VI、节点或连线上放置一个断点，程序运行到该处时暂停执行。在连线上设置断点后，数据流经该连线且暂停按钮为红色时程序将暂停执行。在程序框图上放置一个断点，使程序框图在所有节点执行后暂停执行。此时程序框图边框变为红色，断点不断闪烁以提示断点所在位置。

VI 暂停于某个断点时，程序框图将出现在最前方，同时一个选取框将高亮显示含有断点的节点或连线。光标移动到断点上时，断点工具光标的黑色区域变为白色。

（1）设置断点

① 用断点工具单击需暂停执行的 VI、节点或连线。也可用鼠标右键单击 VI、节点

或连线，从快捷菜单中选择【设置断点】。

② 运行 VI。程序执行到一个断点时，VI 将暂停执行，同时暂停按钮显示为红色。VI 的背景和边框开始闪烁。此时，可进行下列操作：

- 用【单步执行】按钮单步执行程序；
- 查看连线上在 VI 运行前事先放置的探针的实时值；
- 若启用了保存连线值选项，则可在 VI 运行结束后，查看连线上探针的实时值；
- 改变前面板控件的值；
- 检查调用列表下拉菜单，查看停止在断点处调用该 VI 的 VI 列表；
- 单击【暂停】按钮，可继续运行到下一个断点处或直到 VI 运行结束。

（2）启用和禁用断点

若要禁用断点，使 VI 在断点处暂停执行后继续，可用鼠标右键单击断点所在的对象，从快捷菜单中选择【断点】→【禁用断点】。若要启用之前禁用的断点，用鼠标右键单击程序框图对象，从快捷菜单中选择【断点】→【禁用断点】。可一次禁用或启用一个断点，也可使用【断点管理器】窗口一次禁用或启用全部断点。

（3）删除断点

用断点工具单击一个现有断点并将其删除。也可用定位工具鼠标右键单击断点，从快捷菜单中选择【断点】→【清除断点】将其删除。选择【编辑】→【从层次结构中删除断点】，可删除 VI 层次结构中所有的断点。可使用【断点管理器】窗口，一次移除 VI 层次结构中的所有断点。

3.3.3 创建和调用子 VI

可将新创建的 VI 用于另一个 VI。一个 VI 被其他 VI 在程序框图中调用，则称该 VI 为子 VI。子 VI 相当于文本编程语言中的子程序，可重复调用。子 VI 的控件和函数从调用该 VI 的程序框图中接收数据，并将数据返回至该程序框图。

1．创建子 VI

构造一个子 VI 的主要工作就是需先为子 VI 创建图标和连线板。

每个 VI 在前面板窗口和程序框图窗口的右上角都有一个图标  。图标是 VI 的图形化表示，可包含文字、图形或图文组合。若将 VI 当作子 VI 调用，程序框图上将显示该子 VI 的图标。

默认图标中有一个数字，表明 LabVIEW 启动后打开新 VI 的个数。用鼠标右键单击前面板窗口或程序框图窗口右上角的图标并从快捷菜单中选择【编辑图标】，或双击前面板窗口右上角的图标，会弹出一个图标编辑器，如图 3.13 所示。在图标编辑器中可创建用户自己的图标。

在完成图标创建后，将其作为子 VI 调用的主要工作就是创建连线板。

用鼠标右键单击前面板窗口右上角的连线板  ，并从快捷菜单中选择【模式】，会出现一个图形化下拉菜单。菜单中列出了 36 种不同的连线板，如图 3.14 所示，可为 VI 选择不同的连线板模式。

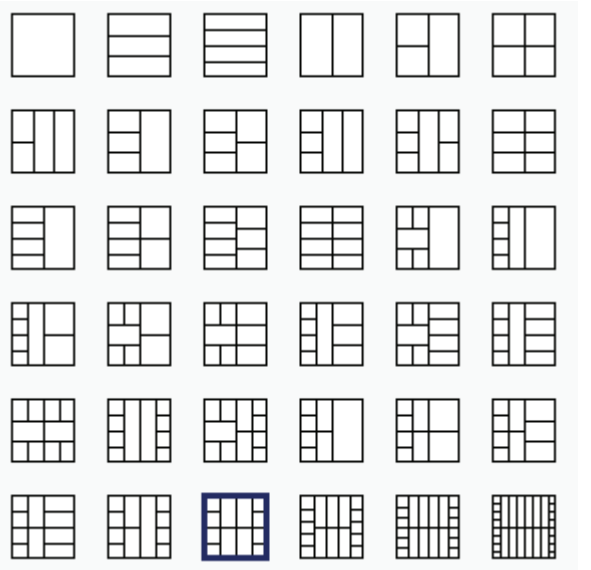


图 3.14　LabVIEW 的连线板

连线板集合了 VI 各个接线端，与 VI 中的控件相互对应，类似文本编程语言中函数调用的参数列表。连线板标明了可与该 VI 连接的输入端和输出端，以便将该 VI 作为子 VI 调用。连线板在其输入端接收数据，然后通过前面板控件将数据传输至程序框图的代码中，从前面板的显示控件中接收运算结果并传递至其输出端。

连线板上的每个单元格代表一个接线端，使用各个单元格分配输入和输出。默认的连线板模式为 4×2×2×4。使用默认模式可保留多余的接线端，当需要为 VI 添加新的输入或输出端时再进行连接。

连线板中最多可设置 28 个接线端。如果前面板上的控件不止 28 个，可将其中的一些对象组合为一个簇，然后将该簇分配至连线板上的一个接线端。

完成了连线板的创建之后，接下来的工作就是定义前面板中的输入控件和显示控件与连线板中各输入端、输出端的关联关系。方法是：单击连线板中的一个矩形（代表一个接线端），光标自动变成连线工具，同时接线端变成黑色，再单击需要连接的前面板对象，此时该前面板对象被虚线框包围，选中的端口的颜色变为与该前面板对象的数据类型一致的颜色，表明该前面板对象和接线端建立起一对一的关系。如果接线端是白色的，则表示没有连接成功。

例如，将前面创建的虚拟仪器例子——两数相加与两数相减 VI 创建为子 VI，将其图标使用图标编辑器中的文本工具编辑为“加减程序”；因该 VI 有两个输入参数（X，Y）与两个输出参数（X+Y，X−Y），其连线板可选择 2×2 模式。为该 VI 创建的图标和连线板如图 3.15 所示。

图 3.15　创建两数相加与两数相减 VI 为子 VI 的图标与连线板

最后，在前面板的菜单中选择【文件】→【保存】，将保存该 VI，即完成了子 VI 的创建。

2．调用子 VI

在函数选板中选择【选择 VI…】，弹出一个名为【选择需打开的 VI】对话框，在对话框中找到需要调用的子 VI，选中后单击【确定】按钮，此时，在光标上会出现这个子 VI 的图标，将其移动到程序框图窗口中的适当位置并单击，将图标加入主 VI 的程序框图中。然后，用连线工具将子 VI 的各个连接端与主 VI 中的其他节点按照一定的逻辑关系连接起来，至此，就完成了子 VI 的调用。

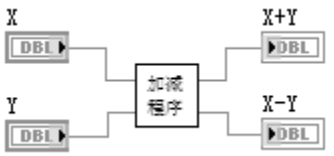


图 3.16　调用两数相加与两数相减子 VI 的 VI 程序框图

例如，调用创建的两数相加与两数相减子 VI 的 VI 程序框图如图 3.16 所示。

3.3.4　VI 创建举例——虚拟温度计

下面以虚拟温度计为例，详细介绍用 LabVIEW 2019 开发虚拟仪器的方法。

实际的温度测量有多种方式，如使用热敏电阻、热电偶、RTD 等。本例采用 AD590 集成温度传感器。AD590 是电流型集成温度传感器的代表产品，其温度测量范围为−55～150℃，线性电流输出为 1μA/K。AD590 以热力学温标零点作为零输出点，在 25℃时的输出电流为 298.2μA。

假设温度测量范围为 0～100℃，按图 3.17 所选定的电路参数，该电路的输出电压灵敏度为 10mV/℃。因为 AD590 直接测量的是热力学温度（单位为 K），因此，为了以摄氏温度读出，其输出必须以 273.15μA 偏置。令 AD590 的输出电流流过 1kΩ电阻，这样将 1μA/K 的电流灵敏度转换为 1mV/K 的电压灵敏度，再将转换后的输出电压连接到 AD524 仪表放大器的同相输入端。基准电压芯片 AD580 输出的 2.5V 基准电压用电阻分压到 273.15mV，接仪表放大器的反相输入端，设置 AD524 的放大倍数为 10，这样，经 AD524 对两输入端的差值放大后，就可将 0～100℃的温度输入转换为 0～1V 的电压输出，即该温度测量电路的输出电压灵敏度为 10mV/℃。

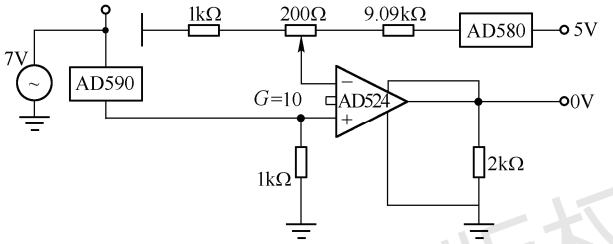


图 3.17　温度测量电路

为了设计方便，本例用一个随机数据代替温度传感器测量电路模拟产生电压输出。设温度测量范围为 0～100℃，VI 设计步骤如下。

1．前面板设计

启动 LabVIEW 2019 后，在启动界面上，选择新建 VI，创建一个新 VI，然后按下面的步骤进行设计。

① 在控件选板的【新式】→【数值】子选板中选择“仪表”控件，放置到前面板窗口的合适位置，将标签“仪表”改为“电压（mV）”。然后，用鼠标右键单击该控件，在弹出的快捷菜单中选择【属性】，在弹出的属性窗口中选择【标尺】，在标尺窗口中设置最小值为 0，最大值为 1000。

② 在控件选板的【新式】→【数值】子选板中选择“温度计”控件，放置到前面板窗口的合适位置。

③ 在控件选板的【新式】→【布尔】子选板中选择“开关按钮”控件，放置到前面板窗口的合适位置，将标签“布尔”改为“开关”，用鼠标右键单击该控件，在弹出的快捷菜单中单击【显示项】→【标签】，隐藏该控件的标签显示。然后，用鼠标右键单击该控件，在弹出的快捷菜单中单击【属性】，在弹出的属性对话框中单击【外观】→【显示布尔文本】，将“开时文本”框中的内容改为“开”，将“关时文本”框中的内容改为“关”，再单击【确定】按钮。

④ 选用工具选板中的“编辑文本”工具，在前面板窗口的合适位置单击，输入文本“虚拟温度计”，再选用工具选板中的“定位/调整大小/选择”工具，单击前面板窗口上的工具栏【文本】，可改变输入文本大小、样式等。

⑤ 在控件选板的【新式】→【修饰】子选板中选择“下凹框”控件，放置到前面板窗口的合适位置，并设置合适的大小。

完成以上 5 个步骤后的虚拟温度计 VI 前面板如图 3.18 所示。

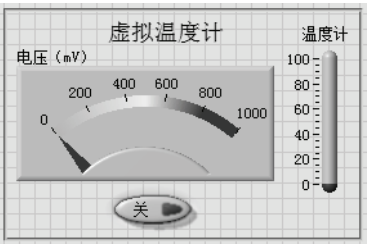


图 3.18　虚拟温度计 VI 前面板

2．程序框图设计

程序框图的设计步骤如下。

① 打开程序框图窗口，调整与前面板相对应的控件图标位置，以便后续摆放函数与连线。

② 在函数选板的【编程】→【数值】子选板中选择“随机数(0-1)”函数，放置到程序框图窗口的合适位置。

③ 在函数选板的【编程】→【数值】子选板中选择“乘”函数，放置到程序框图窗口的合适位置（放置 2 个乘法器函数）。

④ 在函数选板的【编程】→【数值】子选板中选择“数值常量”函数，放置到程序框图窗口的合适位置（放置 3 个数值常量，常数数值分别设置为 0、100、10）。

⑤ 在函数选板的【编程】→【比较】子选板中选择“选择”函数，放置到程序框图窗口的合适位置。

3．数据流编程

选用工具选板中的“连线”工具，根据温度计的设计原理连接各个节点和函数，即可完成数据流编程。

设计好的虚拟温度计 VI 程序框图如图 3.19 所示。

4．编辑 VI 图标

双击前面板窗口右上角的图标，在弹出的图标编辑器对话框右边的工具栏中，单击【选择】，将原来默认图标全部选中，按 Delete 键，删除选中的默认图标。然后，单击【图标文本】选项卡，在“第一行文本”栏中输入“温度”，在“第二行文本”栏中输入“测量”，单

击【确定】按钮，即可将该 VI 的图标编辑为“温度测量”图形化标识。

5．运行和保存 VI

在前面板窗口上，选用工具选板中的“操作值”工具，单击前面板上放置的开关按钮，使其显示为“开”状态，然后单击前面板窗口工具栏上的【运行】按钮，就可运行设计好的虚拟温度计 VI，运行结果如图 3.20 所示。

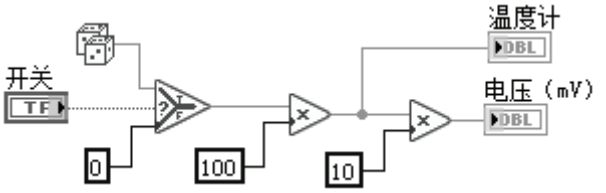


图 3.19　虚拟温度计 VI 程序框图

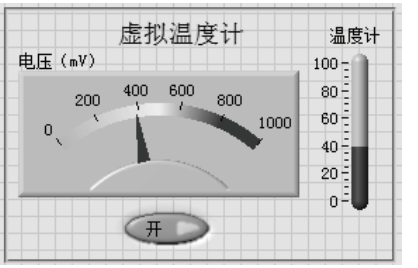


图 3.20　虚拟温度计 VI 运行结果

运行 VI 检验程序设计正确后，可在前面板或程序框图窗口选择菜单上的【文件】→【保存】，在弹出的文件保存对话框中，选择合适的路径，输入文件名，单击【确定】按钮即可保存 VI 文件。

3.4　在 LabVIEW 中管理项目

一个大型的 LabVIEW 应用程序通常包含一个 VI 主程序、多个 VI 子程序、库和相关的其他数据文件，这些文件和数据需要集中统一管理。在 LabVIEW 中，可以通过 LabVIEW 的项目进行管理，也就是可以使用 LabVIEW 中的项目来管理 VI 程序及其他的 LabVIEW 文件和文档，甚至其他可以想到的计算机文件也可以通过 LabVIEW 的项目进行管理。

3.4.1　新建项目

在 LabVIEW 的启动界面中，可以选择创建项目和打开项目；在前面板和程序框图窗口菜单的【项目】中，也有创建、打开、保存、关闭项目等选项。

在 LabVIEW 的启动界面中单击【项目】→【创建项目】，出现如图 3.21 所示的创建项目对话框。

在创建项目对话框可以新建项目、新建 VI，还包含一组模板和范例项目，可帮助用户遵循可靠的设计模式和编程规范。若选择“项目”（新建一个空白项目），单击【完成】按钮，就会弹出项目浏览器窗口，如图 3.22 所示。

在项目浏览器窗口中有“项”和“文件”两个选项卡。“项”选项卡的功能是管理项目，“文件”选项卡用于显示在磁盘上有相应文件的项目项，在该选项卡上可对文件名和目录进行管理。

默认情况下，项目浏览器窗口包括以下各项：

● 项目根目录：包含项目浏览器窗口中所有其他项。项目根目录的标签包括该项目的文件名。

图 3.21　创建项目对话框

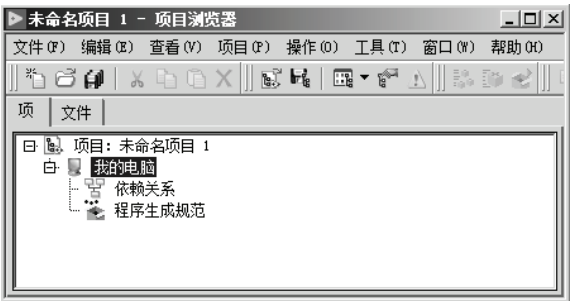


图 3.22　项目浏览器窗口

- 我的电脑：表示可作为项目终端使用的本地计算机。
- 依赖关系：包括某个终端下必需的 VI，如 VI、共享库、LabVIEW 项目库。
- 程序生成规范：包括对源代码发布编译配置以及 LabVIEW 工具包和模块所支持的其他编译形式的配置。如果安装了 LabVIEW 专业版开发系统或应用程序生成器，可使用程序生成规范进行生成独立应用程序、安装程序、.NET 互操作程序集、打包库、共享库、发布源代码、Web 服务、Zip 文件等操作。

在项目浏览器窗口中，单击菜单上的【文件】→【另存为】，并选择项目保存路径和名称后，单击【确定】按钮，即可创建一个新项目。LabVIEW 项目文件的扩展名为.lvproj。例如，将创建的项目命名为 Test1 后的项目浏览器窗口如图 3.23 所示。

3.4.2　管理项目

用户可对项目中的项进行管理，包括添加项至项目、在 LabVIEW 项目中查找项目项、删除项目中的项、删除项目中的依赖关系、项目中的项进行排序、开发和发布应用程序等，根据需要对项进行操作。

图 3.23　新建 Test1 项目的项目浏览器窗口

1．添加项至项目

使用项目浏览器窗口可向 LabVIEW 项目添加 LabVIEW 文件，如 VI 和库，以及非 LabVIEW 特有的文件，如文本文件和电子表格。将文件添加到一个项目中时，LabVIEW 在项目浏览器窗口中创建代表内存中文件的项。

例如，若要将 3.3 节创建的“虚拟温度计”VI 添加至建立的 Test1 项目中，方法是：用鼠标右键单击【我的电脑】，在弹出的快捷菜单中选择【新建】或【添加】，则可进行新建或添加 VI 的操作，如图 3.24 所示。

图 3.24　添加 VI 后的项目浏览器窗口

将 VI 添加至项目时，LabVIEW 自动将整个层次结构添加到项目浏览器的依赖关系下。

2．删除项目中的项

使用项目浏览器窗口可删除 LabVIEW 项目中的项。通过下列方式可删除项目中的项：

- 右键单击要删除的项，在弹出的快捷菜单中选择【从项目中删除】；
- 选择需删除的项，单击【项目】菜单上的【从项目中删除】；
- 选择需删除的项并按 Delete 键。

需要注意的是，从项目中删除项并不会在磁盘中删除相应文件。

3．对项目中的项进行排序

用户可使用名称、类型、路径、自定义、与父项相同等排序选项对项目项进行排序。排序选项不会影响项目在磁盘上的排列，减少了项目中改动时合并和比较操作所需的时间。

4．开发和发布应用程序

用户可使用程序生成规范来创建和发布自定义 LabVIEW 应用程序。注意，生成和发布应用程序必须安装应用程序生成器（LabVIEW 专业版开发系统中含有应用程序生成器）。

LabVIEW 项目可转换为可发布的应用程序用于其他计算机，创建可发布应用程序的类型有：

● 独立应用程序，为其他用户提供 VI 的可执行版本，用户无须安装 LabVIEW 开发系统也可运行 VI。运行独立应用程序，需 LabVIEW 运行引擎。Windows 应用程序以.exe 为扩展名。

● 安装程序，安装程序用于发布通过应用程序生成器创建的独立应用程序、共享库和源代码发布等。包含 LabVIEW 运行引擎的安装程序允许用户在未安装 LabVIEW 的情况下运行应用程序或使用共享库。

● .NET 互操作程序集，.NET 互操作程序集将一组 VI 打包，用于 Microsoft.NET Framework。若要通过应用程序生成器创建.NET 互操作程序集，必须安装.NET Framework 4.0。

● 打包库，使用打包库将多个 LabVIEW 文件打包至一个文件。部署打包库中的 VI 时，只需部署打包库一个文件即可。打包库的顶层文件是一个项目库。打包库包含为特定操作系统编译的一个或多个 VI 层次结构，打包库的扩展名为.lvlibp。

● 共享库（DLL），为非 LabVIEW 编程语言提供了访问 LabVIEW 代码的方式。若需与其他开发人员共享所创建 VI 的功能，可使用共享库。其他开发人员可使用共享库，但不能编辑或查看该库的程序框图，除非编写者在共享库上启用调试。Windows 共享库以.dll 为扩展名。

● 源代码发布，源代码发布是将一系列源文件打包。用户可通过源代码发布将代码发送给其他开发人员在 LabVIEW 中使用。在 VI 设置中可实现添加密码、删除程序框图或应用其他配置等操作。为一个源代码发布中的 VI 可选择不同的目标目录，而且 VI 和子 VI 的连接不会因此中断。

● Zip 文件，压缩文件用于以单个可移植文件的形式发布多个文件或整套 LabVIEW 项目。一个 Zip 文件包括可发送给用户使用的已经压缩了的多个文件。Zip 文件可用于将已选定的源代码文件发布给其他 LabVIEW 用户使用。可使用 Zip VI 通过编程创建 Zip 文件。

生成和发布应用程序的操作方法是：用鼠标右键单击项目浏览器窗口中的【程序生成规范】，在弹出的快捷菜单中选择【新建】下面的子菜单来完成相应的创建操作，如图 3.25 所示。

图 3.25　生成和发布应用程序界面

对于其他项目的管理操作，可以查阅 LabVIEW 的在线帮助，以获得详细解答。

3.5 获取 LabVIEW 帮助

在 LabVIEW 的程序开发过程中，如果遇到疑问或者初次使用的函数，可以通过 LabVIEW 的帮助文档获取详细的信息。了解 LabVIEW 的帮助，用户可以更好地学习和掌握 LabVIEW。

LabVIEW 的帮助信息主要包括显示即时帮助、LabVIEW 帮助、LabVIEW 编程范例及 LabVIEW 网络资源等。这些帮助信息可以通过前面板或程序框图窗口中的“帮助”菜单来获得。

3.5.1 显示即时帮助

显示即时帮助是 LabVIEW 提供的实时快捷帮助窗口，即时帮助信息对于 LabVIEW 的初学者来说非常有用。

单击 LabVIEW 菜单的【帮助】→【显示即时帮助】，即可弹出即时帮助窗口。此时如果用户需要了解当前程序框图中的 VI、节点或控件的帮助信息，只需将光标移动到相关 VI、节点或控件上面，即时帮助窗口将显示其基本的功能说明信息，从这些帮助信息中用户可以了解到 VI、节点或控件的基本功能。

例如，对于一个“+”函数，其即时帮助窗口如图 3.26 所示。

图 3.26 “+”函数的即时帮助窗口

3.5.2 LabVIEW 帮助

LabVIEW 帮助提供了 VI、函数、模板、菜单和工具的全部参考信息。单击 LabVIEW 菜单的【帮助】→【LabVIEW 帮助】，即可弹出 LabVIEW 帮助窗口，如图 3.27 所示。

通过窗口左侧的“目录”“索引”“搜索”选项卡可练习整个帮助系统，用户可以方便地查找到自己感兴趣的帮助信息。

在 LabVIEW 的编程过程中，如果用户想获取某个函数的帮助信息，可以在目标函数节点上单击鼠标右键，在弹出的快捷菜单中选择【帮助】，也可以打开 LabVIEW 的帮助窗口，在这个窗口中将显示该函数相关的帮助信息。

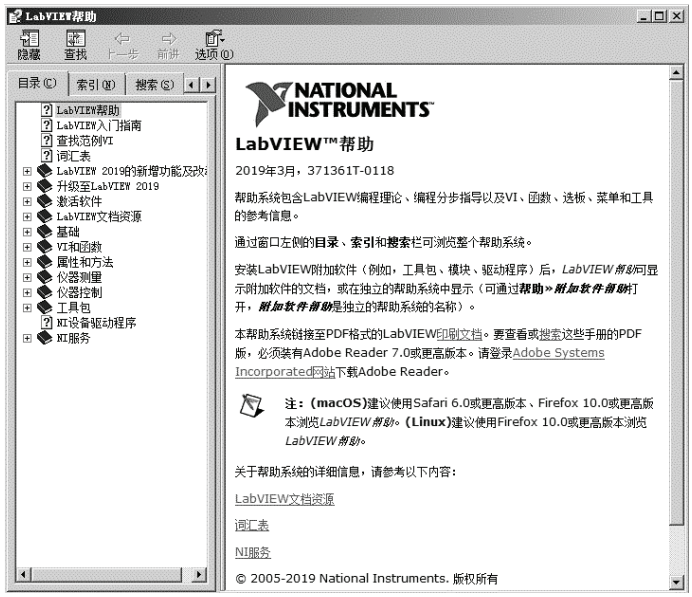


图 3.27　LabVIEW 帮助窗口

3.5.3　LabVIEW 编程范例

LabVIEW 编程范例包含了 LabVIEW 各个功能模块的应用实例，借鉴 LabVIEW 提供的典型范例是快速、深入地学习 LabVIEW 的一个好方法。

单击 LabVIEW 菜单的【帮助】→【查找范例】，即可弹出 NI 范例查找器窗口，如图 3.28 所示。

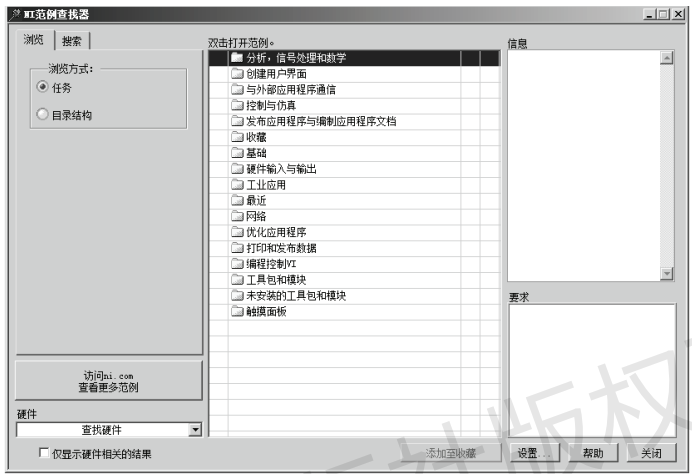


图 3.28　NI 范例查找器窗口

在 NI 范例查找器窗口的左侧，可选择任务或目录结构浏览方式查找 LabVIEW 范例，也可以使用搜索功能查找感兴趣的 LabVIEW 范例。

在 LabVIEW 的学习过程中，用户可以充分利用 LabVIEW 自带例程，帮助掌握 LabVIEW

的编程方法。很多情况下，只需对这些例程稍做修改，就可以直接应用到自己的 VI 中，从而提高程序的开发效率。

3.5.4 LabVIEW 网络资源

单击 LabVIEW 菜单的【帮助】→【网络资源】，即可通过网页浏览器连接到 NI 公司的官网（www.ni.com）。该官网提供了大量的 LabVIEW 的最新技术与应用方案，还能共享代码、获取技术支持等。

本 章 小 结

本章介绍了 LabVIEW 的基本特性，说明了安装与启动 LabVIEW 2019 的方法，通过 LabVIEW 2019 的操作选板、菜单和工具栏，介绍了 LabVIEW 2019 的编程环境。同时详细讲解了创建 VI 的一般步骤，并通过虚拟温度计的创建介绍了 VI 的设计方法。

详细了解 LabVIEW 的编程环境，掌握虚拟仪器的创建步骤和调试方法，是学习和编写 LabVIEW 程序的重要一步，希望读者能够认真体会。

思考题和习题 3

3.1 什么是 LabVIEW？LabVIEW 编写的程序由哪几部分组成？

3.2 LabVIEW 有哪 3 个选板？简述其各自的功能。

3.3 LabVIEW 的前面板与程序框图如何切换？

3.4 简述用 LabVIEW 编写程序的一般步骤。

3.5 程序框图主要由哪几类对象构成？它们分别起什么作用？

3.6 比较前面板工具栏和程序框图工具栏的相同及不同之处。

3.7 简述 LabVIEW 程序调试的基本方法。

3.8 如何利用错误列表快速定位程序框图中的错误？

3.9 什么是子 VI？

3.10 如何更改 VI 的图标？

3.11 在前面板创建 3 个数值控件，分别按上边缘对齐、下边缘对齐、左边缘对齐和右边缘对齐排列这 3 个数值控件。

3.12 设计 VI，把两个输入数值相加，再把和乘以 20。

3.13 设计 VI，输入一个数，判断这个数是否在 10～100 之间。

3.14 设计 VI，比较两个数，如果其中一个数大于另一个数，则点亮 LED 指示灯。

3.15 设计 VI，产生一个 0.0 到 10.0 的随机数与 10.0 相乘，然后通过一个子 VI 将积与 100 相加后开方。

3.16 设计 VI，求 3 个输入数的平均值，并将平均值与一个 0.0 到 1.0 之间的随机数相乘。要求将其中求平均值的部分创建为子 VI 来实现 VI 设计。

3.17 如何获取 LabVIEW 帮助？