

实验二十九

虚拟仪器在物理实验中的应用 ——伏安法测电阻与非线性元件

【目的要求】

- (1) 了解虚拟仪器的概念；
- (2) 了解图形化编程语言 LabVIEW，学习基本的 LabVIEW 编程；
- (3) 完成伏安法测电阻的虚拟仪器设计。

【仪器用具】

计算机（含操作系统）、LabVIEW 2020，数据采集卡， 100Ω 标准电阻，待测电阻 $1k\Omega$ 和 50Ω 各一个，稳压二极管一个，导线若干。

【实验原理】

1. 虚拟仪器的概念

通常的电子测量仪器包括信号输入、分析处理、显示和控制几个部分。例如，在传统的示波器中，利用探头来输入信号，在显示屏上显示信号，有些示波器还具有对输入信号进行处理和分析的高级功能。对于整个测量过程我们可以通过示波器前面板上提供的控制按钮或触摸键进行控制。

迄今为止，我们一般接触到的都是传统的独立仪器，比如，模拟示波器，数字万用表，函数发生器等。传统独立仪器有些显而易见的缺点：数据显示形式比较单一，数据处理功能比较简单，不容易根据需要自行改装，不能共享数据，自动化程度低，等等。随着计算机和网络技术的发展，计算机在处理数据、通信方面的长处日益显露出来，人们开始想把计算机应用到测量上来，一种思路是把计算机装入仪器，这是智能化仪器的思想；一种思路是把仪器装入计算机，这就是虚拟仪器的思想。把仪器装入计算机意味着测量是以计算机为中心的，测量对象的信息必须变成计算机能处理的信息，对测量对象的控制也是由计算机完成，而用户则通过计算机来控制测量和了解测量结果。因此，虚拟仪器实际上就是一种基于计算机的自动化测试仪器系统。它利用通用计算机的强大计算处理功能，通过传感器和接口卡实现信号输入，用键盘、鼠标、显示器等计算机外设实现控制和显示功能。使用者可以根据需要自行设计仪器，体现了“软件就是仪器”的新概念。虚拟仪器设计过程主要包括选择适当的硬件平台和传感器、接口卡，以及编制应用软件。比如，用虚拟仪器方法实现示波器的功能，只需要一台个人电脑，一块模数/数模转换卡，加上专为示波器功能编写的软件就可以；而同样的硬件设备，加上为实现函数发生器功能而编写的软件，还可以成为一台函数发生器。虚拟仪器相比较于传统仪器，其功能更加灵活多变、易于扩展，而且由于和网络技术结合，它还能够与外界进行数据通信，实现远程操作。因此，虚拟仪器技术已经成为自动化测量的主流技术。

2. 虚拟仪器的硬件

本实验使用的硬件平台是个人电脑（PC 机），接口卡使用 NI（national instruments）公司的数据采集卡（DAQ 卡）。有些应用中要测量非电学量，还需要使用传感器将非电学量转换成电学量进行测量，比如测量温度，就可以利用铂电阻传感器连接成非平衡电桥电路将温度信号转换成电桥的非平衡电压，再用采集卡测量电压。如果用采集卡输出电压，在所需输出电压或者功率很大的情况下，还需要在输出端加上电压或者功率放大电路。

本实验所用的 DAQ 卡(NI USB-6343 型)采用 USB 接口传输数据,使用起来非常方便、灵活,便于携带。DAQ 卡可以实现对输入电压信号的采集、放大以及模/数转换,以及将计算机数字信号转换为模拟电压信号进行输出的功能,因此既可以进行测量,也可以实现控制。

数据采集卡的主要性能指标包括:输入通道数、信号输入方式、输入范围、阻抗、A/D 转换器的采样速率和位数、分辨率和精度等。本实验采用的数据采集卡可同时采集 32 路单端或 16 路差分输入信号,采集的信号电压范围从 $\pm 0.2\text{ V}$ 到 $\pm 10\text{ V}$ 分成四挡,分辨率为 16 位,最高采样率为 500 kS/s (S/s 即采样点数/秒);可同时输出 4 路模拟信号,最高输出速率为 900 kS/s 。

使用数据采集卡之前要正确安装硬件和驱动程序。同时还要根据需要在采集卡管理软件中恰当设置信号输入方式,比如是采用单端参考地输入还是双端差分输入。

采集原理详细内容请参考相关参考书。

3. 虚拟仪器的软件

用于虚拟仪器系统开发的比较成熟的软件开发平台主要有两大类:一类是通用的可视化软件编程环境,如 VC++, VB 等;另一类是专用于虚拟仪器开发的软件编程环境,如 Agilent 的图形化编程环境 Agilent VEE 和 NI 公司的 LabVIEW,以及后者的文本编程环境 LabWindows/CVI。

本实验使用的是 LabVIEW (laboratory virtual instrument engineering workshop) 开发平台,它将计算机数据分析和显示能力与仪器驱动程序整合在一起,为针对仪器的编程提供了很大的便利。而且,LabVIEW 是一种图形化编程语言,编程过程也就是设计流程图,即使初学者也能很快入门。

用 LabVIEW 开发平台编制的虚拟仪器程序简称为 VI。VI 包括三个部分:前面板 (Front Panel)、程序框图 (Block Diagram) 和图标/连接器。

前面板用于设置输入数值和显示输出量,相当于真实仪表的前面板。前面板上的图标,分为两类:输入类 (Controls, 用于输入) 和显示类 (Indicators, 用于输出),具体可以是开关、旋钮、按钮、图形、图表等表现形式。程序框图相当于仪器的内部功能结构,其中的端口用来和前面板的输入对象和显示对象传递数据,节点用来实现函数和功能子程序调用,框图用来实现结构化程序控制命令,连线则代表程序执行过程中的数据流。

关于用 LabVIEW 编写程序的进一步知识请查看附录。

4. 利用虚拟仪器测量伏安特性

本实验中利用采集卡的一个输出通道为整个测量电路供电,利用两个输入通道分别测量总电压和标准电阻上的电压;利用测量得到的电压数值和标准电阻数值就可以得到电路中的电流以及待测电阻上的电压。在程序控制下,电路电压由 0 V 开始逐渐增加到设定的上限值,输出电压每改变一次,测得一组电压、电流值,最后得到一个数组,经过线性拟合后就可以得到待测电阻值。测量原理见图 29-1。使用单端输入方式时,采集卡的各个输入通道共用地线,各通道测量的都是对地的电压,连线时要加以注意。



图 29-1 用虚拟仪器测量待测元件伏安特性的原理图。

【实验内容】

1. 初步熟悉 LabVIEW 开发环境的基本操作和编程方法

由开始>>所有程序>>NI LabVIEW 2020 SP1 (64-bit)启动 LabVIEW 程序。点击 **Create Project**，然后选择 **Blank VI**，点击 finish 进入 LabVIEW 环境。首先看到的是灰色的前面板（Front Panel）。按键盘上的快捷键<Ctrl-E>（或点击 **Window>>Show Block Diagram**）可以显示程序框图（Block Diagram），默认背景色是白色。

在前面板，点击 **View>>Controls Palette** 和 **Tools Palette** 可以分别显示控件选板和工具选板（参见图 29-2）。旋钮、开关等控制量在控件选板上，用鼠标左键单击这些图标并在前面板空白处再次单击鼠标左键，将其放置在前面板上。按<Ctrl-E>切换到程序框图，那么相应的端子和图标会出现在程序框图上，输入控件图标右侧出现指向外部的三角形，显示控件图标左侧出现向内的三角形。通过这些控件图标可以在前面板输入控制程序中的数据，或者将程序运行结果显示出来。

在程序框图窗口中，点击 **View>>Functions Palette** 显示函数选板，利用函数选板提供的循环、数学运算、比较以及公式节点等函数功能可以创建程序框图。

正式编程前，请先自己试着熟悉一下各选板上的图标和名称，学习选择并放置控件、点击右键查看快捷菜单，**学习使用工具选板上的操作工具、标签工具、定位工具、连线工具，熟悉各种快捷键，各种操作的详细说明请见附录。**

以下每个部分的实验内容请按文本给出的步骤操作。每个步骤的介绍中，通常首先说明的是该步骤的目的，然后是具体操作步骤。为保证程序不出错且执行正常，**请按顺序一步一步操作，不要跳步或打乱顺序。**框图仅用作摆放控件和连线的参考，**不要对照框图寻找控件。**

如果要删除某个对象，先选定位工具，鼠标放在要删除的对象上，单击左键，对象周围会出现虚框，按键盘上的 **Delete** 键，即可删除。如果要撤销前一次操作，按快捷键<Ctrl-Z>。

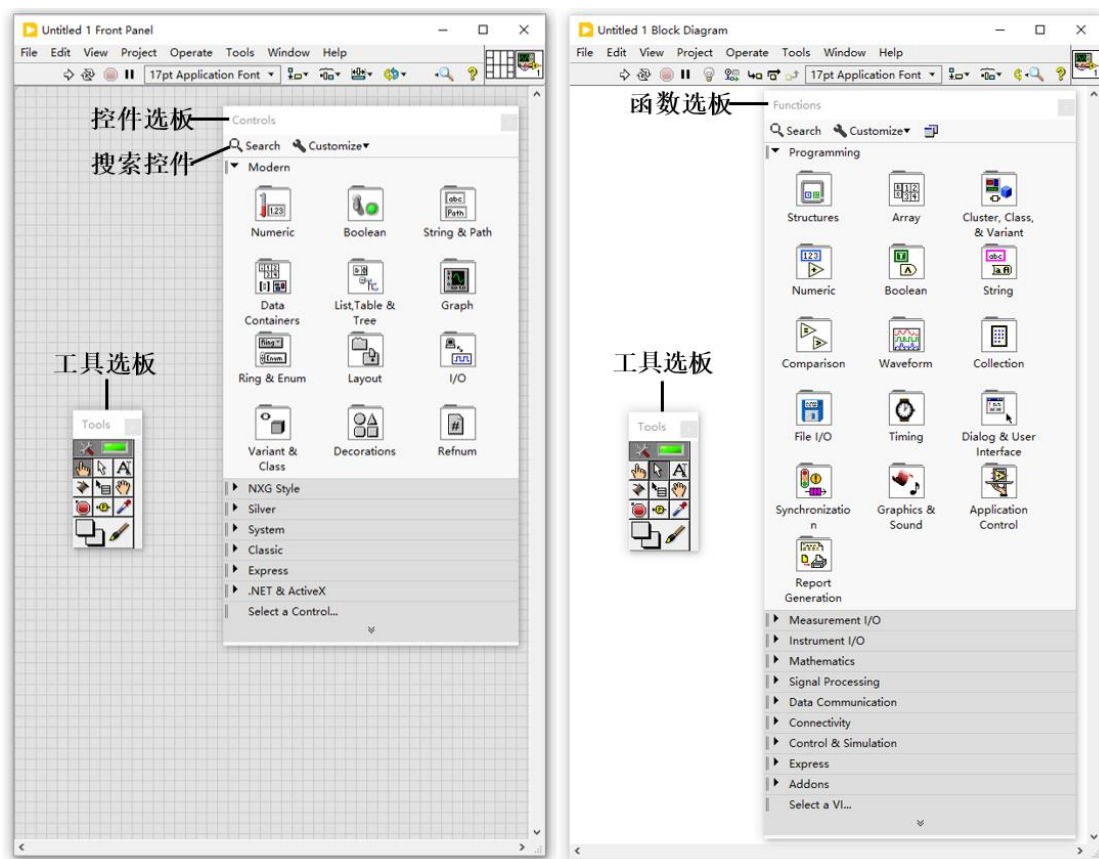


图 29-2 LabVIEW 2020 SP1 开发环境图。左侧是前面板窗口。右侧是程序框图窗口。

2. 用虚拟仪器实现波形发生器和示波器

2.1 编写波形发生器程序，包括前面板和程序框图

前面板图 and 程序框图如图 29-3 和图 29-4 所示。

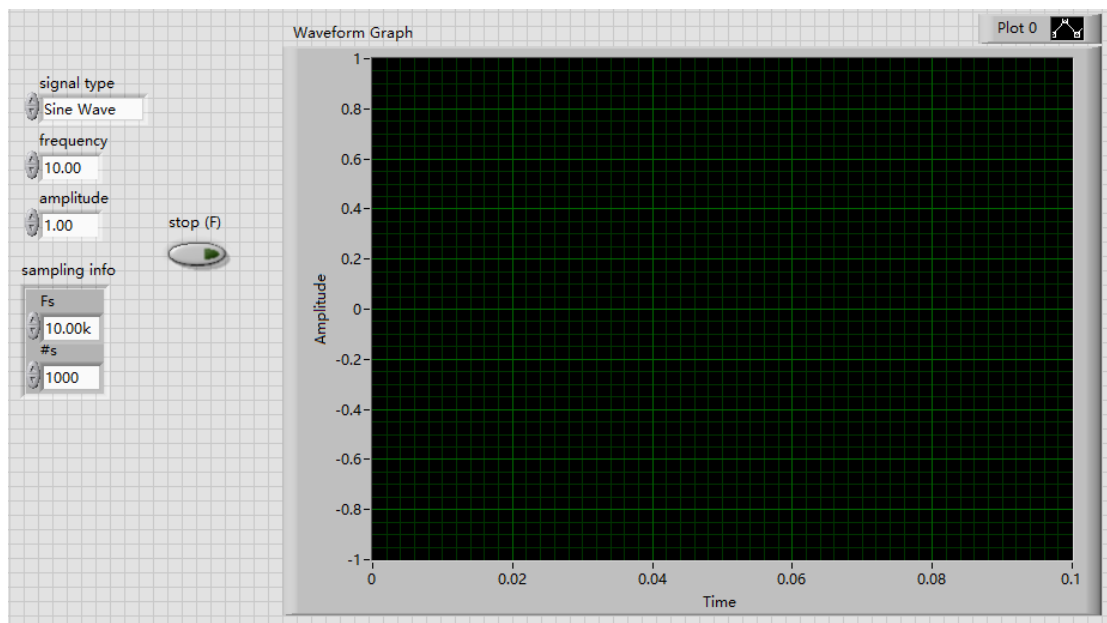


图 29-3 波形发生器前面板。

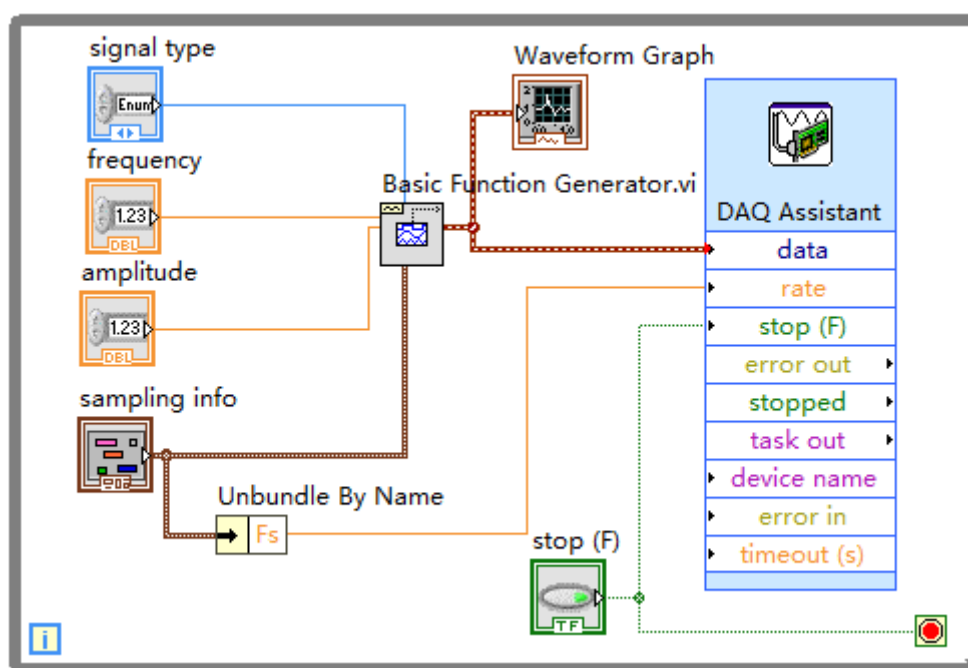


图 29-4 波形发生器程序框图。

2.1.1 创建前面板:

打开 LabVIEW 2020，点击创建一个 **project**，并选择 **Blank VI**。点击 File 菜单，选择 Save 保存，同时给该 VI 命名。编程过程中每操作几步，就按键盘上的<Ctrl-S>保存一下。



打开前面板窗口，在空白处点右键，弹出控件选板，在控件选板中选择下列控件，放在前面板上：


放上一个波形显示控件 **Waveform Graph** (**Controls>>Modern>>Graph**)。括号中是控件所在的文件夹，你也可按照控件名称在控件选板中搜索控件。找控件时一定要确认其名称




正确，因为有些控件的图标仅有细微区别。有时候同一个控件可能在多个文件夹都有，而且名称可能不完全一样，以下每个控件名称后的括号中也给出了可能的其他名称。找到控件，确认名称正确后，用鼠标左键单击控件，再在前面板上需要放置该控件的位置再次单击左键，控件即可添加到前面板上。

2.1.2 创建程序框图：

按快捷键<Ctrl-E>，打开程序框图窗口。

产生生成波形所需数据。放上一个 **Basic Function Generator**(或 **Basic FuncGen**) (**Functions>>Programming>>Waveform>>Analog Waveform**(或 **Analog Wfm**)>>**Waveform Generation**(或 **Generation**)。在控件上点鼠标右键，在弹出的菜单中选择 **Visible Items>>Label**，显示其名称。选择工具选板中的连线工具，把鼠标依次放在 **Basic Function Generator** 的其中四个接线端上 (signal type, frequency, amplitude, sampling info)，当该接线端闪烁并显示相应端口名称时，点击鼠标右键，选择 **Creat Control**，创建相应的输入控件（创建的四个输入控件会显示在前面板上，用于设置波形参数）。选择连线工具，把鼠标放在 **Basic Function Generator** 的 signal out 端口上，当该接线端闪烁并显示相应名称时，单击鼠标左键，拉出一根虚线（如果要中途停止连线操作，可单击鼠标右键），然后把鼠标移动到 **Waveform Graph** 显示控件上，点击左键，即可完成连线。

让采集卡连续输出波形信号。先打开采集卡的电源开关（开关在采集卡后方右侧），不然程序找不到采集卡。若采集卡开关打开，且与电脑 usb 口连接正常，则其前方右侧的蓝灯会亮。在框图中放入一个 **DAQ Assist** (**Functions>>Express>>Output**)，用于输出模拟电压信号，单击鼠标放置图标后程序显示 initializing，之后在生成的右侧对话框内点击 **Generate Signals>>Analog Output>>Voltage**，然后在生成的对话框 **Physical** 内点击选择 **Dev1** (USB-6343)的 ao0，表示使用设备号为 **Dev1** 的设备的 ao0 通道输出模拟电压信号，点击 finish 后生成界面对输出的模拟电压信号进行具体设置。首先在中间偏右侧的对话框 **Configuration>>Channel Settings>>Voltage Output Setup** 内将 **Signal Output Range** 设置为 -10~+10V，**Terminal Configuration** 默认为 **RSE**，表示输出的电压信号是 ao0 通道相对于地（也就是输出 GND 通道）的电压。然后在下方的对话框 **Timing Settings** 内将 **Generation Mode** 选为 **Continuous Samples**，表示该通道将连续输出数据，同时不勾选  (Use Waveform Timing)。最后点击 OK 按钮完成设置，程序自动生成相应的子 VI。此时会弹框询问“该任务所配置的模式要求将 DAQ 助手 Express VI 置于循环内。是否创建循环？该 VI 将不会再出现提示”，可以选择“否”，随后我们将手动创建循环。后期如果需要对 **DAQ Assist** 的设置进行修改，可以双击 **DAQ Assistant** 图标或者用右键点击 **DAQ Assistant** 图标再选择 **Properties** 重新进入设置页面。另外，**DAQ Assist** 在每次程序启动后会先初始化再运行，运行当中其采集或输出设置将保持不变，即使其设置相关的输入端数据（如 rate, number of samples 等）发生变化，它也不会响应。

设置 **DAQ Assist** 的外部参数。选定位工具，鼠标放在 **DAQ Assist** 控件下边框的蓝色方块上，按住左键，向下把 **DAQ Assist** 拽长，使它的各个端口名称显示完整后，松开左键。用连线工具连接 **Basic Function Generator** 的 signal out 端口和 **DAQ Assist** 的 data 端口。在 **sampling info** 控件上点右键，选择 **Cluster, Class & Variant Palette>> Unbundle By Name**，用连线工具连接 **sampling info** 和 **Unbundle By Name** 控件。用操作工具点击 **Unbundle**

By Name 的右方框，选中 Fs，这样可以从 cluster 类型的 sampling info 中提取其中 Fs 元素的数据。连接 Unbundle By Name 的 Fs 端和 DAQ Assist 的 rate 端口，使采集卡输出数据速率与波形数据的采样率一致。

设置循环结构。选择 **While Loop** 结构（**Functions>>Programming>>Structures**），按住鼠标左键用鼠标从左上到右下把所有的控件都框起来然后释放左键。选用连线工具，在 DAQ Assist 的 stop(F)端口点右键，选择 **Creat Control**，就创建了一个开关。把这个开关和 **While Loop** 结构右下角的 Conditional Terminal连接起来，这样当这个开关值为真时可以同时让信号输出和循环终止。

2.2 编写示波器程序，包括前面板和程序框图

前面板图 and 程序框图如图 29-5 和图 29-6 所示。

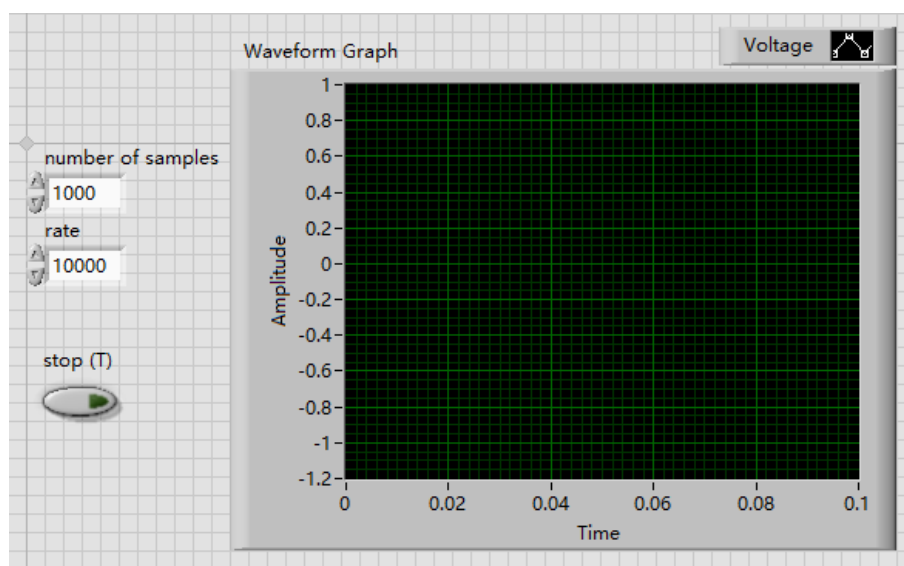


图 29-5 示波器前面板。

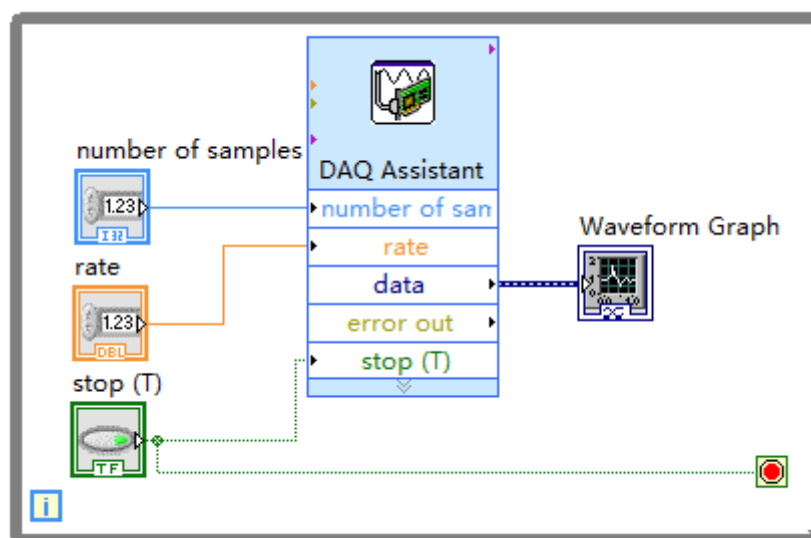


图 29-6 示波器程序框图。

打开 LabVIEW 2020，点击创建一个 **project**，并选择 **Blank VI**。点击 File 菜单，选择保存，同时给该 VI 命名。编程过程中每操作几步，就按键盘上的<Ctrl-S>保存一下。

本程序不需要在前面板创建控件。

创建程序框图：

打开框图窗口。

让采集卡连续采集信号。**打开采集卡电源开关**。在框图中放入一个 **DAQ Assist** (**Functions>>Express>>Input**)，用于采集模拟电压信号，单击鼠标放置图标后程序显示 initializing，之后在生成的右侧对话框内点击 **Acquire Signals>>Analog Input>>Voltage**，然后在生成的对话框 **Physical** 内点击选择 Dev1 (USB-6343) 的 ai0，表示使用设备号为 Dev1 的设备的 ai0 通道采集模拟电压信号，点击 finish 后生成界面对输出的模拟电压信号进行具体设置。首先在中间偏右侧的对话框 **Configuration>>Channel Settings>>Voltage Input Setup** 内将 Signal Input Range 设置为 -10~+10V，Terminal Configuration 选择为 **RSE**（注意，不同于前面信号输出，输入采集时默认设置不是 RSE，而是差分，这里必须改为共地 RSE），表示输入的电压信号是 ai0 通道相对于地（也就是输入 GND 通道）的电压。然后在下方的对话框 **Timing Settings** 内将 Acquisition Mode 选为 **N Samples**，并将 Samples to Read 的值设为 1000，表示一次采集 1000 个样本，Rate (Hz) 设为 10000，表示测量速率是每秒 10000 个样本。最后点击 OK 按钮完成设置，程序自动生成相应的子 VI。

设置 **DAQ Assist** 外部参数。用定位工具，向下把 **DAQ Assist** 拽长，使它的各个端口名称显示完整。用连线工具，在三个端口（number of samples, rate, stop(T)）上点右键，选择 **Creat Control**。在 data 端口上点右键，选择 **Creat>>Graph Indicator**。这样，在前面板上会出现两个参数设置端口、一个开关和一个波形图。

设置循环结构。选择 **While Loop** 结构 (**Functions>>Programming>>Structures**)，把所有的控件都框起来。用连线工具，把开关和 **While Loop** 结构右下角的条件接线端 (Conditional Terminal) 连接起来，这样当这个开关值为真时可以同时让采集和循环终止。

整个程序创建完毕。最后可以整理一下图标位置和连线（在需要整理的连线上单击右键，在快捷菜单中选择 **Clean Up Wire**）。


2.3 连电路，用示波器程序测量波形发生器程序输出的信号

关闭采集卡电源开关。


波形发生器程序使用采集卡的两个输出端口输出电压信号：AO GND 已通过采集卡连接大地，AO 0 输出相对于 AO GND 的电压（可正可负）。示波器程序使用采集卡的两个输入端口测量电压，AI GND 已通过采集卡连接大地，AI 0 与电路中需要测量电压的某点相连。找两个彼此不导通的接线柱，将 AI GND（端口号 19）和 AO GND（端口号 32）连到一个接线柱上，将 AI 0（端口号 1）和 AO 0（端口号 15）连到另一个接线柱上。


检查电路无误后，打开采集卡电源开关。


2.4 设置参数，运行程序

按照图 29-3 和图 29-5 设置两个程序前面板上的参数。让开关为 False（假值）状态（在开关上点右键，Visible Items>>Boolean Text，可显示开关状态）。确认 While 循环结构右下角的 Conditional Terminal 为图示状态  (Stop if True)，如果不是，可用操作工具点击切换。

可将当前参数值设为默认值：菜单栏点击 Edit 菜单，选 **Make Current Values Default**，这样每次重新打开程序后不必重设参数。

点击两个程序上方工具栏里的 Run（运行）右箭头按钮  运行程序，**让它们同时运行**。

改变波形（用操作工具 ，左键点击波形发生器程序前面板中的 signal type 控件白框内部，分别设为正弦波 Sine Wave、三角波 Triangle Wave 等），观察程序运行结果。

因为程序不会自动终止，最后要点击两个程序上方工具栏里的 Abort Execution（终止执行）红色圆点按钮，终止程序运行。保存程序，关闭程序窗口。关闭采集卡电源开关。

3. 用虚拟仪器测量电阻的伏安特性

3.1 编写程序，包括前面板和程序框图

前面板图和程序框图如图 29-7 和图 29-8 所示。

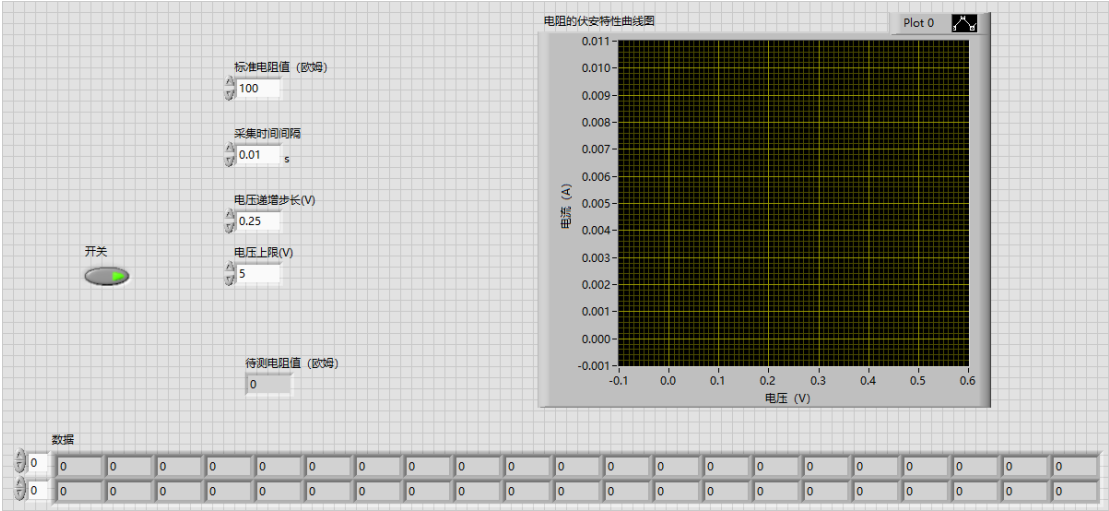


图 29-7 伏安法测电阻前面板图。

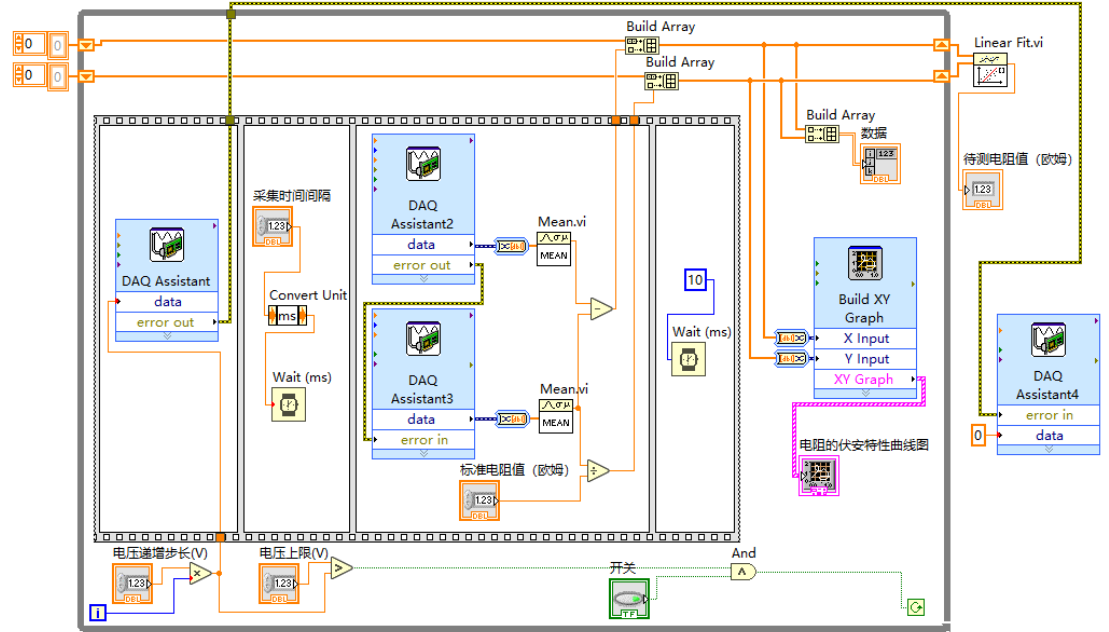



图 29-8 伏安法测电阻程序框图。


3.1.1 创建前面板

打开 LabVIEW 2020，点击创建一个 **project**，并选择 **Blank VI**。点击 File 菜单，选择保存，同时给该 VI 命名。编程过程中每操作几步，就按键盘上的<Ctrl-S>保存一下。

打开前面板窗口，在空白处点右键，弹出控件选板，在控件选板中选择下列控件，放在

前面板上：

放上一个 **Ex XY Graph** (或 **Express XY Graph**) (**Controls>>Modern>>Graph**)，用于显示电流-电压图。用标签工具，将图上方名字改成“电阻的伏安特性曲线图”，并将横坐标和纵坐标分别改成“电压 (V)”和“电流 (A)”。在图的右上角 Plot 0 右边的图线标志处，点右键，选择 **Common Plots**，选“点+线”模式。

放入一个开关 **Push Button** (**Controls>>Modern>>Boolean**)，用于手动控制程序进程，用标签工具，把它的标签 (Label) 改为“开关”。此开关状态为“on”时，框图的 while 循环才可继续。

放入四个数值型输入控件 **Numeric Control** (**Controls>>Modern>>Numeric**)，分别将四个输入控件的名字改成“标准电阻值 (欧姆)”、“采集时间间隔”、“电压递增步长(V)”、“电压上限(V)”。测量时，标准电阻值设为实际所用的值，本实验中为 100 欧姆。采集时间间隔用来设置每次输出电压改变和采集数据之间的时间间隔，让电路充分达到平衡再进行测量。在“采集时间间隔”控件上点右键，选择 **Visible Item>>Unit Label**，此时会在控件右方出现一个白色方块，直接输入“s”（输入前勿点鼠标，输入法必须是半角），这样采集时间间隔成为一个单位为秒的量。仅含电阻元件的电路稳定较快，采集时间间隔的值可以设为 0.01 s。电压递增步长、电压上限用来设置采集卡输出电压的递增步长和上限，请根据需要设置。由于采集卡输出会限制电流，实际输出电压不一定能达到设定的上限值。

放入一个数值型显示控件 **Numeric Indicator** (**Controls>>Modern>>Numeric**) 用于显示待测电阻值，将该显示控件的名字改成“待测电阻值 (欧姆)”。

用一个二维数组显示控件显示测量的电压和电流。先放入数组控件 **Array** (**Controls>>Modern>>Data Containers**)，再将它变成数值型的数组显示控件（创建一个数值型显示控件 **Numeric Indicator**，并用定位工具把它直接拖放到数组框内），再改变数组维数为二维（用定位工具，鼠标放在左侧索引框（底色为白色，左侧有箭头）下方出现的蓝色小方块处，按住左键向下拖拽使它有两个索引值，或者在索引框上点击右键弹出快捷菜单选择 **Add Dimension**），最后用定位工具放在数组边框右下角蓝色小方块处向右拖拽数据框使数据有 2 行，列数足够多，用于显示测量得到的数据点。把数组的名字“Array”改成“数据”。

完成后并输入了参数的前面板如图 29-7 所示。

3.1.2 创建程序框图

根据实验思路，先输出一个电压，等到稳定后分别测量总电压和标准电阻上的电压。控制程序执行顺序可以通过顺序结构来实现。在程序框图中放入一个顺序结构 **Flat Sequence** (**Functions>>Programming>>Structures**)，在其边框上点击鼠标右键，选择弹出菜单中的 **Add Frame After** 使顺序结构有四帧，用定位工具拉动边框调整结构大小。之前在程序框图中已经有一些控件图标，它们分别对应于前面板上的各个控件，现在可以按图 29-8 把这些图标移动到顺序结构的各帧之中。

首先，让 DAQ 卡输出电压。打开采集卡电源开关。在顺序结构的第 0 帧中放入一个 **DAQ Assist** (**Functions>>Express>>Output**)，用于输出模拟电压信号，单击鼠标放置图标后程序显示 initializing，之后在生成的右侧对话框内点击 **Generate Signals>>Analog Output>>Voltage**，然后在生成的对话框 Physical 内点击选择 Dev1 (USB-6343) 的 ao0，表示使用设备号为 Dev1 的设备的 ao0 通道输出模拟电压信号，点击 finish 后生成界面对输出的模拟电压信号进行具体设置。首先在中间偏右侧的对话框 **Configuration>>Channel Settings>>Voltage Output Setup** 内将 Signal Output Range 设置为 -10~+10V，Terminal Configuration 默认为 **RSE**，表示输出的电压信号是 ao0 通道相对于地（也就是输出 GND 通

道) 的电压。然后在下方的对话框 **Timing Settings** 内将 **Generation Mode** 选为 **1 Sample (On Demand)**, 表示该通道一次输出一个固定的电压值。最后点击 **OK** 按钮完成设置, 程序自动生成相应的子 VI。具体的输出电压值我们以后再设置。后期如果需要对上述设置进行修改, 可以双击 **DAQ Assistant** 图标或者用右键点击 **DAQ Assistant** 图标再选择 **Properties** 重新进入设置页面。

然后, 让程序等待一定时间, 用于等待电阻上的电流达到稳定。在顺序结构的第 1 帧中放入一个 **Wait(ms)** (**Functions>>Programming>>Timing**), 再放入一个 **Covert Unit** (在 **Functions>>Programming>>Numeric>>Conversion** 下), 在模块中间格子输入“ms”, 用于将单位 s 转换成 ms, 将 **Covert Unit** 的输入端和“采集时间间隔”的输出相连, 输出端和 **Wait(ms)** 的 **Milliseconds to wait** 端相连。

然后, 用 DAQ 卡采集总电压和标准电阻上的电压, 计算待测电阻上的电压、电流值。在顺序结构的第 2 帧中先后放入两个 **DAQ Assist** (**Functions>>Express>>Input**), 分别用来测量总电压和标准电阻上的电压。单击鼠标放置第一个 **DAQ Assist** 图标后在生成的右侧对话框内点击 **Acquire Signals>>Analog Input>>Voltage**, 并在生成的对话框 **Physical** 内点击选择 **Dev1 (USB-6343)** 下的 **ai0**, 表示使用设备号为 **Dev1** 的设备的 **ai0** 通道采集输入的模拟电压信号, DAQ 卡外部接线时将这个通道连接到总电压, 点击 **finish** 后生成界面对输入电压信号进行具体设置。首先在中间偏右侧的对话框 **Configuration>>Channel Settings>>Voltage Input Setup** 内将 **Signal Input Range** 设置为 **-10~+10V**, **Terminal Configuration** 选择为 **RSE** (注意, 不同于前面信号输出, 输入采集时默认设置不是 **RSE**, 而是差分, 这里必须改为共地 **RSE**), 表示输入的电压信号是 **ai0** 通道相对于地 (也就是输入 **GND** 通道) 的电压。然后在下方的对话框 **Timing Settings** 内将 **Acquisition Mode** 选为 **N Samples**, 并将 **Samples to Read** 的值设为 100, 表示一次采集 100 个样本, 后面我们将把这 100 个样本的平均值作为一个数据采集点的结果, 以提高信噪比; **Rate (Hz)** 设为 1000, 表示测量速率是每秒 1000 个样本, 这样采集一个数据点也就是 100 个样本需要的时间是 0.1 s, 点击 **OK** 按钮完成设置。对第二个 **DAQ Assist** 做类似的设置, 只是在对话框 **Physical** 内改为选择 **Dev1 (USB-6343)** 下的 **ai1**, 表示这个子 VI 使用 **ai1** 通道采集输入的模拟电压信号, DAQ 卡外部接线时将这个通道连接到标准电阻上的电压。两个采集信号的 **DAQ Assistant** 设置完成后, 将前一个 **DAQ Assistant** 的 **error out** 输出端连接到后一个 **DAQ Assistant** 的 **error in** 输入 (将 **DAQ Assistant** 图标的下沿向下拉可以显示出全部输入、输出端口), 通过上述 **error** 数据流的设置, 可以控制程序的同一个部分按照 **error** 数据流的顺序先后运行多个数据采集过程; 否则, 如果在程序的同一个部分同时运行多个数据采集过程, 采集卡无法正常工作, 程序将报错。上述两个数据采集的 **DAQ Assistant** 得到的结果是 100 个数 (因为前面的 **Samples to Read** 的值设为 100), 需要对它们求平均值, 放入两个 **Mean** (**Functions>>Mathematics>>Prob & Stat**), 把两个 **DAQ Assistant** 的 **data** 输出端和 **Mean** 的 **X** 输入端相连, 则 **Mean** 的 **mean** 输出端输出的是电压平均值。连线时由于 **DAQ Assistant** 的 **data** 端输出数据类型与 **Mean** 的输入数据类型不同, 连线时中间会自动出现一个数据类型转换节点。之后用总电压的平均值减去标准电阻上的电压平均值得到待测电阻上的电压, 再把标准电阻上的电压平均值除以标准电阻, 得到电流值。减法 (**Subtract**) 和除法 (**Divide**) 函数都在 **Functions>>Programming>>Numeric** 目录下。给函数连线时, 注意看一下输入输出端口名称, 以区分被减数和减数, 或者把鼠标放在函数图标上, 按 <Ctrl-H> 看简要说明。

然后, 再让程序等待一定时间, 以减少对数据测量过程的影响。在顺序结构的第 3 帧中放入 **Wait(ms)**, 在 **Wait(ms)** 的输入端点右键创建常数, 选择 **Create>>Constant**, 并将其常数改改为 10 (表示等待 10 ms)。

利用上面的顺序结构我们只完成了一组数据点的测量过程, 实验中我们需要逐渐改变电

压，获得电阻的一条完整的伏安特性曲线。下面我们来实现电压值的改变。

我们通过 **While Loop** 来实现电压的改变。放入的 **While Loop** 要包含先前的顺序结构、“数据”、“电阻的伏安特性曲线图”和“开关”。先在 **Functions>>Programming>>Structures** 下单击选择 **While Loop** 控件，然后在程序框图中，在顺序结构左上角以外点击鼠标左键并按住，向顺序结构右下角拖拉 **While Loop** 框，将顺序结构等对象完全包含之后释放鼠标左键，这样就把顺序结构等对象放在了 **While Loop** 中。或者，先放一个空的 **While Loop**，再用定位工具拉出一个框框住整个顺序结构，把它拖进 **While Loop** 中。如果你只想删除 **While Loop** 结构框，而保留其中的顺序结构，可以在 **While Loop** 边框上点右键，在快捷菜单中选择 **Remove While Loop**。

我们希望 DAQ 卡输出电压从 0 V 开始增加到某个上限值，每隔一定步长测一组数据。等间隔递增的实现：在 **While Loop** 框左下角放入 **Multiply**（乘法）函数（**Functions>>Programming>>Numeric**），将循环变量 *i* 和“电压递增步长(V)”分别连入乘法函数的左侧两个输入端，将乘积也就是乘法函数的输出端与顺序结构第 0 帧中的 **DAQ Assistant** 的 data 输入端相连，从而控制 **DAQ Assistant** 输出的电压值。**While Loop** 的循环变量 *i* 从 0 开始计数，循环条件在框内语句都执行之后才进行判断，因此 **While Loop** 至少执行一次。限制电压范围不超过上限值：放入 **Greater?**（大于）函数（**Functions>>Programming>>Comparison**），将“电压上限(V)”连入“大于函数”的左上输入端 *x*，将“乘法函数”的输出端连入“大于函数”的左下输入端 *y*；“大于函数”的输出结果再和开关控件作逻辑与运算（**And** 函数，在 **Functions>>Programming>>Boolean** 下）后和 **While Loop** 框内右下角的 **Conditional Terminal** 相连，用于控制循环。注意用右键点击右下角的 **Conditional Terminal**，在菜单中把循环条件设为 **Continue If True**。

用移位寄存器 **Shift Register** 实现数据的传递和实时显示，**Shift Register** 的功能是在相邻两次循环之间传递数据。在 **While Loop** 左边框（或右边框）上点右键选择 **Add Shift Register**，加上两个 **Shift Register**，分别用来存储并传递电压和电流的测量数据。在循环中放入两个 **Build Array**（**Functions>>Programming>>Array**），用定位工具，向下拖放其图标使其有两个输入端，将图标上方的 **element** 端口和左边的 **Shift Register** 相连、下方 **element** 端口和电压（或是电流）相连，图标输出端 **appended array** 和右端的 **Shift Register** 相连（此处 **Build Array** 的作用是在来自 **Shift Register** 的原来一维数组中的数据后面串接上来自下方 **element** 输入端的新测量数据，成为一个新的一维数组）。注意在左右两端 **Shift Register** 之间的线完全连好以前，连线会显示为断线；两边都连好之后，连线才会显示正常。分别在左边两个 **Shift Register** 的小三角上点右键，选择 **Create>>Constant**，创建两个空的数组，用于初始化数据，清空 **Shift Register**。

显示测量数据。在程序框图再放入第三个 **Build Array**，用定位工具，向下拖放其图标使其有两个输入端。把之前两个 **Build Array** 输出的电压和电流数组分别和第三个 **Build Array** 的输入端相连，把 **Build Array** 的输出和名为“数据”的数组显示控件相连（由于输入的两个一维数组大小相同，此处 **Build Array** 的作用是将来自两个 **element** 输入端的电压一维数组和电流一维数组合并成一个二维数组）。

显示伏安曲线。把 **Build Array** 输出的电压一维数组和 **Build XY Graph** 的 **X Input** 相连，电流一维数组和 **Y Input** 相连。由于显示数组控件“数据”和显示图形控件“电阻的伏安特性曲线图”都在 **While Loop** 以内，因此每次循环中“数据”和“电阻的伏安特性曲线图”都会更新一次。

计算电阻值。在 **While Loop** 框外右边放入一个 **Linear Fit**（**Functions>>Mathematics>>Fitting**），将 **While Loop** 的右边框上的 **Shift Register** 传递出来的电流数组和 **Linear Fit** 的 **X** 输入端相连，电压数组和 **Y** 输入端相连，把 **Linear Fit** 的



Slope 输出端和“待测电阻值（欧姆）”显示控件相连，显示线性拟合得到的电阻测量结果。

结束测量后让采集卡输出电压归零。用定位工具，左键单击选中第 0 帧中的 **DAQ Assistant**，按快捷键<Ctrl-C>复制，在 **While Loop** 框外右边空白处点击一下左键，按快捷键<Ctrl-V>粘贴，即创建出了对同一个输出通道 AO0 进行相同模式设置的 **DAQ Assistant4**。用连线工具，在其 data 输入端点右键创建常数 0，并将第 0 帧中 **DAQ Assistant** 的 error out 输出端和 **DAQ Assistant4** 的 error in 输入端相连。

3.2 正确连接采集卡和外部电路

关闭采集卡电源开关，按图 29-1 连接电路。以 100 欧姆电阻为标准电阻，1k 欧姆和 50 欧姆的电阻为待测电阻。图 29-1 中，“LabVIEW 供电”表示用 DAQ 卡输出电压：模拟输出的 AO GND 端口连到图中接地标记处（相当于电源接地端），模拟输出端口 AO 0 连到电路另一端（相当于电源正端）。“测总电压”和“测电压算电流”表示用 DAQ 卡两个输入通道分别测量总电压和标准电阻上的电压：将模拟输入的 AI GND 端口（相当于电压表负端或接地端）连到图中接地标记处，模拟输入端口 AI 0（测总电压）和 AI 1（测标准电阻上电压）分别连到图中所示位置（AI 0 和 AI 1 相当于电压表正端）。DAQ 卡各个连线对应的端口参见 DAQ 卡上方的铭牌，上面标出了不同数字端口对应的端口标号。**检查电路无误后，打开采集卡电源开关。**

3.3 运行程序

先测 1k 欧姆电阻，后测 50 欧姆电阻。将前面板窗口中的**标准电阻值**设为 100 欧姆，采集时间间隔设为 0.01s，**根据需要设置“电压递增步长(V)”和“电压上限(V)”**，选择操作工具  并点击开关使其为“on”状态。点击屏幕左上角的右箭头  运行程序。循环结束程序会自动停止。

完成测量后，可以调整一下曲线图的显示范围（数据点显示完整且充满图面）和显示数据的精度（避免刻度显示全为零、无法读数）。在前面板中显示结果的曲线图上点击鼠标右键并选择 **Properties**，然后在弹出的窗口中设置：（1）显示范围设置：**Properties>>Scales**，上方选择要设置的图轴（X 轴或 Y 轴），右边不勾选 Autoscale，输入图轴的最小值（Minimum）和最大值（Maximum），Scaling Factors 不要修改；（2）数据精度设置：**Properties>>Display Format**，对话框下方选择 Default editing mode，上方左边 Type 选 Floating point，右边 Precision Type 选 Digits of precision，然后中间 Digits 设置位数（此时表示小数点后的位数）。点下方 OK 确定。

分别对 1k 欧姆大电阻和 50 欧姆小电阻重复测量三次，分析实验结果。测 50 欧姆小电阻时，电压增大到一定程度后不再增加，为什么？试将电压递增步长改为恰当的大小再测量。

保存程序并退出。**关闭采集卡电源开关。**

***4. 选做内容。**对前面的程序稍做修改，测量并绘制稳压二极管伏安特性曲线。实验中需要限制稳压二极管正向和反向通过的**最大电流不超过 8mA**，稳压二极管反向稳定电压大概为 5V 多。

【思考题】

- （1）虚拟仪器系统与传统仪器的区别体现在哪些方面，请简要说明。
- （2）你对虚拟仪器系统在物理实验中的应用有何设想？能否结合具体的实验给出一个简单的设计思路。
- （3）本实验中为什么不同时用采集卡两个通道直接测量待测电阻上的电压和标准电阻上的电压？
- （4）利用虚拟仪器系统测量电阻伏安特性并计算其阻值时，结果的误差可能来源于哪

些方面？

(5) 实验内容 4 中测量并绘制二极管伏安特性曲线时要注意什么？

【参考资料】

- 1 石博强等编著. LabVIEW 6.1 编程技术实用教程. 北京：中国铁道出版社，2002.
- 2 汪敏生等译著. LabVIEW 基础教程. 北京：电子工业出版社，2002.
- 3 刘君华等编著. 虚拟仪器图形化编程语言 LabVIEW 教程. 西安：西安电子科技大学出版社，2001.
- 4 [美] Robert H. Bishop 著，乔瑞萍等译. LabVIEW 6i 实用教程. 北京：电子工业出版社，2003.
- 5 赵会兵编著. 虚拟仪器技术规范与系统集成. 北京：清华大学出版社/北方交通大学出版社，2003.
- 6 [美] John Essick 著，邓科等译. 精通 LabVIEW 教程——由浅入深的范例学习（第二版）. 北京：电子工业出版社，2017.

【附录】

LabVIEW 编程简介

(1) 新建 LabVIEW 程序

双击 LabVIEW 2020 图标，出现启动画面，点击左侧窗口的 **Create Project**，然后选择 **Blank VI**，点击 finish 进入 LabVIEW 环境，就可以打开一个新的 LabVIEW 程序，可以看到它由前面板（Front Panel）和程序框图（Block Diagram）组成。如果要打开一个已有程序，点击右侧窗口的 **Open Existing**，然后选择要打开的程序。

(2) 前面板窗口（Front Panel）和控件选板（Controls Palette）

前面板窗口是用户界面，也就是 VI 的虚拟仪器面板。前面板上上面有两类控件：输入控件（controls）和显示控件（indicators）。输入控件有开关、旋钮、数字或文本输入框等，用来让用户进行输入或控制操作。显示控件有图形、数字或文本输出框等，用来显示测量或计算结果。

所有控件都要通过控件选板进行选择，再添加到前面板上。控件选板可以用两种方法打开：1) 在前面板空白处点击右键；2) 选择菜单 **View>>Controls Palette**。

控件选板如图 29-2 左图所示。其上的控件主要按照数据类型分组。寻找控件有两种方式：1) 根据所属类型找到其分组，再在该分组中找到该控件并点击；2) 如果知道控件名称，还可以利用控件选板上的搜索栏（Search）直接搜索，双击搜索结果可高亮显示其在选板中的位置。

向前面板添加控件的方法：鼠标左键点一下控件图标并松开，控件就粘附在光标上了；将光标移到要放控件的位置，再点一下鼠标左键，就将控件释放了。

对于某些数据类型的输入（或显示）控件，可以在该控件上单击鼠标右键，选择 **Change to indicator**（或 **control**），将其改变为同一数据类型的显示（或输入）控件。

(3) 程序框图（Block Diagram）和函数选板（Functions Palette）

创建前面板窗口后，还需要在程序框图中编写图形化代码，用来控制前面板中的对象。前面板和程序框图之间的切换，可以直接用鼠标在不同窗口点击；如果窗口已经全屏，可以点击 **Window>>Show Front Panel**（或 **Show Block Diagram**）进行切换。在程序框图中，可以看到添加在前面板上的控件在程序框图中已经生成了相应的输入、输出端。此外，可能还需要在程序框图中添加一些子 VI、函数、常量、结构和连线（用来在对象间传递数据），才能实现仪器的输入和输出功能。

在程序框图中，我们可以看到输入控件和显示控件的图标有两个显著区别：1) 输入控

件的边框较粗，显示控件的边框较细；2）它们的图标上都有一个表示数据流向的接线端箭头，输入控件的箭头向外，而显示控件的箭头向内。

向程序框图中添加子 VI、函数、常量、结构，需要通过函数选板来选择和添加。函数选板如图 29-2 右图所示。打开函数选板也有两种方法：1）在程序框图空白处点击右键；2）选择菜单 **View>>Functions Palette**。

如果知道对象名称，可以使用函数选板上的搜索功能查找对象。双击搜索结果可高亮显示其在选板中的位置。

如果要查看函数或 VI 的功能，可将鼠标移到控件上，按快捷键<Ctrl-H>，即可出现 **Context Help** 窗口并出现相应的说明，点击窗口中的 **Detailed help** 将出现更详细的说明信息。

（4）数据流、连线和数据类型

① 数据流工作方式













LabVIEW 按照数据流模式运行程序，只有当模块要求的输入数据完全到达这个模块时才能执行，然后向其所有的输出端口输出数据，这些数据再沿数据线流向其它模块。数据流经节点的过程决定了程序框图上 VI 和函数的执行顺序。

在数据流模式下，两个彼此没有数据依赖的程序块可能会同时运行，如果要强制它们的运行顺序，可以使用顺序结构（Flat Sequence）或错误簇（Error）等编程技术。

② 连线

连线用于在程序对象之间传递数据。连线上数据的传递是有方向的。输入端子、常数、函数或子 VI 的输出端都可以作为数据源。数据接收端可以是显示端子、函数或 VI 的输入端。一条连线只有一个数据源，但可以连接多个数据接收端。此外，连线上传递的数据具有一定类型和维数，其数据源和数据接收端的数据类型和数据维数必须相同。因此，正确的连线必须保证两点：1）连线方向正确；2）连线输入和输出端必须与连线上传递的数据兼容。表 29-1 显示了最常见的连线类型。

表 29-1 常用连线类型

类型	标量	一维数组	二维数组	颜色
整型				蓝色
浮点型				橙色
布尔型				绿色
字符型				红色

当连线错误时，连线会表示成断线，它是一条中间带小红叉的黑色虚线。这时你应将连线工具移动到它上面，LabVIEW 会给出两端的数据类型和流向信息，你可以检查并改正。

添加新的模块和连线可能会对已设置好的部分有影响，而且原来的错误连线也可能对新添加的连线有影响。

如果已经连了一个接线端，想取消连线，可单击鼠标右键。如果想删除连线，用定位工具选中它，即可删除。


连线结束后，右键单击连线，从快捷菜单中选择 **Clean Up Wire**，可使 LabVIEW 自动选择连线路径。按快捷键<Ctrl-B>可删除在程序框图中的所有断线。

③ 数据类型


LabVIEW 的基本数据类型有五种：Numeric（数值）、Boolean（逻辑）、String（字符串）、Enum（枚举）、Ring（环型枚举），Ring 和 Enum 很类似，可以循环枚举。


Numeric 类型的数据按精度又分若干种类型，与标准 C++ 的数据类型基本是一致的。其代表符号直观的表现其类型。可以在数值对象上点右键，通过弹出菜单中 **Representation** 修改类型。在程序框图中，不同数据类型的对象有不同颜色，其颜色和数据类型的对应关系与连线相同。


（5）工具选板功能介绍：


在前面板窗口和程序框图中都可以使用工具选板（见图 29-2），可以通过 **View>>Tools Palette** 打开和关闭它。它提供了各种用于创建、修改和调试 VI 程序的工具。当从选项板内选择了任一种工具后，鼠标箭头就会变成该工具相应的形状。工具选板上的  是控制自动选择工具的开关：当灯亮时，鼠标如果移到某个模块上，程序会自动选择工具，方便操作。


下面介绍常用工具按钮的功能：


：操作工具（Operate Value），用于操作前面板的控制和显示，比如按下开关、扭动旋钮等；或者用于改变程序框图中布尔常量的值。当使用它向数字或字符串控件中键入值时，工具会变成标签工具。

：定位工具（Position/Size/Select），用于选择、移动或改变对象的大小。1）选择对象：点击一个控件，就可选中它；如果想选择多个控件，可以用鼠标在屏幕上拉一个框，这时框中的控件同时都被选中。选中对象后，可以移动、复制或删除该对象。2）移动对象：左键点击对象并一直按住，待移动到目标位置之后再松开。3）改变对象大小：选择对象后，将光标移动到边框上，此时矩形边框四角和四边中央会出现小方块，将光标移到小方块上时，光标会变成相应的双箭头形状，此时可以沿所需的方向对对象拉伸或压缩。

：标签工具（Edit Text），用于向输入控件中输入文本、编辑文本或者创建自由标签。

：连线工具（Connect Wire），用于在程序框图中连接对象。连线方法：将连线工具移至接线端，此时将出现含有接线端名称的提示框，而且图标上的接线端将会闪烁。由于有的函数或 VI 有多个接线端，因此要注意选对所需的接线端，当所需接线端闪烁时，单击鼠标即可。将连线工具在两个对象的接线端上各单击一次，就可在这两个对象之间创建连线。如果已经连了一个接线端，想取消连线，可单击鼠标右键取消。

：断点工具（Set/Clear Breakpoint），用于在 VI、函数、节点、连线和结构中设置断点，断点位置将暂停运行。

：探针工具（Probe Data），用于在程序框图的连线上创建探针。使用探针工具可查看 VI 运行中连线上传递的数据值。



（6）快捷菜单和属性对话框


所有的 LabVIEW 对象都有快捷菜单，右键单击对象，即可呼出快捷菜单。通过快捷菜单，可以改变对象的外观或运行方式。比如，选择可见项目（visible items）、为某个输入端创建常数（create constants）、改变数据精度类型（representation）、设置属性（properties）等等。

在快捷菜单中选择设置属性后，会弹出属性对话框，其选项与快捷菜单中的选项类似。


（7）前面板窗口、程序框图工具栏和程序调试工具



前面板窗口和程序框图上方都有一条工具栏，上面有运行、中止、调试程序的工具按钮，它们的用法如下：




：运行（Run）。如果 VI 有编译错误，此按钮将变成；此时单击该断箭头按钮，会显示错误列表；双击错误列表中的项目，程序中出错的地方会高亮显示出来。

：连续运行（Run Continuously）。再次单击此按钮可停止连续运行。

：中止执行（Abort Execution）。点击可终止程序运行。

：暂停（Pause）。单击此按钮后，程序暂停，按钮变成红色；再次单击后，程序继续运行。

：高亮显示执行过程（Highlight Execution），只出现在程序框图工具栏中。单击此按钮，按钮会变成，此时，可以看到每个步骤的数据值以及数据在连线上的流动情况。注意，在高亮显示执行过程状态下，程序执行速度会显著降低。

小结一下调试程序的常用方法：1）程序编译出错时，可单击按钮找到错误之处并修正；2）调试程序时，可以用高亮显示执行过程按钮检测程序运行情况；3）对于特别关注的某些数据值，可以使用探针工具。其它还有设置断点和单步运行等方法，具体请查阅软件帮助文档。

（8）快捷键和编程小技巧

<Ctrl-S>：保存文件。

<Ctrl-R>：运行 VI。

<Ctrl-E>：切换前面板窗口和程序框图。

<Ctrl-H>：打开帮助窗口（Context Help），当把鼠标放到任何感兴趣的模块对象上时，就会在帮助窗口中显示相应的帮助信息。

TAB 或 Spacebar：切换选择工具。多按几下直到鼠标变成你想要的工具的形状。

<Ctrl-Z>：撤销前一操作。

<Ctrl-Shift-Z>：重做刚撤销的操作。

<Ctrl-B>：去除所有错误的连线。

如果要观察或改变模块或控件的各种属性，在其图标上按右键。

双击某个控件/模块可以看到与它对应的模块/控件端。