第 3 章

数据类型和操作

本章学习目标

- 熟练掌握 LabVIEW 的基本数据类型及操作
- 熟练掌握数组、簇和波形数据的操作

LabVIEW 作为一种通用的编程语言，与其他文本编程语言一样，支持所有的数据类型，数据类型决定数据的存储空间大小与操作方式。选择合适的数据类型不但能提高程序的执行效率，而且还能减少内存空间的占用。在程序设计开发过程中，数据操作是最基本的操作，不同的数据类型在程序框图中对应不同的端口图标和颜色，且具有不同的数据操作方式。本章主要介绍 LabVIEW 中常用的数据类型，以及与这些数据类型相关的数据运算。

3.1 LabVIEW 的基本数据类型

LabVIEW 的基本数据类型包括数值型、布尔型、字符串、文件路径等几类。在 LabVIEW 中，不同的数据类型通常采用不同的颜色显示，见表 3-1。

表 3-1 各种数据显示的颜色

数据类型	颜 色	标 量	一维数组	二维数组
整数型数值	蓝色			
浮点型数值	橙色			
布尔型	绿色			
字符串	粉红色			

3.1.1 数值型

数值型是 LabVIEW 的一种基本的数据类型，可以分为浮点型、整数型和复数型 3 种基本形式，其类型的详细分类见表 3-2。

表 3-2　数值类型表

数值类型	图　标	特　性
有符号 64 位整数		用 64 位存储整数数据，可为正数也可为负数
有符号 32 位整数		用 32 位存储整数数据，可为正数也可为负数
有符号 16 位整数		用 16 位存储整数数据，可为正数也可为负数
有符号 8 位整数		用 8 位存储整数数据，可为正数也可为负数
无符号 64 位整数		用 64 位存储整数数据，仅表示非负整数
无符号 32 位整数		用 32 位存储整数数据，仅表示非负整数
无符号 16 位整数		用 16 位存储整数数据，仅表示非负整数
无符号 8 位整数		用 8 位存储整数数据，仅表示非负整数
双精度浮点型		64 位 IEEE 双精度格式
单精度浮点型		32 位 IEEE 单精度格式
扩展精度浮点型		保存为独立于平台的 128 位格式
定点数		64 位或 72 位存储数据，数值大小根据平台有所不同
复数双精度浮点型		128 位，实部与虚部分别与双精度浮点型相同
复数单精度浮点型		64 位，实部与虚部分别与单精度浮点型相同
复数扩展精度浮点型		256 位，实部与虚部分别与扩展精度浮点型相同

显然，不同数据类型的差别在于存储数据使用的位数和表示值的范围。

控件选板上的数值子选板分别位于“新式”→“数值”，“银色”→“数值”，“系统”→“数值”，“经典”→“经典数值”及 Express→“数值输入控件”和“数值显示控件”子选板中，以满足编程的需要。在每个子选板中包含了多种不同形式的数值输入控件和显示控件，它们的外观各不相同，有数字输入框、滚动条、滑动杆、进度条、旋钮、转盘、仪表、量表、液罐、温度计、颜色盒等，如图 3-1 所示。但是这些对象在本质上是完全相同的，都是数值型。对象属性的设置方法也基本相同，均通过其快捷菜单来设置。

在编程语言中，数据通常分为变量与常量，LabVIEW 控件选板中的控件相当于变量，而常量则位于程序框图函数选板的“编程”→“数值”中，如图 3-2 所示。

在一个数值控件上单击鼠标右键，弹出如图 3-3 所示的快捷菜单。其中，“显示项”给出了数值控件可以添加的所有附加元素的开关选项列表，有标签、标题、单位标签、基数和增量/减量；“查找接线端”用于从前面板窗口定位该控件在程序框图中的接线端子，在程序框图接线端上弹出的快捷菜单里，该选项为“查找输入控件”，用来从程序框图定位前面板上的控件；“转换为显示控件”可以将输入控件变换为显示控件，对于显示控件，该选项为“转换为输入控件”，则将显示控件转换为输入控件；“转换为数组”是将单个数值控件转换为同数值类型的一维数组；“说明和提示”可以定义控件的说明和提示；“创建”子菜单可以为数值控件创建局部变量、引用、属性节点及调用节点；“替换”子菜单可以把数值控件变换为其他的前面板控件；“数据操作”子菜单下的“重新初始化为默认值”选项把数值输入

图 3-1 “数值”子选板

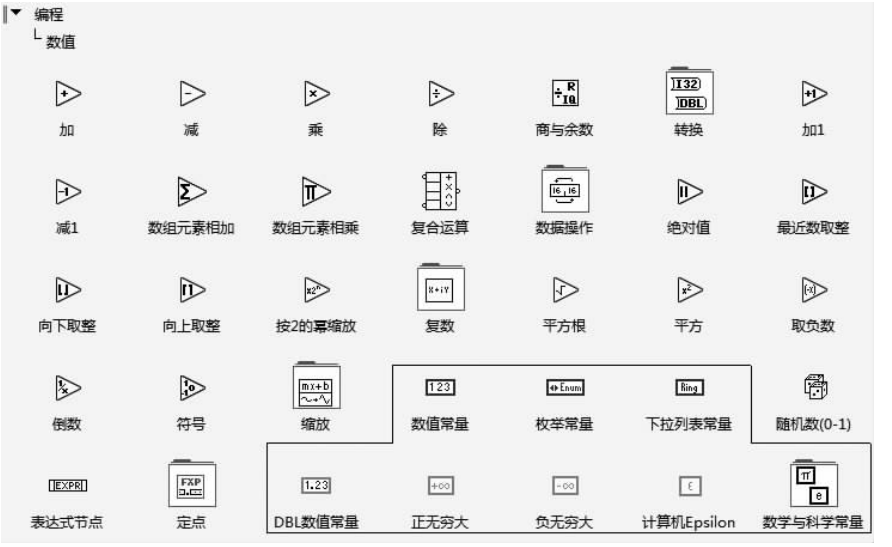


图 3-2 “函数”选板中的“数值”常量

控件还原为默认值，“当前值设为默认值”选项把当前值设置为默认值，“剪切数据”“复制数据”和“粘贴数据”选项则用于在数值控件之间复制数据。

从快捷菜单中可以打开“属性”对话框，在对话框中定义控件的各种属性，如图 3-4 所

示。很多快捷菜单选项都能在这里找到，在快捷菜单里和在属性对话框里定义这些控件属性和参数没有任何区别。

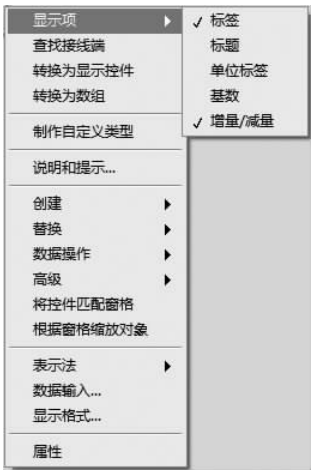


图 3-3　“数值”控件的快捷菜单

图 3-4　“数值”控件的属性对话框

（1）外观。可设置数值控件的标签和标题，并能将其设置为可见或不可见。“启用状态”选项用来设置用户是否对该控件进行操作，其中“启用”表示用户可以操作该控件；“禁用”表示该控件能在前面板显示，但不能对其操作；“禁用并变灰”表示不仅不能操作该控件，而且在前面板上呈灰色。“显示基数”选项用来改变对象的格式，如八进制、十进制、十六进制等。“显示增量/减量按钮”选项用来显示控件的增量/减量按钮，单击按钮可修改控件值。

（2）数据类型。定义数据的表示法，单击“表示法”图标，在弹出的窗口中选择数据类型，取值范围见表 3-2。

（3）数据输入。修改数值控件的默认值，可设置默认的最小值、最大值和增量。“对超出界限的值的响应”选项用来设置当用户的输入值超出数据界限时处理数值的方式。

（4）显示格式。用于改变数值控件的类型和精度。数值对象的类型有浮点、科学计数法、自动格式、SI 符号、二进制、八进制、十进制、十六进制、绝对时间和相对时间等。“精度类型”用来设置显示精度位数或有效数字，当精度类型为精度位数，“位数”可指定小数点后显示的数字位数；如精度类型为有效数字，“位数”可指定显示的有效数字位数。“隐藏无效零”用来删除数据末尾的无效零。

（5）说明信息。用于描述对象的目的并给出使用说明。

（6）数据绑定。用于将前面板对象绑定至共享变量引擎（NI-PSP，订阅协议，由 NI 公司发布）或 DataSocket。

（7）快捷键。用于设置控件的快捷键。

【实训练习】

（1）数值型常量的数据类型定义：在程序框图中放置一个数值常量，取其值为2，并设定其数据类型为双字节整型。

（2）数值型变量的数据类型定义：定义数值输入控件的数据类型为单精度浮点型，最大值为10，最小值为0，默认值为0，并设定增量的大小为0.005，精度位数为3。

实训练习.mp4

3.1.2 布尔型

布尔数据类型比较简单，只有“真”(True)和“假”(False)，或者“1”和“0”两种取值，也叫逻辑型数据类型。在LabVIEW中，布尔型数据在前面板中出现较多，位于“控件”选板上的“新式”→“布尔”、“银色”→“布尔”、“系统”→“布尔”、“经典”→“经典布尔”及Express→“按钮与开关”和“指示灯”子选板中，包括开关按钮、翘板开关、指示灯、摇杆开关、按钮及单选按钮等，如图3-5所示。与数值型的前面板对象类似，这些不同的布尔控件也是外观不同，属性相同，只有True和False两个值。

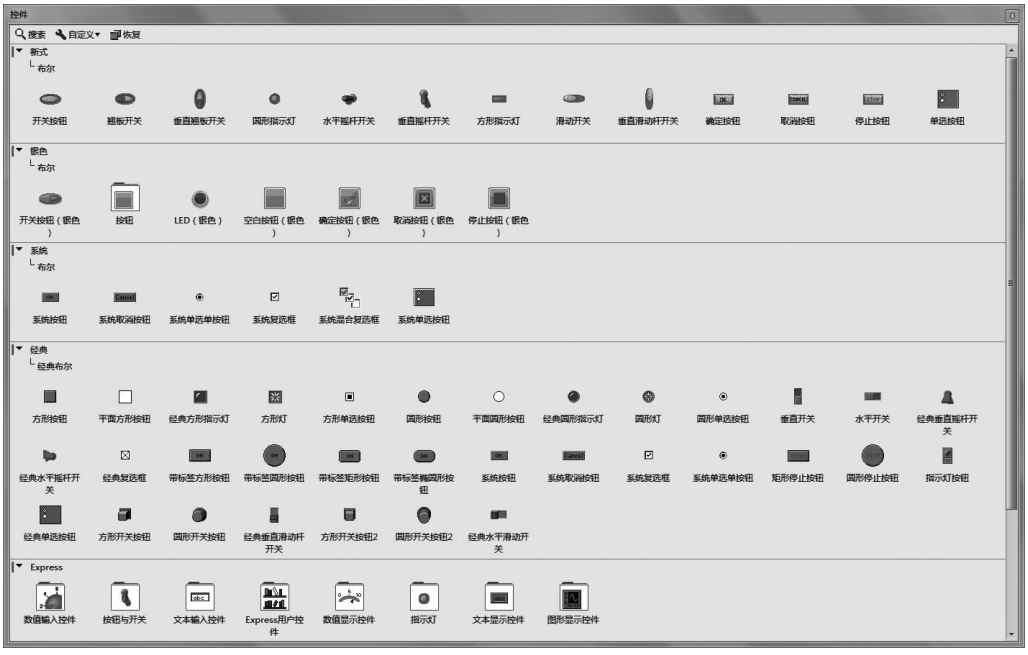


图3-5 “布尔”子选板

布尔常量位于“函数”选板的“编程”→“布尔”子选板中，包括“真常量”和“假常量”，如图3-6所示。

布尔输入控件的一个重要属性是机械动作，正确配置这一属性将有助于更精确地模拟物理仪器上的开关器件。该属性位于布尔输入控件的右键快捷菜单里，如图3-7所示。图标中字母含义为：m(mouse)表示鼠标的动作，v(value)表示控件输出值，RD(read)表示VI读取控件的时刻。

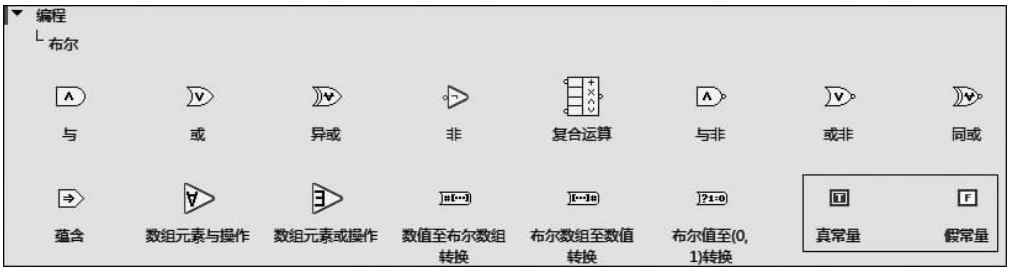


图 3-6　“函数”选板中的“布尔”常量

图 3-7　“布尔”控件的“机械动作”

表 3-3 给出了“布尔”控件的“机械动作”。

表 3-3　“布尔”控件的“机械动作”

图标	名　　称	说　　明
	单击时转换	单击时立即改变控件当前值，且保留新值直至下一次单击控件
	释放时转换	释放鼠标按钮时改变控件当前值，且保留新值直至下一次单击控件
	保持转换直至释放	只在单击鼠标并保持鼠标按钮按下期间改变当前值并保持新值，释放鼠标后将恢复原值
	单击时触发	单击时立即改变控件当前值，且在 VI 读取该控件新值后恢复原值
	释放时触发	释放鼠标时改变控件当前值，且在 VI 读取该控件新值后恢复原值
	保持触发直到释放	只在单击鼠标并保持鼠标按钮按下期间改变当前值并保持新值。释放鼠标按钮且 VI 读取控件值后将恢复原值

在“布尔”控件属性设置对话框的“操作”选项卡中也可以设置“机械动作”，如图 3-8 所示。在“操作”选项卡中，选中的动作为“布尔”控件当前使用的“机械动作”，选中某按钮动作，窗口右侧将给出该动作的详细解释，同时还有所选动作效果预览。

图 3-8 “布尔”控件属性设置对话框

实训练习.mp4

【实训练习】

设置“布尔”控件的“机械动作”：在前面板放置一个水平摇杆开关和一个布尔指示灯，在程序框图中将开关和指示灯圈入 While 循环中，设置机械动作后分别观察运行程序时指示灯做出的反应。

3.1.3 字符串与路径

字符串是 LabVIEW 的一种基本的数据类型。LabVIEW 为用户提供了功能强大的字符串控件和字符串运算功能函数。

路径是一种特殊的字符串，专门用于对文件路径的处理。

在 LabVIEW 中，字符串与路径主要包含在控件选板的“新式”→“字符串与路径”、“银色”→“字符串与路径”、“系统”→“字符串与路径”、“经典”→“经典字符串及路径”及 Express→“文本输入控件”和“文本显示控件”子选板中，如图 3-9 所示，字符串常量位于函数选板的“编程”→“字符串”子选板中，如图 3-10 所示。

从图 3-9 中可以看出，“控件”选板的“字符串与路径”共有 3 种对象：字符串控件（输入/显示）、组合框控件和文件路径控件（输入/显示）。

1. 字符串控件

字符串对象用于处理和显示各种字符串，用数据操作工具或文本编辑工具单击字符串对象的显示区，即可在对象显示区的光标位置进行字符串的输入和修改。

字符串有四种显示模式：正常显示、“\”（反斜杠）代码显示、密码显示、十六进制显示，字符串显示模式可以通过快捷菜单项在这四种模式之间切换。

图 3-9　“控件”选板中的“字符串与路径”控件

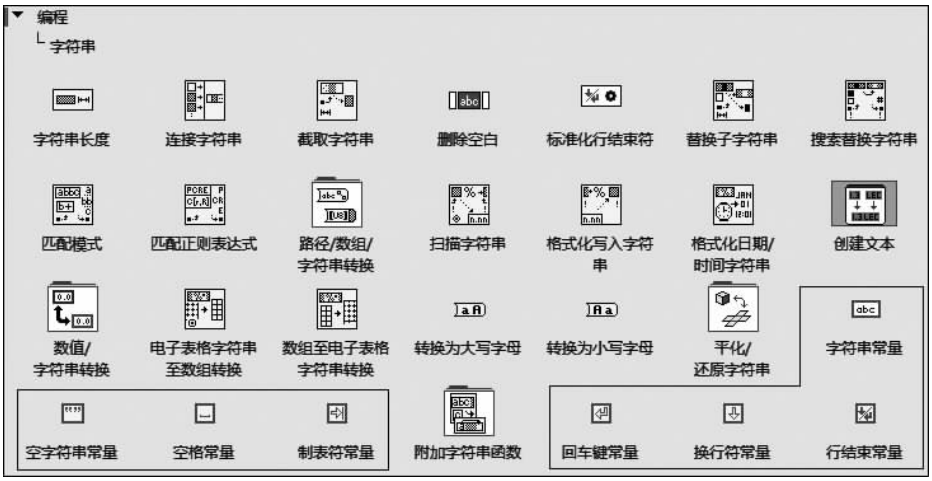


图 3-10　“函数”选板中的“字符串”常量

(1) 正常显示。

在该显示模式下，除了一些不可显示的字符如制表符、Esc 等之外，字符串控件显示键入的所有字符。

(2) “\”代码显示。

在该显示模式，LabVIEW 将反斜杠(\)后紧接的字符视作不可显示字符的代码。该模式适用于调试 VI 及把不可显示字符发送至仪器、串口及其他设备。表 3-4 列出了 LabVIEW 对不同代码的解释。

表 3-4　特殊字符表

代　码	LabVIEW 解释
\00-\FF	8 位字符的十六进制值；必须大写
\b	退格符（ASCII BS，相当于\08）
\f	换页符（ASCII FF，相当于\08）
\n	换行符（ASCII LF，相当于\0A）。格式化写入文件函数自动将此代码转换为独立于平台的行结束字符
\r	回车符（ASCII CR，相当于\0D）
\t	Tab 制表符（ASCII HT，相当于\09）
\s	空格符（相当于\20）
\\	反斜扛（ASCII \，相当于\5C）

反斜杠后的大写字母用于十六进制字符，小写字母用于换行、退格等特殊字符。例如，LabVIEW 将\BFare 视为 hex BF，其后为字符 are。将\bFare 和\bfare 分别视为退格符和 Fare 及退格符和 fare。而在\Bfare 中，\B 不是退格符代码，\Bf 也就不是有效的十六进制代码，在这种情况下，当反斜杠后仅有部分有效十六进制字符时，LabVIEW 将认为反斜杠后带有 0 而将\B 解释为 hex 0B。如果反斜杠后既不是合法的十六进制字符，也不是特殊字符，LabVIEW 将忽略该反斜杠字符。

不论是否选中“\”代码显示，都可通过键盘将表中列出的不可显示字符输入到一个字符串输入控件中。但是，如在显示窗口含有文本的情况下启用反斜杠模式，则 LabVIEW 将重绘显示窗口，显示不可显示字符在反斜杠模式下的表示法及\字符本身。

（3）密码显示。

该模式将使输入字符串控件的每个字符（包括空格）显示为星号（*）。从程序框图中读取字符串数据时，实际上读取的是用户输入的数据。如从控件复制数据，LabVIEW 将只复制 * 字符。

（4）十六进制显示。

该模式下，将显示输入字符的 ASCII 码值，而不是字符本身。调试或与仪器通信时，可使用十六进制显示。

图 3-11 给出了字符串 Display Model of String 及一个回车符在 4 种不同显示模式下的显示结果。

2. 组合框控件

“组合框”控件用来创建一个字符串列表，在前面板上可按次序循环浏览该列表。在字符串列表中，可以预先设定几个预定的字符串，供用户选择，如图 3-12 所示。单击“组合框”右侧的“下拉”按钮，会出现一个下拉列表，列出了预先设定的字符串选项。在“组合框”控件右键快捷菜单中选择“编辑项”，将弹出属性设置对话框并打开“编辑项”选项卡。在该选项卡中，可以编辑、预设组合框对象中可选择的字符串条目，如图 3-13 所示。

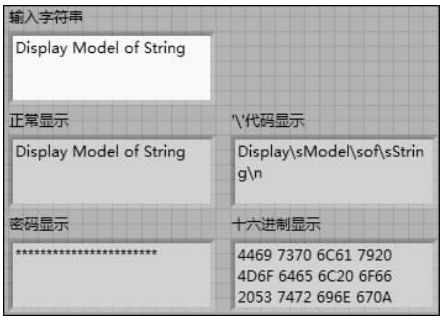


图 3-11　同一字符串在 4 种显示模式下的显示结果

图 3-12　“组合框”控件

图 3-13　“组合框”控件“编辑项”选项卡

在编辑区域，“项”为在组合框中显示的字符串，“值”为组合框实际存储的值。当“值与项值匹配”复选框被选中时，“值”中的字符串选项与“项”中的内容保持一致。另外，“允许在运行时有未定义的值”复选框被选中时，允许用户在为控件定义的值列表中输入不存在的值。如需用户在为控件定义的值列表中选择，可取消勾选该复选框。

3. 文件路径控件

文件路径对象也是一种特殊的字符串对象，专门用于处理文件的路径，文件路径控件用于输入或返回文件或目录的地址，可与文件 I/O 节点配合使用，如图 3-14 如示。用户可以直接在“文件路径输入控件”中输入文件的路径，也可以通过单击浏览按钮 打开一个 Windows 标准文件对话框，在对话框中查找需要的文件。

图 3-14　文件路径控件

路径通常分为以下 3 种类型。

（1）非法路径。

如函数未成功返回路径，该函数将在显示控件中返回一个非法路径值，非法路径值可作为一个路径控件的默认值检测用户何时未提供有效路径，并显示一个带有选择路径选项的文件对话框。要显示文件对话框，可使用“文件对话框”Express VI。

（2）空路径。

空路径可用于提示用户指定一个路径。将一个空路径与文件 I/O 函数相连时，空路径将指向映射到计算机的驱动器列表。

（3）相对路径和绝对路径。

相对路径是文件或目录在文件系统中相对于任意位置的地址。绝对路径描述的是从文件系统的目录开始的文件或目录地址。使用相对路径可避免在另一台计算机上创建应用程序或运行 VI 时重新指定路径。

3.2 数据运算

LabVIEW 提供了丰富的数据运算功能，除了基本的数据运算外，还有许多功能强大的函数节点。与文本语言编程不同的是，LabVIEW 的运算是按照数据流的方向顺序执行，不具有优先级和结合性的概念。

3.2.1 数值运算

基本数值运算节点主要实现加、减、乘、除等基本数值运算。在 LabVIEW 中，数值运算符包含在程序框图的“编程”→“数值”子选板中，如图 3-15 所示。“数值”子选板中除一些实现基本数值运算的函数节点外，还包括数值常量、数学与科学常量节点、转换节点、数据操作节点、复数节点等几个子选板。

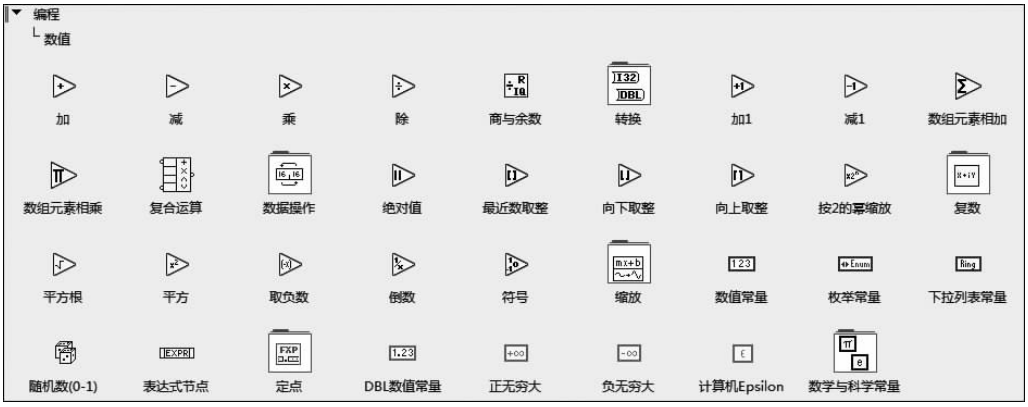


图 3-15 “数值”子选板

数值运算函数的输入都是数值型数据。除了函数中明确指出的一些特例以外，输出数据的表示法默认与输入数据的表示法相同，如果输入数据包含多种不同的数值类型，那么输出数据的类型与能表示较大数值的输入数据类型相同。例如，将一个 8 位整数和一

个 16 位整数相加，默认的输出类型为 16 位整数。如对函数的输出进行配置，则指定的设置将覆盖原有的默认设置。

数值运算功能不仅支持单一的数值量输入，还支持处理不同类型的复合型数值量，例如由数值型数据组成的数组、簇或簇数组等。下面对几个较特殊的函数进行说明。

(1) 加。加函数能分别计算数值、数组、矩阵和簇的和；也能计算数值与数组、数值与簇之和，此时数值与数组或簇的每个元素相加。其他节点与此类似。

(2) 乘。不能进行矩阵相乘。若输入的是两个矩阵，该节点仅将两个矩阵第一行的第一个元素相乘。

(3) 复合运算。通过快捷菜单的“更改模式”命令，可任选加、减、乘、与、或、异或等 5 种模式之一进行运算。用对象操作工具拖动节点图标下边缘(或上边缘)上的尺寸控制点，或在输入端口的右键快捷菜单中选择“增加输入”命令，可添加输入端口。在输入端口或输出端口的右键快捷菜单中选择“反转”命令，可使该端口对数据进行“非”操作。

(4) 最近数取整。输入值向最近的整数取整。当输入值为两个整数的中间值时，该函数返回最近的偶数。例如，数字为 1.5 或 2.5，最接近的整数值为 2。向下取整、向上取整分别返回小于等于 x 的最大整数和大于等于 x 的最大整数。

表 3-5 列出了“数值”子选板中 6 个子选板的功能说明。

表 3-5　“数值”子选板中 6 个子选板的功能说明

图标	子选板	功 能 说 明
	转换	转换 VI 和函数用于数据类型的转换，包括转换为长整型、转换为单精度浮点、转换为单精度复数、单位转换、布尔值至(0,1)转换、RGB 至颜色转换等实现数据类型转换的函数节点
	数据操作	数据操作函数用于改变 LabVIEW 使用的数据类型，包括强制类型转换、转化至字符串、从字符串还原等函数节点
	复数	复数函数用于根据两个直角坐标或极坐标的值创建复数或将复数分为直角坐标或极坐标的两个分量，包括复共轭、极坐标至复数转换、复数至极坐标转换等函数节点
	缩放	缩放 VI 可将电压读数转换为温度或其他应变单位，包括转换 RTD 读数、转换热电偶读数、转换热敏电阻读数、转换应变计读数等 4 个函数节点
	定点	定点函数可对定点数字的溢出状态进行操作，包括定点溢出状态、定点转换为整型、清除定点溢出状态、删除定点溢出状态、整型转换为定点 5 个函数节点
	数学与科学常量	数学与科学常量用于创建 LabVIEW 应用程序，包含了科学数据委员会(CODATA)制定的一些常量，有些常量带有单位

【实训练习】

编写程序计算 $y=ax^2+bx-2$，输入变量为 a、b 和 x。

实训练习.mp4

3.2.2　比较运算

比较运算也称为关系运算，比较运算函数节点包含在“函数”→“编程”→“比较”子选板中，如图 3-16 所示。

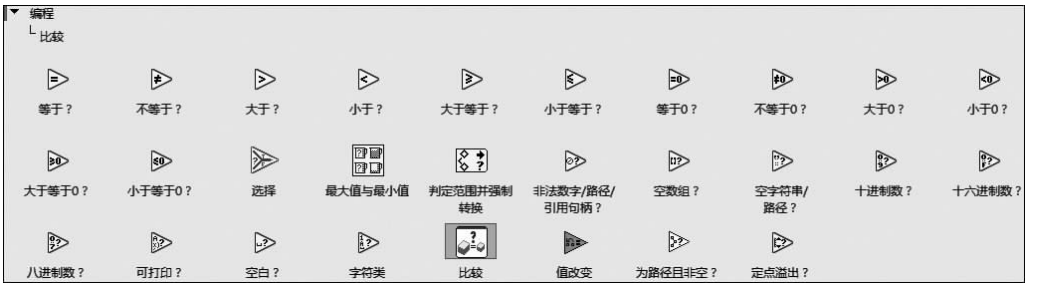


图 3-16 “比较”子选板

在 LabVIEW 中可以进行数值比较、布尔值比较、字符串比较、数组比较和簇比较、值改变。不同数据类型的数据在进行比较时适用的规则不同，下面对这些规则进行简单介绍。

（1）数值比较。

数值比较是相同数据类型的比较。数据类型不同时，比较函数的输入端能够自动进行强制性数据类型转换，然后再进行比较。对于带有非法数值（NaN）的一个或两个输入，其比较将返回不相等的结果。

（2）布尔值比较。

在布尔比较中，布尔值 True 值比 False 值大，实际上就是 0 和 1 两个值的比较。

（3）字符串比较。

字符串的比较是依据 ASCII 字符码的值对字符串进行比较。在比较时从字符串的第 0 个元素开始，逐个比较，直到两个字符不相等或者直至一个字符串的末尾为止。

（4）数组比较和簇比较。

与字符串的比较类似，从数组或簇的第 0 个元素开始比较，直到有不相等的元素为止。某些比较函数节点有两种比较数组或簇的模式，在“比较集合”模式下，比较两个数组或簇时，函数返回的是标量；在“比较元素”模式下（默认），函数返回值为所有比较结果的相应布尔值构成的数组或簇。

比较多维数组时，每个连接到函数的数组必须有相同的维数；进行簇的比较时，簇中的元素个数、元素的数据类型及顺序必须相同。

（5）值改变。

如首次调用该 VI，或输入值自上一次调用 VI 后发生改变，则返回 True。

实训练习.mp4

【实训练习】

设计一个 VI，用于比较两个输入数值的大小，并输出较大值的平方值。

3.2.3 布尔运算

布尔运算又称逻辑运算，传统编程语言使用逻辑运算符将关系表达式或逻辑量连接

起来形成逻辑表达式。逻辑运算包括与、或、非等。逻辑运算函数节点包含在“函数”→“编程”→“布尔”子选板中，LabVIEW 中逻辑运算函数节点的图标与数字电路中的逻辑运算符的图标类似，如图 3-17 所示。

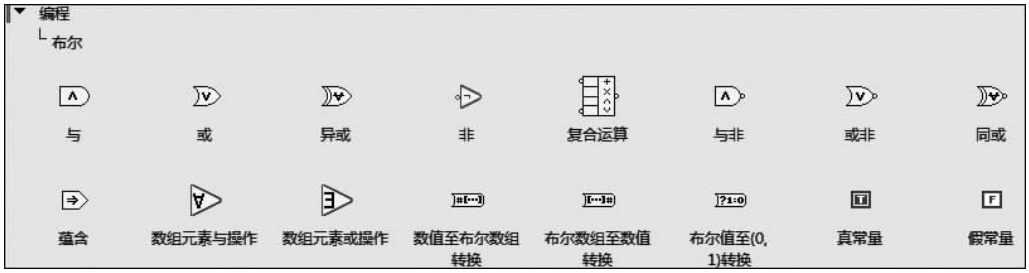


图 3-17　“布尔”子选板

【实训练习】

（1）编写一个用于判断数值大小的程序，当两个值都大于 100 时，绿指示灯亮；当有一个数值大于 100 时，红指示灯亮。

（2）实现两个 8 位无符号整数的布尔运算及布尔变量之间的布尔运算。

实训练习 1.mp4　　实训练习 2.mp4

3.2.4　字符串运算

虚拟仪器控制软件经常需要实现与各种仪器进行通信，处理各种不同的文本命令，这些命令通常由字符串组成，对字符串进行合成、分解、变换是软件开发人员经常遇到的问题。因此，LabVIEW 为用户提供了丰富的字符串运算函数，这些字符串函数提供包括合并两个或两个以上字符串、从字符串中提取子字符串、将数据转换为字符串、将字符串格式化用于文字处理或电子表格等功能。

字符串运算函数节点包含在“函数”→“编程”→“字符串”子选板中，如图 3-18 所示。其中包括基本字符串操作函数、数值/字符串转换、路径/数组/字符串转换和附加字符串函数。

下面对一些常用的字符串函数的使用方法进行简要说明。

1. 字符串长度

字符串长度函数的图标如图 3-19 所示，其功能是用于返回字符串、数组字符串、簇字符串所包含的字符个数，以字节为单位。因此，在字符串中，一个汉字的长度是“2”。图 3-20 所示为字符串长度函数的使用。

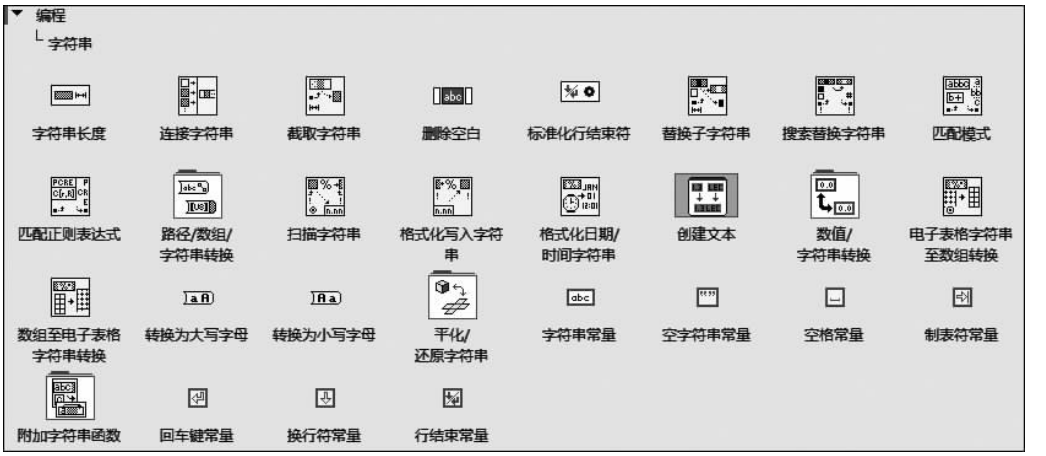


图 3-18 “字符串”子选板

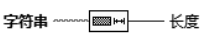


图 3-19 字符串长度函数的图标

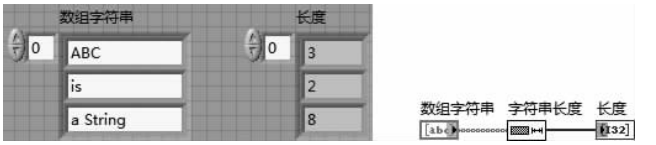


图 3-20 字符串长度函数的使用

2. 连接字符串

连接字符串函数的图标如图 3-21 所示，其功能是将两个或多个字符串连成一个新的输出字符串，输出字符串包含所有连接的输入字符串，顺序与连线至节点的顺序（从上到下）一致，如图 3-22 所示。对于数组输入，该函数连接数组中的每个元素。

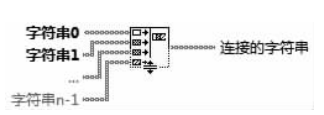


图 3-21 连接字符串函数的图标

图 3-22 连接字符串函数的使用

右击“函数”，在快捷菜单中选择添加输入，或向外拖动函数上、下边框，均可向函数增加输入端。

3. 截取字符串

截取字符串函数的图标如图 3-23 所示，其功能是返回输入字符串的子字符串，从偏移量位置开始，包含长度指定的字符，如图 3-24 所示。

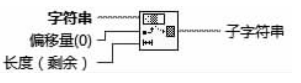


图 3-23 截取字符串函数的图标

图 3-24 截取字符串函数的使用

4. 替换子字符串

替换子字符串函数的图标如图 3-25 所示，其功能是插入、删除或替换子字符串，位置由偏移量指定。

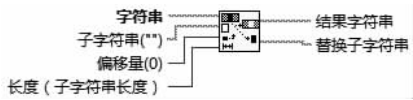


图 3-25　替换子字符串函数的图标

函数从偏移量位置开始在字符串中删除长度指定的字符，并用子字符串替换删除的部分，如图 3-26 所示。如长度为 0，替换子字符串函数在偏移量位置插入子字符串。如子字符串为空，该函数在偏移量位置删除长度指定的字符，如图 3-27 所示。

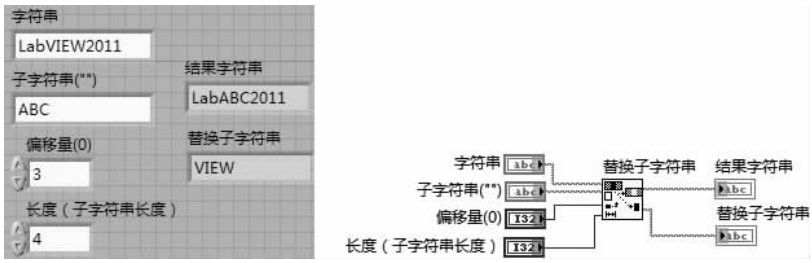


图 3-26　将指定的字符串替换

5. 搜索替换字符串

搜索替换字符串函数(图 3-28)与替换子字符串函数的不同之处在于，它不是按照位置和长度替换字符串，而是查找与“搜索字符串”一致的字符串，用“替换字符串”替换。如图 3-29 所示，如果“替换字符串”不连接，就删除“搜索字符串”输入的字符串。

图 3-27　将指定的字符串删除

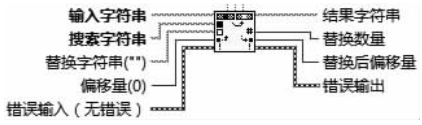


图 3-28　搜索替换字符串函数的图标

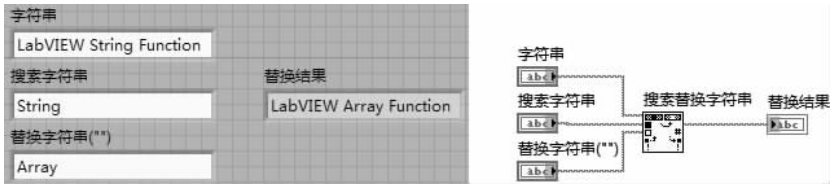


图 3-29　搜索替换字符串函数的使用方法

6. 格式化日期/时间字符串

格式化日期/时间字符串函数的图标如图 3-30 所示，通过“时间格式字符串”指定输出字符串的格式，按照该格式使时间标识的值或数值显示为时间。时间格式代码为：%H(小时，24 小时制)，%I(小时，12 小时制)，%p(AM/PM)，%M(分钟)，%s(秒)，%d(日)，%m(月份)，%b(月名缩写)，%y(两位年份)，%Y(四位年份)，%a(星期名缩写)。输入时间格式字符串时，如果插入其他字符，则将其原样输出，如图 3-31 所示为该函数的使用示例。

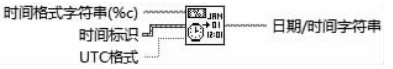


图 3-30　格式化日期/时间字符串函数的图标

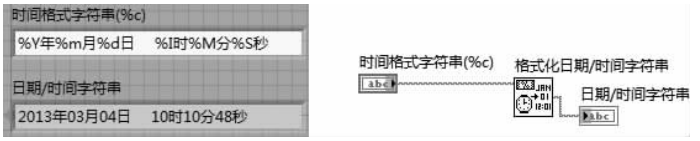


图 3-31　格式化日期/时间字符串函数的使用方法

7. 格式化写入字符串

格式化写入字符串函数的图标如图 3-32 所示，其功能是将字符串路径、枚举型、时间标识、布尔或任意数值数据类型转化为文本输出。“格式字符串”指定函数转换输入参数为结果字符串的方法；“初始字符串”指定可通过扩展参数组成结果字符串的基本字符串。应用示例如图 3-33 所示。

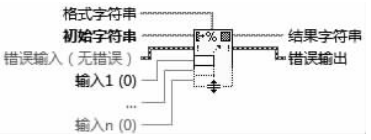


图 3-32　格式化写入字符串函数的图标

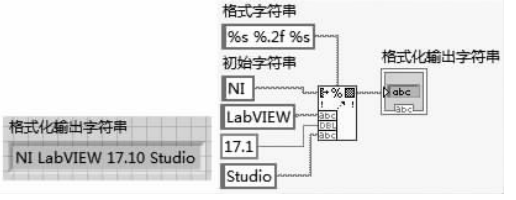


图 3-33　格式化写入字符串函数的使用方法

8. 数值至小数字符串转换

该函数位于“函数”→“编程”→“字符串”→“数值/字符串转换”子选板中，如图 3-34 所示，其功能是使数字转换为小数格式的浮点型字符串，其中输入数据端口“数字”表示待转换的数据，“精度”输入数据端口表示转换后数据的精度，即小数点后保留几位，在图 3-35 中转换前的数字为“3.141592”，按照 5 位精度转换后的字符串为“3.14159”。

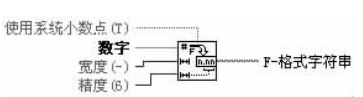


图 3-34　数值至小数字符串转换函数的图标

图 3-35　数值至小数字符串转换函数的使用方法

【实训练习】

输入两个字符串：LabVIEW 2014 和 Studio，将它们拼接成一个字符串后测量该字符串的长度；将结果字符串的 2014 替换成 2017，并删除字符 Studio。

实训练习.mp4

3.3　数组

LabVIEW 主要的数据类型除了数值型、字符型和布尔型等基本数据类型外，还有结构类型（包括一个以上的元素）数据，如数组、簇和 LabVIEW 特有的动态数据类型等复杂数据类型。

3.3.1　数组数据的组成

数组是相同类型元素的集合，由元素和维度组成。元素是组成数组的数据，维度是数组的长度、高度或深度。数组可以是一维或多维的，在内存允许的情况下每一维度可有多达 $2^{31}-1$ 个元素。在 LabVIEW 中，数组元素可以是数值型、布尔型、字符串型及其他数据类型，但是不能将数组、图表或波形数据等作为数组元素。

数组中的每个元素都有其唯一的索引数值，对每个数组元素的访问都是通过数组索引进行访问的。索引的范围是 0 到 $n-1$，其中 n 是数组中元素的个数。图 3-36 所示是一个由数值型数据构成的一维数组。注意，第一个元素的索引号为 0，第二个是 1，依此类推。虽然数组的元素可以是数值型、布尔型、字符串型等数据类型，但其所有元素的数据类型必须一致。数组由 3 部分组成：数据、数据索引和数据类型（隐含在数据中）。通过索引可以很容易地访问数据中的任何一个元素。

索引	0	1	2	3	4	5	6	7	8	9
10个元素的数组	1.2	3.2	8.2	5.2	8.0	1.2	5.1	6.0	2.1	1.7

图 3-36　一维数据结构示意图

在创建一个数组时，首先要建立一个数据框架，然后在这个框架中置入数组元素（数值、字符串等）。通过将一个数组的框架和数据对象结合起来，可以创建一个数组控件。

3.3.2　数组的创建

在 LabVIEW 中，可以用多种方法来创建数组数据。其中常用的有以下 3 种方式：第一，在前面板上创建数组数据；第二，在程序框图上创建数组数据；第三，用函数、VIs 以及 Express VIs 动态生成数组数据。

1. 前面板数组对象的创建

通过以下两个步骤可以完成一个简单前面板数组对象的创建。

（1）创建一个数组框架。

要创建一个数组控件，首先必须在前面板放置一个数组框架。数组框架位于“控件”→“新式”→“数组、矩阵与簇”子选板、“控件”→“银色”→“数组、矩阵与簇”子选板和

“控件”→“经典”→“经典矩阵与簇”子选板，如图 3-37 所示。

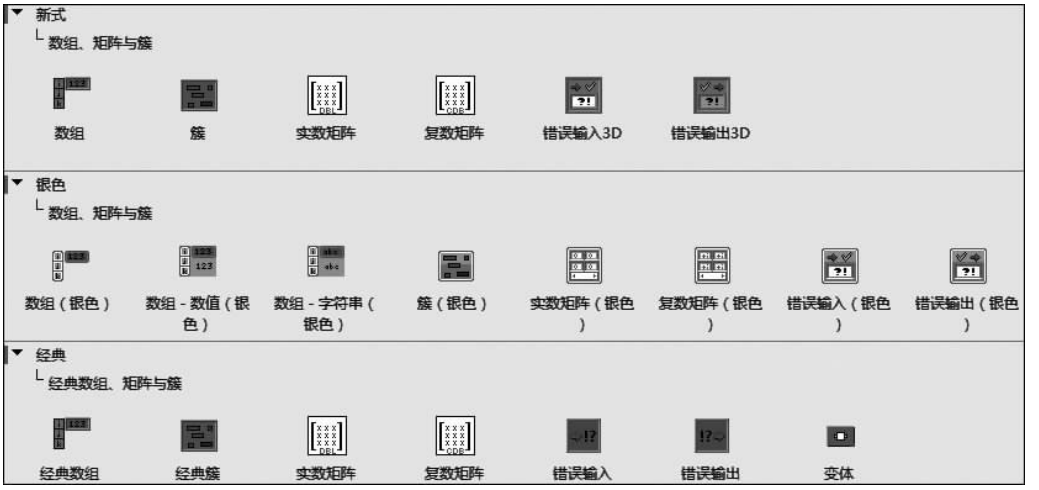


图 3-37 “数组、矩阵与簇”子选板

单击“数组”控件后移动鼠标到前面板窗口，在前面板上再次单击，则在前面板上创建了一个数组控件。此时创建的仅仅是一个数组的框架，不包含任何内容，对应在程序框图中的接线端口为黑色边框的图标，如图 3-38 所示。

（2）将一个数据对象或元素放入该数组框架中。

当创建好一个数组框架后，可以直接从“控件”选板中选择对象放进数组框架内，也可以把前面板上已有的对象拖曳到框架中，即完成数组的创建。这个数组的数据类型以及它是输入控件还是显示器完全取决于放入的对象。如图 3-39 所示为数值型数组控件（数组的属性为输入），数组在程序框图中相应的端口也变为相应颜色和数据类型的图标了。

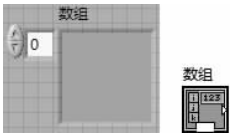


图 3-38 数组框架

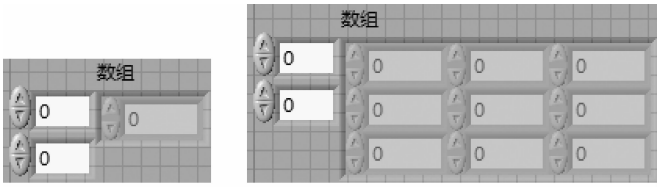
图 3-39 创建数值型数组

数组在创建之初都是一维数组，如果需要创建一个多维数组，把定位工具放在数组索引框任意一角轻微移动，向上或向下拖动鼠标增加索引框数量就可增加数组的维数，或者在索引框上弹出的快捷菜单中选择添加维度命令，如图 3-40(a)所示。

两个索引框中上一个是行索引，下一个是列索引。光标放在数组索引框左侧时不仅可以上下拖动增加索引框数量，还可以向左拖动扩大索引框面积。刚刚创建的数组只显示一个成员，如果需要显示更多的数组成员，可以把定位工具放在数组数据显示区任意一角，当光标形状变成网状折角时，向任意方向拖动就可以显示更多的数据成员，如图 3-40(b)所示。数组索引框显示的是左上角的数组成员的索引号。

2. 在程序框图中创建数组常量

数组常量只能在程序框图中出现，其创建方法与前面板创建数组控件的方法类似。



(a) 增加数组维数　　(b) 增加显示成员数量

图 3-40　增加数组成员

在“函数”→“编程”→“数组”子选板中选中“数组常量”并放置到程序框图中，即创建一个数组常量框架。将“常量”(如数值常量、布尔常量、字符串常量等)拖入数组常量框架中，即完成一个数组常量的创建。

利用数组常量的索引区和边框上快捷菜单“转换为输入控件”和“转换为显示控件”选项可分别把数组常量变为前面板上的数组输入控件和显示控件。

3. 数组成员赋值

用上述方法创建的数组是空的，从外观上看，数组成员都显示为灰色，根据需要用操作值工具为数组成员逐个赋值。若跳过前面的成员为后面的成员赋值，则前面成员数据类型自动赋一个空值，例如，0(数值)、F(布尔值)或空字符串。数组赋值后，在赋值范围外的成员显示仍然是灰色的。

其他创建数组的方法：用数组函数创建数组，用某些 VI 的输出参数创建数组，用程序结构创建数组。

3.3.3　数组数据的使用

LabVIEW 给出了大量的数组处理函数，可以实现数组的各种操作。数组函数位于“函数”→“编程”→“数组”子选板中，如图 3-41 所示。

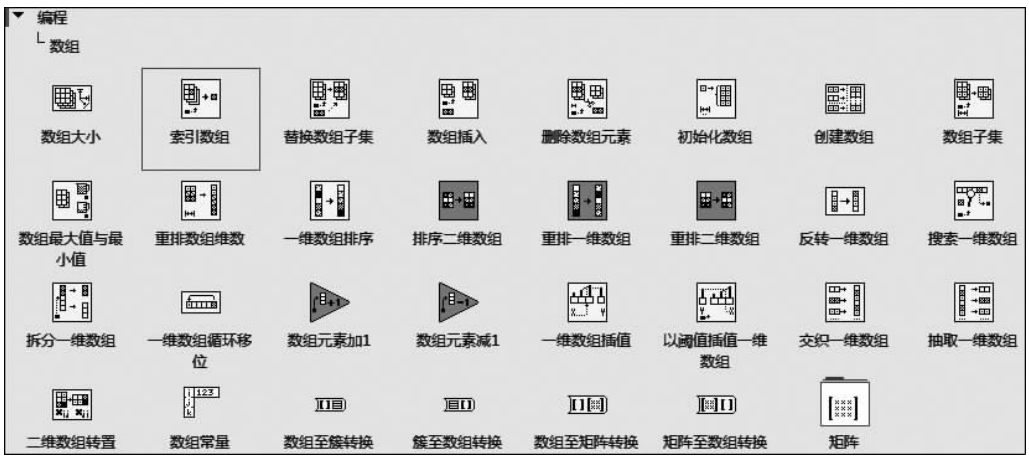


图 3-41　“数组”子选板