

# 虚拟仪器技术

王峰

**Email:** wangfeng@nju.edu.cn

现代工程与应用科学学院

## 第2章 LabVIEW程序对象的基本操作

**VI**是**LabVIEW**编程中的基础程序单元。

本章主要内容：

- 1.在**前面板**中创建和编辑控件对象
- 2.控件对象在**程序框图**中的连接方法
- 3.如何运行和调试**VI**程序。

一个图标，两个界面，三个选板

## ■ 2.1 VI的创建

### 2.1.1 VI的创建方法

在**LabVIEW**中新建一个**VI**，有以下几种方法。

(1) 在前面板的“文件”菜单中选择“新建**VI**”。

## ■ 2.1 VI的创建

- (2) 在如图1-7所示的启动方式选择对话框中的新建文件窗口直接点击**VI**建立一个新的**VI**。
- (3) 在启动方式选择对话框中单击“项目”，创建一个项目后，在“我的电脑”上单击鼠标右键，从弹出的快捷菜单中选择“新建→**VI**”，即可创建新的**VI**，如图2-1所示。

## ■ 2.1 VI的创建

(4) 从图1-7所示的启动方式选择对话框中选择“基于模板的VI”项，在如图2-3所示的模板选择对话框中选择“基于模板的VI”，模板针对不同的应用需求设计了不同的程序框架，可以根据需要选择不同的模板并在模板中添加程序，这样可以大大提高编写程序的效率。

## ■ 2.1 VI的创建

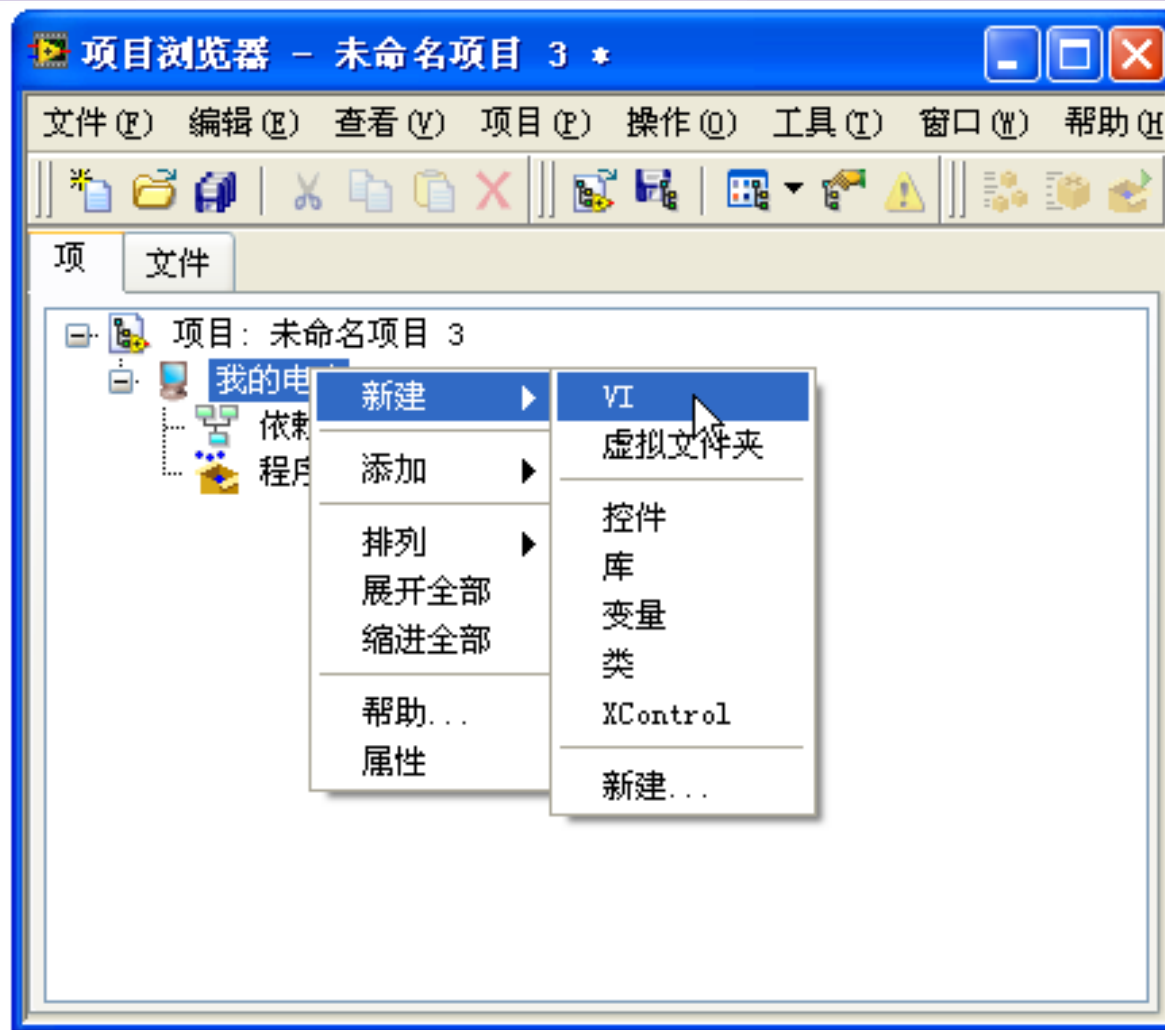


图2-1 新建VI

## ■ 2.1 VI的创建

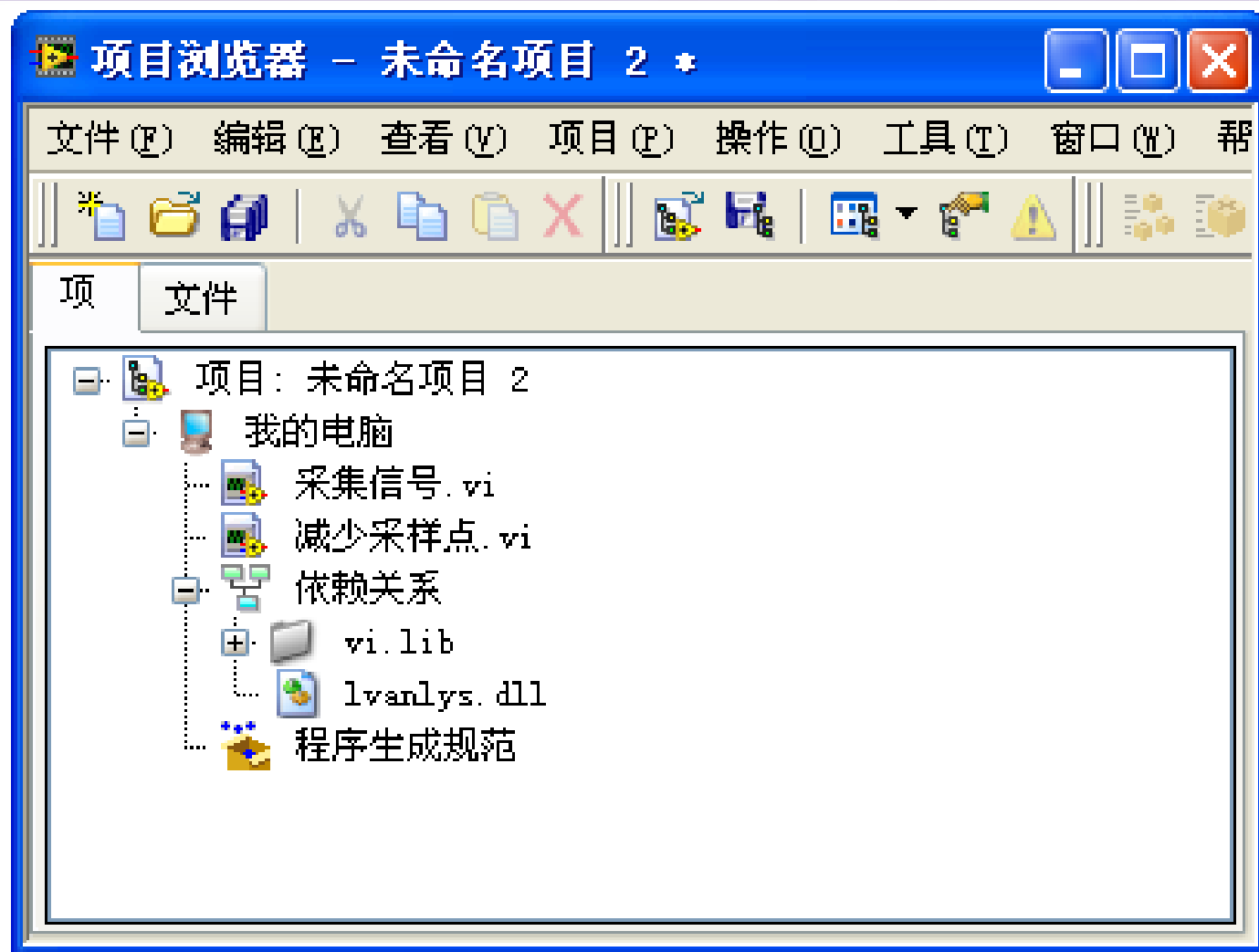


图2-2 项目浏览器

## 2.1 VI的创建

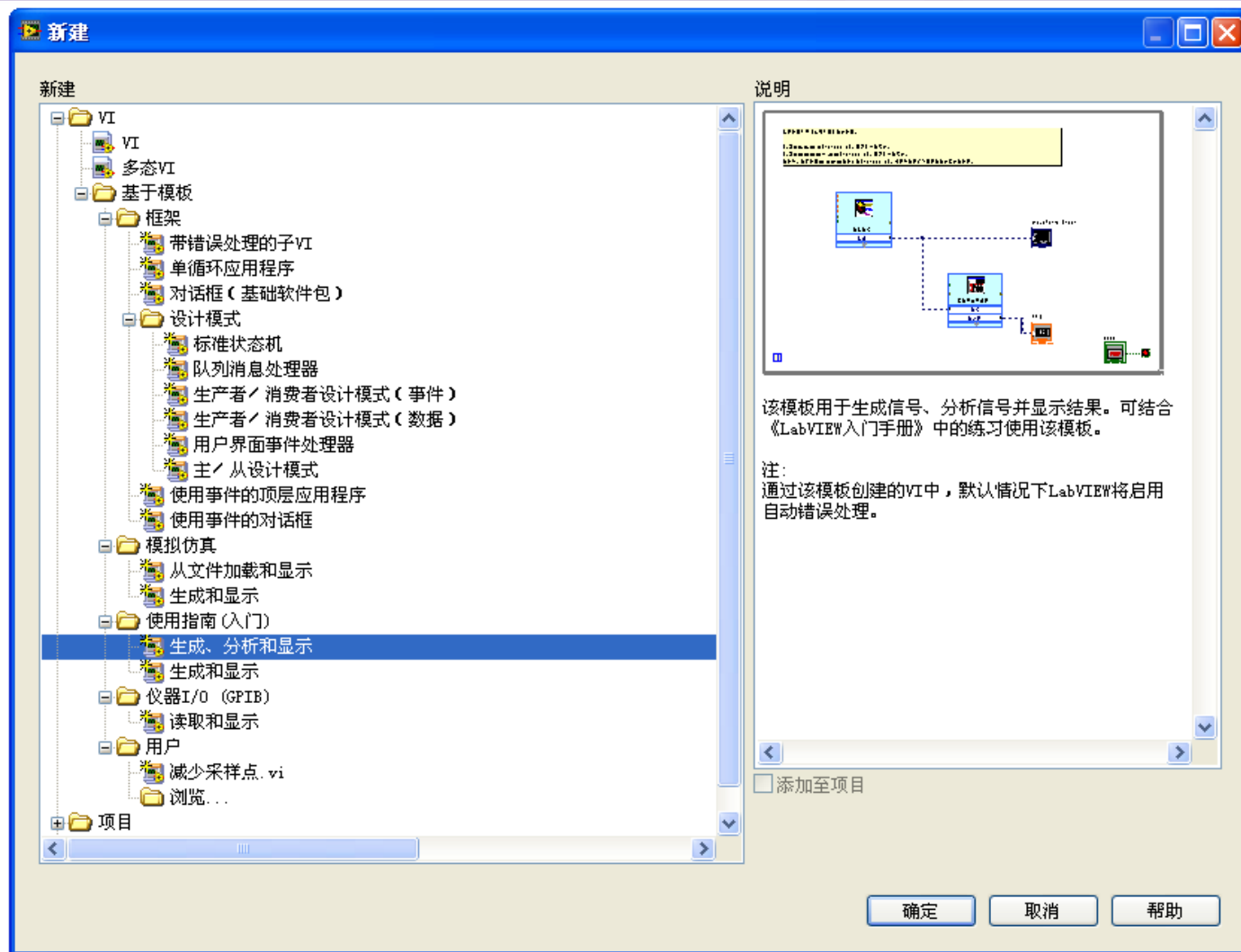


图2-3 模板选择对话框



## 2.1.2 VI示例的创建步骤

**【例2-1】** 创建一个简单的VI，该VI将完成下列功能：

- (1) 将两个输入的数字相减并显示结果；
- (2) 将同样的两个输入数字相乘并显示结果；
- (3) 比较两个输入数字，如果数字相等，则**LED**指示灯变亮。

## 2.1.2 VI示例的创建步骤

创建此**VI**的步骤如下。

- (1) 建立新的**VI**。
- (2) 前面板创建数字输入控件和指示器。
- (3) 创建布尔型**LED**。

## 2.1.2 VI示例的创建步骤

- (4) 从“窗口”下拉菜单中选择“显示程序框图”以切换到程序框图窗口  
(**CTRL+E**)。
- (5) 在程序框图上放置“减”和“乘”的函数。
- (6) 从函数选板的比较子选板中选择“等于？”并将其拖放到程序框图上。
- (7) 连线。

## 2.1.2 VI示例的创建步骤

- (8) 通过选择“窗口”下拉菜单的“显示前面板”或单击前面板窗口的任何位置切换回前面板窗口。
- (9) 保存此VI。
- (10) 输入数据并运行VI。
- (11) 完成实验，关闭该VI。

## 2.1.2 VI示例的创建步骤

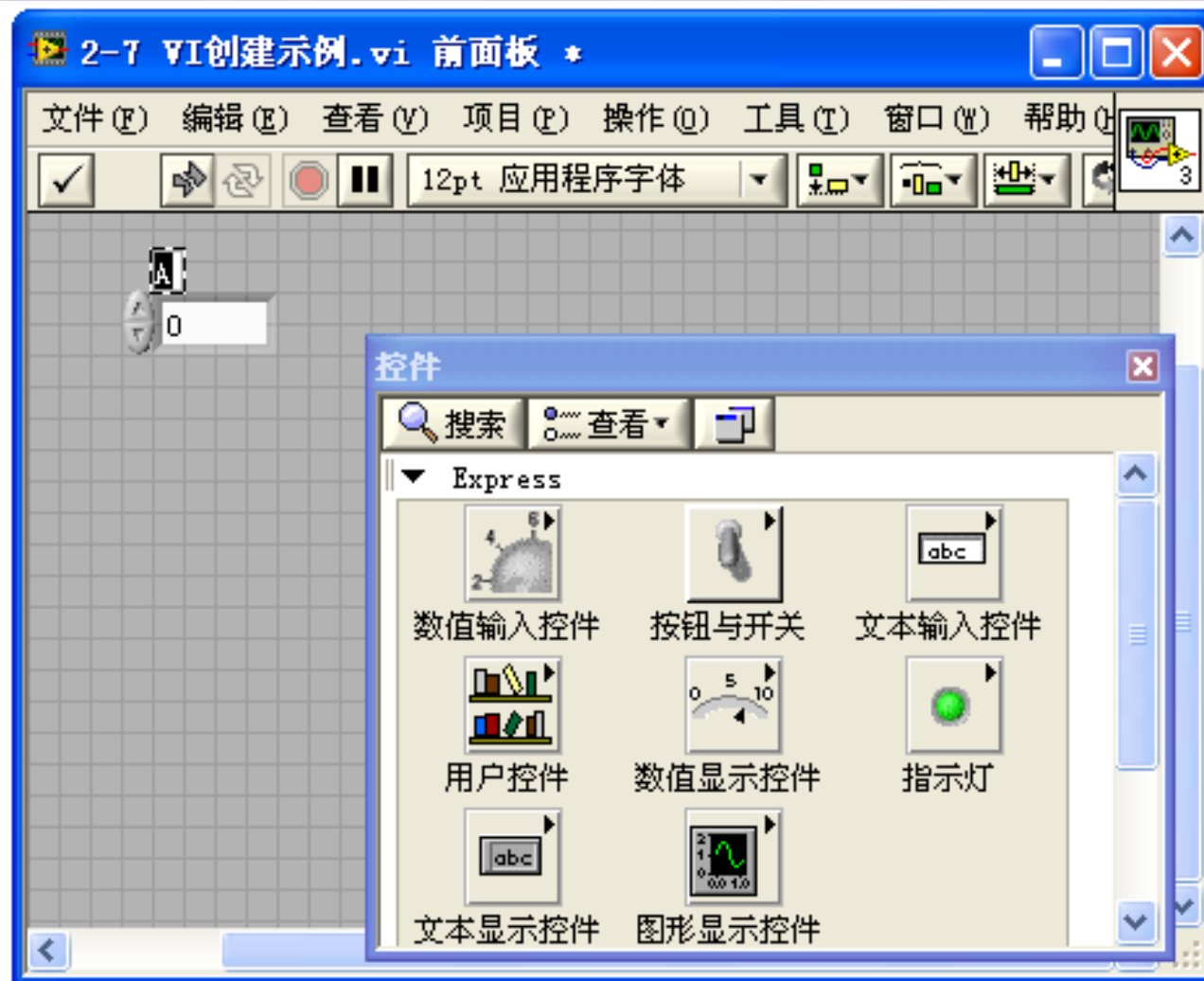


图2-4 在前面板上放置控件

## 2.1.2 VI示例的创建步骤



图2-5 在程序框图上放置函数

## 2.1.2 VI示例的创建步骤

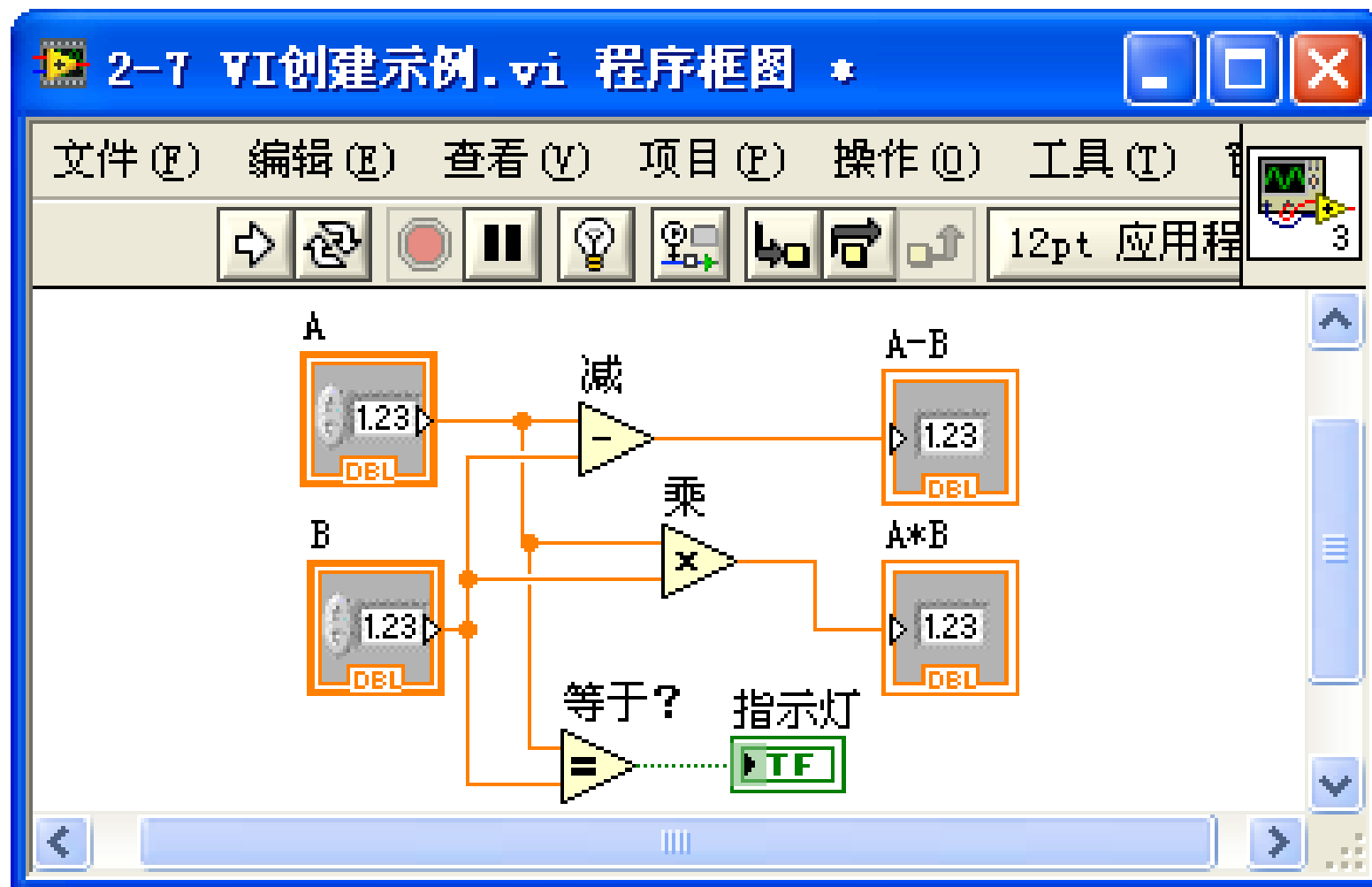


图2-6 连线后的框图

## 2.1.2 VI示例的创建步骤



图2-7 VI运行结果



一个图标，两个界面，三个选板

数据流运行过程，

实例演示



## ■ 2.2 前面板的编辑

**LabVIEW**中的前面板是图形化的**人机界面( 虚拟面板)**，用户利用前面板提供的控件选板可以**设置基本的输入数据和显示输出数据**。同样，也可以在基本控件的基础上**创建自定义的控件**模拟真实仪器或生产过程。

## 2.2.1 输入控件和显示控件

利用**输入控件**可以实现相应数据的输入，如**数值量、布尔量、字符串、文件路径**等。主要的输入控件有数值输入控件、文本输入控件、旋钮与开关控件等，如图2-8所示。

**显示控件**主要是用来实现数据的**输出-显示**功能。



**输入控件和显示控件可以相互转换。**

## 2.2.1 输入控件和显示控件

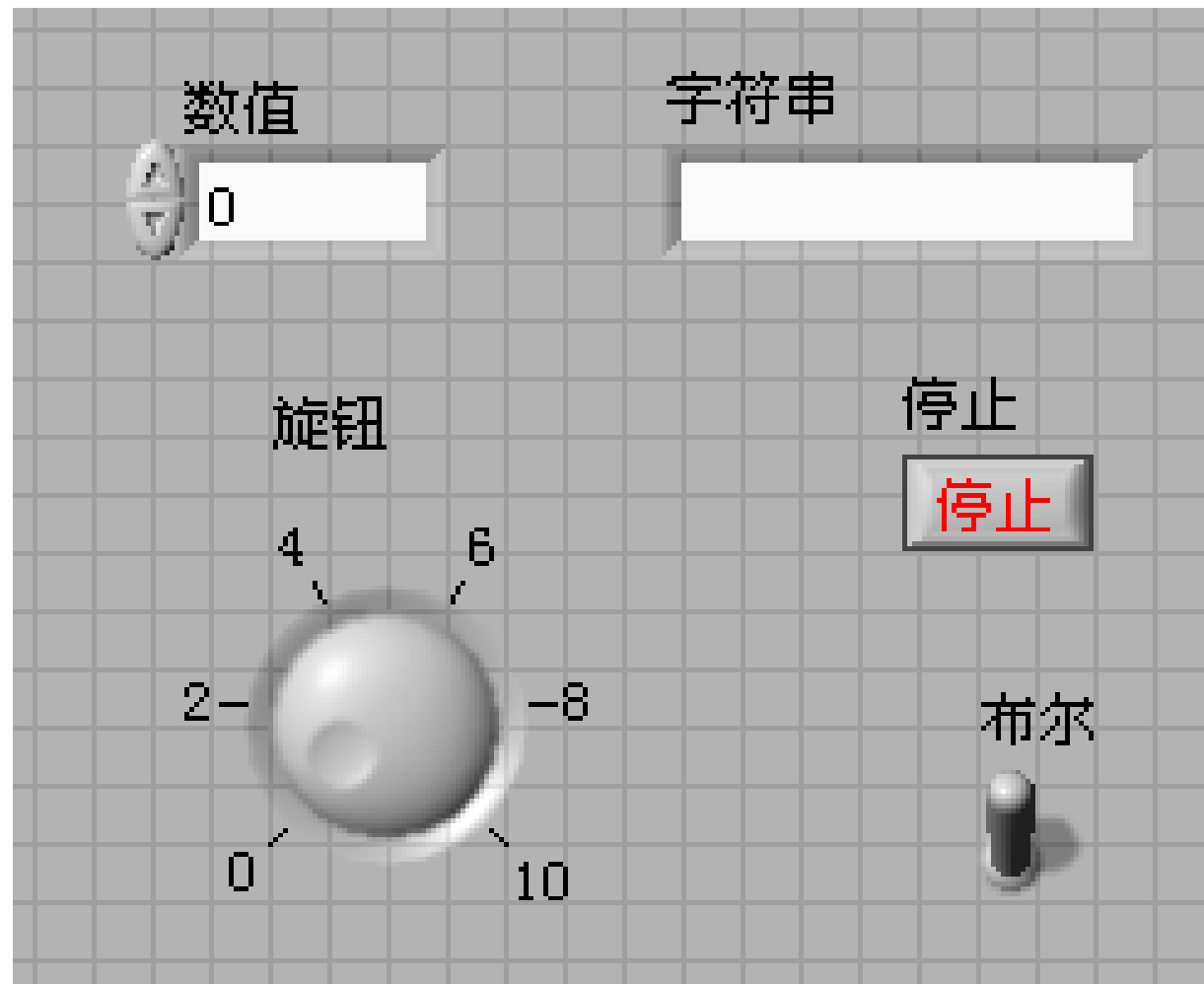


图2-8 输入控件示意图

## 2.2.1 输入控件和显示控件

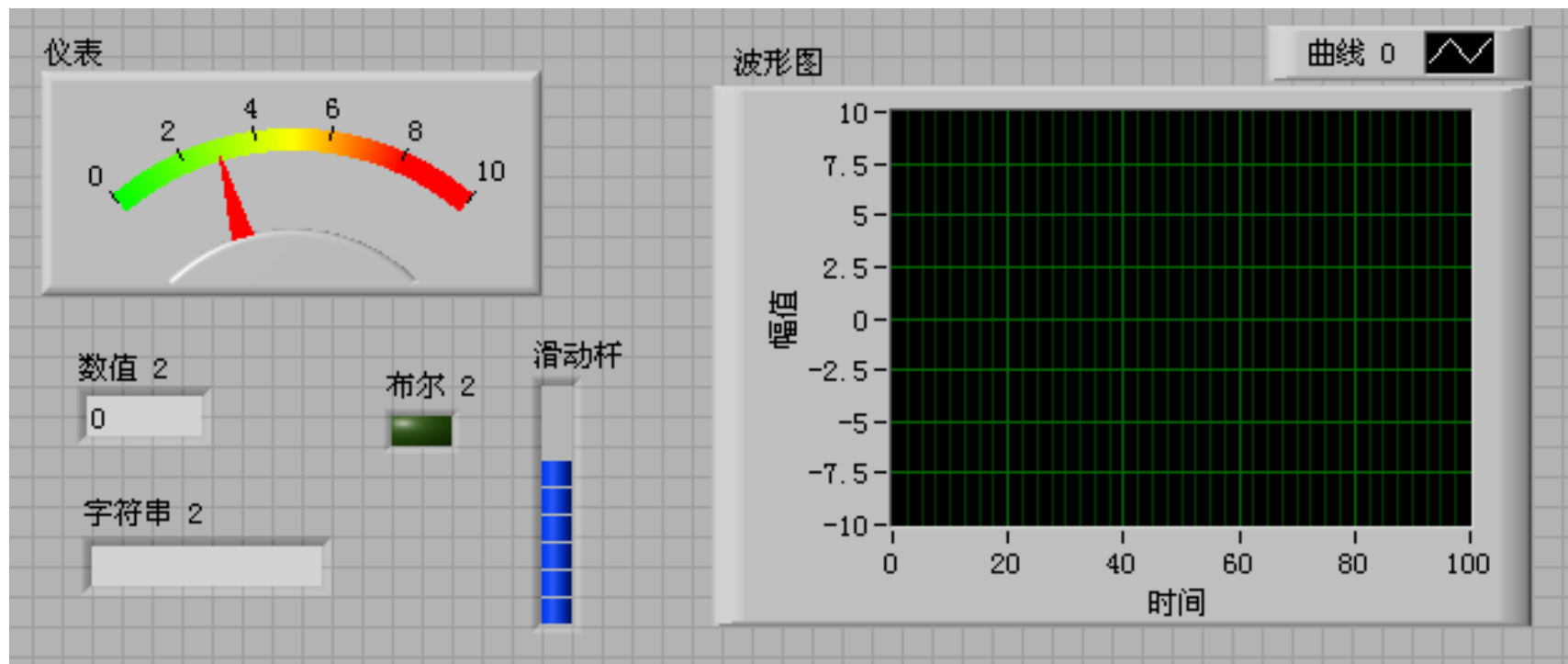


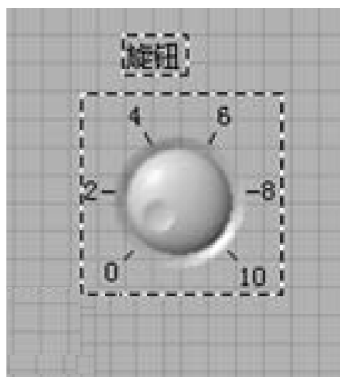
图2-9 显示控件示意图

## 2.2.2 对象的放置与选择

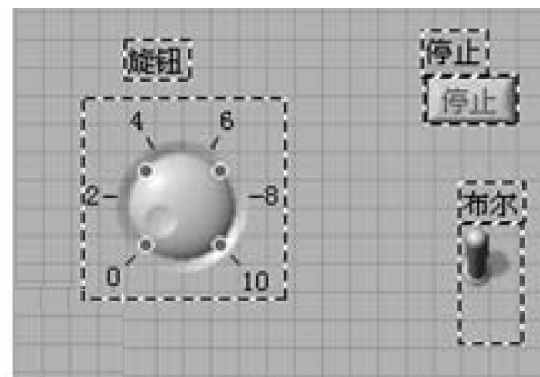
编辑前面板时需要用到控件选板，我们可以用鼠标选择控件选板上的对象，然后将其拖放到前面板上即可。

在前面板中，一般使用**定位工具**来选择对象。

## 2.2.2 对象的放置与选择

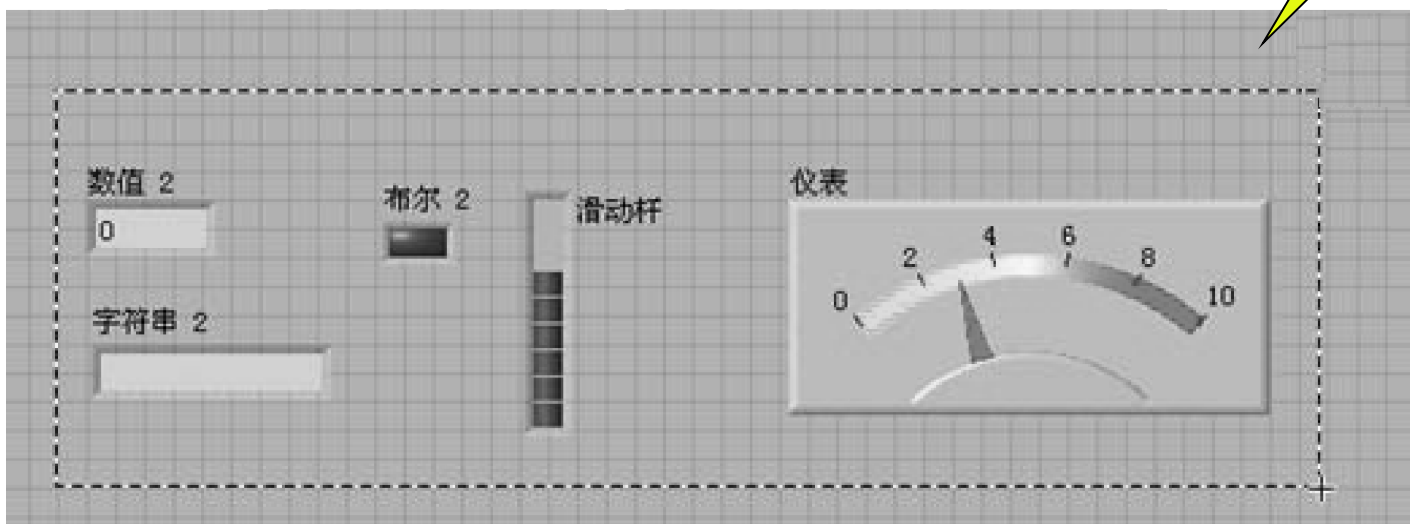


(a) 单个对象被选择



(b) 用shift键选择多个对象

闪动的虚线表示选中的对象控件。



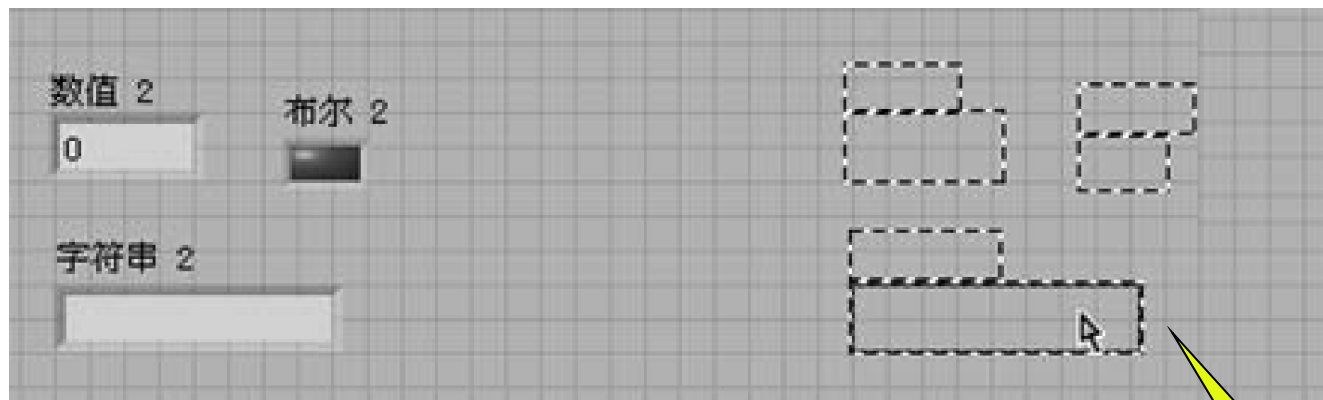
(c) 用鼠标左键拖动选择多个对象

图2-10 使用定位工具选择对象

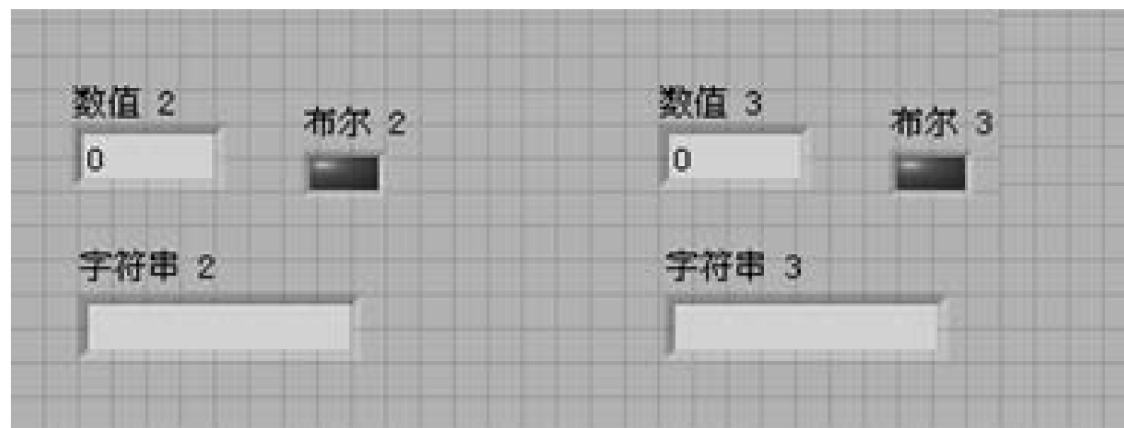
## 2.2.3 对象的移动、复制与删除

利用定位工具单击选择对象，鼠标形状为**左箭头**时，按住鼠标左键不放，**拖动**鼠标，我们可以将被选对象移动到窗口中希望的位置，如图**2-11（a）**所示。

## 2.2.3 对象的移动、复制与删除



(a) 用鼠标拖动实现



(b) 对象复制

左箭头时，  
代表选中。  
**CTRL**键+拖  
动代表复制。

图2-11 对象的移动、复制



## 2.2.3 对象的移动、复制与删除

多数对象都可以复制，支持**ctrl+v**，**ctrl+x**，**ctrl+C**操作。

用户选中对象后，然后“**Delete**”键，即可删除对象。

但控件对象处于“锁定”状态时，无法删除。

## 2.2.4 对象大小的调整

一般情况下，控件选板上的对象是以默认的大小被拖放到前面板上的，但是这些属性不一定适合某一具体程序的需要，其外观的一些属性都可以通过简单的操作来进行修改。

## 2.2.4 对象大小的调整

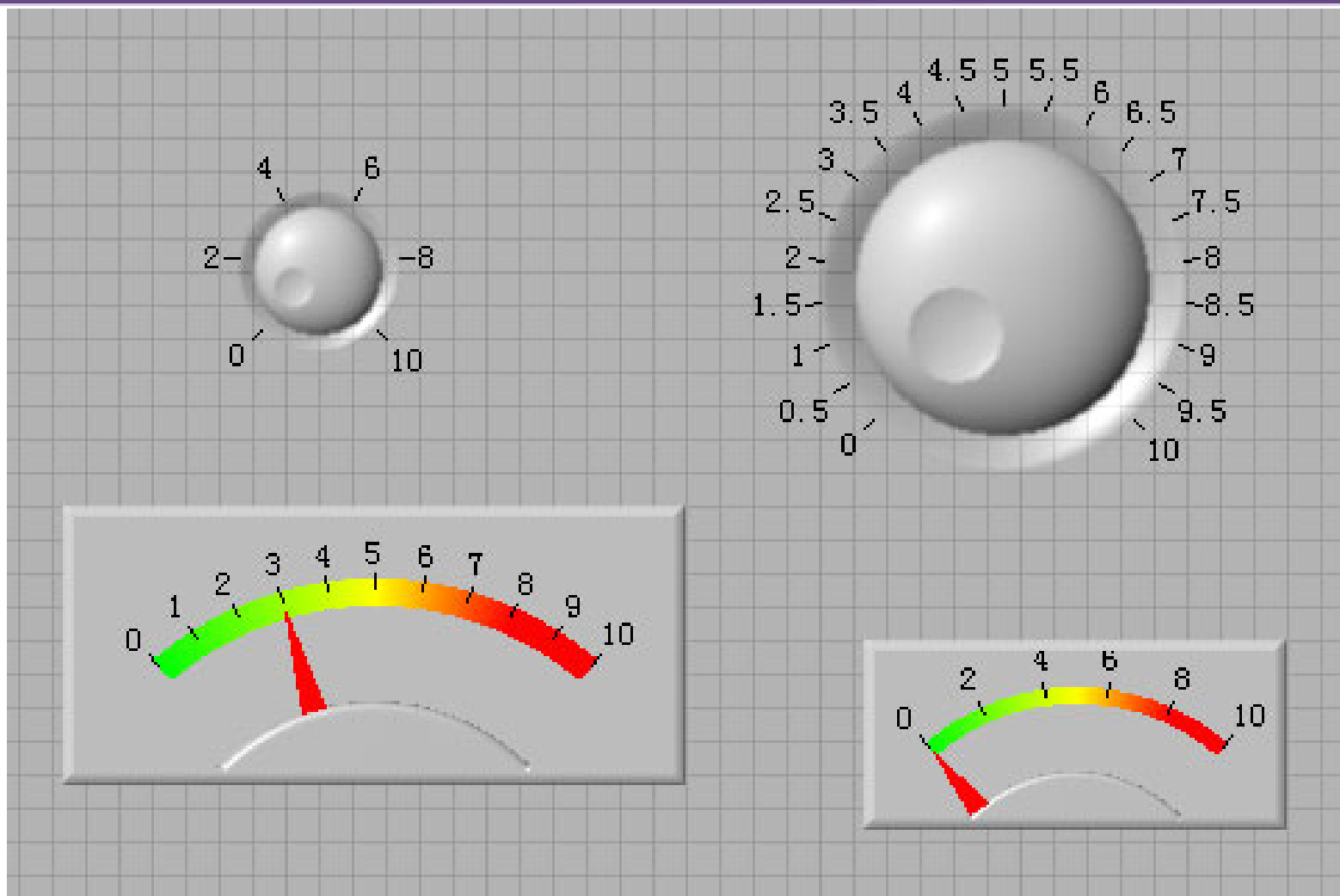


图2-12 对象大小的调整

## 2.2.5 对象的着色与字体选择

在建立前面板和框图对象时，**LabVIEW**会自动给对象着色。用户可利用**工具选板**中提供的**颜色工具**来编辑前面板和前面板控件的颜色，但不是**编辑系统风格控件的颜色**，因为系统风格控件的颜色由操作系统决定。

也可以在“鼠标关闭自动选择”时，利用“**Tab**”键来轮巡切换鼠标的功能性质。

可以“**shift**”+**右键**，调出“**工具选板**”。

## 2.2.5 对象的着色与字体选择

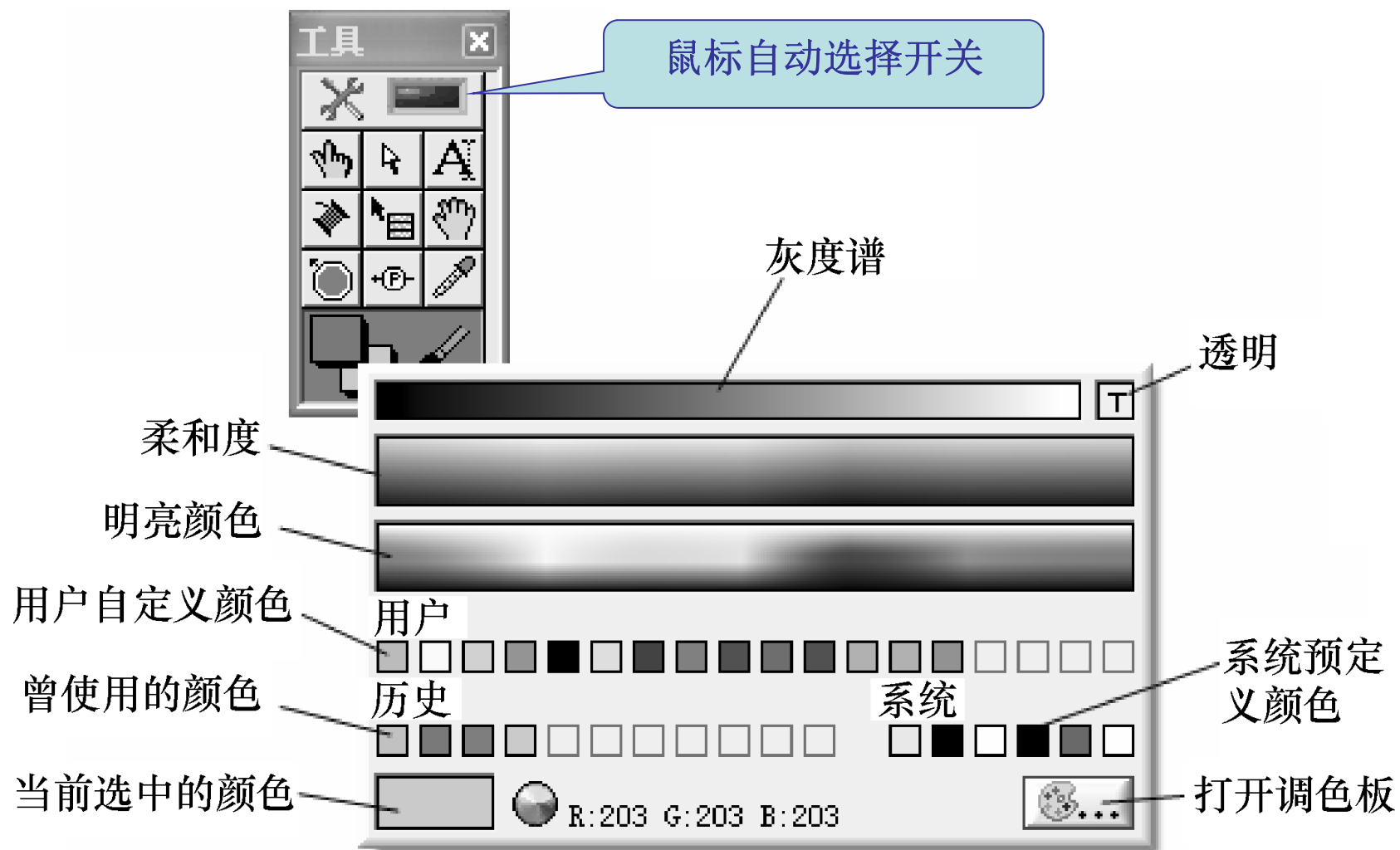


图2-13 颜色面板

## 2.2.7 控件属性的设置

前面板上的每个控件都有自己的属性，如大小、字体、颜色、显示模式等。

属性有两种修改方式，静态、动态。

**静态时**，在程序没运行时用鼠标修改。

**动态**指在程序运行的过程中，利用编程的改进。

## 2.2.7 控件属性的设置

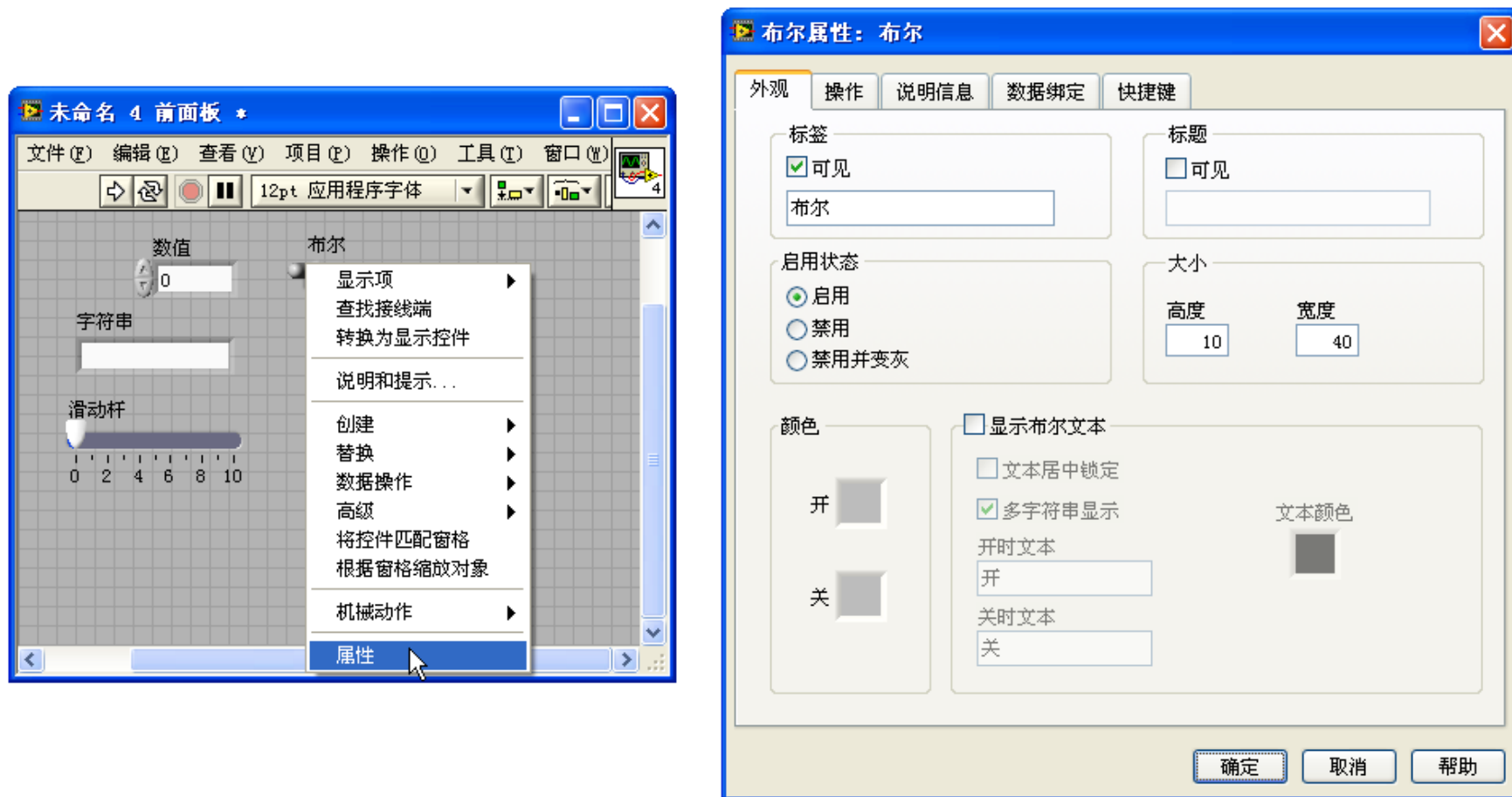


图2-17 控件属性对话框

## 2.2.7 控件属性的设置

在属性对话框中，操作功能界面包括以下几项内容：

- (1) “外观”：设置控件开/关时的颜色和文本、控件标签的显示和隐藏等功能；
- (2) “操作”：设置布尔控件的机械动作方式；



## 2.2.7 控件属性的设置

- (3) “说明信息”：为控件添加描述和提示信息；
- (4) “数据绑定”：将该控件与网络上相应的数据源相连接；
- (5) “快捷键”：设置该控件的快捷键。

## 2.2.7 控件属性的设置

控件的隐藏

按钮的动作

## ■ 2.3 程序框图的编辑

在前面板中添加控件后，为了实现对前面板中对象的控制，用户必须要创建程序框图，**程序框图是图形化代码的集合。**

### 2.3.1 前面板对象在框图中的显示

通过选择“窗口”下拉菜单中的“显示程序框图”或**鼠标双击对象框图窗口**中任意位置，可实现前面板与程序框图的切换。

**快捷键切换 “ctrl+E”**

## ■ 2.3 程序框图的编辑

当用户在前面板中创建控件对象后，**LabVIEW**会在程序框图中自动添加该对象的编程图标（节点），如图**2-18**所示。

**注意：**修饰控件在程序框图上没有对应的节点图标。

常量图标在前面板上也没有对应的控件。

## ■ 2.3 程序框图的编辑

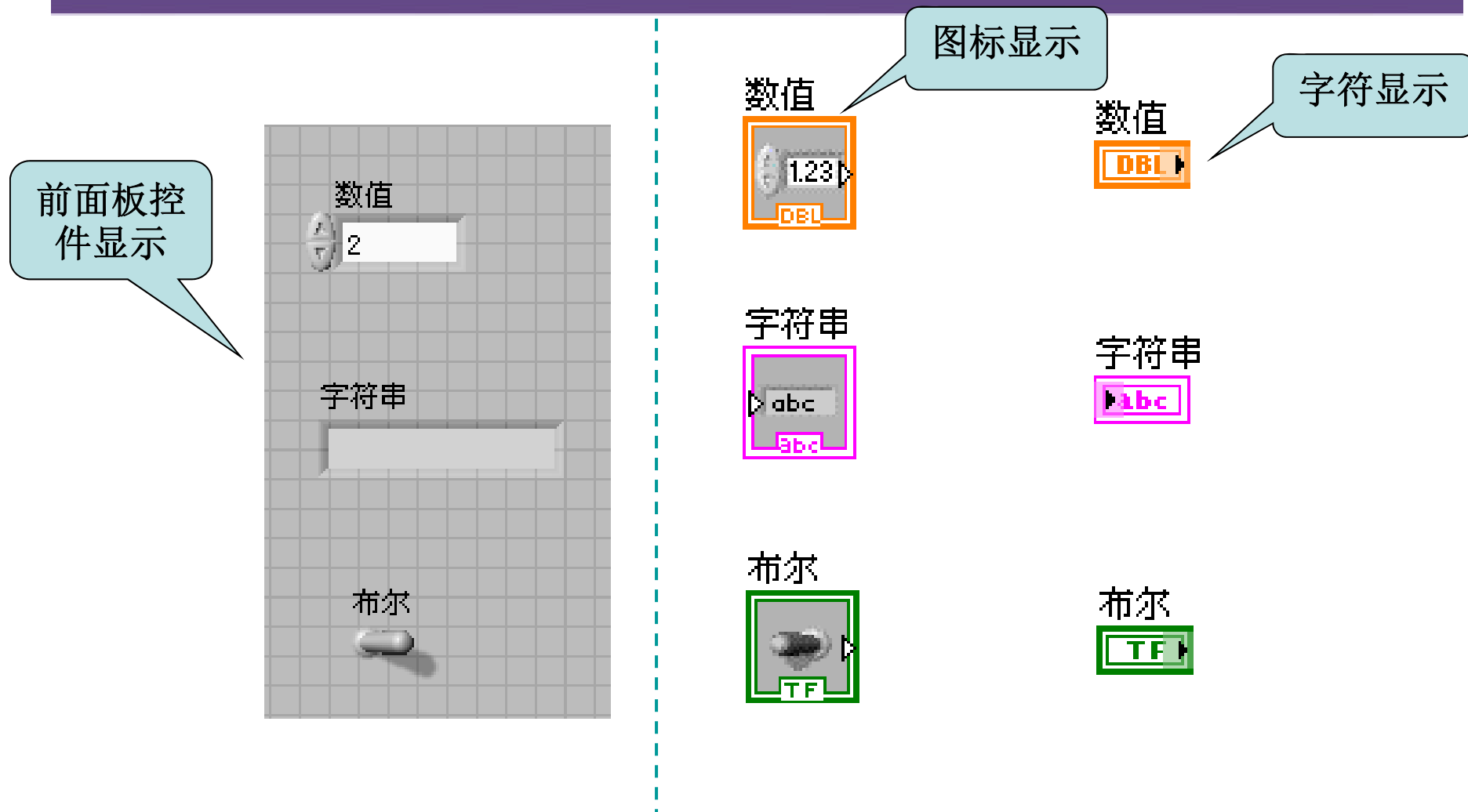


图2-18 程序框图中的控件

## ■ 2.3 程序框图的编辑

VI的运行机制是“**数据流**”，只有节点的输入端口的流入数据全部为有效输入时，节点功能才会执行，之后，输出端口才有效。这就是**LABVIEW**的“**数据流驱动**”机制。


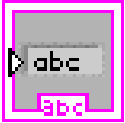


如果我们在程序中设计了若干个“独立的”数据流过程，那么它们将同时按照**各自的数据流程**运行。数据流完成，程序结束。

在程序框图中，数据流的表示就是“数据连线”，连线的“规范”是“**从左到右**”设计。但实际运行取决于节点的数据端口。

程序框图中，接线端**不同的颜色代表着不同的数据类型**，各种常用数据类型的代表颜色如表**2-1**所示。



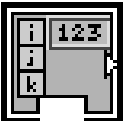

## ■ 2.3 程序框图的编辑

表2-1 与前面板控件对应的常用数据类型

数据类型	端口图标	图标颜色	默认值	简要说明
布尔量		绿色	False	存储为8位数值（一个字节），数值0表示False，其余数值代表True
字符串		粉红	空字符串	存储为一个指针，指向的结构包括一个数值和一个数组。数组每个成员表达一个字符，数值表达数组长度
时间标记		棕色	本地日期时间	存储为一个簇，前两个整数用64位表示s的整数部分，后两个整数用64位表示s的小数部分
枚举		蓝色	—	列出所有的可选值，它的数字表达是无符号整型数

## ■ 2.3 程序框图的编辑

续表

数据类型	端口图标	图标颜色	默认值	简要说明
路径		浅绿	—	使用标准的Windows语法，可以使用相对路径和绝对路径。存储为路径类型、路径成员数和路径成员
参考号		浅绿	—	LabVIEW进行操作的对象，例如文件、设备等的标识 (类似于Windows的句柄)
数组		随成员变	—	图标的方括号内包含数组成员的类型，图标的颜色随成员的数据类型改变
簇		棕或粉红	—	包含多个不同数据类型的成员，如果成员都是数值，颜色为棕色，否则为粉红色（结构变量）



## 2.3 程序框图的编辑

续表

数据类型	端口图标	图标颜色	默认值	简要说明
波形		棕色	—	簇，成员包括 $t_0$ ， $\Delta t$ 和 数据 含时间成员的特殊簇。
数字波形		粉红	—	簇，成员包括 $t_0$ ， $\Delta t$ 和 数字数据
数字数据		深绿	—	用表格形式显示数字信号
输入输出名		紫色	—	表示与输入输出设备有关的名称 代表数据采集硬件的名称。
变体		紫色	—	可以容纳各种不同的数据类型，存 储控件名称、数据类型和数据本身

## 2.3.2 程序框图中连线的创建

连线工具是工具模板提供的工具之一。

对象连线实现了程序框图中各对象之间的数据传递。

对象之间的连线可以采用自动连线和手动连线。

连线代表数据流、反映数据流的方向。

# 1. 自动连线

在**LabVIEW**的编程环境中，**默认的连接方式是自动连线**。

**LabVIEW**也可以对程序框图上已有对象进行“**自动连线**”。

**自动连线**是指两个节点图标靠近时，自动把**2**个节点的数据端口用数据流线连接起来。

# 1. 自动连线



自动连线 and 手动连线通过“**空格**”来切换，通过对象外框的虚线有无来表示“自动连线状态”

图2-19 自动连线

## 2. 手动连线

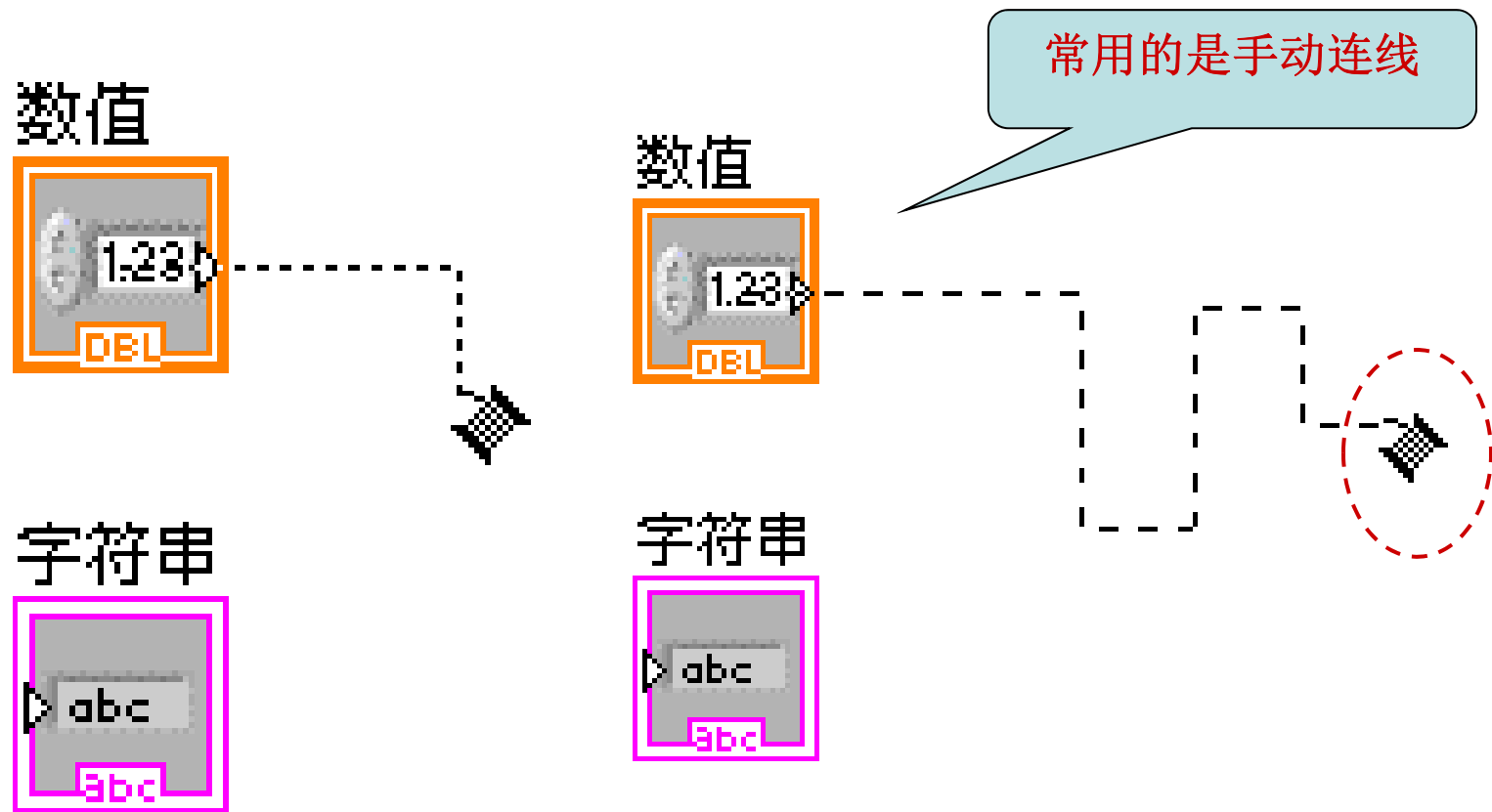


图2-20 手动连线

## 2. 手动连线

连线只能是水平或垂直的，但为了使代码清晰、可读性强，对于连线的排列和分布需要做一定的编辑。要移动或删除连线时，必须先选择该目标连线。

注意：连线的源数据类型和目标数据的类型要一致或相容。

### 2.3.3 程序框图中连线的编辑

选择定位工具，鼠标左键单击某段连线则该段连线变成流动虚线，即表示选中该段，如图2-21（a）所示；鼠标左键双击某段连线则表示选中该连线的一个分支，如图2-21（b）所示；鼠标左键连续单击3次则表示选择全部连线，如图2-21（c）所示。

## 2.3.3 程序框图中连线的编辑

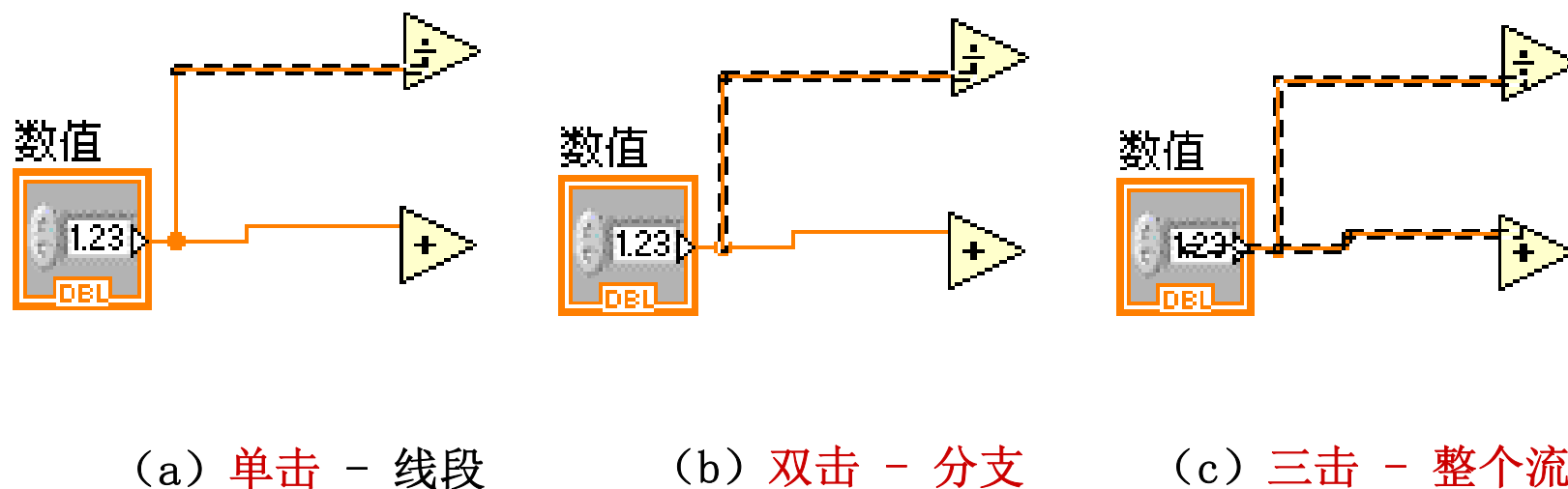


图2-21 连线的编辑 (1)



### 2.3.3 程序框图中连线的编辑

对于被选择的连线或连线段，可以按“**Backspace**”或“**Delete**”键进行删除；也可以用鼠标将其拖动到新的位置。如果连线失败或不可用，就会变成断线。

断线的存在会阻碍程序的运行。

断线显示为黑色的虚线，中间有个红色的“×”。

有断线存在，代表数据流的中断，系统就不会运行。

## 2.3.3 程序框图中连线的编辑

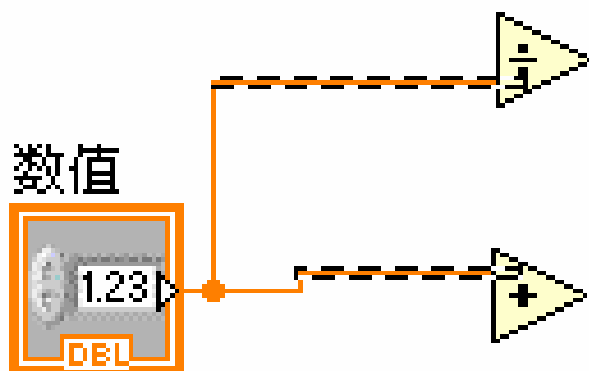


图2-22 连线的编辑 (2)

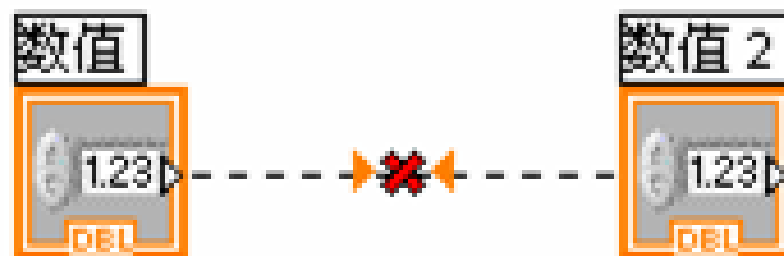


图2-23 数据源错误引起的断线

## 2.3.3 程序框图中连线的编辑

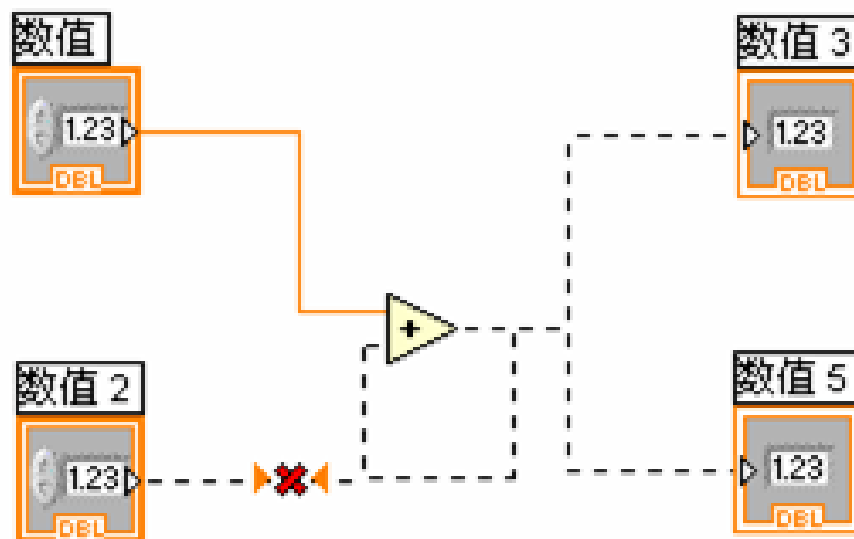
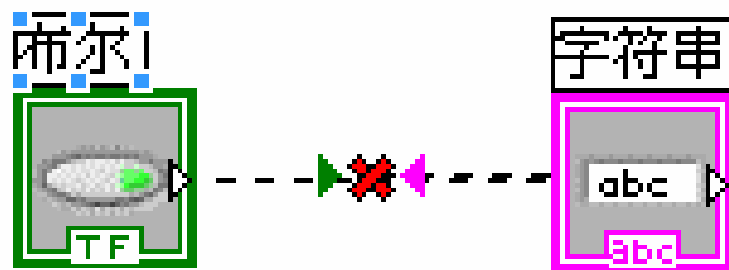


图2-24 数据类型不兼容引起的断线      图2-25 数据回环引起的断线

### 2.3.3 程序框图中连线的编辑

**LabVIEW**还为用户提供了整理连线的功能，此功能可以把框图中混杂不清的连线清楚的显示出来。

## 2.3.3 程序框图中连线的编辑

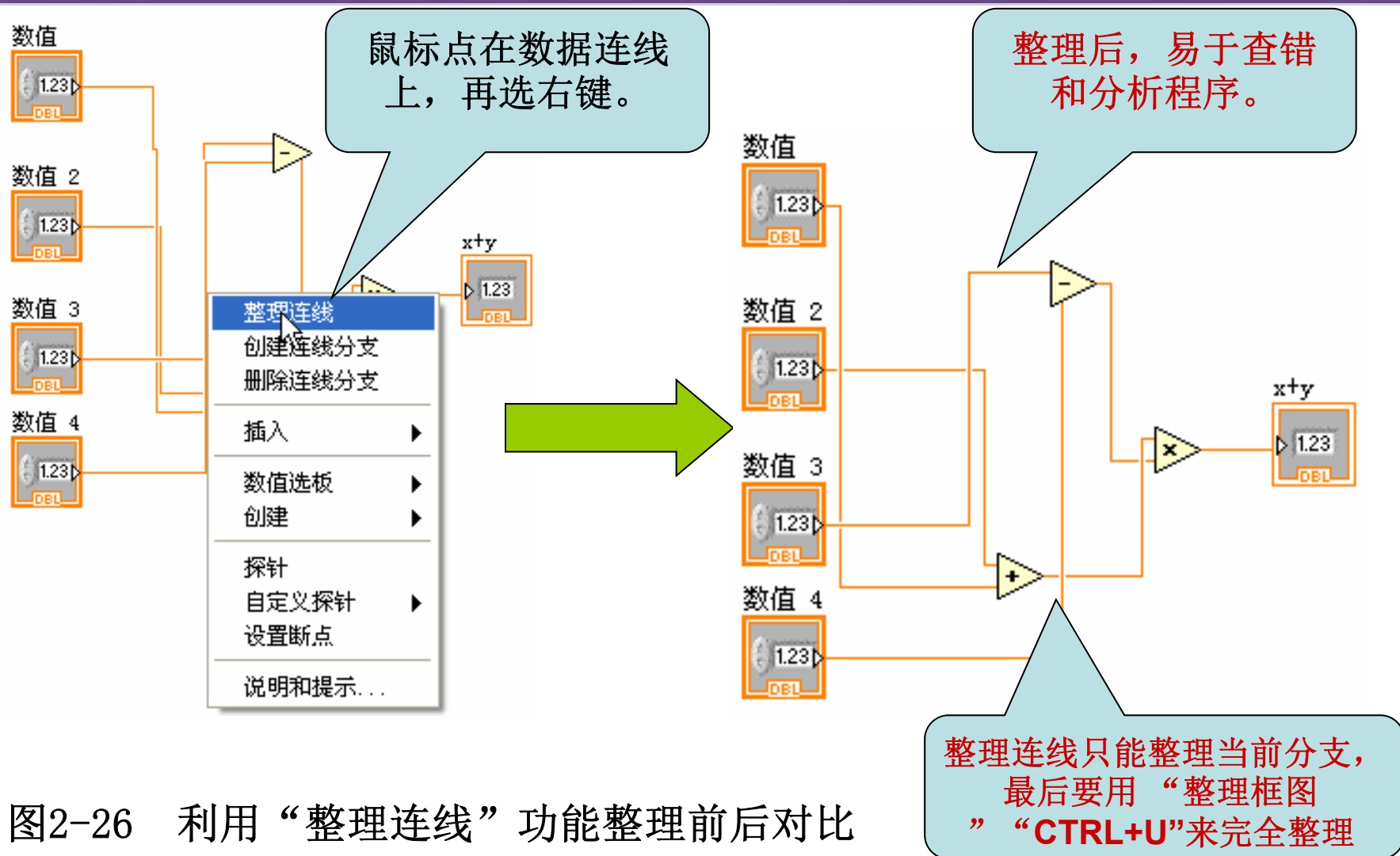


图2-26 利用“整理连线”功能整理前后对比

## 2.3.3 程序框图中连线的编辑



整理连线

## 2.4 VI的运行与调试

在程序编写工作完成后，我们必须经过运行和调试来测试编写的程序是否能够产生预期的运行结果从而找出程序中的一些错误。**LabVIEW**提供了许多的工具来帮助我们完成程序的调试。


## 2.4.1 VI程序的运行

**LabVIEW**中**VI**程序的运行主要包括下列操作。

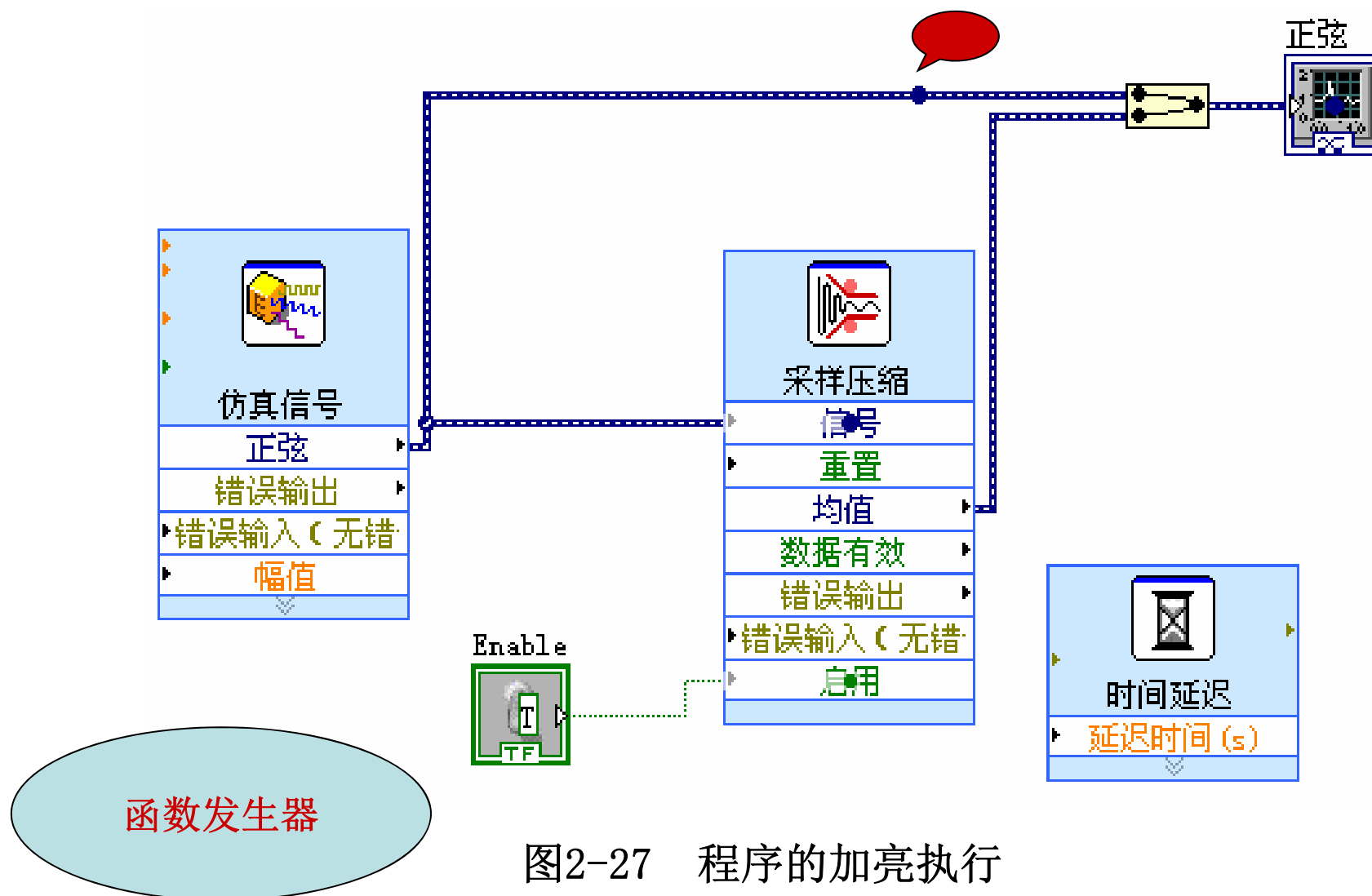
- (1) **VI**的运行。
- (2) **VI**的连续运行。
- (3) **VI**的停止。
- (4) **VI**运行的暂停。



## 2.4.2 VI程序的加亮执行

当单击框图窗口工具条上的加亮执行按钮 , 即可打开执行加亮功能。执行加亮时, 对节点之间的数据流动采用在连线上移动的气泡加以形象表示。

## 2.4.2 VI程序的加亮执行



## 2.4.3 设置VI断点与探针

我们可以在VI的某个位置**设置断点**，来看清程序执行情况。

当数据流过框图连线时，也可使用**探针工具**检查VI运行时的**即时数据**。

## 2.4.3 设置VI断点与探针

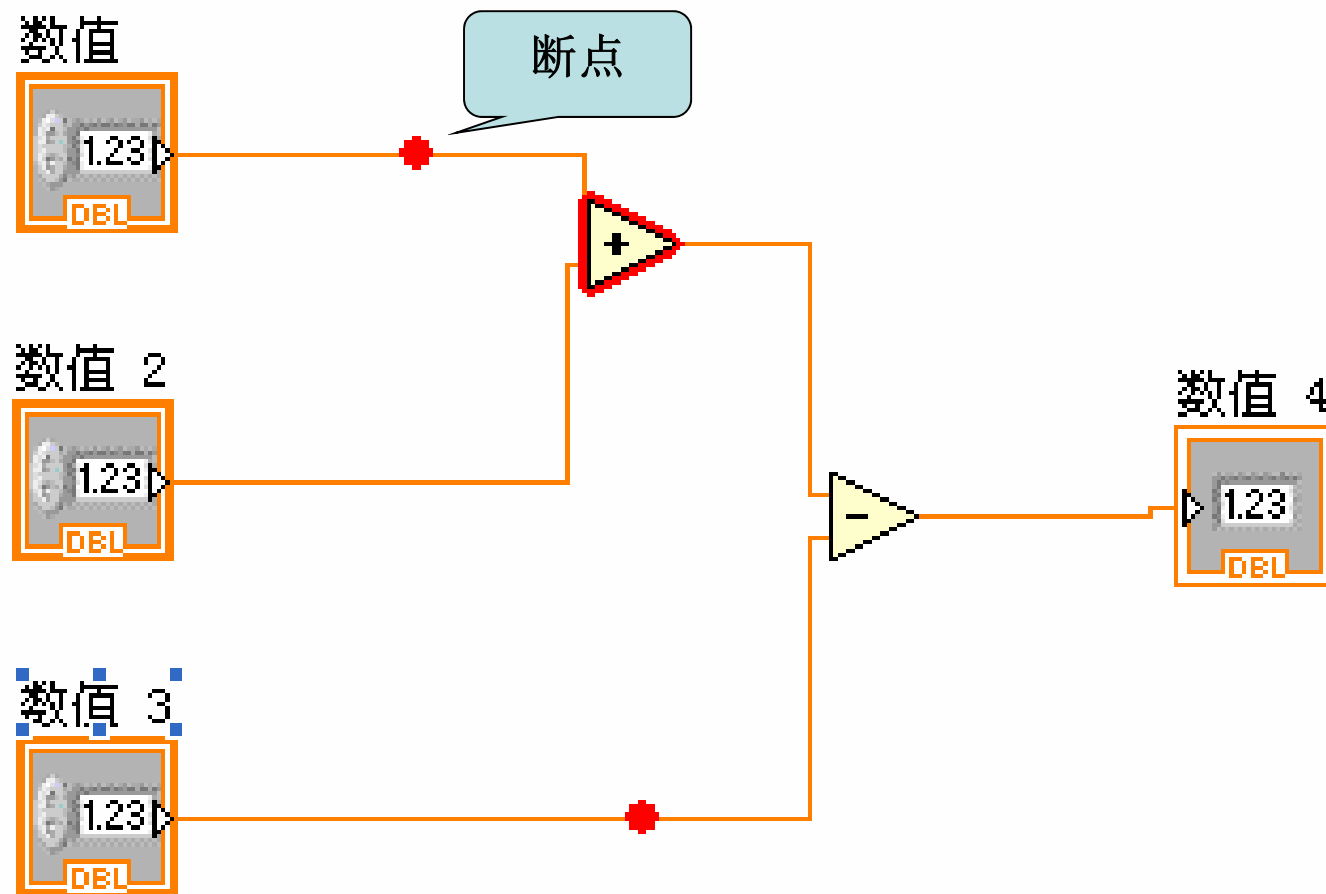


图2-28 程序中的断点

## 2.4.3 设置VI断点与探针

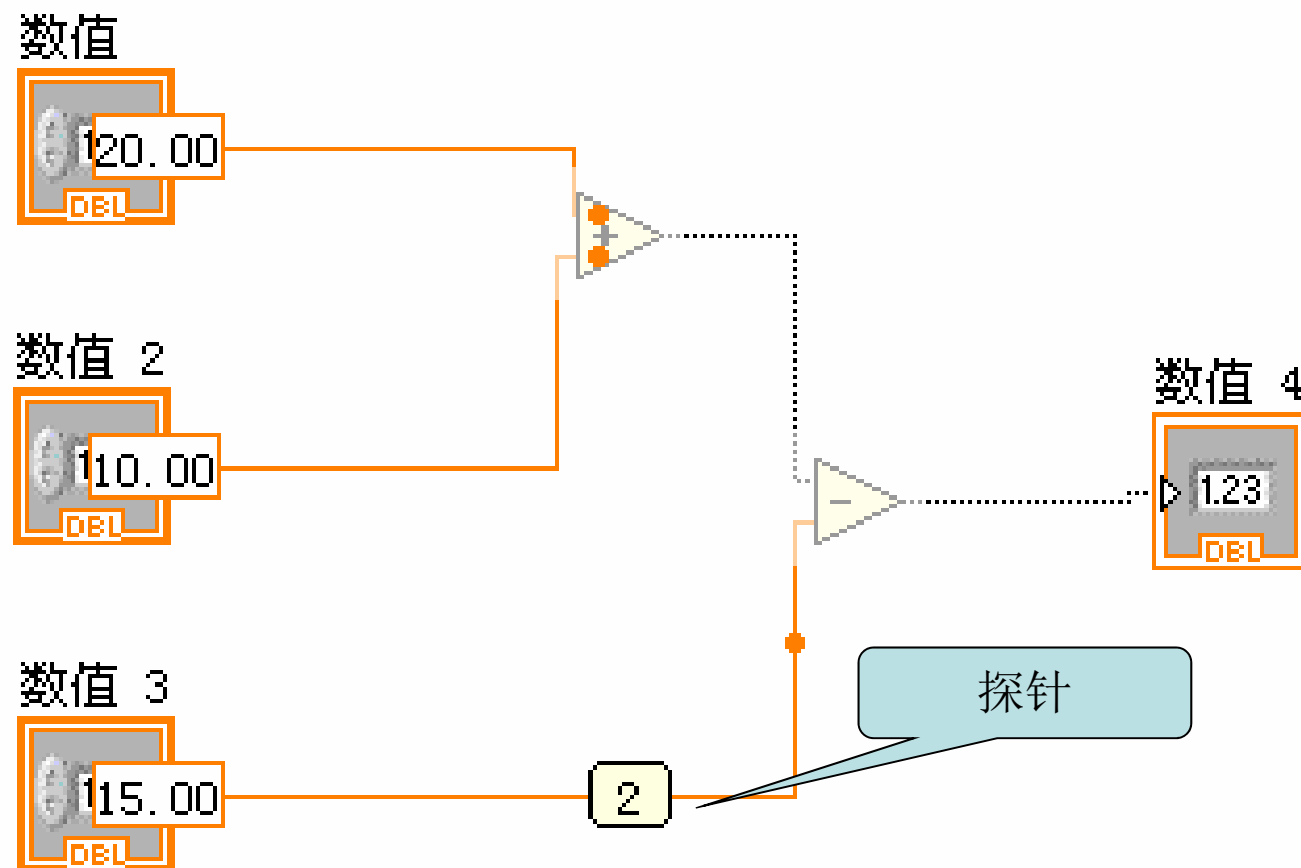


图2-29 程序中的探针

## 2.4.3 设置VI断点与探针



图2-30 探针窗口

算术运算

## 2.4.4 错误列表窗口的使用

程序错误一般分为两种：一种为程序编辑错误或编辑结果不符合语法，程序无法正常运行；另一种错误为语义和逻辑上的错误，或者是程序运行时某种外部条件得不到满足引起的运行错误，这种错误**LABVIEW**很难查出。

## 2.4.4 错误列表窗口的使用

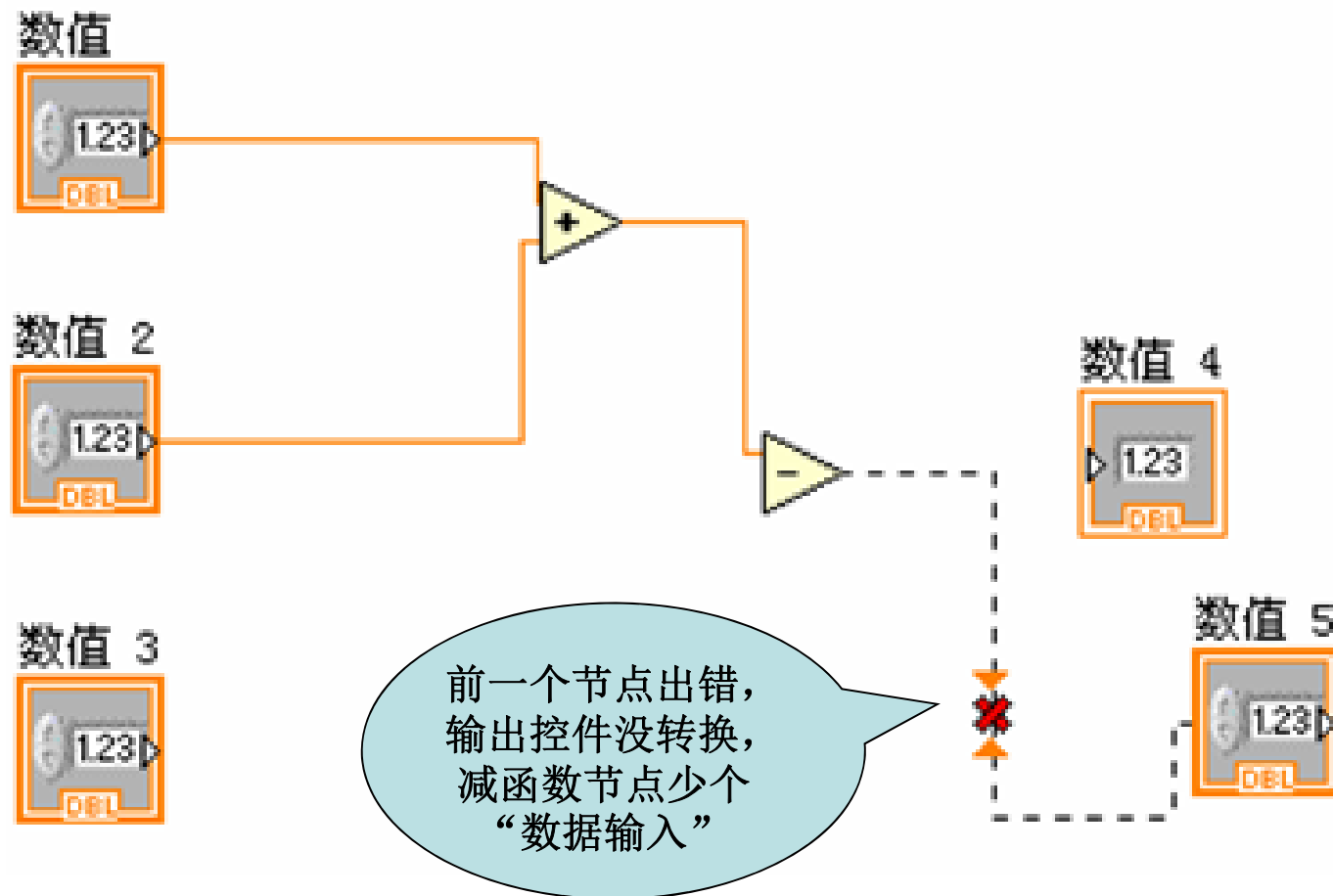


图2-31 一个LabVIEW程序



## 2.4.4 错误列表窗口的使用




双击错误条目，可以切换到对应的错误节点上去。

图2-32 程序的错误列表



## 2.4.5 VI的单步执行

如果想使程序逐个节点执行，可以采用单步执行。在**单步执行**时，可以查看全部代码的执行细节。单步执行方式有**3**种类型。

## 2.4.5 VI的单步执行

- (1) 单击  单步步入执行方式，打开一个要执行的程序节点并暂停，不运行当前节点。
- (2) 单击  单步步过执行方式，首先执行打开的程序节点，然后停止在下一个节点处。
- (3) 单击  单步步出执行方式，执行完当前节点内容立即暂停。

## 2.4.6 VI程序调试技巧

**LabVIEW**对用户的编程过程进行即时语法检查，对于不符合语法规则的连线或没有连接必须要连接的端子时，工具栏中的运行按钮将由  变为 。

## 2.4.6 VI程序调试技巧

系统对于错误的准确定位，能够有效提高调试程序的效率。

在大多数情况下，程序虽然可以运行，但是无法得出期望的结果。这种错误一般较难发现，对于这种**逻辑错误**，查找过程可以按以下步骤进行。

## 2.4.6 VI程序调试技巧

(1) 检查连线是否连接适当。

利用整理框图 (**CTRL+U**) 功能。

(2) 使用“帮助”下拉菜单中的“显示即时帮助”功能 (**CTRL+H**) 来动态显示鼠标所指向的函数或子程序的用法介绍以及各端口的定义，然后对比当前的连线检查连线的正确性。

(3) 检查某些函数或子程序的端口默认值，尤其是当函数或子程序的端口类型是可选类型的时候。

## 2.4.6 VI程序调试技巧

(4) 对于复杂的VI,选择“查看”下拉菜单中的“VI层次结构”。

(5) 通过使用加亮执行方式、单步执行方式以及设置断点等手段来检查程序是否是按预定要求运行的。

(6) 通过VI分析器，对VI扫描，查错。

## 2.4.6 VI程序调试技巧

(6) 通过使用探针工具来获取连线上的即时数以及检查函数或子程序的输出是否存在错误。

(7) 检查函数或子程序输出的数据是否有意义的数据。



## 2.4.6 VI程序调试技巧

- (8) 检查控件和指示器的数据是否有溢出。
- (9) 当For循环的循环次数为0时，需要注意此时将会产生一个空数组，当调用该空数组时需要事先作特殊的处理。
- (10) 检查簇成员的顺序是否与目标端口一致。
- (11) 检查是否有未连线的VI子程序。

# 本章小结

**VI**是**LabVIEW**程序的基础单元。本章通过多个**VI**创建示例介绍了创建**VI**的典型过程，着重介绍了如何在前面板中添加、删除以及编辑控件对象，如何在程序框图中编辑连线，最后介绍了如何运行**VI**、调试**VI**、运行错误的查找和解决方法以及一些常用的**VI**调试技巧。