

NI LabVIEW Environment: Detailed Explanation

Virtual Instruments (VIs)

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. LabVIEW contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data as well as tools to help you troubleshoot the code you write.

When you create a new VI, you see two windows: the front panel window and the block diagram.

Front Panel

When you open a new or existing VI, the front panel window of the VI appears. The front panel window is the user interface for the VI. **Figure 1** shows an example of a front panel window.

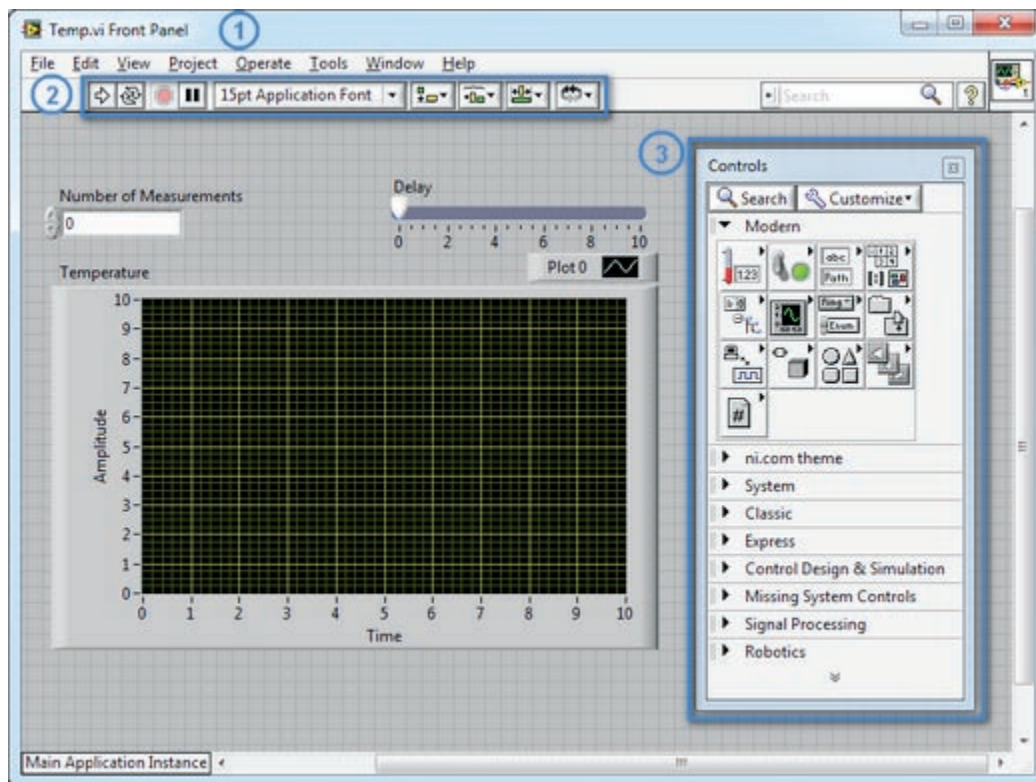


Figure 1. Front Panel Window, Toolbar, and Controls Palette

Controls Palette

The Controls palette contains the controls and indicators you use to create the front panel. You access the Controls palette from the front panel window by selecting **View >> Controls Palette** or by right-clicking on any empty space in the front panel window. The Controls palette is broken into various categories; you can expose some or all of these categories to suit your needs. **Figure 2** shows a Controls palette with all of the categories exposed and the Modern category expanded.

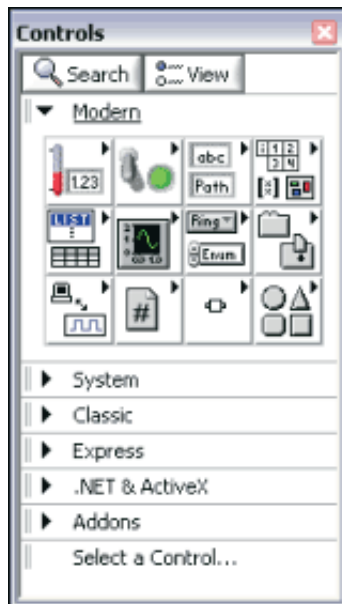


Figure 2. Controls Palette

To view or hide categories (subpalettes), select the View button on the palette and select or deselect in the Always Visible Categories option.

Controls and Indicators

You create the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data that the block diagram acquires or generates.

Figure 1 has two controls: Number of Measurements and Delay (sec). It has one indicator: an XY graph named Temperature Graph.

The user can change the input value for the Number of Measurements and Delay (sec) controls. The user can see the value generated by the VI on the Temperature Graph indicator. The VI generates the values for the indicators based on the code created on the block diagram.

Every control or indicator has a data type associated with it. For example, the Delay (sec) horizontal slide is a numeric data type. The most commonly used data types are numeric, Boolean value, and string.

Numeric Controls and Indicators

The numeric data type can represent numbers of various types, such as integer or real. The two common numeric objects are the numeric control and the numeric indicator, as shown in **Figure 3**. Objects such as meters and dials also represent numeric data.

To enter or change values in a numeric control, click the increment and decrement or double-click the number, enter a new number, and press the <Enter> key.

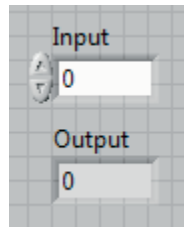


Figure 3. Numeric Control Increment/Decrement, Control, Indicator

Boolean Controls and Indicators

The Boolean data type represents data that has only two possible states, such as TRUE and FALSE or ON and OFF. Use Boolean controls and indicators to enter and display Boolean values. Boolean objects simulate switches, push buttons, and LEDs. The vertical toggle switch and the round LED Boolean objects are shown in **Figure 4**.



Figure 4. Boolean Controls and Indicators

String Controls and Indicators

The string data type is a sequence of ASCII characters. Use string controls to receive text from the user such as a password or user name. Use string indicators to display text to the user. The most common string objects are tables and text entry boxes, as shown in **Figure 5**.

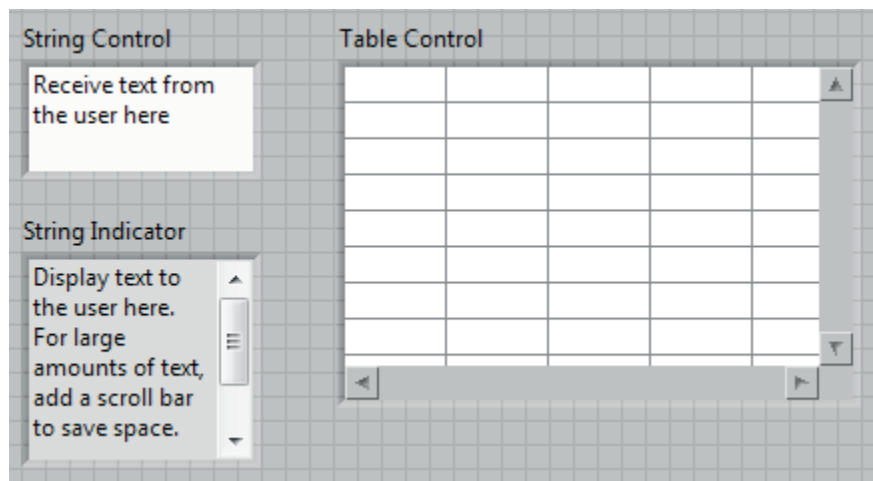


Figure 5. String Controls and Indicators

Block Diagram

Block diagram objects include terminals, subVIs, functions, constants, structures, and wires, which transfer data among other block diagram objects.

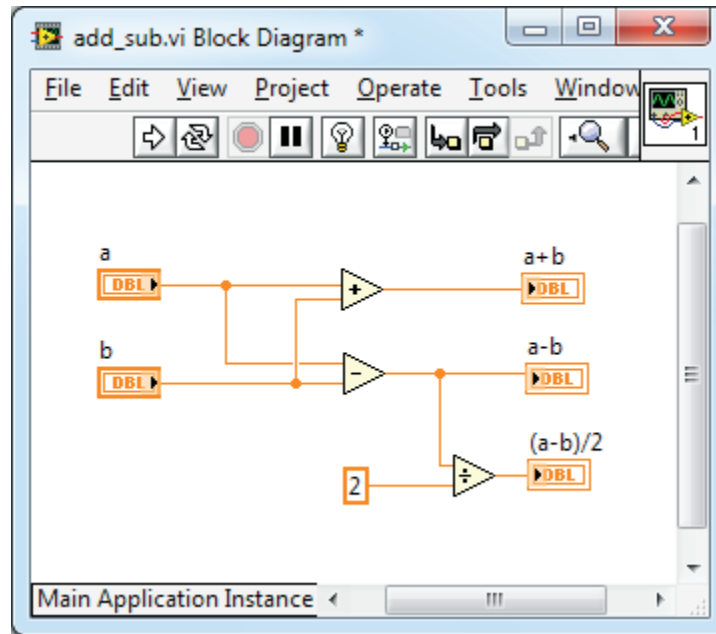


Figure 6. Indicator Terminals, Wires, Nodes, Control Terminals

Block Diagram Window

After you create the front panel window, you add code using graphical representations of functions to control the front panel objects. The block diagram window contains this graphical source code.

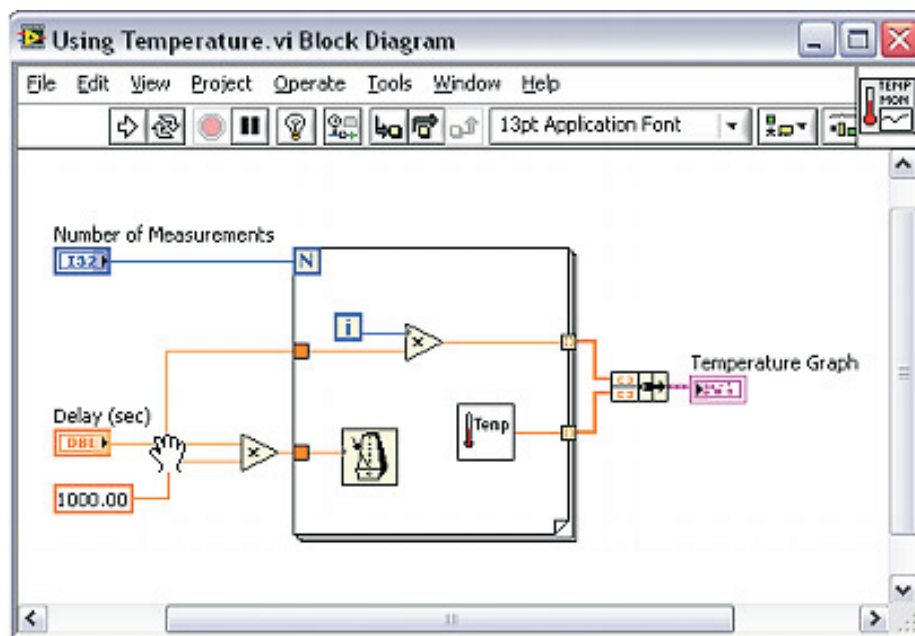


Figure 7. Block Diagram

Block Diagram Terminals

Objects on the front panel window appear as terminals on the block diagram. Terminals are entry and exit ports that exchange information between the front panel and block diagram. Terminals are analogous to parameters and constants in text-based programming languages.

Types of terminals include control or indicator terminals and node terminals. Control and indicator terminals belong to front panel controls and indicators. Data you enter into the front panel controls (a and b in the previous front panel) enters the block diagram through the control terminals. The data then enters the Add and Subtract functions. When the Add and Subtract functions complete their calculations, they produce new data values. The data values flow to the indicator terminals, where they update the front panel indicators (a+b and a-b in the previous front panel).

Controls, Indicators, and Constants

Controls, indicators, and constants behave as inputs and outputs of the block diagram algorithm. Consider the implementation of the algorithm for the area of a triangle:

$$\text{Area} = .5 * \text{Base} * \text{Height}$$

In this algorithm, Base and Height are inputs and Area is an output, as shown in **Figure 8**.

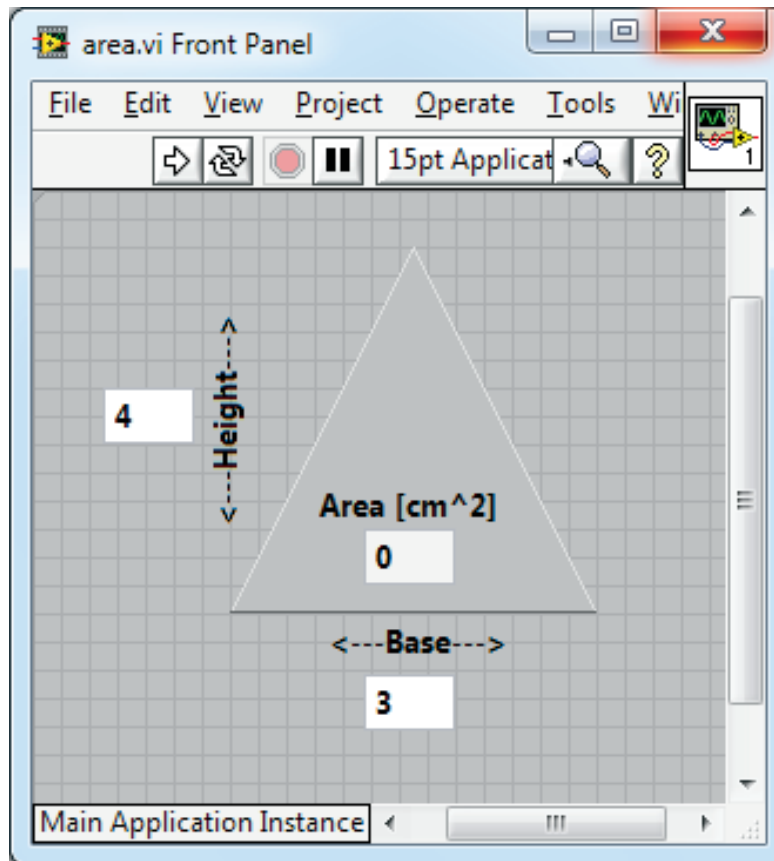


Figure 8. Area of a Triangle Front Panel

The user does not change or access the constant .5, so it does not appear on the front panel unless included as documentation of the algorithm.

Figure 9 shows a possible implementation of this algorithm on a LabVIEW block diagram. This block diagram has four different terminals created by two controls, one constant, and one indicator.

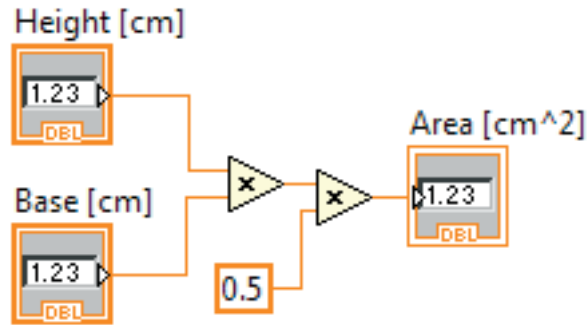


Figure 9. Control, Indicator, Constant

Notice that the Base (cm) and Height (cm) block diagram terminals have a different appearance from the Area (cm²) terminal. There are two distinguishing characteristics between a control and an indicator on the block diagram. The first is an arrow on the terminal that indicates the direction of data flow. The controls have arrows showing the data leaving the terminal, whereas the indicator has an arrow showing the data entering the terminal. The second distinguishing characteristic is the border around the terminal. Controls have a thick border and indicators have a thin border.

You can view terminals with or without icon view. **Figure 10** shows the same block diagram without using the icon view of the terminals; however, the same distinguishing characteristics between controls and indicators exist.

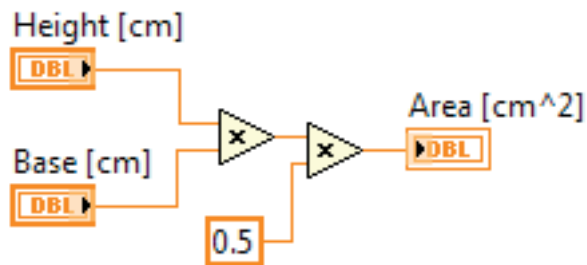


Figure 10. Area of a Triangle Block Diagram Without Icon Terminal View

Block Diagram Nodes

Nodes are objects on the block diagram that have inputs and/or outputs and perform operations when a VI runs. They are analogous to statements, operators, functions, and subroutines in text-based programming languages. Nodes can be functions, subVIs, or structures. Structures are process control elements, such as Case structures, For Loops, or While Loops. The Add and Subtract functions in **Figure 6** are function nodes.

Functions

Functions are the fundamental operating elements of LabVIEW. They do not have front panel windows or block diagram windows but they do have connector panes. Double-clicking a function only selects the function. A function has a pale yellow background on its icon.

SubVIs

SubVIs are VIs that you create to use inside another VI or that you access on the Functions palette. A subVI is similar to a function in a text-based programming language.

Any VI has the potential to be used as a subVI. When you double-click a subVI on the block diagram, its front panel window appears. The front panel includes controls and indicators. The block diagram includes wires, icons, functions, possibly subVIs, and other LabVIEW objects.



Every VI displays an icon in the upper right corner of the front panel window and block diagram window. An example of the default icon is shown above. An icon is a graphical representation of a VI. The icon can contain both text and images. If you use a VI as a subVI, the icon identifies the subVI on the block diagram of the VI. The default icon contains a number that indicates how many new VIs you opened after launching LabVIEW.



To use a VI as a subVI, you need to build a connector pane, shown above. The connector pane is a set of terminals on the icon that corresponds to the controls and indicators of that VI, similar to the parameter list of a function call in text-based programming languages. Access the connector pane by right-clicking the icon in the upper right corner of the front panel window. You cannot access the connector pane from the icon in the block diagram window.

SubVIs also can be Express VIs. Express VIs are nodes that require minimal wiring because you configure them with dialog boxes. Use Express VIs for common measurement tasks. You can save the configuration of an Express VI as a subVI. Refer to the Express VIs topic of the LabVIEW Help for more information about creating a subVI from an Express VI configuration.

LabVIEW uses colored icons to distinguish between Express VIs and other VIs on the block diagram. Icons for Express VIs appear on the block diagram as icons surrounded by a blue field, whereas subVI icons have a yellow field.

Functions Palette

The Functions palette contains the VIs, functions, and constants you use to create the block diagram. You access the Functions palette from the block diagram by selecting **View >> Functions Palette**. The palette is broken into various categories; you can show and hide categories to suit your needs. **Figure 11** shows a Functions palette with all of the categories exposed and the Programming category expanded.

To view or hide categories, click the **View** button on the palette and select or deselect the **Change Visible Categories** option.

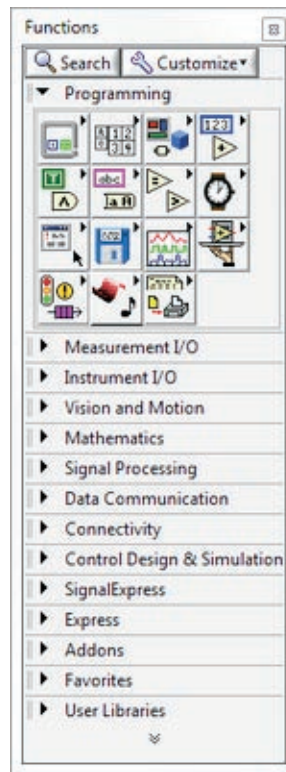
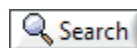


Figure 11. Functions Palette

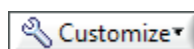
Searching for Controls, VIs, and Functions

When you select **View >> Controls** or **View >> Functions** to open the Controls and Functions palettes, two buttons appear at the top of the palette.

The Search button changes the palette to search mode so you can perform text-based searches to locate controls, VIs, or functions on the palettes. While a palette is in search mode, click the Return button to exit search mode and return to the palette.



The Customize button provides options for selecting a format for the current palette, showing and hiding categories for all palettes, and sorting items in the Text and Tree formats alphabetically. Select Options from the shortcut menu to display the Controls/Functions palettes page of the Options dialog box, in which you can select a format for all palettes. This button appears only if you click the thumbtack in the upper left corner of a palette to pin the palette.



Until you are familiar with the location of VIs and functions, search for the function or VI using the Search button. For example, if you want to find the Random Number function, click the Search button on the Functions palette toolbar and start typing Random Number in the text box at the top of the palette. LabVIEW lists all matching items that either start with or contain the text you typed. You can click one of the search results and drag it to the block diagram, as shown in **Figure 12**. Double-click the search result to highlight its location on the palette.

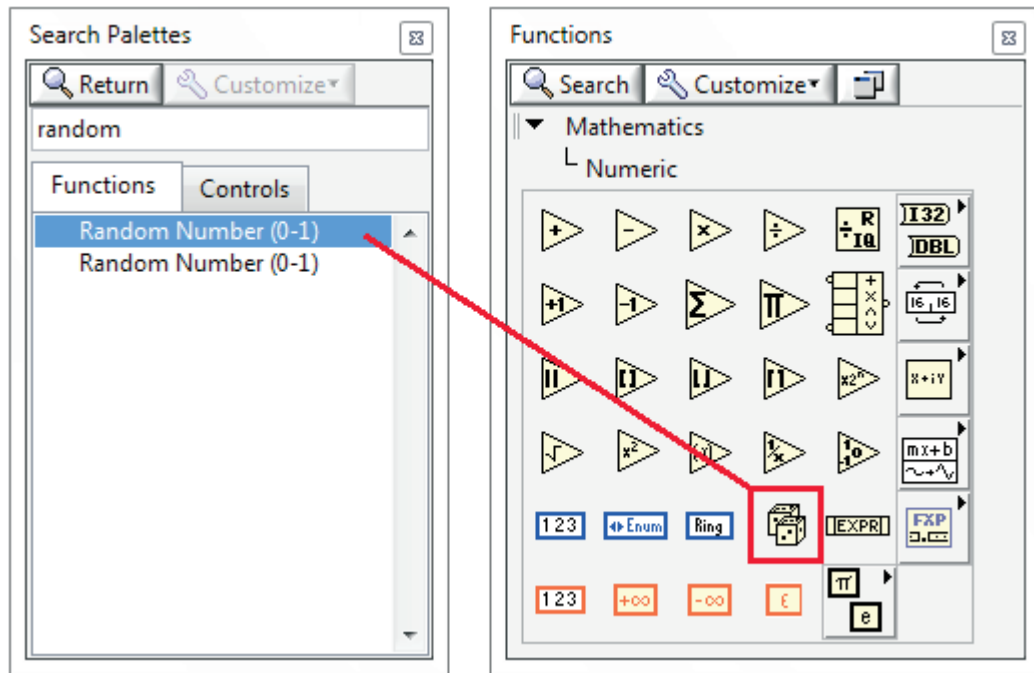


Figure 12. Searching for an Object in the Functions Palette