

The LabVIEW Programming Environment and Basic Operations

Introduction

In this Exercise, you will learn about the LabVIEW programming environment. You will also write a simple Virtual Instrument (VI) to incorporate basic operations in LabVIEW. Finally, you will convert your programs into subVIs.

Objectives

- Learn the three parts of a VI
- Learn the three palettes.
- Learn how data is passed in LabVIEW.
- Distinguish between controls and indicators on the front panel and block diagram.
- Create a subVI using two different methods.

Theory: LabVIEW Programming Basics

LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabVIEW uses dataflow programming, where the flow of data determines execution.

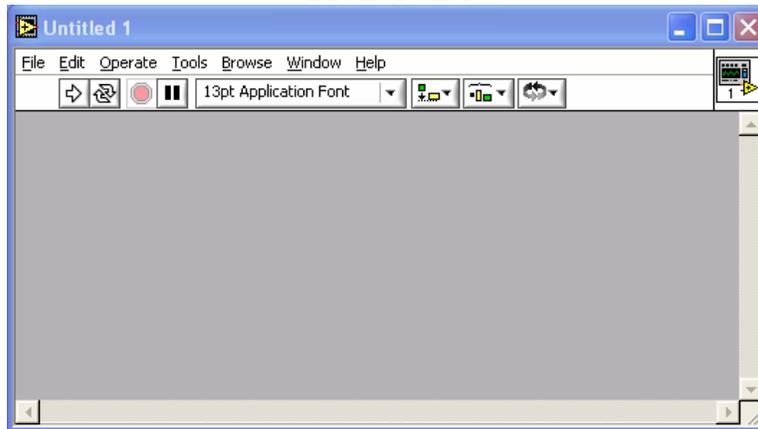
In LabVIEW, you build a user interface by using a set of tools and objects. The user interface is known as the front panel. You then add code using graphical representations of functions to control the front panel objects. The block diagram contains this code. In some ways, the block diagram resembles a flowchart.

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. Every VI uses functions that manipulate input from the user interface or other sources and display that information or move it to other files or other computers.

A VI contains the following three components:

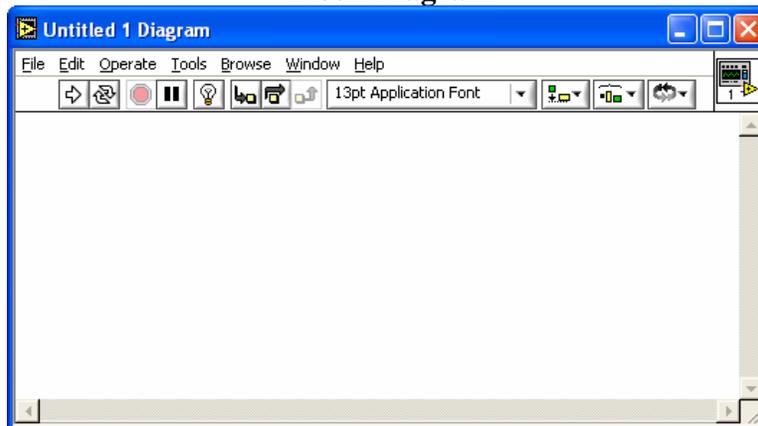
- **Front panel** — Serves as the user interface.
- **Block diagram** — Contains the graphical source code that defines the functionality of the VI.
- **Icon and connector pane** — Identifies the VI so that you can use the VI in another VI. A VI within another VI is called a subVI. A subVI corresponds to a subroutine in text-based programming languages.

Front Panel



The front panel is the user interface of the VI. You build the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, pushbuttons, 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 the block diagram acquires or generates.

Block Diagram



After you build the front panel, you add code using graphical representations of functions to control the front panel objects. The block diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram.

Additionally, the block diagram contains functions and structures from built-in LabVIEW VI libraries. Wires connect each of the nodes on the block diagram, including control and indicator terminals, functions, and structures.

LabVIEW Palettes

LabVIEW palettes give you the options you need to create and edit the front panel and block diagram.

The **Tools** palette is available on the front panel and the block diagram.

A tool is a special operating mode of the mouse cursor. When you select a tool, the cursor icon changes to the tool icon. Use the tools to operate and modify front panel and block diagram objects.

Select **Window»Show Tools Palette** to display the **Tools** palette. You can place the **Tools** palette anywhere on the screen.

If automatic tool selection is enabled and you move the cursor over objects on the front panel or block diagram, LabVIEW automatically selects the corresponding tool from the Tools palette.

The **Controls** palette is available only on the front panel. The **Controls** palette contains the controls and indicators you use to create the front panel. Select **Window»Show Controls Palette** or right-click the front panel workspace to display the **Controls** palette. You can place the **Controls** palette anywhere on the screen.

The **Functions** palette is available only on the block diagram. The **Functions** palette contains the VIs and functions you use to build the block diagram. Select **Window»Show Functions Palette** or right-click the block diagram workspace to display the **Functions** palette. You can place the **Functions** palette anywhere on the screen.

Dataflow Programming

LabVIEW follows a dataflow model for running VIs. A block diagram node executes when all its inputs are available. When a node completes execution, it supplies data to its output terminals and passes the output data to the next node in the dataflow path.

Creating SubVIs

After you build a VI and create its icon and connector pane, you can use it in another VI. A VI called from the block diagram of another VI is called a subVI. A subVI corresponds to a subroutine in text-based programming languages. A subVI node corresponds to a subroutine call in text-based programming languages. The node is not the subVI itself, just as a subroutine call statement in a program is not the subroutine itself.

Setting up the Connector Pane



To use a VI as a subVI, you need to build a connector pane. The connector pane is a set of terminals that corresponds to the controls and indicators of that VI, similar to the parameter list of a function call in text-based programming languages. The connector pane defines the inputs and outputs you can wire to the VI so you can use it as a subVI.

Define connections by assigning a front panel control or indicator to each of the connector pane terminals. To define a connector pane, right-click the icon in the upper right corner of the front panel window and select **Show Connector** from the shortcut menu. The connector pane replaces the icon. Each rectangle on the connector pane represents a terminal. Use the rectangles to assign inputs and outputs. The number of terminals LabVIEW displays on the connector pane depends on the number of controls and indicators on the front panel.

The connector pane has, at most, 28 terminals. If your front panel contains more than 28 controls and indicators that you want to use programmatically, group some of them into a cluster and assign the cluster to a terminal on the connector pane.

Select a different terminal pattern for a VI by right-clicking the connector pane and selecting **Patterns** from the shortcut menu. Select a connector pane pattern with extra terminals. You can leave the extra terminals unconnected until you need them. This flexibility enables you to make changes with minimal effect on the hierarchy of the VIs.

If you create a group of subVIs that you use together often, give the subVIs a consistent connector pane with common inputs in the same location to help you remember where to locate each input. If you create a subVI that produces an output another subVI uses as the input, align the input and output connections to simplify the wiring patterns. Place the **error in** clusters on the lower left corner of the front panel and the **error out** clusters on the lower right corner.

Setting Required, Recommended, and Optional Inputs and Outputs

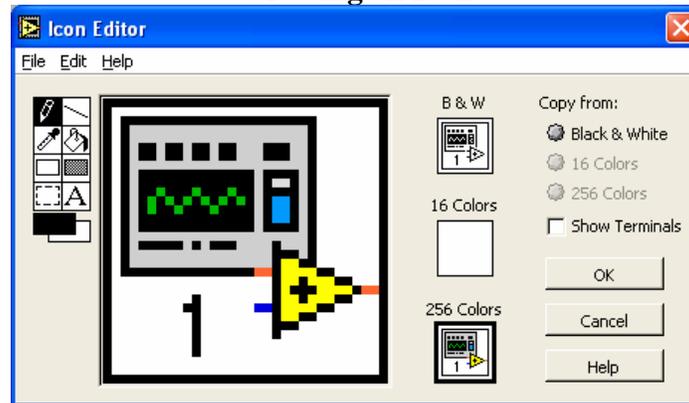
You can designate which inputs and outputs are required, recommended, and optional to prevent users from forgetting to wire subVI connections. Right-click a terminal in the connector pane and select **This Connection Is** from the shortcut menu. A checkmark indicates the terminal setting. Select **Required**, **Recommended**, or **Optional**.

For terminal inputs, required means that the block diagram on which the subVI is dropped will be broken if the required inputs are not wired. Required is not available for terminal outputs. For terminal inputs and outputs, recommended means that the block diagram on which the subVI is dropped can be run if the input is not wired but the **Warnings** dialog box will generate a warning that the input has not been wired. Optional means that the block diagram on which the subVI is dropped can be run and will not generate any warnings if the terminal input or output is not wired.

Inputs and outputs of VIs in vi.lib are already marked as **Required**, **Recommended**, or **Optional**. LabVIEW sets inputs and outputs of VIs you create to **Recommended** by default. Set a terminal setting to required only if the VI must have the input or output to run properly.

In the **Context Help** window, required connections are bold, recommended connections are plain text, and optional connections are dimmed if you have the **Detailed** view selected or do not appear if you have the **Simple** view selected.

Creating an Icon



Every VI displays an icon in the upper right corner of the front panel and block diagram windows. An icon is a graphical representation of a VI. It can contain text, images, or a combination of both. 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 have opened since launching LabVIEW. Create custom icons to replace the default icon by right-clicking the icon in the upper right corner of the front panel or block diagram and selecting **Edit Icon** from the shortcut menu or by double-clicking the icon in the upper right corner of the front panel.

You also can drag a graphic from anywhere in your file system and drop it in the upper right corner of the front panel or block diagram. LabVIEW converts the graphic to a 32 X 32 pixel icon.

Depending on the type of monitor you use, you can design a separate icon for monochrome, 16-color, and 256-color mode. LabVIEW uses the monochrome icon for printing unless you have a color printer.

Creating SubVIs from Sections of a VI

Convert a section of a VI into a subVI by using the Positioning tool to select the section of the block diagram you want to reuse and selecting **Edit>Create SubVI**. An icon for the new subVI replaces the selected section of the block diagram. LabVIEW creates controls and indicators for the new subVI and wires the subVI to the existing wires.

Creating a subVI from a selection is convenient but still requires careful planning to create a logical hierarchy of VIs. Consider which objects to include in the selection and avoid changing the functionality of the resulting VI.

Equations: Converting °C to °F

The formula for converting degrees Celsius to degrees Fahrenheit is as follows:

$$^{\circ}\text{F} = (1.8 * ^{\circ}\text{C}) + 32$$

For example, to convert a Celsius temperature of 100 degrees into degrees Fahrenheit, first multiply the Celsius temperature reading by 1.8 to get 180. Then add 32 to 180 and get 212 degrees Fahrenheit.

Slope of a Line

The formula for the slope of a line is as follows:

$$\text{Slope} = (Y2 - Y1) / (X2 - X1)$$

where (X1, Y1) and (X2, Y2) are points on the line.

Exercises:

1. Converting °C to °F

In this exercise, you will create a VI that can be used later as a sub-VI.

1. Launch LabVIEW on your PC. Click **New VI** to open a new front panel.
2. (Optional) Select **Window»Tile Left and Right** to display the front panel and block diagram side by side.
3. Creating a numeric digital control: You will use this control to enter the value for degrees Centigrade.
 - a. Select the digital control on the **Controls»Numeric** palette. If the **Controls** palette is not visible, right-click an open area on the front panel to display it.
 - b. Move the control to the front panel and click to place the control.
 - c. Type deg C inside the label and click outside the label or click the **Enter** button on the toolbar, shown at left. If you do not type the name immediately, LabVIEW uses a default label. You can edit a label at any time by using the Labeling tool, shown at left.
4. Creating a numeric digital indicator: You will use this indicator to display the value for degrees Fahrenheit.
 - a. Select the digital indicator on the **Controls»Numeric** palette.
 - b. Move the indicator to the front panel and click to place the indicator.
 - c. Type deg F inside the label and click outside the label or click the **Enter** button.

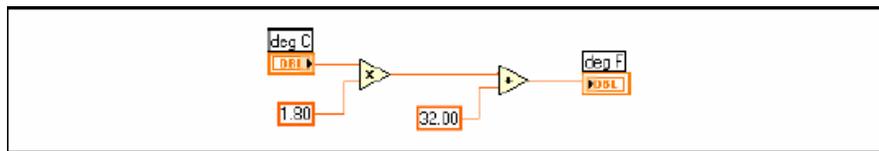


LabVIEW creates corresponding control and indicator terminals on the block diagram. The terminals represent the data type of the control or indicator. For example, a DBL terminal, shown at left, represents a double-precision, floating-point numeric control or indicator.

5. Display the block diagram by clicking it or by selecting **Window»Show Diagram**.

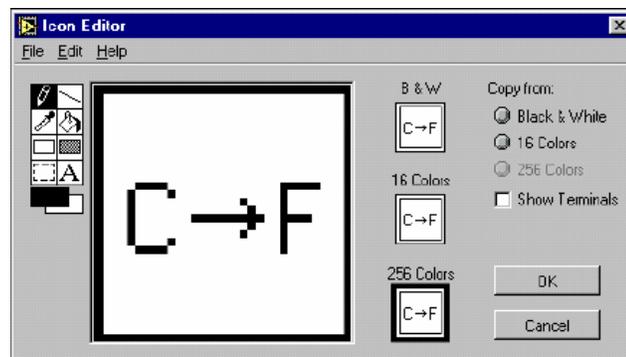
6. Select the Multiply and Add functions on the **Functions»Numeric** palette and place them on the block diagram. If the **Functions** palette is not visible, right-click an open area on the block diagram to display it.
7. Select the numeric constant on the **Functions»Numeric** palette and place two of them on the block diagram. When you first place the numeric constant, it is highlighted so you can type a value.
8. Type 1.8 in one constant and 32.0 in the other.

If you moved the constants before you typed a value, use the Labeling tool to enter the values.



9. Use the Wiring tool to wire the icons as shown in the block diagram.
 - To wire from one terminal to another, use the Wiring tool to click the first terminal, move the tool to the second terminal, and click the second terminal, as shown in the following illustration. You can start wiring at either terminal.
 - You can bend a wire by clicking to tack the wire down and moving the cursor in a perpendicular direction. Press the spacebar to toggle the wire direction.
 - To identify terminals on the nodes, right-click the Multiply and Add functions and select **Visible Items»Terminals** from the shortcut menu to display the connector pane. Return to the icons after wiring by right-clicking the functions and selecting **Visible Items»Terminals** from the shortcut menu to remove the checkmark.
 - When you move the Wiring tool over a terminal, the terminal area blinks, indicating that clicking will connect the wire to that terminal and a tip strip appears, listing the name of the terminal.
 - To cancel a wire you started, press the <Esc> key, right-click, or click the source terminal.
10. Display the front panel by clicking it or by selecting **Window»Show Panel**.
11. Save the VI because you will use this VI later in the course. Select **File»Save**. Type `Convert C to F.vi` in the dialog box. Click the **Save** button.
12. Enter a number in the digital control and run the VI.

- a. Use the Operating tool or the Labeling tool to double-click the digital control and type a new number.
 - b. Click the **Run** button to run the VI.
 - c. Try several different numbers and run the VI again.
13. Right-click the icon in the upper right corner of the front panel and select **Edit Icon** from the shortcut menu. The **Icon Editor** dialog box appears.
 14. Double-click the Select tool on the left side of the **Icon Editor** dialog box to select the default icon.
 15. Press the <Delete> key to remove the default icon.
 16. Double-click the Rectangle tool to redraw the border.
 17. Create the following icon.



- a. Use the Text tool to click the editing area.
- b. Type C and F.
- c. Double-click the Text tool and change the font to **Small Fonts**.
- d. Use the Pencil tool to create the arrow.

Note To draw horizontal or vertical straight lines, press the <Shift> key while you use the Pencil tool to drag the cursor.

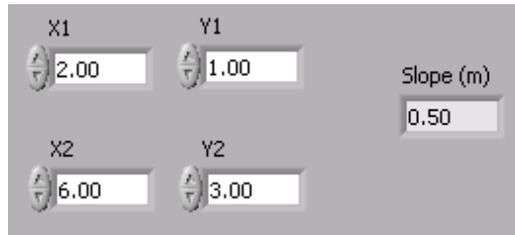
- e. Use the Select tool and the arrow keys to move the text and arrow you created.
- f. Select the **B&W** icon and select **256 Colors** in the **Copy from** field to create a black and white icon, which LabVIEW uses for printing unless you have a color printer.

- g. When the icon is complete, click the **OK** button to close the **Icon Editor** dialog box. The icon appears in the upper right corner of the front panel and block diagram.
18. Right-click the icon on the front panel and select **Show Connector** from the shortcut menu to define the connector pane terminal pattern. LabVIEW selects a connector pane pattern based on the number of controls and indicators on the front panel. For example, this front panel has two terminals, **deg C** and **deg F**, so LabVIEW selects a connector pane pattern with two terminals.
19. Assign the terminals to the digital control and digital indicator.
- Select **Help»Show Context Help** to display the **Context Help** window. View each connection in the **Context Help** window as you make it.
 - Click the left terminal in the connector pane. The tool automatically changes to the Wiring tool, and the terminal turns black.
 - Click the **deg C** control. The left terminal turns orange, and a marquee highlights the control.
 - Click an open area of the front panel. The marquee disappears, and the terminal changes to the data type color of the control to indicate that you connected the terminal.
 - Click the right terminal in the connector pane and click the **deg F** indicator. The right terminal turns orange.
 - Click an open area on the front panel. Both terminals are orange.
 - Move the cursor over the connector pane. The **Context Help** window shows that both terminals are connected to floating-point values.
20. Select **File»Save** to save the VI.
21. Select **File»Close** to close the VI.

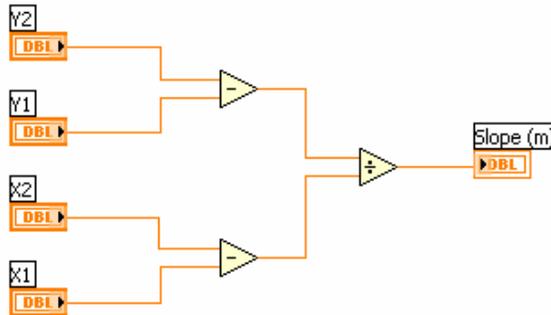
2. Calculating the Slope of a Line

Using the techniques you learned in Exercise 1, create a VI to calculate the slope of a line. Then, transform the code into a subVI using the **Edit>Create SubVI** method. Save the VI as `Slope.vi`. Save the subVI as `SlopeSub.vi`. The equation for slope is given in the Theory section.

Open a new VI and complete the front panel and block diagram as follows:

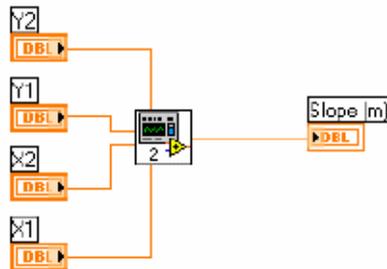


Front Panel



Block Diagram before subVI

After creating the subVI, your block diagram will resemble the one below.



Block Diagram with subVI

3. Calculator VI

Write a VI that adds, subtracts, multiplies, divides, and averages two input numbers and displays the results on the front panel. Save your VI as `Calculator.vi`.