Introduction

LabVIEW (short for Laboratory Virtual Instrumentation Engineering Workbench) is a graphical programming environment from National Instruments, which allows engineers and scientists to rapidly build “instruments” originally for computer-controlled data acquisition, automated testing and hardware interfacing.

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation often imitate physical instruments, such as oscilloscopes. A VI and its components are analogous to main programs and functions from text programming languages like C. A VI has both an interactive user interface – known as the front panel window, and the source code – written in the graphical (G) dataflow programming language on the block diagram window, as shown in Figure 1. Use Ctrl + E to switch between two windows.

1. Front Panel

A front panel houses various graphical objects ranging from simple buttons to complex graphs. Various options are available for changing the look and feel of the objects on the front panel to match the needs of any application.

2. Block Diagram

A block diagram represents some program logic that serves to modify data as it flows from sources to sinks. It can house a pipeline structure of sources, sinks, VIs (executable icons or
nodes), and structures wired together to define this logic. Most importantly, every data source or sink on the front panel has its corresponding analog source or sink on the block diagram. This representation allows the input values from users to be accessed by the block diagram. Likewise, new output values executed by the code in the diagram can be shown on the front panel.

3. Palettes

3.1 Controls Palette and Functions Palette

The Controls palette contains the controls and indicators used to create the Front Panel. The Functions palette contains the terminals (mapped from the controls and indicators on the Front Panel), SubVIs, functions, constants, structures and wires used to create the Block Diagram.

Figure 2. Controls and Functions Palettes
Users can access the *Controls palette* from the Front Panel window by selecting View » Controls Palette or by right clicking on any empty space in this window; Similarly, users access the *Functions palette* from the Block Diagram window by selecting “View » Functions Palette” or by right clicking on any empty space in this window. The contents of both palettes are broken into various categories. Users can also expose all or some of these categories by selecting “Customize” on the corresponding palette.

### 3.2 Tools Palette

In LabVIEW, the mouse is the main way to interact with the programming environment. That said, the mouse pointer must be able to accomplish many different tasks, such as selecting, wiring, highlighting text, and so on. The operating mode of the cursor corresponds to the icon of the tool selected on the Tools Palette. Users can select “View » Tools Palette” to display the Tools Palette.

The top item is the Automatic Tools Selection button. When users enable this, LabVIEW automatically chooses a tool based on the current location of the mouse. Users can disable this automatic tool selection by deselecting this button or by selecting another item in the palette.

![Figure 3. Tools Palette](image)

### 4. Controls and Indicators

In LabVIEW, the most common form of a *data source* is a Control. This element appears as some type of graphical element on the Front Panel of a VI that can *receive input from* a user or even another VI. Every control has an associated data type that determines what kind of data flows from it on the block diagram, and every data type also has an associated color shown on the block diagram.

The most common form of a *data sink* is an Indicator. This element appears as some type of graphical element on the Front Panel of a VI that can *display output to* a user or even another VI. Similar to a Control, every Indicator also has an associated data type that determines what kind of data can be written to it on the block diagram. Some commonly used Controls and Indicators are shown in Figure 3. We will discuss Array and Cluster in following lessons.
4.1 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. Objects such as meters and dials also represent numeric data.

4.2 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. Boolean controls and indicators are used to enter and display Boolean values. Boolean objects simulate switches, push buttons, and LEDs.

4.3 String Controls and Indicators

The string data type is a sequence of ASCII characters. String controls are used to receive text from the user such as a password or user name. String indicators are used to display text to the user. The most common string objects are tables and text entry boxes.

Exercises

1. Get familiar with the LabVIEW environment: panel and diagram windows, pop-up menus (Change to Control/Indicator/Constant, etc), pull-down menus (Windows, Tools>>Options, View, etc), controls palettes (Modern, Express), functions palette (Programming, Express), tools palette, LabVIEW help options.

2. Please follow the following steps to create a non-trivial VI program based on a template VI program. This exercise is derived from the “Getting Started with LabVIEW 2012” manual from NI.
Launching LabVIEW

1. Launch LabVIEW. Select File >> New… to display the New dialog box as the following.

2. From the Create New List, select VI>>From Template>>Tutorial (Getting Started)>>Generate and Display, as shown in the following.
3. Click the **OK** button to create a VI from the template. LabVIEW displays two windows: front panel window and block diagram window.

4. Examine the Front Panel (user interface) window. Note: if the front panel is not visible, you can display it by selecting **Window>>Show Front Panel**, or by pressing <Ctrl-E> keys.

5. Select **Window>>Show Block Diagram** and examine the Block Diagram of the VI. It includes Vis and structures that control the front panel objects.

6. On the Front Panel toolbar, click the **Run** button. You also can press <Ctrl-R> keys to run a VI.

7. Stop the VI by clicking the STOP button on the front panel.

**Adding a control on the Front Panel**

Front panel controls emulate the input mechanisms on a physical instrument and supply data to the block diagram of the VI. We now add a knob into the front panel.

1. Open the **Controls Palette** by selecting **View>>Controls Palette** if it is invisible.

2. Select the **Express** palette. Move the cursor over the icons on the **Numeric Controls** subpalette. Click it to open this subpalette.

3. Click the **Knob** control and add it to the front panel to the left of the waveform graph, and inside the while loop structure (the gray loop).

4. Select **File>>Save As** and save the modified VI.

**Changing a Signal Type**

1. Display the block diagram by pressing <Ctrl-E> keys. Locate the **Simulate Signal Express VI**. This library VI simulates a signal based on the configuration that you specify. Right-click this VI and select **Properties** from the shortcut menu to display the dialog box as shown in the following.
2. Select **Sawtooth** from the signal type pull-down menu. Click **OK** button to save the current configuration and close the dialog box.

3. Move the cursor over the down arrows at the bottom of the **Simulate Signal** Express VI. The down arrows indicate you can reveal hidden inputs and outputs by extending the border of the Express VI.

   ![Simulate Signal Express VI](image)

4. Drag the down arrows to add two rows. When releasing the border, the **Amplitude** input appears as shown above.

**Wiring Objects on the block diagram**

1. On the block diagram, move the cursor over the right arrow on the **Knob** terminal.

   The cursor appears as a wire spool tool. Then move to the **Amplitude** input of the Simulate Signal Express VI and wire these two blocks together. This sets up the data flow between these two Vis.

2. Select **File>>Save** to save the VI and start to run.

**Modifying a Signal**

1. In the block diagram, use the Positioning tool to click the wire that connects the Simulate Signal Express VI to the Waveform Graph terminal. As shown below.

   ![Waveform Graph](image)

2. Press the **<Delete>** key to delete this wire.

3. Open the **Functions palette >> Express palette** as shown below.
4. On the **Arithmetic & Comparison** palette, select the **Formula Express VI**, place it between Simulate Signal Express VI and the Waveform Graph terminal. The configure Formula dialog box appears automatically when you place this Express VI on the block diagram.

![Configure Formula Dialog Box](image)

5. Change the text in the Label column of the dialog box option you read about, as below, from \(x_1\) to \(sawtooth\) to indicate the input value to the Formula Express VI. When you click in the Formula text box at the top of the **Configure Formula dialog box**, the text changes to match the label you entered.

![Formula Text Box](image)

6. Define the value of the scaling factor by entering \(*10\) after **Sawtooth** in the Formula text box as shown below.

![Scaled Formula](image)

7. Click the OK button to save the current configuration and close the Configuration Formula dialog box.

8. Move the cursor over the arrow on the Sawtooth output of the Simulate Signal Express VI.
9. When the wiring tool appears, click this arrow and then click the arrow on the Sawtooth input of the Formula Express VI, to wire the two objects together.

10. Use the Wiring tool to wire the Result output of the Formula Express VI to the Waveform Graph terminal as below.

Display two signals on a graph
1. In the block diagram, move the cursor over the arrow on the Sawtooth output of the Simulate Signal Express VI.
2. Use the Wiring tool to wire the Sawtooth output to the Waveform Graph input. The Merge signals function, which takes the two separate signals and combines them so that both can display on the same graph, appears automatically where the two wires connect.
3. Save the VI and return to the front panel, run the VI, and turn the knob control.

Customizing a Knob control
1. Right-click the front panel knob and select Properties from the shortcut menu to display the Knob Properties dialog box.
2. Click the **Appearance** tab to display the page. In the Label selection on this page, delete the label knob, and enter **Amplitude** in the text box.

3. Click the **Scale** tab. In the Scale style section, check the **Show color ramp** checkbox.

4. Click the **OK** button to save the current configuration and close the **Knob Properties** dialog box.

5. Save the VI.

**Customizing a Waveform graph**

1. In the front panel window, move the cursor over the top of the plot legend on the waveform graph. When a double-headed arrow appears, click and drag the border of the plot legend to add one item to the legend. When you release the mouse button, the second plot name appears; change the label name.

2. Right-click the waveform graph and select **Properties** from the shortcut menu to display the **Graph Properties** dialog box.
   
   On the **Plots** page, select **Sawtooth** from the top pull-down menu. In the **colors** section, click the **Line** color box to display the color picker. Select a new line color; then select **Sawtooth(Formula Result)** from the top pull-down menu; Check the **Do not use waveform names for plot names** checkbox. This allows editing the labels on the graph; In the **Name** text box, delete the current label and change the name of this plot to **Scaled Sawtooth**.

3. Click the **OK** button to save the current configuration and close the **Graph Properties** dialog box.

4. Save and Run the VI.