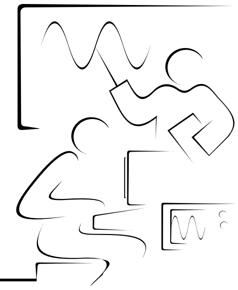


Lesson 1

Introduction to LabVIEW

Exercises **Page**

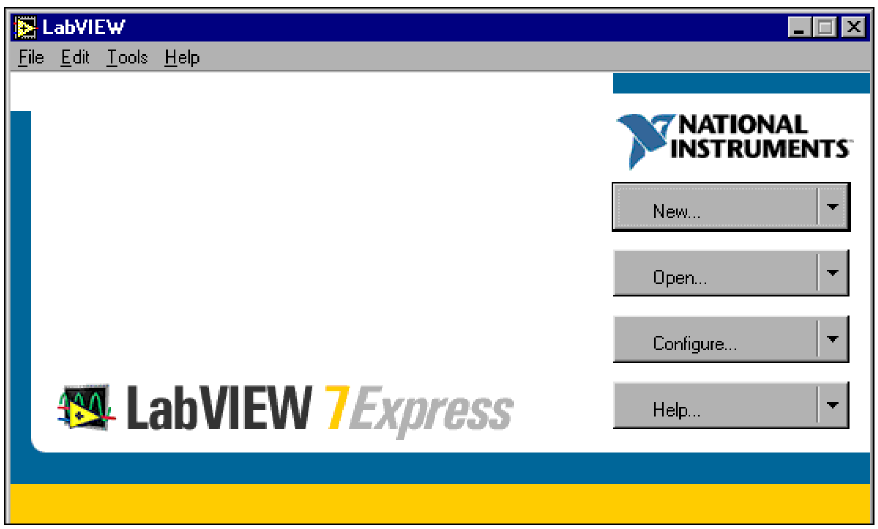
Exercise 1-1: Acquiring a Signal VI

Objective: Explore the LabVIEW environment by creating a VI that generates a signal and displays it on the front panel.

In the following exercise, you will build a VI that generates a signal and displays that signal in a graph. LabVIEW provides templates containing information from which you can build a VI. These templates help you get started with LabVIEW.

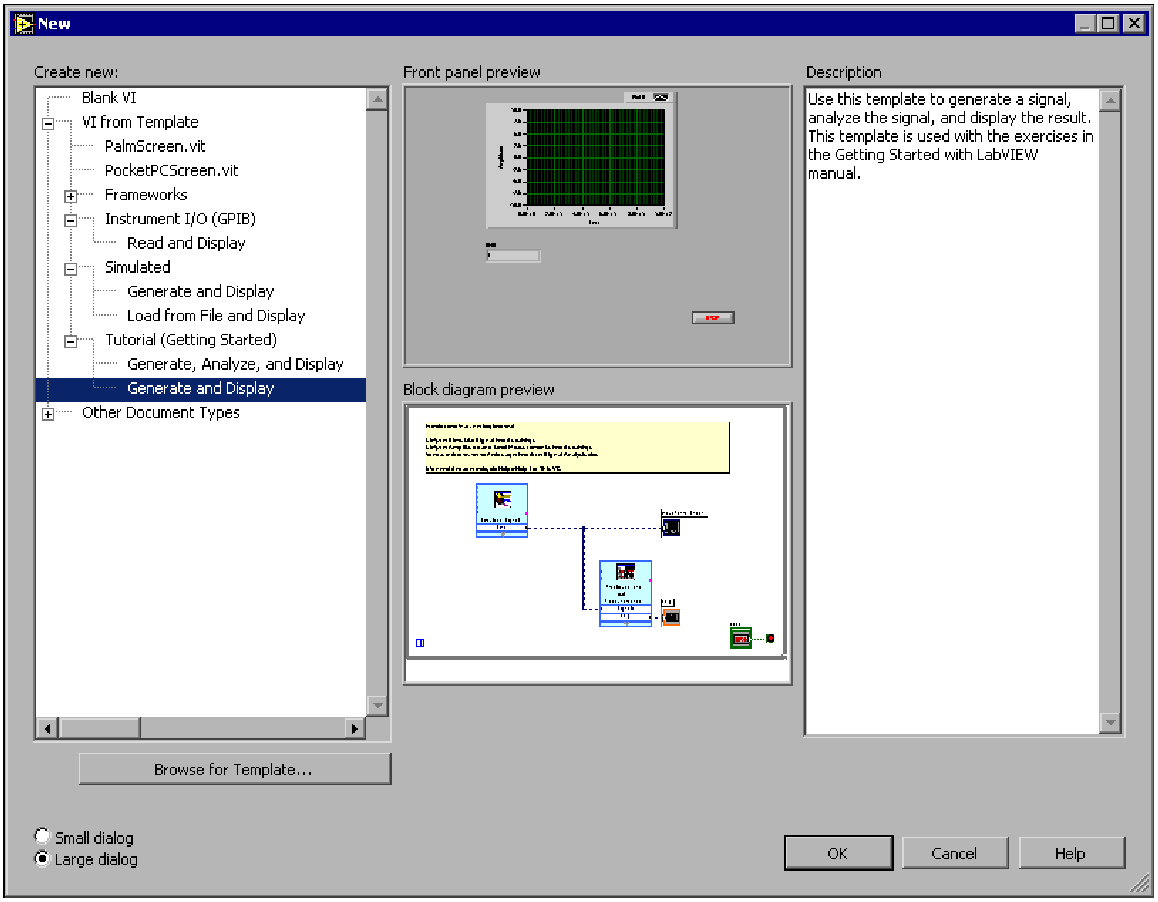
Complete the following steps to create a VI that generates a signal and displays it on the front panel.

1. Launch LabVIEW.
2. In the **LabVIEW** dialog box that appears, shown in the following figure, click the **New** button to display the **New** dialog box.



3. Select **VI from Template»Tutorial (Getting Started)»Generate and Display** in the **Create new** list. This template VI generates and displays a signal.

   Notice that previews of the template VI appear in the **Front panel preview** and the **Block diagram preview** sections. The following figure shows the **New** dialog box and the Generate and Display template VI.



4. Click the **OK** button to open the template. You also can double-click the name of the template VI in the **Create new** list to open the template.
5. Examine the front panel of the VI.

   The user interface, or front panel, appears with a gray background and includes controls and indicators. The title bar of the front panel indicates that this window is the front panel for the Generate and Display VI.

Note If the front panel is not visible, you can display the front panel by selecting **Window»Show Front Panel**.

6. Examine the block diagram of the VI.

   The block diagram appears with a white background and includes VIs and structures that control the front panel objects. The title bar of the block diagram indicates that this window is the block diagram for the Generate and Display VI.

Note If the block diagram is not visible, you can display the block diagram by selecting **Window»Show Block Diagram**.

7. On the front panel toolbar, click the **Run** button, shown at left.

   Notice that a sine wave appears on the graph.

8. Stop the VI by clicking the **STOP** button, shown at left, on the front panel.

Adding a Control to the Front Panel

Controls on the front panel simulate the input devices on a physical instrument and supply data to the block diagram of the VI. Many physical instruments have knobs you can turn to change an input value. Complete the following steps to add a knob control to the front panel.

Tip Throughout these exercises, you can undo recent edits by selecting **Edit»Undo** or pressing the <Ctrl-Z> keys.

1. If the **Controls** palette is not visible on the front panel, select **Window» Show Controls Palette** to display it.
2. Move the cursor over the icons on the **Controls** palette to locate the **Numeric Controls** palette.

   Notice that when you move the cursor over icons on the **Controls** palette, the name of that subpalette appears in the gray space above all the icons on the palette. When you idle the cursor over any icon on any palette, the full name of the subpalette, control, or indicator appears.
3. Click the **Numeric Controls** icon to access the **Numeric Controls** palette.
4. Select the knob control on the **Numeric Controls** palette and place it on the front panel to the left of the waveform graph.

   You will use this knob in a later exercise to control the amplitude of a signal.
5. Select **File»Save As** and save this VI as `Acquiring a Signal.vi` in the `C:\Exercises\LV Basics I` directory.

Note Save all the VIs you edit or create in this course in the `C:\Exercises\LV Basics I` directory.

Changing the Signal Type

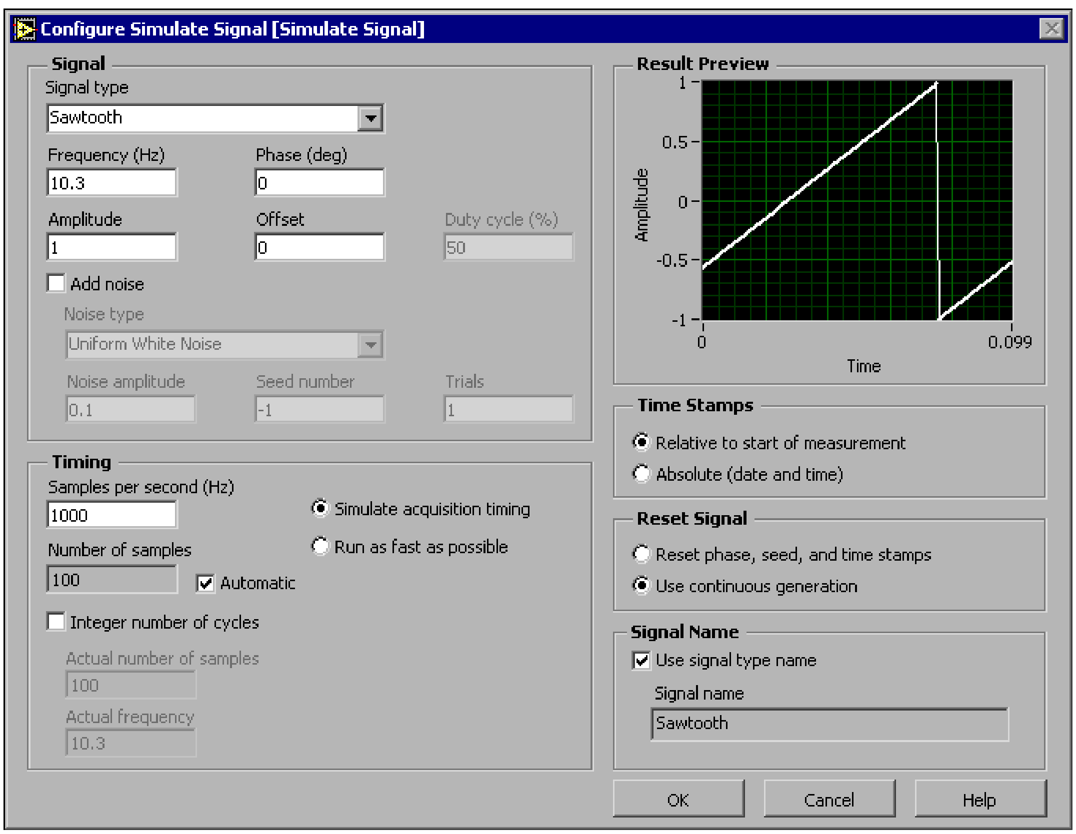
The block diagram has a blue icon labeled **Simulate Signal**. This icon represents the Simulate Signal Express VI. The Simulate Signal Express VI simulates a sine wave by default. Complete the following steps to change this signal to a sawtooth wave.

1. Display the block diagram by selecting **Window»Show Block Diagram** or by clicking the block diagram.

   Notice the Simulate Signal Express VI, shown at left. An Express VI is a component of the block diagram that you can configure to perform common measurement tasks. The Simulate Signal Express VI simulates a signal based on the configuration that you specify.

2. Right-click the Simulate Signal Express VI and select **Properties** from the shortcut menu to display the **Configure Simulate Signal** dialog box.
3. Select **Sawtooth** from the **Signal type** pull-down menu.

   Notice that the waveform on the graph in the **Result Preview** section changes to a sawtooth wave. The **Configure Simulate Signal** dialog box should appear similar to the following figure.



4. Click the **OK** button to apply the current configuration and close the **Configure Simulate Signal** dialog box.

5. Move the cursor over the down arrows at the bottom of the Simulate Signal Express VI.

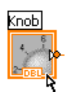
6. When a double-headed arrow appears, shown at left, click and drag the border of the Express VI until the **Amplitude** input appears.

   Notice how you expanded the Simulate Signal Express VI to display a new input. Because the **Amplitude** input appears on the block diagram, you can configure the amplitude of the sawtooth wave on the block diagram.

   In the previous figure, notice how **Amplitude** is an option in the **Configure Simulate Signal** dialog box. When inputs, such as **Amplitude**, appear on the block diagram and in the configuration dialog box, you can configure the inputs in either location.

Wiring Objects on the Block Diagram

To use the knob control to change the amplitude of the signal, you must connect the two objects on the block diagram. Complete the following steps to wire the knob to the **Amplitude** input on the Simulate Signal Express VI.



1. Move the cursor over the **Knob** terminal, shown at left, until the Positioning tool appears.

   Notice how the cursor becomes an arrow, or the Positioning tool, shown at left. Use the Positioning tool to select, position, and resize objects.

2. Click the **Knob** terminal to select it, then drag the terminal to the left of the Simulate Signal Express VI. Make sure the **Knob** terminal is inside the loop, shown at left.

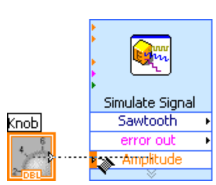
   The terminals are representations of front panel controls and indicators. Terminals are entry and exit ports that exchange information between the front panel and block diagram.

3. Deselect the **Knob** terminal by clicking a blank space on the block diagram.

4. Move the cursor over the arrow of the **Knob** terminal, shown at left.

   Notice how the cursor becomes a wire spool, or the Wiring tool, shown at left. Use the Wiring tool to wire objects together on the block diagram.

Note The cursor does not switch to another tool while an object is selected.



5. When the Wiring tool appears, click the arrow and then click the **Amplitude** input of the Simulate Signal Express VI, shown at left, to wire the two objects together.

   Notice that a wire appears and connects the two objects. Data flows along this wire from the terminal to the Express VI.

6. Select **File»Save** to save this VI.

Running the VI

Running a VI executes your solution. Complete the following steps to run the Acquiring a Signal VI.

1. Display the front panel by selecting **Window»Show Front Panel** or by clicking the front panel.

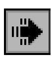
Tip Press the <Ctrl-E> keys to switch from the front panel to the block diagram or from the block diagram to the front panel.

2. Click the **Run** button.
3. Move the cursor over the knob control.

Notice how the cursor becomes a hand, or the Operating tool, shown at left. Use the Operating tool to change the value of a control or select the text within a control.

4. Using the Operating tool, turn the knob to adjust the amplitude of the sawtooth wave.

Notice how the amplitude of the sawtooth wave changes as you turn the knob. Also notice that the y-axis on the graph autoscales to account for the change in amplitude.



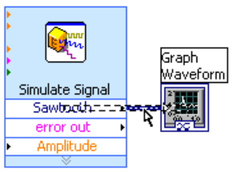
To indicate that the VI is running, the **Run** button changes to a darkened arrow, shown at left. You cannot edit the front panel or block diagram while the VI runs.

5. Click the **STOP** button, shown at left, to stop the VI.

Note Although the **Abort Execution** button  looks like a stop button, the **Abort Execution** button does not always properly close the VI. National Instruments recommends stopping your VIs using the **STOP** button on the front panel. Use the **Abort Execution** button only when errors prevent you from terminating the application using the **STOP** button.

Modifying the Signal

Complete the following steps to add scaling to the signal and display the results in the graph on the front panel.



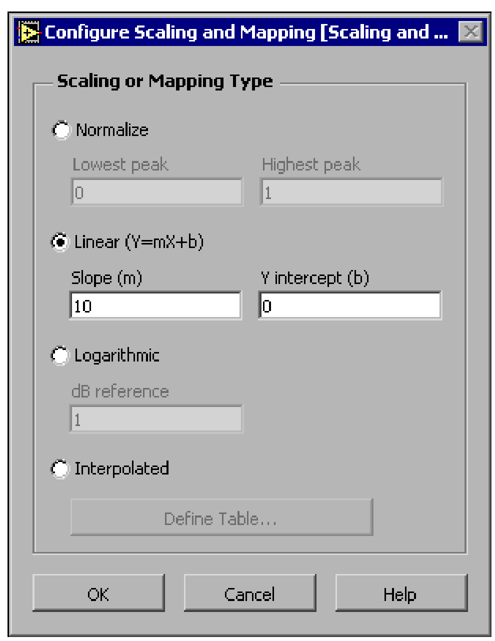
1. On the block diagram, use the Positioning tool to double-click the wire that connects the Simulate Signal Express VI to the **Waveform Graph** terminal, shown at left.
2. Press the <Delete> key to delete this wire.
3. If the **Functions** palette is not visible on the block diagram, select **Window»Show Functions Palette** to display it.

4. Select the Scaling and Mapping Express VI, shown at left, on the **Arithmetic & Comparison** palette and place it on the block diagram inside the loop between the Simulate Signal Express VI and the **Waveform Graph** terminal. If there is no room between the Express VI and the terminal, move the **Waveform Graph** terminal to the right.

   Notice that the **Configure Scaling and Mapping** dialog box automatically opens when you place the Express VI on the block diagram.

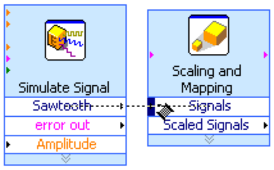
5. Define the value of the scaling factor by entering `10` in the **Slope (m)** text box.

   The **Configure Scaling and Mapping** dialog box should appear similar to the following figure.



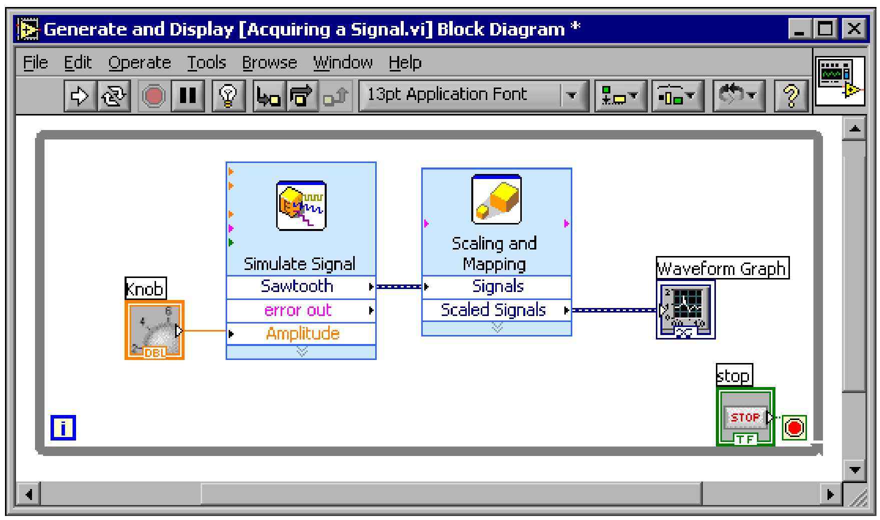
6. Click the **OK** button to apply the current configuration and close the **Configure Scaling and Mapping** dialog box.

7. Move the cursor over the arrow on the **Sawtooth** output of the Simulate Signal Express VI.



8. When the Wiring tool appears, click the arrow and then click the arrow on the **Signals** input of the Scaling and Mapping Express VI, shown at left, to wire the two objects together.
9. Using the Wiring tool, wire the **Scaled Signals** output of the Scaling and Mapping Express VI to the **Waveform Graph** terminal.

   Notice the wires connecting the Express VIs and terminals. The arrows on the Express VIs and terminals indicate the direction that the data flows along these wires. The block diagram should appear similar to the following figure.

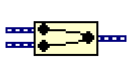


10. Select **File»Save** to save this VI.

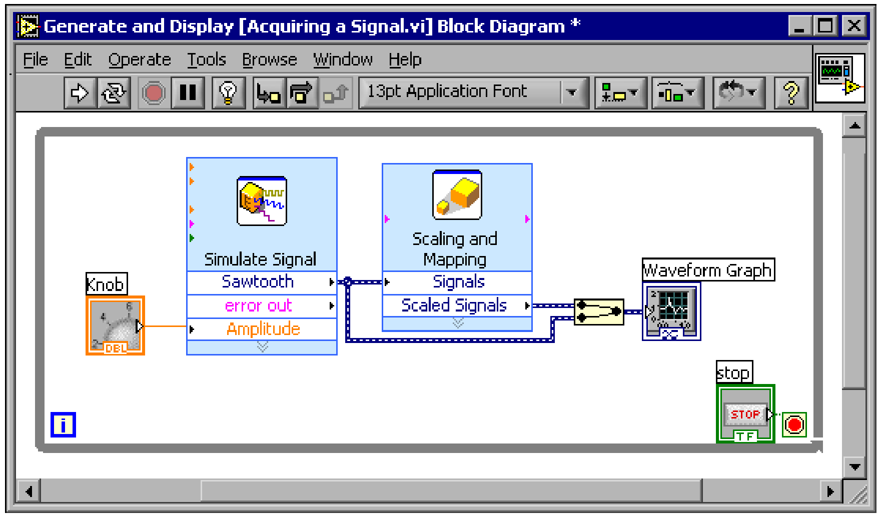
Displaying Two Signals on the Graph

To compare the signal generated by the Simulate Signal Express VI and the signal modified by the Scaling and Mapping Express VI on the same graph, use the Merge Signals function. Complete the following steps to display two signals on the same graph.

1. Move the cursor over the arrow on the **Sawtooth** output of the Simulate Signal Express VI.
2. Using the Wiring tool, wire the **Sawtooth** output to the **Waveform Graph** terminal.



   The Merge Signals function, shown at left, appears where the two wires connect. This function takes the two separate signals and combines them so that both can be displayed on the same graph. The block diagram should appear similar to the following figure.



3. Select **File»Save** to save this VI. You also can press the <Ctrl-S> keys to save a VI.
4. Return to the front panel, run the VI, and turn the knob control.

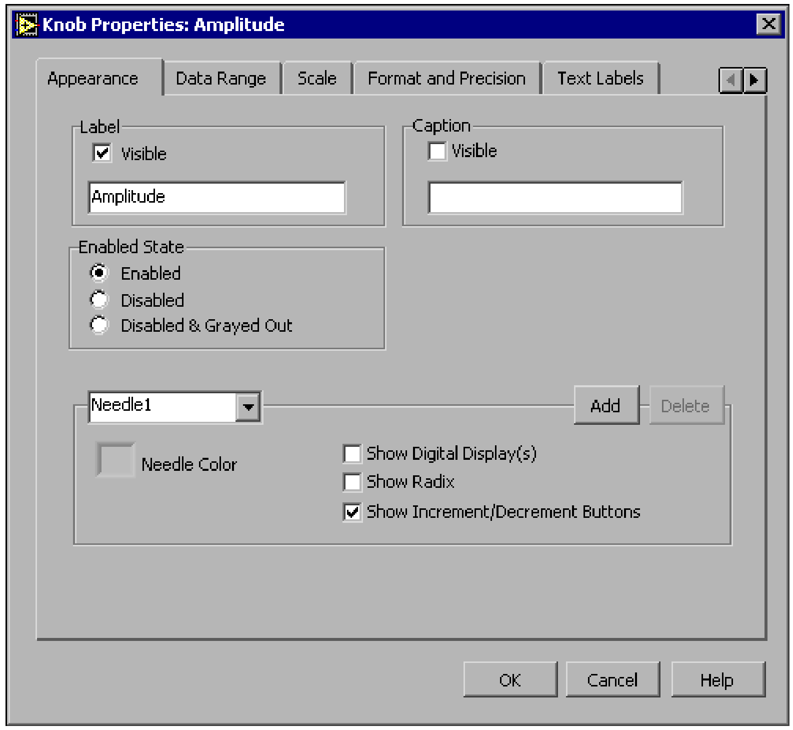
   Notice that the graph plots the sawtooth wave and the scaled signal. Also notice that the maximum value on the y-axis automatically changes to be 10 times the knob value. This scaling occurs because you set the slope to 10 in the Scaling and Mapping Express VI.
5. Click the **STOP** button.

Customizing the Knob

The knob control changes the amplitude of the sawtooth wave so labeling it **Amplitude** accurately describes the function of the knob. Complete the following steps to customize the appearance of a control on the front panel.

1. Right-click the knob and select **Properties** from the shortcut menu to display the **Knob Properties** dialog box.
2. In the **Label** section on the **Appearance** tab, delete the label **`Knob`**, and type `Amplitude` in the text box.

   The **Knob Properties** dialog box should appear similar to the following figure.



3. Click the **Scale** tab and in the **Scale Range** section, change the maximum value to `5.0`.

   Notice how the knob on the front panel instantly updates to reflect these changes.

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

5. Save this VI.

Tip As you build VIs, you can experiment with different properties and configurations. You also can add and delete objects. Remember you can undo recent edits by selecting **Edit»Undo** or pressing the <Ctrl-Z> keys.

6. Experiment with other properties of the knob by using the **Knob Properties** dialog box. For example, try changing the colors for the **Marker Text Color** by clicking the color box located on the **Scale** tab.

7. Click the **Cancel** button to avoid applying the changes you made while experimenting. If you want to keep the changes you made, click the **OK** button.

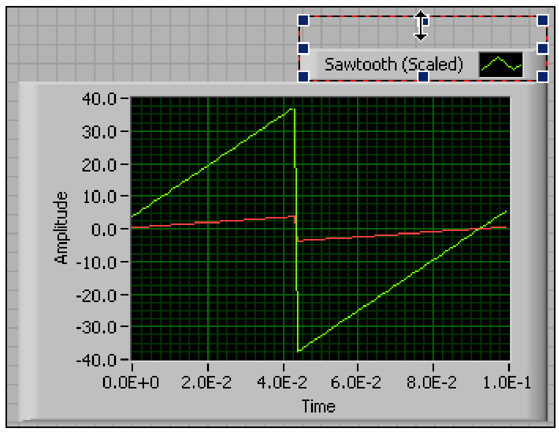
Customizing the Waveform Graph

The waveform graph indicator displays the two signals. To indicate which plot is the scaled signal and which is the simulated signal, you customize the plots. Complete the following steps to customize the appearance of an indicator on the front panel.

1. Move the cursor over the top of the plot legend on the waveform graph.

   Notice that while there are two plots on the graph, the plot legend displays only one plot.

2. When a double-headed arrow appears, shown in the following figure, click and drag the border of the plot legend until the second plot name appears.



3. Right-click the waveform graph and select **Properties** from the shortcut menu to display the **Graph Properties** dialog box.

4. On the **Plots** tab, select **Sawtooth** from the pull-down menu. Click the **Line Color** color box to display the color picker. Select a new line color.

5. Select **Sawtooth (Scaled)** from the pull-down menu.

6. Place a checkmark in the **Don't use waveform names for plot names** checkbox.

7. In the **Name** text box, delete the current label and change the name of this plot to `Scaled Sawtooth`.

8. Click the **OK** button to apply the current configuration and close the **Graph Properties** dialog box.

   Notice how the plot color on the front panel changes.

9. Experiment with other properties of the graph by using the **Graph Properties** dialog box. For example, try disabling the autoscale feature located on the **Scales** tab.

10. Click the **Cancel** button to avoid applying the changes you made while experimenting. If you want to keep the changes you made, click the **OK** button.
11. Save and close the VI.

End of Exercise 1-1

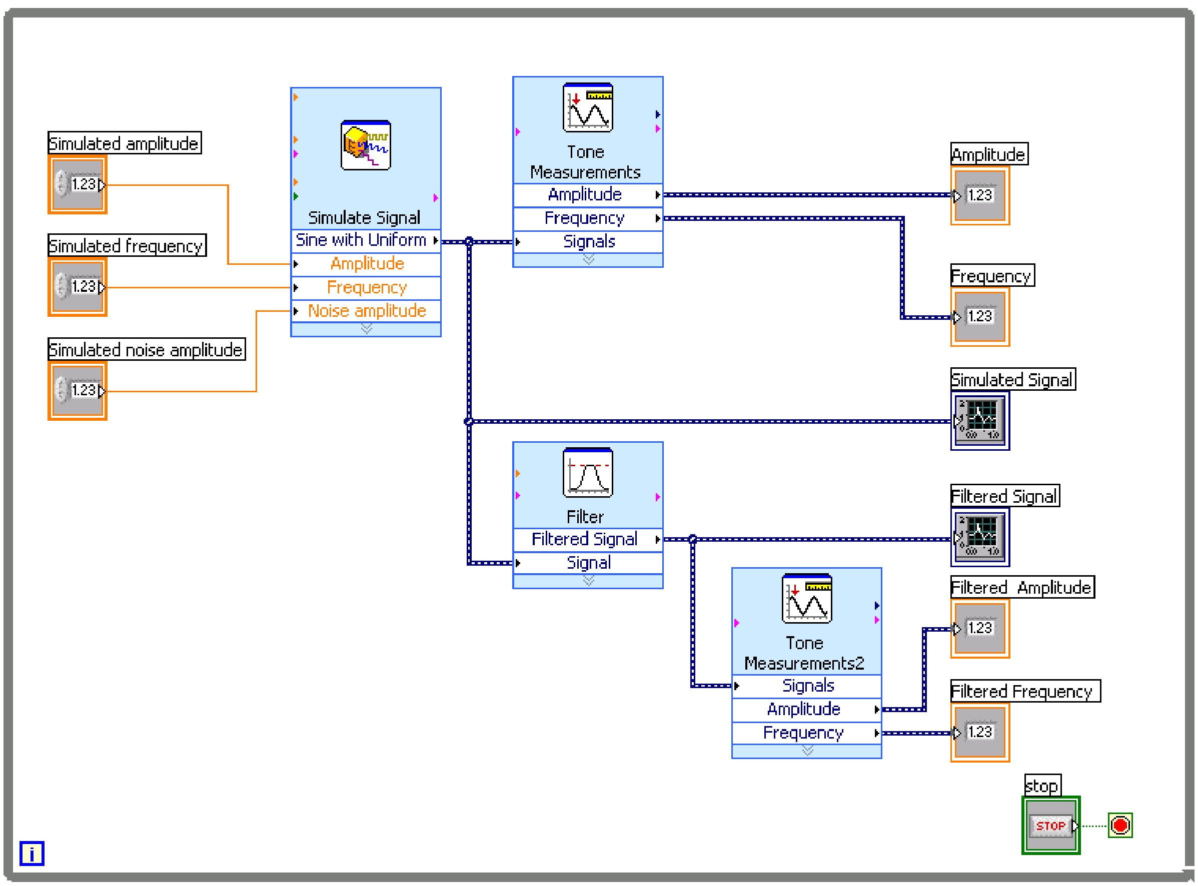
Exercise 1-2: Express Filter VI

Objective: Explore the data flow of an example VI.

Complete the following steps to use the NI Example Finder to search for an example VI that generates a sine wave with a frequency of 10 Hz and an amplitude of 10 volts with white noise of 1 volts of amplitude and applies a filter.

1. Select **Help»Find Examples** to open the NI Example Finder.
2. Click the **Search** tab and type `filter` in the **Type in the word(s) to search for** text box.

   Notice that this word choice reflects what you want this Express VI to do–filter a signal.
3. Select `filter` to display the example VIs that include filter in the title.
4. Find the example VI called `Express Filter.vi` and double-click to open it.
5. Open the block diagram of the VI, shown in the following figure.



6. Click the **Highlight Execution** button, shown at left, on the toolbar to slow down the execution of the program so you can observe the execution order on the block diagram.

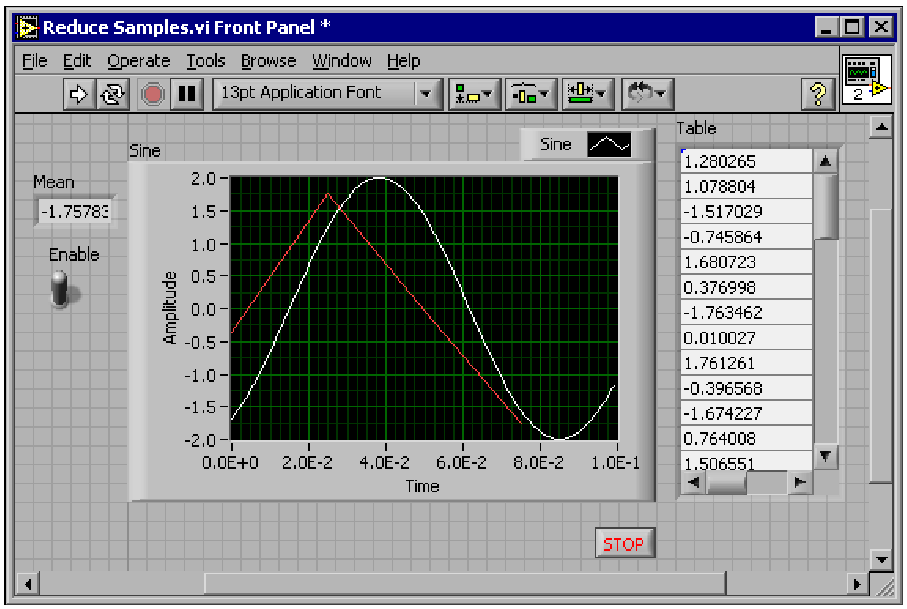
7. Click the **Run** button.
8. Observe the block diagram. Notice the flow of data on the block diagram. For example, notice that the Tone Measurements2 Express VI cannot output data until it receives data from Filter.
9. Close the VI. Do not save changes.

End of Exercise 1-2

Exercise 1-3: Reduce Samples VI

Objective: Use the LabVIEW documentation resources to build a VI that generates a signal, reduces the number of samples in the signal, and displays the resulting data in a table on the front panel.

In the following exercises, you will open a blank VI and add Express VIs and structures to the block diagram to build a new VI. When you complete the exercise, the front panel of the VI will appear similar to the following figure.

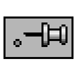


Opening a Blank VI

If no template is available for the task you want to create, you can start with a blank VI and add Express VIs to accomplish the specific task. Complete the following steps to open a blank VI.

1. In the **LabVIEW** dialog box, click the arrow on the **New** button and select **Blank VI** from the shortcut menu or press the <Ctrl-N> keys to open a blank VI.

   Notice that a blank front panel and block diagram appear.



2. If the **Functions** palette is not visible, right-click any blank space on the block diagram to display the **Functions** palette. Click the thumbtack, shown at left, in the upper left corner of the **Functions** palette to place the palette on the screen.

Note You can right-click a blank space on the block diagram or the front panel to display the **Functions** or **Controls** palettes.

Adding an Express VI that Simulates a Signal

Complete the following steps to find the Express VI you want to use and then add it to the block diagram.

1. If the **Context Help** window is not visible, press the <Ctrl-H> keys to open the **Context Help** window. You also can press the **Show Context Help Window** button, shown at left, to open the **Context Help** window.

2. Select the **Input** palette on the **Functions** palette and move the cursor over the Express VIs on the **Input** palette.

   Notice that the **Context Help** window displays information about the function of each Express VI.
3. From the information provided in the **Context Help** window, find the Express VI that can simulate a sine wave signal.
4. Select the Express VI and place it on the block diagram. The **Configure Simulate Signal** dialog box appears.
5. Idle the cursor over the various options in the **Configure Simulate Signal** dialog box, such as **Frequency (Hz)**, **Amplitude**, and **Samples per second (Hz)**. Read the information that appears in the **Context Help** window.
6. Configure the Simulate Signal Express VI to generate a sine wave with a frequency of `10.7` and amplitude of `2`.
7. Notice how the signal displayed in the **Result Preview** window changes to reflect the configured sine wave.
8. Close the **Configure Simulate Signal** dialog box by clicking the **OK** button.
9. Move the cursor over the Simulate Signal Express VI and read the information that appears in the **Context Help** window.

   Notice that the **Context Help** window now displays the configuration of the Simulate Signal Express VI.
10. Save this VI as `Reduce Samples.vi` in the `C:\Exercises\LV Basics I` directory.

Modifying the Signal

Complete the following steps to use the *LabVIEW Help* to search for the Express VI that reduces the number of samples in a signal.

1. Select **Help»VI, Function, & How-To Help** to open the *LabVIEW Help*.
2. Click the **Search** tab and type `sample compression` in the **Type in the word(s) to search for** text box.

   Notice that this word choice reflects what you want this Express VI to do—compress, or reduce, the number of samples in a signal.

3. To begin the search, press the <Enter> key or click the **List Topics** button.
4. Double-click the **Sample Compression** topic to display the topic that describes the Sample Compression Express VI.
5. After you read the description of the Express VI, click the **Place on the block diagram** button to select the Express VI.
6. Move the cursor to the block diagram.

   Notice how LabVIEW attaches the Sample Compression Express VI to the cursor.
7. Place the Sample Compression Express VI on the block diagram to the right of the Simulate Signal Express VI.
8. Configure the Sample Compression Express VI to reduce the signal by a factor of 25 using the mean of these values.
9. Close the **Configure Sample Compression** dialog box.
10. Using the Wiring tool, wire the **Sine** output in the Simulate Signal Express VI to the **Signals** input in the Sample Compression Express VI.

Customizing the Front Panel

In a previous exercise, you added controls and indicators to the front panel using the **Controls** palette. You also can add controls and indicators from the block diagram. Complete the following steps to create controls and indicators.

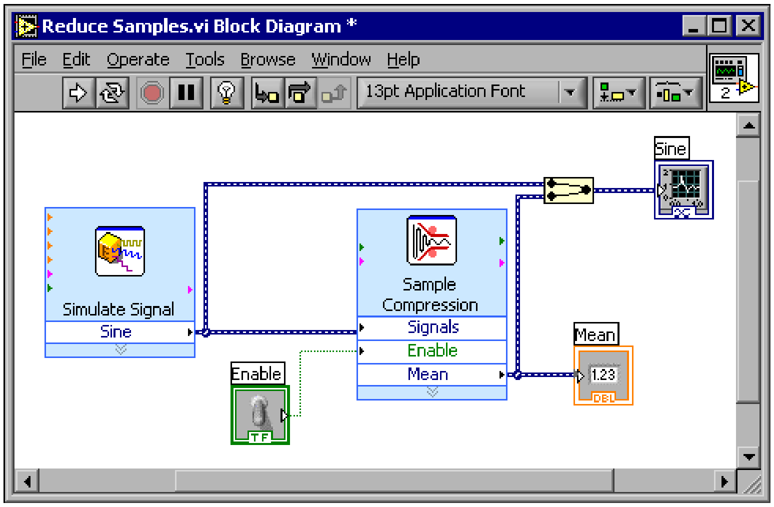
1. Right-click the **Mean** output in the Sample Compression Express VI and select **Create»Numeric Indicator** from the shortcut menu to create a numeric indicator.
2. Right-click the **Mean** output of the Sample Compression Express VI and select **Insert Input/Output** from the shortcut menu to insert the **Enable** input.
3. Right-click the **Enable** input and select **Create»Control** from the shortcut menu to create the **Enable** switch.
4. Right-click the wire linking the **Sine** output in the Simulate Signal Express VI to the **Signals** input in the Signal Compression Express VI and select **Create»Graph Indicator** from the shortcut menu.

   Notice that you can create controls and indicators from the block diagram. When you create controls and indicators using this method, LabVIEW automatically creates terminals that are labeled and formatted correctly.
5. Using the Wiring tool, wire the **Mean** output in the Sample Compression Express VI to the **Sine** terminal.

   Notice that the Merge Signals function appears.

6. Arrange the objects on the block diagram so that they appear similar to the following figure.

Tip You can right-click any wire and select **Clean Up Wire** from the shortcut menu to automatically route an existing wire.



7. Display the front panel.

   Notice that the controls and indicators you added automatically appear on the front panel with labels that correspond to their function.

8. Save this VI.

Configuring the VI to Run Continuously Until the User Stops It

In the current state, the VI runs once, generates one signal, then stops executing. To run the VI until a condition is met, you can add a While Loop to the block diagram. Complete the following steps to add a While Loop.

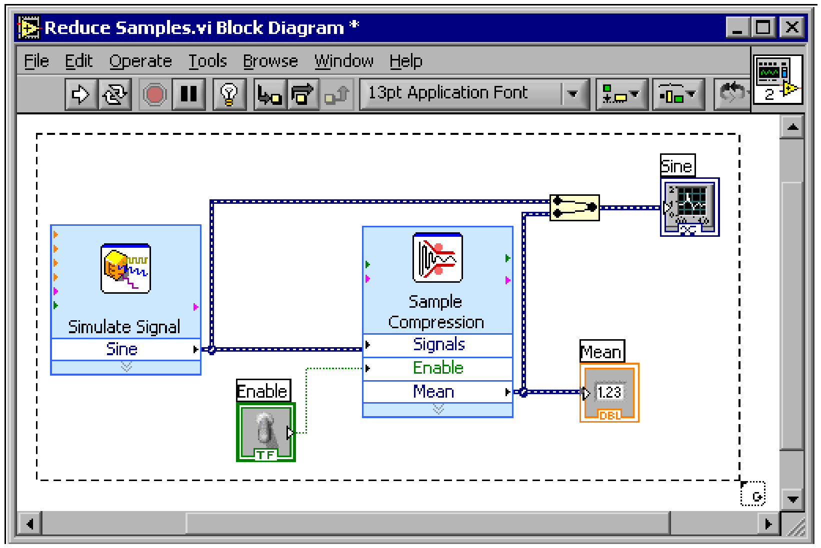
1. Display the front panel and run the VI.

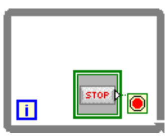
   Notice how the VI runs once and then stops. Also notice how there is no **STOP** button.

2. Display the block diagram and select the While Loop on the **Functions» Execution Control** palette.

3. Move the cursor to the upper left corner of the block diagram. Place the top left corner of the While Loop here.

4. Click and drag the cursor diagonally to enclose *all* the Express VIs and wires, as shown the following figure.





Notice that the While Loop, shown at left, appears with a **STOP** button wired to the condition terminal. This While Loop is configured to stop when the user clicks the **STOP** button.

5. Display the front panel and run the VI.

   Notice that the VI now runs until you click the **STOP** button. A While Loop executes the functions inside the loop until the user presses the **STOP** button. Refer to Lesson 3, *Loops and Charts* for more information about While Loops.

Controlling the Speed of Execution

To plot the points on the waveform graph more slowly, you can add a time delay to the block diagram. Complete the following steps to control the speed at which the VI executes.

1. On the block diagram, select the Time Delay Express VI on the **Functions»Execution Control** palette and place it inside the loop.
2. Type `.250` in the **Time delay (seconds)** text box.

   This time delay specifies how fast the loop runs. With a .250 second time delay, the loop iterates once every quarter of a second.
3. Close the **Configure Time Delay** dialog box.
4. Save this VI.
5. Display the front panel and run the VI.

6. Click the **Enable** switch and notice the change on the graph.

   Notice how if the **Enable** switch is on, the graph displays the reduced signal. If the **Enable** switch is off, the graph does not display the reduced signal.

7. Click the **STOP** button to stop the VI.

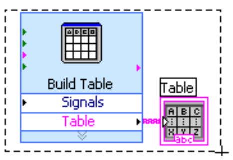
Using a Table to Display Data

Complete the following steps to display a collection of mean values in a table on the front panel.

1. On the front panel, select the Express Table indicator on the **Controls» Text Indicators** palette and place it on the front panel to the right of the waveform graph.

2. Display the block diagram.

   Notice that the **Table** terminal appears wired to the Build Table Express VI automatically.



3. If the Build Table Express VI and the **Table** terminal are not selected already, click an open space on the block diagram to the left of the Build Table Express VI and the **Table** terminal. Drag the cursor diagonally until the selection rectangle encloses the Build Table Express VI and the **Table** terminal, shown at left.

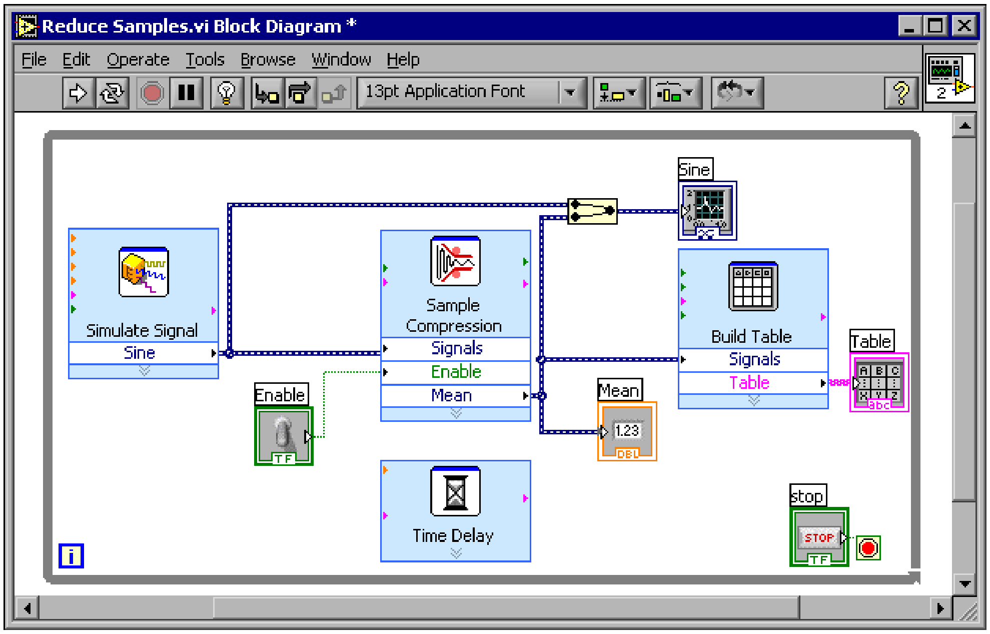
   A moving dashed outline called a marquee highlights the Build Table Express VI, the **Table** terminal, and the wire joining the two.

4. Drag the objects into the While Loop to the right of the **Mean** terminal.

   Notice that the While Loop automatically resizes to enclose the Build Table Express VI and the **Table** terminal.

5. Using the Wiring tool, wire the **Mean** terminal of the Sample Compression Express VI to the **Signals** input of the Build Table Express VI.

The block diagram should appear similar to the following figure.



6. Display the front panel and run the VI.
7. Click the **Enable** switch.

   The table displays the mean values of every 25 samples of the sine wave. Notice if the **Enable** switch is off, the table does not record the mean values.
8. Stop the VI.
9. Experiment with properties of the table by using the **Table Properties** dialog box. For example, try changing the number of columns to one.
10. Save and close the VI.

End of Exercise 1-3

Exercise 1-4: Debug Exercise (Main) VI

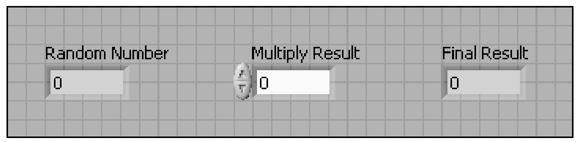
Objective: To practice debugging techniques.

Complete the following steps to load a broken VI and correct the error. Use single-stepping and execution highlighting to step through the VI.

Front Panel

1. Select **File»Open** and navigate to `C:\Exercises\LV Basics I` to open the Debug Exercise (Main) VI.

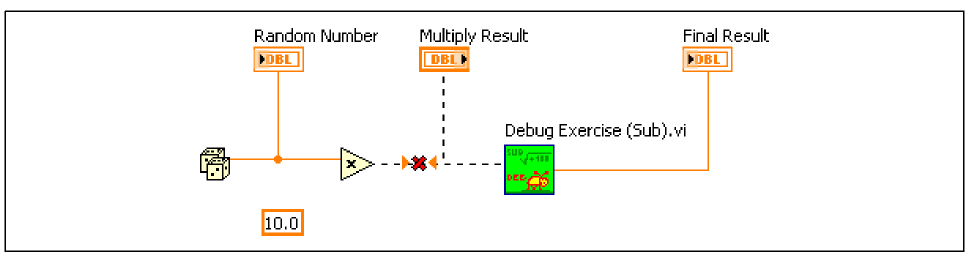
   The following front panel appears.



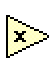
Notice the **Run** button on the toolbar appears broken, shown at left, indicating that the VI is broken and cannot run.

Block Diagram

2. Select **Window»Show Block Diagram** to display the following block diagram.



The Random Number (0-1) function, located on the **Functions» Arithmetic & Comparison»Express Numeric** palette, produces a random number between 0 and 1.



The Multiply function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, multiplies the random number by `10.0`.

The numeric constant, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, is the number to multiply by the random number.

The Debug Exercise (Sub) VI, located in the `C:\Exercises\LV Basics I` directory, adds `100.0` and calculates the square root of the value.

3. Find and fix each error.
   a. Click the broken **Run** button to display the **Error list** window, which lists all the errors.
   b. Select an error description in the **Error list** window. The **Details** section describes the error and in some cases recommends how to correct the error.
   c. Click the **Help** button to display a topic in the *LabVIEW Help* that describes the error in detail and includes step-by-step instructions for correcting the error.
   d. Click the **Show Error** button or double-click the error description to highlight the area on the block diagram that contains the error.
   e. Use the **Error list** window to fix each error.
4. Select **File»Save** to save the VI.
5. Display the front panel by clicking it or by selecting **Window»Show Front Panel**.

Run the VI

6. Click the **Run** button to run the VI several times.
7. Select **Window»Show Block Diagram** to display the block diagram.
8. Animate the flow of data through the block diagram.

   a. Click the **Highlight Execution** button, shown at left, on the toolbar to enable execution highlighting.

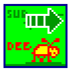
   b. Click the **Step Into** button, shown at left, to start single-stepping. Execution highlighting shows the movement of data on the block diagram from one node to another using bubbles that move along the wires. Nodes blink to indicate they are ready to execute.

   c. Click the **Step Over** button, shown at left, after each node to step through the entire block diagram. Each time you click the **Step Over** button, the current node executes and pauses at the next node.

   Data appear on the front panel as you step through the VI. The VI generates a random number and multiplies it by `10.0`. The subVI adds `100.0` and takes the square root of the result.

   d. When a blinking border surrounds the entire block diagram, click the **Step Out** button, shown at left, to stop single-stepping through the Debug Exercise (Main) VI.
9. Single-step through the VI and its subVI.
   a. Click the **Step Into** button to start single-stepping.

   b. When the Debug Exercise (Sub) VI blinks, click the **Step Into** button. Notice the run button on the subVI.



c. Display the Debug Exercise (Main) VI block diagram by clicking it. A green glyph, shown at left, appears on the subVI icon on the Debug Exercise (Main) VI block diagram, indicating that the subVI is running.

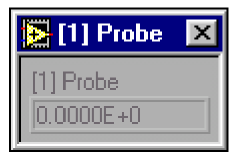
d. Display the Debug Exercise (Sub) VI block diagram by clicking it.

e. Click the **Step Out** button twice to finish single-stepping through the subVI block diagram. The Debug Exercise (Main) VI block diagram is active.

f. Click the **Step Out** button to stop single-stepping.

10. Use a probe to check intermediate values on a wire as a VI runs.

a. Use the Probe tool, shown at left, to click any wire. A window similar to the following window appears.



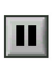
LabVIEW numbers the **Probe** window automatically and displays the same number in a glyph on the wire you clicked.

b. Single-step through the VI again. The **Probe** window displays data passed along the wire.

11. Place breakpoints on the block diagram to pause execution at that location.

a. Use the Breakpoint tool, shown at left, to click nodes or wires. Place a breakpoint on the block diagram to pause execution after all nodes on the block diagram execute.

b. Click the **Run** button to run the VI. When you reach a breakpoint during execution, the VI pauses and the **Pause** button on the toolbar appears red.



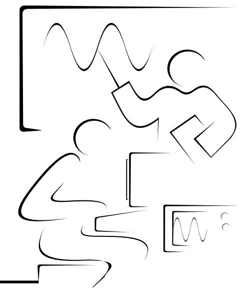
c. Click the **Continue** button, shown at left, to continue running to the next breakpoint or until the VI finishes running.

d. Use the Breakpoint tool to click the breakpoints you set and remove them.

12. Click the **Highlight Execution** button to disable execution highlighting.

13. Select **File»Close** to close the VI and all open windows.

End of Exercise 1-4



Lesson 2

Modular Programming

Exercises **Page**

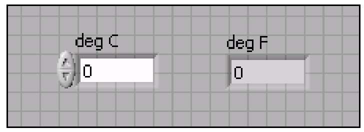
Exercise 2-1: Convert C to F VI

Objective: To create a VI and the icon and connector pane needed to use as a subVI.

Complete the following steps to create a VI that takes a number representing degrees Celsius and converts it to a number representing degrees Fahrenheit.

Front Panel

1. Open a blank VI and begin building the following front panel.



2. (Optional) Select **Window»Tile Left and Right** to display the front panel and block diagram side by side or **Window»Tile Up and Down** to display the front panel and block diagram stacked.
3. Create a numeric control. You will use this control to enter the value for degrees Celsius.
   a. Select **Controls»Numeric Controls** to display the **Numeric Controls** palette. If the **Controls** palette is not visible, right-click an open space on the front panel workspace to display it.

   b. Select the Numeric Control. Move the control to the front panel and click to place the control.

   c. Type `deg C` inside the label of the control and press the <Enter> key or click the **Enter** button, shown at left, on the toolbar. If you do not type the name immediately, LabVIEW uses a default label.

Tip You can edit a label at any time by double-clicking the label, using the Labeling tool, or right-clicking and selecting **Properties** from the shortcut menu to display the property dialog box.

4. Create a numeric indicator. You will use this indicator to display the value for degrees Fahrenheit.

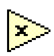
   a. Select the Numeric Indicator located on the **Controls»Numeric Indicators** palette.
   b. Move the indicator to the front panel and click to place the indicator.
   c. Type `deg F` inside the label and press the <Enter> key or click the **Enter** button.

Block Diagram

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

   LabVIEW creates corresponding control and indicator terminal icons on the block diagram when you place controls and indicators on the front panel. The terminals represent the data type of the control or indicator. You should see two double-precision, floating-point terminals on the block diagram, one indicator, and one control.

Note Control terminals have a thicker border than indicator terminals.



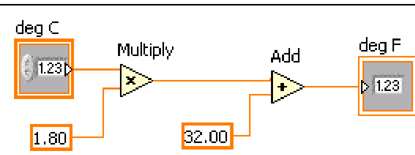
6. Place the Multiply function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram to the right of the `deg C` indicator. If the **Functions** palette is not visible, right-click an open space on the block diagram workspace to display it.

7. Place the Add function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram to the right of the Multiply function.

8. Place a Numeric Constant, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, to the lower left of the Multiply function. Type `1.80` in the constant. When you first place a numeric constant, it is highlighted so you can type a value. If the constant is no longer highlighted, double-click the constant to activate the Labeling tool.

9. Place a Numeric Constant, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, to the left of the Add function. Type `32.0` in the constant.

10. Use the Wiring tool, shown at left, to wire the icons as shown in the following 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. You can start wiring at either terminal.
- You can bend a wire by clicking to tack down the wire 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 on the block diagram. 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, displaying the name of the terminal. If the Context Help window is open, the terminal area also blinks in the Context Help window.
- To cancel a wire you started, press the <Esc> key, right-click, or click the terminal where you started the wire.

11. Display the front panel by clicking it or by selecting **Window»Show Front Panel**.
12. Save the VI as `Convert C to F.vi` in the `C:\Exercises\LV Basics I` directory.

Run the VI

13. Enter a number in the numeric control and run the VI.

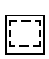
   a. Use the Operating tool, shown at left, or the Labeling tool to double-click the numeric control and type a new number.

   b. Click the **Run** button, shown at left, to run the VI.

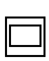
   c. Try several different numbers and run the VI again.

Icon and Connector Pane

14. Right-click the icon in the upper right corner of the front panel window and select **Edit Icon** from the shortcut menu. The **Icon Editor** dialog box appears.

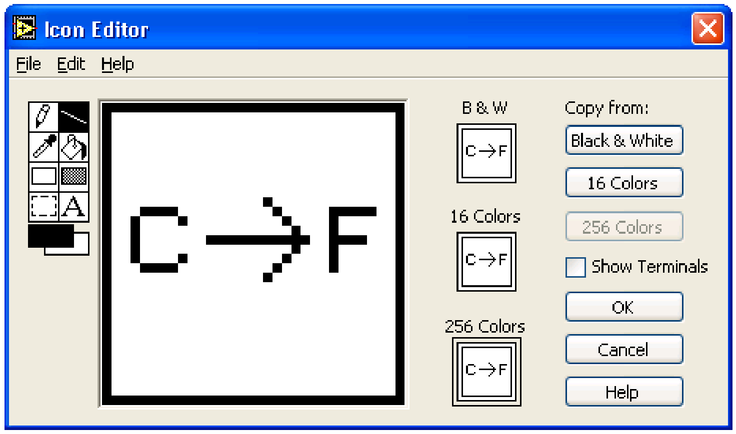


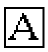
15. Double-click the Select tool, shown at left, on the left side of the **Icon Editor** dialog box to select the default icon.
16. Press the <Delete> key to remove the default icon.



17. Double-click the Rectangle tool, shown at left, to redraw the border.

18. Create the following icon.





a. Double-click the Text tool, shown at left, and change the font to **Small Fonts**.

b. Use the Text tool to click the editing area where you will begin typing.

c. Type `C` and `F`. While the text is active, you can move the text by pressing the arrow keys.

d. Use the Pencil tool, shown at left, 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.

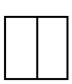
f. Select the **B & W** icon and click the **256 Colors** button in the **Copy from** section to create a black and white icon, which LabVIEW uses for printing unless you have a color printer.

g. Select the **16 Colors** icon and click the **256 Colors** button in the **Copy from** section.

h. When you complete the icon, 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.

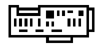
19. Right-click the icon on the front panel and select **Show Connector** from the shortcut menu to define the connector pane terminal pattern.

LabVIEW selects a default 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, shown at left.



20. Assign the terminals to the numeric control and numeric indicator.
    a. Select **Help»Show Context Help** to display the **Context Help** window.
    b. Click the left terminal in the connector pane. The tool automatically changes to the Wiring tool, and the terminal turns black.
    c. Click the **deg C** control. A marquee highlights the control on the front panel.
    d. Click an open space on 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.
    e. Click the right terminal in the connector pane and click the **deg F** indicator.
    f. Click an open space on the front panel. Both terminals of the connector pane are orange.
    g. Move the cursor over the connector pane. The **Context Help** window shows that both terminals are connected to double-precision, floating-point values.
21. Save and close the VI. You will use this VI later in the course.

End of Exercise 2-1

Exercise 2-2:Thermometer VI 

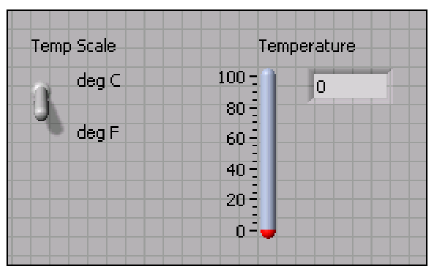
Objective: To build a VI using the Convert C to F subVI to read a temperature from the DAQ Signal Accessory in Celsius or Fahrenheit.

Complete the following steps to create a VI that reads a measurement from the temperature sensor on the DAQ Signal Accessory and displays the temperature in Celsius or Fahrenheit.

The sensor returns a voltage proportional to temperature. For example, if the temperature is 23 °C, the sensor output voltage is 0.23 V. The sensor is connected to Channel 0 of Device 1. Device 1 is the DAQ device. On some systems, the DAQ device may have another device number.

Front Panel

1. In the **LabVIEW** dialog box, click the arrow on the **New** button and select **Blank VI** from the shortcut menu or press the <Ctrl-N> keys to open a blank VI.
2. Create the following front panel.



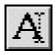
   a. Place a thermometer, located on the **Controls»Numeric Indicators** palette, on the front panel.

   b. Type `Temperature` in the label and press the <Enter> key or click the **Enter** button on the toolbar, shown at left.

   c. Right-click the thermometer and select **Visible Items»Digital Display** from the shortcut menu to show the digital display for the thermometer.

   d. Place a vertical toggle switch control, located on the **Controls» Buttons & Switches** palette, on the front panel.

   e. Type `Temp Scale` in the label and press the <Enter> key or click the **Enter** button.



f. Use the Labeling tool, shown at left, to place a free label, `deg C`, next to the TRUE position of the switch. If you are using automatic tool selection, double-click the blank area of the front panel to begin typing a free label.

g. Place a free label, `deg F`, next to the FALSE position of the switch.

User Documentation

3. Document the VI so a description appears in the **Context Help** window when you move the cursor over the VI icon.

   a. Select **File»VI Properties** to display the **VI Properties** dialog box.

   b. Select **Documentation** from the **Category** pull-down menu.

   c. Type the following description for the VI in the **VI description** text box:

   ```
   This VI measures temperature using the temperature
   sensor on the DAQ Signal Accessory.
   ```

   d. Click the **OK** button.

4. Document the thermometer indicator and switch control so descriptions appear in the **Context Help** window when you move the cursor over the objects and tip strips appear on the front panel or block diagram when you move the cursor over the objects while the VI runs.

   a. Right-click the thermometer indicator and select **Description and Tip** from the shortcut menu.

   b. Type the following description for the thermometer in the **Description** text box:

   ```
   Displays the temperature measurement.
   ```

   c. Type `temperature` in the **Tip** text box.

   d. Click the **OK** button.

   e. Right-click the switch control and select **Description and Tip** from the shortcut menu.

   f. Type the following description for the vertical switch control in the **Description** text box:

   ```
   Determines the scale (Fahrenheit or Celsius) to
   use for the temperature measurement.
   ```

   g. Type `scale – C or F` in the **Tip** text box.

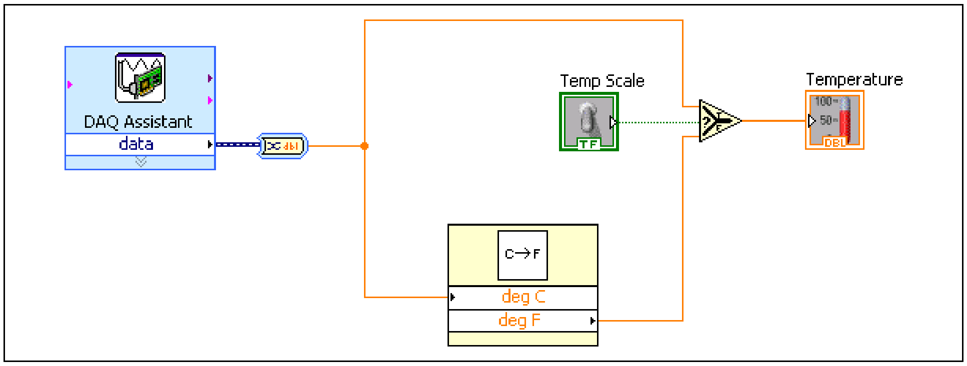
   h. Click the **OK** button.

5. Select **Help»Show Context Help** to display the **Context Help** window.

6. Move the cursor over the front panel objects and the VI icon to display the descriptions in the **Context Help** window.

Block Diagram

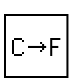
7. Select **Window»Show Block Diagram** to display the block diagram.



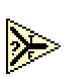
8. Place the DAQ Assistant Express VI, located on the **Functions»Input** palette, on the block diagram. When you place this Express VI on the block diagram the DAQ Assistant configuration dialog box appears.
   a. Select **Analog Input»Voltage** for the type of measurement to make.
   b. Select **Dev1»ai0** (or **Dev2»ai0**) for the physical channel and click the **Finish** button.
   c. You must multiply the temperature by 100 to convert it from voltage to Celsius. On the **Settings** tab, select **Custom Scaling»Create New**. Select a **Linear** scale. Name the scale `Temperature`. Enter a slope scale of `100`. Click the **OK** button.
   d. Select the **Acquire 1 Sample** option on the **Task Timing** tab. Click the **OK** button.

Note If you do not have a DAQ device with a temperature sensor connected to your computer, use the (Demo) Read Voltage VI, located in the `C:\Exercises\LV Basics I` directory.

9. Place the Convert from Dynamic Data Express VI, located on the **Functions»Signal Manipulation** palette, on the block diagram. This VI converts the dynamic data type. In the configuration dialog box, select **Single scalar** in the **Resulting data type** listbox.



10. Place the Convert C to F VI on the block diagram. Select **Functions»All Functions»Select a VI**, navigate to `C:\Exercises\LV Basics I\Convert C to F.vi`. This VI converts the Celsius readings to Fahrenheit.



11. Place the Select function, located on the **Functions»Arithmetic & Comparison»Express Comparison** palette, on the block diagram. This function returns either the Fahrenheit (FALSE) or Celsius (TRUE) temperature value, depending on the value of **Temp Scale**.

Use the Positioning tool to place the icons as shown in the previous block diagram and use the Wiring tool to wire them together.

Tip To display terminals for a node, right-click the icon and select **Visible Items» Terminals** from the shortcut menu.

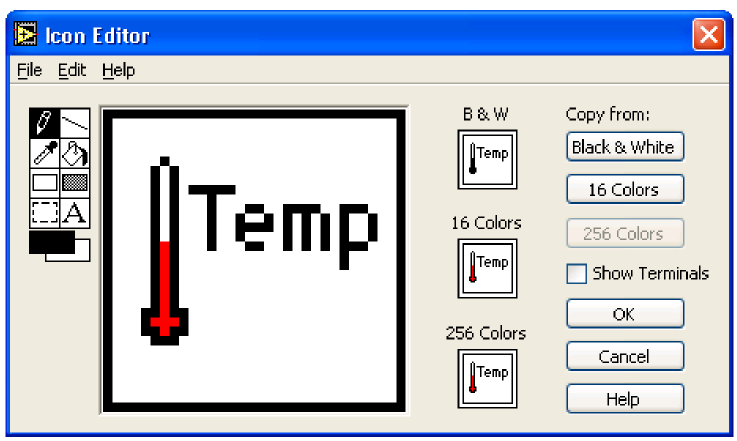
Front Panel

12. Display the front panel by clicking it or by selecting **Window»Show Front Panel**.

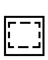
13. Click the **Run Continuously** button, shown at left, to run the VI continuously.
14. Put your finger on the temperature sensor and notice the temperature increase.
15. Click the **Run Continuously** button again to stop the VI. This allows the VI to finish the current run of the VI.

Icon and Connector Pane

16. Create an icon so you can use the Thermometer VI as a subVI. The following icon is an example. If necessary, create a simpler icon to save time.

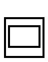


a. Right-click the icon in the upper right corner of the front panel and select **Edit Icon** from the shortcut menu. The **Icon Editor** dialog box appears.



b. Double-click the Select tool, shown at left, on the left side of the **Icon Editor** dialog box to select the default icon.

c. Press the <Delete> key to remove the default icon.



d. Double-click the Rectangle tool, shown at left, to redraw the border.

e. Use the Pencil tool, shown at left, to draw an icon that represents the thermometer.

f. Use the Foreground and Fill tools to color the thermometer red.

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

g. Double-click the Text tool, shown at left, and change the font to **Small Fonts**.

h. Type `Temp`. Reposition the text if necessary.

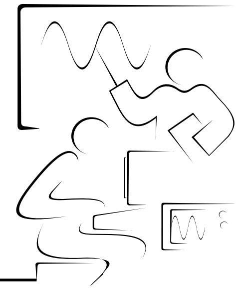
i. Select the **B & W** icon and select **256 Colors** in the **Copy from** section to create a black and white icon, which LabVIEW uses for printing unless you have a color printer.

j. When the icon is complete, click the **OK** button. The icon appears in the upper right corner of the front panel.

17. Right-click the icon and select **Show Connector** from the shortcut menu and assign terminals to the switch and the thermometer.
    a. Click the left terminal in the connector pane.
    b. Click the **Temp Scale** control. The left terminal turns green.
    c. Click the right terminal in the connector pane.
    d. Click the **Temperature** indicator. The right terminal turns orange.
    e. Click an open space on the front panel.
18. Save the VI as `Thermometer.vi` in the `C:\Exercises\LV Basics I` directory.You will use this VI later in the course.
19. Close the VI.

End of Exercise 2-2

Lesson 3



Loops and Charts

Exercises **Page**

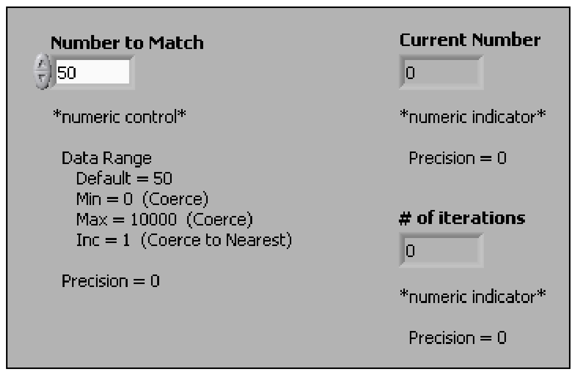
Exercise 3-1: Auto Match VI

Objective: To pass data out of a While Loop through a tunnel.

Complete the following steps to build a VI that generates random numbers until the number generated matches a number you specify. The iteration terminal records the number of random numbers generated until a match occurs.

Front Panel

1. Open a blank VI and build the following front panel. Modify the controls and indicators as shown in the following front panel and as described in the following steps.



   a. Place a numeric control, located on the **Controls»Numeric Controls** palette, on the front panel. Label the control `Number to Match`. This control specifies the number to match.

   b. Place a numeric indicator, located on the **Controls»Numeric Indicators** palette, on the front panel. Label the indicator `Current Number`. This indicator displays the current random number.

   c. Place another numeric indicator on the front panel. Label the indicator `# of iterations`. This indicator displays the number of iterations before a match.

Setting the Data Range

Set a data range for a control to prevent the user from selecting a value that is not compatible with a range or increment. You can choose to ignore a value that is out of range or coerce it to within the range. Complete the following steps to set the range between `0` and `10,000` with an increment of `1` and a default value of `50`.

2. Right-click the **Number to Match** control and select **Data Range** from the shortcut menu. The **Data Range** page of the **Numeric Properties** dialog box appears.
   a. Remove the checkmark from the **Use Default Range** checkbox.
   b. Set the **Default Value** to `50`.
   c. Set the **Minimum** value to `0` and select **Coerce** from the **Out of Range Action** pull-down menu.
   d. Set the **Maximum** value to `10,000` and select **Coerce** from the **Out of Range Action** pull-down menu.
   e. Set the **Increment** value to `1` and select **Coerce to Nearest** from the **Out of Range Action** pull-down menu. Do not close the dialog box.

Modifying Digits of Precision

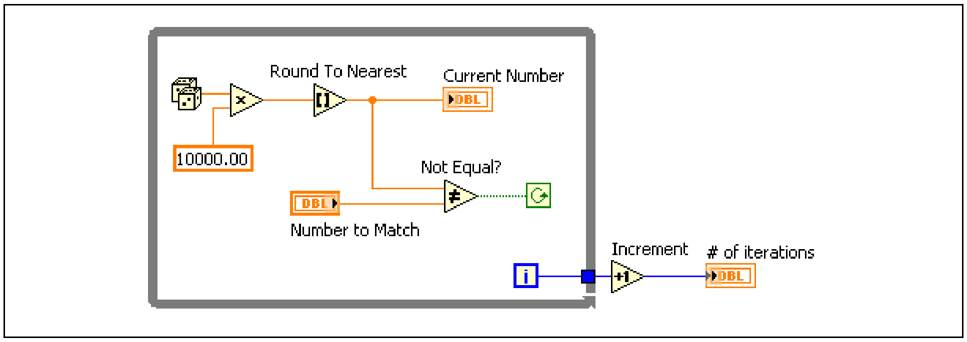
By default, LabVIEW automatically formats numeric controls. You also can specify the precision or notation. You can display numeric values in floating-point, scientific, or SI notation. Complete the following steps to change the precision to `0`.

3. Select the **Format and Precision** tab.

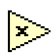
   If you closed the **Numeric Properties** dialog box, right-click the **Current Number** indicator and select **Format & Precision** from the shortcut menu. The **Format & Precision** page of the **Numeric Properties** dialog box appears.
   a. Select **Floating Point** and change **Significant digits** to **Digits of precision**.
   b. Type `0` in the **Digits of precision** text box and click the **OK** button.
4. Repeat step 3 to set the precision for **Current Number** and **# of iterations** indicators.

Block Diagram

5. Build the following block diagram.

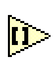


a. Place the Random Number (0-1) function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram. This function produces a random number between 0 and 1.

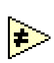


b. Place the Multiply function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram. This function multiplies the random number by `10,000` to produce a random number between 0 and 10,000.

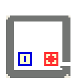
c. Right-click the **y** terminal of the Multiply function, select **Create» Constant** from the shortcut menu, type `10000`, and press the <Enter> key to create a numeric constant.



d. Place the Round To Nearest function, located on the **Functions» Arithmetic & Comparison»Express Numeric** palette, on the block diagram. This function rounds the random number to the nearest integer.



e. Place the Not Equal? function, located on the **Functions» Arithmetic & Comparison»Express Comparison** palette, on the block diagram. This function compares the random number with **Number to Match** and returns TRUE if the numbers are not equal; otherwise, it returns FALSE.



f. Place the While Loop, located on the **Functions»Execution Control** palette, on the block diagram. Right-click the conditional terminal and select **Continue if True** from the shortcut menu.

g. Wire the iteration terminal to the border of the While Loop. A blue tunnel appears on the While Loop border. You will wire the tunnel to the Increment function. Each time the loop executes, the iteration terminal increments by one. The iteration count passes out of the loop upon completion. Increment this value by one outside the loop because the count starts at `0`.

h. Place the Increment function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram. This function adds `1` to the While Loop count. A coercion dot appears on the **# of iterations** output to indicate that LabVIEW coerced the numeric representation of the iteration terminal to match the numeric representation of the **# of iterations** output. Refer to the *For Loops* section for more information about numeric conversion.

6. Save the VI as `Auto Match.vi` in the `C:\Exercises\LV Basics I` directory.

Run the VI

7. Display the front panel and change the number in **Number to Match**.
8. Run the VI. Change **Number to Match** and run the VI again. **Current Number** updates at every iteration of the loop because it is inside the loop. **# of iterations** updates upon completion because it is outside the loop.
9. To see how the VI updates the indicators, enable execution highlighting. On the block diagram toolbar, click the **Highlight Execution** button, shown at left, to enable execution highlighting. Execution highlighting shows the movement of data on the block diagram from one node to another so you can see each number as the VI generates it.

10. Change **Number to Match** to a number that is out of the data range, which is `0` to `10,000` with an increment of `1`.
11. Run the VI. LabVIEW coerces the out-of-range value to the nearest value in the specified data range.
12. Close the VI.

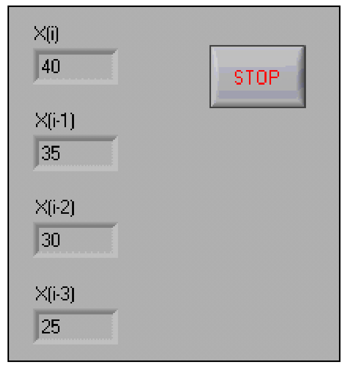
End of Exercise 3-1

Exercise 3-2: Shift Register Example VI

Objective: To use shift registers to access values from previous iterations.

Front Panel

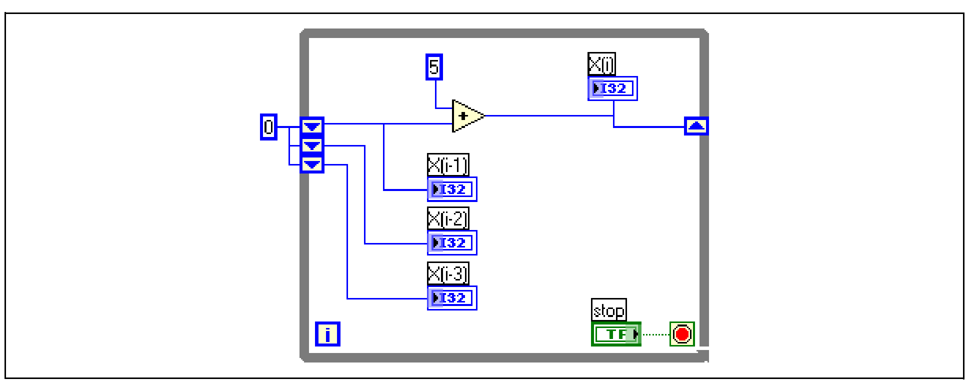
1. Open the Shift Register Example VI located in the `C:\Exercises\LV Basics I` directory. The following front panel is already built.



The **X(i)** indicator displays the current value, which shifts to the left terminal at the beginning of the next iteration. The **X(i-1)** indicator displays the value one iteration ago, the **X(i-2)** indicator displays the value two iterations ago, and so on.

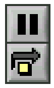
Block Diagram

2. Display the following block diagram and make sure both the front panel and block diagram are visible. If necessary, close or move the **Tools** and **Functions** palettes.



The `0` wired to the left terminals initializes the elements of the shift register to `0`.

3. Click the **Highlight Execution** button, shown at left, to enable execution highlighting.



4. Run the VI and watch the bubbles that move along the wires. If the bubbles are moving too fast, click the **Pause** and **Step Over** buttons, shown at left, to slow the execution.

   In each iteration of the While Loop, the VI funnels the previous values through the left terminals of the shift register. Each iteration of the loop adds `5` to the current data, **X(i)**. This value shifts to the left terminal, **X(i-1)**, at the beginning of the next iteration. The values at the left terminal funnel downward through the terminals. This VI retains the last three values. To retain more values, add more elements to the left terminal of the shift register by right-clicking the left terminal and selecting **Add Element** from the shortcut menu.

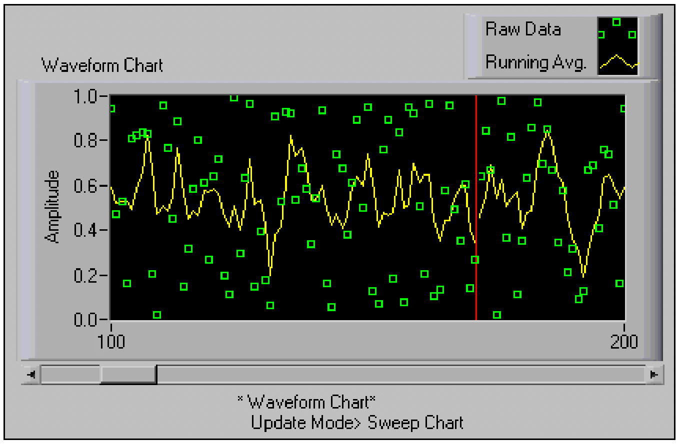
5. Close the VI. Do not save changes.

End of Exercise 3-2

Exercise 3-3: Random Average VI

Objective: To build a VI that displays two plots, a random plot and a running average of the last four points, on a waveform chart in sweep update mode.

1. Build this VI, using the following tips:
   - Use a For Loop ($n = 200$) instead of a While Loop. The sweep chart should be similar to the following chart.



   - Use a shift register with three left terminals to average the last four data points.
   - Use the Random Number (0-1) function located on the **Functions» Arithmetic & Comparison»Express Numeric** palette to generate the data.
   - Use the Bundle function located on the **Functions»All Functions» Cluster** palette to group the random data with the averaged data before plotting.
2. Save the VI and name it `Random Average.vi`.
3. Close the VI.

End of Exercise 3-3

Exercise 3-4: Temperature Running Average VI

Objective: To use shift registers to perform a running average.

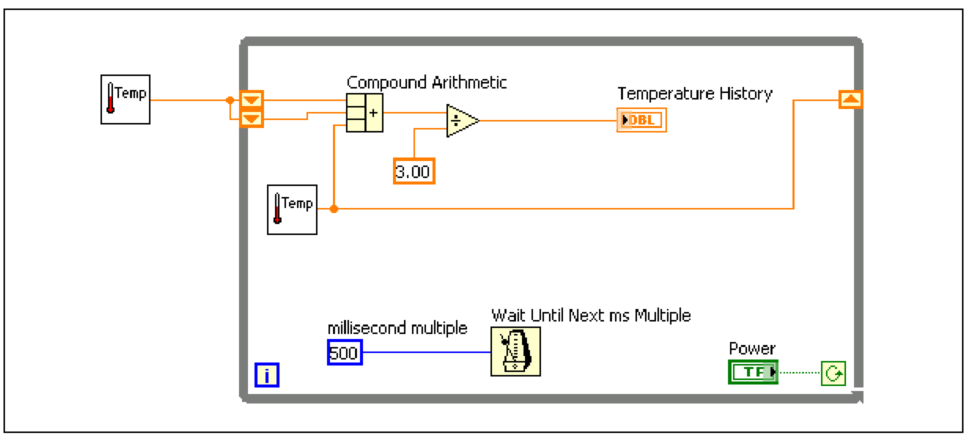
Complete the following steps to modify the Temperature Monitor VI to average the last three temperature measurements and display the average on a waveform chart.

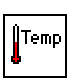
Front Panel

1. Open the Temperature Monitor VI in the `C:\Exercises\LV Basics I` directory.
2. Select **File»Save As** and rename the VI `Temperature Running Average.vi`.

Block Diagram

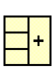
3. Display the block diagram.
4. Right-click the right or left border of the While Loop and select **Add Shift Register** from the shortcut menu to create a shift register.
5. Right-click the left terminal of the shift register and select **Add Element** from the shortcut menu to add an element to the shift register.
6. Modify the block diagram as follows.





a. Press the <Ctrl> key while you click the Thermometer VI and drag it outside the While Loop to create a copy of the subVI.

The Thermometer VI returns one temperature measurement from the temperature sensor and initializes the left shift registers before the loop starts.



b. Place the Compound Arithmetic function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram. This function returns the sum of the current temperature and the two previous temperature readings. Use the Positioning tool to resize the function to have three left terminals.

c. Place the Divide function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram. This function returns the average of the last three temperature readings.

d. Right-click the **y** terminal of the Divide function, select **Create» Constant**, type `3`, and press the <Enter> key.

7. Save the VI.

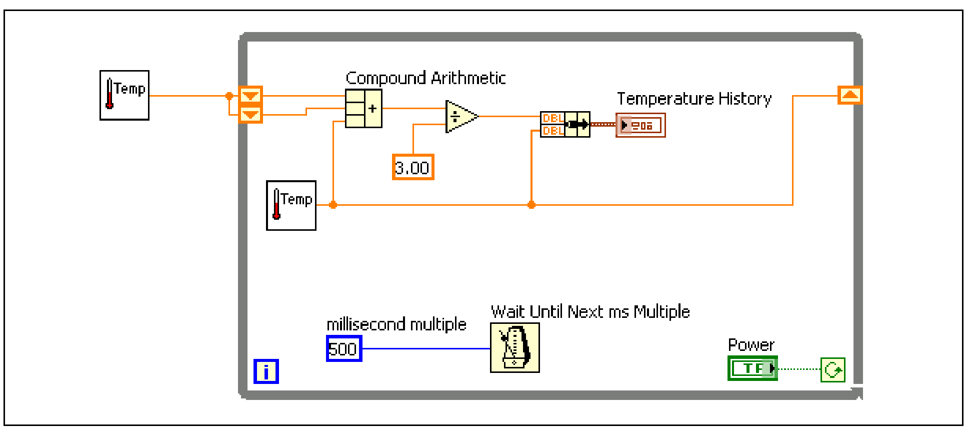
Run the VI

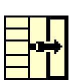
8. Run the VI.

During each iteration of the While Loop, the Thermometer VI takes one temperature measurement. The VI adds this value to the last two measurements stored in the left terminals of the shift register. The VI divides the result by three to find the average of the three measurements, the current measurement plus the previous two. The VI displays the average on the waveform chart. Notice that the VI initializes the shift register with a temperature measurement.

Block Diagram

9. Modify the block diagram as shown in the following illustration.





a. Place the Bundle function, located on the **Functions»All Functions»Cluster** palette, on the block diagram. This function bundles the average and current temperature for plotting on the waveform chart.

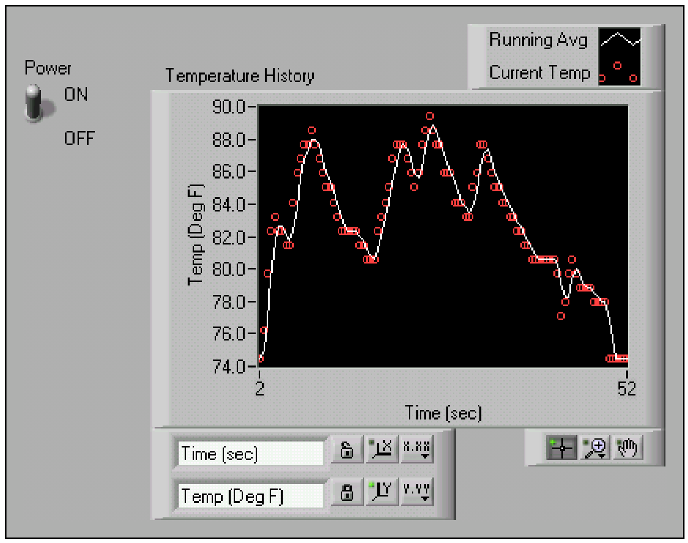
10. Save the VI. You will use this VI later in the course.

Run the VI

11. Run the VI. The VI displays two plots on the waveform chart. The plots are overlaid. That is, they share the same vertical scale.
12. If time permits, complete the optional steps. Otherwise, close the VI.

Optional

Customize the waveform chart as shown in the following front panel. You can display a plot legend, a scale legend, a graph palette, a digital display, and a scrollbar. By default, a waveform chart displays the plot legend.



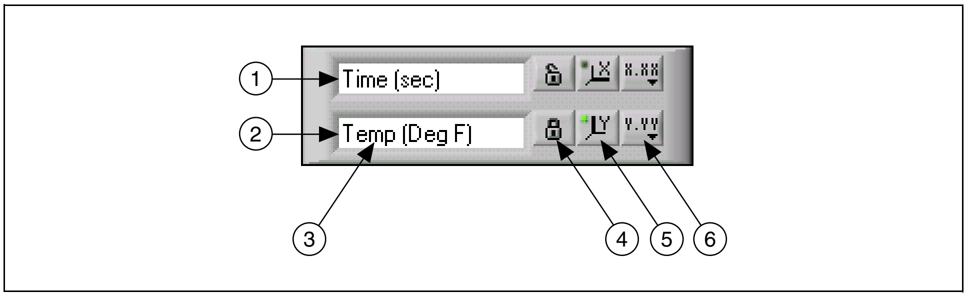
13. Customize the y-axis.

   a. Use the Labeling tool to double-click `70.0` in the y-axis, type `75.0`, and press the <Enter> key.
   b. Use the Labeling tool to double-click the second number from the bottom on the y-axis, type `80.0`, and press the <Enter> key. This number determines the numerical spacing of the y-axis divisions.

      For example, if the number above 75.0 is 77.5, it indicates a y-axis division of 2.5, changing the 77.5 to 80.0 reformats the y-axis to multiples of 5.0 (75.0, 80.0, 85.0, and so on).

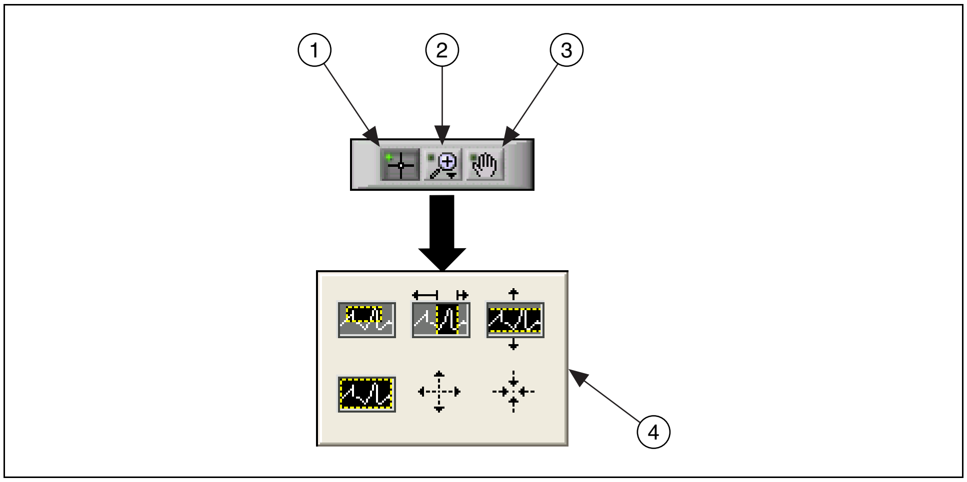
Note The waveform chart size has a direct effect on the display of axis scales. Increase the waveform chart size if you encounter problems while customizing the axis.

14. Right-click the waveform chart and select **Visible Items»Scale Legend** from the shortcut menu to display the scale legend, as shown in the following illustration. You can place the scale legend anywhere on the front panel.



1 X-axis	3 Scale Labels	5 Autoscale Button
2 Y-axis	4 Scale Lock Button	6 Scale Format Button

15. Use the scale legend to customize each axis.
    a. Make sure the **Lock Autoscale** button appears locked and the **Autoscale** LED is green so the y-axis adjusts the minimum and maximum values to fit the data in the chart.
    b. Click the **Scale Format** button to change the format, precision, mapping mode, scale visibility, and grid options for each axis.
16. Use the plot legend to customize the plots.
    a. Use the Positioning tool to resize the plot legend to include two plots.
    b. Use the Labeling tool to change `Temp` to `Running Avg` and to change `Plot 1` to `Current Temp`. If the text does not fit, use the Positioning tool to resize the plot legend.
    c. Right-click the plot in the plot legend to set the line and point styles and the color of the plot background or traces.
17. Right-click the waveform chart and select **Visible Items»Graph Palette** from the shortcut menu to display the graph palette, as shown in the following illustration. You can place the graph palette anywhere on the front panel.



1 Cursor Movement Tool	3 Panning Tool
2 Zoom Button	4 Zoom Pull-down Menu

Use the **Zoom** button on the graph palette to zoom in or out of sections of the chart or the whole chart. Use the Panning tool to pick up the plot and move it around on the display. Use the Cursor Movement tool to move the cursor on the graph.

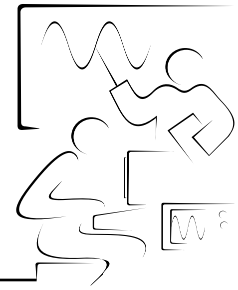
18. Run the VI. While the VI runs, use the buttons in the scale legend and graph palette to modify the waveform chart.

Note If you modify the axis labels, the display might become larger than the maximum size that the VI can correctly present.

19. Use the Operating tool to click the **Power** switch and stop the VI.
20. Save and close the VI.

End of Exercise 3-4

Lesson 4



Arrays, Graphs and Clusters

Exercises	Page

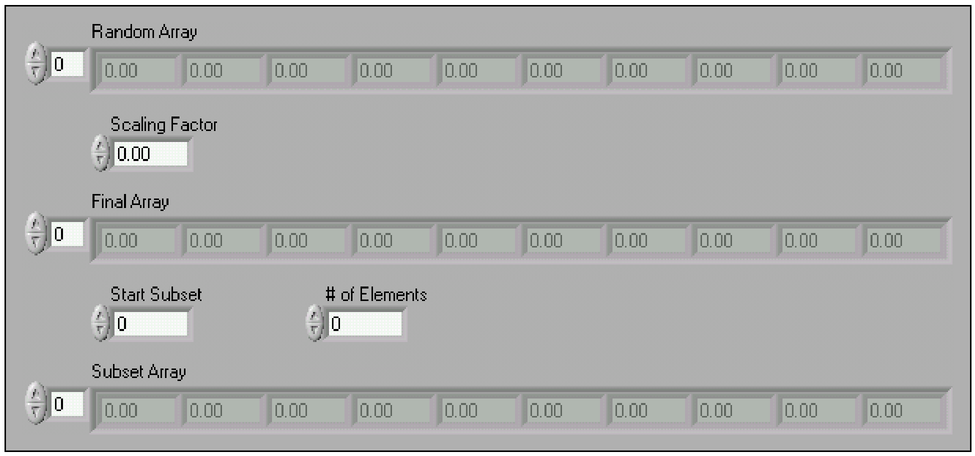
Exercise 4-1: Array Exercise VI

Objective: To create arrays and become familiar with the Array functions.

Complete the following steps to build a VI that creates an array of random numbers, scales the resulting array, and takes a subset of that final array.

Front Panel

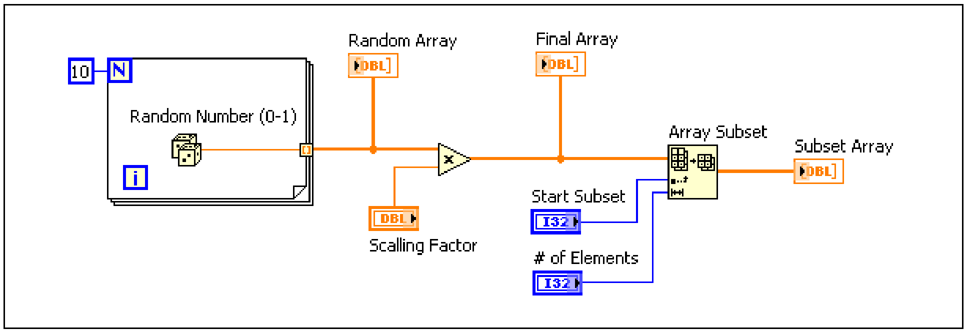
1. Open a blank VI and build the following front panel.



   a. Place an array, located on the **Controls»All Controls»Array & Cluster** palette, on the front panel.
   b. Label the array `Random Array`.
   c. Place a numeric indicator, located on the **Controls»Numeric Indicators** palette, in the array shell.
   d. Use the Positioning tool to resize the array control to contain 10 numeric indicators.
   e. Press the <Ctrl> key while you click and drag the **Random Array** control to create two copies of the control.
   f. Label the copies `Final Array` and `Subset Array`.
   g. Place three numeric controls, located on the **Controls»Numeric Controls** palette, and label them `Scaling Factor`, `Start Subset`, and `# of Elements`.
   h. Right-click the **Start Subset** and **# of Elements** controls and select **Representation»I32** from the shortcut menu.
   i. Do not change the values of the front panel controls.

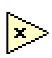
Block Diagram

2. Build the following block diagram.

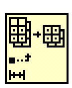


a. Place the Random Number (0-1) function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram. This function generates a random number between 0 and 1.

b. Place the For Loop, located on the **Functions»All Functions» Structures** palette, on the block diagram. The loop accumulates an array of 10 random numbers at the output tunnel. Create a constant of `10` for the count terminal.



c. Place the Multiply function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram. In this exercise this function multiplies Random Array by Scaling Factor and returns Final Array.



d. Place the Array Subset function, located on the **Functions»All Functions»Array** palette, on the block diagram. This function returns a portion of an array starting at **Start Subset** and containing **# of Elements** elements.

3. Save the VI as `Array Exercise.vi` in the `C:\Exercises\ LV Basics I` directory.

Run the VI

4. Display the front panel, change the values of the controls, and run the VI a few times.

   The For Loop runs for 10 iterations. Each iteration generates a random number and stores it at the output tunnel. **Random Array** displays an array of 10 random numbers. The VI multiplies each value in **Random Array** by **Scaling Factor** to create **Final Array**. The VI takes a subset of **Final Array** starting at **Start Subset** for **# of Elements** and displays the subset in **Subset Array**.

5. Close the VI.

End of Exercise 4-1

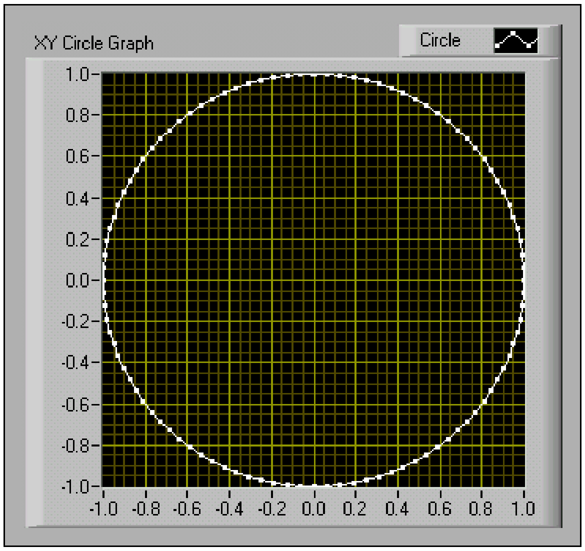
Exercise 4-2: Graph Circle VI

Objective: To plot data using an XY Graph.

Complete the following steps to build a VI that plots a circle using independent *x* and *y* arrays.

Front Panel

1. Open a blank VI and build the following front panel.



   a. Place an XY Graph, located on the **Controls»Graph Indicators** palette, on the front panel.

   b. Label the graph `XY Circle Graph`.

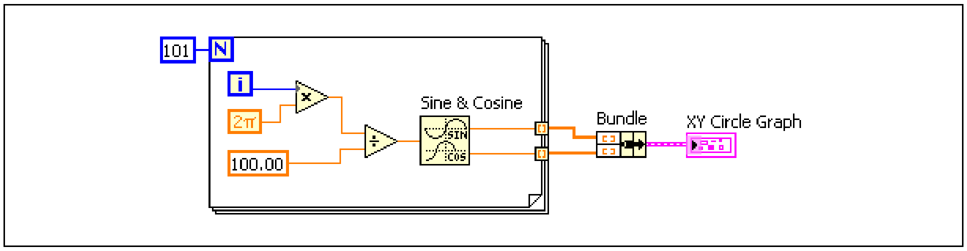
   c. Change `Plot 0` to `Circle` in the plot legend.

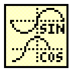
   d. Right-click the plot in the plot legend, select **Point Style** from the shortcut menu, and select the small square.

   e. Change the scale labels and ranges, as shown in the previous figure.

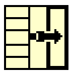
Block Diagram

2. Build the following block diagram.





a. Place the Sine & Cosine function, located on the **Functions»Arithmetic & Comparison»Express Numeric»Express Trigonometric** palette, on the block diagram. This function builds an array of points that represents one cycle of a sine wave and a cosine wave.



b. Place the Bundle function, located on the **Functions»All Functions»Cluster** palette, on the block diagram. This function assembles the sine array and the cosine array to plot the sine array against the cosine array to produce a circle.

c. Place the Pi Multiplied by 2 constant, located on the **Functions»Arithmetic & Comparison»Express Numeric»Express Numeric Constants** palette, on the block diagram.

3. Save the VI as `Graph Circle.vi` in the `C:\Exercises\LV Basics I` directory.
4. Display the front panel and run the VI.
5. Close the VI.

End of Exercise 4-2

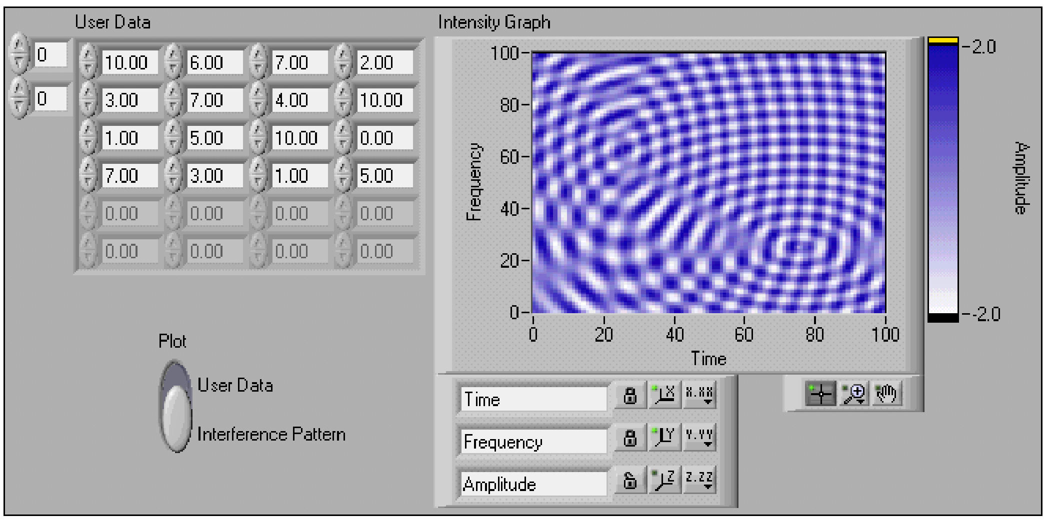
Exercise 4-3: Intensity Graph Example VI

Objective: To use an intensity graph.

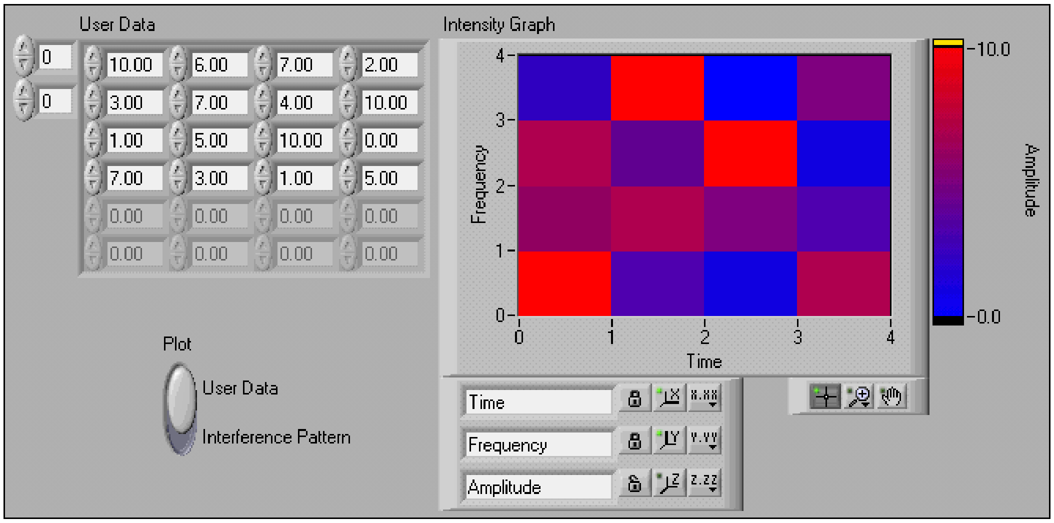
In this exercise, you use a VI that displays a wave interference pattern on an intensity graph. You also use the VI to plot a 2D array of data on the graph.

Front Panel

1. Open and run the Intensity Graph Example VI located in the `C:\Exercises\LV Basics I` directory. By default, the VI plots an interference waveform. A Property Node on the block diagram defines the color range used in the intensity graph. You can change the color range by opening the block diagram and modifying the Color Array constant.



2. Change the Plot switch on the front panel to User Data and enter values between `0.0` and `10.0` in the User Data array control. Run the VI. Notice how the magnitude of each element is mapped to the intensity graph.



3. Close the VI. Do not save changes.

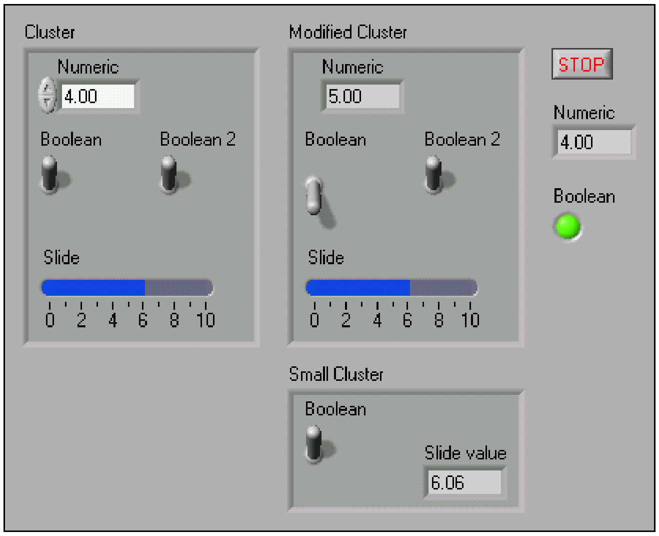
End of Exercise 4-3

Exercise 4-4: Cluster Exercise VI

Objective: To create clusters on the front panel and use the Cluster functions to assemble and disassemble clusters.

Front Panel

1. Open a blank VI and build the following front panel.



a. Place a stop button, located on the **Controls»Buttons & Switches** palette, a numeric indicator, located on the **Controls»Numeric Indicators** palette, and a round LED, located on the **Controls» LEDs** palette, on the front panel.

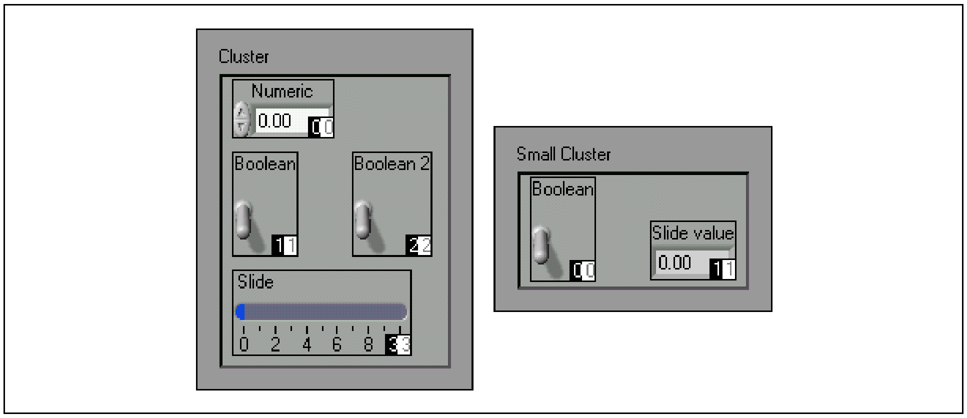
b. Place a cluster, located on the **Controls»All Controls»Array & Cluster** palette, on the front panel.

c. Place a numeric control, located on the **Controls»Numeric Controls** palette, two vertical toggle switches, located on the **Controls»Buttons & Switches** palette, and a horizontal fill slide, located on the **Controls»Numeric Controls** palette, in the cluster.

d. Create the **Modified Cluster** by duplicating the first cluster and relabeling it. Right-click the shell of **Modified Cluster**, and select **Change to Indicator** from the shortcut menu.

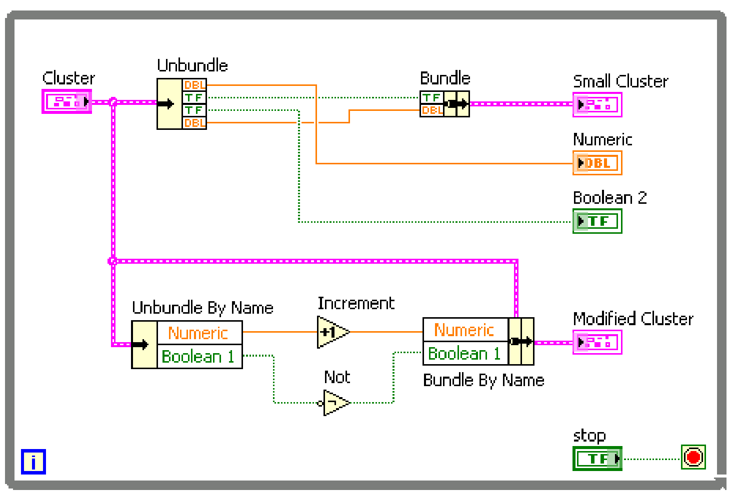
e. Copy **Modified Cluster** and relabel it to create **Small Cluster**. Remove the second toggle switch and horizontal fill slide indicators. Relabel the numeric indicator to `Slide value`. Resize the cluster as shown in the previous front panel.

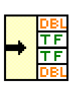
2. Verify the cluster order of **Cluster** and **Small Cluster**. **Modified Cluster** should have the same order as **Cluster**.
   a. Right-click the boundary of each cluster and select **Reorder Controls in Cluster** from the shortcut menu.
   b. Confirm the following cluster orders.



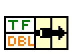
Block Diagram

3. Build the following block diagram.

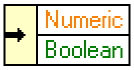




   a. Place the Unbundle function, located on the **Functions»All Functions»Cluster** palette, on the block diagram. This function disassembles Cluster. Use the Positioning tool to resize this function to four output terminals or wire the input cluster to resize the function automatically.



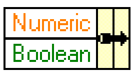
   b. Place the Bundle function, located on the **Functions»All Functions»Cluster** palette, on the block diagram. This function assembles Small Cluster.



c. Place the Unbundle by Name function, located on the **Functions» All Functions»Cluster** palette, on the block diagram. This function returns two elements from Cluster. Resize this function to have two output terminals. If a label name is not correct, right-click the name and select the correct name from the **Select Item** shortcut menu.

d. Place the Increment function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram. This function adds one to the value of Numeric.

e. Place the Not function, located on the **Functions»Arithmetic & Comparison»Express Boolean** palette, on the block diagram. This function returns the logical opposite of the value of the **Boolean** terminal of the Unbundle by Name function.



f. Place the Bundle by Name function, located on the **Functions» All Functions»Cluster** palette, on the block diagram. This function replaces the values of Numeric and Boolean in Cluster and creates Modified Cluster. Resize this function to have two input terminals. If a label name is not correct, right-click the name and select the correct name from the **Select Item** shortcut menu.

g. Complete the block diagram and wire the objects as shown in the previous figure.

4. Save the VI as `Cluster Exercise.vi` in the `C:\Exercises\LV Basics I` directory.

Run the VI

5. Display the front panel and run the VI.
6. Enter different values in Cluster and run the VI again. Notice how values entered in Cluster affect the Modified Cluster and Small Cluster indicators. Is this the behavior you expected?
7. Try changing the cluster order of Modified Cluster. Run the VI. How did the changed order affect the behavior?
8. Close the VI. Do not save changes.

End of Exercise 4-4

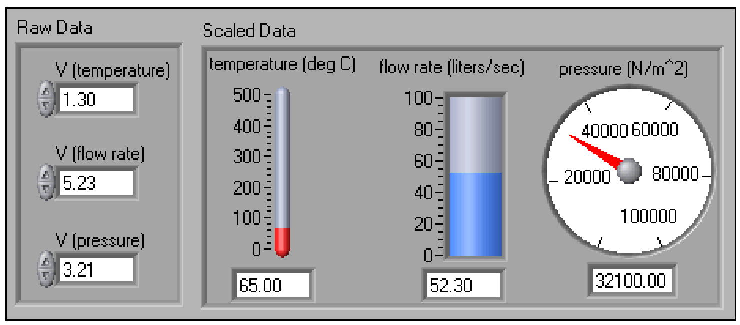
Exercise 4-5: Cluster Scaling VI

Objective: To build a VI that uses polymorphism with clusters.

Complete the following steps to build a VI that scales values stored in a cluster, where each cluster element has a different scale factor. Assume that the voltages were measured from transducers that measure the pressure, flow rate, and temperature. The VI then scales these values to get the actual values present in the system.

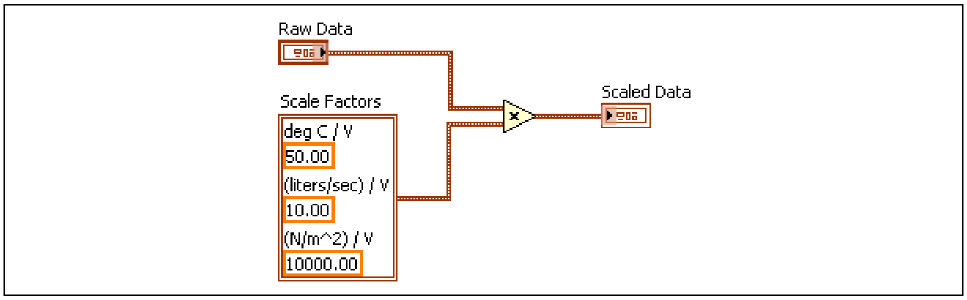
Front Panel

1. Open the Cluster Scaling VI located in the `C:\Exercises\LV Basics I` directory. The front panel is already built.
2. Change the controls as shown in the following front panel.



Block Diagram

3. Build the following block diagram. Make sure you apply the correct scale factors to each element in the Raw Data cluster.

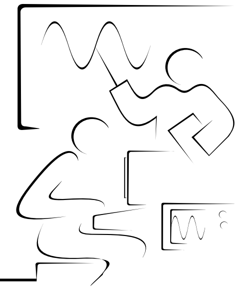


4. Save the VI.
5. Display the front panel and run the VI.
6. Change the front panel controls and run the VI again.
7. Close the VI.

End of Exercise 4-5

Lesson 5

Case and Sequence Structures



Exercises **Page**

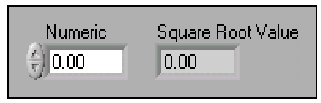
Exercise 5-1: Square Root VI

Objective: To use the Case structure in a VI.

Complete the following steps to build a VI that checks whether a number is positive. If the number is positive, the VI calculates the square root of the number. Otherwise, the VI returns an error message.

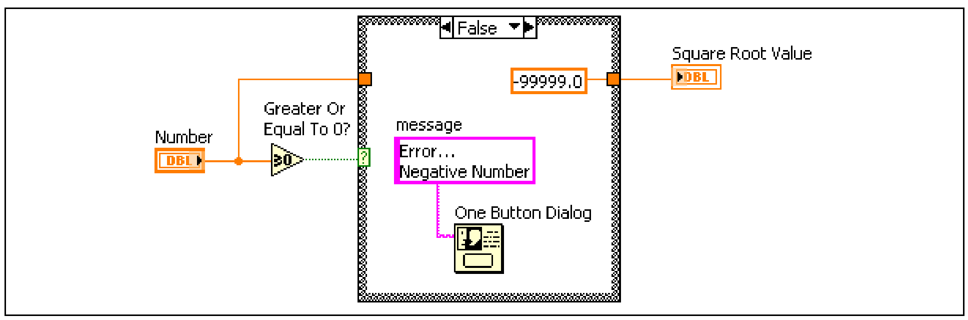
Front Panel

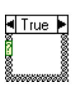
1. Open a blank VI and build the following front panel.



Block Diagram

2. Build the following block diagram.

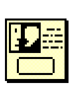




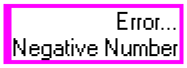
a. Place the Case structure, located on the **Functions»Execution Control** palette, on the block diagram. Click the decrement or increment button to select the FALSE case.

b. Place the Greater or Equal to 0? function, located on the **Functions» Arithmetic & Comparison»Express Comparison** palette, on the block diagram. This function returns TRUE if **Numeric** is greater than or equal to `0`.

c. Right-click the numeric constant and select **Properties** from the shortcut menu. Select the **Format and Precision** tab. Set **Digits of precision** to `1`, select **Floating point** notation, and click the **OK** button to ensure there is no data conversion between the constant and the numeric indicator outside the Case structure.



d. Place the One Button Dialog function, located on the **Functions» All Functions»Time & Dialog** palette, on the block diagram. This function displays a dialog box that contains the message `Error...Negative Number`.

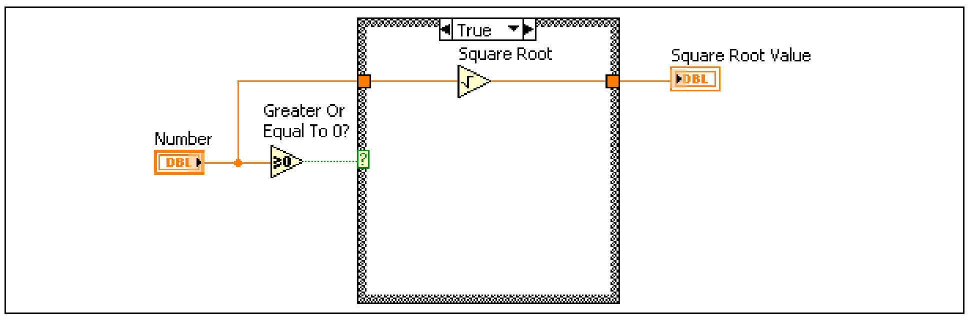


e. Right-click the **message** terminal of the One Button Dialog function, select **Create»Constant** from the shortcut menu, type `Error...Negative Number` in the constant and click the **Enter** button on the toolbar or click outside the control. Refer to Lesson6, *Strings and File I/O* for more information about strings.

f. Complete the diagram as shown in the previous figure.

3. Select the TRUE case of the Case structure.

Place the Square Root function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram. This function returns the square root of **Numeric**. Wire the function as shown in the following block diagram.



4. Save the VI as `Square Root.vi` in the `C:\Exercises\LV Basics I` directory.

Run the VI

5. Display the front panel and run the VI.

Caution Do *not* run this VI continuously. Under certain circumstances, continuously running this VI could result in an endless loop.

If **Numeric** is positive, the VI executes the TRUE case and returns the square root of **Numeric**. If **Numeric** is negative, the VI executes the FALSE case, returns `-99999`, and displays a dialog box with the message **`Error...Negative Number`**.

6. Close the VI.

End of Exercise 5-1

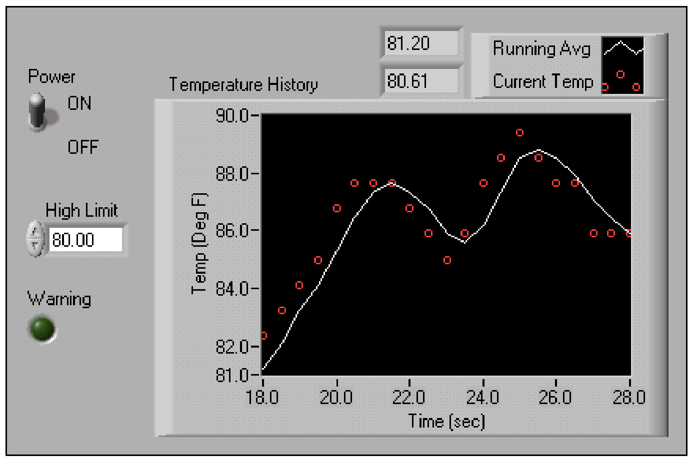
Exercise 5-2: Temperature Control VI

Objective: To use the Case structure.

Complete the following steps to build a VI that detects when a temperature is out of range. If the temperature exceeds the limit, an LED turns on and a beep sounds.

Front Panel

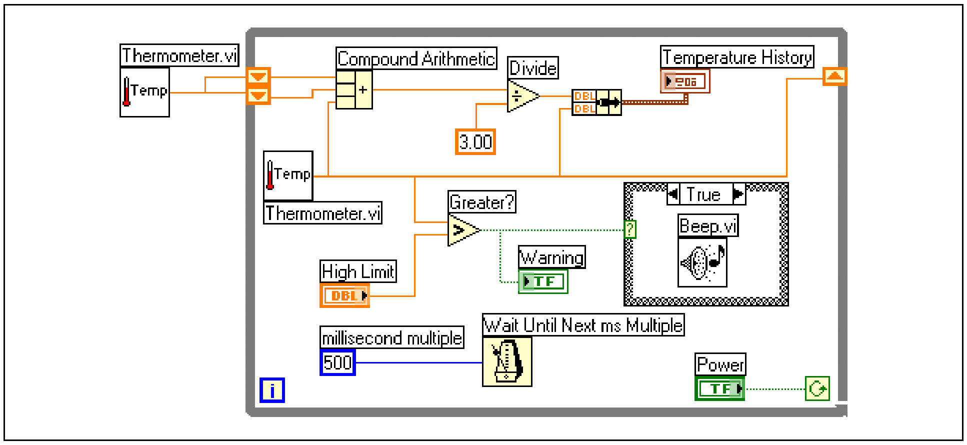
1. Open the Temperature Running Average VI, which you built in Exercise 3-4.
2. Modify the front panel as follows.



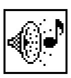
3. Right-click the chart display and select **Visible Items»Digital Display** from the shortcut menu to display the digital values.
4. Save the VI as `Temperature Control.vi`.

Block Diagram

5. Modify the block diagram as follows. The FALSE case of the Case structure is empty.



a. Place the Greater? function located on the **Functions»Arithmetic & Comparison»Express Comparison** palette. This function returns TRUE if the temperature exceeds **High Limit**. Otherwise, the function returns FALSE.



b. Place the Beep VI located on the **Functions»All Functions» Graphics & Sound»Sound** palette. This VI sounds a beep if the selector terminal of the Case structure receives TRUE.

c. **(MacOS)** Provide values for the Beep VI input terminals.

6. Save the VI because you will use this VI later in the course.

7. Display the front panel, enter `80` in **High Limit**, and run the VI.

   If the VI returns a temperature greater than **High Limit**, **Warning** turns on, the VI executes the TRUE case, and a beep sounds. If the temperature is less than **High Limit**, **Warning** turns off, the VI executes the FALSE case, and no beep sounds.

8. Close the VI.

End of Exercise 5-2

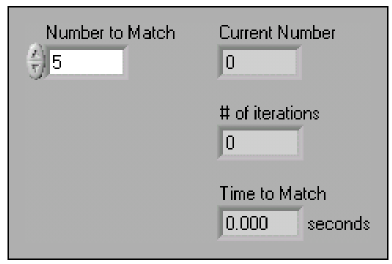
Exercise 5-3: Time to Match VI

Objective: To use the Sequence structure.

Complete the following steps to build a VI that computes the time it takes to generate a random number that matches a number you specify.

Front Panel

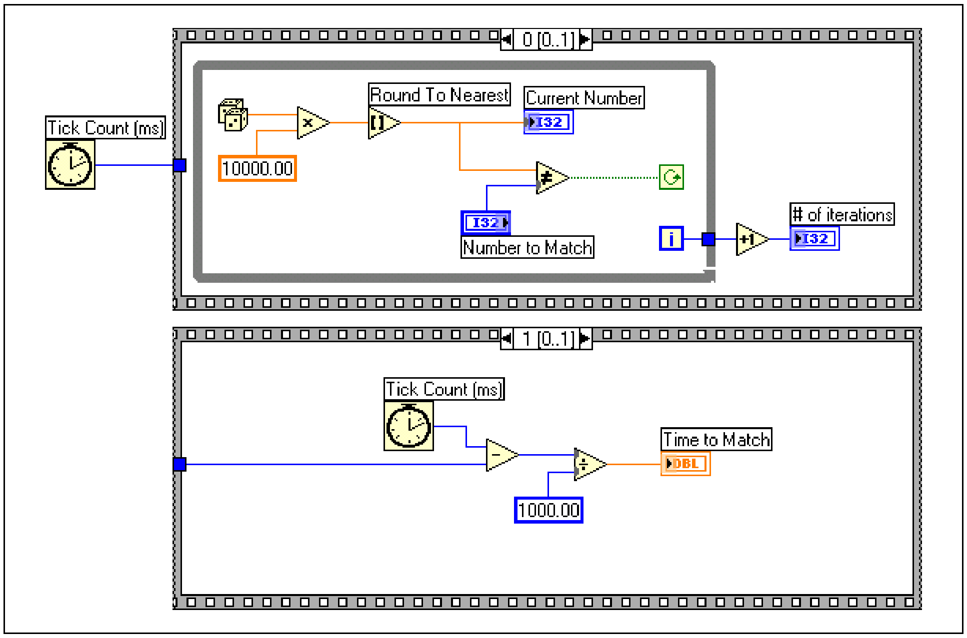
1. Open the Auto Match VI, which you built in Exercise 3-1.
2. Modify the front panel as follows.



   a. Change **Number to Match**, **Current Number**, and **# of iterations** to I32 representation.
   b. Change **Time to Match** to DBL representation and 3 digits of precision.
3. Save the VI as `Time to Match.vi`.

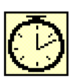
Block Diagram

4. Modify the block diagram as follows.



a. Place a Stacked Sequence structure located on the **Functions»All Functions»Structures** palette.

b. Right-click the structure border and select **Add Frame After** from the shortcut menu to add a frame.



c. Place the Tick Count (ms) function located on the **Functions»All Functions»Time & Dialog** palette. This function reads the current value of the operating system clock and returns the value in milliseconds.

5. Save the VI.

6. Display the front panel, enter a number in **Number to Match**, and run the VI.

In frame 0, the VI executes the While Loop while **Current Number** does not match **Number to Match**. In frame 1, the Tick Count (ms) function reads the operating system clock. The VI subtracts the new value from the initial time read and returns the elapsed time in seconds.

Note If **Time to Match** is always `0.000`, the VI might be running too quickly. Either run the VI with execution highlighting enabled or increase the numeric constant wired to the Multiply function in frame 0 to a large value, such as `1000000`.

7. Close the VI.

End of Exercise 5-3

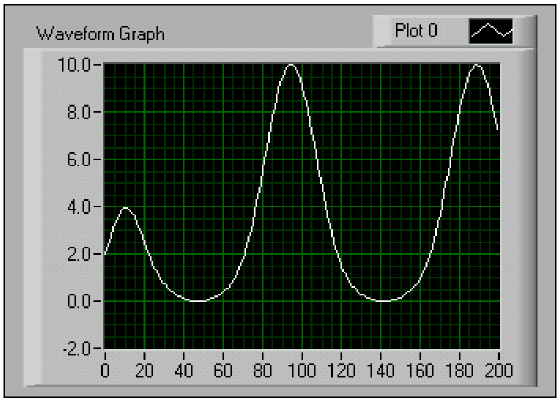
Exercise 5-4: Formula Node VI

Objective: To use the Formula Node in a VI.

Complete the following steps to build a VI that uses the Formula Node to perform a complex mathematical operation and graphs the results.

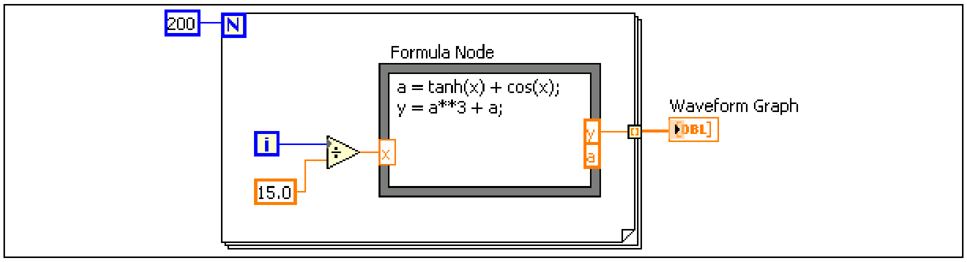
Front Panel

1. Open a blank VI and build the following front panel.



Block Diagram

1. Build the following block diagram.



   a. Place the Formula Node, located on the **Functions»All Functions» Structures** palette, on the block diagram.

   b. Create the **x** input terminal by right-clicking the left border and selecting **Add Input** from the shortcut menu. Type `x` into the box that appears.

   c. Create the **y** and **a** output terminals by right-clicking the right border and selecting **Add Output** from the shortcut menu. Enter `y` and `a`, respectively, in the boxes that appear. You must create output terminals for temporary variables like *a*.

Note When you create an input or output terminal, you must use a variable name that exactly matches the one in the equation. Variable names are case sensitive.

d. Type the following equations in the Formula Node, where `**` is the exponentiation operator. Refer to the *LabVIEW Help* for more information about syntax for the Formula Node.

```
a = tanh(x) + cos(x);
y = a**3 + a;
```

e. Complete the block diagram as shown.

2. Save the VI as `Formula Node Exercise.vi` in the `C:\Exercises\LV Basics I` directory.

Run the VI

3. Display the front panel and run the VI. The graph displays the plot of the equation $y = f(x)^3 + f(x)$, where $f(x) = \tanh(x) + \cos(x)$.

   During each iteration, the VI divides the iteration terminal value by 15.0. The quotient is wired to the Formula Node, which calculates the function value. The VI plots the array as a graph.

4. Close the VI.

End of Exercise 5-4

Lesson 6

Strings and File I/O

Exercises **Page**

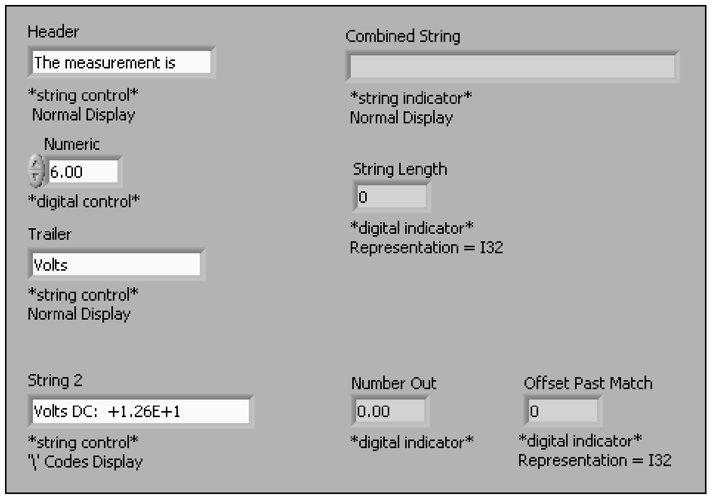
Exercise 6-1: Build String VI

Objective: To use the Format Into String, Concatenate Strings, and String Length functions.

Complete the following steps to build a VI that converts a numeric to a string, concatenates the string to other strings to form a single output string, and determines the output string length. The VI also matches a pattern in a string and converts the remaining string to a numeric.

Front Panel

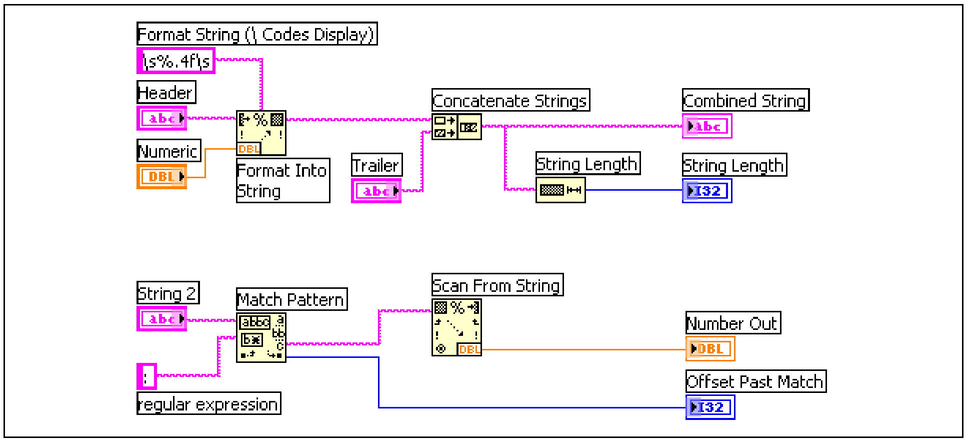
1. Open a new VI and build the following front panel.



   a. Right-click **String 2** and select **'\' Codes Display** from the shortcut menu.
   b. Change **String Length** and **Offset Past Match** to I32 representation.

Block Diagram

2. Build the following block diagram.



a. Place the Format Into String function located on the **Functions»All Functions»String** palette. This function converts **Number** to a string.

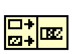
b. Right-click the Format Into String function and select **Edit Format String** from the shortcut menu to display the **Edit Format String** dialog box.

c. Place a checkmark in the **Use specified precision** checkbox and type `4` in the corresponding text box to create a **format string** that converts **Number** to a string with four digits after the decimal point.

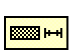
d. Click the **OK** button. LabVIEW creates a **format string** of `%.4f` using the options you selected.

e. Use the Labeling tool to type a space on either side of the `%.4f` constant so **Number** will appear with spaces on either side in **Combined String**.

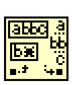
f. Right-click the constant and select **'\' Codes Display** from the shortcut menu. The spaces you typed change to `\s`.



g. Place the Concatenate Strings function located on the **Functions» All Functions»String** palette. This function concatenates input strings into a single output string.

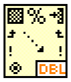


h. Place the String Length function located on the **Functions»All Functions»String** palette. This function returns the number of characters in **Combined String**.



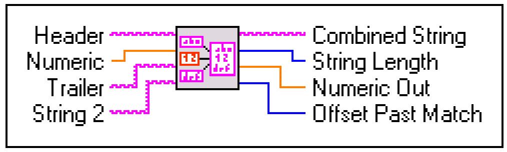
i. Place the Match Pattern function located on the **Functions»All Functions»String** palette. This function searches **String 2** for a colon.

j. Right-click the **regular expression** input terminal, select **Create» Constant** from the shortcut menu and type a colon (`:`).



k. Place the Scan from String function located on the **Functions»All Functions»String** palette. This function converts the string after the colon to a numeric.

3. Display the front panel and create the following icon and connector pane so you can use the VI as a subVI later in this course. Refer to Lesson 2, *Modular Programming*, for more information about creating icons and connector panes.



4. Save the VI as `Build String.vi` because you will use this VI later in the course.
5. Change the values of the front panel controls and run the VI.

   The VI concatenates **Header**, **Number**, and **Trailer** into **Combined String** and displays the string length.

   The VI also searches **String 2** for a colon, converts the string following the colon to **Numeric Out**, and displays the index of the first character after the colon in **Offset Past Match**.
6. Save and close the VI.

End of Exercise 6-1

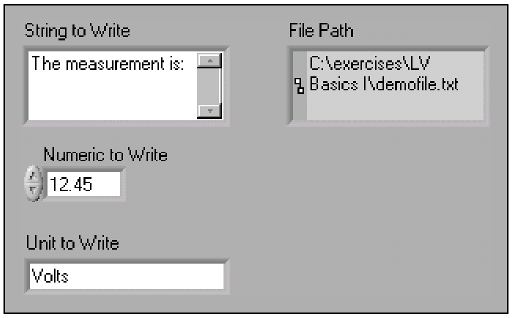
Exercise 6-2: File Writer VI

Objective: To write data to a file.

Complete the following steps to build a VI that concatenates a message string, a numeric value, and a unit string to a file. In Exercise 6-3, you will build a VI that reads the file and displays its contents.

Front Panel

1. Open a blank VI and build the following front panel.

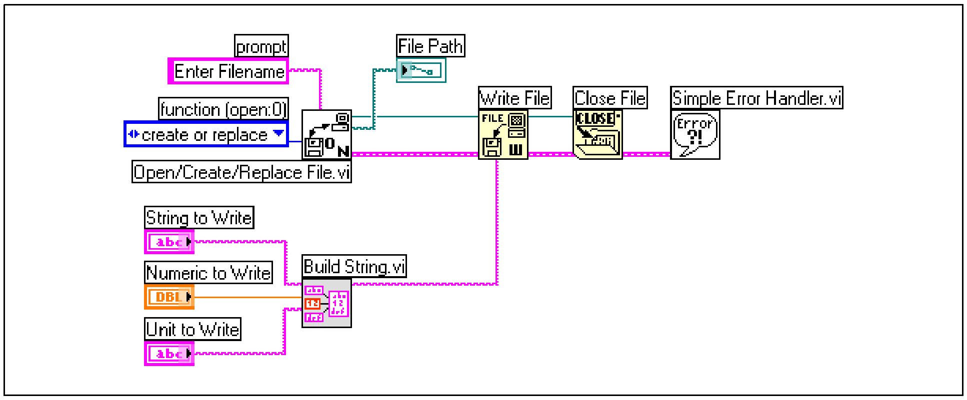


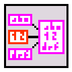
   a. Place a path indicator located on the **Controls»Text Indicators** palette on the front panel. This indicator displays the path for the data file you create.

   b. Right-click the **String to Write** control and select **Visible Items» Scrollbar** from the shortcut menu to display a scrollbar.

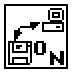
Block Diagram

2. Build the following block diagram.





a. Place the Build String VI from Exercise 6-1 on the block diagram. Select **Functions»All Functions»Select a VI** and navigate to `C:\Exercises\LV Basics I\Build String.vi`. This subVI concatenates the three input strings to one combined string.

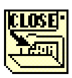


b. Place the Open/Create/Replace File VI, located on the **Functions» All Functions»File I/O** palette, on the block diagram. This VI displays a dialog box to open or create a file.

   Right-click the **prompt** input, select **Create»Constant** from the shortcut menu, and type `Enter Filename` in the constant. When the VI runs, a file navigation dialog box appears with `Enter Filename` as the title of the window.

   Right-click the **function** input, select **Create»Constant** from the shortcut menu, and click the constant with the Operating tool to select **create or replace**.

c. Place the Write File function, located on the **Functions»All Functions»File I/O** palette, on the block diagram. This function writes the concatenated strings to the file.



d. Place the Close File function, located on the **Functions»All Functions»File I/O** palette, on the block diagram. This function closes the file.

e. Place the Simple Error Handler VI, located on the **Functions»All Functions»Time & Dialog** palette, on the block diagram. This function checks the error cluster and displays a dialog box if an error occurs.

f. Complete the block diagram as shown in the previous figure.

3. Save the VI as `File Writer.vi` in the `C:\Exercises\LV Basics I` directory.

Run the VI

4. Change the values of the front panel controls and run the VI. An **Enter Filename** dialog box appears.
5. Type `demofile.txt` and click the **Save** or **OK** button to save the file.

   The VI writes the **String to Write**, **Numeric to Write**, and **Unit to Write** values to the file.
6. Close the VI.

End of Exercise 6-2

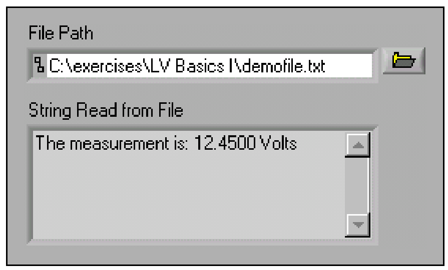
Exercise 6-3: File Reader VI

Objective: To build a VI that reads data from a file.

Complete the following steps to build a VI that reads the file created in Exercise 6-2 and displays the information in a string indicator.

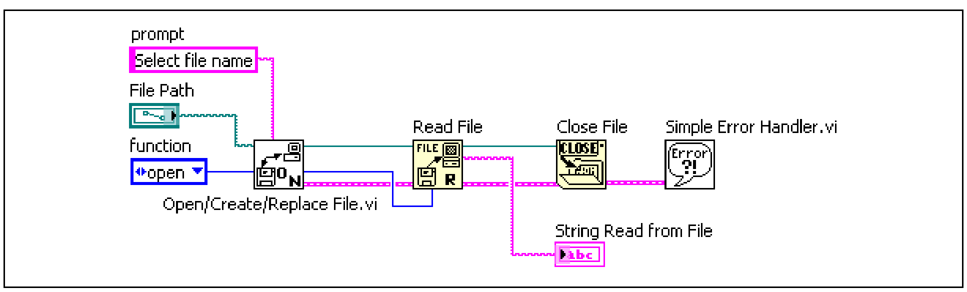
Front Panel

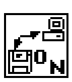
1. Open a blank VI and build the following front panel using the file path control located on the **Controls»Text Controls** palette and a string indicator located on the **Controls»Text Indicators** palette.



Block Diagram

2. Build the following block diagram.





a. Place the Open/Create/Replace File VI, located on the **Functions» All Functions»File I/O** palette, on the block diagram. This VI displays a dialog box that you use to open or create a file. Right-click the **prompt** input, select **Create»Constant** from the shortcut menu, and type `Select Filename` in the constant. Right-click the **function** input, select **Create»Constant** from the shortcut menu, and click the constant with the Operating tool to select **open**.

b. Place the Read File function, located on the **Functions»All Functions»File I/O** palette, on the block diagram. This function reads **count** bytes of data from the file starting at the beginning of the file.

c. Place the Close File function, located on the **Functions»All Functions»File I/O** palette, on the block diagram. This function closes the file.

d. Place the Simple Error Handler VI, located on the **Functions»All Functions»Time & Dialog** palette, on the block diagram. This VI checks the error cluster and displays a dialog box if an error occurs.

e. Complete the block diagram as shown in the previous figure.

3. Save the VI as `File Reader.vi` in the `C:\Exercises\LV Basics I` directory.

Run the VI

4. Display the front panel and use the Operating tool to click the **Browse** button in the path control.
5. Navigate to `demofile.txt` and click the **Open** or **OK** button.
6. Run the VI. **String Read from File** displays the contents of the file.
7. If time permits, complete the following challenge step. Otherwise, save and close the VI.

Challenge

8. Modify the VI so it parses the numeric value and displays the numeric value in a numeric indicator. Save and close the VI.

Tip Use the Match Pattern function to search for the first numeric character.

End of Exercise 6-3

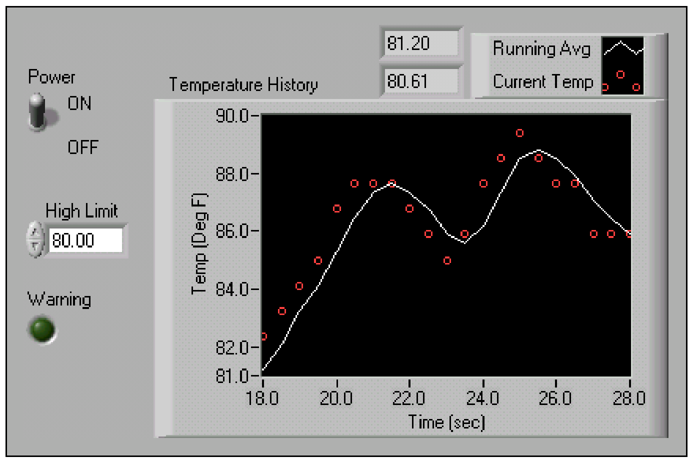
Exercise 6-4: Temperature Logger VI

Objective: To save data to a file in a form that a spreadsheet or a word processor can access.

Complete the following steps to save the time and current temperature to a data file.

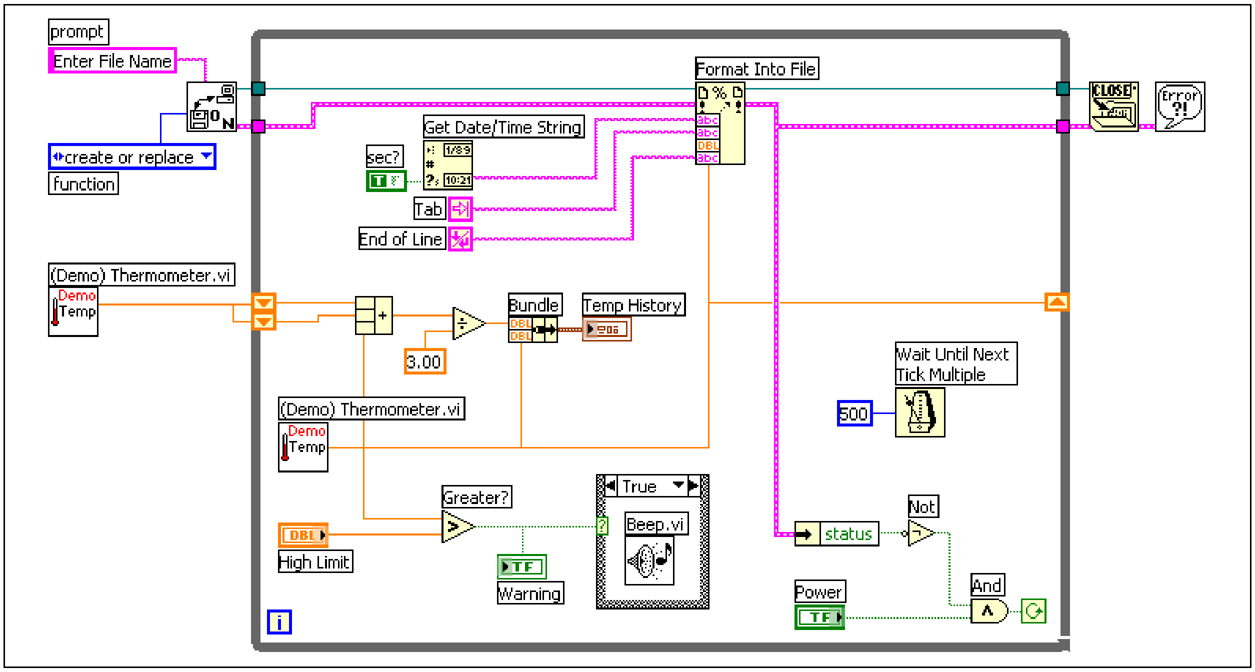
Front Panel

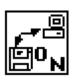
1. Open the Temperature Control VI, which you built in Exercise 5-2, and save it as `Temperature Logger.vi`. You do not need to modify the following front panel.



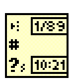
Block Diagram

2. Modify the block diagram as follows.



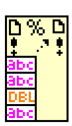


a. Place the Open/Create/Replace File VI located on the **Functions»All Functions»File I/O** palette. This VI displays a dialog box that you use to open or create a file.

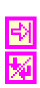


b. Place the Get Date/Time String function located on the **Functions»All Functions»Time & Dialog** palette. This function returns the time, in string format, when the temperature measurement was taken.

c. Place the TRUE Boolean constant located on the **Functions»Arithmetic & Comparison»Express Boolean** palette. This constant sets the function to include seconds in the string.

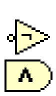


d. Place the Format Into File function located on the **Functions»All Functions»File I/O** palette. This function converts the temperature measurement to a string and builds and writes to file a formatted data string.



e. Place the tab constant and end of line constant located on the **Functions»All Functions»String** palette.

f. Place the Unbundle by Name function located on the **Functions»All Functions»Cluster** palette. This function removes **status** from the error cluster.



g. Place the Not and And functions located on the **Functions»Arithmetic & Comparison»Express Boolean** palette. The Not and

And functions set the conditional terminal to continue while **Power** is TRUE and no error occurs.

h. Place the Close File function located on the **Functions»All Functions»File I/O** palette. This function closes the file.

i. Place the Simple Error Handler VI located on the **Functions»All Functions»Time & Dialog** palette. This VI checks the error cluster and displays a dialog box if an error occurs.

3. Save the VI because you will use this VI later in the course.
4. Run the VI. An **Enter Filename** dialog box appears.
5. Type `temp.txt` and click the **Save** or **OK** button.

   The VI creates a file called `temp.txt`. The VI takes readings every half-second and saves the time and temperature data to a file until you click the **Power** switch or an error occurs. When the VI finishes, it closes the file.
6. Close the VI.
7. Open a word processor or spreadsheet application, such as **(Windows)** Notepad or WordPad, **(MacOS)** SimpleText, or **(UNIX)** Text Editor.
8. Open `temp.txt`. The time appears in the first column, and the temperature data appears in the second column.
9. Exit the word processor or spreadsheet application and return to LabVIEW.

End of Exercise 6-4

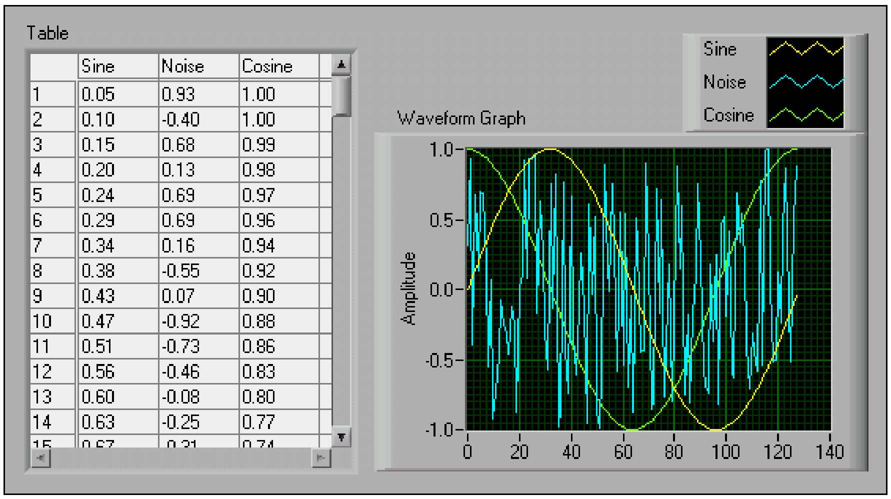
Exercise 6-5: Spreadsheet Example VI

Objective: To save a 2D array in a text file so a spreadsheet application can access the file and to display numeric data in a table.

Complete the following steps to examine a VI that saves numeric arrays to a file in a format you can access with a spreadsheet.

Front Panel

1. Open the Spreadsheet Example VI located in the `C:\Exercises\LV Basics I` directory. The following front panel is already built.



Run the VI

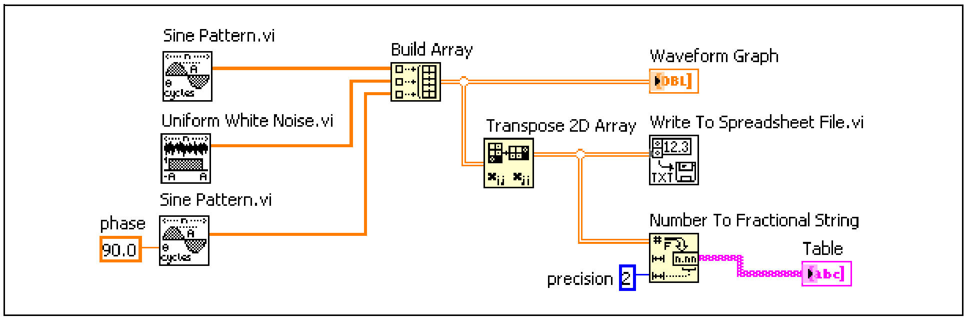
2. Run the VI.

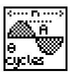
   The VI generates a 2D array of 128 rows × 3 columns. The first column contains data for a sine waveform, the second column contains data for a noise waveform, and the third column contains data for a cosine waveform. The VI plots each column in a graph and displays the data in a table.

3. When the **Choose file to write** dialog box appears, save the file as `wave.txt` in the `C:\Exercises\LV Basics I` directory and click the **OK** button. Later, you will examine this file.

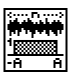
Block Diagram

4. Display and examine the block diagram for this VI.

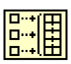




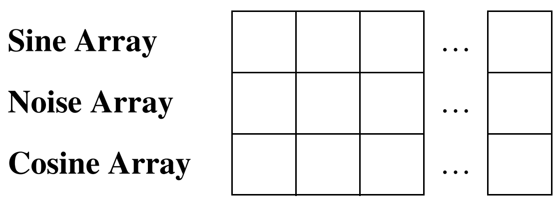
The Sine Pattern VI located on the **Functions»All Functions»Analyze» Signal Processing»Signal Generation** palette returns a numeric array of 128 elements containing a sine pattern. The constant `90.0`, in the second instance of the Sine Pattern VI specifies the phase of the sine pattern or cosine pattern.

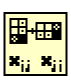


The Uniform White Noise VI located on the **Functions»All Functions» Analyze»Signal Processing»Signal Generation** palette returns a numeric array of 128 elements containing a noise pattern.

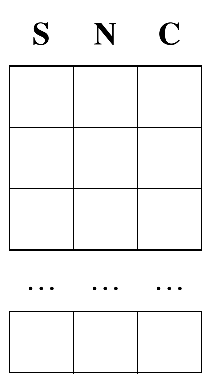


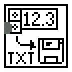
The Build Array function located on the **Functions»All Functions» Array** palette builds the following 2D array from the sine array, noise array, and cosine array.



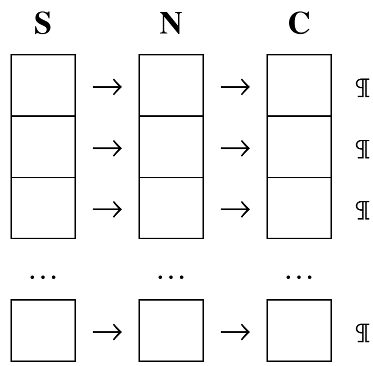


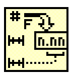
The Transpose 2D Array function located on the **Functions»All Functions»Array** palette rearranges the elements of the 2D array so element `[i,j]` becomes element `[j,i]`, as follows.





The Write To Spreadsheet File VI located on the **Functions»All Functions»File I/O** palette formats the 2D array into a spreadsheet string and writes the string to a file. The string has the following format, where an arrow (→) indicates a tab, and a paragraph symbol (¶) indicates an end of line character.





The Number To Fractional String function located on the **Functions»All Functions»String»String/Number Conversion** palette converts an array of numeric values to an array of strings that the table displays.

5. Close the VI. Do not save changes.

Note This example stores only three arrays in the file. To include more arrays, increase the number of inputs to the Build Array function.

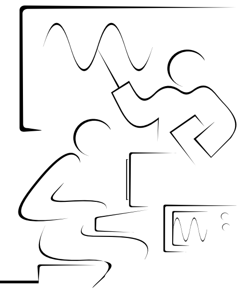
Optional

Open the `wave.txt` file using a word processor or spreadsheet application and view its contents.

6. Open a word processor or spreadsheet application, such as **(Windows)** Notepad or WordPad, **(Mac OS)** SimpleText, or **(UNIX)** Text Editor.
7. Open `wave.txt`. The sine waveform data appear in the first column, the random waveform data appear in the second column, and the cosine waveform data appear in the third column.
8. Exit the word processor or spreadsheet application and return to LabVIEW.

End of Exercise 6-5

Lesson 7



Data Acquisition and Waveforms

Exercises **Page**

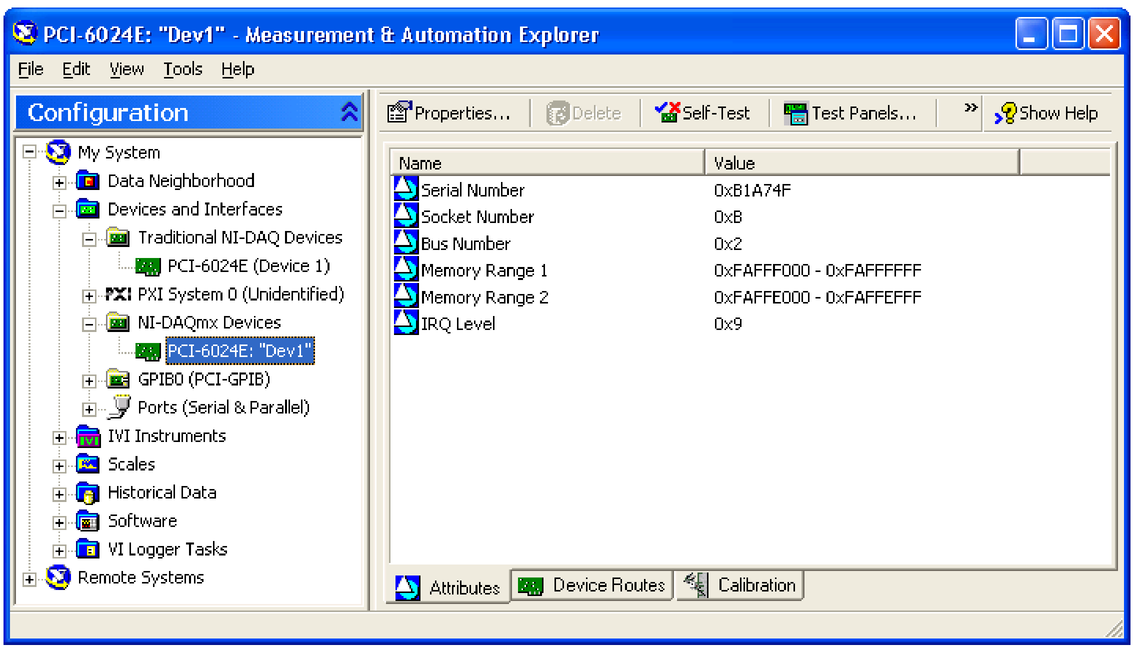
Exercise 7-1: Measurement & Automation Explorer

Objective: To use MAX to examine the current DAQ configuration and test the device interactively.

Complete the following steps to examine the configuration for the DAQ device in the computer using MAX and use the test routines in MAX to confirm operation of the device.

Part A. Examining the DAQ Device Settings

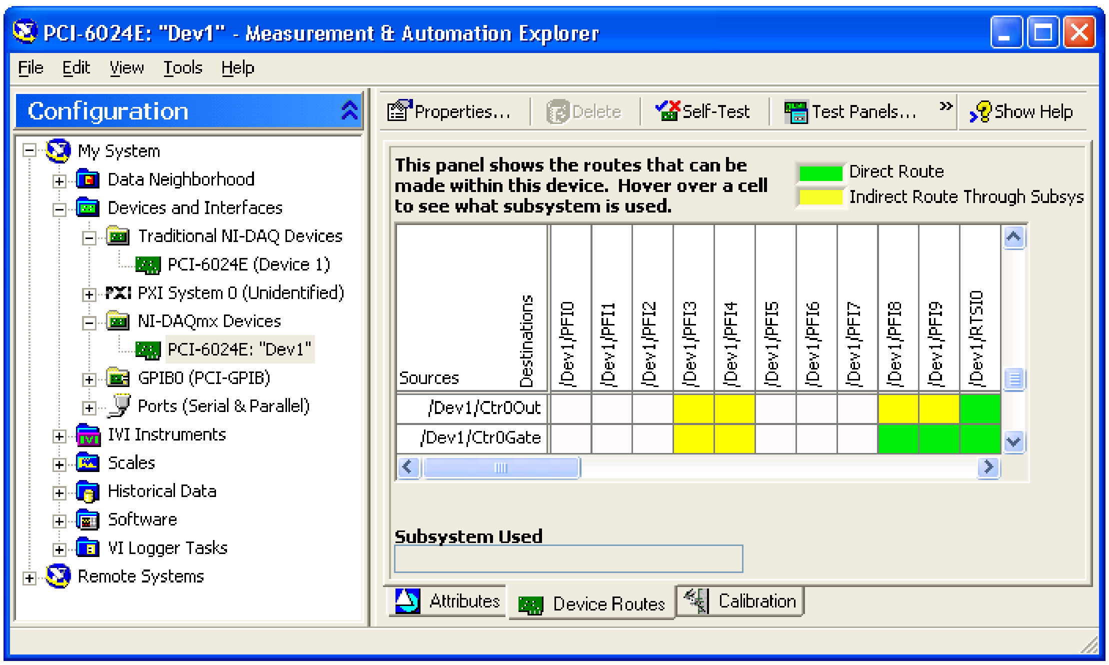
1. Launch MAX by double-clicking the icon on the desktop or by selecting **Tools»Measurement & Automation Explorer** in LabVIEW. The utility searches the computer for installed National Instruments hardware and displays the information.
2. Expand the **Devices and Interfaces** section to view the installed National Instruments devices. The following example shows the PCI-6024E and a PCI-GPIB device.



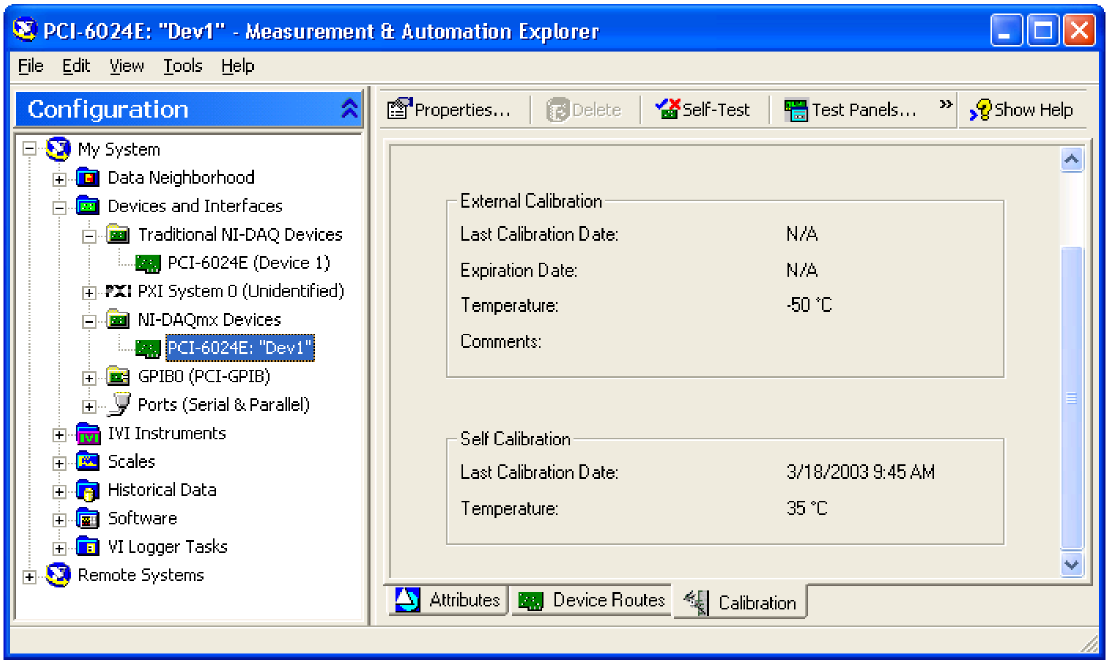
MAX displays the National Instruments hardware and software in the computer. The device number appears in quotes following the device name. The Data Acquisition VIs use this device number to determine which device performs DAQ operations. MAX also displays the attributes of the device such as the system resources that are being used by the device.

Note You might have a different device installed, and some of the options shown might be different. Click the **Show Help/Hide Help** button in the top right corner of MAX to hide the online help and show the DAQ device information.

3. The **Device Routes** tab provides detailed information about the internal signals that can be routed to other destinations on the device. This is a powerful resource that gives you a visual representation of the signals that are available to provide timing and synchronization with components that are on the device and other external devices.



4. The **Calibration** tab provides information about the last time the device was calibrated both internally and externally.



5. Right-click the NI-DAQmx device in the configuration tree and select **Self Calibrate** to calibrate the DAQ device using a precision voltage reference source and update the built-in calibration constants. Once the device has been calibrated, the **Self Calibration** information updates in the **Calibration** tab.

Part B. Testing the DAQ Device Components

6. Click the **Self-Test** button to test the device. This tests the system resources assigned to the device. The device should pass the test because it is already configured.
7. Click the **Test Panels** button to test the individual functions of the DAQ device, such as analog input and output. The **Test Panels** dialog box appears.
   a. Use the **Analog Input** tab to test the various analog input channels on the DAQ device. Channel Dev1/ai0 is connected to the temperature sensor on the DAQ Signal Accessory. Click the **Start** button to acquire data from analog input channel 0. Place your finger on the sensor to see the voltage rise. You also can move the **Noise** switch to **On** on the DAQ Signal Accessory to see the signal change in this tab. When you are finished click the **Stop** button.

b. Click the **Analog Output** tab to set up a single voltage or sine wave on one of the DAQ device analog output channels.

   Change the **Output Mode** to **Sinewave Generation** and click the **Start Sine Generator** button. LabVIEW generates a continuous sine wave on analog output channel 0.

c. On the external DAQ Signal Accessory box, wire Analog Out Ch0 to Analog In Ch1.

d. Click the **Analog Input** tab and change the channel to Dev1/ai1. Click the **Start** button to acquire data from analog input channel 1. LabVIEW displays the sine wave from analog output channel 0.

e. Click the **Digital I/O** tab to test the digital lines on the DAQ device.

f. Set lines 0 through 3 as output and toggle the **Logic Level** checkboxes. As you toggle the boxes, the LEDs on the DAQ signal accessory turn on or off. The LEDs use negative logic.

g. Click the **Close** button to close the **Test Panel** and return to MAX.

8. Click the **Counter I/O** tab to determine if the DAQ device counter/timers are functioning properly. To verify counter/timer operation, change the **Counter Mode** tab to **Edge Counting** and click the **Start** button. The **Counter Value** increments rapidly. Click **Stop** to stop the counter test.

9. Close MAX by selecting **File»Exit**.

End of Exercise 7-1

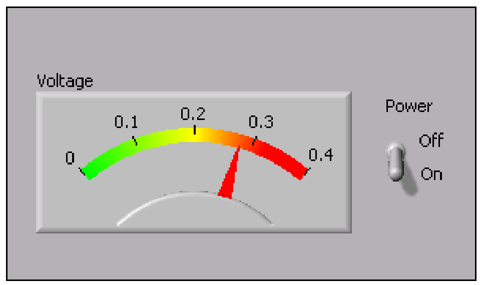
Exercise 7-2: Voltmeter VI

Objective: To acquire an analog signal using a DAQ device.

Complete the following steps to build a VI that measures the voltage that the temperature sensor on the DAQ Signal Accessory outputs. The temperature sensor outputs a voltage proportional to the temperature. The sensor is hard-wired to channel 0 of the DAQ device.

Front Panel

1. Open a blank VI and build the following front panel.



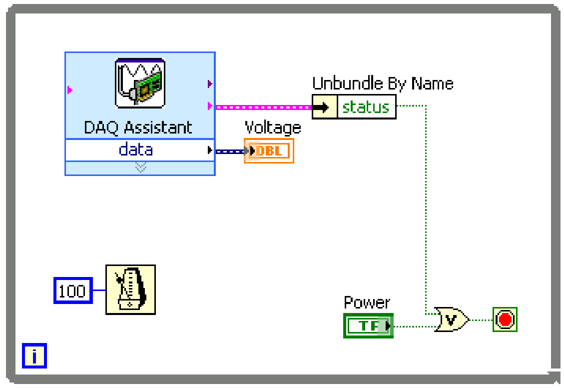
   a. Place the Meter, located on the **Controls»Numeric Indicators** palette, on the front panel. Configure the meter scale for `0.0` to `0.4`. Use the Labeling tool to double-click `10.0` and type `0.4`. You might need to enlarge the meter to display the scale as shown in the example.

   b. Place a Vertical Toggle Switch, located on the **Controls»Buttons & Switches** palette, on the front panel. Configure the toggle switch to a default value of FALSE and a mechanical action of **Latch When Pressed**.

   c. Create two free labels, **Off** and **On**, using the Labeling tool.

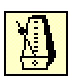
Block Diagram

2. Build the following block diagram.



a. Place the DAQ Assistant Express VI located on the **Functions» Input** palette, on the block diagram. Configure this VI to read an analog input channel and return the voltage.

- Select **Analog Input»Voltage** for the measurement to make.
- Select **Dev1»ai0** for the physical channel.
- Click the **Finish** button.
- The **Analog Input Voltage Task** dialog box appears. Configure the **Task Timing** to **Acquire 1 Sample**.
- Click the **OK** button to close the **Analog Input Voltage Task Configuration** dialog box. This saves the settings specified for the task in the DAQ Assistant Express VI.



b. Place the Wait Until Next ms Multiple function, located on the **Functions»All Functions»Time & Dialog** palette, on the block diagram. Right-click the input and select **Create Constant** from the shortcut menu. Type `100` in the constant to cause the loop to execute every 100 ms.

c. Place the Unbundle by Name function, located on the **Functions» All Functions»Cluster** palette, on the block diagram. Use this function to access the **status** from the error cluster.

d. Place the Or function, located on the **Functions»Arithmetic & Comparison»Express Boolean** palette, on the block diagram. This function stops the loop if an error occurs or the user clicks the power switch on the front panel.

3. Save the VI as `Voltmeter.vi` in the `C:\Exercises\LV Basics I` directory. You will use this VI later in the course.
4. Display the front panel and run the VI.

   The meter displays the voltage the temperature sensor outputs. Place your finger on the temperature sensor and notice that the voltage increases.
5. Stop the VI by clicking the power switch.

Scales

The temperature sensor on the DAQ Signal Accessory outputs the voltage in degrees Celsius, scaled by 100. In order to convert the voltage into degrees Celsius, it is necessary to multiply the voltage by 100. You could multiply the output of the DAQ Assistant Express VI by 100, or configure the DAQ Assistant Express VI to automatically scale the voltage. Using the capabilities that exist within the VI reduces block diagram clutter.

6. Double-click the DAQ Assistant to display the **Analog Input Voltage Task Configuration** dialog box.
7. Select **Create New** in the **Custom Scaling** pull-down menu.
8. Select **Linear** and name the scale `temperature`. Click the **Finish** button.
9. A dialog box appears where you can scale the data by a multiplier and an offset.
   a. Set the slope to **100** and the **Scaled Units** to **Celsius**.
   b. Click the **OK** button to close the dialog box.
10. In the **Analog Input Voltage Task Configuration** dialog box, set the minimum input range to `0`, set the maximum input range to `100`, and click the **OK** button to return to the block diagram.
11. Run the VI. The temperature displays in the meter. The temperature values are 100 times greater than the voltage values. Change the meter scale to see the correct values.
12. Stop the VI. Save the VI but do not close it. You will use the VI in the next exercise.

End of Exercise 7-2

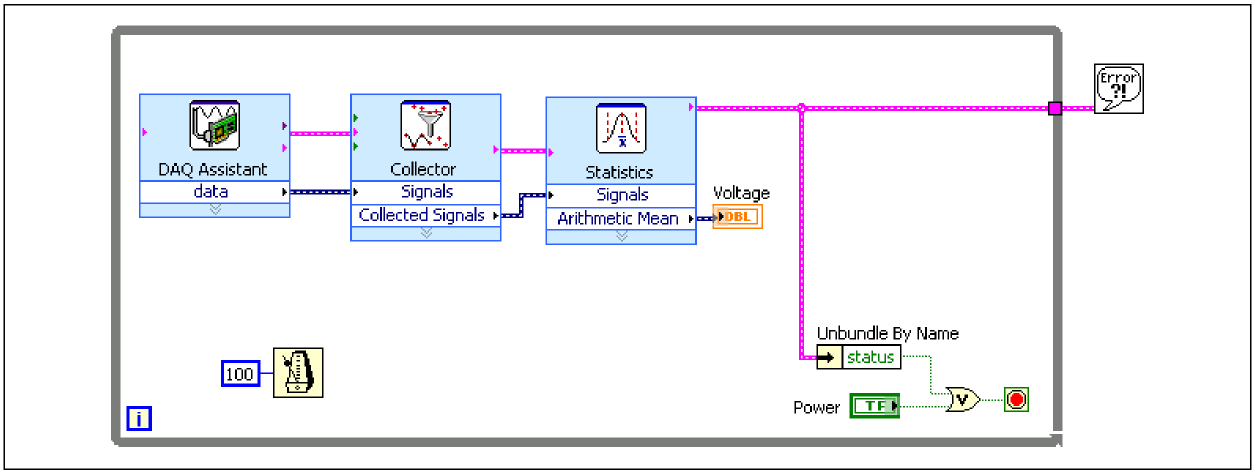
Exercise 7-3: Measurement Averaging VI

Objective: To reduce noise in analog measurements by averaging.

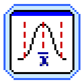
1. Run the Voltmeter VI that you completed in Exercise 7-2.
2. Introduce noise into the temperature measurement by changing the Temp Sensor Noise switch on the DAQ Signal Accessory to the ON position. The measurements begin to fluctuate with noise spikes.

Block Diagram

3. Stop the VI and display the block diagram. Modify the block diagram to calculate the average of 100 measurements.



a. Place the Collector Express VI located on the **Functions»Signal Manipulation** palette, on the block diagram. This Express VI creates an internal buffer to store the individual points. When the maximum number of input points is collected, the Express VI discards the oldest points and adds the newest points. In the **Configure Collector** dialog box that appears, set the **Maximum number of samples** to `100`. Click the **OK** button to close the dialog box.



b. Place the Statistics Express VI, located on the **Functions»Analysis** palette, on the block diagram. In the **Configure Statistics** dialog box that appears, place a checkmark in the **Arithmetic Mean** checkbox to perform averaging on the collected data. Click the **OK** button to close the dialog box.

4. Select **File»Save As** to save the VI as `Measurement Averaging.vi` in the `C:\Exercises\LV Basics I` directory.
5. Display the front panel and run the VI. Notice that the noise spikes are reduced when the Temp Sensor Noise switch is turned on.
6. Stop and close the VI.

End of Exercise 7-3

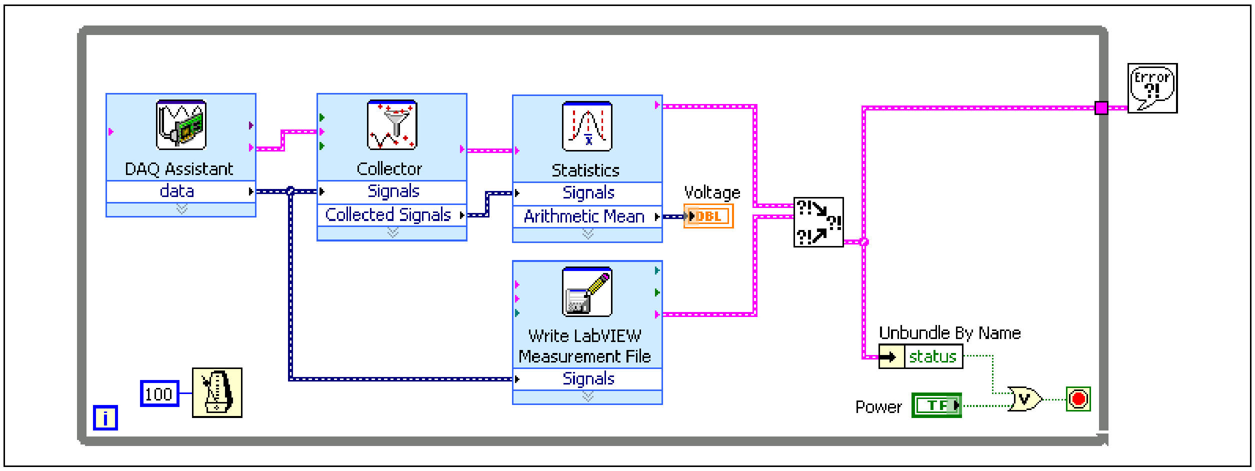
Exercise 7-4: Simple Data Logger VI

Objective: To learn to use a LabVIEW Measurement File.

Complete the following steps to modify the Measurement Averaging VI that you created in Exercise 7-3 to log the acquired data to a LabVIEW Measurement File. Create another VI that reads the data file.

Simple Data Logger Block Diagram

1. Open the Measurement Averaging VI located in the `C:\Exercises\LV Basics I` directory.
2. Modify the block diagram to log the acquired data as shown in the following figure.



Place the Write LabVIEW Measurement File Express VI, located on the **Functions»Output** palette, on the block diagram. This Express VI stores the data acquired from the DAQ device. In the **Configure Write LabVIEW Measurement File** dialog box that appears, set the following options:

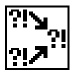
a. Set the **Action** to **Ask user to choose file** for the filename.

b. Set the **Segment Headers** to **One header only** to provide a header for all of the data. The header contains information about the sampling rate and the time when the sample was taken.

c. Set **X Value Columns** to **One column per channel** to provide a table of data that can be read by any spreadsheet editor or an ASCII text file editor.

d. Set the **Delimiter** to **Tab** to make it easy for a spreadsheet editor to determine where a column of data starts in the file.

e. Click the **OK** button to close the dialog box.

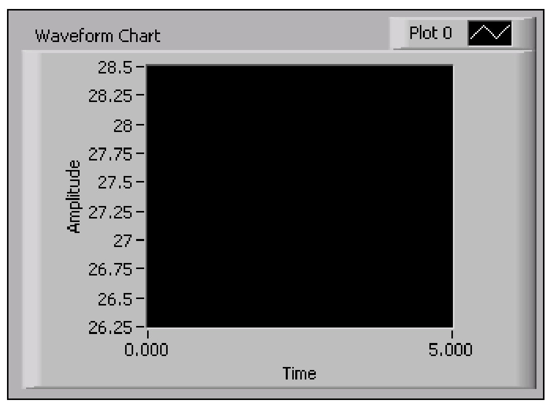


Place the Merge Errors VI, located on the **Functions»All Functions»Time & Dialog** palette, on the block diagram. It is important to catch errors with both DAQ and file I/O, and because the code has a parallel structure it is necessary to merge the errors from all of the parallel operations to determine if the code is functioning properly.

3. Select **File»Save As** to save the VI as `Simple Data Logger.vi` in the `C:\Exercises\LV Basics I` directory.
4. Run the VI. A filename prompt appears. Name the file `logger.lvm` in the `C:\Exercises\LV Basics I` directory.
5. Stop and close the VI.

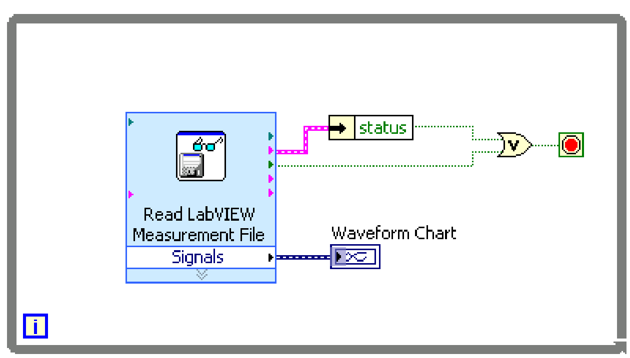
Simple Data Reader Front Panel

6. Open a blank VI and build the following front panel by placing a waveform chart, located on the **Controls»Graph Indicators** palette, on the front panel.



Block Diagram

7. Build the following block diagram.



a. Place the Read LabVIEW Measurement File Express VI, located on the **Functions»Input** palette, on the block diagram. Because this Express VI reads data located in a LabVIEW measurement file one data point at a time it must be placed in a loop. In the **Configure Read LabVIEW Measurement File** dialog box that appears, set the following options:

- In the **Action** section, place a checkmark in the **Ask user to choose file** checkbox.
- Set the **Segment Size** to **Retrieve segments of original size** so that all the data stored in the file is retrieved.
- Set **Time Stamps** to **Relative to start of measurement**. Because the dynamic data type stores information about the signal timing, this setting aligns the data with the time of the measurement.
- In the **Generic Text File** section, remove the checkmark from the **Read generic text files** checkbox because the data is stored in a LabVIEW measurement file.
- Click the **OK** button to close the dialog box.

b. Place the Unbundle by Name function, located on the **Functions»All Functions»Cluster** palette, on the block diagram. Wire the Error Out to the Unbundle by Name function.

c. Place the Or function, located on the **Functions»Arithmetic & Comparison»Express Boolean** palette, on the block diagram.

d. Wire the **EOF?** output to the Or function. The EOF? outputs a true value when the entire LabVIEW Measurement File has been read.

e. Finish wiring the block diagram as shown.

8. Save the VI as `Simple Data Reader.vi` in the `C:\Exercises\LV Basics I` directory.
9. Display the front panel, and run the VI. In the filename prompt that appears, select the `logger.lvm` file that you created in step 4.
10. The data that was stored in the LabVIEW Measurement File appears in the waveform chart.

Note You might need to rescale or autoscale the y-axis of the waveform chart to display the data.

11. Close the Simple Data Reader VI.

End of Exercise 7-4

Exercise 7-5: Voltage Output

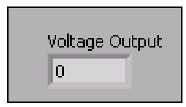
Objective: To output an analog voltage using a DAQ device.

Complete the following steps to finish a VI that outputs voltage from 0 to 9.5 V in 0.5 V steps.

1. Connect Analog Out CH0 to Analog In CH1 on the DAQ Signal Accessory.

Front Panel

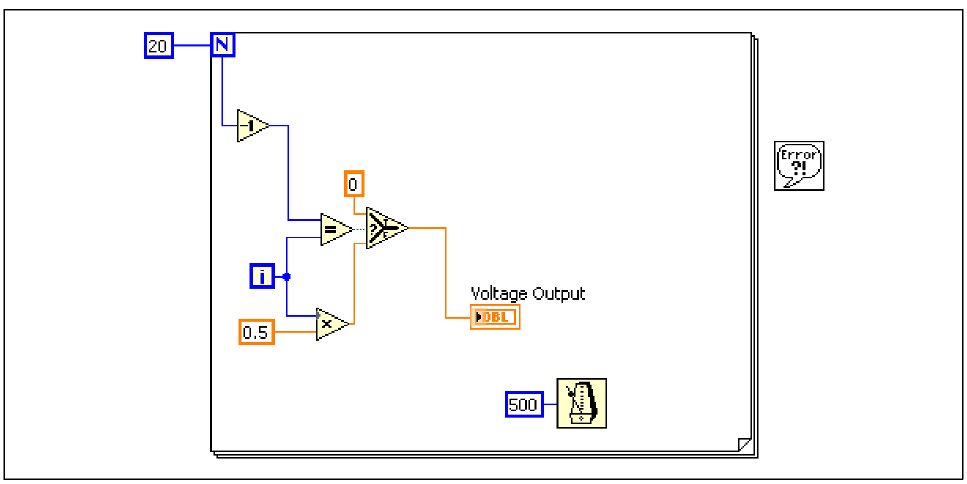
2. Open the Voltage Output VI located in the `C:\Exercises\LV Basics I` directory. The following front panel is already built.

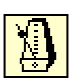


Voltage Output displays the current voltage output.

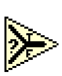
Block Diagram

3. Display and examine the block diagram.



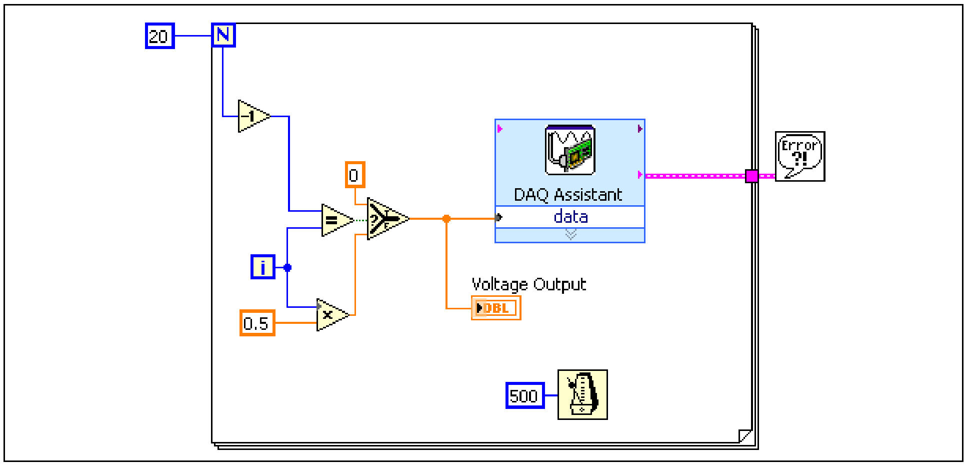


The Wait Until Next ms Multiple function located on the **Functions» All Functions»Time & Dialog** palette causes the For Loop to execute every 500 ms.



The Select VI located on the **Functions»Arithmetic & Comparison» Express Comparison** palette checks if the loop is in its last iteration. If the loop is in its last iteration, then the DAQ device outputs 0 volts. This is a good technique to reset the output voltage to a known level. It is always a good idea to reset the output voltage to something that won't damage a device that is connected to the DAQ device.

4. Modify the block diagram as shown in the following figure.



Place the DAQ Assistant Express VI, located on the **Functions»Input** palette, in the For Loop. Complete the following steps to configure this Express VI to generate an analog output voltage.

a. Select **Analog Output»Voltage** for the measurement to make.

b. Select **Dev1»ao0** for the physical channel and click the **Finish** button.

c. In the **Analog Output Voltage Task Configuration** dialog box that appears, configure the **Task Timing** to **Generate 1 Sample**. Change the output range minimum to `0` and maximum to `10`.

d. Click the **OK** button to close the **Analog Output Voltage Task Configuration** dialog box. This saves the settings specified for the task in the DAQ Assistant Express VI.

5. Save the VI.

6. Close the block diagram but leave the front panel open.

Front Panel

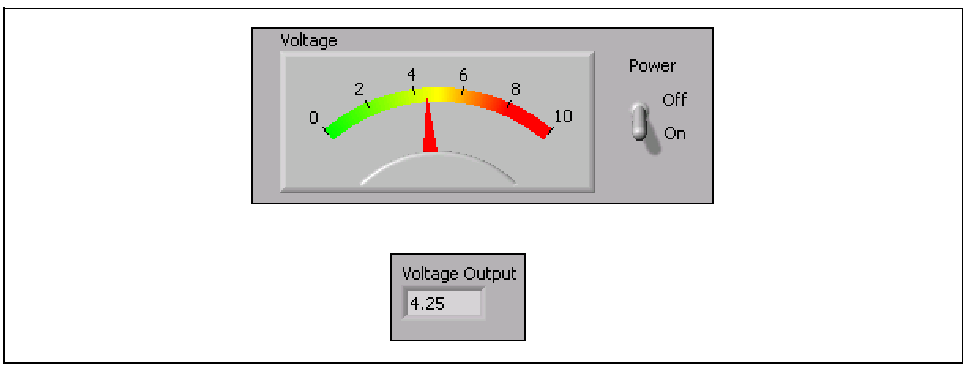
7. Open the Voltmeter VI that you completed in Exercise 7-2.

8. Configure the meter scale minimum to `0.0` and maximum to `10.0`.

Block Diagram

9. Display the block diagram for the Voltmeter VI and double-click the DAQ Assistant Express VI to open the **Analog Input Voltage Task Configuration** dialog box.
10. Right-click **Voltage** in the **Channel List** section and select **Change Physical Channel**. Select **ai1** for the channel because you wired the DAQ signal accessory to output a voltage on Analog Out CH0 and acquire the voltage from Analog In CH1.
11. Select **No Scale** from the **Custom Scaling** pull-down menu.
12. Change the voltage range to `0` to `10`.
13. Click the **OK** button to close the dialog box.
14. Display the front panel and run the Voltmeter VI.
15. To acquire and display the voltage output, run the Voltage Output VI.

    The Voltage Output VI outputs the voltage in 0.5 V increments from 0 to 9.5 V. When the For Loop executes its last iteration, the VI outputs 0 V to reset the analog output channel.



16. Close both VIs.

End of Exercise 7-5

Exercise 7-6: Simple Event Counting

Objective: To create a simple event counting VI.

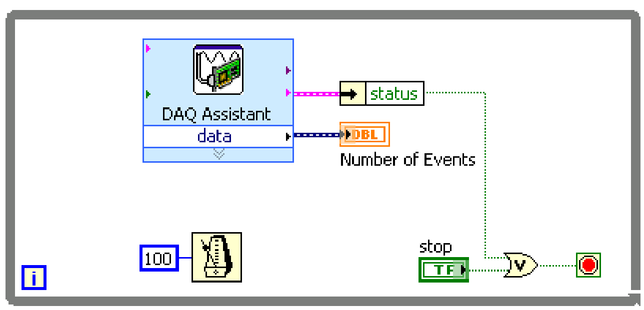
Complete the following steps to build a VI that counts pulses from the quadrature encoder on the DAQ Signal Accessory.

Front Panel

1. Open a blank VI and build the following front panel.

Block Diagram

2. Build the following block diagram.



Place the DAQ Assistant Express VI, located on the **Functions»Input** palette, in a While Loop. Complete the following steps to configure the counter to perform event counting.

a. Select **Counter Input»Edge Count** for the measurement to make.

b. Select **Dev1»ctr0** for the physical channel.

c. In the **Counter Input Edge Count Task Configuration** dialog box that appears, leave the settings as they are. The default settings define the source of the counter as being Programmable Function Input (PFI) 8, which is the default source for counter 0. The DAQ Signal Accessory connects counter 0 source input to PFI 8.

d. Click the **OK** button to close the **Counter Input Edge Count Task Configuration** dialog box. This saves all the settings specified for the task in the DAQ Assistant Express VI.

3. Save the VI as `Simple Event Counting.vi` in the `C:\Exercises\LV Basics I` directory.

4. On the DAQ Signal Accessory, wire the A output of the quadrature encoder to the SOURCE input of counter 0.
5. Run the VI. Rotate the quadrature encoder knob on the DAQ Signal Accessory. Notice that the Number of Events indicator increments as you rotate the knob. The quadrature encoder knob produces pulses as you rotate the knob. The counter counts these pulses.
6. Stop the VI.
7. Double-click the DAQ Assistant Express VI, and change the **Count Direction** pull-down menu to **Externally Controlled**. Click the **OK** button to close the configuration dialog box.

   The DAQ Signal Accessory internally connects phase B of the quadrature encoder to the Up/Down line for counter 0. This can be used to determine the direction the knob has turned.
8. Run the VI. Rotate the quadrature encoder knob on the DAQ Signal Accessory.

   Notice that the Number of Events indicator decrements when you rotate the knob clockwise, and increments when you rotate the knob counterclockwise.
9. Save and close the VI.

End of Exercise 7-6

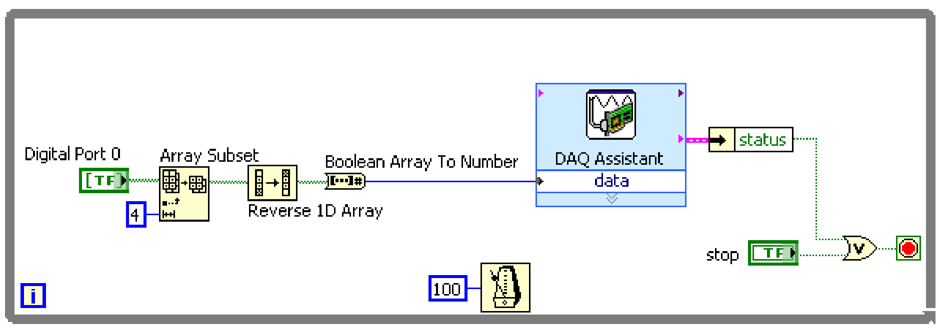
Exercise 7-7: Digital Example

Objective: To control the digital I/O lines on the DAQ device.

Complete the following steps to complete a VI that turns on the LEDs of Port 0 on the DAQ Signal Accessory based on the digital value set on the front panel. Each LED is wired to a digital line on the DAQ device. The lines are numbered 0, 1, 2, and 3, starting with the LED on the right.

Note The LEDs use negative logic. That is, writing a 1 to the LED digital line turns off the LED. Writing a 0 to the LED digital line turns on the LED.

1. Open the Digital Example VI, located in the `C:\Exercises\LV Basics I` directory, and modify the block diagram as shown in the following figure.



Place the DAQ Assistant Express VI, located on the **Functions»Input** palette, in the While Loop. Complete the following steps to configure the counter to perform event counting.

a. Select **Digital I/O»Port Output** for the measurement to make.

b. Select **Dev1»port0** for the physical channel and click the **Finish** button.

c. In the **Digital Output Port Task Configuration** dialog box that appears, select **Invert All Lines In Port** because the LEDs use negative logic.

d. Click the **OK** button to close the configuration dialog box. All of the settings specified for the task are saved internally in the DAQ Assistant VI.

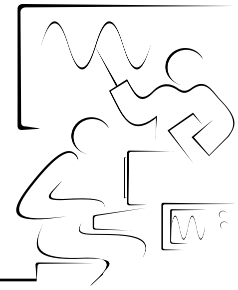
The Boolean buttons on the front panel are stored in an array to simplify the code. The Array Subset function extracts only the first four elements in the array. The output of the array subset needs to be reversed since element 0 of the array is the most significant bit. The array is then converted to a number with the Boolean Array to Number function, which passes its output to the DAQ Assistant Express VI to write that value to the port.

2. Save the VI.
3. Display the front panel and run the VI. Turn the Boolean LEDs on and off and observe the changes on the DAQ Signal Accessory.
4. Stop and close the VI.

End of Exercise 7-7

Lesson 8

Instrument Control

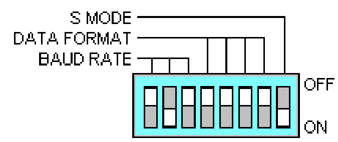


Exercises **Page**

Exercise 8-1: Using Measurement & Automation Explorer

Objective: To use MAX to examine the GPIB interface settings, detect instruments, and communicate with an instrument.

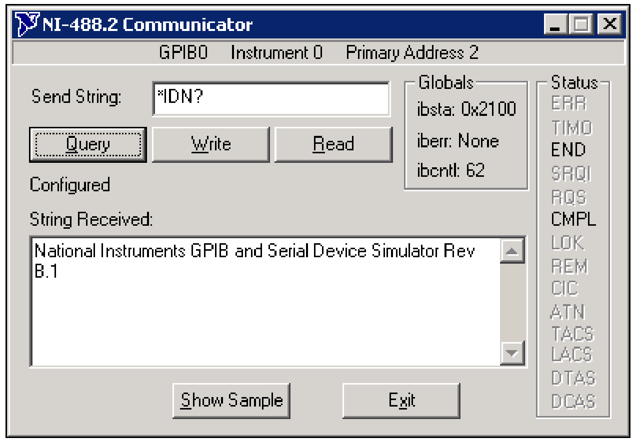
1. Power off the NI Instrument Simulator and configure it to communicate through GPIB by setting the following left bank of switches on the side of the box.



2. Power on the NI Instrument Simulator and verify that both the Power and Ready LEDs are lit.
3. Launch MAX by either double-clicking the icon on the desktop or by selecting **Tools»Measurement & Automation Explorer** in LabVIEW.
4. Expand the **Devices and Interfaces** section to display the installed interfaces. If a GPIB interface is listed, the NI-488.2 software is correctly loaded on the computer.
5. Select the GPIB interface and click the **Properties** button on the toolbar to display the **Properties** dialog box.
6. Examine but do not change the settings for the GPIB interface, and click the **OK** button.
7. Make sure the GPIB interface is still selected in the **Devices and Interfaces** section and click the **Scan for Instruments** button on the toolbar.
8. Expand the GPIB board section. One instrument named `Instrument0` appears.
9. Click **Instrument0** to display information about it in the right pane of MAX.

   The NI Instrument Simulator has a GPIB primary address (PAD) of 2.
10. Click the **Communicate with Instrument** button on the toolbar. An interactive window appears. You can use it to query, write to, and read from that instrument.

11. Type `*IDN?` in **Send String** and click the **Query** button. The instrument returns its make and model number in **String Received**. You can use this window to debug instrument problems or to verify that specific commands work as described in the instrument documentation.



12. Type `MEAS:DC?` in **Send String** and click the **Query** button. The NI Instrument Simulator returns a simulated voltage measurement.
13. Click the **Query** button again to return a different value.
14. Click the **Exit** button.
15. Set a VISA alias for the NI Instrument Simulator so you can use the alias instead of having to remember the primary address.
    a. While **Instrument0** is selected in MAX, click the **VISA Properties** button to display the **Properties** dialog box.
    b. Type `devsim` in the **VISA Alias** field and click the **OK** button. You will use this alias throughout this lesson.
16. Select **File»Exit** to exit MAX.

End of Exercise 8-1

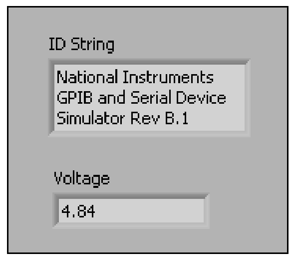
Exercise 8-2: Using the I/O Assistant

Objective: To build a VI that uses the Instrument I/O Assistant to communicate with a GPIB interface.

Complete the following steps to build a VI that acquires data from the NI Instrument Simulator.

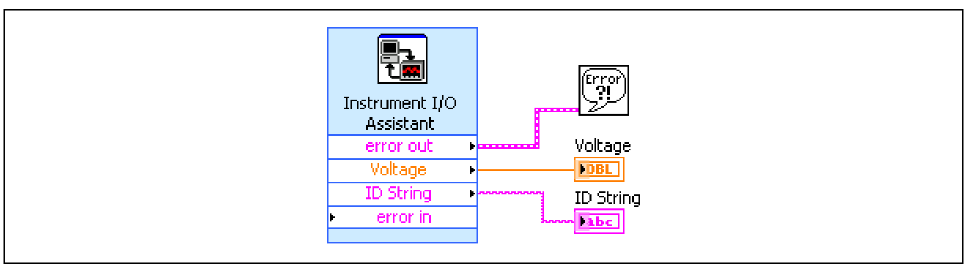
Front Panel

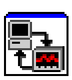
1. Open a blank VI.
2. The following front panel will result from building the block diagram.



Block Diagram

3. Display and build the following block diagram.





a. Place the Instrument I/O Assistant Express VI, located on the **Functions»Input** palette, on the block diagram. Complete the following steps to configure the Express VI in the **Instrument I/O Assistant** dialog box.

(1) Select **devsim** from the **Select an instrument** pull-down menu and select **VISA Code Generation** from the **Code generation type** pull-down menu.

(2) Click the **Add Step** button. Click **Query and Parse** to write and read from the Instrument Simulator.

(3) Type `*IDN?` as the command, select `\n` as the **Termination character**, and click the **Run this step** button. If no error warning appears in the lower half of the dialog box, this step has successfully completed.

(4) To parse the data received, click the **Auto parse** button.

Notice that `Token` now appears in the **Outputs** pane on the left side of the dialog box. This value represents the string returned from the identification query. Rename `Token` by typing `ID String` in the **Token name** text box.

(5) Click the **Add Step** button. Click **Query and Parse**. Type `MEAS:DC?` as the command and click the **Run this step** button.

(6) To parse the data received, click the **Auto parse** button. The data returned is a random numeric value. Rename `Token` by typing `Voltage` in the **Token name** text box.

(7) Click the **OK** button to exit the I/O Assistant and return to the block diagram.

b. Right-click the **ID String** output and select **Create»Indicator** from the shortcut menu.

c. Right-click the **Voltage** output and select **Create»Indicator** from the shortcut menu.

d. Wire the **Error Out** output to the Simple Error Handler VI.

4. Display the front panel and run the VI. Resize the string indicator if necessary.
5. Save the VI as `Read Instrument Data.vi` in the `C:\Exercises\LV Basics I` directory.
6. Right-click the I/O Assistant and select **Show Front Panel**. Click the **Convert** button when asked if you want to convert to a subVI.
7. View the code generated by the I/O Assistant. Where is the command `*IDN?` written to the Instrument Simulator? Where is the voltage being read?
8. Select **File»Exit** to exit the subVI. Do not save changes.

End of Exercise 8-2

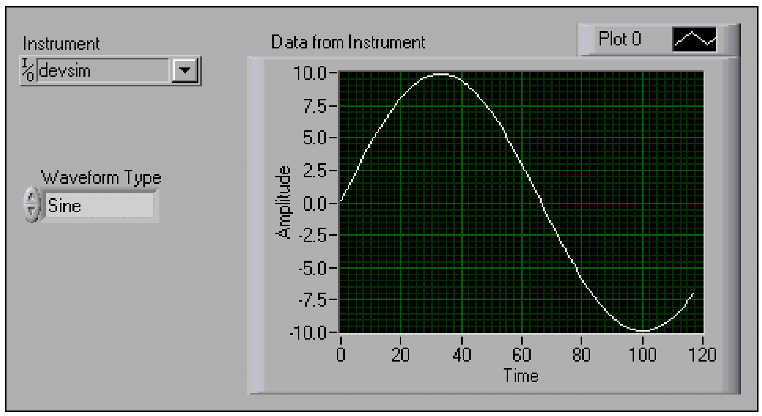
Exercise 8-3: Programming with VISA

Objective: To build a VI that uses the VISA functions to communicate with a GPIB interface.

Complete the following steps to build a VI that acquires a waveform from the NI Instrument Simulator.

Front Panel

1. Open a new VI and build the following front panel.



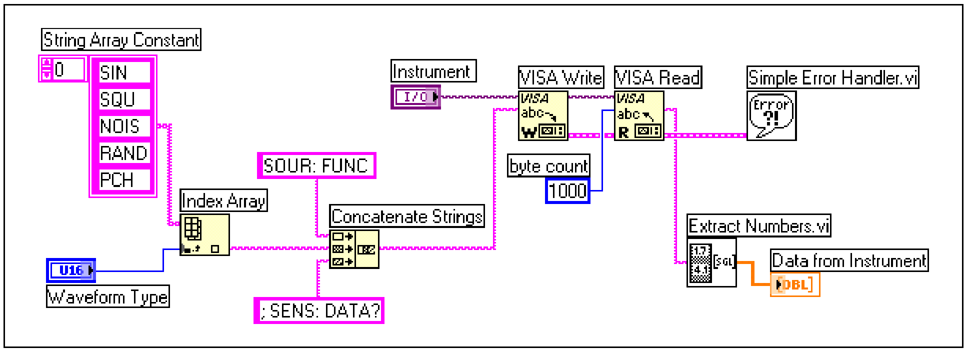
   a. Use the Labeling tool to enter the following values for the **Waveform Type** text ring located on the **Controls»All Controls» Ring & Enum** palette:

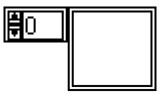
```
0 = Sine
1 = Square
2 = Noisy Sine
3 = Random
4 = Chirp
```

   b. Type the first entry into the text ring. Right-click the ring control, select **Add Item After** from the shortcut menu, and type the second entry into the ring control.

Block Diagram

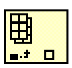
2. Build the following block diagram.



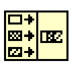


a. Place an array constant located on the **Functions»All Functions» Array** palette. This constant builds the command string for the NI Instrument Simulator. Each of the five waveform types is represented by an element in this array.

b. Place three string constants located on the **Functions»All Functions»String** palette. Place one in the array constant and resize the array constant to show five elements. Place the other two string constants as shown in the previous block diagram.



c. Place the Index Array function located on the **Functions»All Functions»Array** palette. This function extracts the string array element that matches the value of **Waveform Type**.



d. Place the Concatenate Strings function located on the **Functions» All Functions»String** palette. This function combines the string fragments into the complete command string for the NI Instrument Simulator.

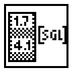
e. Place the VISA Write function located on the **Functions»All Functions»Instrument I/O»VISA** palette. This function writes the command string to the NI Instrument Simulator.

Note If you do not have a GPIB interface or an NI Instrument Simulator, place the (Demo) VISA Write VI located on the **Functions»All Functions»User Libraries» Basics I Course** palette to simulate writing a command to the instrument.

f. Place the VISA Read function located on the **Functions»All Functions»Instrument I/O»VISA** palette. This function reads the response from the NI Instrument Simulator.

Note If you do not have a GPIB interface or an NI Instrument Simulator, place the (Demo) VISA Read VI located on the **Functions»All Functions»User Libraries» Basics I Course** palette to simulate reading a string from the instrument.

g. Place the Simple Error Handler VI located on the **Functions»All Functions»Time & Dialog** palette. This VI displays a dialog box if an error occurs and displays the error information.



h. Place the Extract Numbers VI located on the **Functions»All Functions»User Libraries»Basics I Course** palette. This VI converts the comma delimited string returned from the NI Instrument Simulator into an array of numbers that the VI plots on the waveform graph.

3. Save this VI as `Read VISA Waveform.vi`.
4. Display the front panel, type either `devsim` or `GPIB::2::INSTR` in **Instrument**, and run the VI. The VI plots a waveform from the NI Instrument Simulator that matches the waveform type you selected.

Note If the VISA functions return an error, the command string might not be formatted correctly. Carefully check the spelling, punctuation, spaces, and capitalization. Sometimes an instrument locks up or gets into a confused state if the wrong command string is sent. Reinitialize the instrument by turning the Power switch off and on again.

5. Run the VI a few times requesting different waveforms each time to see how the waveform is read from the NI Instrument Simulator. It takes a second or so for the instrument to process the information and send the waveform to your computer.
6. Close the VI.

End of Exercise 8-3

Exercise 8-4: NI-DEVSIM Getting Started

Objective: To examine the installed LabVIEW instrument drivers and use the example VI from the NI DevSim instrument driver.

Note If you have a different instrument, install a free LabVIEW driver for it from the National Instruments Web site or the LabVIEW CD. In LabVIEW, select **Tools» Instrumentation»Instrument Driver Network** to visit the Instrument Driver Network at `ni.com`.

1. Open a new VI and display the block diagram.
2. Select the **Functions»Input»Instrument Drivers** palette and record which instrument drivers are installed:

   ____

   ____

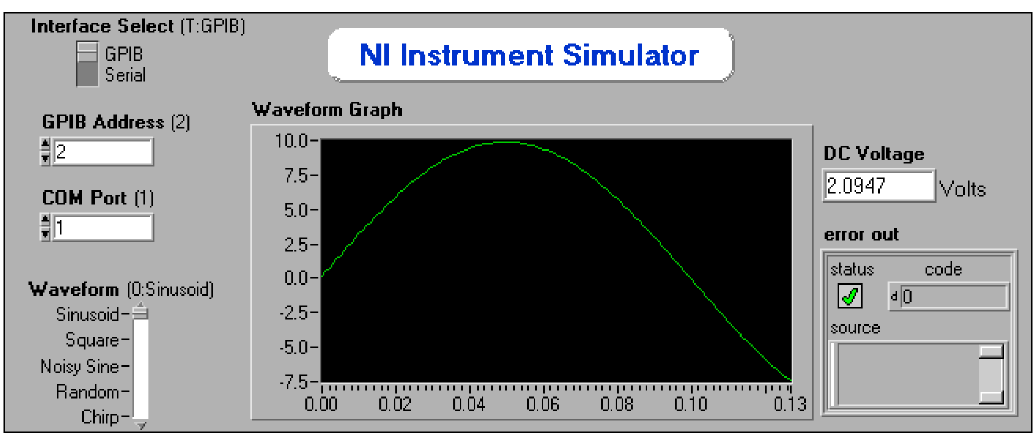
   ____

   ____

3. Select the **NI Device Simulator** subpalette and examine the categories of instrument driver VIs.
4. Select the **Application Examples** subpalette and place the NI DEVSIM Getting Started VI on the block diagram.

Front Panel

5. Double-click the NI DEVSIM Getting Started VI to display and examine the following front panel.

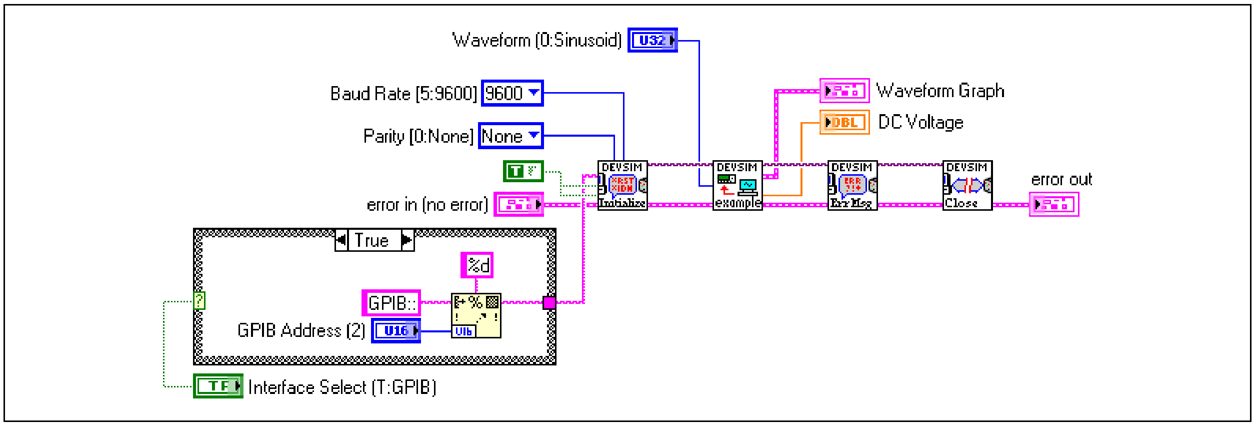


6. Run the VI. The simulator supplies a random DC voltage and generates the requested waveform on the graph. The simulator might take several seconds to acquire the waveform.

   You can simulate different waveforms by moving the **Waveform** slider and running the VI again.

Block Diagram

7. Display and examine the following block diagram.



The Initialize VI initializes the device, and the Application Example VI sends commands to configure and request information from the instrument. The Close VI ends the communication. All VIs using instrument drivers implement this structure of initialization, communication, and shutdown.

8. Close the VI. Do not save changes.

End of Exercise 8-4

Exercise 8-5: Voltage Monitor VI

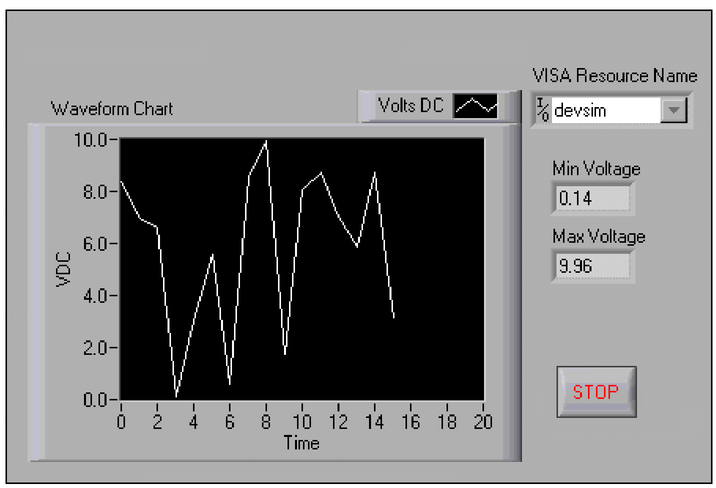
Objective: To explore a VI that uses the DevSim instrument driver VIs to acquire and plot voltages.

This exercise demonstrates a VI that uses the DEVSIM instrument driver VIs to acquire a DC voltage measurement from the NI Instrument Simulator once every second and plot it in a waveform chart. As each value is acquired, the VI compares it with the previous minimum and maximum values. The VI calculates and displays the minimum and maximum values continuously on the front panel.

You must have an NI Instrument Simulator to run this VI.

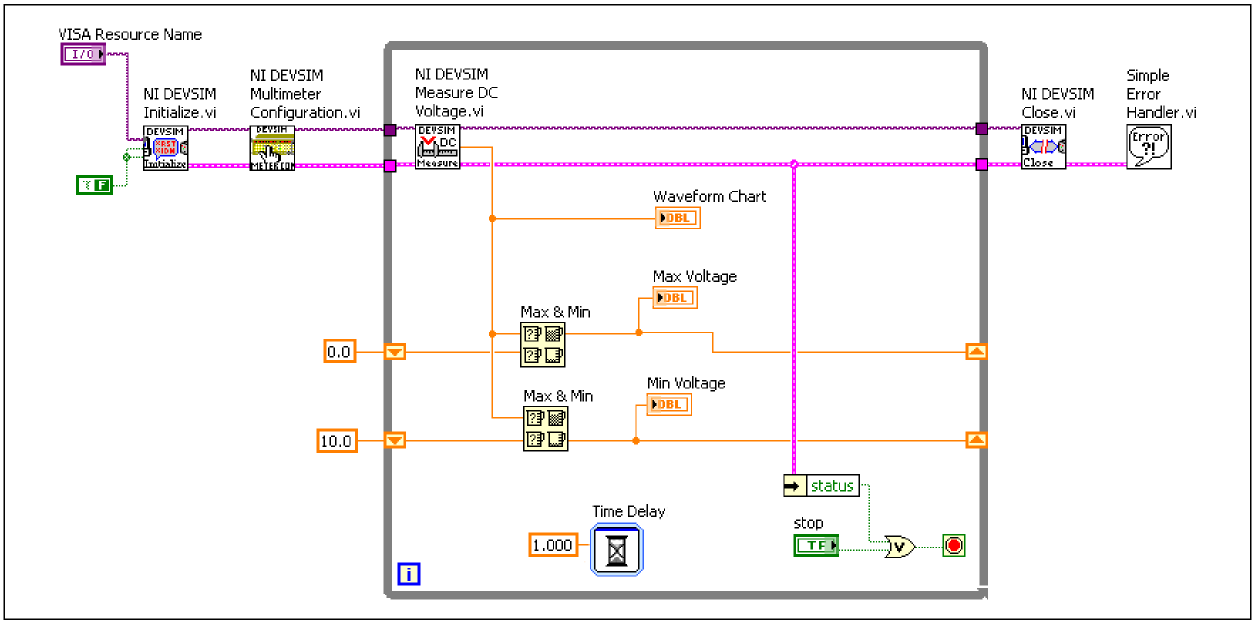
Front Panel

1. Open the Voltage Monitor VI located in the `C:/Exercises/LV Basics I` directory.
2. The front panel contains a VISA Resource Name control, a Waveform Chart, Max Voltage and Min Voltage numeric indicators, and a Stop button, as shown in the following figure.



Block Diagram

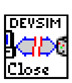
3. Display and examine the block diagram.



The NI DEVSIM Initialize VI opens communication between LabVIEW and the NI Instrument Simulator.

The NI DEVSIM Multimeter Configuration VI configures the range of voltage measurements that the NI Instrument Simulator generates. The default is `0.0` to `10.0` V DC.

The NI DEVSIM Measure DC Voltage VI returns a simulated voltage measurement from the NI Instrument Simulator. The voltage is output to a waveform chart and to the Max & Min functions. Shift registers on the While Loop store minimum and maximum voltage values. The Max & Min functions check the current voltage against the minimum and maximum values stored in the shift registers and output updated values to the Max Voltage and Min Voltage numeric indicators.



The NI DEVSIM Close VI ends communication between LabVIEW and the NI Instrument Simulator.

4. Make sure the NI Instrument Simulator is powered on.
5. Display the front panel and run the VI. The LEDs alternate between Listen and Talk as LabVIEW communicates with the GPIB instrument once a second to get a simulated voltage reading. This voltage displays on the chart, and the minimum and maximum values update accordingly.
6. Stop and close the VI.

End of Exercise 8-5

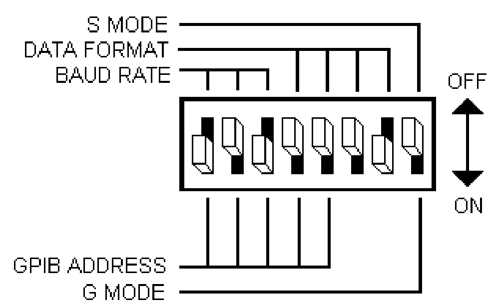
Exercise 8-6: Serial Read & Write

Objective: To build a VI that communicates with an RS-232 device.

Complete the following steps to use the Instrument I/O Assistant to build a VI that communicates with the NI Instrument Simulator.

NI Instrument Simulator

1. Power off the NI Instrument Simulator and configure it to communicate through the serial port by setting the following switches on the side of the box.



These switch settings configure the instrument as a serial device with the following settings:

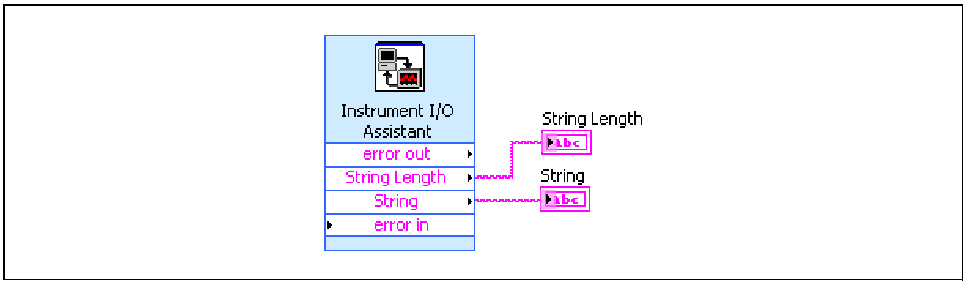
- Baud rate = 9,600
- Data bits = 8
- Parity = no parity
- Stop bits = 1
- Flow control parameters = hardware handshaking

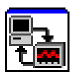
Handshaking is a means of data flow control. Software handshaking involves embedding control characters in transmitted data. For example, XON/XOFF flow control works by enclosing a transmitted message between the two control characters XON and XOFF. Hardware handshaking uses voltages on physical wires to control data flow. The RTS and CTS lines of the RS-232 device are frequently used for this purpose. Most lab equipment uses hardware handshaking.

2. Make sure the NI Instrument Simulator is connected to a serial port on the computer with a serial cable. Make a note of the port number.
3. Power on the NI Instrument Simulator. The Power, Ready, and Listen LEDs are lit to indicate that the device is in serial communication mode.

Block Diagram

4. Open a blank VI and build the following block diagram.





   a. Place the Instrument I/O Express VI, located on the **Functions» Input** palette, on the block diagram. Complete the following steps in the **Instrument I/O Assistant** dialog box that appears to configure the Express VI.

      (1) Choose **COM1** (or **COM2** depending on the connection port of the NI Instrument Simulator) from the **Select an instrument** pull-down menu.

      (2) Click the **Add Step** button and click **Write**. In the command field, type `*IDN?` and select `\n` as the **Termination character**.

      (3) Click the **Add Step** button and click **Read and Parse**.

      (4) Click the **Add Step** button and click **Read and Parse** again.

Note The Instrument Simulator returns the byte size of the response, the termination character, the response, then another termination character. Therefore, after `*IDN?` is sent to the instrument, the response must be read twice.

      (5) Click the **Run** button (not the **Run this step** button). The Run button runs the entire sequence.

      (6) Return to the first **Read and Parse** step.

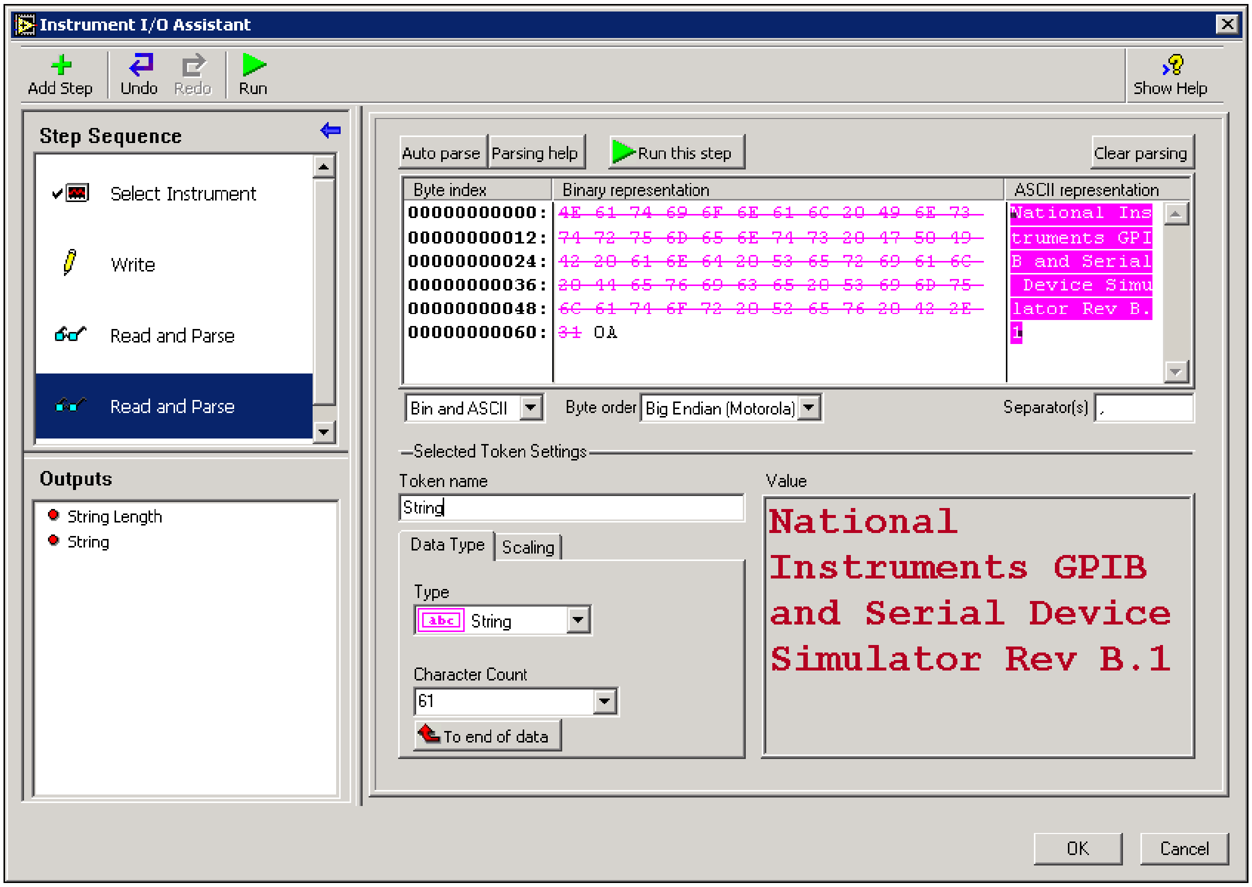
      (7) Click the **Auto parse** button. The value returned is the size in bytes of the query response.

      (8) Rename `Token` to `String Length` in the **Token name** text box.

      (9) Select the second **Read and Parse** step.

      (10) Click the **Auto parse** button. The value returned is the identification string of the NI Instrument Simulator.

(11) Rename `Token` to `String` in the **Token name** text box. The configuration window should be similar to the following figure.



(12) Select **OK** to return to the block diagram.

b. Right-click the **String** output and select **Create»Indicator** from the shortcut menu.

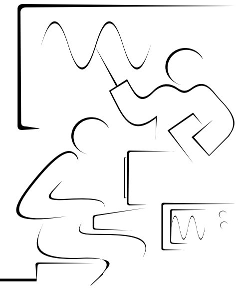
c. Right-click the **String Length** output and select **Create»Indicator** from the shortcut menu.

Tip Since LabVIEW is set to handle errors automatically, there is no need to connect a Simple Error Handler VI to **error out**.

5. Display the front panel and run the VI.
6. Save the VI as `Serial Communication.vi` in the `C:\Exercises\LV Basics I` directory.
7. Close the VI.

End of Exercise 8-6

Lesson 9



VI Customization

Exercises **Page**

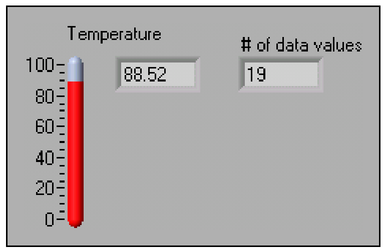
Exercise 9-1: Pop-Up Graph VI

Objective: To display a subVI front panel when a VI runs.

Complete the following steps to build a VI that acquires temperature once every 0.5 seconds for 10 seconds, displays a subVI front panel that shows the acquired data in a graph, and keeps the front panel open until you click a button.

Front Panel

1. Open a blank VI and build the following front panel.

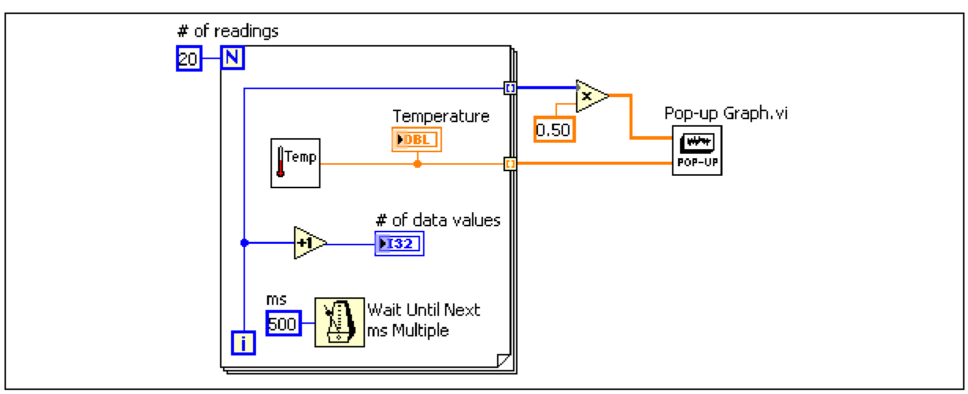


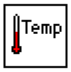
Use the following guidelines to assist you in building the front panel.

- The indicator to the right of the thermometer is a digital display belonging to the thermometer. Right-click the thermometer and select **Visible Items»Digital Display** from the shortcut menu to display the digital value.
- Change **# of data values** to signed 32-bit integer (I32) representation.

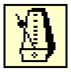
Block Diagram

2. Build the following block diagram.





a. Place the Thermometer VI, which you built in Exercise 2-2, on the block diagram. This VI acquires the current temperature value.



b. Place the Wait Until Next ms Multiple function, located on the **Functions»All Functions»Time & Dialog** palette, on the block diagram. Right-click the input, select **Create»Constant**, and type `500` in the constant to cause the For Loop to execute every 500 ms.

c. Place the Multiply function, located on the **Functions»Arithmetic & Comparison»Express Numeric** palette, on the block diagram. This function multiplies each element of the output array by `0.50` to scale the *x* values to represent the time interval at which the VI takes the measurements.

d. Place the Pop-up Graph VI, located in the `C:\Exercises\LV Basics I` directory, on the block diagram. This VI plots the temperature data on an XY graph.

e. Complete the block diagram as shown in the previous figure.

3. Save the VI as `Use Pop-up Graph.vi` in the `C:\Exercises\LV Basics I` directory.

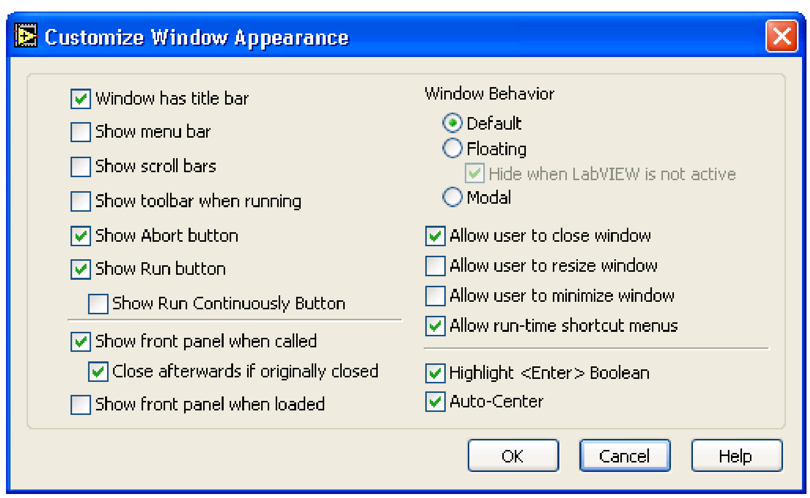
4. Configure the subVI to display its front panel when called.

   a. Double-click the Pop-up Graph subVI to open its front panel.

   b. Select **File»VI Properties**.

   c. Select **Window Appearance** from the **Category** pull-down menu.

   d. Click the **Customize** button. Configure the window appearance as shown in the following dialog box.



   e. Click the **OK** button twice and save and close the subVI. If the front panel is not closed, it will not close after the subVI runs.

5. Run the Use Pop-up Graph VI. After the VI acquires 10 seconds of temperature data, the front panel of the Pop-up Graph VI displays and plots the temperature data. Click the **DONE** button to return to the calling VI.
6. Change the window appearance settings for the Pop-up Graph subVI to the **Dialog** window style.
7. Save and close the subVI.
8. Run the Use Pop-up Graph VI again. The Pop-up Graph subVI front panel window behaves as a dialog box. For example, the window stays on top of all other windows and uses the system colors.
9. Close all open VIs.

End of Exercise 9-1

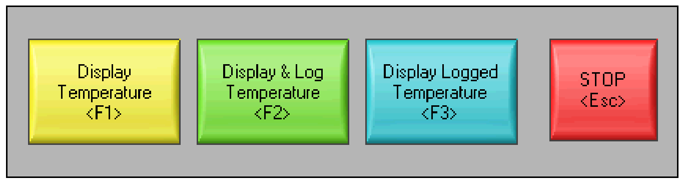
Exercise 9-2: Temperature System

Objective: To set keyboard shortcuts for front panel controls and display a subVI front panel when a VI runs.

Complete the following steps to build a temperature monitoring system you can use to view three different tests on request.

Front Panel

1. Open the Temperature System VI. The following front panel is already built.

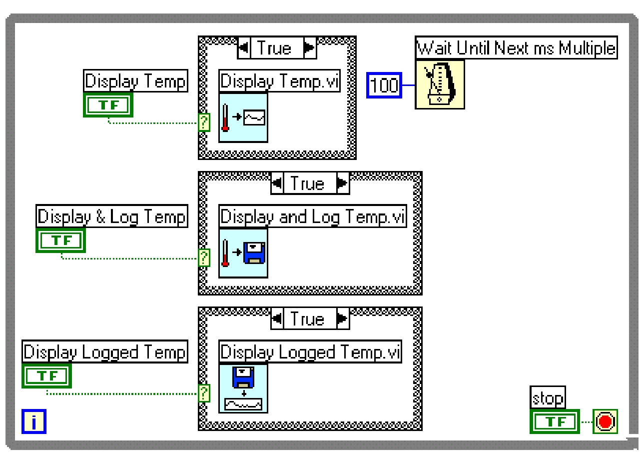


   The front panel contains four Boolean buttons. The mechanical action of the first three buttons is **Latch When Pressed**, and the **STOP** button is **Latch When Released**.

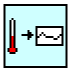
2. Right-click a control and select **Advanced»Key Navigation** from the shortcut menu to display the **Key Navigation** dialog box.
3. In the **Key Assignment** section, assign the shortcut key shown in the previous front panel.
4. Repeat steps 2 and 3 for each control.

Block Diagram

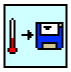
5. Build the following block diagram, leaving all the FALSE cases empty.



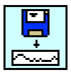
a. Place a While Loop and three Case structures located on the **Functions»Execution Control** palette.



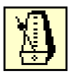
b. Place the Display Temp VI located on the **Functions»All Functions»User Libraries»Basics I Course** palette. This VI simulates a temperature measurement every 500 ms and plots it on a strip chart.



c. Place the Display and Log Temp VI located on the **Functions»All Functions»User Libraries»Basics I Course** palette. This VI simulates a temperature measurement every 500 ms, plots it on a strip chart, and logs it to a file.



d. Place the Display Logged Temp VI located on the **Functions»All Functions»User Libraries»Basics I Course** palette. This VI opens a file that you select, reads the logged data, and displays them on a graph.



e. Place the Wait Until Next ms Multiple function located on the **Functions»All Functions»Time & Dialog** palette. This function causes the While Loop to execute every 100 ms.

6. Double-click each subVI icon to open its front panel, examine the block diagram, and close the subVI.
7. Configure each subVI to display its front panel when called.
   a. Right-click the subVI and select **SubVI Node Setup** from the shortcut menu.
   b. Place a checkmark in the **Show Front Panel when called** and **Close afterwards if originally closed** checkboxes.
   c. Repeat steps a and b for the remaining two subVIs.
8. Save the VI.
9. Display the front panel and run the VI.
10. Click each button and press the corresponding keyboard shortcuts.

Note The three subVIs return to the Temperature System VI front panel when you press the <Enter> key. Try pressing the <Enter> key to do so.

11. Stop the VI.
12. Configure the Temperature System VI to run automatically when you open the VI.
    a. Select **File»VI Properties**.
    b. Select **Execution** from the **Category** pull-down menu.
    c. Place a checkmark in the **Run When Opened** checkbox.

13. Configure the VI so the menu bar and toolbar are not visible while the VI runs.
    a. Select **Window Appearance** from the **Category** pull-down menu.
    b. Click the **Customize** button.
    c. Remove the checkmarks from the **Show Menu Bar** and **Show Toolbar When Running** checkboxes.
    d. Click the **OK** button twice.
14. Save and close all VIs.
15. Open the Temperature System VI again. The VI runs automatically when you open it.
16. Click the buttons on the front panel or use the keyboard shortcuts.
17. Stop and close all VIs.

End of Exercise 9-2

Exercise 9-3: Edit Me VI

Objective: To edit a VI with properties that make it difficult to edit.

Complete the following steps to modify a VI that is configured to run when opened and quit LabVIEW after it runs.

Front Panel

1. Close any open VIs and open the Edit Me VI, located in the `C:\Exercises\LV Basics I` directory. The following front panel is already built.

   The VI is already running when it opens. While the VI runs, you cannot use the menu bar, toolbar, or keyboard shortcuts to edit or abort the VI.

2. Click the **Start** button. After 10 seconds, the VI stops running and quits LabVIEW.
3. Relaunch LabVIEW and open a blank VI.
4. If the VI you want to edit either does not have subVIs or you do not know what it contains, complete steps 5 through 13.

   However, if the VI you want to edit has subVIs, open one of the subVIs and modify the block diagram to break the subVI. For example, place an Add function on the block diagram and do not wire the inputs. Open the VI you want to edit. Because its subVI is nonexecutable, the VI that calls it is also nonexecutable. It opens in edit mode and the **Run** button appears broken. Make sure to fix the subVI after you edit the calling VI.

5. Display the block diagram of the new VI.
6. Place the Edit Me VI, which is already built, on the block diagram. The front panel for the Edit Me VI displays.

   Although you can display the block diagram of the Edit Me VI, you cannot edit it.

7. Select **Operate»Change to Edit Mode**. A dialog box informs you that the VI is locked.

8. Click the **Unlock** button. You now can edit the VI. You also can unlock a VI by selecting **File»VI Properties** and selecting **Security** from the **Category** pull-down menu.
9. Select and delete the Quit LabVIEW function from the block diagram.
10. Save and close the Edit Me VI. Close the new VI and do not save changes.
11. Open the Edit Me VI again.
12. After the VI runs, try to edit it.
13. Close the Edit Me VI.

End of Exercise 9-3

Notes