

Exercise 1 – Convert °C to °F

Build a VI



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

In wiring illustrations, the arrow at the end of this mouse icon shows where to click and the number on the arrow indicates how many times to click.

Front Panel

1. Select **File»New VI** to open a new front panel.



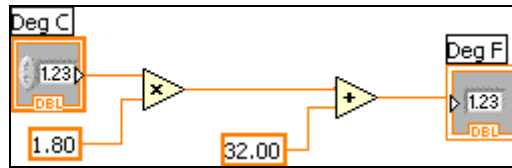
2. (Optional) Select **Window»Tile Left and Right** to display the front panel and block diagram side by side.
3. Create a numeric digital control. You will use this control to enter the value for degrees Centigrade.
 - a. Select the numeric control on the **Controls»Numeric Controls** palette. If the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

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

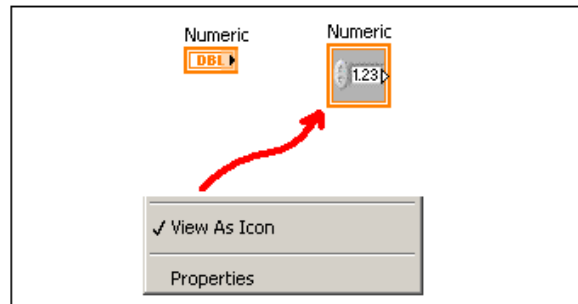
Note Control terminals have a thicker border than indicator terminals.

Block Diagram

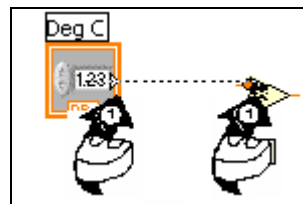
5. Display the block diagram by clicking, selecting **Window»Show Diagram**, or hitting **Ctrl-E**.



Note: Block Diagram terminals can be viewed as icons or as terminals. To change the way LabVIEW displays these objects right click on a terminal and select **View As Icon**.



6. Select the Multiply and Add functions on the **Functions» Programming» Numeric** palette and place them on the block diagram. If the **Functions** palette is not visible, right-click an open area on the block diagram to display it.
7. Select the numeric constant on the **Functions»Programming»Numeric** palette and place two of them on the block diagram. When you first place the numeric constant, it is highlighted so you can type a value.
8. Type 1.8 in one constant and 32.0 in the other.
If you moved the constants before you typed a value, simply double click inside the constant to allow editing.
9. Use the Wiring tool to wire the icons as shown in the previous block diagram.
 - To wire from one terminal to another, use the Wiring tool to click the first terminal, move the tool to the second terminal, and click the second terminal, as shown in the following illustration. You can start wiring at either terminal.



- You can bend a wire by clicking to tack the wire down and moving the cursor in a perpendicular direction. Press the spacebar to toggle the wire direction.
 - To identify terminals on the nodes, right-click the Multiply and Add functions and select **Visible Items»Terminals** from the shortcut menu to display the connector pane. Return to the icons after wiring by right-clicking the functions and selecting **Visible Items»Terminals** from the shortcut menu to remove the checkmark.
 - When you move the Wiring tool over a terminal, the terminal area blinks, indicating that clicking will connect the wire to that terminal and a tip strip appears, listing the name of the terminal.
 - To cancel a wire you started, press the <Esc> key, right-click, or click the source terminal.
10. Display the front panel.
 11. Save the VI because you will use this VI later in the course.
 - a. Select **File»Save**.
 - b. Navigate to your user drive.
- Note** Save all the VIs you edit in this course in your user drive.
- c. Type `Convert C to F.vi` in the dialog box.
 - d. Click the **Save** button.
12. Enter a number in the digital control and run the VI.
 - a. Use the Operating tool or the Labeling tool to double-click the digital control and type a new number.
 - b. Click the **Run** button to run the VI.
 - c. Try several different numbers and run the VI again.
 13. Select **File»Close** to close the Convert C to F VI.

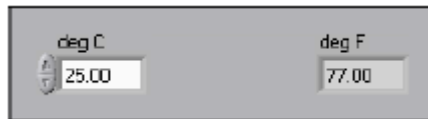
End of Exercise



Exercise 2a – Create a SubVI

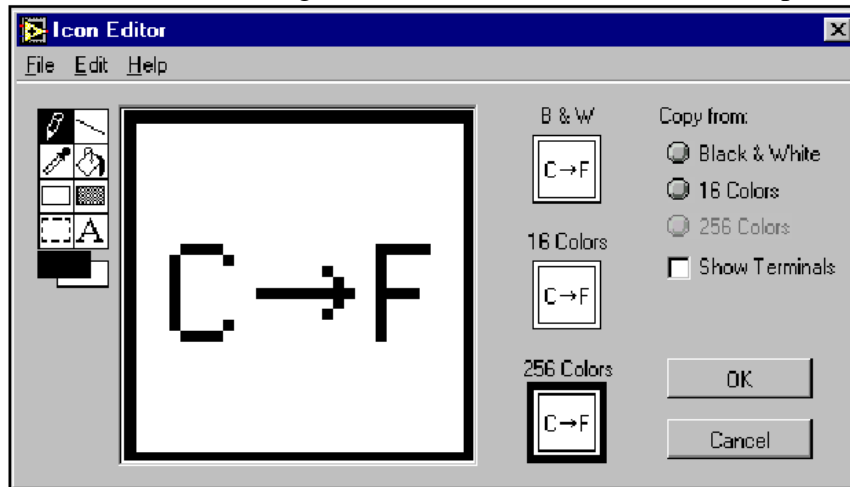
Front Panel

1. Select **File>Open** and navigate to your user drive to open the Convert C to F VI.
If you closed all open VIs, click the **Open...** button on the **LabVIEW** dialog box.

The following front panel appears.



2. 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.
3. Double-click the Select tool () on the left side of the **Icon Editor** dialog box to select the default icon.
4. Press the <Delete> key to remove the default icon.
5. Double-click the Rectangle tool to redraw the border.
6. Create the following icon. Note: It does not need to look perfect!



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

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

- e. Use the Select tool and the arrow keys to move the text and arrow you created.

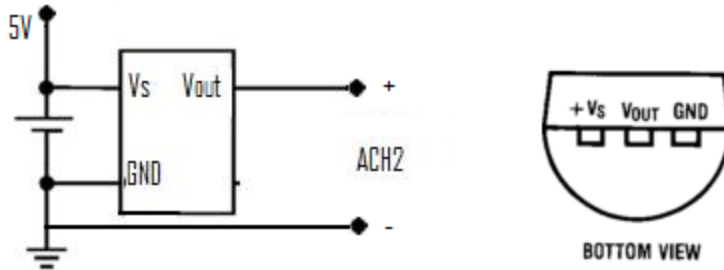
- f. When the icon is complete, click the **OK** button to close the **Icon Editor** dialog box. The icon appears in the upper right corner of the front panel and block diagram.
7. Right-click the icon on the front panel and select **Show Connector** from the shortcut menu to define the connector pane terminal pattern. Now, right click on the terminal icon, select patterns, and select an appropriate pattern. As this VI will have one input and one output, a good choice would be the fourth pattern on top.
8. Assign the terminals to the digital control and digital indicator.
 - a. Select **Help>Show Context Help** to display the **Context Help** window. View each connection in the **Context Help** window as you make it.
 - 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. The left terminal turns orange, and a marquee highlights the control.
 - d. Click an open area of the front panel. The marquee disappears, and the terminal changes to the data type color of the control to indicate that you connected the terminal.
 - e. Click the right terminal in the connector pane and click the **deg F** indicator. The right terminal turns orange.
 - f. Click an open area on the front panel. Both terminals are orange.
 - g. Move the cursor over the connector pane. The **Context Help** window shows that both terminals are connected to floating-point values.
9. Select **File>Save** to save the VI because you will use this VI later in the course.
10. Select **File>Close** to close the Convert C to F VI.

End of Exercise

Exercise 2b - Data Acquisition

To complete this exercise, you will be using the circuit sitting at your labstation.

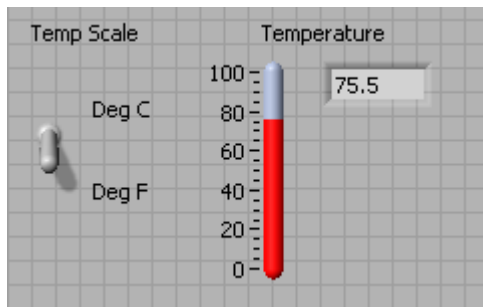
Wiring up the Temperature Sensor



1. Wire the LM19 temperature sensor to the Labstation input ACH2 as shown above using the Labstation's 5V power supply. **Note that the diagram shows a BOTTOM VIEW of the sensor. Wiring the sensor backwards will break the sensor!!!** Turn on the supply.
2. Do a Google search for "lm19", the temperature sensor you will be using. Find a PDF data sheet. What is the equation for calculating temperature from measured voltage?

Front Panel

3. Select **File»New** to open a new front panel.
4. Create the thermometer indicator, as shown on the following front panel.

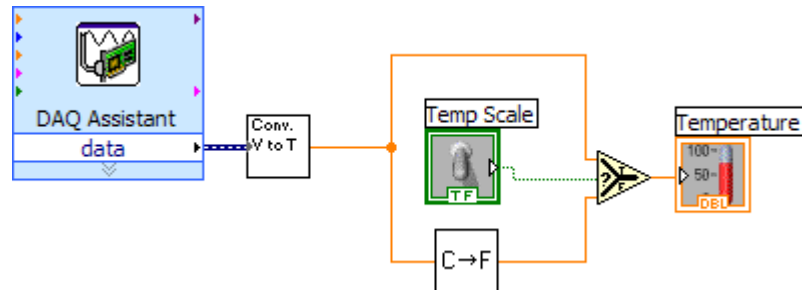


- a. Select the thermometer on the **Controls»Numeric Indicators** palette and place it on the front panel.
 - b. Type `Temperature` inside the label and click outside the label.
 - c. Right-click the thermometer and select **Visible Items»Digital Display** from the shortcut menu to display the digital display for the thermometer.
5. Create the vertical switch control.

- a. Select the vertical toggle switch on the **Controls»Buttons and Switches** palette.
- b. Type `Temp Scale` inside the label and click outside the label.
- c. Select **View»Tools palette** if the tools palette is not visible. Use the Labeling tool (**A**) to place a free label, `deg C`, next to the TRUE position of the switch, as shown in the previous front panel.
- d. Place a free label, `deg F`, next to the FALSE position of the switch.

Block Diagram

6. Select **Window»Show Diagram** to display the block diagram.
7. Build the following block diagram:




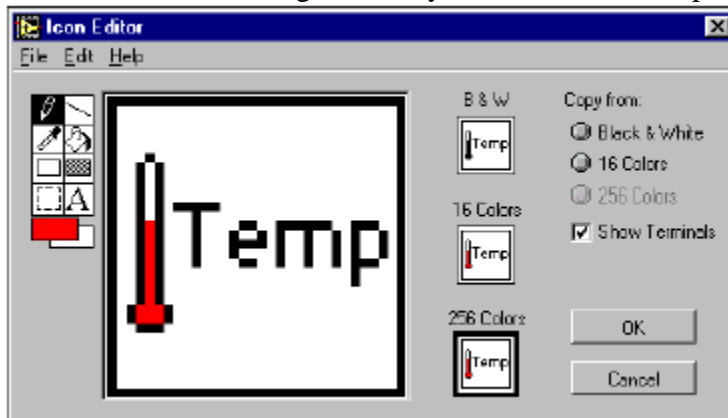
- a. Place the DAQ Assistant on your block diagram. It is located in **Functions»Measurement I/O»Ni-DAQmx»DAQ Assist.** A configuration wizard will automatically pop up. Select **Acquire Signals»»Analog Input»Voltage»Dev1 (PCI-MIO-16E-4)»ai2.** Click finish. In the Dialog box that opens, select **1 Sample (On Demand)** for Acquisition Mode, then hit **OK**. You have now configured a function which reads the voltage on ACH2 every time it is called. Note some of the other controls, such as range and terminal configuration, which in this case can be left in their default settings.
- b. Select **Functions»Select a VI.** Place the Convert V to T(Im19) VI after the AI Sample Channel subVI.
 - i. The VI is in **ITLL Documents\ITLL Modules\LabVIEW Introduction\VIs.** This function converts the voltage we read from our Im19 circuit into a real temperature.
- c. Take a look inside the Convert V to T VI. Is the conversion correct (Hint: Look at the data sheet)? If not, save your own version of the VI and fix the problem.
- d. Select **Functions»Select a VI**, navigate to the Convert C to F VI, which you built in Exercise 2a, and place the VI on the block diagram.
- e. Place the Select function located on the **Functions» Programming» Comparison** palette. This function returns either the Fahrenheit (FALSE) or Celsius (TRUE) temperature value, depending on the value of **Temp Scale**.

- f. 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 identify terminals on the nodes, right-click the icon and select

Visible Items»Terminal from the shortcut menu to display the connector pane.

8. Display the front panel.
9. Click the **Continuous Run** button () to run the VI continuously. Notice that there is significant noise in our signal. This is caused by the use of poorly shielded wiring. What can we do to reduce this noise?
10. Obtain a heat gun from the instructor and use it to heat up the sensor while the VI runs.
11. Click the **Continuous Run** button again to stop the VI.
12. Create the following icon, so you can use the Temperature VI as a subVI.



- 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 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 to redraw the border.
- e. Use the Pencil tool 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.

- a. Double-click the Text tool, shown at left, and change the font to **Small Fonts**.
 - b. When the icon is complete, click the **OK** button. The icon appears in the upper right corner of the front panel.
13. Select **File»Save** to save the VI. Choose a location on your user drive and save the VI as Thermometer.vi.
 14. Select **File»Close** to close the VI.

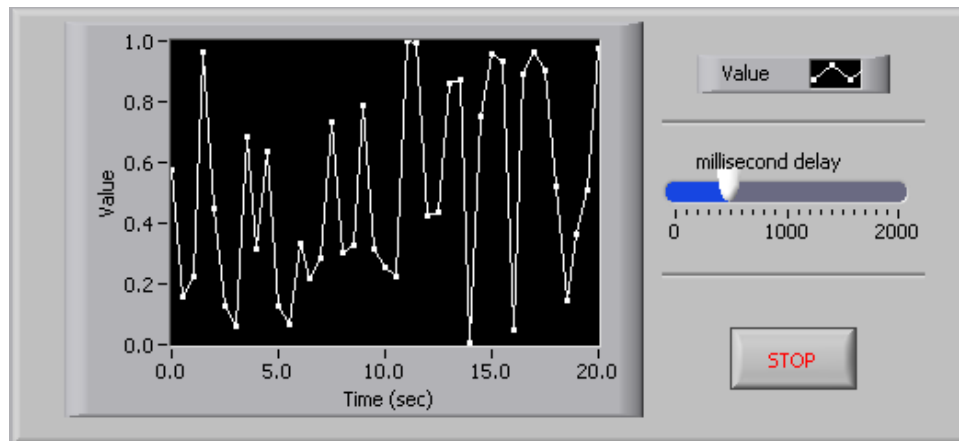
End of Exercise

Exercise 3 – Using Loops

Use a while loop and a waveform chart to build a VI that demonstrates software timing.

Front Panel

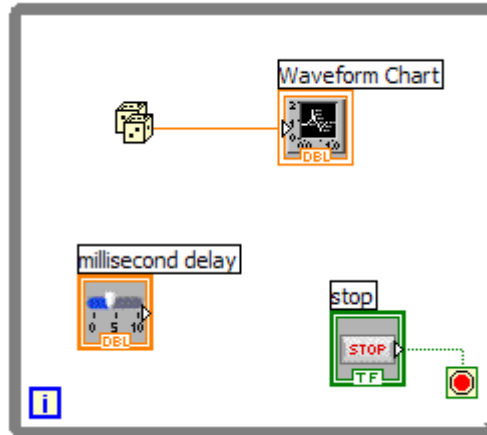
1. Open a new VI.
2. Build the following front panel.



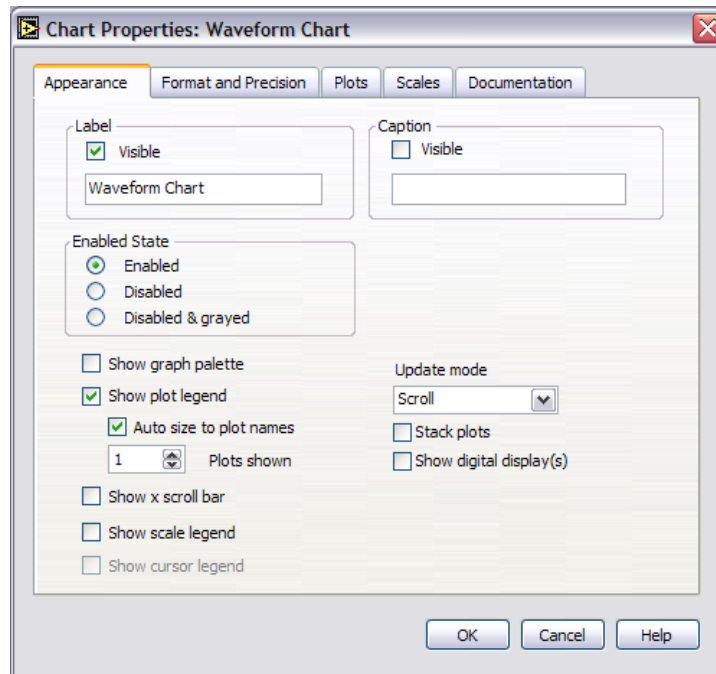
- a. Select the horizontal pointer slide on the **Controls»Numeric Controls** palette and place it on the front panel. You will use the slide to change the software timing.
- b. Type `millisecond delay` inside the label and click outside the label.
- c. Place a Stop Button from the **Controls»Buttons and Switches** palette.
- d. Select a waveform chart on the **Controls»Graph Indicators** palette and place it on the front panel. The waveform chart will display the data in real time.
- e. Type `Value History` inside the label and click outside the label.
- f. The waveform chart legend labels the plot `Plot 0`. Double click `Plot 0` in the chart legend. Type `Value`.
- g. Use the Labeling tool to double-click `10.0` in the y-axis, type `1`, and click outside the label or click the **Enter** button to rescale the chart.
- h. Change `-10.0` in the y-axis to `0`.
- i. Label the y-axis `Value` and the x-axis `Time(sec)` by double clicking on the labels.

Block Diagram

3. Display the block diagram.
4. Enclose the two terminals in a While Loop, as shown in the following block diagram.



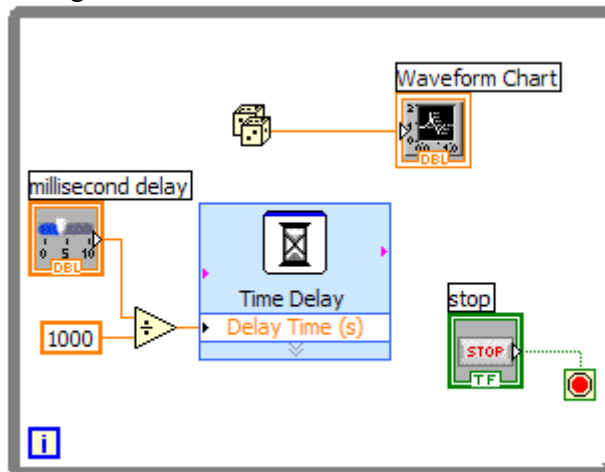
- a. Select the While Loop on the **Functions»Programming»Structures** palette.
 - b. Click and drag a selection rectangle around the two terminals.
 - c. Use the Positioning tool to resize the loop, if necessary.
5. Select the Random Number (0-1) on the **Functions»Programming»Numeric** palette. Alternatively you could use a VI that is gathering data from an external sensor, such as the temperature measuring VI we created earlier.
 6. Wire the block diagram objects as shown above.
 7. Save the VI as `Use a Loop.vi` because you will use this VI later in the course.
 8. Display the front panel by clicking it or by selecting **Window»Show Front Panel**.
 9. Run the VI. The section of the block diagram within the While Loop border executes until the specified condition (in this case the Stop button) is TRUE. For example, while the STOP button is not pressed, the VI returns a new number and displays it on the waveform chart. Note that the loop executes VERY fast. This is due to the fact that all of the code within the loop can execute very fast, and we have not yet put any delays inside the loop to control timing.
 10. Click the STOP button to stop the acquisition. The condition is FALSE, and the loop stops executing.
 11. Format and customize the X and Y scales of the waveform chart.
 - a. Right-click the chart and select **Properties** from the shortcut menu. The following dialog box appears.
 - b. Click the **Scale** tab and select different styles for the y-axis. You also can select different mapping modes, grid options, scaling factors, and formats and precisions. Notice that these will update interactively on the waveform chart
 - c. Select the options you desire and click the **OK** button.



12. Right-click the waveform chart and select **Data Operations»Clear Chart** to clear the display buffer and reset the waveform chart. If the VI is running, you can select **Clear Chart** from the shortcut menu.

Adding Timing

When this VI runs, the While Loop executes as quickly as possible. Complete the following steps to take data at certain intervals, such as once every half-second, as shown in the following block diagram.



- a. Place the Time Delay Express VI located on the **Functions»Programming»Timing** palette. The value you put in the dialog box doesn't matter, as we will be wiring a value that overwrites this value.
- b. Divide the millisecond delay by 1000 to get time in seconds. Connect the output of the divide function to the Delay Time (s) input of the

Time Delay Express VI. This will allow you to adjust the speed of the execution from the pointer slide on the front panel.

13. Save the VI.
14. Run the VI.
15. Try different values for the millisecond delay and run the VI again. Notice how this effects the speed of the number generation and display. Note that you can increase the maximum possible delay time by adjusting the range on the “Millisecond Delay” control. Do this by double clicking the maximum value on the “millisecond delay” control and putting in a different maximum value.
16. Close the VI.

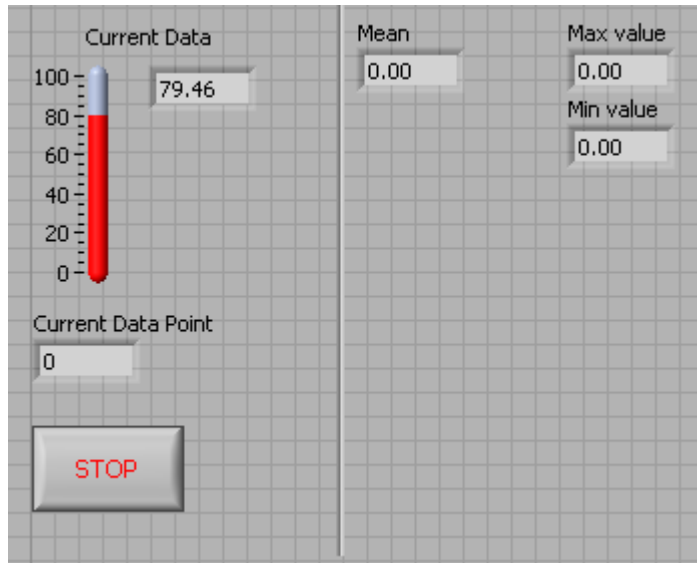
End of Exercise

Exercise 4 - Analyzing and Logging Data

Complete the following steps to build a VI that measures temperature every 0.25 s. During the acquisition, the VI displays the measurements in real time on a waveform chart. After the acquisition is complete, the VI plots the data on a graph and calculates the minimum, maximum, and average temperatures.

Front Panel

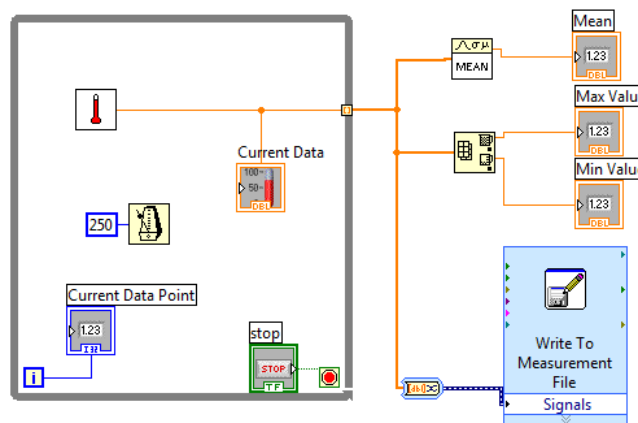
1. Open a new VI and build the following front panel using the following tips.



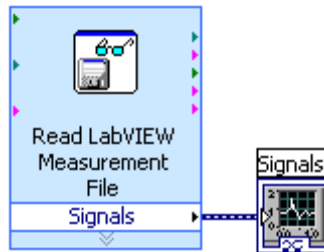
- Do not create the **Mean**, **Max**, and **Min** indicators yet. Create them on the Block Diagram by right clicking on the outputs of the appropriate functions and choosing Create Indicator. Then position them on the Front Panel.

Block Diagram

2. Build the following block diagram.



- a. Select **Functions»Select a VI...** and choose Thermometer.vi (from previous exercise). Note that your thermometer VI has no connectors on it. Place connectors on this SubVI in the same way we did for the “Convert C to F” VI.
 - b. Place the Wait Until Next ms Multiple function located on the **Functions»Programming »Timing** palette and create a constant of 250. Much like the Time Delay Express VI, this function causes the For Loop to execute every 0.25 s (250 ms).
 - c. Place the Array Max & Min function located on the **Functions»Programming»Array** palette. This function returns the maximum and minimum temperature.
 - d. Place the Mean VI located on the **Functions»Mathematics»Probability and Statistics** palette. This VI returns the average of the temperature measurements.
 - e. Right-click the output terminals of the Array Max & Min function and Mean VI and select **Create»Indicator** from the shortcut menu to create the **Max**, **Min**, and **Mean** indicators.
 - f. Place the Write To Measurement File Express VI located on the **Functions»Programming»File I/O** palette. LabVIEW will automatically insert the “To DDT” function into the wire you connect to the Signals input. The “To DDT” function converts the array from an array data type to a Signal data type, which is what the Measurement file function expects on its input.
 - g. Double click on the Write LabVIEW Measurements File VI. A dialog box will open. Select “Ask User to Choose File”.
 - h. Note that when you wire the output of Thermometer.vi to the While Loop output, a solid orange block will show. Right click on this and select enable indexing. This tells our program that we want to pass an array of EVERY temperature value out of our while loop.
 - i. Also note that the “Convert to Dynamic Data” function in front of the “Write to Measurement File” VI automatically appears when you wire the array to the VI input.
3. Save the VI as `Temperature Logger.vi`.
 4. Display the front panel and run the VI.
 5. After pressing STOP a dialog box will appear. Enter the name of the file to save the spreadsheet.
 6. Open the spreadsheet file to make sure the file was properly created by using Notepad, Excel, or by creating a VI to read the file as follows.
 - Create the following block diagram



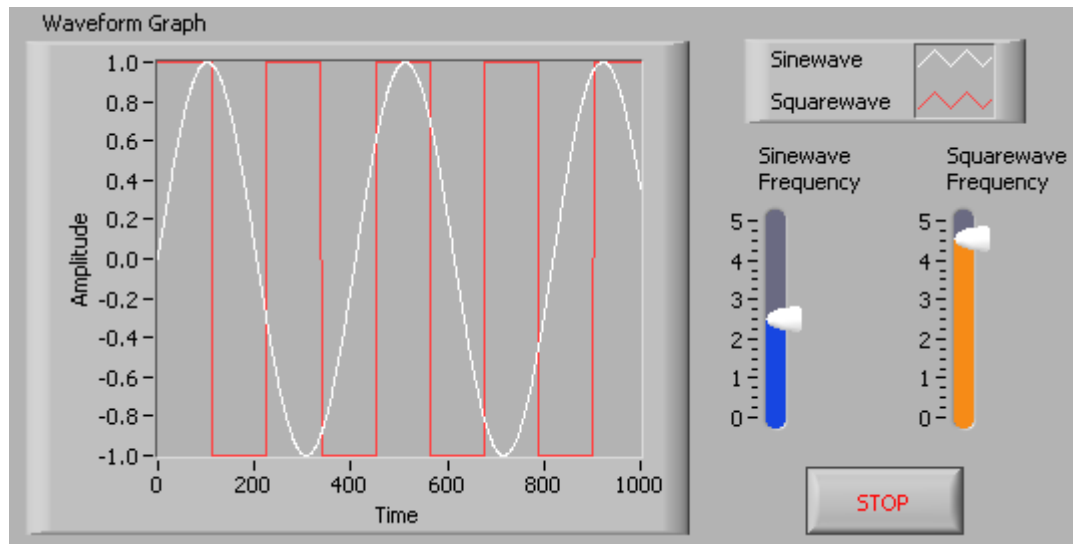
- Place the Read LabVIEW Measurement File Express VI located on the **Functions»Programming»File I/O** palette.
 - Configure the VI to ask the user to choose the file to read and change the delimiter to Tab
 - Right click on the Signals Output and choose create graph indicator
7. Run the VI. This imports the data back into LabVIEW.
 8. Save and close both of the VIs.

End of Exercise

Exercise 5 - Using Waveform Graphs

Front Panel

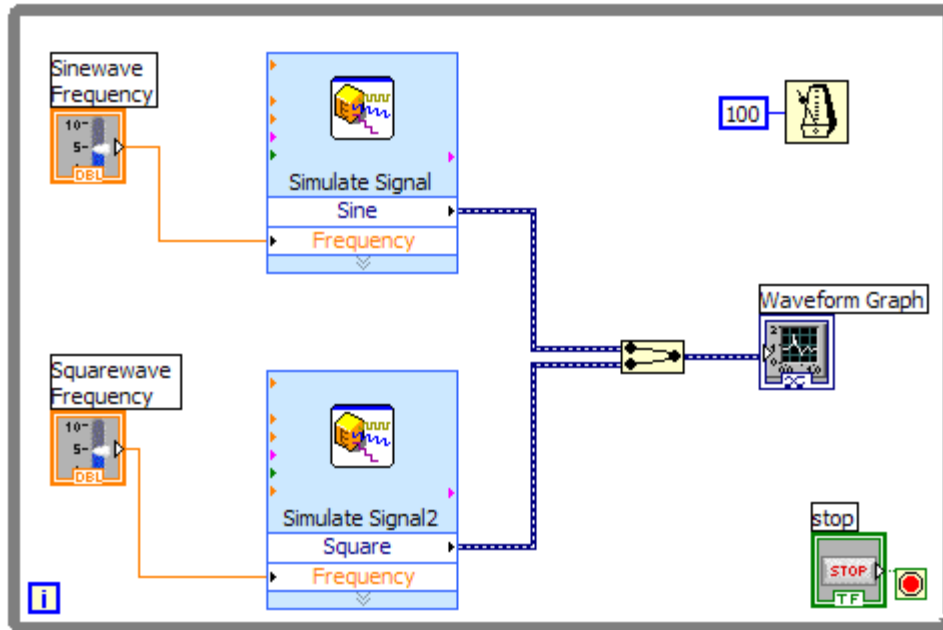
1. Open a new VI and build the following front panel using the following tips.



- a. Create a waveform graph indicator from the **Controls»Graph Indicators** palette. Use the position/size/select tool to move the plot legend to the side, and expand it to display two plots. Use the labeling tool to change the plot names and the properties page to choose different colors for your plots.
- b. Place a Stop button on the front panel.
- c. Place two vertical pointer slides from the **Controls»Numeric Controls** palette. Use the properties page again to change the slide fill color.

Block Diagram

2. Build the following block diagram.



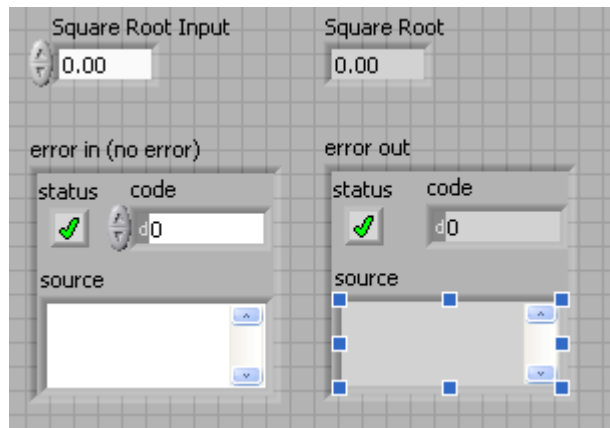
- a. Place a While Loop from **Functions»Programming»Structures** palette.
 - b. Place a **Wait Until Next ms Multiple** from **Functions»Programming»Timing** and create a constant with a value of 100.
 - c. Place two Simulate Signal Express VIs from the **Functions»Express»Input** and leave the Signal type as Sine for the first Simulate Signal VI and change the Signal Type to Square for the second VI. Wire both of the outputs into the waveform graph. A Merge Signals function will automatically be inserted.
 - d. Expand the Simulate Signal Express VIs to show another Input/Output. By default, error out should appear. Change this to Frequency by clicking on error out and choosing **Frequency**.
 - e. Note that we have created a program which runs 10 times a second and displays .1 seconds worth of data. If you wish to display more data, change each function to output more data using the “Number of Samples” control on the dialog box.
3. Save the VI as `Multiplot Graph.vi`.
 4. Display the front panel and run the VI.
 5. Save and close the VI.

End of Exercise

Exercise 6 - Error Clusters & Handling

Front Panel

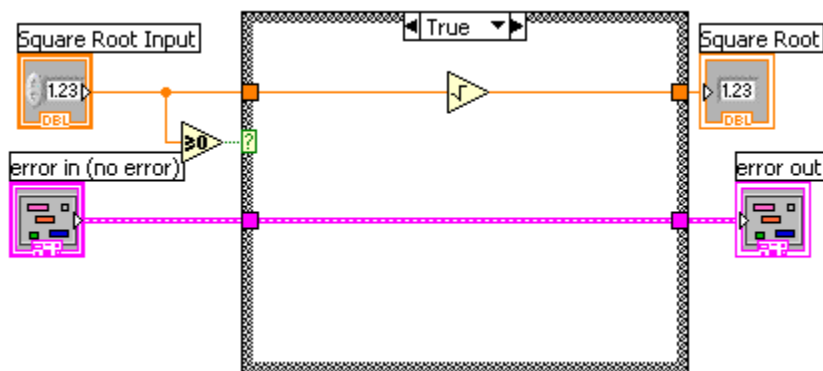
1. Open a new VI and build the following front panel using the following tips.



- a. Create a numeric control and change the Label to Square Root Input. Create a numeric indicator for Square Root.
- b. Place Error In 3D.ctl from **Controls»Modern»Array, Matrix & Cluster**.
- c. Place Error Out 3D.ctl from **Controls»Arrays & Clusters**.

Block Diagram

2. Build the following block diagram.

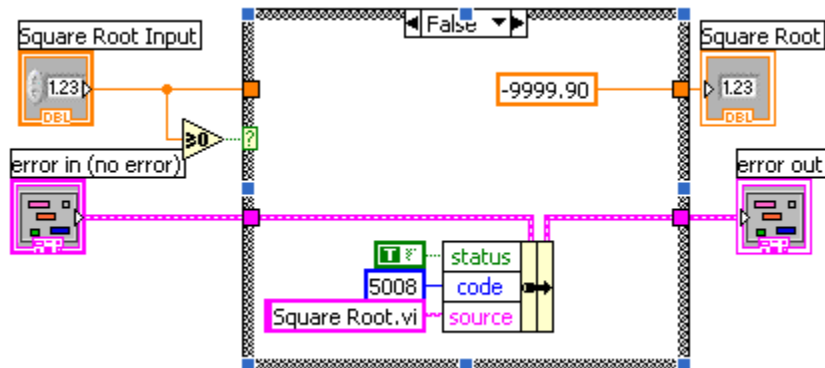


- a. Place a Case Structure from **Functions»Programming»Structures** palette.
- b. Place a **Greater or Equal to 0?** from the **Functions»Programming»Comparison** palette and wire it to the condition terminal of the case structure.

In the True Case:

- c. Place the Square Root function from **Functions»Programming»Numeric** palette.

In the False Case:



- d. Create a numeric constant from **Functions»Programming»Numeric** palette and type -9999.90. Create a TRUE constant from the **Functions»Boolean** palette. Finally, create an integer constant with the value 5008.
- e. Place the Bundle By Name from **Functions»Programming»Cluster and Variant** palette. Wire from “error in” to the center terminal of Bundle by Name to make status show up. Drag the bottom of the function down to make the “code” and “source” terminals visible. Wire from the Error Out indicator to the output of Bundle By name.

3. Save the VI as `Square Root.vi`.

4. Display the front panel and run the VI.

5. Save and close the VI.

6. Why does this function need an error case? Why to you suppose we return the value -9999.9 in the false case? What is the purpose of the Status, Code, and Source variables within the error cluster?

End of Exercise

Introduction to Labview Data Acquisition Using NI-DAQmx

Part 1 – What is Data Acquisition and NI-DAQmx?

Thus far in your Labview education you have learned the basics of programming, such as loops, arrays, clusters, etc. Now the question is, how do we acquire data from the outside world? As you can see on your labstation panel, we have many inputs and outputs available to us. We simply need to know how to write Labview code which reads from our inputs and writes to our outputs.

If you have completed our Labstation tutorial you should have a good idea of what our Labstation is capable of. One of the modules discussed in that document is the SCXI-1141, which is a software programmable low pass filter module. Today we will learn how to use Labview to program the filters in this module and retrieve signal data from our analog to digital converters.

The Labview functions we will use to complete these tasks are part of a data acquisition software suite called NI-DAQmx. The NI stands for National Instruments, the DAQ stands for Data Acquisition, and the MX just sounds really cool and modern! As the hardware and software are created by the same company, you will likely find the programming interface to be powerful and intuitive.

Part 2 – The SCXI Chassis:

You will notice that the labstation breakout panel has connections to four devices called SCXI modules. The SCXI stands for Signal Conditioning Extensions for Instrumentation. The modules are designed to condition low level signals, such as strain gauge and thermocouple signals. Every labstation has an 1121, 1122, and 1141 module. The fourth module varies depending on the labstation.

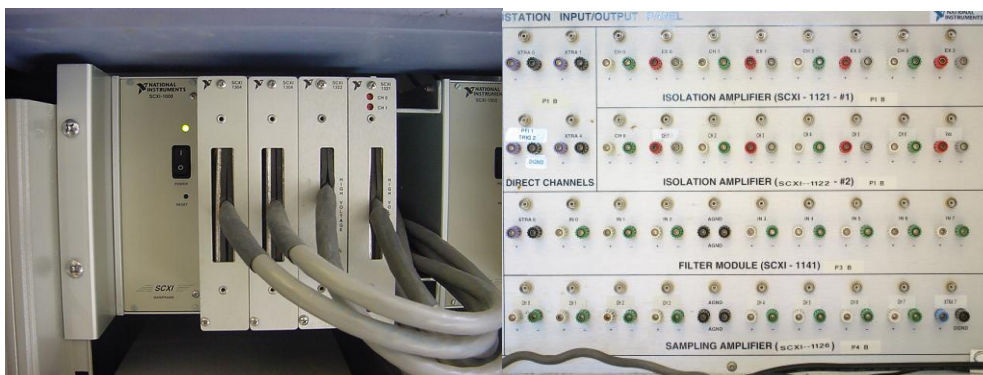


Figure 1: SCXI-1000 Chassis and Modules (Left), and Corresponding Breakout Panel

Part 3 – Tasks and the DAQ Assistant

3.1 Using the DAQ Assistant to Monitor Voltage

We will now learn how to capture a voltage vs. time signal in Labview:

1. Turn on the SCXI Chassis on the side of the lab station.
2. Connect the output of the function generator to the input of the SCXI-1141/Channel 0 input. Set the function generator to produce a 10 Vpp, 10 Hz sine wave.
3. Turn the output of the function generator on by hitting Channel>>Output>>On.
4. Open up Labview and create a new blank VI. Save it as “Capture Voltage.vi”.
5. Place the DAQ Assistant in the VI window by right clicking and selecting **Measurement I/O>>NI-DAQmx>>DAQ Assistant**.
6. Select **Acquire Signals>>Analog Input>>Voltage**. Notice that the DAQ Assistant can also be programmed to read other phenomena such as temperature, strain, etc.
7. Notice the list of devices we have which can measure voltage, including the DAQ Card and SCXI modules.

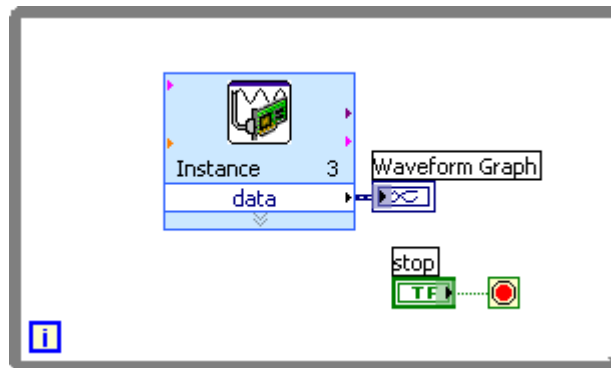


Figure 2: Our Simple Data Acquisition Program

8. Select **SCXI-1141 >> ai0**. Hit finish.
9. Within the DAQ Assistant window, set the input range max and min to 5 and -5. Set samples to read to 100 and rate to 100. For Acquisition Mode, make sure that N samples is selected. Click OK. *Always make sure to set your input max and min to be just above what you expect your signal range to be. Setting this too high will reduce the precision of your measurement. Setting it too low will “clip” your waveform.*

10. Place a while loop around the DAQ Assistant by right clicking and selecting **Structures >> While Loop** (see Figure 2). Right click on the Data output of the DAQ Assistant and select Create >> Graph Indicator. Right click on the Stop terminal and select “Create Control” to create a Stop button.
11. Run your program. Play around with the voltage and sample settings by stopping the VI and re-opening the DAQ Assistant. *What happens if our signal frequency is over half the sampling frequency? Use a sinusoidal signal to test this.*
12. Set the samples to read and rate of the DAQ Assistant back to 100 and 100.

3.2 Trigger Control

Notice that your waveform tends to wander about the waveform graph. This is due to the fact that we start sampling at random times. In order to fix this problem we must use triggering. Triggering will tell our program at what point in the waveform to begin sampling.

1. Open the DAQ Assistant and select Triggering. Settings should be as follows:
Trigger type: Analog Edge
Source: Voltage
Slope: Rising
Level: 0.000
2. Run your VI again and observe its behavior.
3. Play around with the slope and level setting of the trigger and observe. *What happens if our signal never crosses the trigger point?*

3.3 Filtering

The SCXI-1141 module which we have chosen to use for this tutorial contains a software programmable low pass filter. This capability is most often used to reduce high frequency noise, such as the 60 Hz line noise created by the power in the building. This filter is also VERY important in preventing aliasing, which occurs when we try to sample signal data whose frequency is higher than half the sampling frequency.

1. In the DAQ Assistant window select Device >> Lowpass Filter Enable. Select a filter setting of 20 Hz.
2. Turn off the autoscale Y feature of the waveform graph by right clicking on the Y-axis and de-selecting AutoScale Y.

3. Slowly sweep the frequency control on your function above and below your filter frequency and observe what happens. *What happens to a square wave whose frequency is the same as that of the filter? Why does this happen?*

3.4 Using the Analog Outputs to Build a Virtual Function Generator

Now that you have learned to read a real world signal into Labview, we should now learn how to send a software defined voltage signal from Labview back into the real world. In order to do this, we will use DAC 0 output. DAC stands for Digital-to-Analog Converter.

1. Create a new VI. Save it as FunctionGen.VI.
2. Once again place the DAQ Assistant on the block diagram. Select Generate Signals>>Analog Output>>Voltage. Select the Dev1 (our DAQ Card) ao0 as the output and hit Finish. You have now configured the DAC output in your program.
3. Change the Generation Mode to continuous. Hit OK. Hit yes when the next window pops up. Labview will automatically place the DAQ Assistant inside of a loop.
4. Place function Functions>>Waveform>>Analog Waveform>>Waveform Generation>>Basic Function Generator.vi on the block diagram.
5. Build the VI shown in Figure 3. All controls can be created by right clicking on the respective input and selecting Create Control, as can all indicators.
6. Hook the DAC0 output on your labstation panel to the Channel 1 input on your oscilloscope using a BNC to BNC connector.
7. Run your VI, trying different settings for the various inputs to the Basic Function Generator VI. *What is the purpose of the Sampling Info input? How could you change this VI in order to ensure that it always puts out a continuous function?*

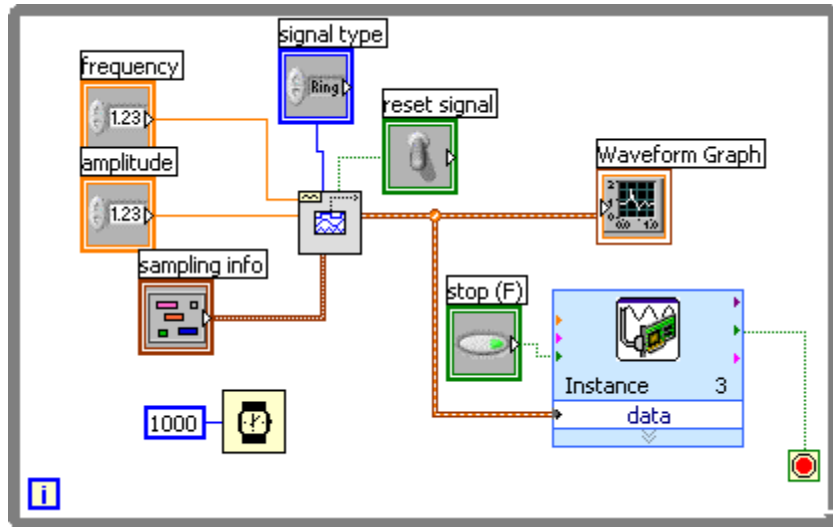


Figure 3: Simple Function Generator VI

Congratulations! We have just implemented a fairly sophisticated program in only a few short steps.

Workshop survey

Please fill out the online survey for this workshop at <http://www.surveymonkey.com/s/labviewv2>. The information you provide will help us provide better instruction in the future!