CONTENTS

SEMINAR OVERVIEW

In this hands-on seminar, you will learn how to build test applications using LabVIEW and NI PXI. At the completion of this course, you should be familiar with acquiring and analyzing data using Soft Front Panels and LabVIEW PXI instrument drivers, building professional user interfaces, and scaling your applications to meet evolving system requirements.

The NI PXI system you are using in this hands-on is equipped to test components of the unit under test (UUT). You will be testing all or a subset of the following components:

- Light Emitting Diode (LED) – validating power supply voltage levels
- Lowpass filter – frequency response measurement
- Transmission Lines – validation of wiring integrity with high speed digital signals
- Thermocouple – temperature measurement

To test these components of the UUT, you will be using the following instruments:

- Digital Multimeter (DMM) – Used to measure the temperature.
- Arbitrary Waveform Generator (FGEN) – Used to generate a frequency sweep to test the filter response.
- Digitizer (SCOPE) – Used to measure the filter signal output and the magnitude of the result.
- Source Measurement Unit (SMU) – Used to supply power and measure the LED.
- High Speed Digital Module (HSDIO) – Outputs a known digital signal and checks that the return signal from the UUT matches the expected waveform.
- Switch – Manages connections between the test sockets containing UUTs and the other hardware. Allows for software controlled switching between UUTs.

EXERCISE 1 – EXPLORE THE LABVIEW ENVIRONMENT

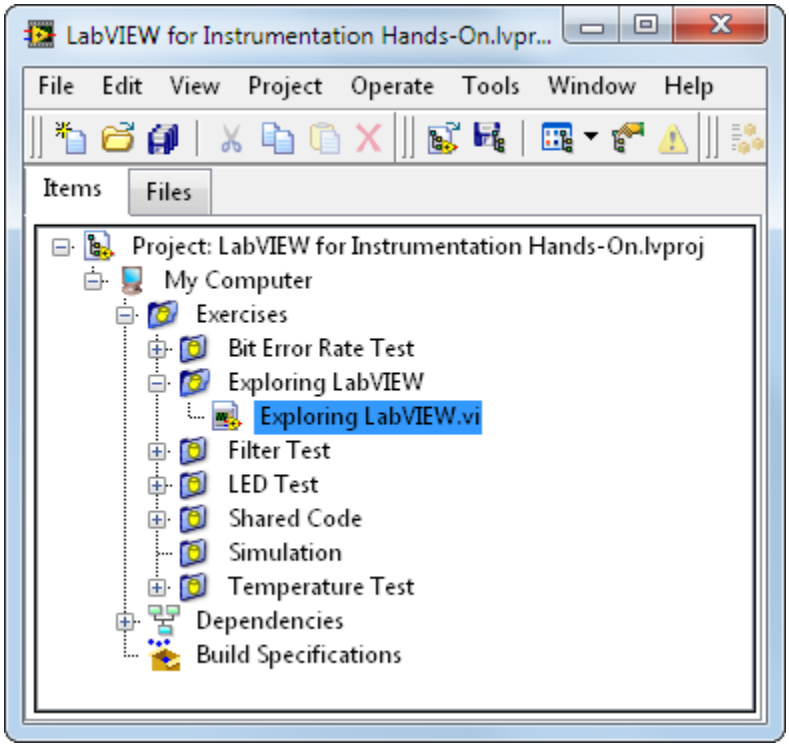
Goal

Explore the LabVIEW environment with an existing application.

Steps

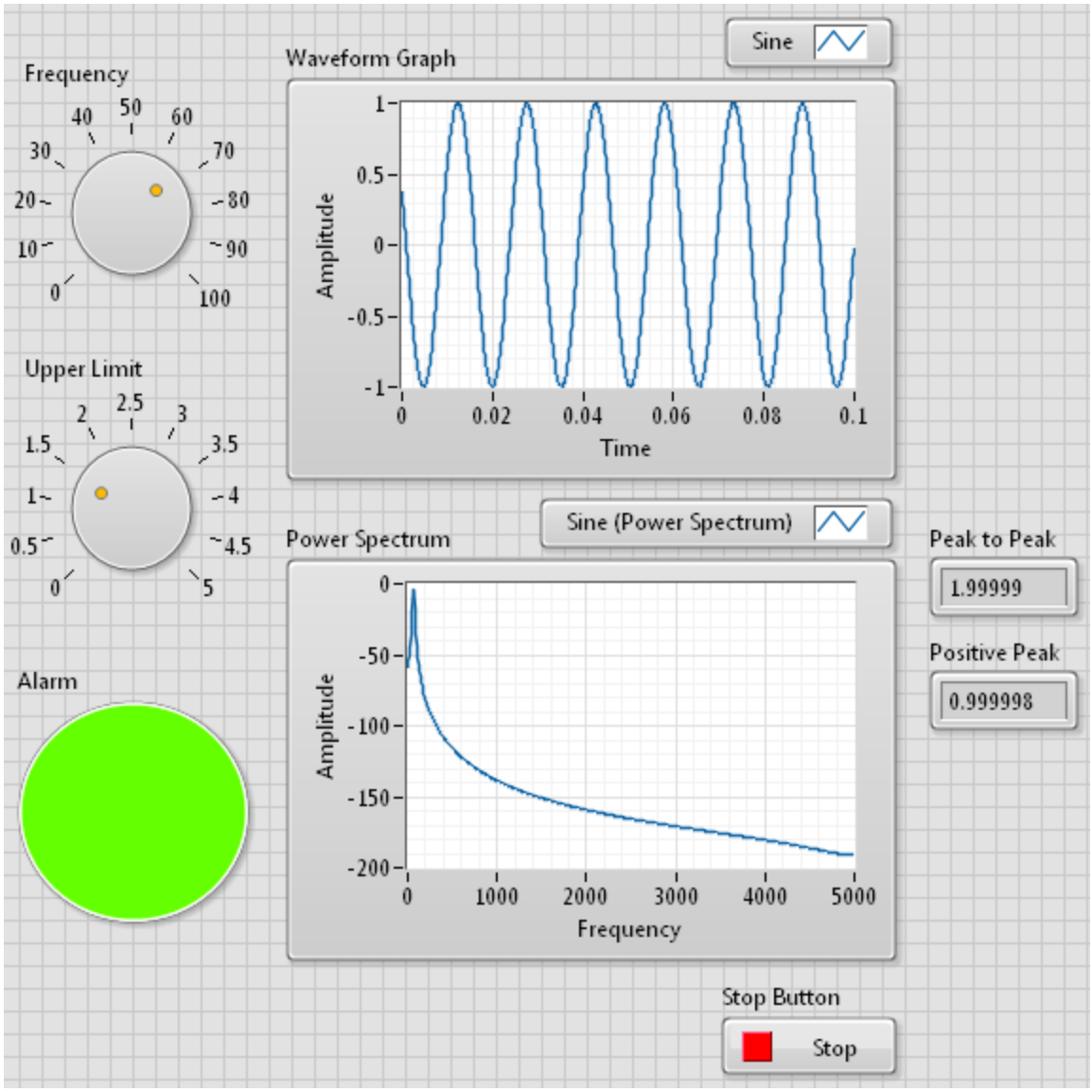
1. Open the LabVIEW for Instrumentation Hands-On Project .
    a. Select **File»Open Project** from the LabVIEW Splash Screen.
    b. Browse to `C:\Seminars\LabVIEW for Instrumentation Hands-On`.
    c. Select LabVIEW for Instrumentation Hands-On.lvprj.
    d. Click **OK**.

    Note: The LabVIEW for Instrumentation Hands-On Project has been created for you so that you can create your LabVIEW tests and organize them in one location. Each folder in the LabVIEW Project file corresponds to a specific test that you will be performing.
2. Open the Exploring LabVIEW application.
    a. Expand the Exploring LabVIEW folder in the Project Explorer window.
    b. Double-click the Exploring LabVIEW.vi.

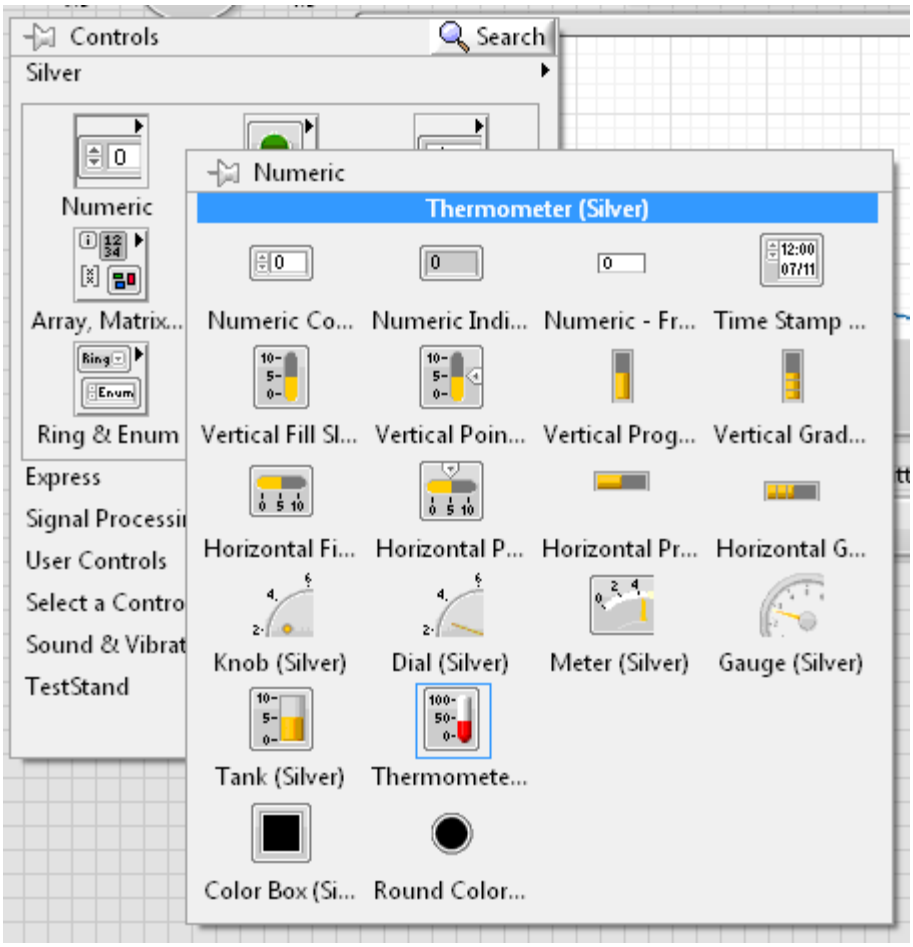


3. Tile the front panel and block diagram by selecting **Window»Tile Left and Right**.

4. The front panel looks similar to the following user interface.

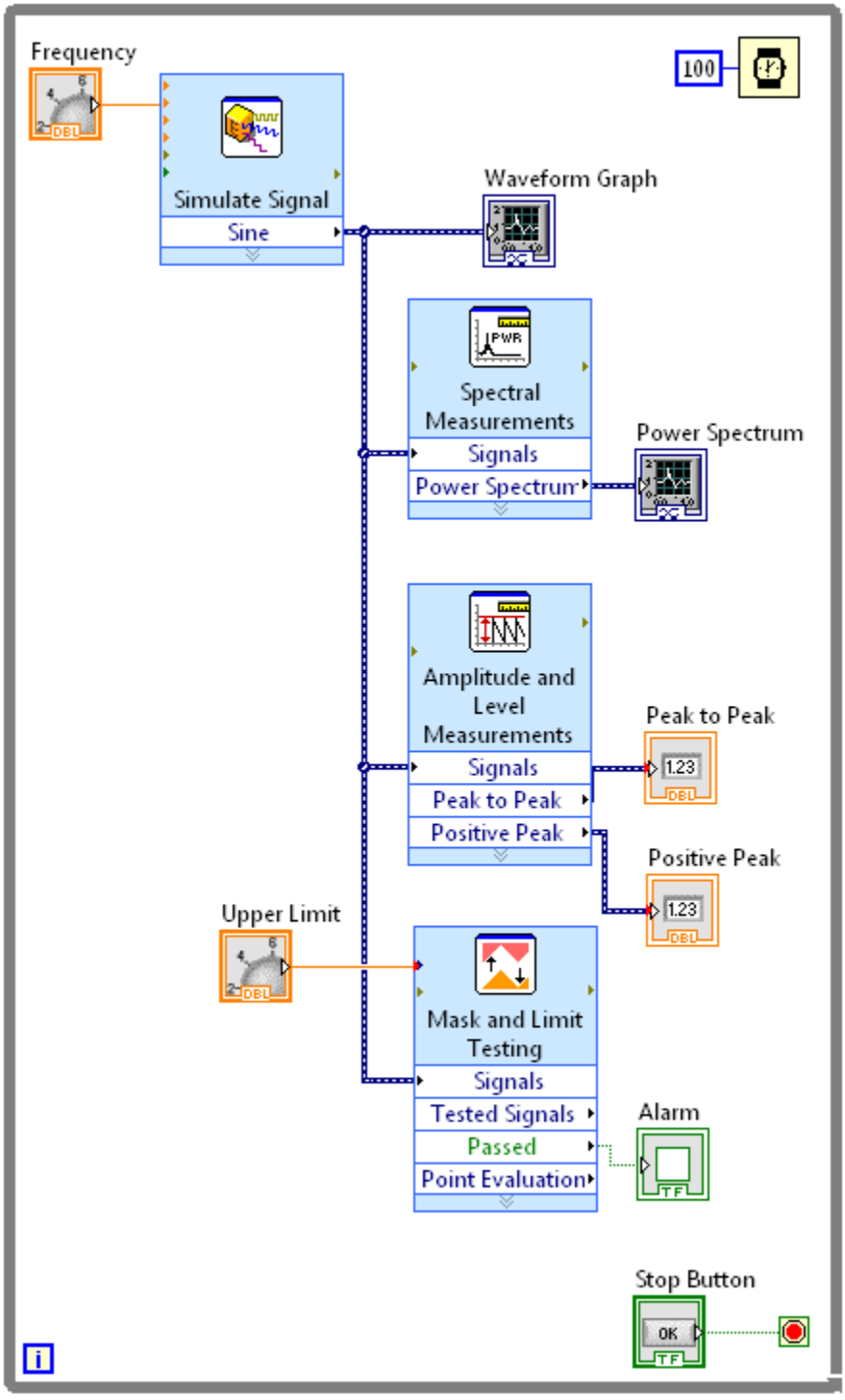


5. Right-click on the front panel and notice that the Controls Palette appears.



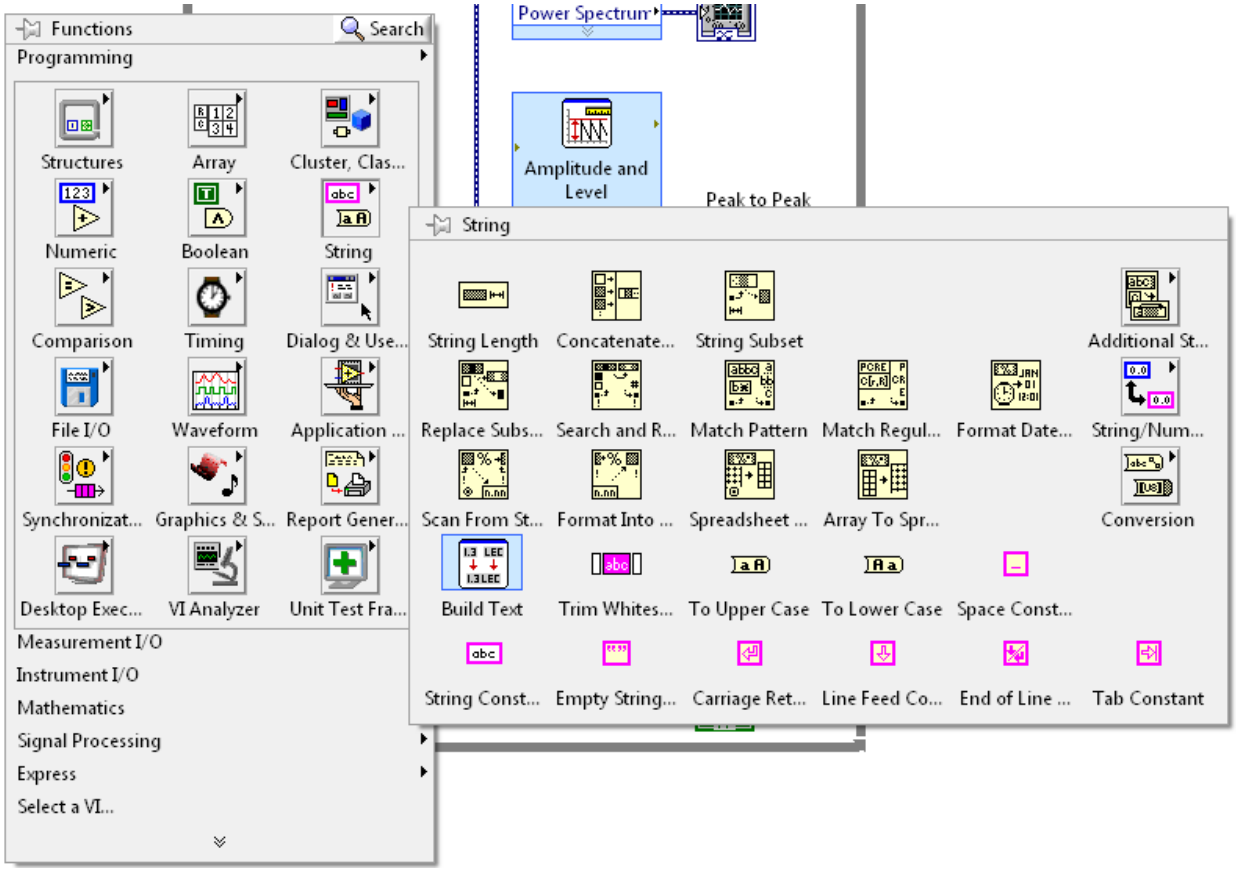
Tip: You can also use <Ctrl-Space> to bring up the Controls Quick Drop window.

6. The block diagram looks similar to the following:



7. Right-click on the block diagram and notice that the Functions Palette appears.

   Tip: You can also use <Ctrl-Space> to bring up the Functions Quick Drop window.



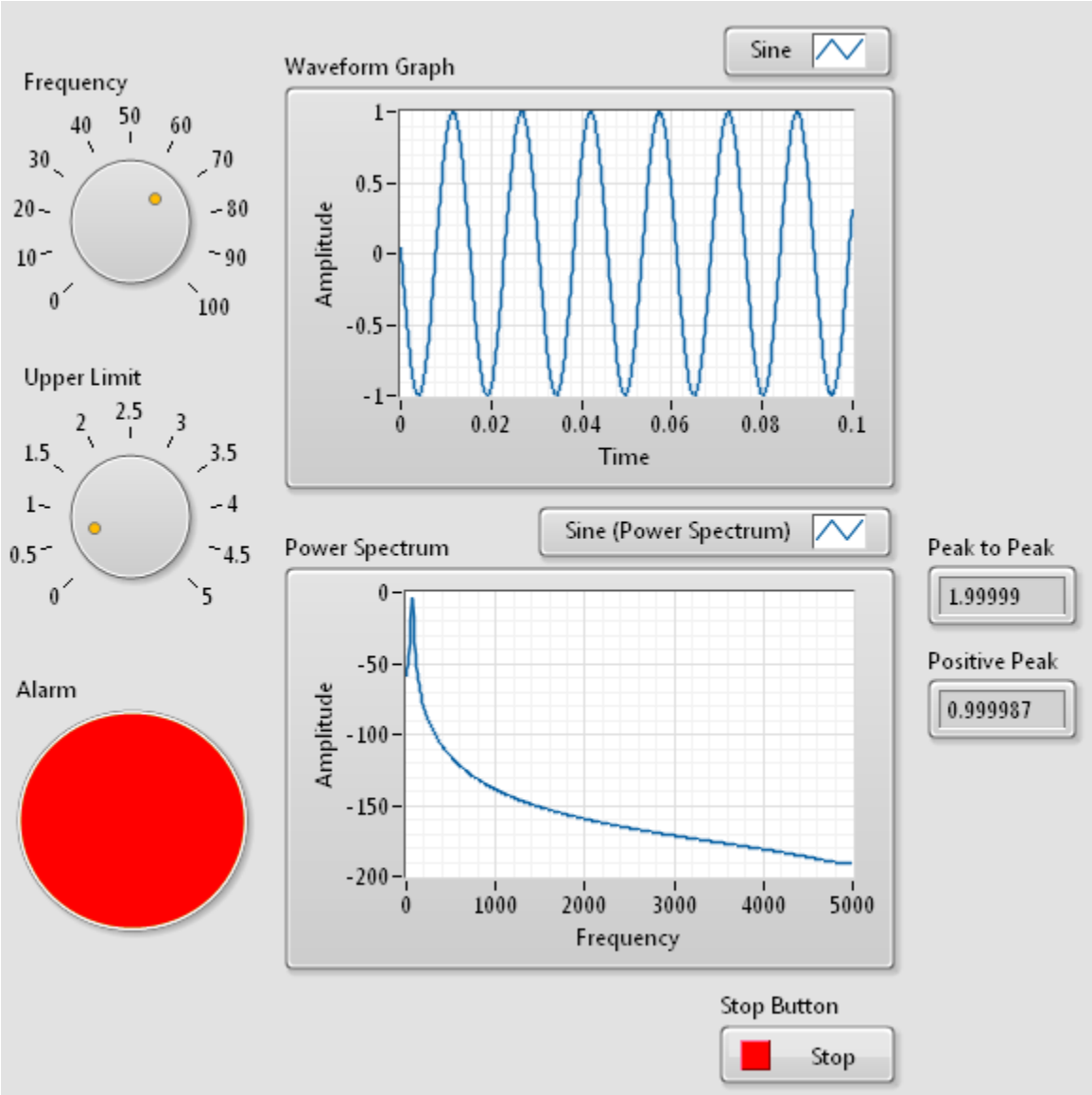
8. Open the Context Help by selecting **Help»Show Context Help**.

   Tip: You can also open the Context Help using <Ctrl-H>.

9. Run the application by selecting the **Run** button.

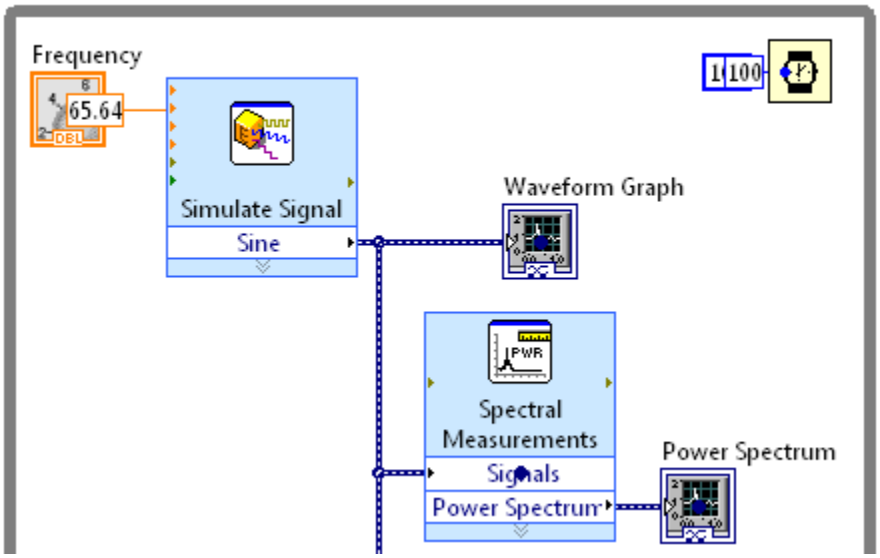
10. Adjust the Frequency and Upper Limit knobs.

11. Notice that the data on the Graphs changes and the Alarm is triggered if the signal exceeds the Upper Limit.



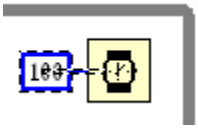
12. Select Highlight Execution by clicking on the light bulb in the block diagram menu.

13. Notice that you can see the data flowing through the block diagram functions.

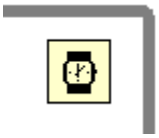


14. Deselect **Highlight Execution**.
15. Click the **Stop** button to stop the application.

16. Delete the Numeric Constant wired to the **milliseconds to wait** input terminal of the **Wait (ms)** function.
    a. Select the **Numeric Constant**.



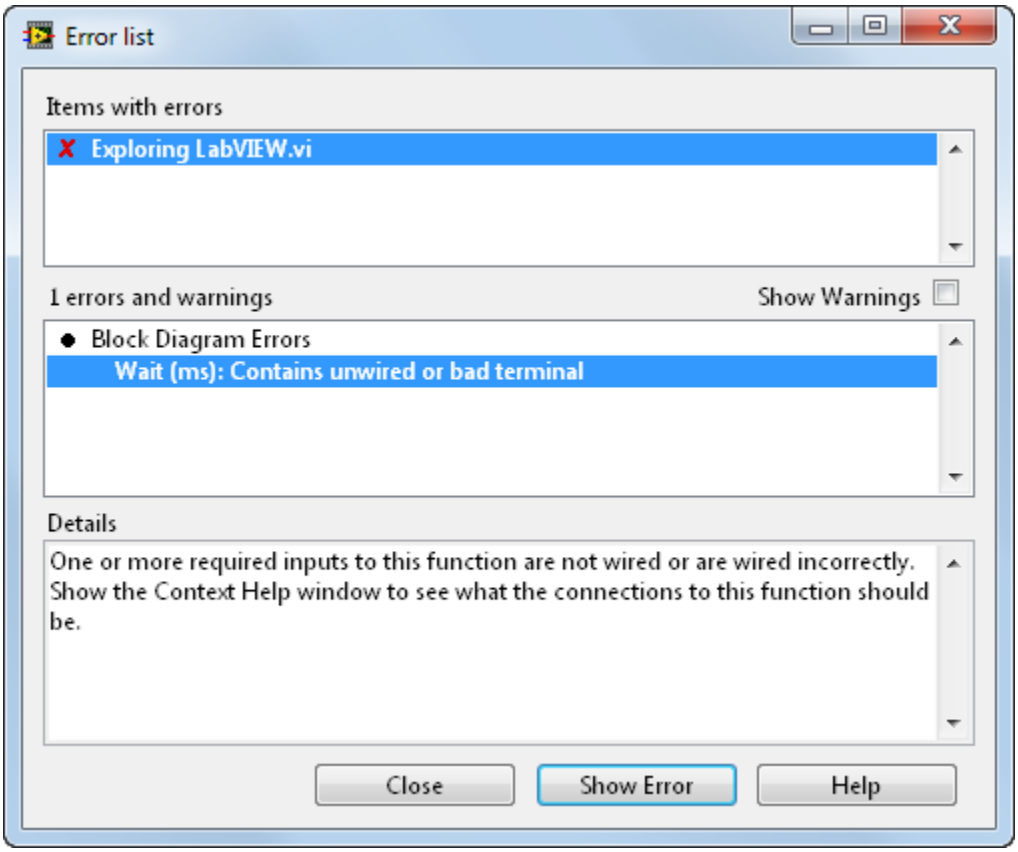
    b. Press **Delete**.



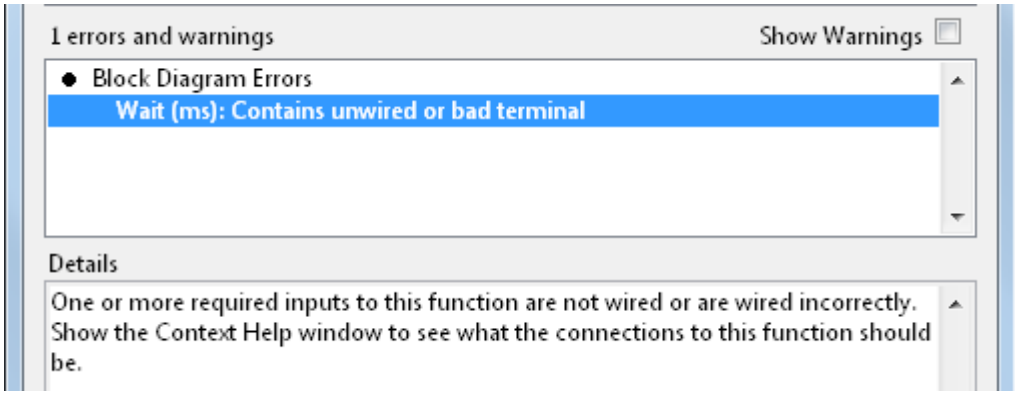
17. Notice that the Run arrow is now broken.

    Note: This means that there is an error in the code and the application will not compile or run.

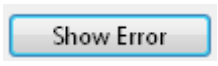
18. Click the broken Run arrow to bring up the Error List.



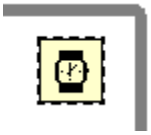
19. Select the **Wait (ms): Contains unwired or bad terminal** error in the errors and warnings list.



20. Select **Show Error**.

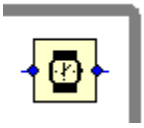


21. Notice that this takes you to the area of the code that has an error – the Wait (ms) function.



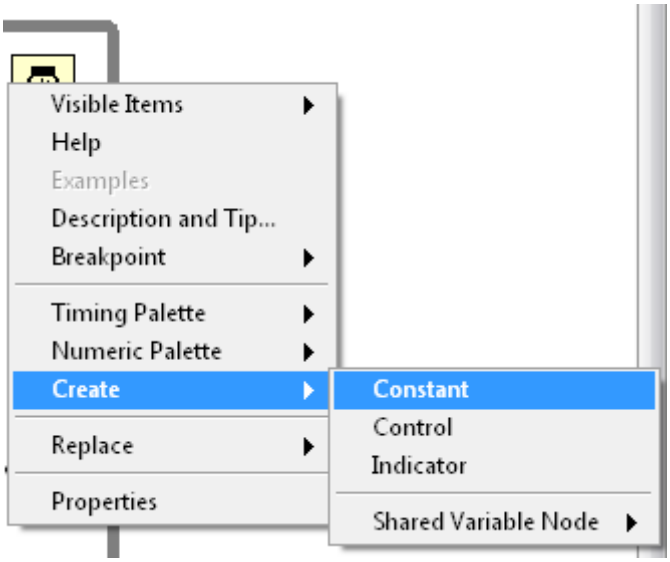
22. Fix the error.

    a. Hover over the Wait (ms) function with your mouse so that you can see the input and output terminals (blue dots).

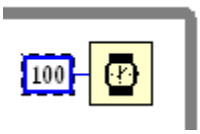


    b. Hover over the milliseconds to wait input terminal on the left side of the Wait (ms) function.

    c. Right-click on the milliseconds to wait input terminal on the left side of the Wait (ms) function and select **Create»Constant**.



    d. Type `100` as the Numeric Constant value.



23. Notice that the Run arrow is no longer broken.

24. **Run** the application by selecting the Run button.
25. Click the **Stop** button to stop the application.

26. **Save** and **Close** the application.

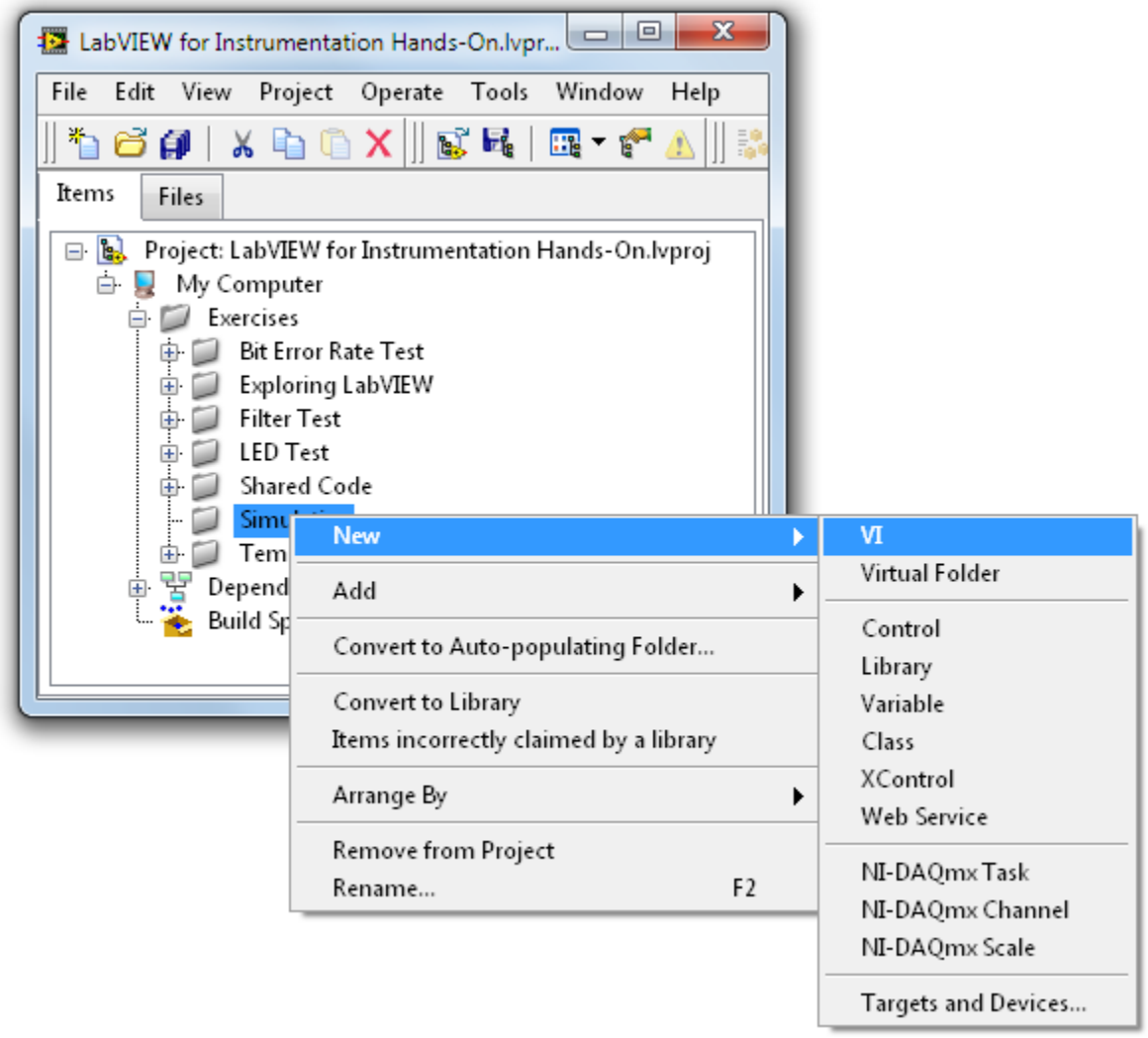
EXERCISE 2 – SIMULATE AND PLOT SIGNAL

Goal

Create a LabVIEW application that will simulate and plot a signal on a Waveform Graph.

Steps

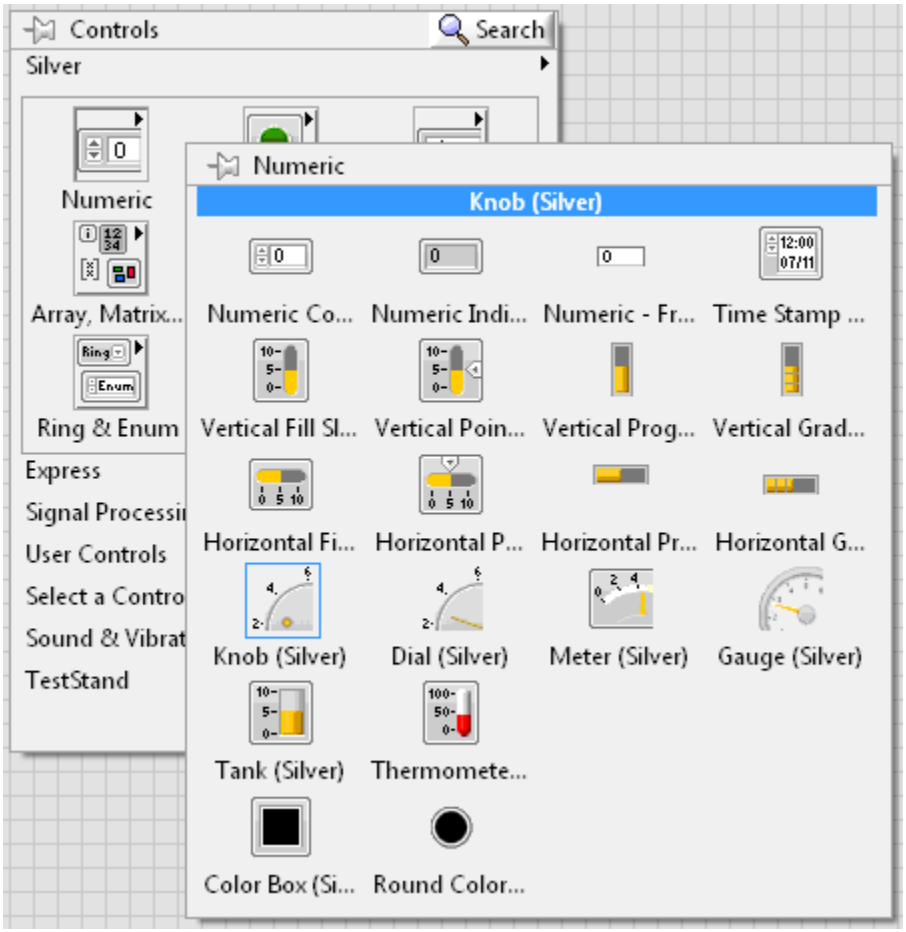
1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browsing to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the LabVIEW for Instrumentation Hands-On.lvprj file.
2. Create a new VI in the Simulation folder by right-clicking on the **Simulation** folder and selecting **New»VI**.



3. On the VI, Select **File»Save** and name the VI **Simulate and Plot Signal**.
4. Tile the front panel and block diagram by selecting **Window»Tile Left and Right**.

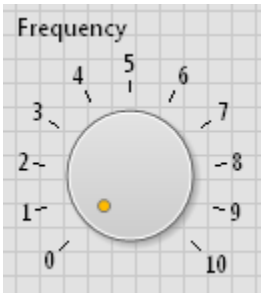
Tip: You can use <Ctrl-T> to tile the front panel and block diagram.

5. Add a Knob to the front panel.
    a. Right-click on the front panel.
    b. Navigate to **Silver»Numeric»Knob**.

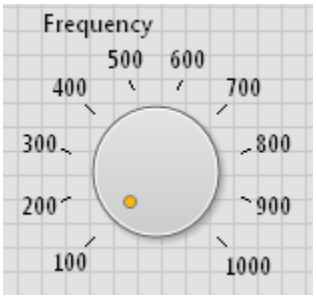


    c. Drag and drop the Knob on to the front panel.

6. Name the Numeric Knob **Frequency**.



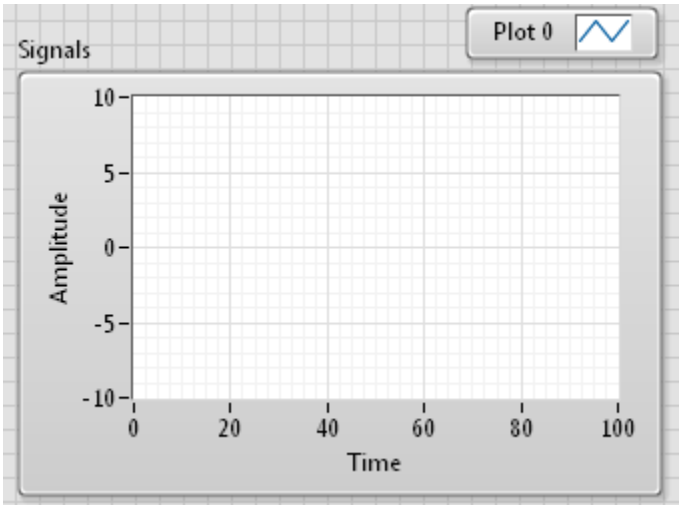
7. Change the range of the Frequency Knob to **[100, 1000]**.
    a. Double-click the **0** on the Frequency Knob range and change the value to **100**.
    b. Double-click the **10** on the Frequency Knob range and change the value to **1000**.



8. Add a **Waveform Graph** to the front panel.
    a. Right-click on the front panel.

    Tip: you can use <Ctrl-Space> to bring up the Controls Quick Drop window.
    b. Navigate to **Silver»Graph»Waveform Graph**.
    c. Drag and drop the Waveform Graph onto the front panel.
9. Name the Waveform Graph **Signals**.

    Tip: You can resize LabVIEW controls on the front panel by dragging the corners.

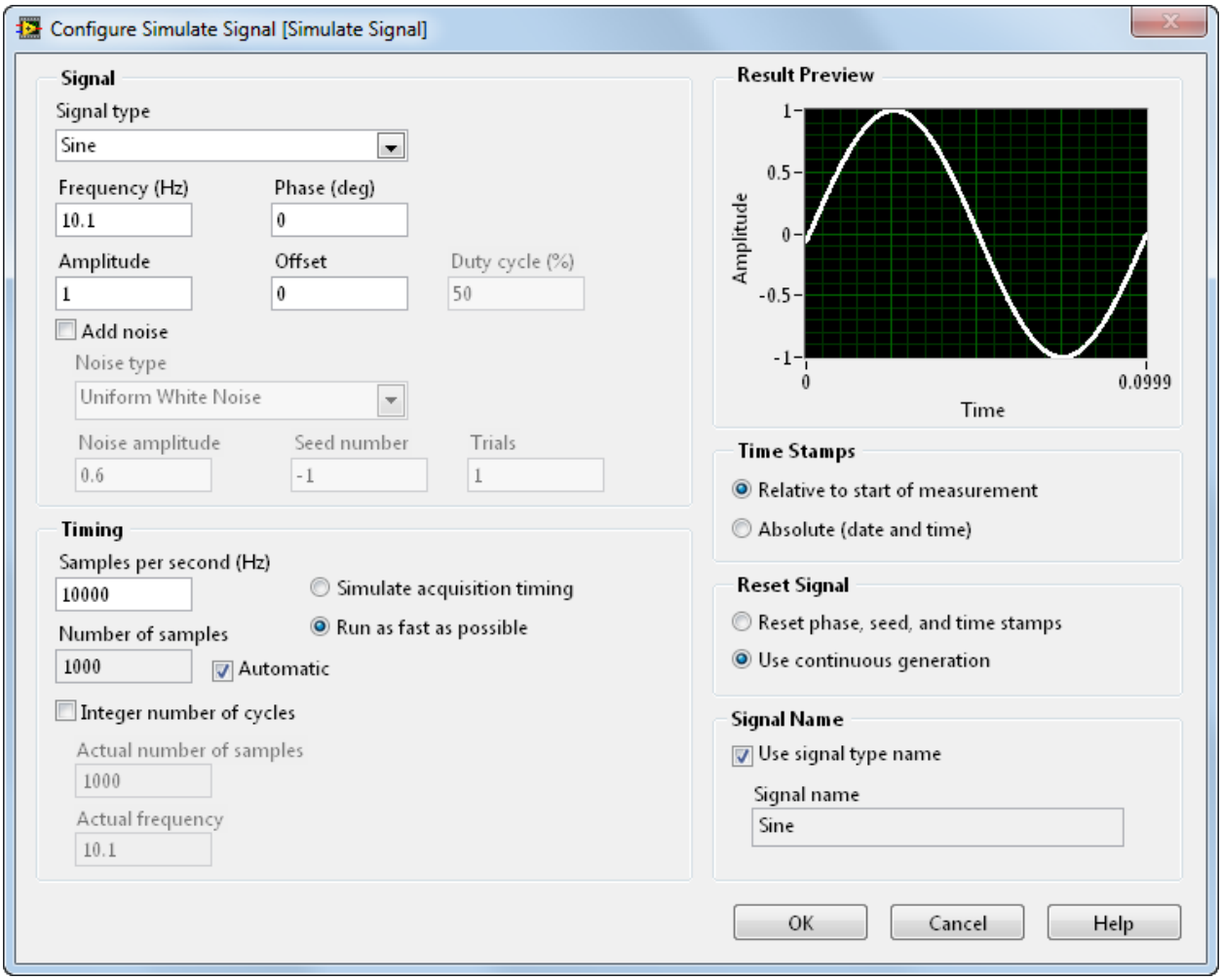


10. If you do not see the block diagram, switch to it by selecting **Window»Show block diagram** from the VI menu.

    Tip: You can also toggle between the front panel and the block diagram using <Ctrl-E>.

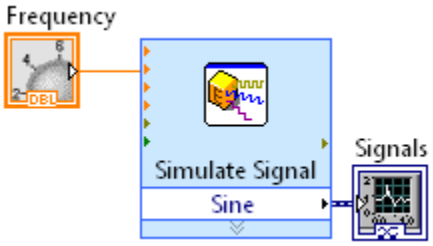
11. Add a **Simulate Signal Express VI** to the block diagram.
    a. Right-click on the block diagram anywhere in the white space.

       Tip: you can use <Ctrl-Space> to bring up the Functions Quick Drop window.
    b. Navigate to **Express»Input»Simulate Signal**.
    c. Drag and drop the Simulate Signal Express VI on to the block diagram.
12. Change the Samples per second (Hz) to **10,000** in the Configure Simulate Signal Window.



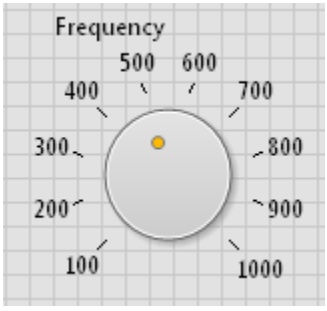
13. Select **OK**.
14. Wire the Frequency control into the Frequency input terminal of the Simulate Signal Express VI.
15. Wire the Sine data from the Simulate Signal Express VI to the Signals Waveform Graph.

16. Clean up your block diagram code by selecting **Edit»Clean Up Diagram** from the block diagram VI menu.



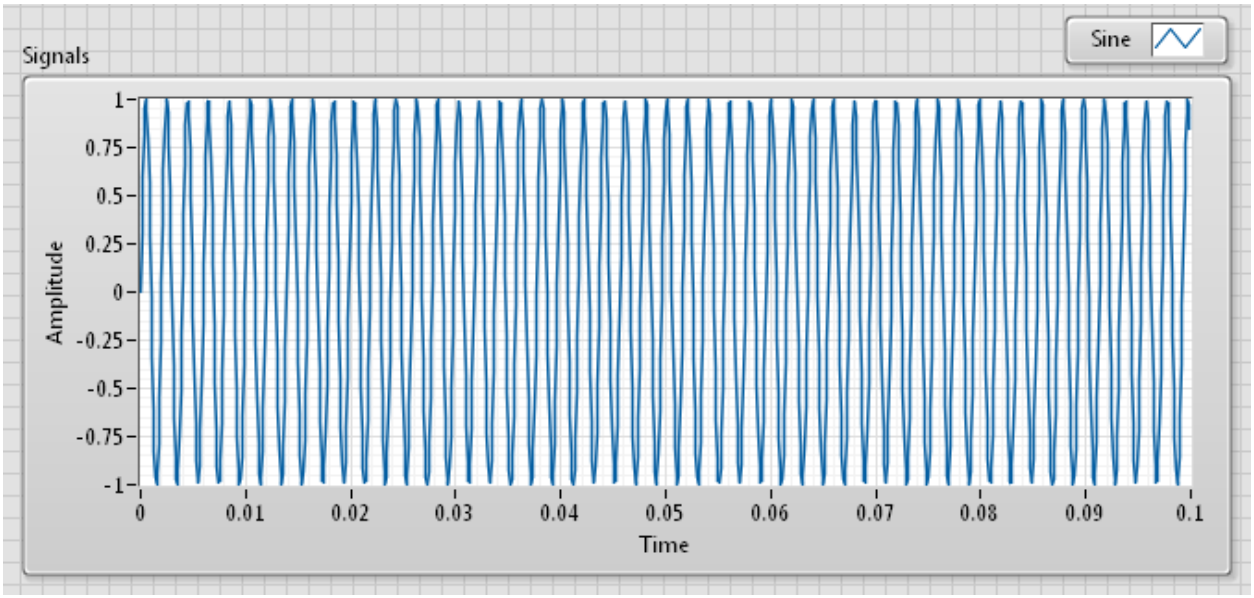
Tip: you can use <Ctrl-U> to clean up the block diagram.

17. Toggle to the front panel by using <Ctrl-E>.
18. Adjust the Frequency Knob on the front panel to approximately **400 Hz**.



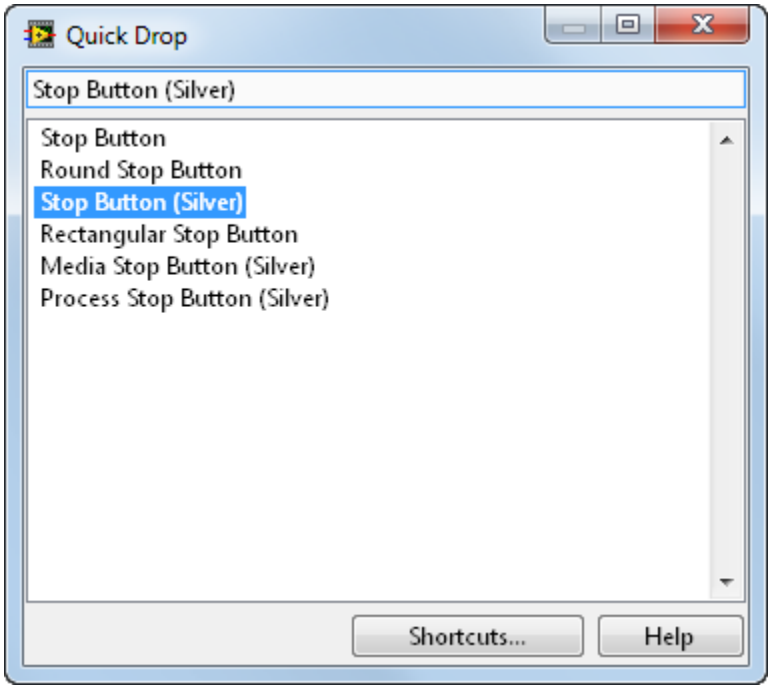
19. Run the application by clicking the **Run** arrow.

20. Observe the simulated signal on the Waveform Graph.

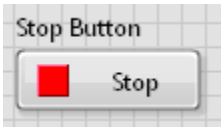


Note: You will notice that this application only generates a finite signal at one frequency. In order to modify the application to run continuously and generate a continuous signal in which the frequency can be changed, we will need to modify our code.

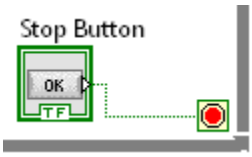
21. Add a Stop Button to the front panel.
    a. Use <Ctrl-Space> to bring up the Controls Quick Drop window.
    b. Type `Stop Button` into the Quick Drop search window.
    c. Double-click **Stop Button (Silver)**.



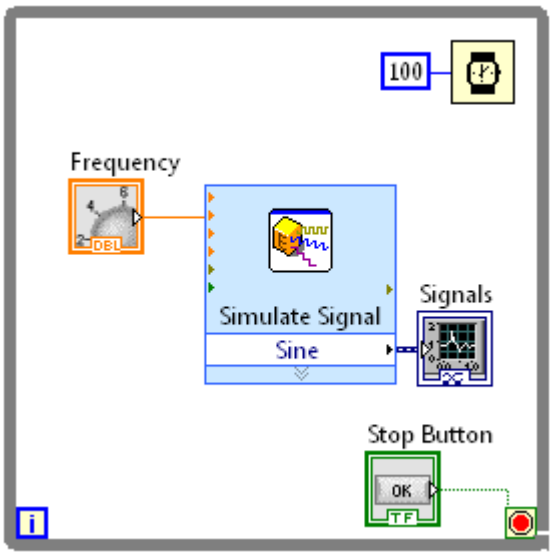
    d. Drop the Stop Button onto the front panel.



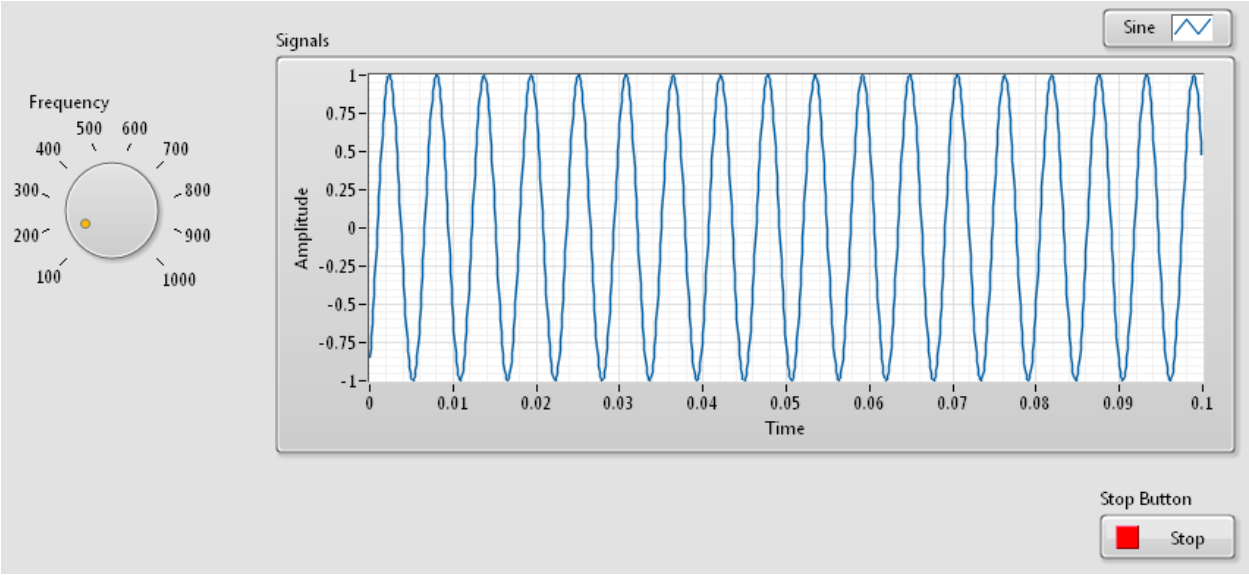
22. Toggle to the block diagram using <Ctrl-E>.
23. Add a While Loop to the block diagram around all of your code.
    a. Right-click on the block diagram.
    b. Navigate to **Programming»Structures»While Loop**.
    c. Drag and draw the While Loop around all of the code.
24. Wire the Stop Button into the While Loop condition.



25. Add timing to the While Loop.
    a. Right-click on the block diagram.
    b. Navigate to **Programming»Timing»Wait (ms)**.
    c. Drag and drop the **Wait (ms)** function in to the While Loop.
    d. Right-click on the milliseconds to wait input terminal of the Wait (ms) function and select **Create»Constant**.
    e. Change the numeric constant value to **100**.



26. Run the application once more and observe that the signal changes as you adjust the Frequency Knob.



27. Press the **Stop** button to stop the application.
28. Save the **Simulate and Plot Signal** application, but *do not close it*.

EXERCISE 3 – SIMULATED FILTER TEST

Goal

The goal of this exercise is to create a simulated lowpass filter test. In the application, you will generate a frequency sweep of simulated signals and supply each to a simulated lowpass filter. The amplitude of each filtered signal will be measured in order to determine when the signal has reached the -3dB point. The frequency of this signal will then be used to perform a limit test in which it will be compared to the lower and upper limits you specify. A pass or fail value will be returned based on the results of the limit test.

Steps

1. Create a copy of the Simulate and Plot Signal VI you just created in Exercise 2.
    a. Select **File»Save As**.
    b. Select **Create unopened disk copy**.
    c. Click **Continue**.
    d. Browse to the Simulation Folder `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Simulation`.
    e. Name the VI **Simulated Filter Test**.

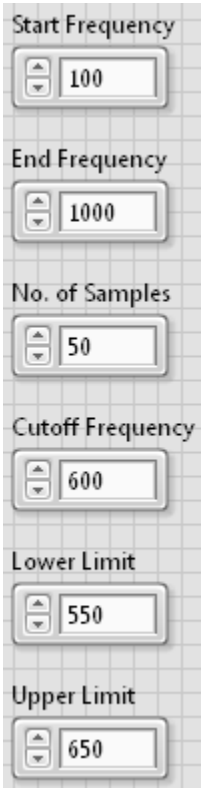
    Note: A solutions folder is available `C:\Seminars\LabVIEW for Instrumentation Hands-On\Solutions` so that you can access any of the completed LabVIEW applications in the event that you are unable to complete the exercises in this hands-on.
    f. Click **OK.**
    g. Add the New VI to the Simulation Folder in the project Explorer.
        i. Right-click the Simulation folder in the Project Explorer.
        ii. Select **Add»File**.
        iii. Browse to Browse to the Simulation Folder `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Simulation`.
        iv. Select Simulated Filter Test.vi.
        v. Click **Add File**.
2. Tile the front panel and block diagram using <Ctrl-T>.
3. Delete the Frequency Knob and the Stop Button from the front panel.
4. Add six Numeric Controls to the front panel.
    a. Right-click on the front panel.
    b. Navigate to **Silver»Numeric»Numeric Control**.

c. Drag and drop the Numeric Control on to the front panel.

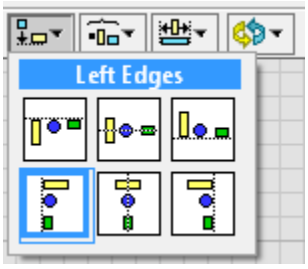
d. Repeat steps 4.a. - 4.c. five times.

Tip: You can make copies of the first Numeric Control you added to the front panel by holding the <Ctrl> key while clicking and dragging a control on the front panel.

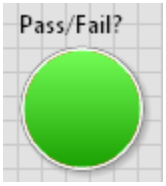
5. Place the Controls on the left side of the front panel and give them the following names and values.



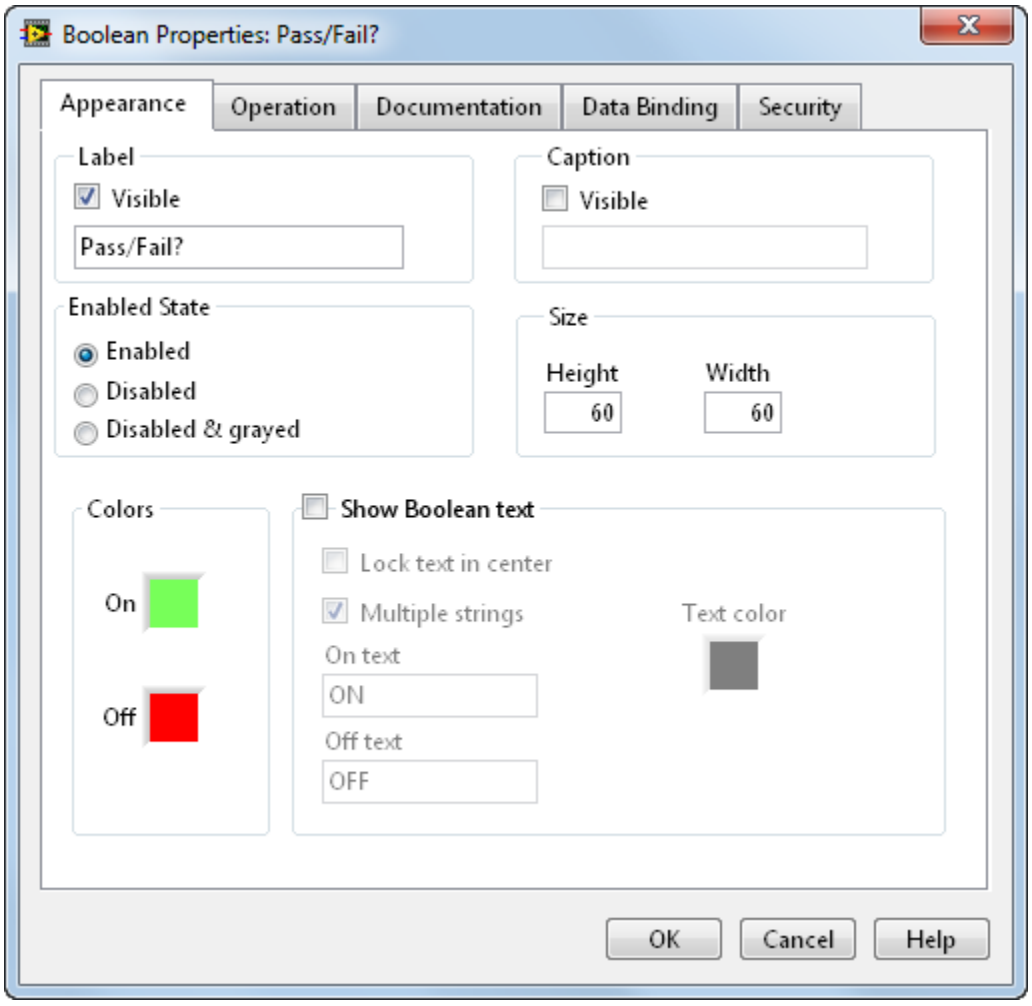
Tip: You can align multiple controls by selecting all of them and then selecting **Align objects»Left Edges** from the front panel Menu.



6. Add an LED to the front panel.
    a. Right-click on the front panel.
    b. Navigate to **Silver»Boolean»LED**.
    c. Drag and drop the LED on to the front panel.
    d. Resize the LED by dragging the circle out.
    e. Name the LED **Pass/Fail?** .



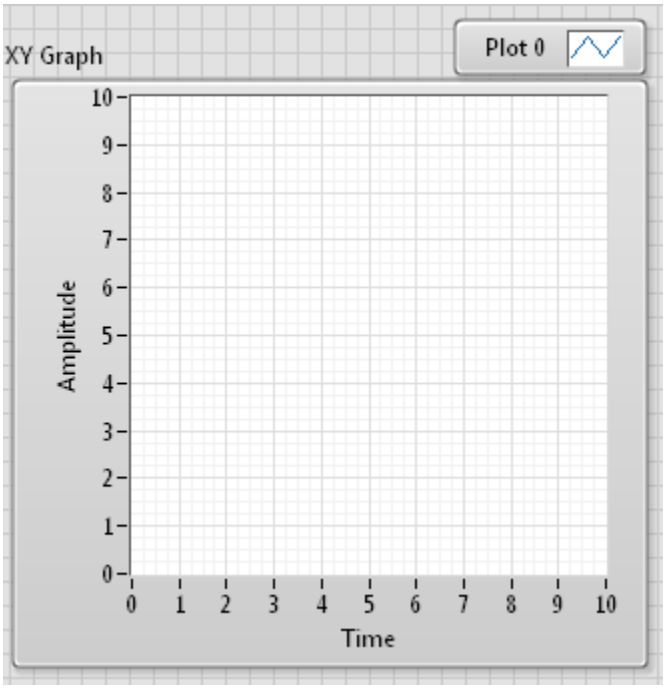
7. Change the Off color of the LED to **Red**.
    a. Right-click the LED and select **Properties**.
    b. Click the Off color and select **Red**.



    c. Click **OK**.

8. Expand the Signals Waveform Graph Legend to show two signals by dragging the Legend upwards.

9. Add an **XY Graph** to the front panel.
    a. Right-click the front panel.
    b. Navigate to **Silver»Graph»XY Graph**.
    c. Drag and drop the XY Graph onto the front panel.

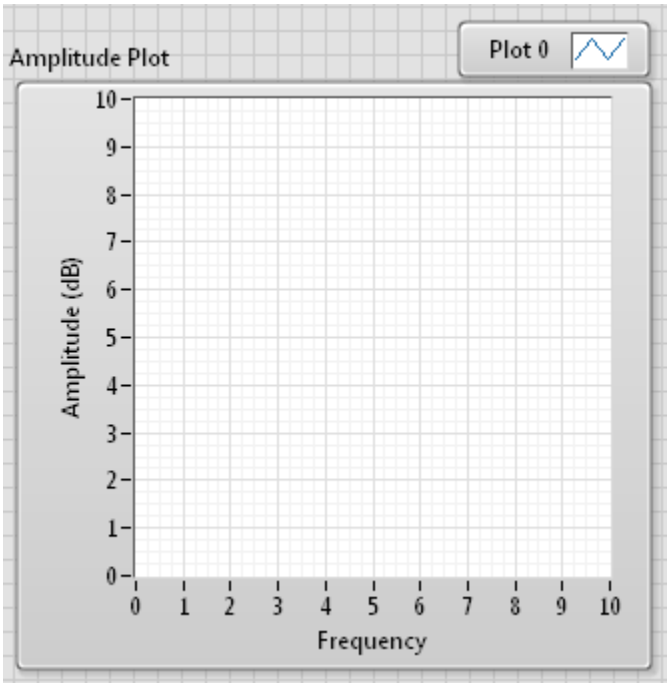


d. Name the XY Graph **Amplitude Plot.**

e. Change the Y-Axis label of the Amplitude Plot XY Graph to **Amplitude(dB).**

Tip: You can modify graph labels directly on the front panel by double-clicking on the label name.

f. Change the X-Axis label of the Amplitude Plot XY Graph to **Frequency.**



g. Change the Amplitude Plot XY Graph X Axis and Y Axis to **not autoscale.**

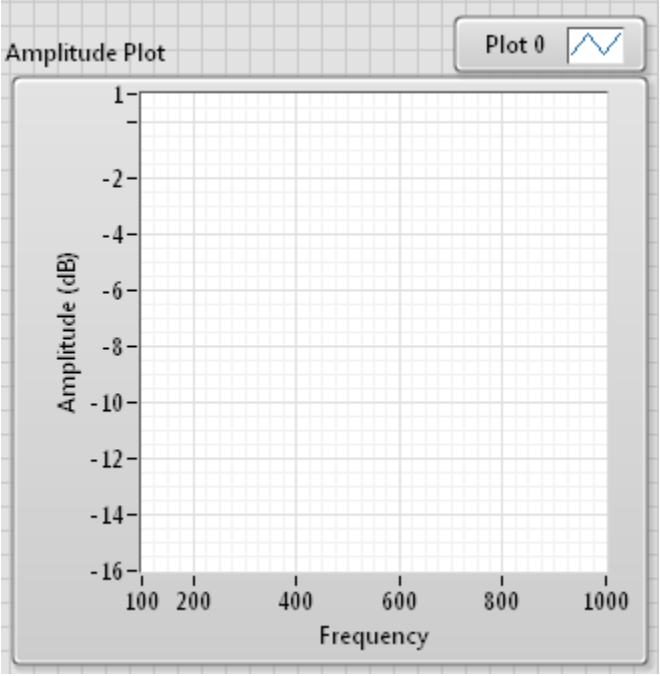
   i. Right-click on the X-Axis and deselect **AutoScale X.**

   ii. Right-click on the Y-Axis and deselect **AutoScale Y.**

h. Change the range of the X-Axis to **[100, 1,000]**.

i. Change the range of the Y-Axis to **[-16, 1]**.

Tip: You can modify the graph axis range directly on the front panel by double-clicking on the axis values.

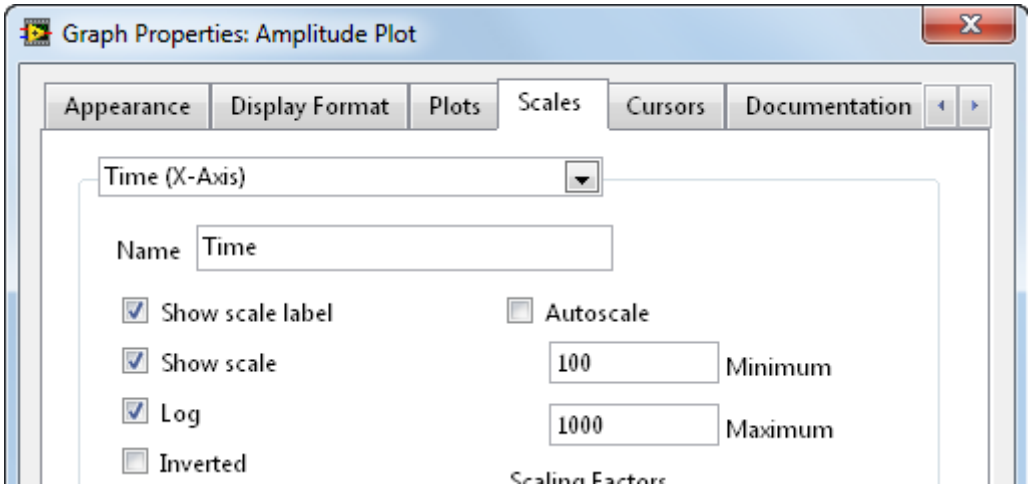


j. Change the scale of X-Axis of the Amplitude Plot XY Graph to **Log**.

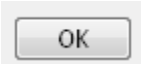
i. Right-click the Amplitude Plot XY Graph and select **Properties**.

ii. Select the **Scales** tab.

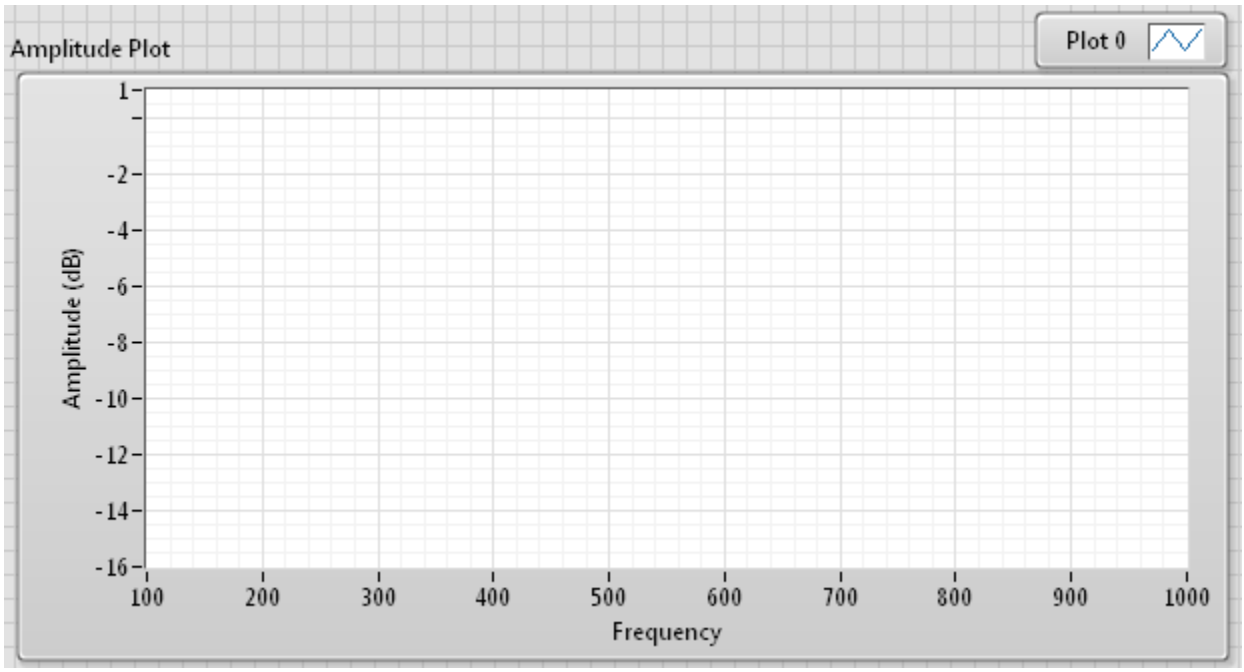
iii. Check **Log**.



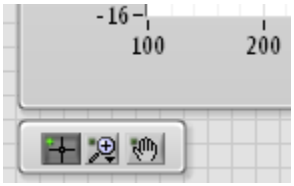
iv. Click **OK**.



10. Resize the Amplitude Plot to be larger by dragging the corners of the XY Graph.

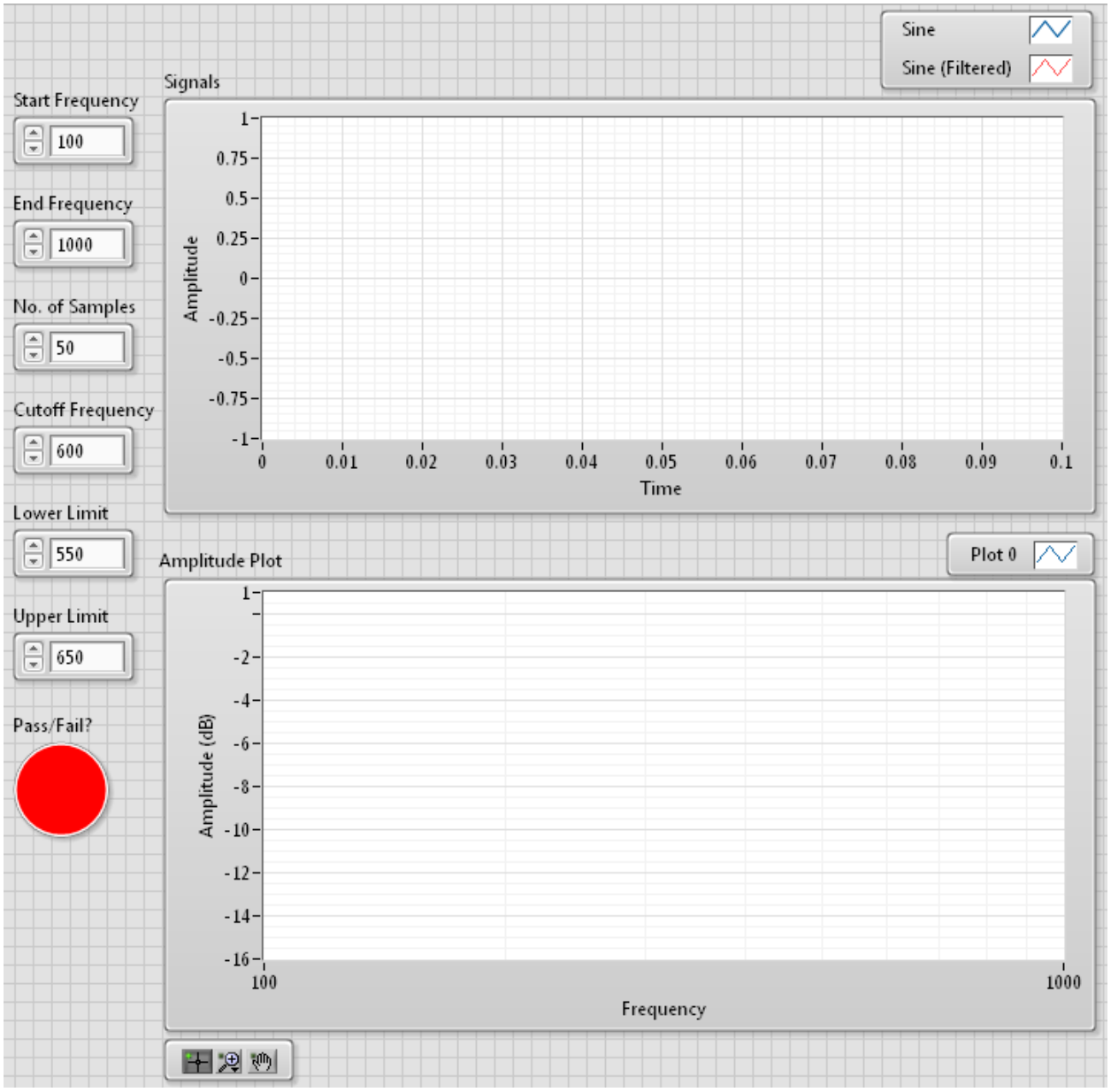


11. Add a Graph Palette to the Amplitude Plot XY Graph.
    a. Right-click the **Amplitude Plot** XY Graph.
    b. Select **Visible Items»Graph Palette**.



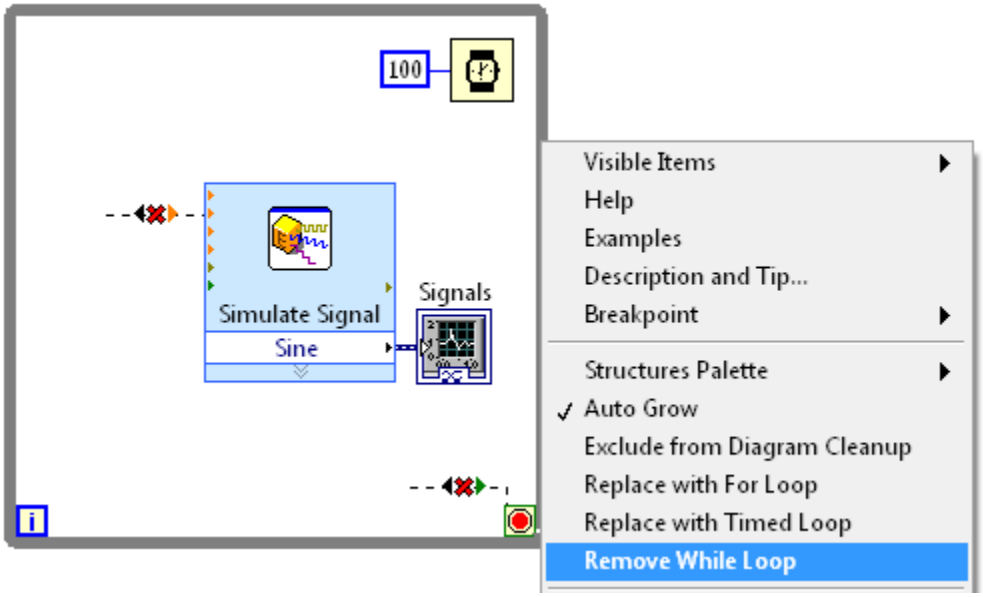
12. Make the current vales default by selecting **Edit»Make Current Values Default**.

13. Your front panel should look similar to the following user interface:



14. Toggle to the block diagram using <Ctrl-E>.

15. Remove the While Loop by right-clicking on the While Loop and selecting **Remove While Loop**.



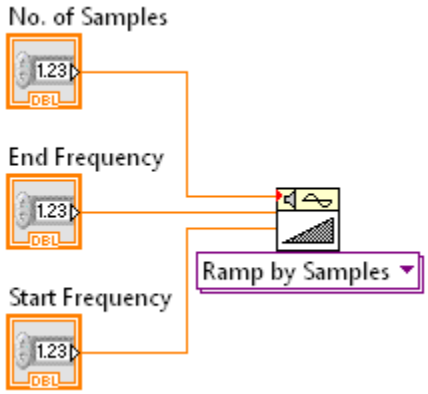
Note: Do not delete the While Loop – this will delete all of the code.

16. Remove the broken wires on the block diagram by selecting **Edit»Remove Broken Wires** from the block diagram menu.

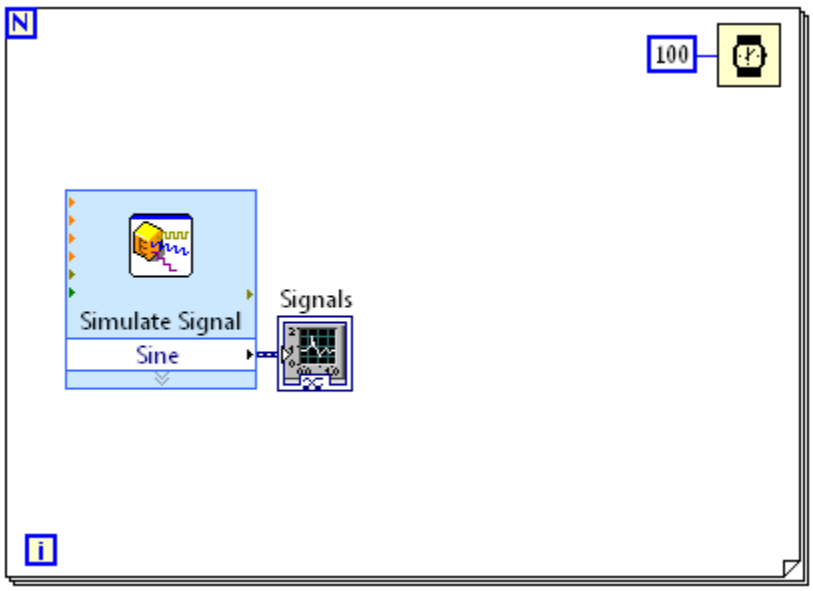
    Tip: You can also remove broken wires using <Ctrl-B>.

17. Add a Ramp Pattern function to the block diagram.
    a. Right-click the block diagram.
    b. Navigate to **Signal Processing»Signal Generation»Ramp Pattern**.
    c. Drag and drop the Ramp Pattern function on to the block diagram.

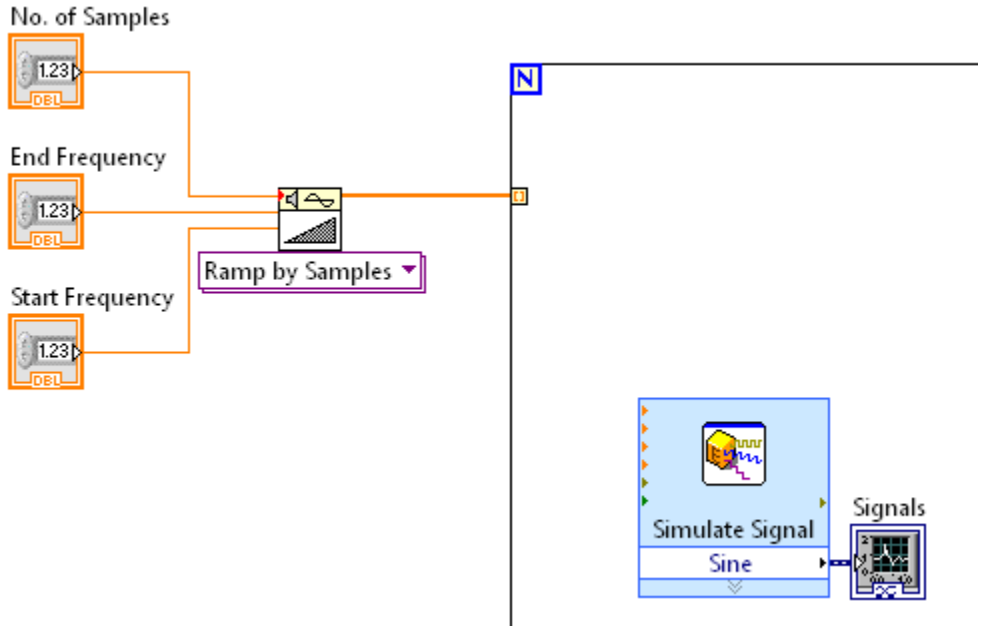
18. Wire the following Icons into the Ramp Pattern function input terminals.
    a. Wire the **No. of Samples** control into the **samples** input terminal of the **Ramp Pattern** function.
    b. Wire the **Start Frequency** control into the **start** input terminal of the **Ramp Pattern** function.
    c. Wire the **End Frequency** control into the **end** input terminal of the **Ramp Pattern** function.



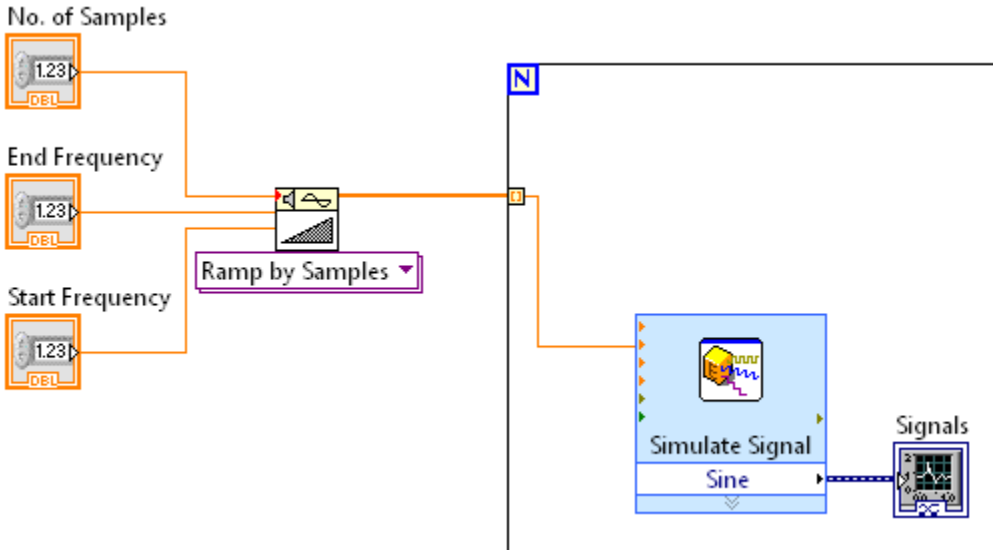
19. Add a For Loop to the block diagram.
    a. Right-click on the block diagram.
    b. Navigate to **Programming»Structures»For Loop**.
    c. Drag and draw the For Loop around the Simulate Signal Express VI, the Signals Waveform Graph**,** the Wait function and numeric constant.



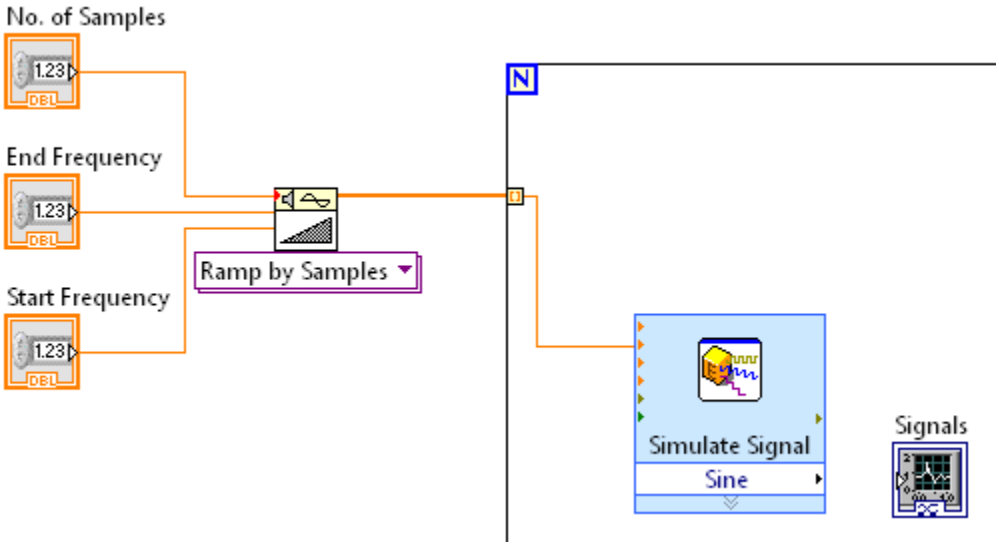
20. Wire the **Ramp Pattern** output terminal of the **Ramp Pattern** function to the left side of the **For Loop**.



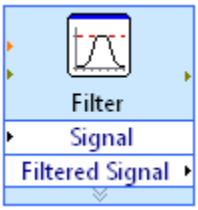
21. Wire the **Ramp Pattern** input from the side of the **For Loop** to the **Frequency** input terminal of the **Simulate Signal Express VI**.



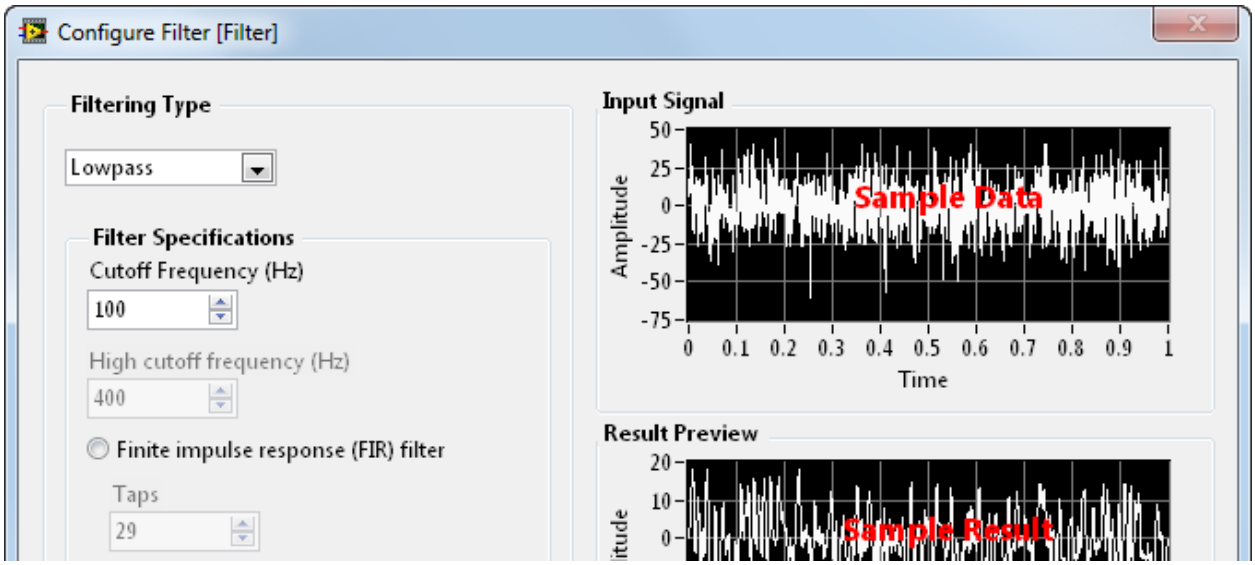
22. Delete the **wire** connecting the **Sine** output terminal of the Simulate Signal Express VI to the **Signals Waveform Graph**.



23. Add a **Filter Express VI** to the For Loop.
    a. Right-click on the block diagram.
    b. Navigate to **Express»Signal Analysis»Filter**.
    c. Drag and drop the Filter Express VI into the For Loop.

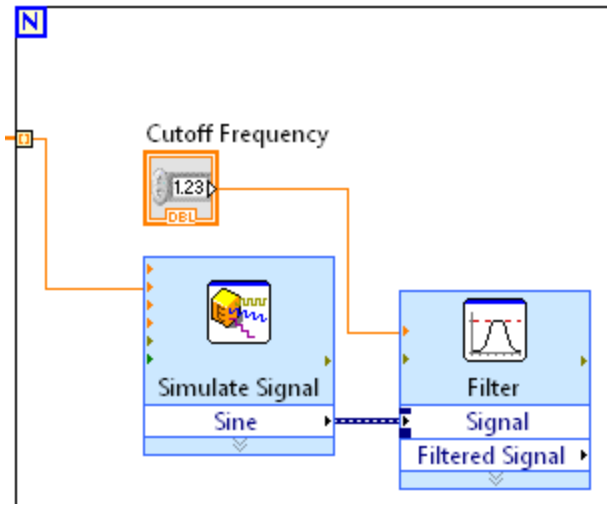


d. In the Configure Filter Window, leave the default settings.

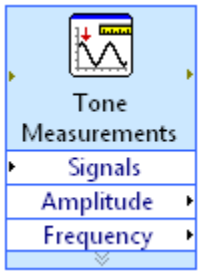


e. Select **OK**.

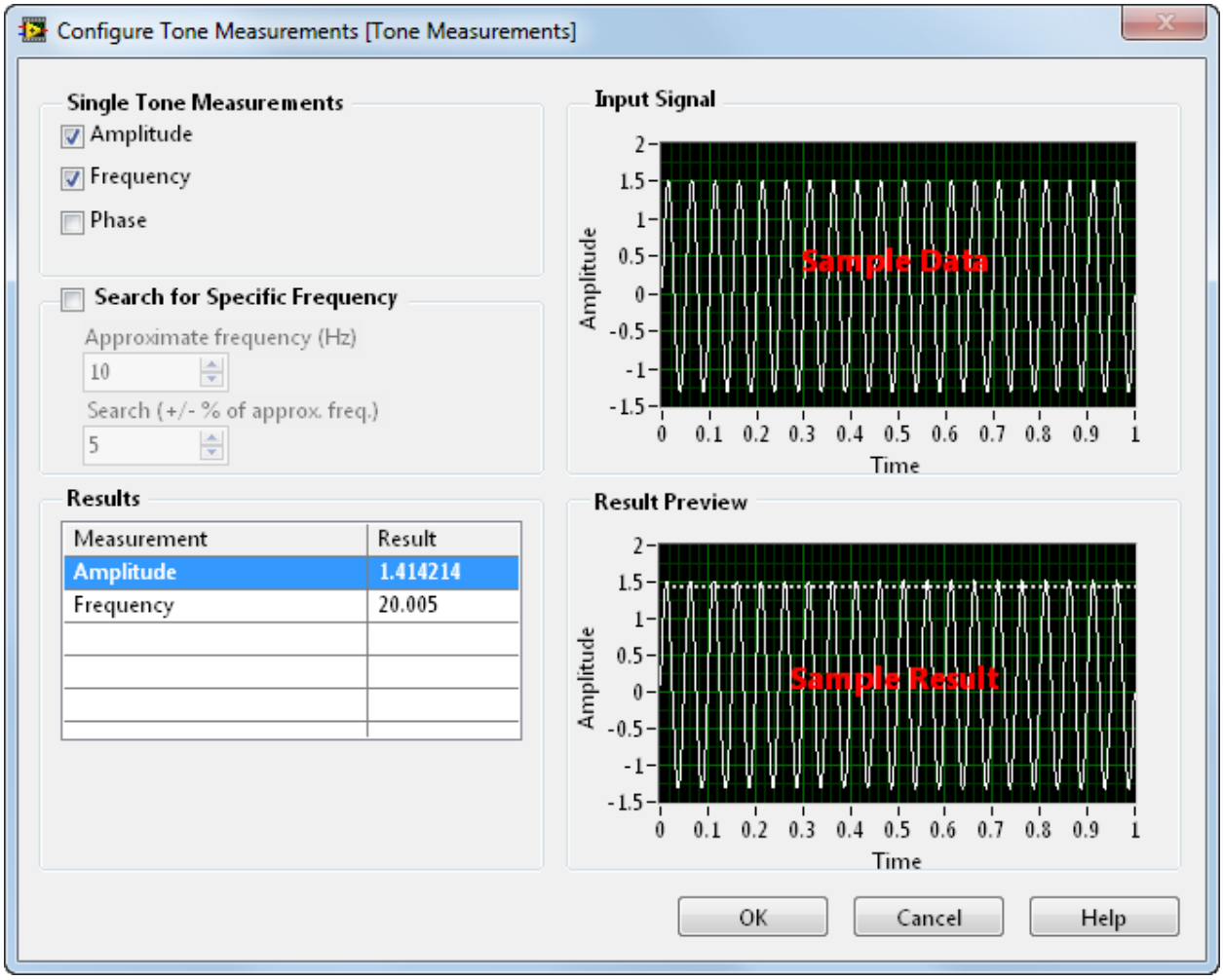
24. Move the Cutoff Frequency Icon into the For Loop and wire it to the **Lower Cut-Off** input terminal of the **Filter Express VI**.
25. Wire the **Sine** output terminal of the **Simulate Signal Express VI** to the **Signal** input terminal of the **Filter Signal Express VI**.



26. Add a **Tone Measurements Express VI** to the For Loop to find the Frequency and Amplitude of the filtered signal.
    a. Right-click the block diagram.
    b. Navigate to **Express»Signal Analysis»Tone Measurements**.
    c. Drag and drop a Tone Measurements Express VI into the For Loop.



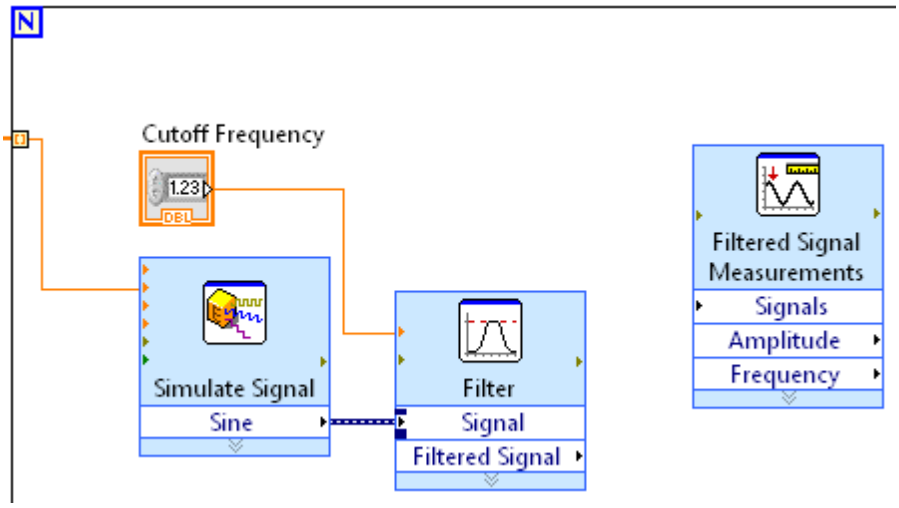
    d. Select **Frequency** and **Amplitude** in the **Configure Tone Measurements** window.



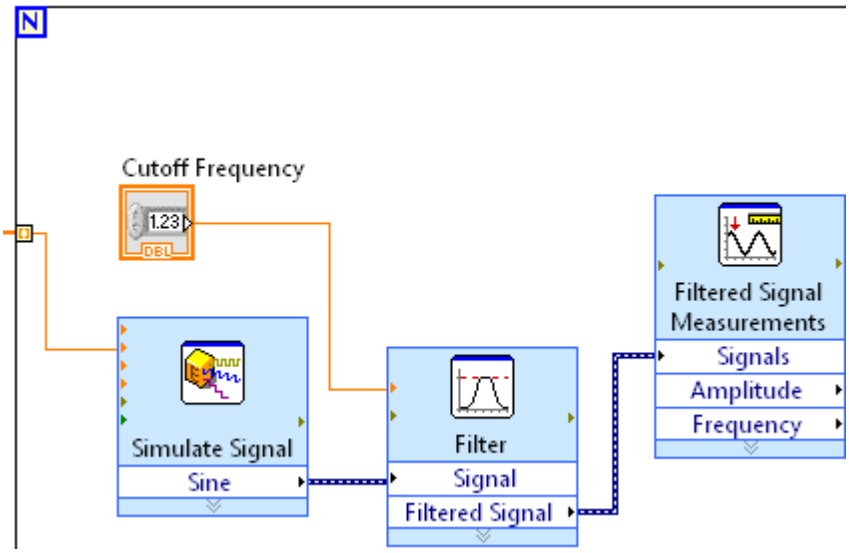
    e. Select **OK**.

f. Name the Tone Measurements Express VI **Filtered Signal Measurements**.

Tip: To rename Express VIs you can double-click the Express VI text.

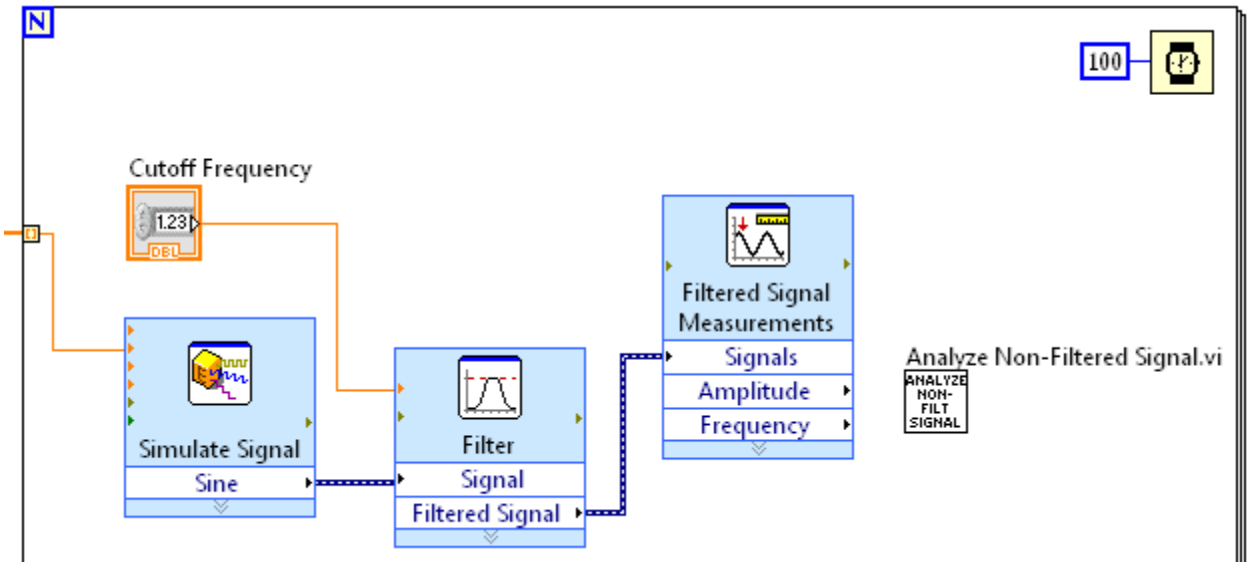


27. Wire the Filtered Signal output terminal of the Filter Express VI to the Signals input terminal of the Filtered Signal Measurements (Tone Measurement) Express VI.

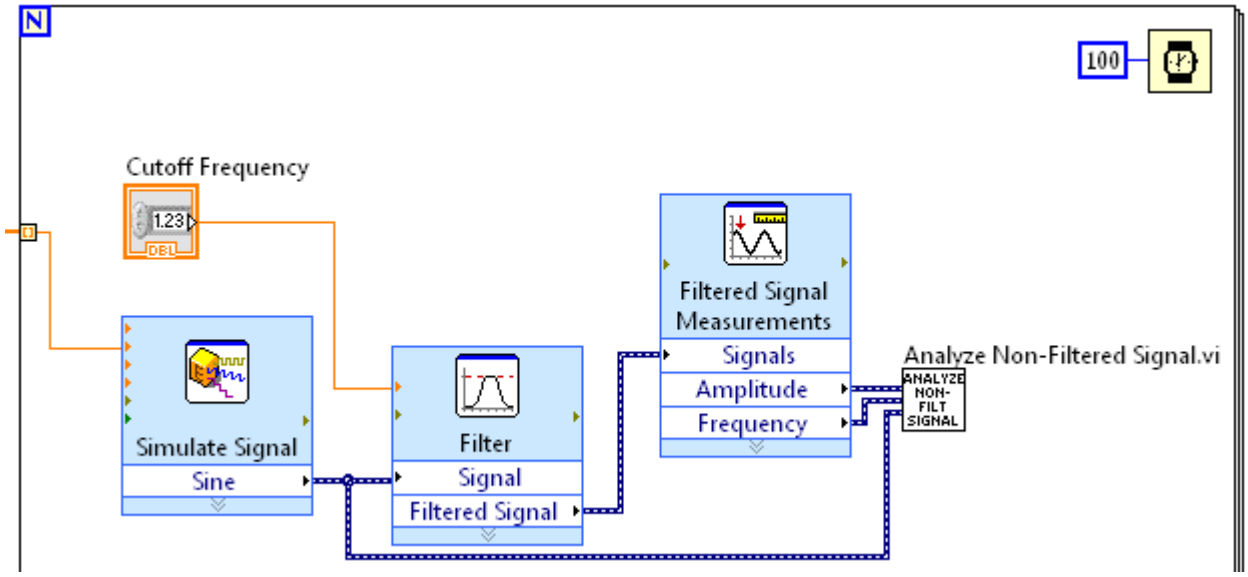


28. Add a Analyze Non-Filtered Measurements SubVI to the For Loop.
    a. Right-click the block diagram.
    b. Navigate to **Select a VI**.
    c. Browse to `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Shared Code` and select **Analyze Non-Filtered Measurements**.
    d. Select **OK**.

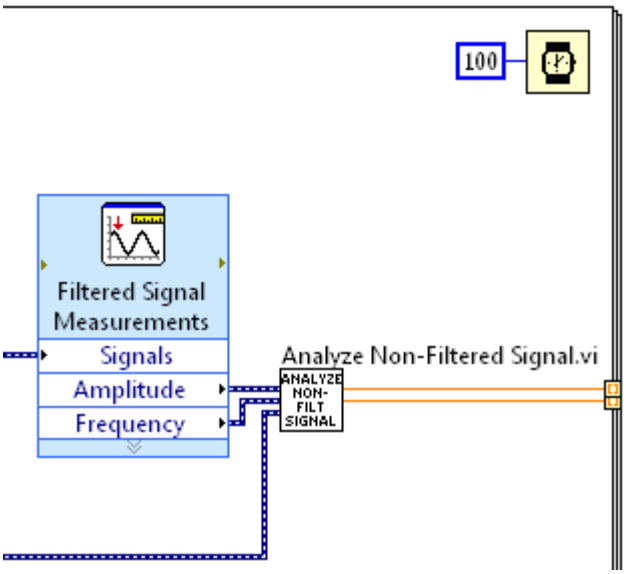
e. Drop the Analyze Non-Filtered Measurements SubVI into the For Loop.



29. Wire the following inputs into the Analyze Non-Filtered Signal SubVI.
    a. **Amplitude** output terminal of the **Filtered Signal Measurements (Tone Measurement) Express VI** to **Amplitude** input terminal of the **Analyze Non-Filtered Signal** SubVI.
    b. **Frequency** output terminal of the **Filtered Signal Measurements (Tone Measurement) Express VI** to **Frequency** input terminal of the **Analyze Non-Filtered Signal** SubVI.
    c. **Sine** output terminal of the **Simulate Signal Express VI** to **sine** input terminal of the **Analyze Non-Filtered Signal** SubVI.



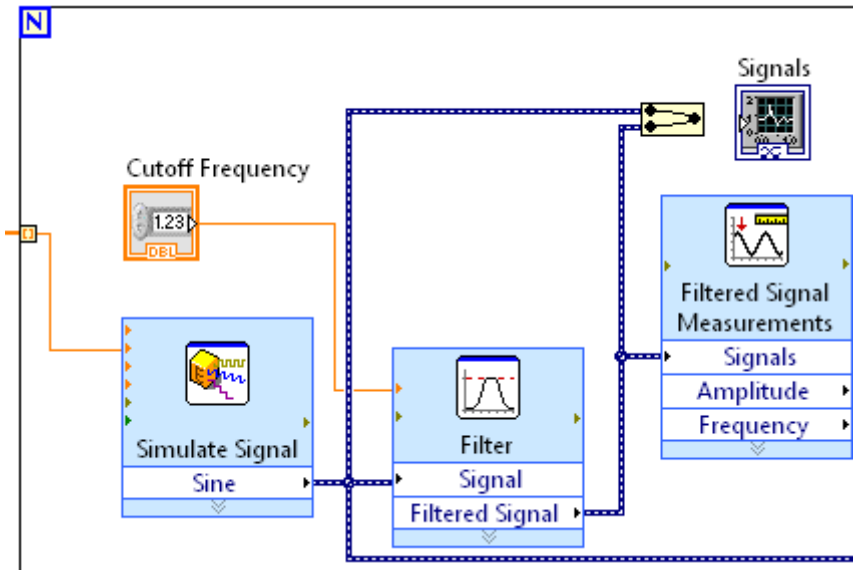
30. Wire the **Frequency** out and the **Amplitude** out from the **Analyze Non-Filtered Signal** SubVI to the right-side of the For Loop.



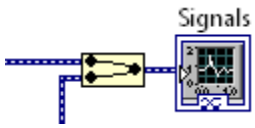
31. Add a Merge Signals function to the For Loop.
    a. Use <CTRL-Space> to bring up the Functions Quick Drop window.
    b. Type `Merge Signals` into the Quick Drop Search window.
    c. Double-click **Merge Signals**.
    d. Drop the Merge Signals function into the For Loop.

32. Wire the **Sine** output terminal from the **Simulate Signal Express VI** to the **top** input terminal of the **Merge Signals** function.

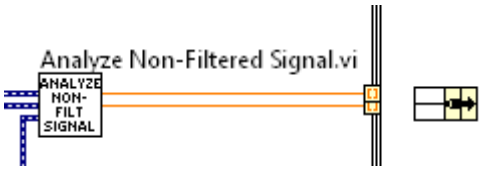
33. Wire the **Filtered Signal** output terminal from the **Filter Express VI** to the **bottom** input terminal of the **Merge Signals** function.



34. Wire the **Merge Signals** output terminal to the **Signals** Waveform Graph.



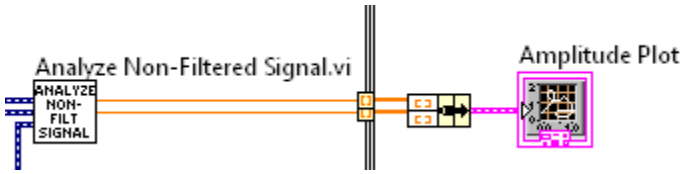
35. Add a Bundle function to the right of the For Loop on the block diagram.
    a. Right-click on the block diagram.
    b. Navigate to **Programming»Cluster, Class & variant»Bundle**.
    c. Drag and drop the Bundle function onto the block diagram.



36. Wire the **Frequency Array** and **Amplitude Array** from the side of the For Loop to the **Bundle** function.

    Note: The Frequencies and Amplitude elements of each iteration of the For Loop are placed into an array.

37. Wire the **output cluster** of the **Bundle** function to the **Amplitude Plot** XY Graph.



38. Create a Cursor List Property Node for the Amplitude Plot.
    a. Right-click on the Amplitude Plot Icon.
    b. Navigate to **Create»Property Node»Cursor List**.
    c. Drag and Drop the Cursor List Property Node onto the block diagram.

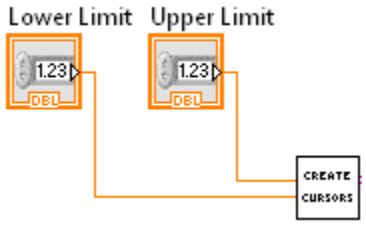
Note: A property node gets (reads) and/or sets (writes) properties of a reference. In this example, the property node will be used to set the Amplitude Plot Cursor List.

39. Change the Cursor List Property Node for the Amplitude Plot to **Write**.
    a. Right-click on the **Cursor List Property Node**.
    b. Select **Change All to Write**.

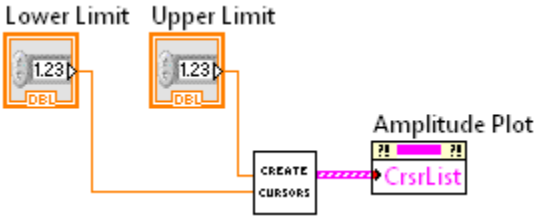
40. To the right of the For Loop, add a Create Cursors SubVI to the block diagram.
    a. Right-click the block diagram.
    b. Navigate to Select a VI.
    c. Browse to `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Shared Code` and select **Create Cursors**.
    d. Select **OK**.
    e. Drop the Create Cursors Sub VI onto the block diagram.

Note: the Create Cursors SubVI creates cursors for the Amplitude plot based on the upper and lower limits you specify.

41. Wire the **Upper Limit** and **Lower Limit** Controls to the **Upper Limit** and **Lower Limit** input terminals of the **Create Cursors** SubVI.



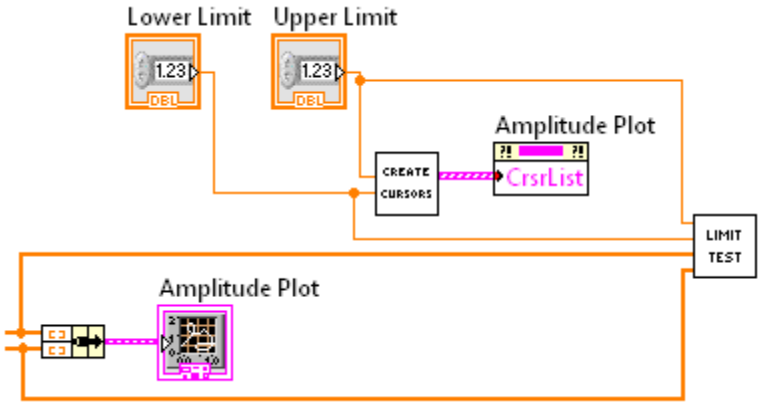
42. Wire the **Plot Cursors** output terminal of the **Create Cursors** SubVI to the **Amplitude Plot Cursor List Property Node**.



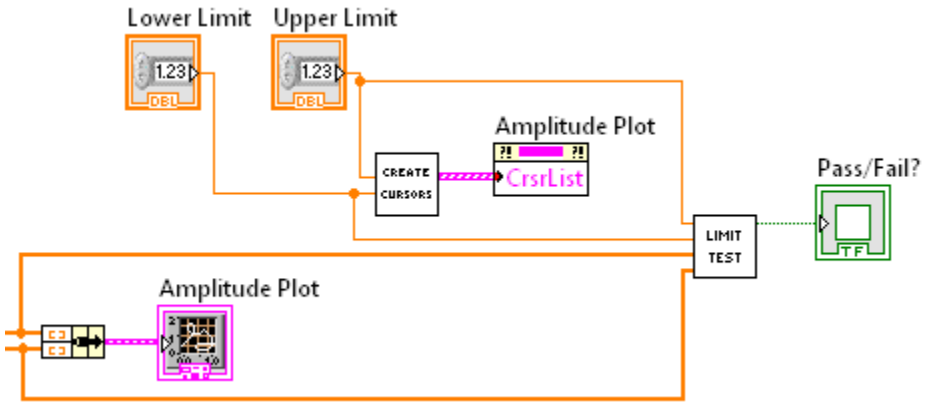
43. Add a Limit Test SubVI to the block diagram.
    a. Right-click the block diagram.
    b. Navigate to Select a VI.
    c. Browse to `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Shared Code` and select **Limit Test**.
    d. Select **OK**.
    e. Drop the Limit Test Sub VI onto the block diagram.

Note: The Limit Test SubVI determines the frequency at which the filtered signal reaches the -3 dB point. The frequency of this signal is used to perform a limit test in which it will be compared to the lower and upper limits you specify. A pass or fail value will be returned based on the results of the limit test.

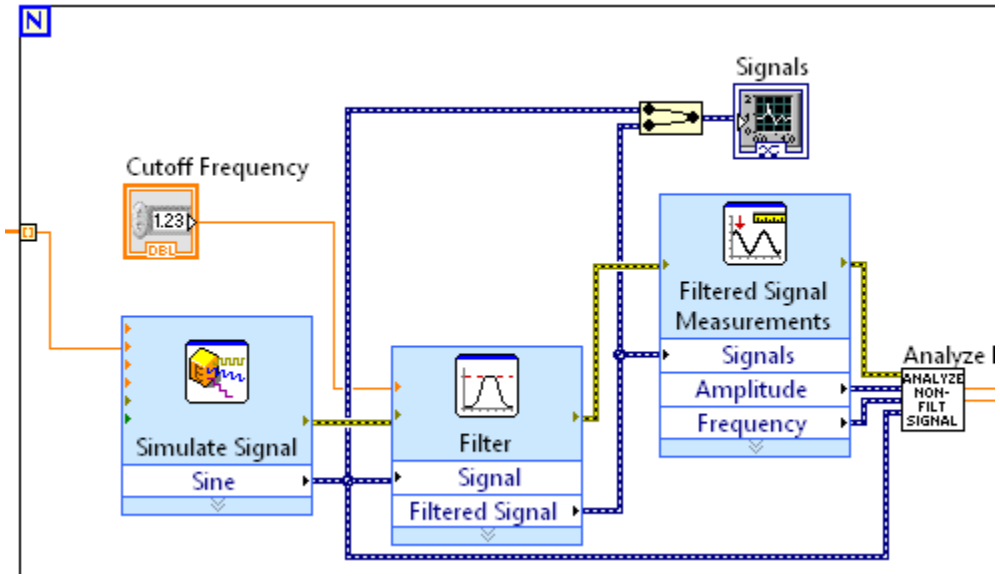
44. Wire the following data to the Limit Test SubVI.
    a. Wire the **Upper Limit** and **Lower Limit** Controls to the **Upper Limit** and **Lower Limit** input terminals of the **Limit Test** SubVI.
    b. Wire the **Frequency** and **Amplitude** Arrays to the **Frequency** and **Amplitude** input terminals of the **Limit Test Sub** VI.



45. Wire the **Pass?** output terminal of the **Limit Test** SubVI to the **Pass/Fail?** LED.

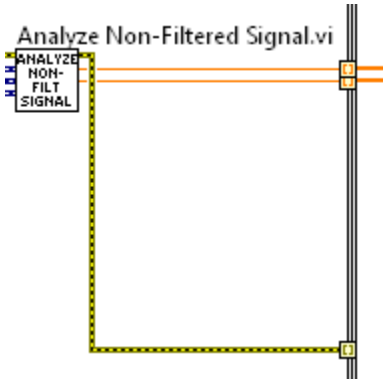


46. Add Error Handling to the application by wiring the Error Cluster through all of the block diagram functions in the For Loop.



Note: The Error Cluster is the green wire and is passed between functions using the Error out and Error in tunnels.

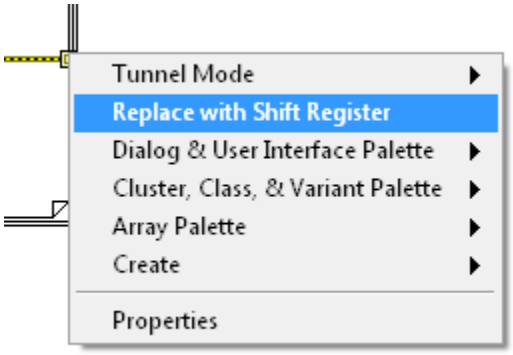
47. Wire the **Error Cluster** from the **Analyze Non-Filtered Signal** SubVI to the right side of the For Loop.



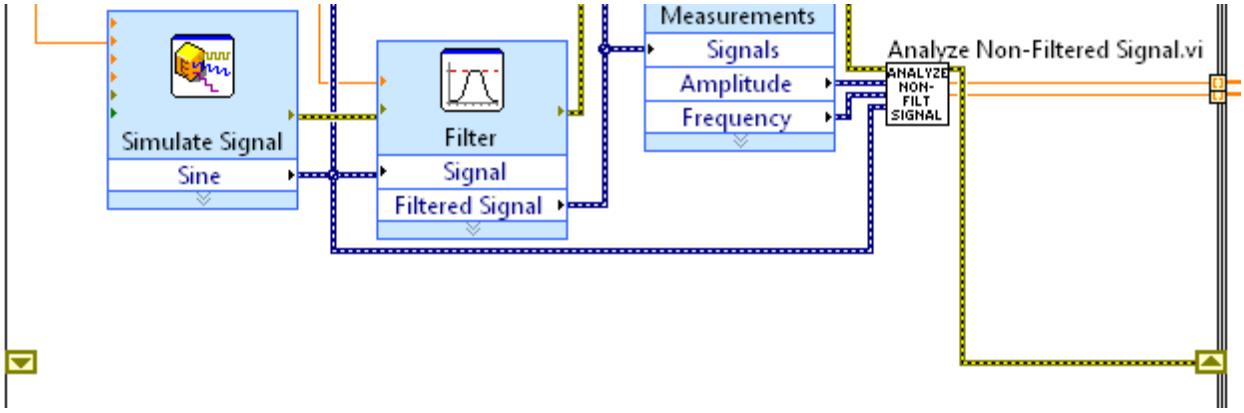
48. Change the Error Cluster to retain the value of the Error with each loop.

    a. Right-click on the **Error Cluster Loop tunnel**.

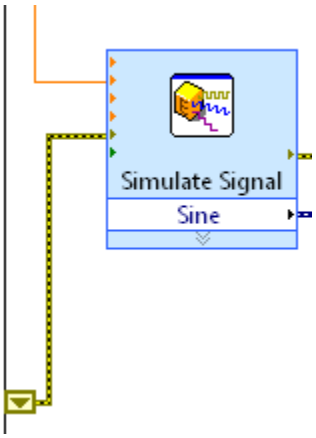
    b. Select **Replace with Shift Register**.



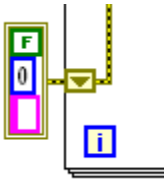
    c. Click on the left side of the For Loop to have a matching input Shift Register for the Error Cluster.



49. Wire the **Error Cluster** on the left side of the Fop Loop to the **Error In** input terminal of the **Simulate Signal Express VI**.

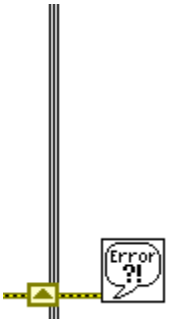


50. Initialize the Error Cluster.
    a. Right-click on the left shift register for the Error Cluster.
    b. Select **Create Constant**.

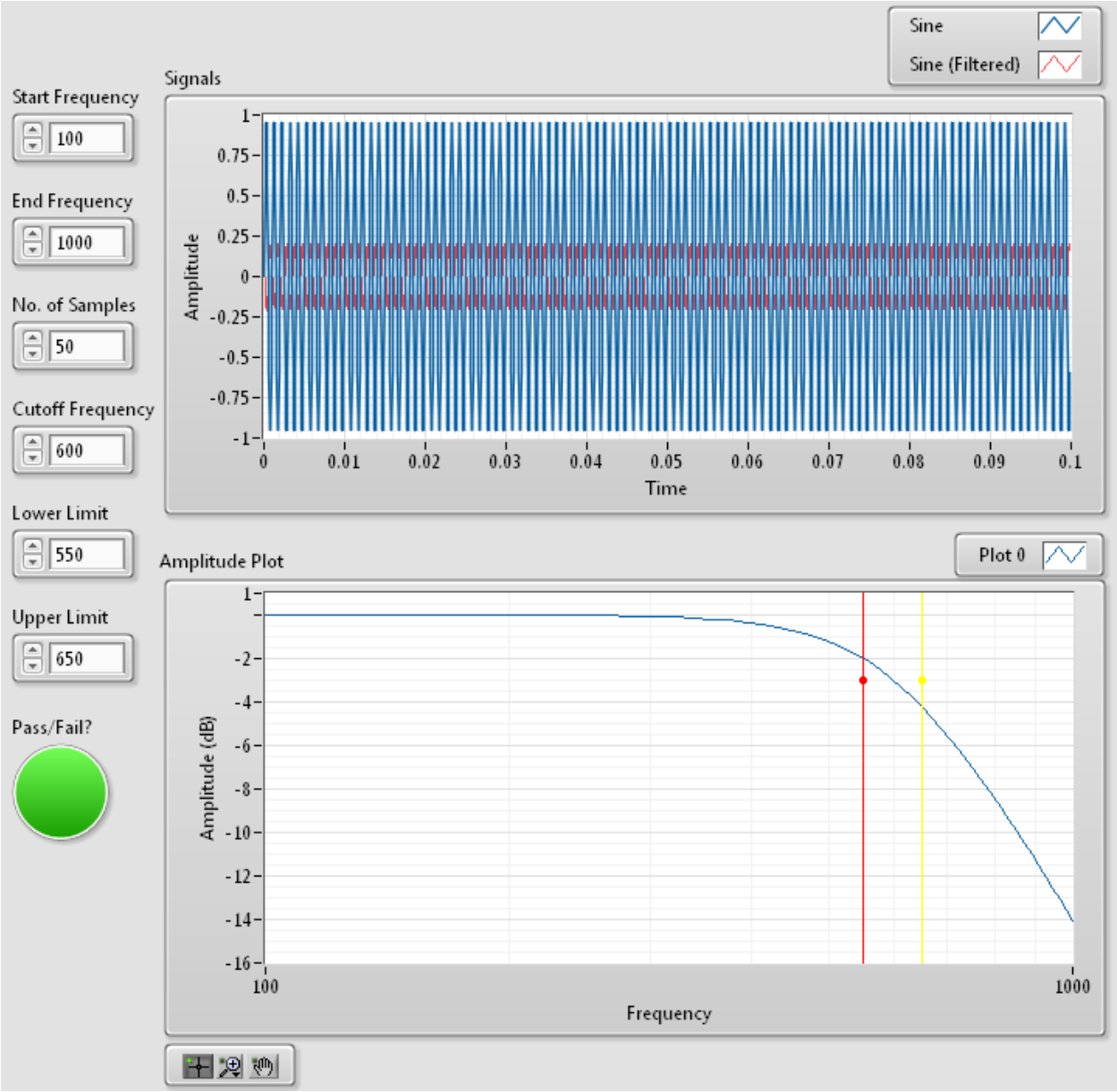


51. To the right of the For Loop, add a Simple Error Handler function to the block diagram.
    a. Right-click the block diagram.
    b. Navigate to **Programming» Dialog & User Interface»Simple Error Handler**.
    c. Drag and Drop the Simple Error Handler function onto the block diagram.

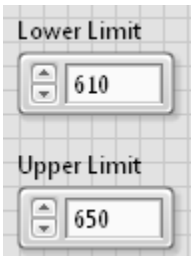
52. Wire the **Error Cluster** from the **For Loop** to the **error in** input terminal of the **Simple Error Handler**.



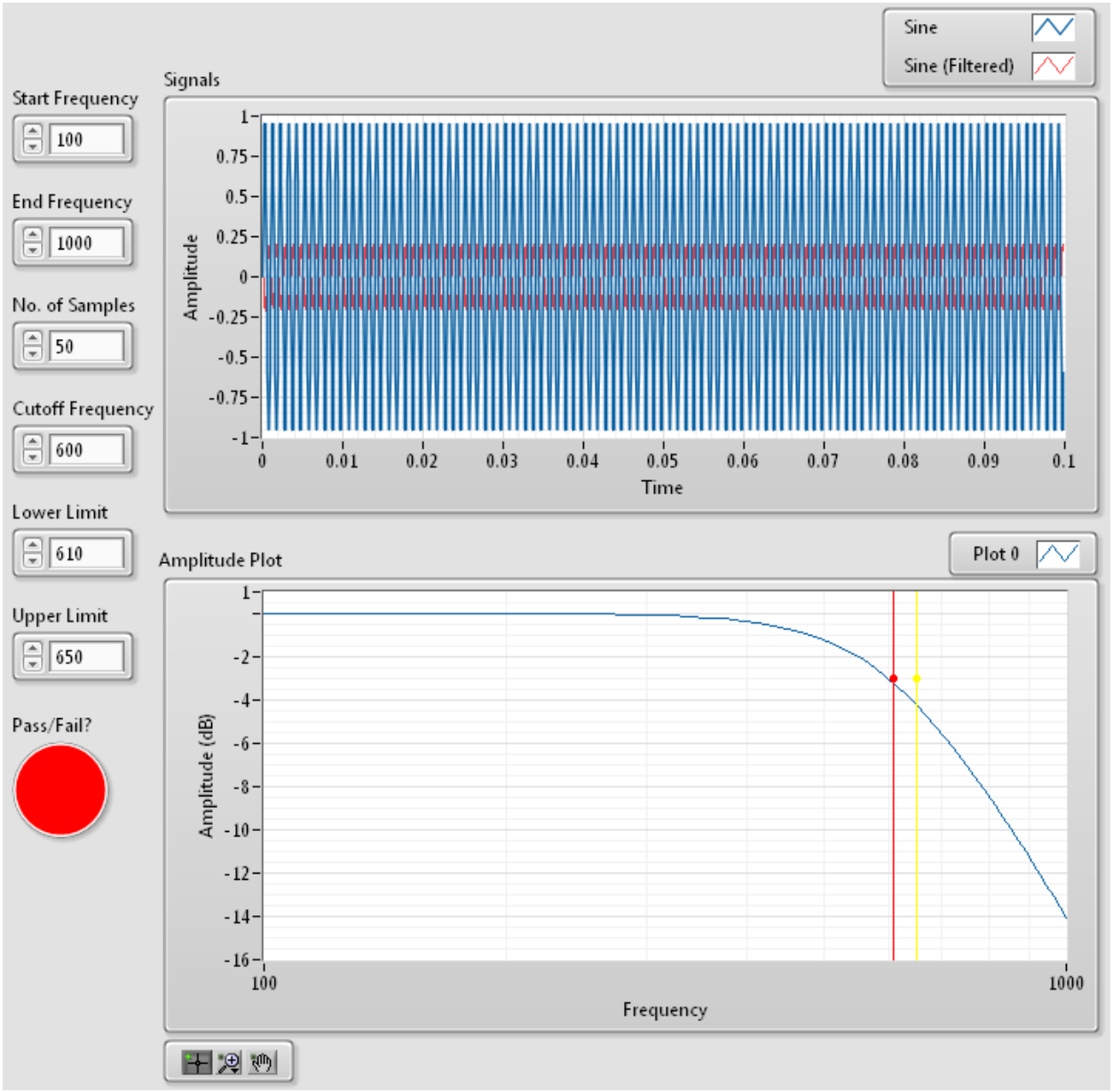
53. Run the Simulated Filtered Test application and observe the Frequency Sweep and Amplitude values.



54. Change the Lower Limit to **610** and leave the Upper Limit at **650**.

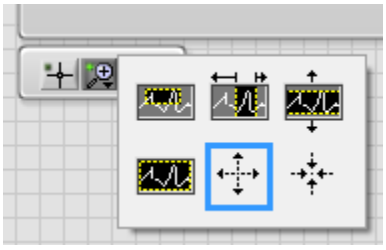


55. **Run** the application once more and observe that the simulated filter test fails.

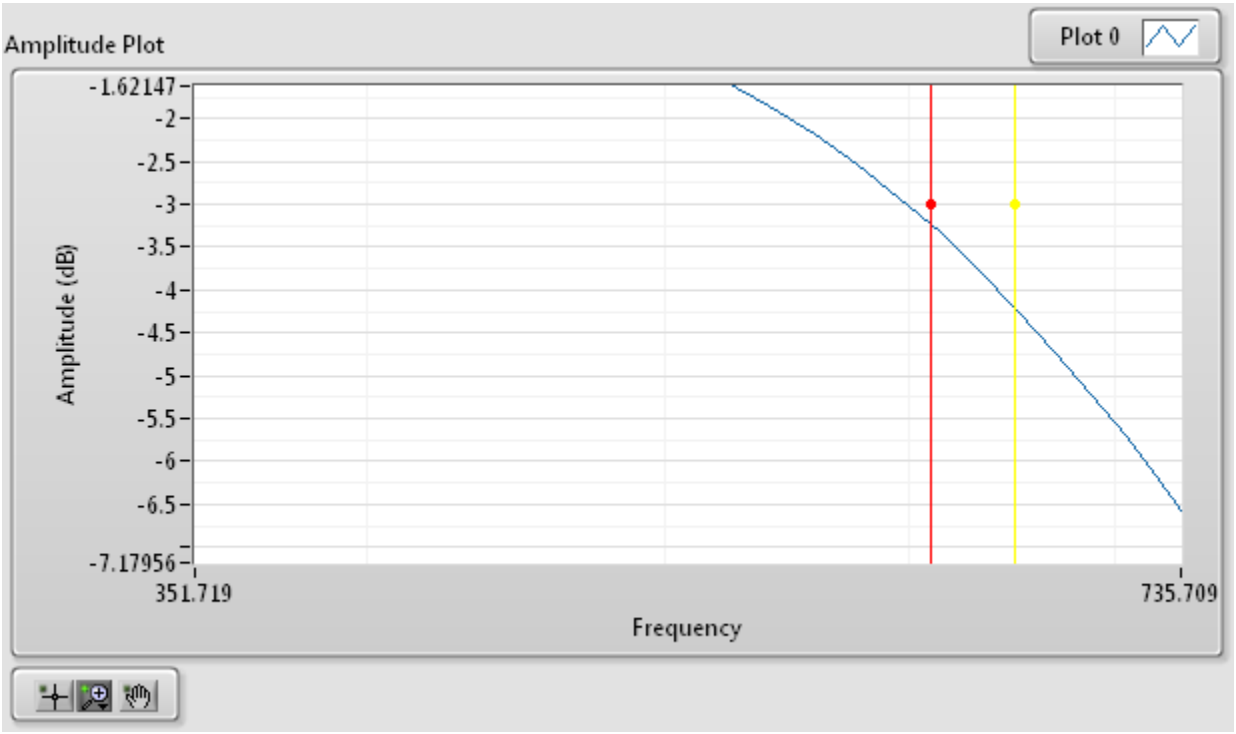


56. Zoom in on the Amplitude Plot to observe the point at which the signal failed.

    a. Select **Zoom In** on the Graph Palette.



    b. Click on the **Amplitude Plot** near the Red and Yellow Cursors to zoom in.



57. Save the Simulated Filtered Test application and minimize it.

EXERCISE 4.1 - GENERATE AND MEASURE A SIGNAL USING THE FGEN AND SCOPE SOFT FRONT PANELS

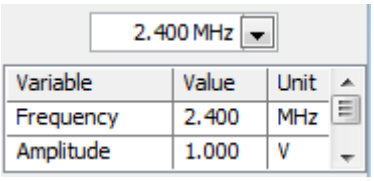
Goal

The goal of this exercise is to generate a signal on the PXI Arbitrary Waveform Generator using the FGEN Soft Front Panel. You will then measure and observe the supplied signal with the PXI Digitizer using the SCOPE Soft Front Panel.

Steps

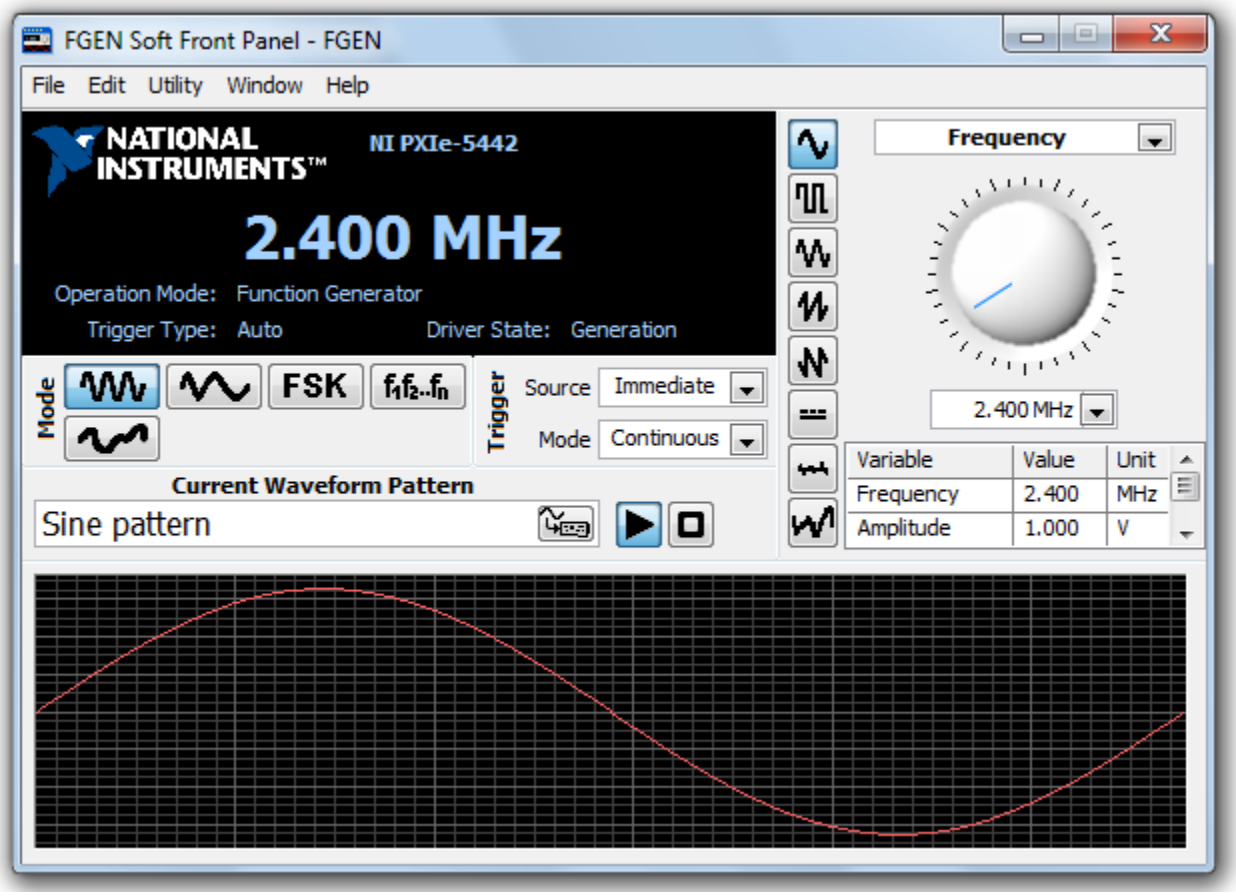
1. Open the FGEN Soft Front Panel from **Start»All Programs»National Instruments» NI-FGEN»FGEN Soft Front Panel**.
2. Select **sine waveform pattern**.

3. Change the Frequency of the sine wave to **2.400 MHz**.

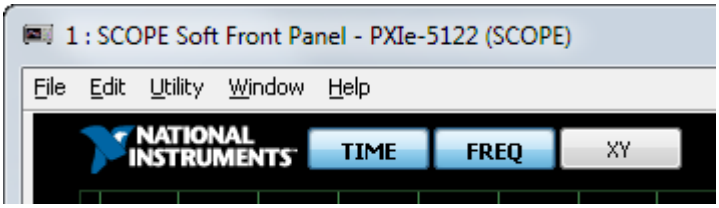


4. Select **Start** waveform generation.

5. You should see the following on the FGEN Soft Front Panel.

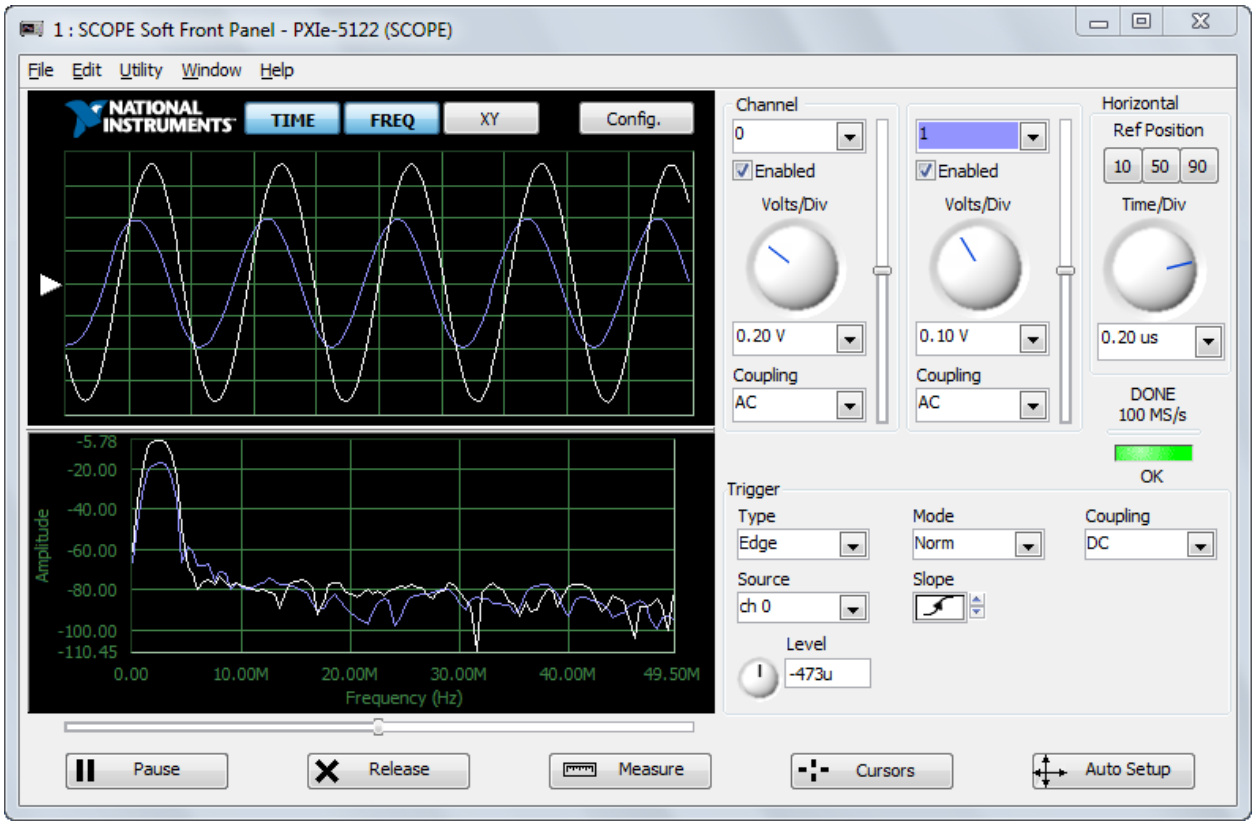


6. Open the SCOPE Soft Front Panel from **Start»All Programs»National Instruments» NI-SCOPE»SCOPE Soft Front Panel**.
7. Make sure that **Time** and **Frequency** are selected.

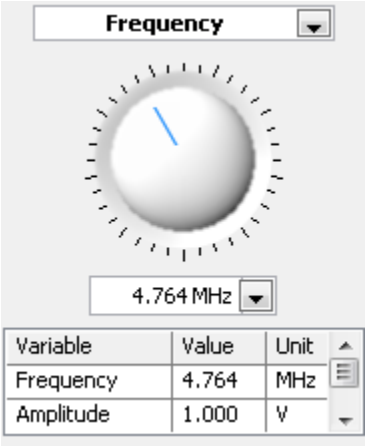


8. Select **Auto Setup**.

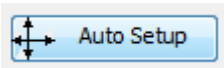
9. Notice the signals being acquired from channel 0 and channel 1. Channel 0 is the raw signal and Channel 1 is the filtered signal.



10. Return to the FGEN Soft Front Panel, and adjust the Frequency of the signal being generated.

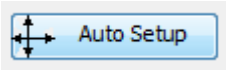


11. Notice that the signal being generated and acquired changes.
12. Select **Auto Setup** on the SCOPE Soft Front Panel.



13. On the FGEN Soft Front Panel, select **square waveform pattern**.

14. Notice that a square wave is now being generated and acquired.
15. Select **Auto Setup** on the SCOPE Soft Front Panel.



16. Spend some time adjusting the values of the FGEN Soft Front Panel and the SCOPE Soft Front Panel.
17. Close the SCOPE Soft Front Panel and the FGEN Soft Front Panel.

EXERCISE 4.2 – LOWPASS FILTER TEST

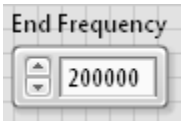
Goal

The goal of this exercise is to create a lowpass filter test. In the application, you will generate a frequency sweep of signals and supply each to a simulated lowpass filter. The amplitude of each filtered signal will be measured in order to determine when the signal has reached the -3dB point. The frequency of this signal will then be used to perform a limit test in which it will be compared to the lower and upper limits you specify. A pass or fail value will be returned based on the results of the limit test.

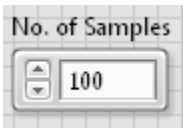
Steps

1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browse to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the LabVIEW for Instrumentation Hands-On.lvprj file.
2. Open the Simulated Filter Test you previously created in Exercise 3.
   Note: You can also open the Simulated Filter Test from the Solutions folder, if you did not complete the previous exercise.
3. Create a copy of this exercise in the Filter Test Folder.
   a. Select **File»Save As**.
   b. Select **Create unopened disk copy**.
   c. Click **Continue**.
   d. Browse to `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Filter Test`.
   e. Name the LabVIEW VI **Filter Test**.
   f. Click **OK**.
   g. Add the New VI to the Filter Test Folder in the Project Explorer.
      i. Right-click the Filter Test folder in the Project Explorer.
      ii. Select **Add»File**.
      iii. Browse to Browse to the Filter Test Folder `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Filter Test`.
      iv. Select Filter Test.vi.
      v. Click **Add File**.
4. Modify the front panel as follows.
   a. Change the Start Frequency to **1,000**.

b. Change the End Frequency to **200,000**.

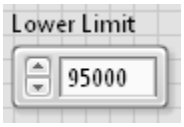


c. Change the No. of Samples to **100**.

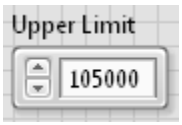


d. Delete the Cutoff Frequency Control.

e. Change the Lower Limit to **95,000**.

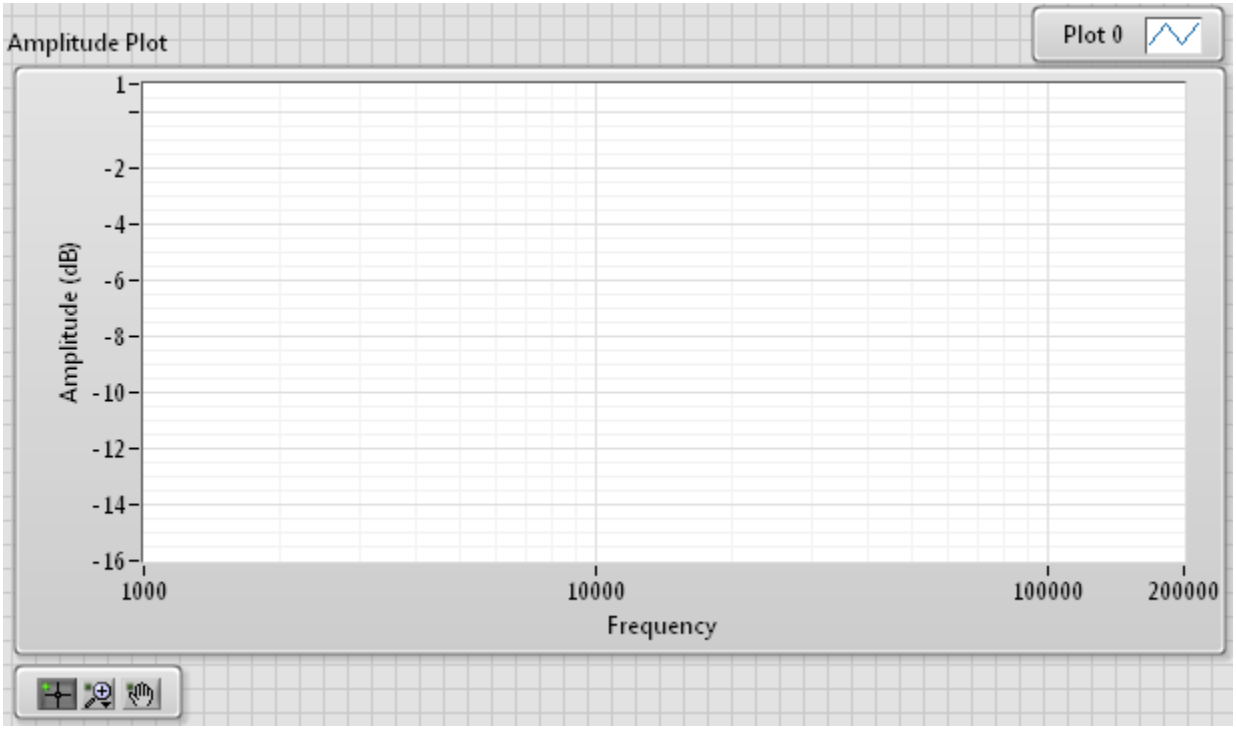


f. Change the Upper Limit to **105,000**.



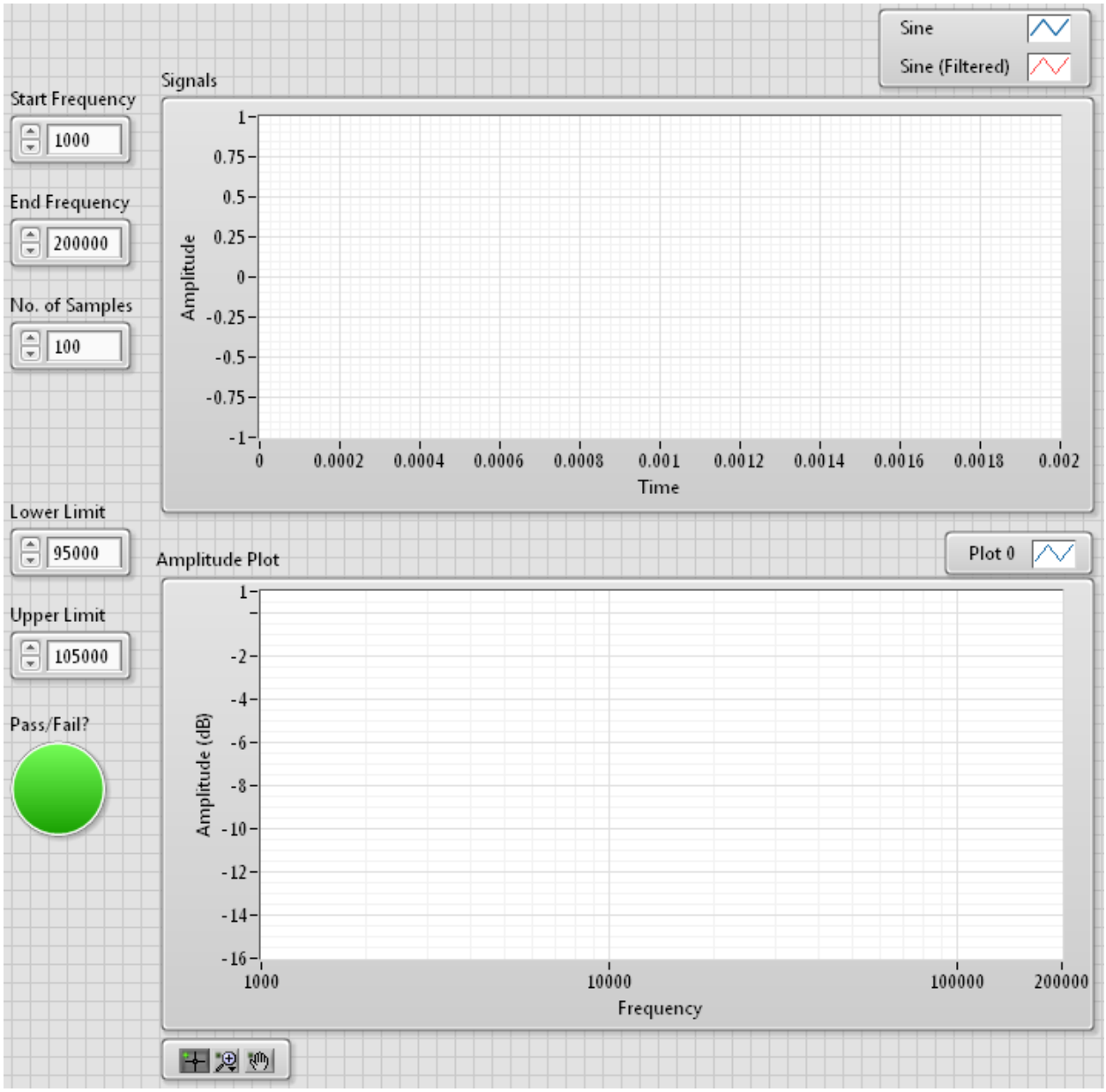
g. Change the X-Axis scale of the Amplitude Plot to **[1,000, 200,000]**.

h. Change the Y-Axis scale of the Amplitude Plot to **[-16, 1]**.

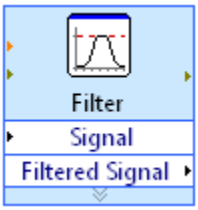


5. Make the current values default by selecting **Edit»Make Current Values Default**.

6. Your front panel should look similar to the following user interface.

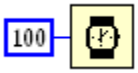


7. Toggle to the block diagram using <Ctrl-E>.
8. On the block diagram, delete the following functions.
    a. Delete the Filter Express VI.

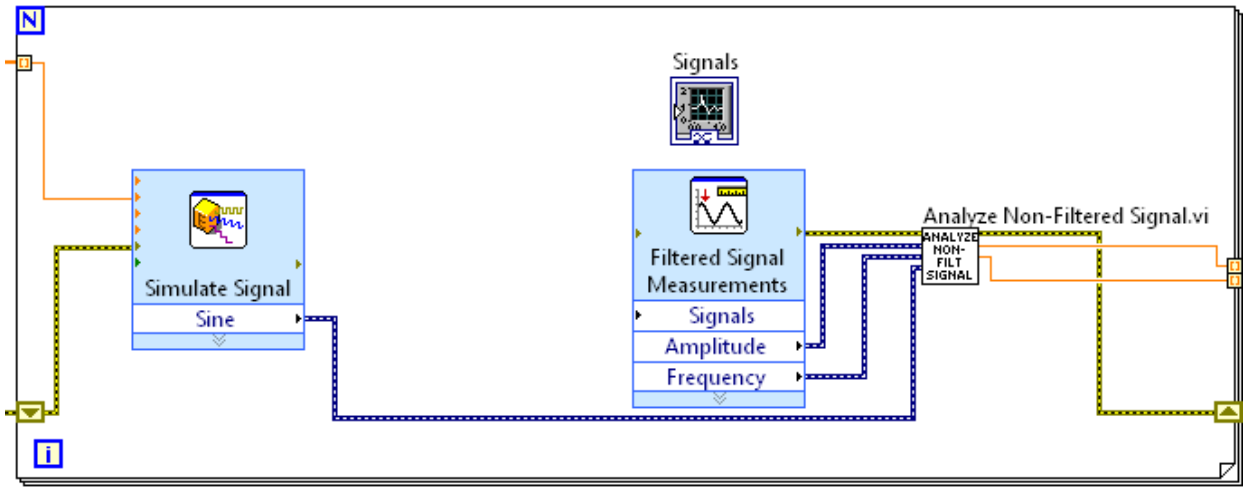


b. Delete the **Merge Signals** function.

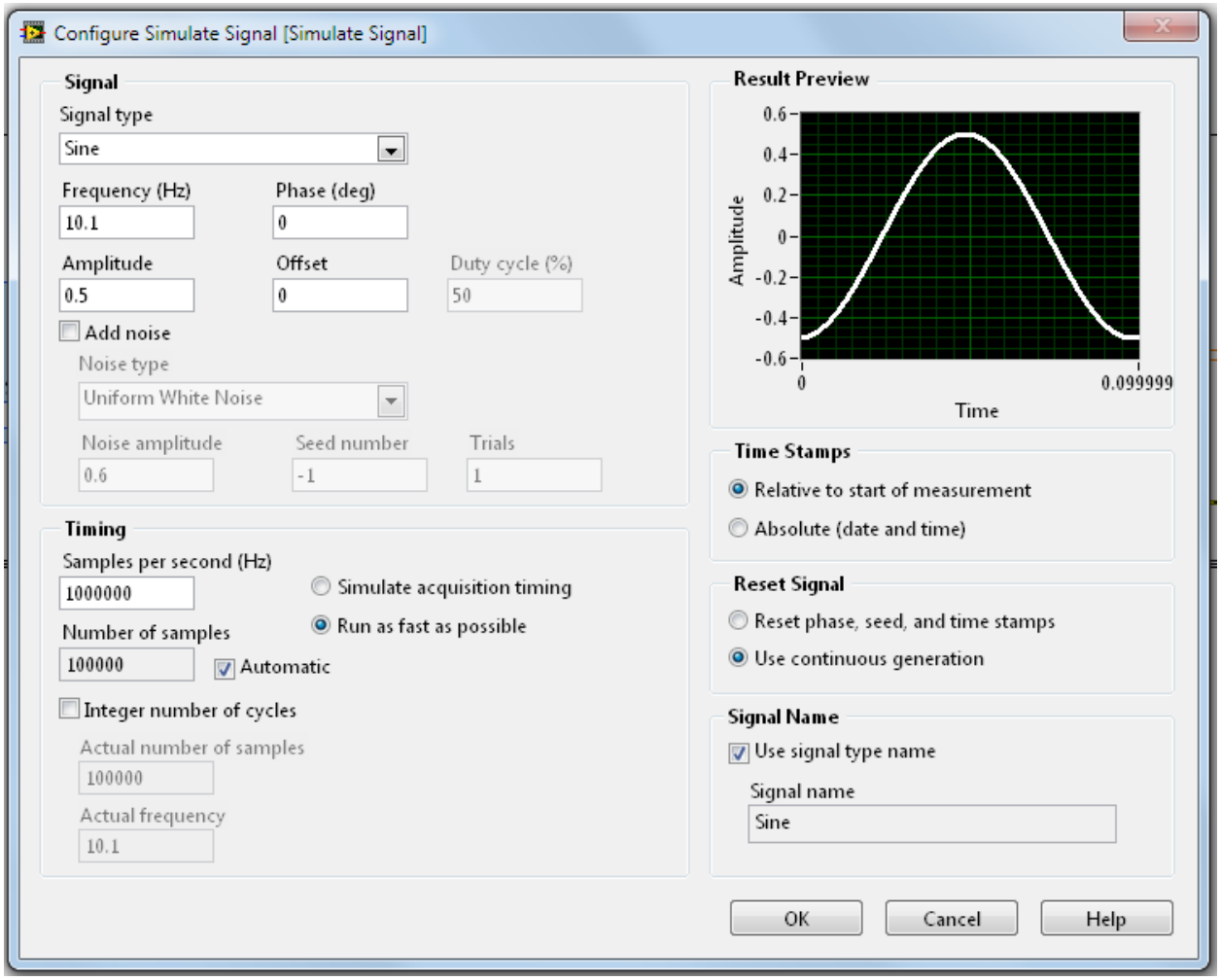
c. Delete the **Wait** function and **timing** constant.



9. Remove the broken wires using <Ctrl-B>.



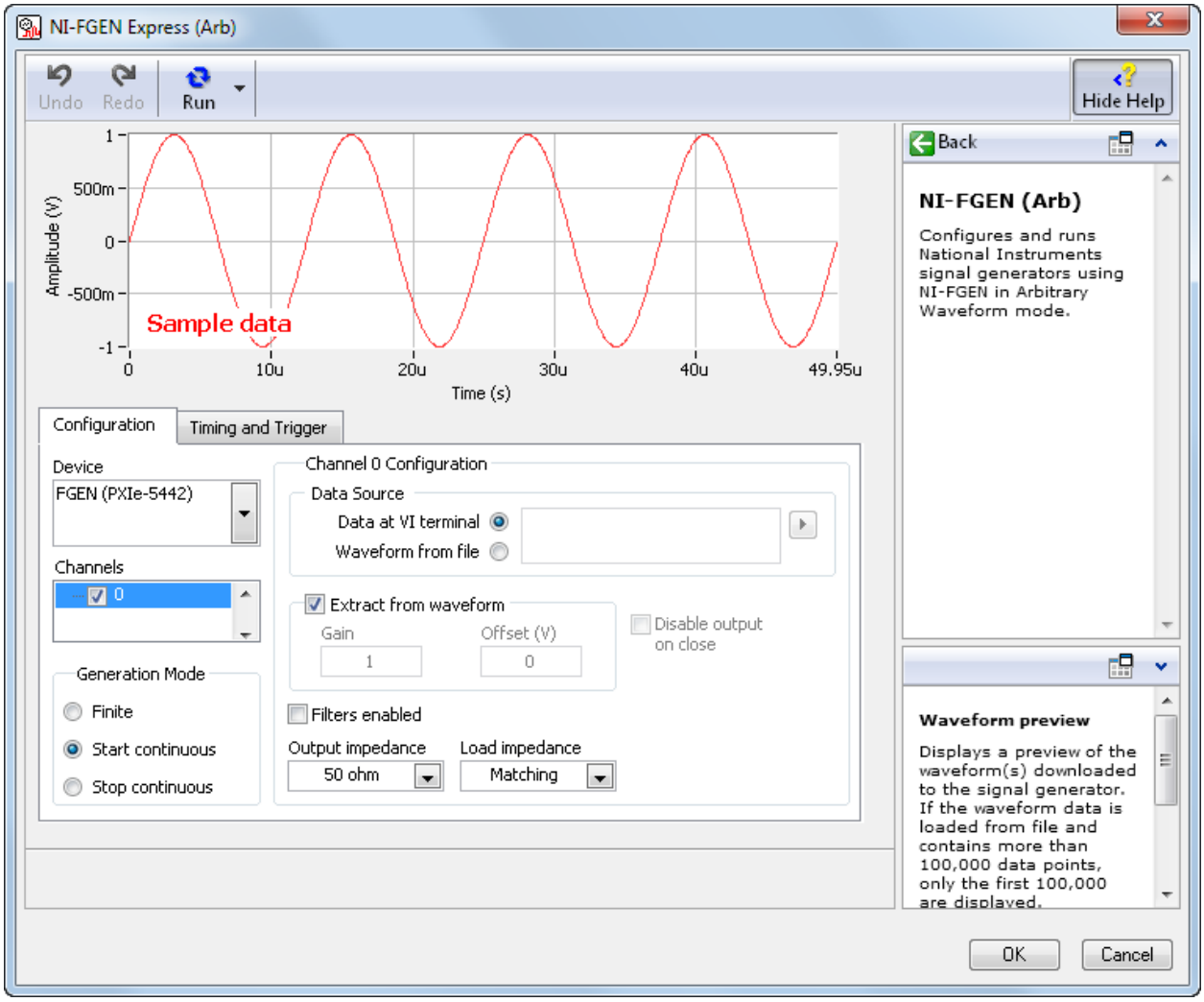
10. Modify the Timing of the Simulate Signal Express VI.
    a. Double-click the **Simulate Signal Express VI**.
    b. Change the Samples per second (Hz) to **1,000,000.**
    c. Change the Amplitude to **0.5.**



    d. Select **OK**.

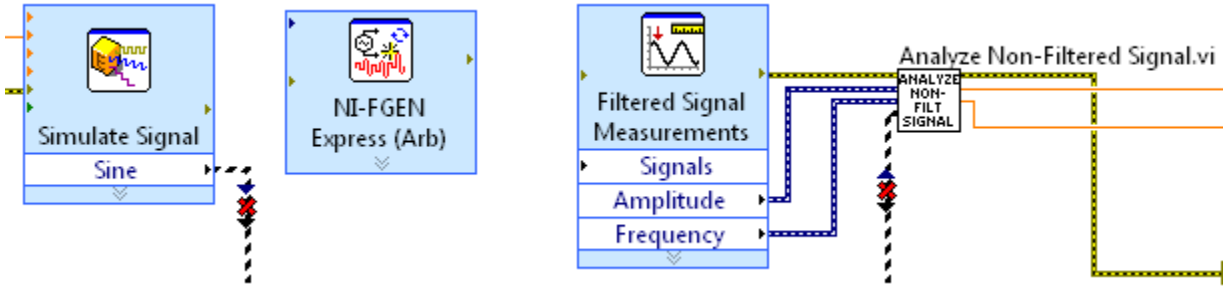
11. Add an NI-FGEN Express (arb) VI to the For Loop to start generation.
    a. Right-click the block diagram.
    b. Navigate to **Measurement I/O»NI-FGEN»NI-FGEN Express (Arb)**.
    c. Drag and drop the NI-FGEN Express (Arb) VI into the For Loop.

d. Leave the default settings in the configuration window.

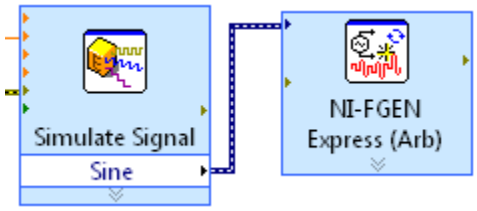


e. Select **OK**.

12. Delete the wire running from the Sine output terminal of the Simulate Signal Express VI to the Analyze Non-Filtered Signal SubVI.



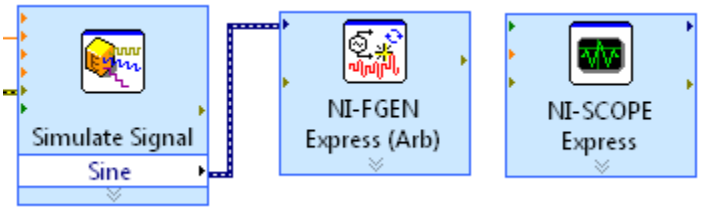
13. Wire the **Sine** output terminal of the **Simulate Signal Express VI** to the **data** input terminal of the **NI FGEN Express (Arb) VI**.



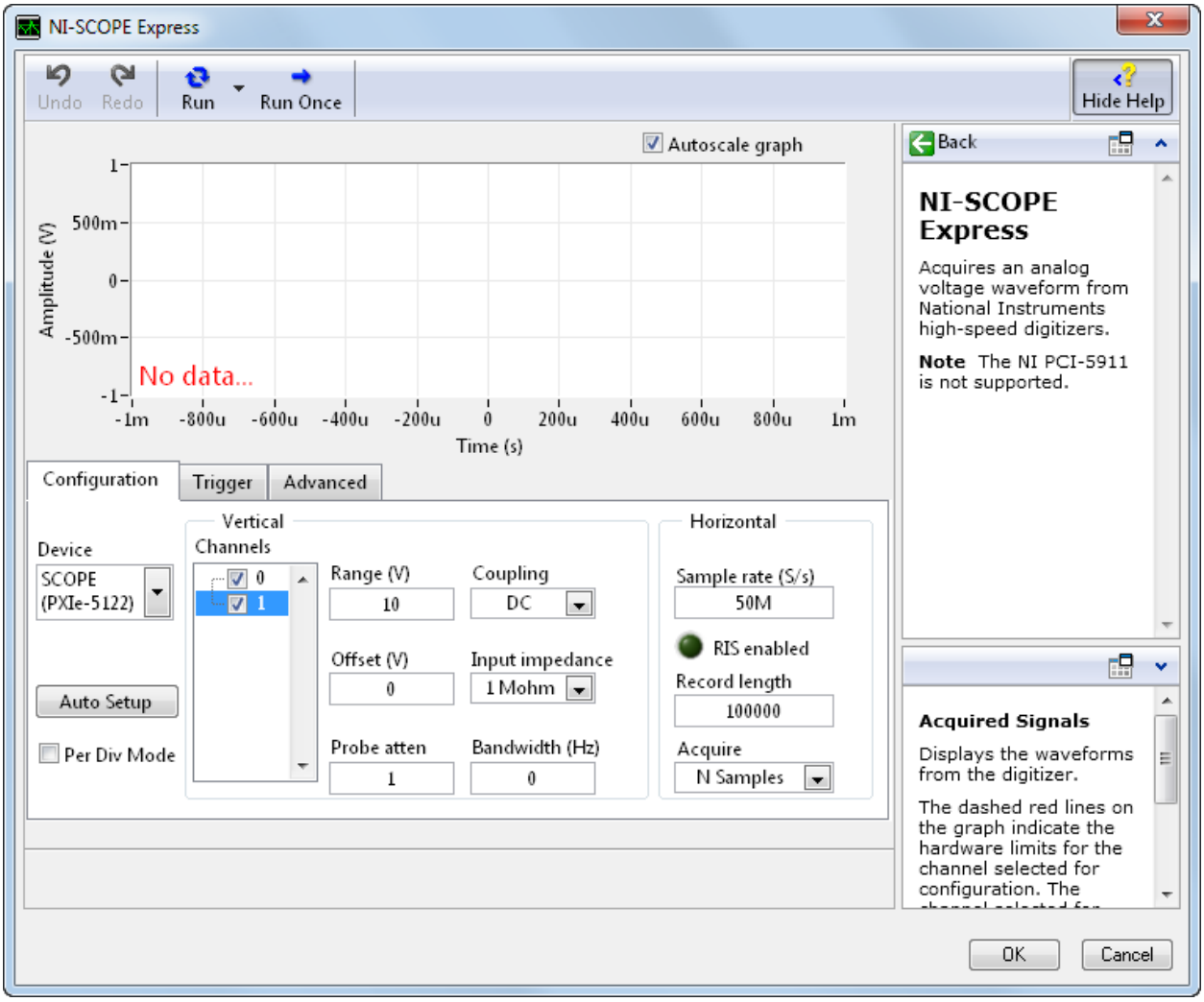
14. Create additional space in the For Loop to the right of the NI FGEN Express (Arb) VI.

    Tip: Hold the <Ctrl> key then click-and-drag the mouse to automatically move the icons on the block diagram.

15. Add an NI-SCOPE Express VI to the For Loop.
    a. Right-click the block diagram.
    b. Navigate to **Measurement I/O»NI-SCOPE»NI-SCOPE Express (Express Block)**.
    c. Drag and drop the NI-SCOPE Express VI into the For Loop to the right of the NI-FGEN Express (Arb) VI.

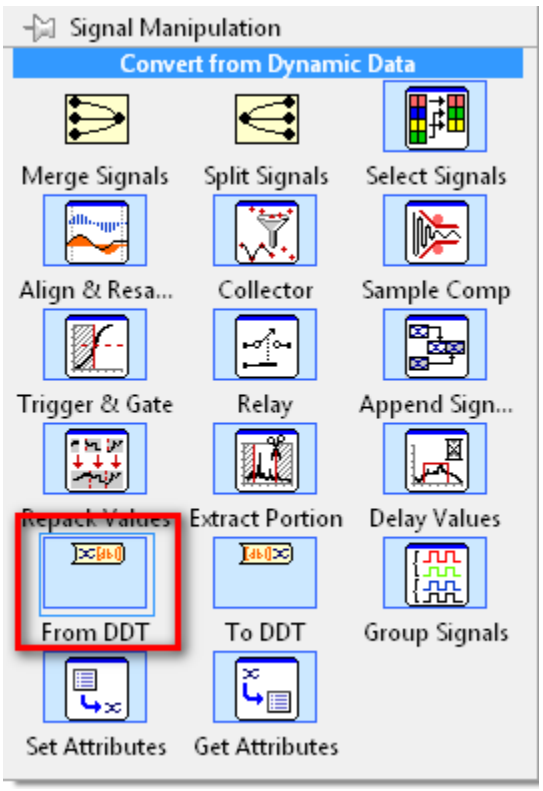


16. Modify the NI-SCOPE Express VI as follows in the configuration window.
    a. Select Channels **0** and **1**.
    b. Change the Range to **10**.
    c. Change the Sample Rate to **50M**.
    d. Change the Record Length to **100,000**.

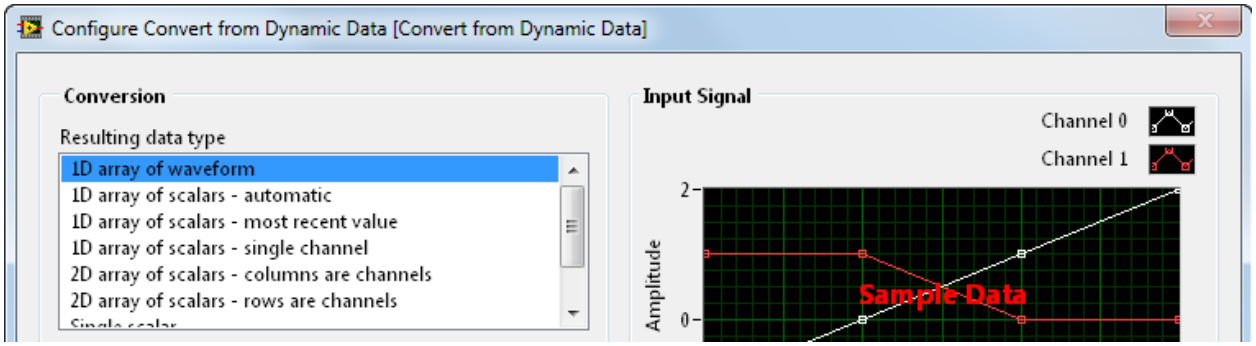


    e. Select **OK**.

17. Add a Convert from Dynamic Data function to the For Loop.
    a. Right-click the block diagram.
    b. Navigate to **Express»Signal Manipulation»Convert from Dynamic Data (From DDT)**.

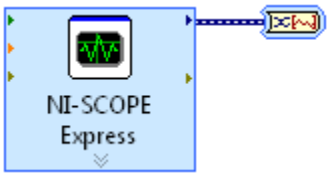


    c. Drag and drop the Convert from Dynamic Data function into the For Loop.
    d. Select **1D array of waveform**.

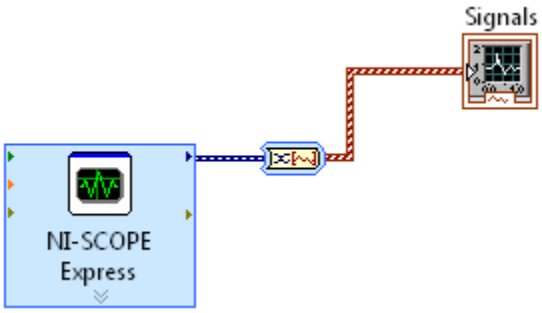


    e. Select **OK**.

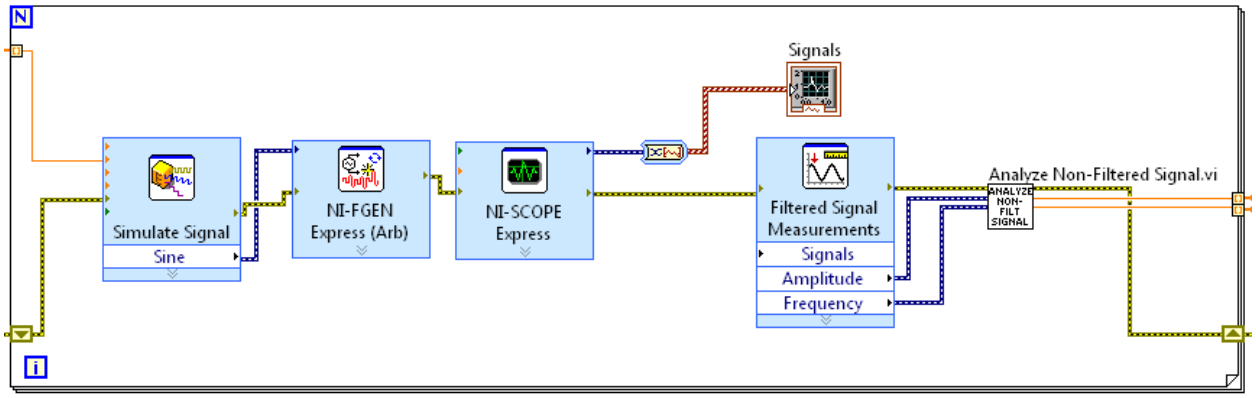
18. Wire the **Signal** output terminal of the **NI-SCOPE Express VI** to the **Convert from Dynamic Data** function.



19. Wire the **Array of Waveform** output terminal from the **Convert from Dynamic Data** to the **Signals** Waveform Graph.

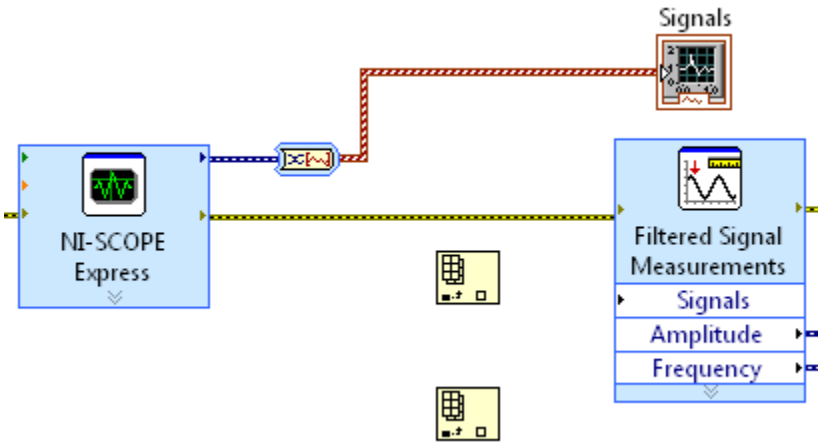


20. Wire the **Error Cluster** through the **block diagram functions** in the **For Loop**.

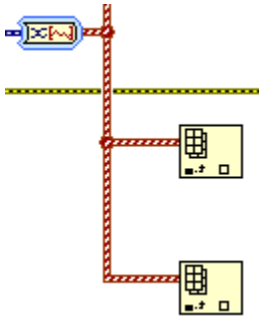


21. Add two Index Array functions to the For Loop.
    a. Right-click the block diagram.
    b. Navigate to **Programming»Array»Index Array**.
    c. Drag and drop the Index Array function into the For Loop.

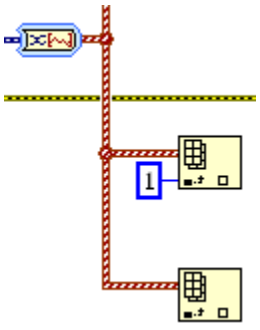
d. Create a copy of the Index Array function you just added to the For Loop by selecting the function and then holding <Ctrl> while dragging.



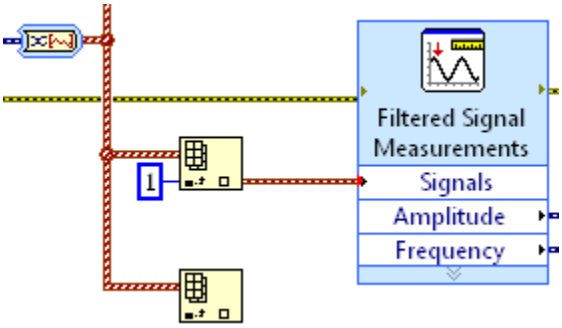
22. Wire the **Array of Waveform** output terminal from the **Convert from Dynamic Data** to the **array** input terminal on each of the **Index Array** functions.



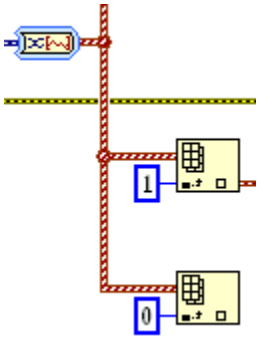
23. Add a constant to the first Index Array function to obtain the 1st element.
    a. Right-click on the index input terminal of the first Index Array function.
    b. Select **Create»Constant**.
    c. Change the value of the numeric constant to **1**.



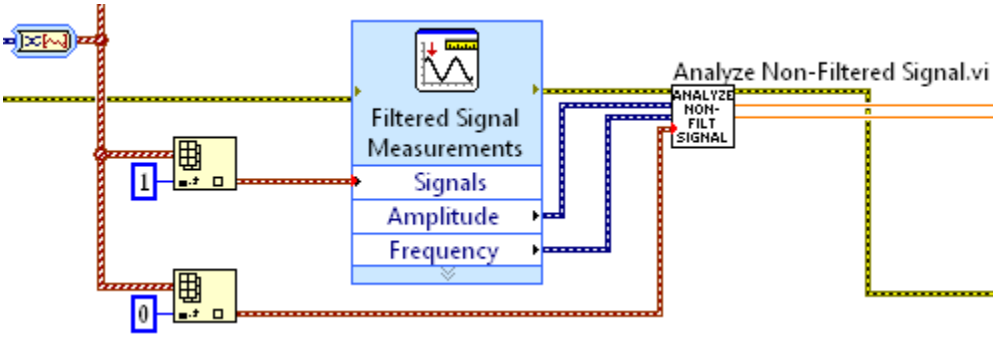
24. Wire the **element** output terminal of the first **Index Array** function to the **Signals** input terminal of the **Filtered Signal Measurements (Tone Measurements) Express VI**.



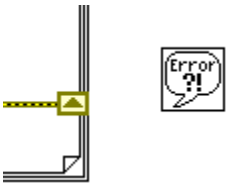
25. Add a constant to the second Index Array function to obtain the 0th element.
    a. Right-click on the **index** input terminal of the second Index Array function.
    b. Select **Create»Constant**.
    c. Change the value of the numeric constant to **0**.



26. Wire the **element** output terminal of the second **Index Array** function to the **Sine** input terminal of the **Analyze Non-Filtered Signal SubVI**.

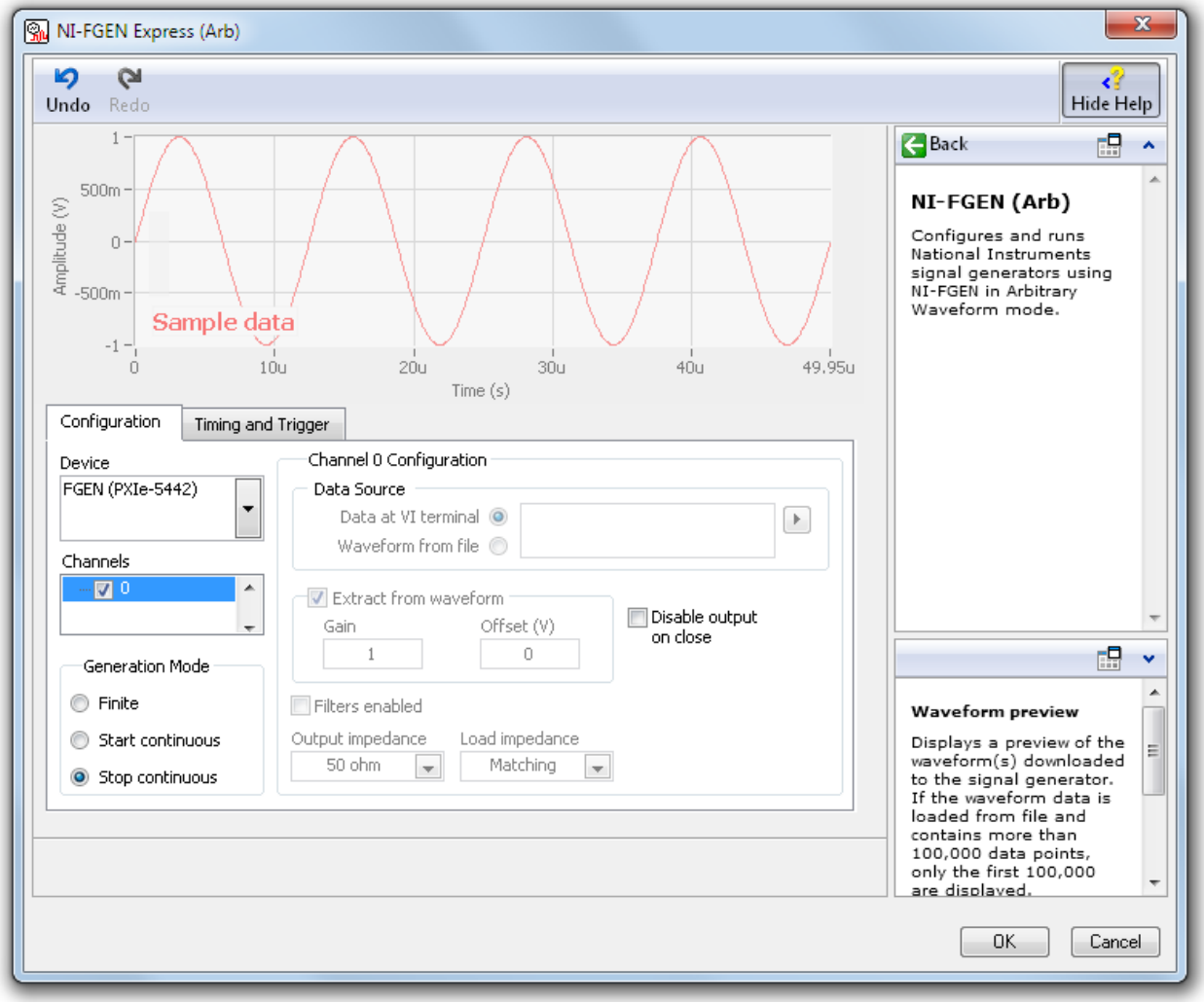


27. Delete the Error wire from the right side of the For Loop to the Simple Error Handler function.



28. Create space between the For Loop and the Simple Error Handler function.
29. Add an NI-FGEN Express (arb) VI to right of the For Loop to stop generation.
    a. Right-click the block diagram.
    b. Navigate to **Measurement I/O»NI-FGEN»NI-FGEN Express (Arb)**.
    c. Drag and drop the NI-FGEN Express (Arb) VI between the For Loop and the Simple Error Handler function.

d. Select **Stop continuous** under Generation mode.

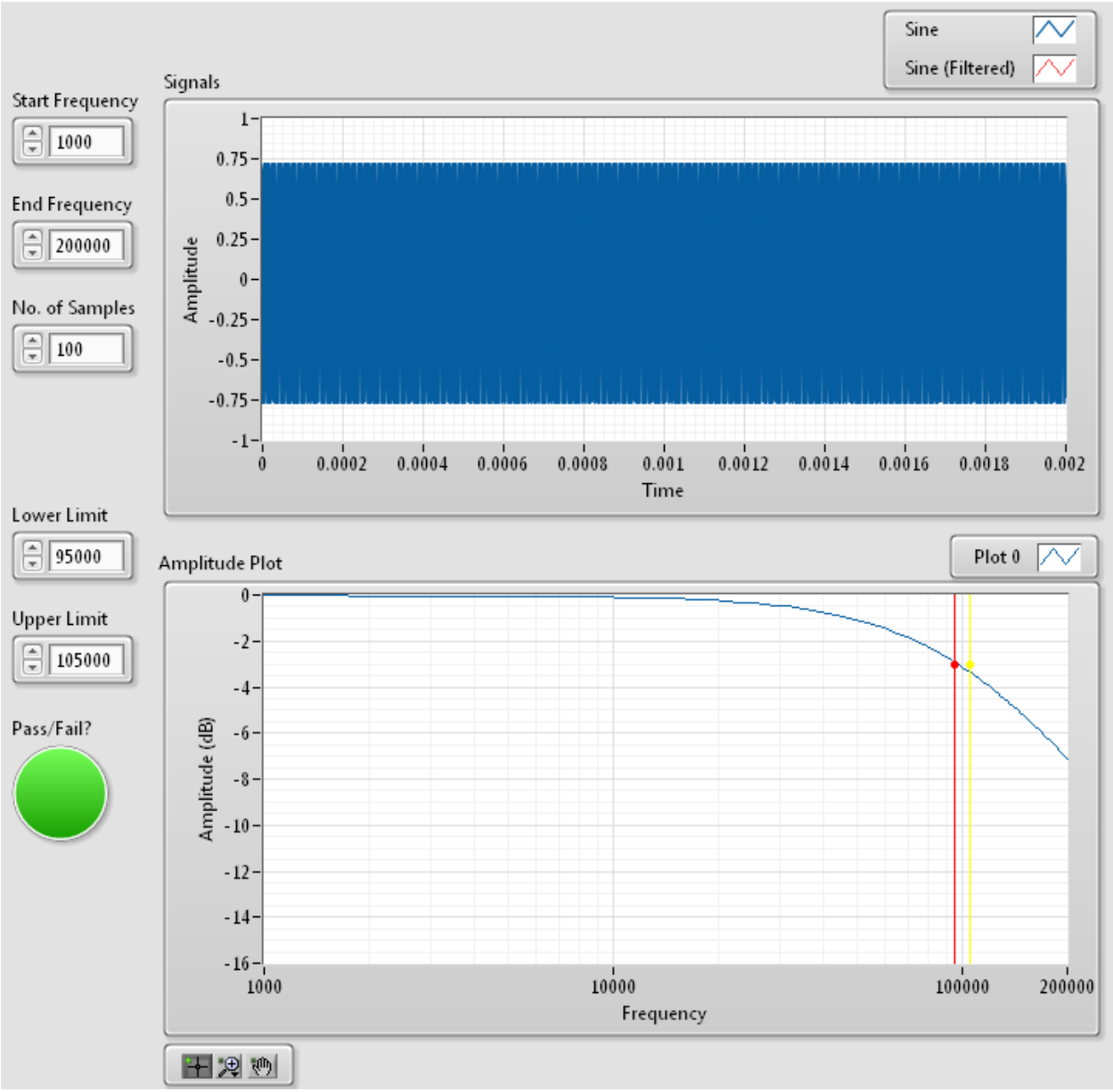


e. Select **OK** .

30. Wire the **Error** wire from the Fop Loop through the **NI-FGEN Express (Arb) VI** and to the **Simple Error Handler** function.

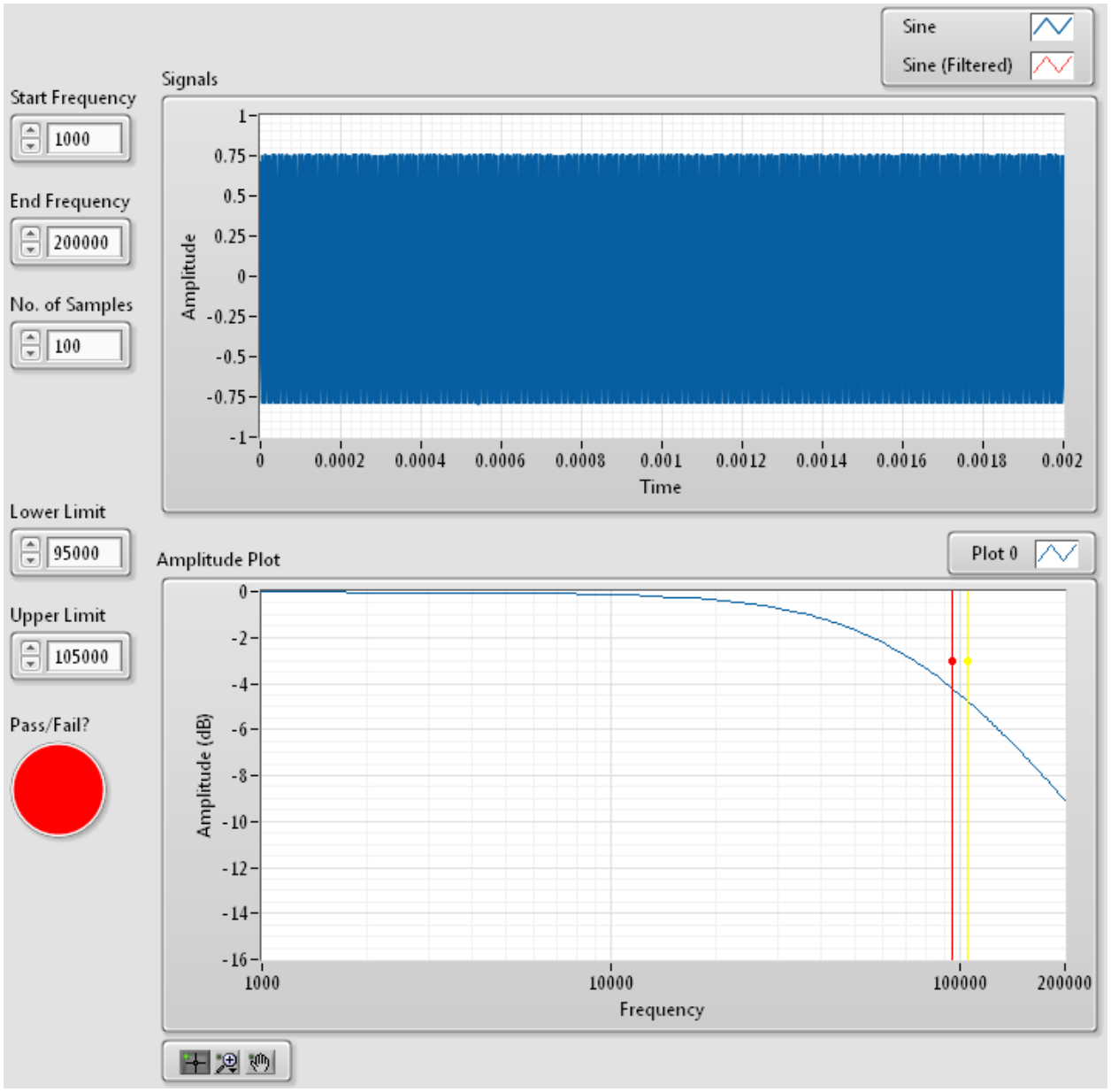
31. On the PXI system Automated Test DEMO Socket 0 UUT, set the Filter potentiometer to the **pass** position.

32. **Run** the Filter Test application and observe that the filter test passes.



33. Change the Filter potentiometer to the **fail** position.

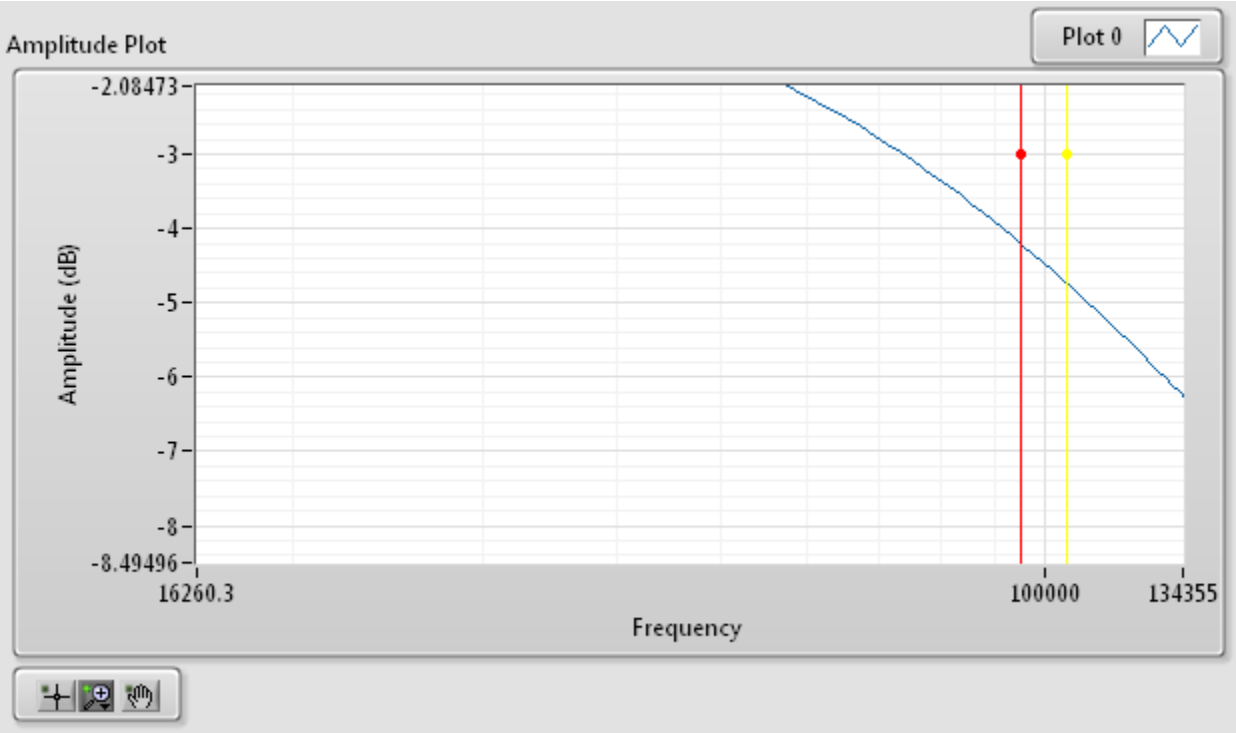
34. **Run** the Filter Test application one more time and observe that the filter test fails.



35. Zoom in on the Amplitude Plot to observe the point at which the signal failed.

    a. Select **Zoom In** on the Graph Palette.

b. Click on the **Amplitude Plot** near the red and yellow cursors to zoom in.



36. **Save** and **Close** the application.

EXERCISE 4.3 – LOWPASS FILTER TEST WITH STANDARD VIS

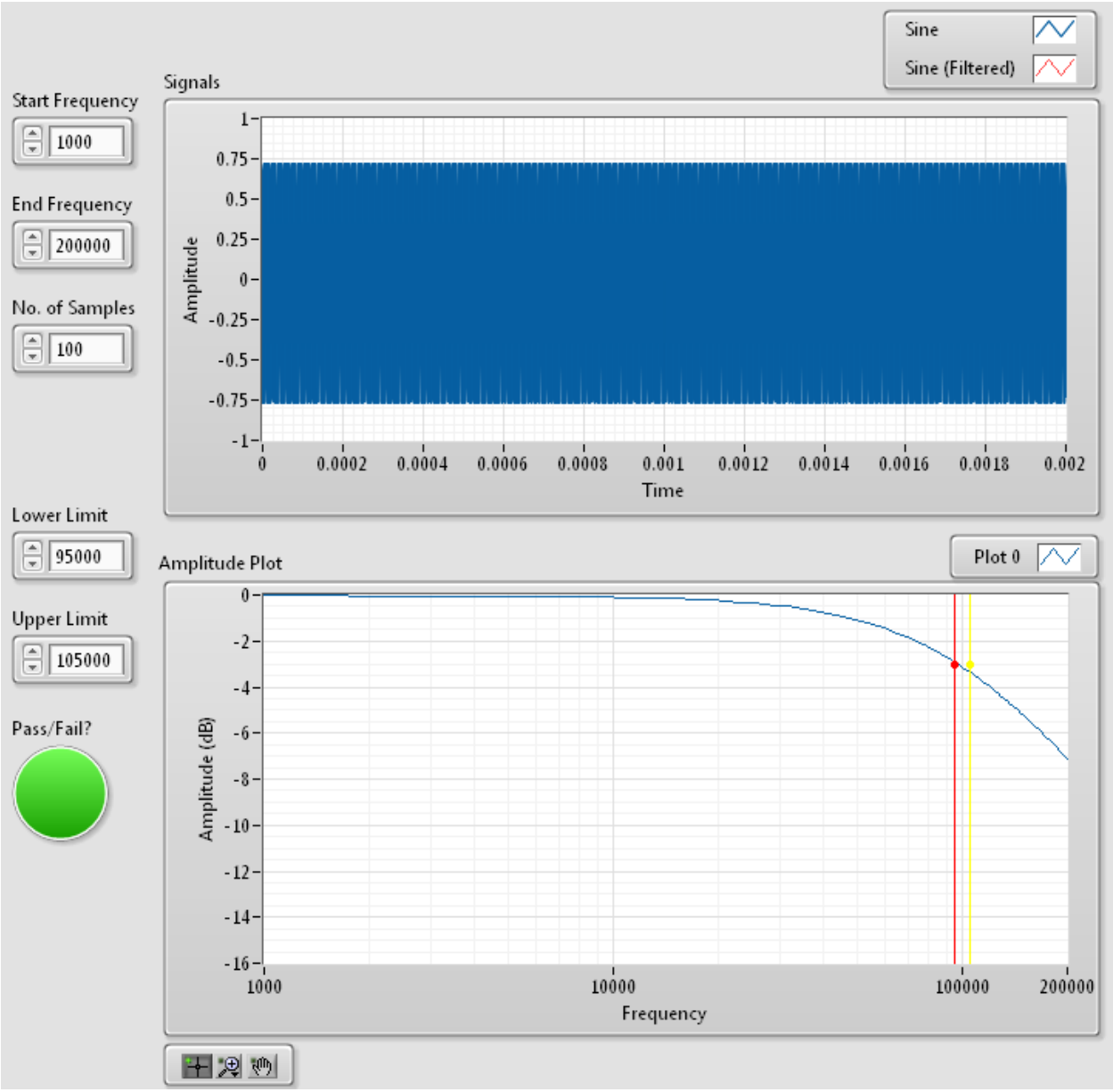
Goal

The goal of this exercise is to open and run a pre-built example that shows how you could implement a filter test using Standard VIs, rather than Express VIs.

Steps

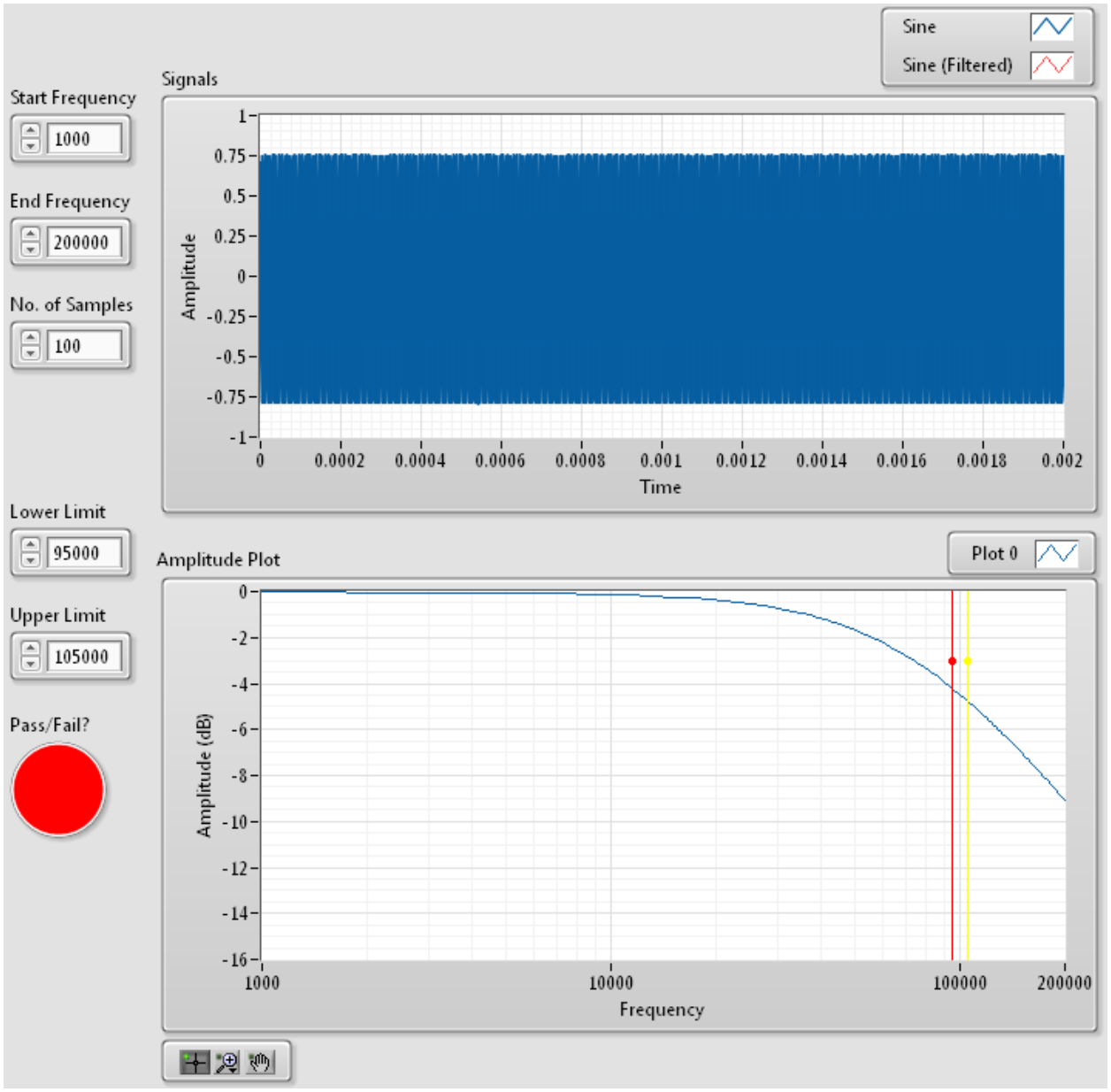
1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browsing to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the **LabVIEW for Instrumentation Hands-On.lvprj** file.
2. In the Filter Test folder, open the Filter Test – Standard VIs.
3. Notice that the front panel looks similar to the Filter Test you previously created with the Express VIs.
4. Toggle to the block diagram using <Ctrl-E>
5. Observe that the source code is implemented using Standard VIs from the FGEN and SCOPE instrument drivers.
6. On the AT DEMO UUT in Socket 0, set the Filter potentiometer to the pass position.

7. **Run** the Filter Test application and observe that the filter test passes.



8. Change the Filter potentiometer to the fail position.

9. **Run** the **Filter Test** application one more time and observe that the filter test fails.



10. **Close** the application and *do not save*.

EXERCISE 5.1 – MEASURE TEMPERATURE USING THE DMM SOFT FRONT PANEL

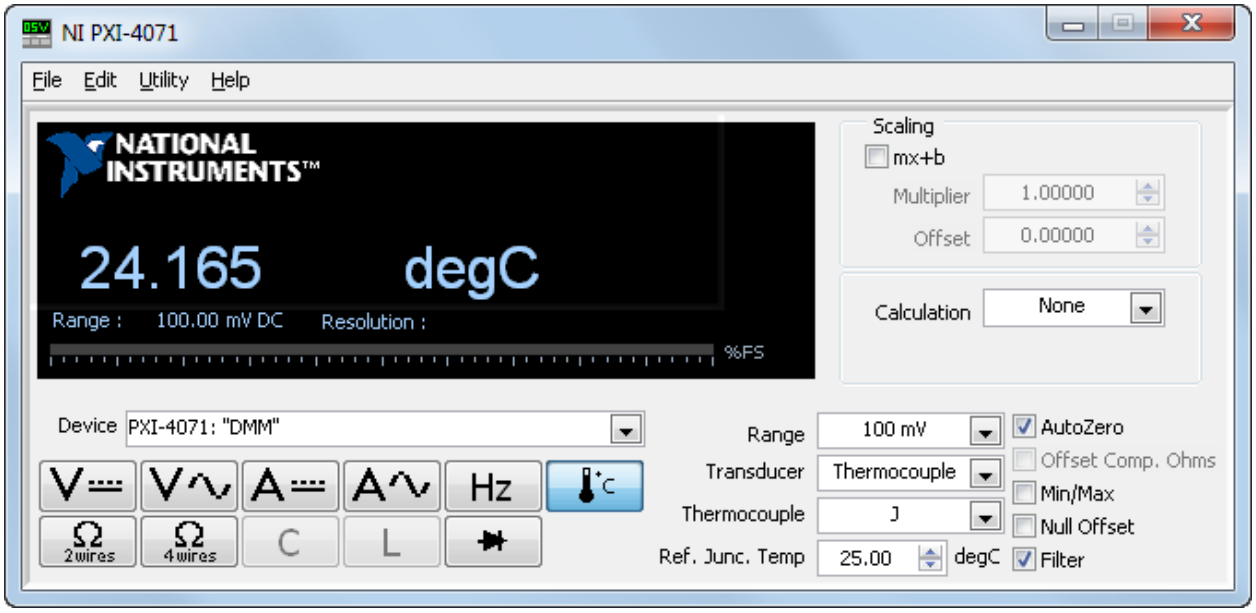
Goal

The goal of this exercise is to use the DMM Soft Front Panel to measure the temperature .

Steps

1. Open the **DMM Soft Front Panel** from **Start»All Programs»National Instruments»NI-DMM»DMM Soft Front Panel.**
2. Change the measurement type to **Temperature**.

3. Ensure that the **DMM Soft Front Panel** is configured as follows.
    a. **Range = 100 mV**
    b. **Transducer = Thermocouple**
    c. **Thermocouple = J**
    d. **Ref Junc. Temp = 25.00**



4. Hold the thermocouple and observe that the temperature measurement increases.
5. Close the DMM Soft Front Panel.

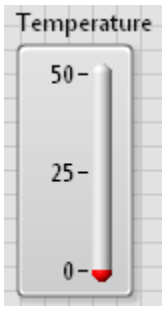
EXERCISE 5.2 – READ TEMPERATURE

Goal

The goal of this exercise is to create a LabVIEW application that will take temperature measurements.

Steps

1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browsing to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the LabVIEW for Instrumentation Hands-On.lvprj file.
2. Create a new VI in the Temperature Test folder by right–clicking on the folder and selecting **New»VI**.
3. Select **File»Save** and name the VI Read Temperature.
4. Tile the front panel and block diagram by selecting **Window»Tile Left and Right**.
   Tip: You can also tile the front panel and block diagram using <Ctrl-T>.
5. Add a Numeric Thermometer to the front panel.
   a. Right-click the front panel.
   b. Navigate to **Silver»Numeric»Thermometer.**
   c. Drag and drop the Thermometer onto the front panel**.**
   d. Name the Thermometer **Temperature.**
   e. Change the range of the Temperature Thermometer to **[0, 50].**

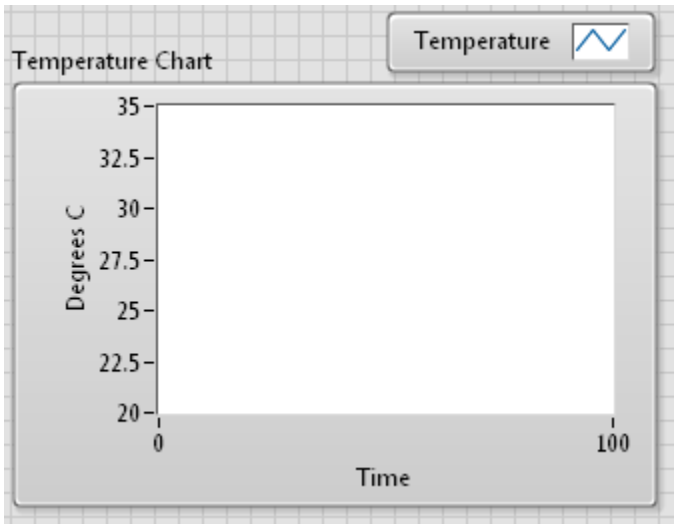


6. Add a Waveform Chart to the front panel.
   a. Right-click the front panel**.**
   b. Navigate to **Silver»Graph»Waveform Chart.**
   c. Drag and drop the Waveform Chart onto the front panel**.**
   d. Name the Waveform Chart **Temperature Chart**.
   e. Change Plot 0 to **Temperature** in the Plot Legend.

f. Change the Y-Axis title to **Degrees C.**

g. Change the Y-Axis to not auto scale by right-clicking on the Y-Axis and deselecting **AutoScale.**

h. Change the range of the Y-Axis to **[20, 35]**.

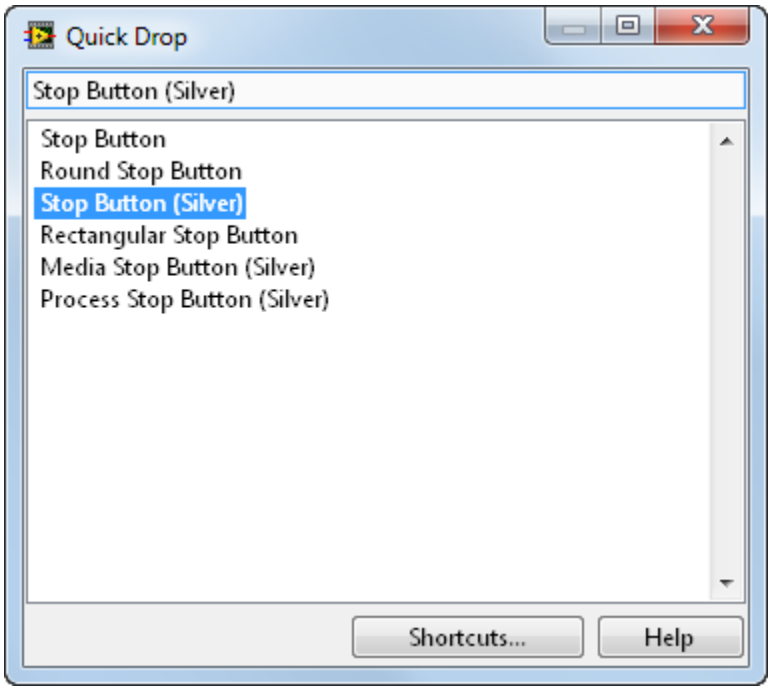


7. Add a Stop Button to the front panel.

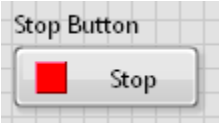
a. Use <Ctrl-Space> to bring up the **Controls Quick Drop** window.

b. Type `Stop Button` into the Quick Drop search window.

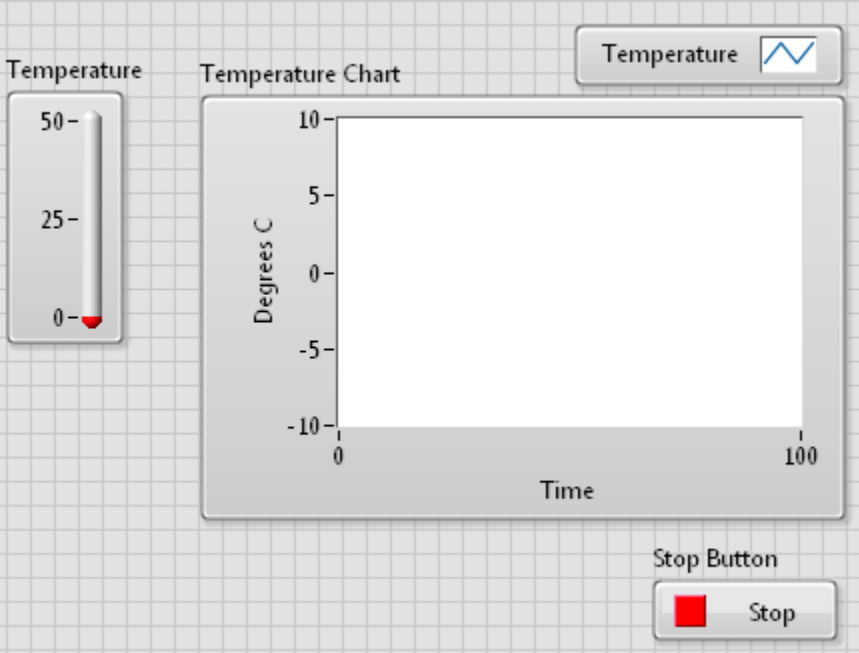
c. Double-click **Stop Button (Silver)**.



d. Drop the Stop Button onto the front panel.

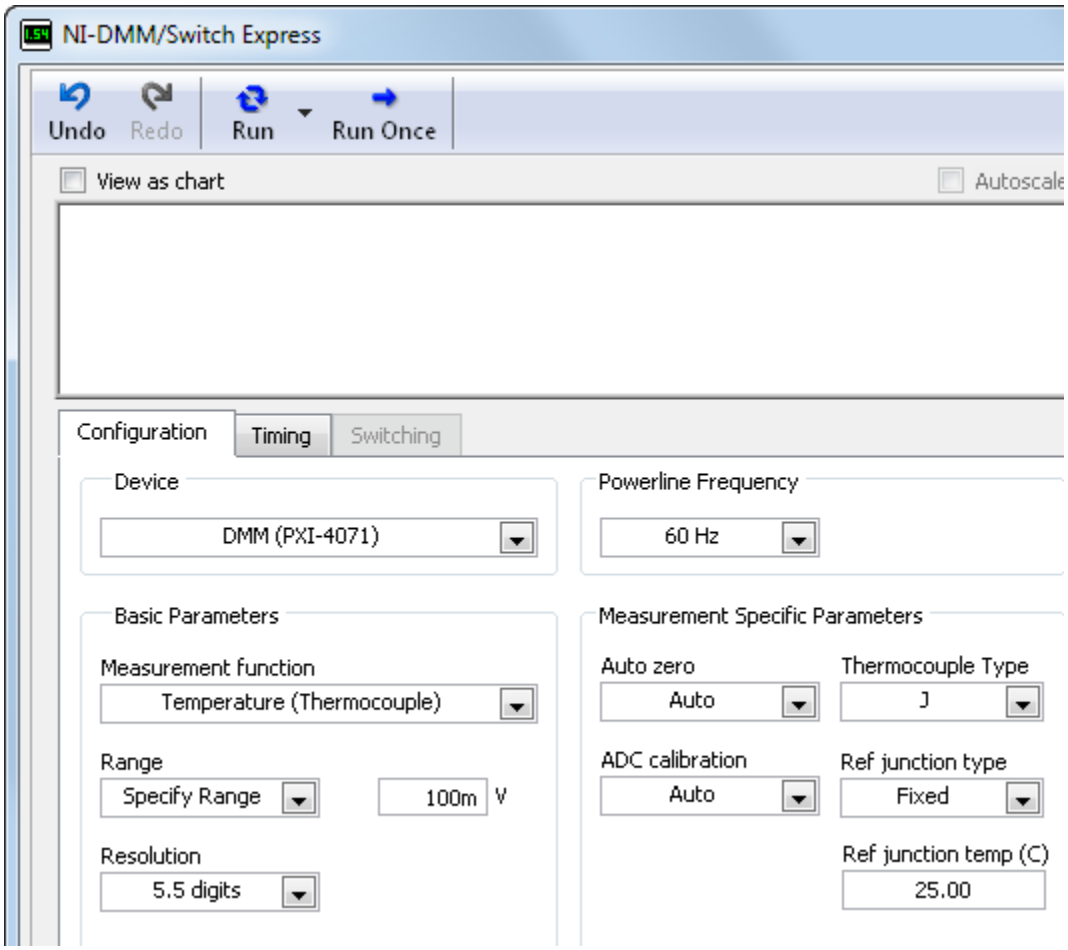


8. Your front panel should look similar to the following user interface.



9. Toggle to the block diagram using <Ctrl-E>.
10. Add a DMM Express VI to the block diagram.
    a. Right-click the block diagram.
    b. Navigate to **Measurement I/O»NI-DMM»DMM/Switch Express.**
    c. Drag and drop the DMM/Switch Express VI onto the block diagram**.**

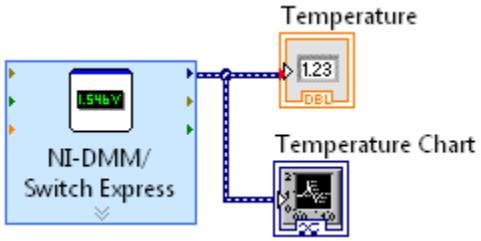
11. Configure the following parameters in the NI-DMM/Switch Express VI window.
    a. **Measurement function = Temperature (Thermocouple)**
    b. **Range = 100 mV**
    c. **Resolution = 5.5 digits**
    d. **Thermocouple Type = J**



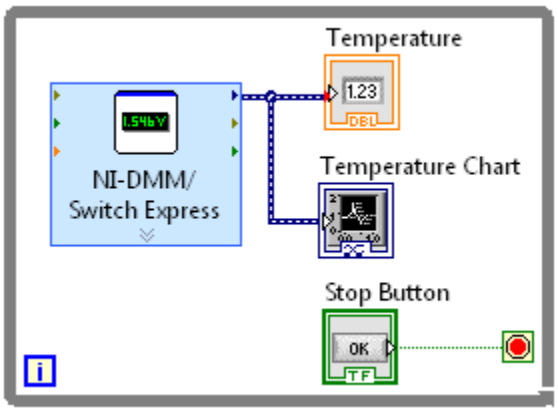
    e. Select **OK.**

12. Wire the **signals out** output terminal from the **DMM Express VI** to the **Temperature** Numeric Indicator.

13. Wire the **signals out** output terminal from the **DMM Express VI** to the **Temperature Chart** Waveform Chart.

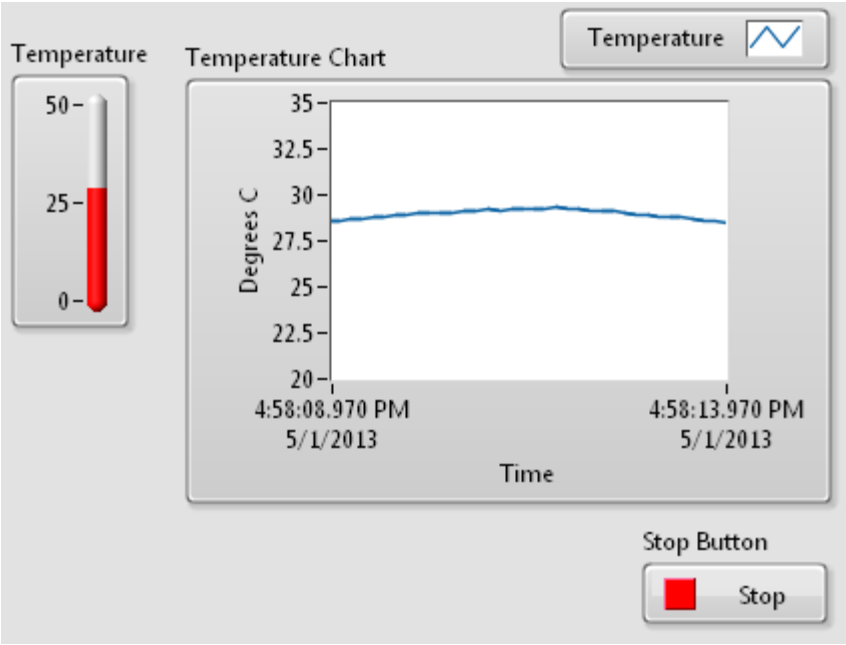


14. Add a While Loop to the block diagram around all of your code.
    a. Right-click the block diagram.
    b. Navigate to **Programming»Structures»While Loop.**
    c. Drag and draw the While Loop around all of your code on the block diagram**.**
15. Wire the Stop Button to the Loop Condition of the While Loop.
16. Clean up the block diagram code by selecting **Edit»Clean Up Diagram** from the block diagram menu or by using <Ctrl-U>.



17. **Run** the application.

18. Hold the thermocouple and observe that the temperature measurement increases.



37. Press the **Stop Button** to stop the application.
38. **Save** the application, but *do not close it*.

EXERCISE 5.3 – TEMPERATURE TEST

Goal

The goal of this exercise is to create a LabVIEW application that will use the DMM to continuously take temperature measurements and throw an alarm if the temperature exceeds a given limit.

Steps

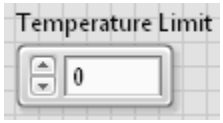
1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browsing to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the LabVIEW for Instrumentation Hands-On.lvprj file.
2. Create a copy of the Read Temperature VI you created in Exercise 5.2**.**
    a. Open the Read Temperature VI in the Temperature Test folder .
    b. Select **File»Save As**.
    c. Select **Create unopened disk copy**.
    d. Click **Continue**.
    e. Brose to `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Temperature Test`.
    f. Name the VI **Temperature Test.**
    g. Select **OK**.
    h. Add the New VI to the Temperature Test Folder in the Project Explorer.
        i. Right-click the Temperature Test folder in the Project Explorer.
        ii. Select **Add»File**.
        iii. Browse to Browse to the Temperature Test Folder `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Temperature Test`.
        iv. Select Temperature Test.vi.
        v. Click **Add File**.
    i.

Note: The Solutions folder has completed LabVIEW applications for you to use in the event that you do not complete the exercises `C:\Seminars\LabVIEW for Instrumentation Hands-On\Solutions`.

3. Tile the front panel and block diagram using <Ctrl-T>.
4. Add a **Numeric Control** to the front panel**.**
    a. Right-click the front panel.
    b. Navigate to **Silver»Numeric»Numeric Control.**

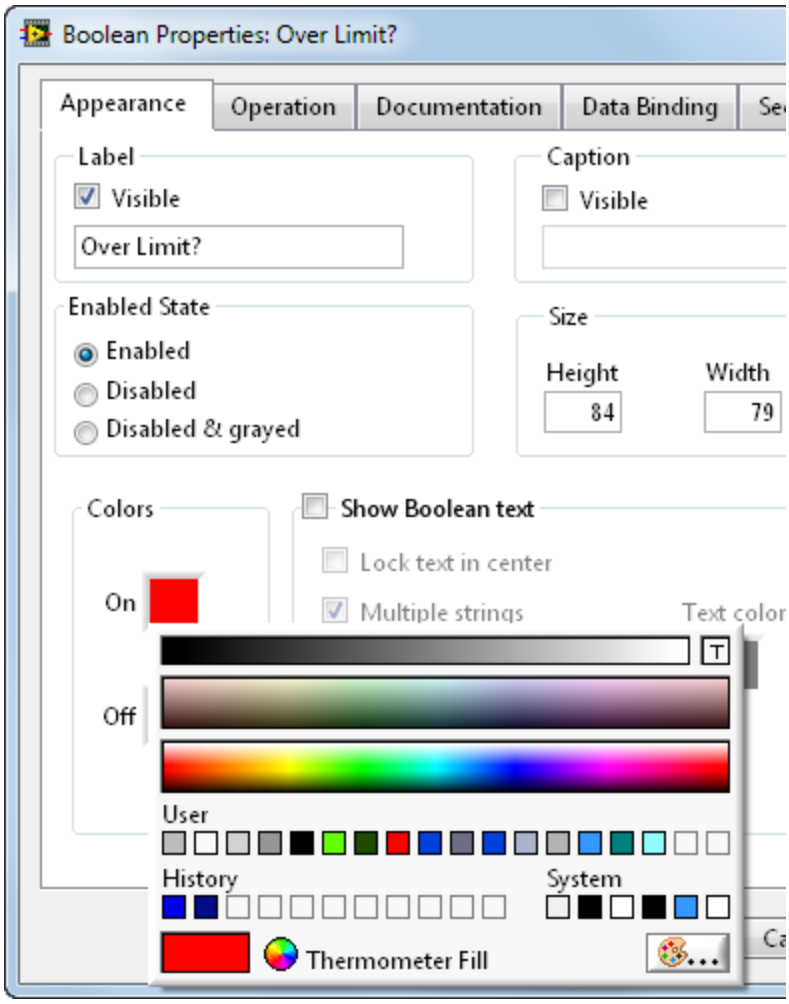
c. Drag and drop the Numeric Control onto the front panel**.**

d. Name the Numeric Control **Temperature Limit.**



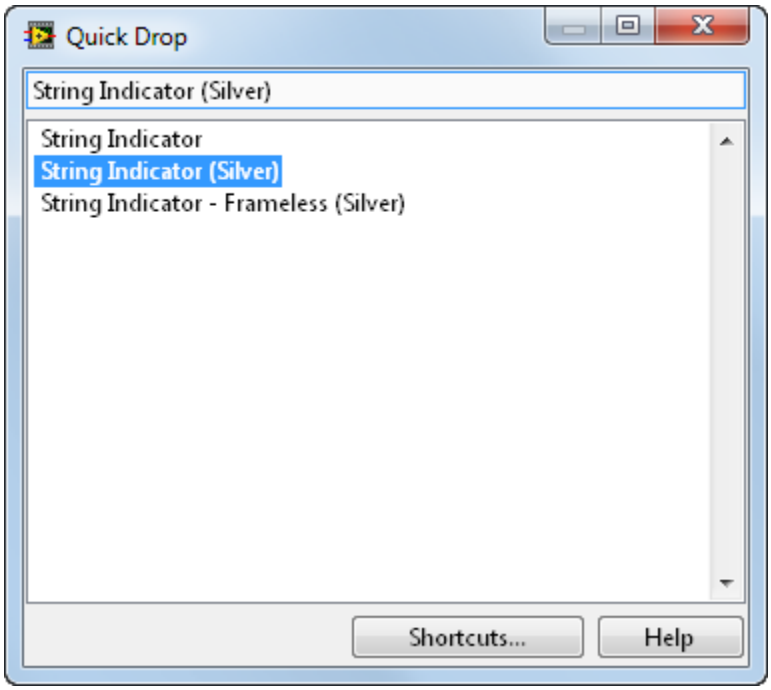
5. Add an LED to the front panel.
    a. Right-click the front panel.
    b. Navigate to **Silver»Boolean»LED.**
    c. Drag and drop the LED onto the front panel**.**
    d. Name the LED **Over Limit?.**
    e. Resize the LED by dragging the corners**.**

6. Change the On color of the LED to **Red**.
    a. Right-click the LED and select **Properties**.
    b. Change the On color to **Red**.

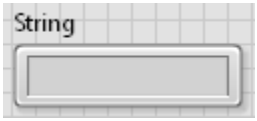

Boolean Properties: Over Limit?
Appearance
Operation
Documentation
Data Binding
Label
Visible
Over Limit?
Caption
Visible
Enabled State
Enabled
Disabled
Disabled & grayed
Size
Height
Width
84
79
Colors
On
Off
Show Boolean text
Lock text in center
Multiple strings
Text color
User
History
System
Thermometer Fill

c. Select **OK**.

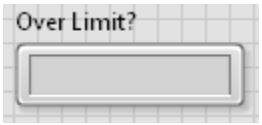
7. Add a String Indicator to the front panel**.**
    a. Use <Ctrl-Space> to bring up the Controls Quick Drop window.
    b. Type `String Indicator` into the Quick Drop search window.
    c. Double-click **String Indicator (Silver)**.



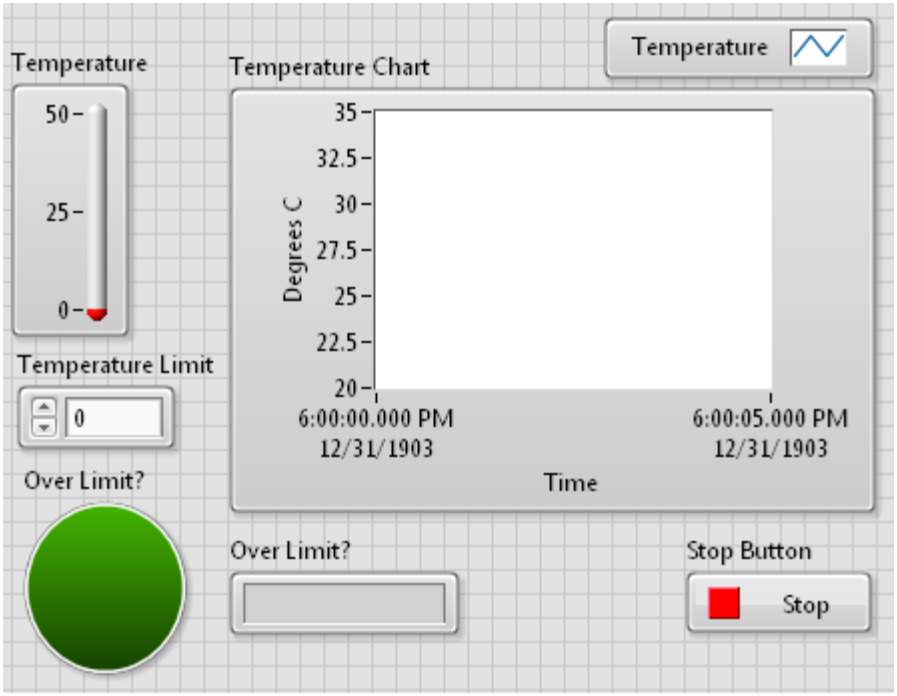
    d. Drop the String Indicator onto the front panel.



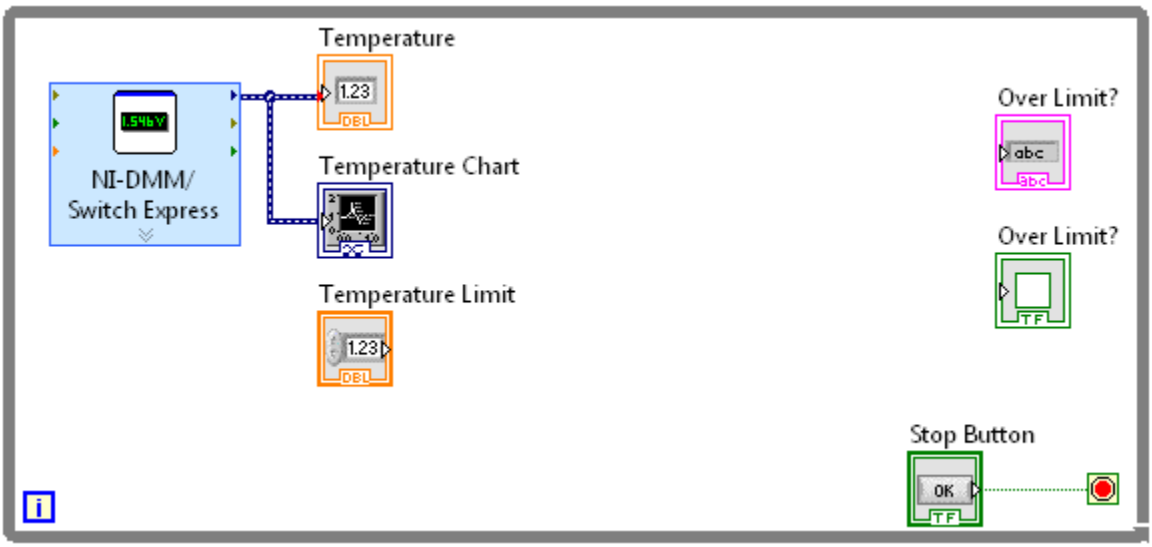
    e. Name the String Indicator **Over Limit?.**



8. Your front panel should look similar to the following user interface.



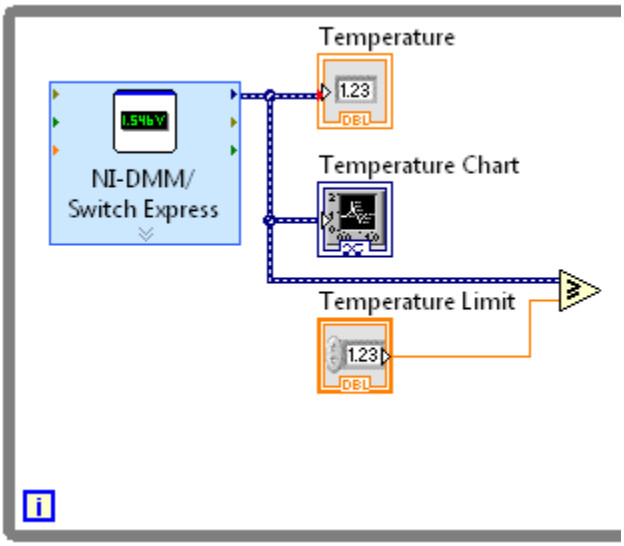
9. Toggle to the block diagram using <Ctrl-E>.
10. On the block diagram, drag all of the new icons in to the While Loop.



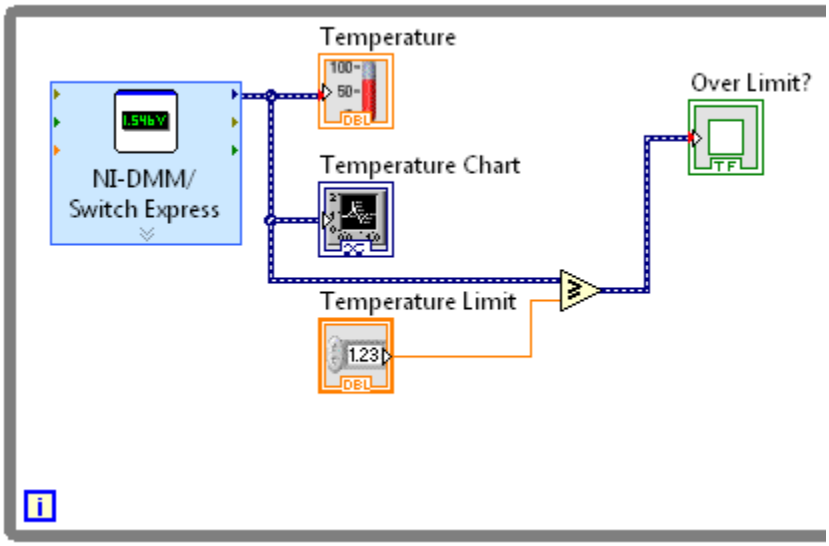
11. Add a Greater Or Equal? function to the block diagram**.**
    a. Right-click the block diagram.
    b. Navigate to **Programming»Comparison»Greater Or Equal?.**

c. Drag and drop the Greater Or Equal? function onto the block diagram.

12. Wire the **signals out** output terminal of the DMM Express VI to the **x** input terminal of the **Greater Or Equal?** function.
13. Wire the **Temperature Limit** Numeric Control to the **y** input terminal of the **Greater Or Equal?** function.

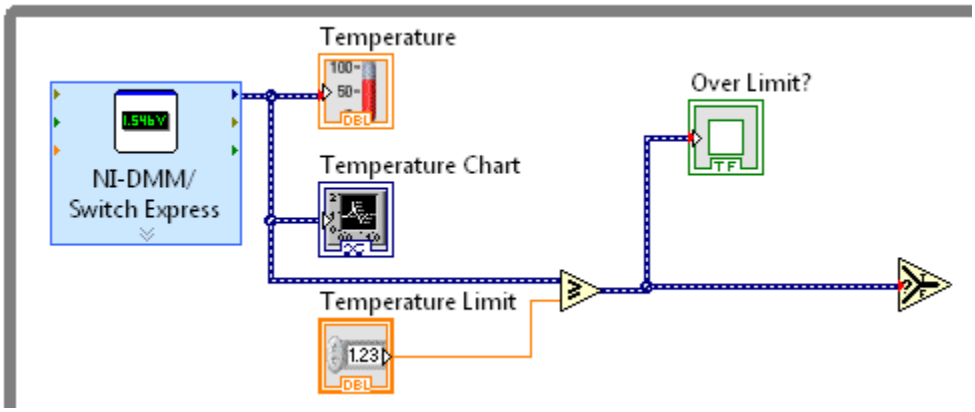


14. Wire the **x >= y?** output terminal of the **Greater Or Equal?** function to the **Over Limit?** Indicator .

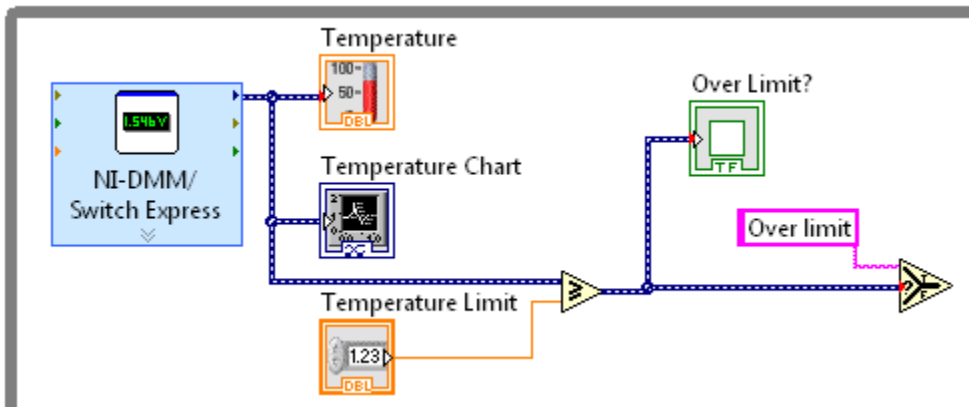


15. Add a Select function to the block diagram.
    a. Right-click the block diagram.
    b. Navigate to **Programming»Comparison**.
    c. Drag and drop the Select function onto the block diagram.

16. Wire the **x >= y?** output terminal of the **Greater Or Equal?** function to the **s** input terminal of the **Select** function .

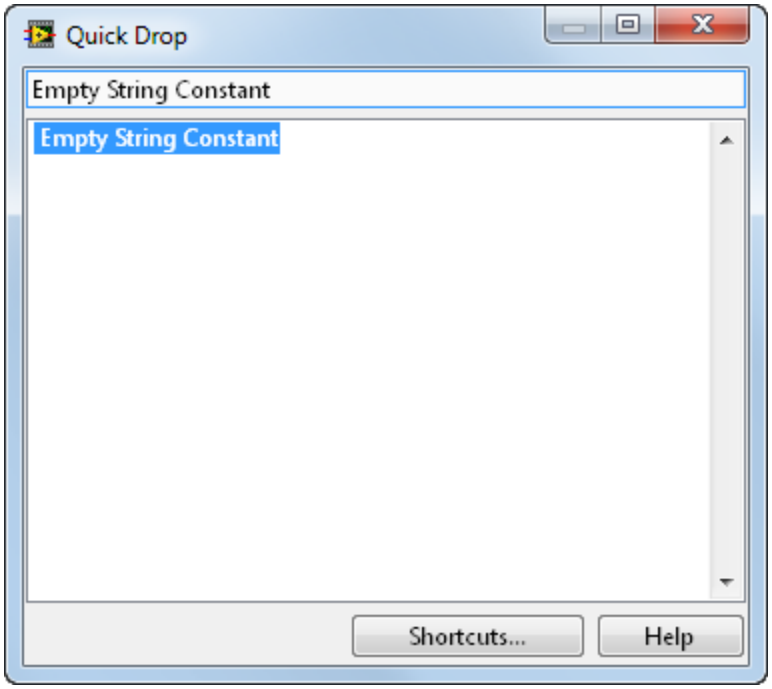


17. Add a String Constant to the block diagram.
    a. Right click the block diagram.
    b. Navigate to **Programming»String»String Constant**.
    c. Drag and drop a String Constant onto the block diagram .
    d. Type `Over Limit` in the String Constant.
    e. Wire the Over Limit String Constant to the **t** input terminal of the **Select** function .



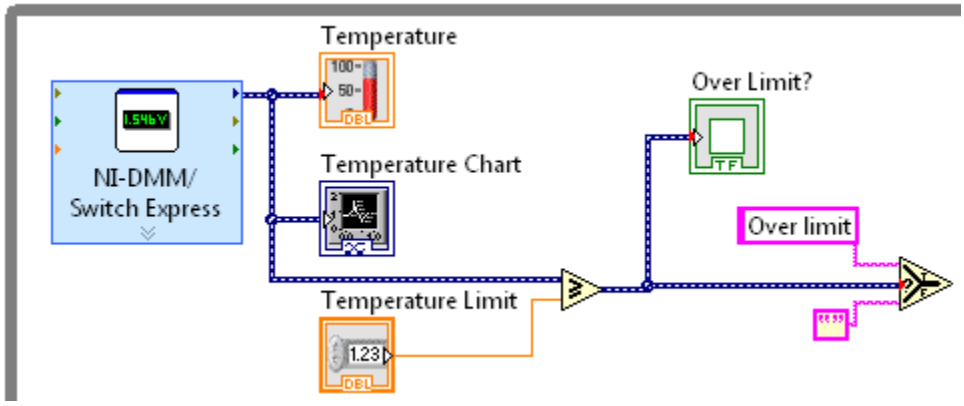
18. Add an Empty String Constant to the block diagram.
    a. Use <Ctrl-Space> to bring up the Functions Quick Drop window.
    b. Type `Empty String Constant` into the Quick Drop search window.

c. Double-click **Empty String Constant**.

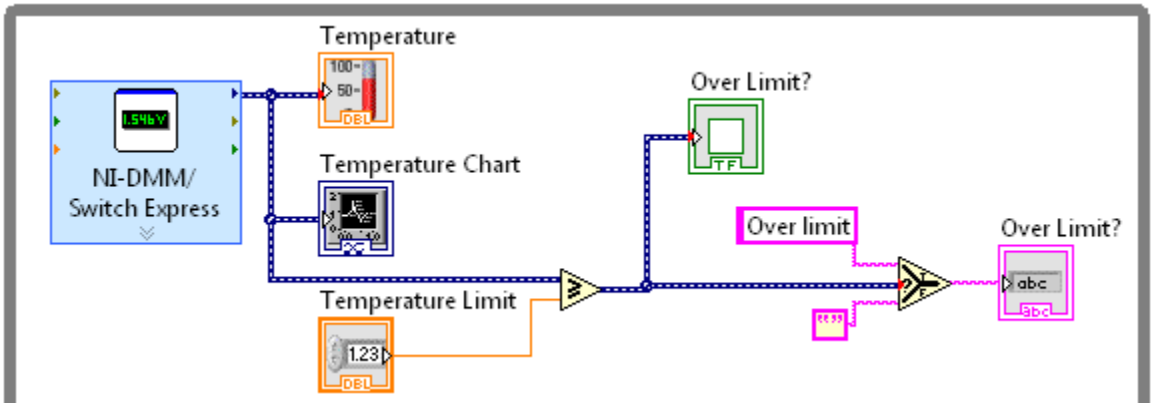


d. Drop the Empty String Constant onto the block diagram.

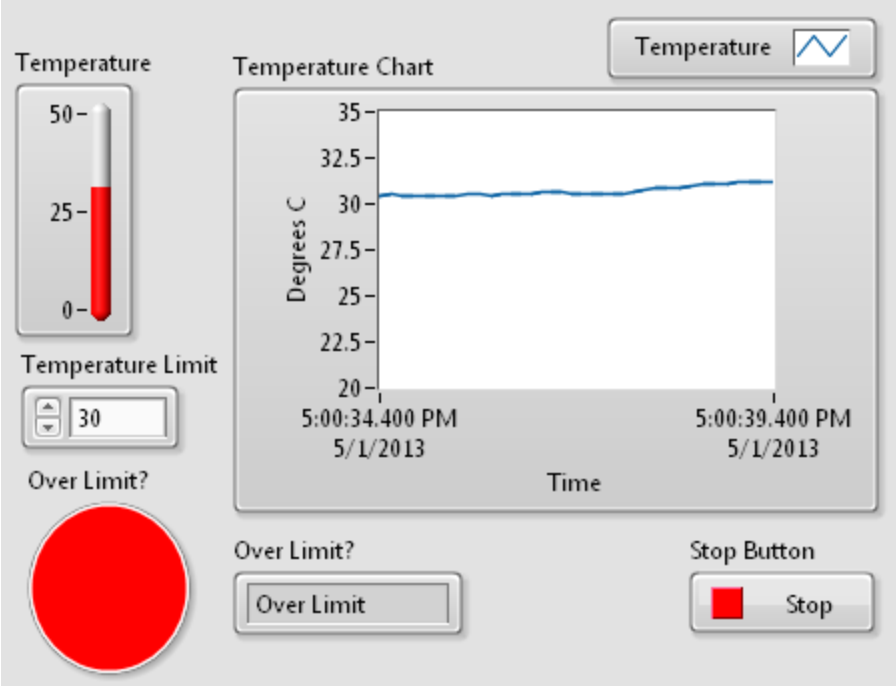
e. Wire the **Empty String Constant** to the **f** input terminal of the **Select** function.



19. Wire the **s?t:f** output terminal of the **Select** function to the **Over Limit?** String Indicator.



20. Toggle to the front panel using <Ctrl-E>.
21. On the front panel, change the Temperature Limit value to **30**.
22. **Run** the application.
23. Hold the thermocouple and observe that the temperature measurement increases and that the Over Limit? alarm is thrown if the temperature exceeds 30 degrees C.



39. Press the **Stop Button** to stop the application.
40. **Save** and **Close** the application.

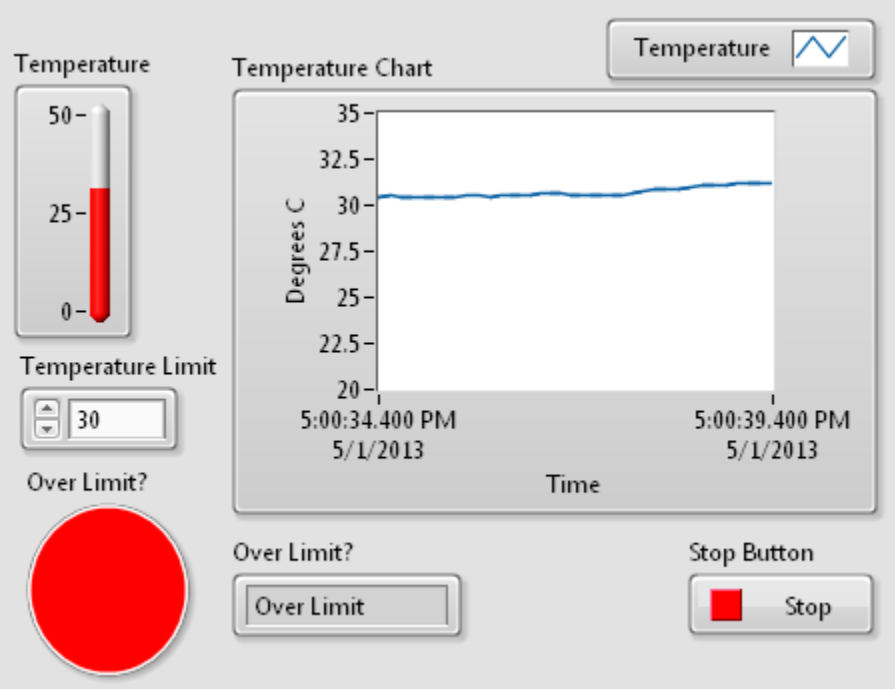
EXERCISE 5.4 – TEMPERATURE TEST WITH STANDARD VIS

Goal

The goal of this exercise is to open and run a pre-built example that shows how you could implement a Temperature test using Standard VIs, rather than Express VIs.

Steps

1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browsing to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the LabVIEW for Instrumentation Hands-On.lvprj file.
2. In the Temperature Test folder, open the Temperature Test – Standard VIs.
3. Open the block diagram and observe that the source code is implemented using Standard VIs from the DMM instrument driver.
4. On the front panel, change the Temperature Limit value to **30**.
5. **Run** the application.
6. Hold the thermocouple and observe that the temperature measurement increases and that the Over Limit? alarm is thrown if the temperature exceeds 30 degrees C.



7. Press the **Stop Button** the stop the application.
8. **Close** the application and *do not save*.

Exercise 6.1 – Supply Power to an LED Using the DCPower Soft Front Panel

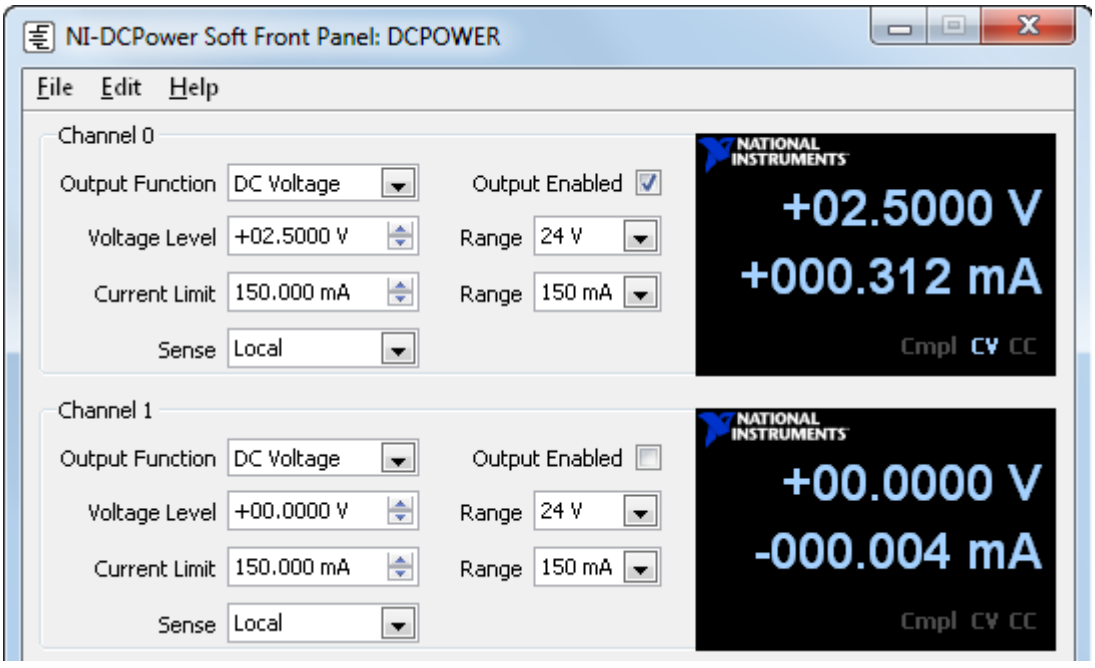
Goal

The goal of this exercise is to use the DCPower Soft front panel to control the NI-SMU and supply power to an LED on the UUT to light it up.

Steps

1. Open the DCPower Soft Front Panel from **Start»All Programs»National Instruemnts» NI-DCPower»NI-DCPower Soft Front Panel**.
2. On **Channel 0**, configure the following:
    a. Check the **Output Enabled** checkbox.
    b. Change the Voltage Level to **2.5** V.

    Warning! Do not supply more than 5 V.
3. Notice that the LED on the Socket 0 UUT lights up.



4. Power the LED off by Changing the Channel 0 Voltage Level to **0**.
5. Close the DCPower Soft Front Panel**.**

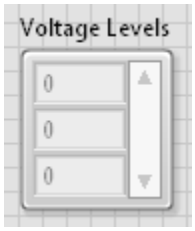
EXERCISE 6.2 – SUPPLY POWER TO AN LED

Goal

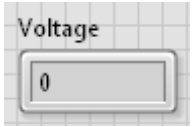
The goal of this exercise is to create a LabVIEW application that will supply power to an LED on the UUT and light it up.

Steps

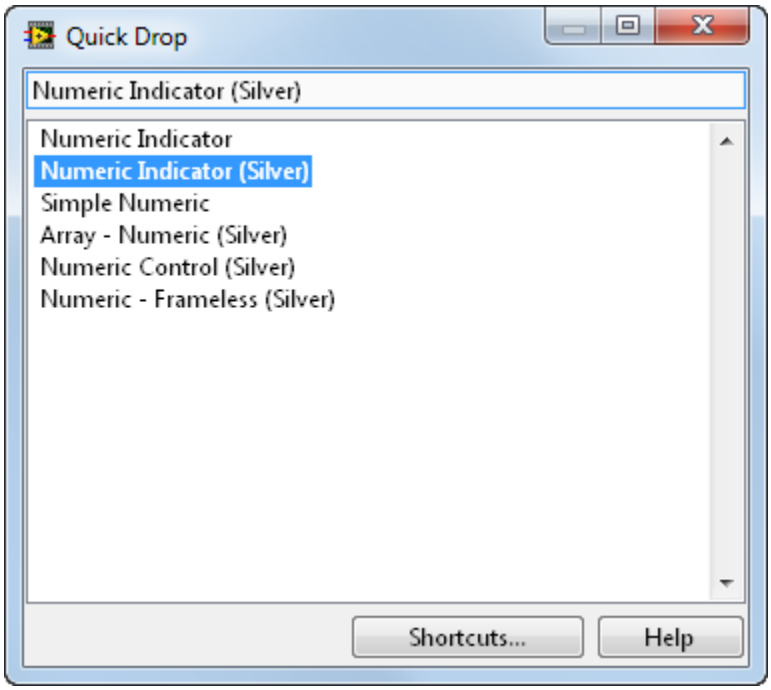
1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browsing to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the LabVIEW for Instrumentation Hands-On.lvprj file.
2. Create a new VI in the LED Test folder by right–clicking on the folder and selecting **New»VI**.
3. Select **File»Save** and name the VI **Turn LED On**.
4. Tile the front panel and block diagram using <Ctrl-T>.
5. Add a **Numeric Array** to the front panel.
    a. Right-click on the front panel.
    b. Navigate to **Silver»Array, Matrix & Cluster»Array-Numeric**.
    c. Drag and drop the Numeric Array onto the front panel.
    d. Name the Numeric Array **Voltage Levels**.



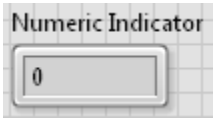
6. Add a **Numeric Indicator** to the front panel to display the Voltage.
    a. Right-click on the front panel.
    b. Navigate to **Silver»Numeric»Numeric Indicator**.
    c. Drag and drop the Numeric Indicator onto the front panel.
    d. Name the Numeric Indicator **Voltage**.



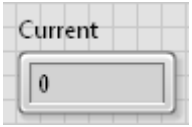
7. Add a **Numeric Indicator** to the front panel to display the Current.
    a. Use <Ctrl-Space> to bring up the Controls Quick Drop window.
    b. Type `Numeric Indicator` into the Quick Drop search window.
    c. Double-click **Numeric Indicator (Silver)**.



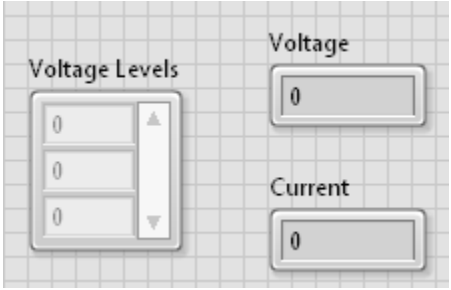
    d. Drop the Numeric Indicator onto the front panel.



    e. Name the Numeric Indicator **Current.**

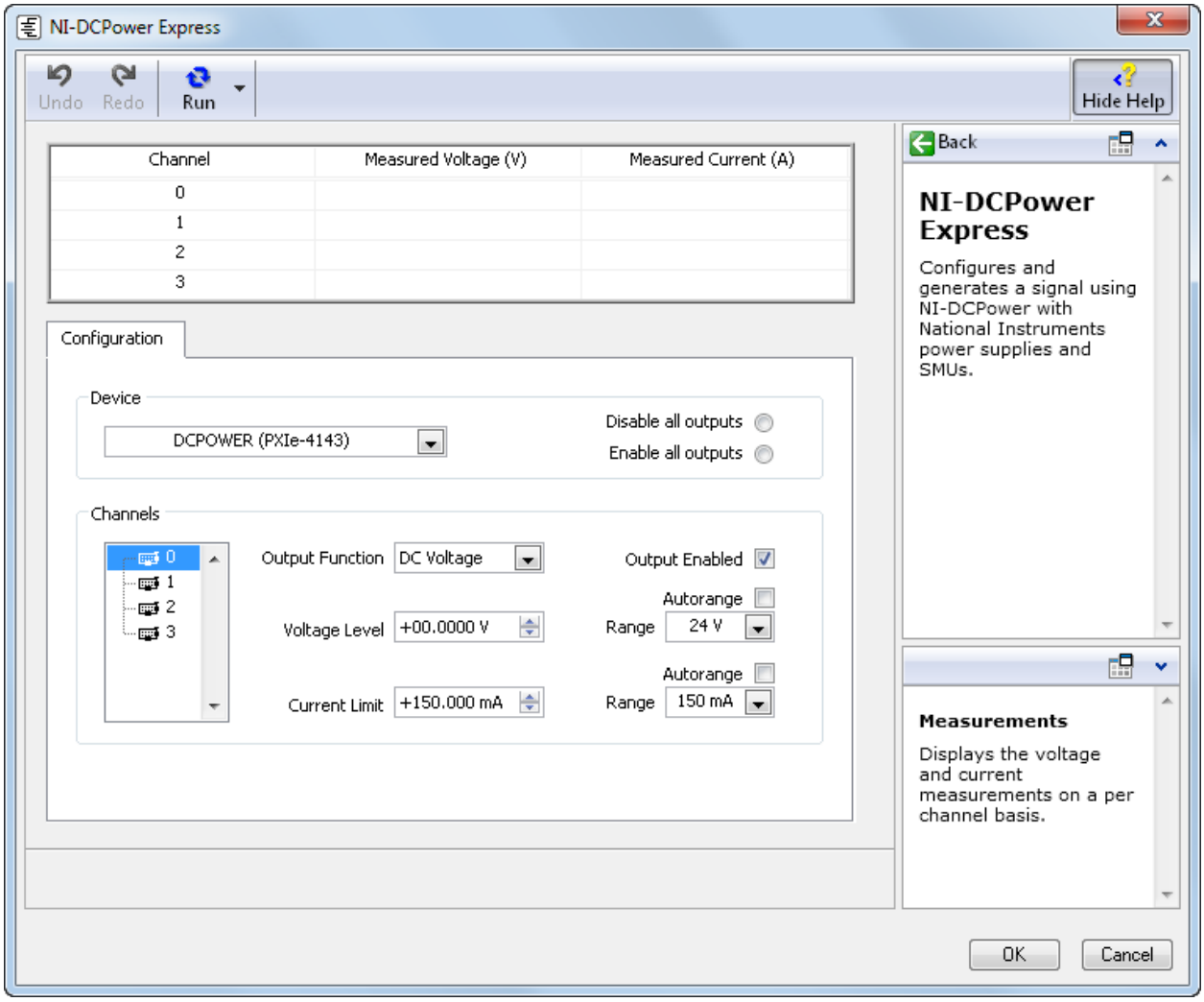


8. Your front panel should look similar to the following user interface.



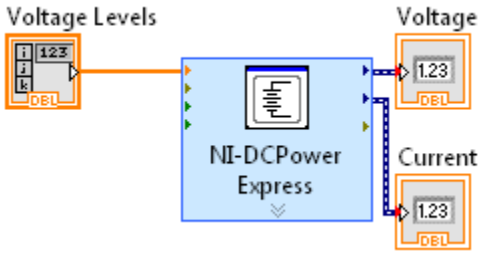
9. Toggle to the block diagram using <Ctrl-E>.

10. Add a **DCPower Express VI** to the block diagram.
    a. Right-click on the block diagram.
    b. Navigate to **Measurement IO»NI-DCPower»NI-DCPower Express**.
    c. Drag and drop the DCPower Express VI onto the block diagram.
11. In the DCPower Express configuration window:
    a. Select **Channel 0**.
    b. Check the **Output Enabled** checkbox.

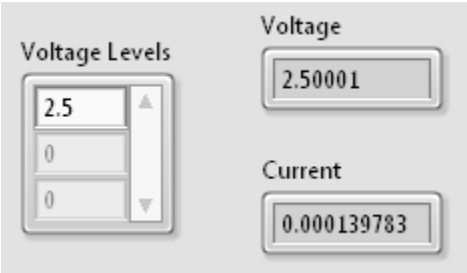


    c. Click **OK**.
12. Wire the **Voltage Levels** Numeric Array into the **voltage/current levels** input terminal of the **DCPower Express VI**.
13. Wire the **measured voltage** output terminal of the **DCPower Express VI** to the **Voltage Numeric Indicator**.

14. Wire the **measured current** output terminal of the DCPower Express VI to the **Current Numeric Indicator**.
15. Clean up the block diagram code using <Ctrl-U>.



16. Toggle to the front panel using <Ctrl-E>.
17. Change the first element of the Voltage Levels Numeric Array to **2.5**.
18. **Run** the application and observe that the LED on the Socket 0 UUT lights up.



19. Try entering Voltage Levels between the range of **2.5** and **4** (must be done by changing the value of the first element of the Voltage Levels Numeric Array). Notice that the brightness of the LED increases as the voltage is increased..

    Warning! Do not supply more than 5 V.
20. Power off the LED by changing the first element of the Voltage Levels Numeric Array to **0** and running the application.
21. **Save** and **Close** the application.

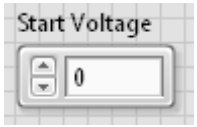
EXERCISE 6.3 – LED VOLTAGE SWEEP TEST

Goal

The goal of this exercise is to create a LabVIEW application that will use the SMU to supply a sweep of voltages to the LED and measure the voltage drop across the LED.

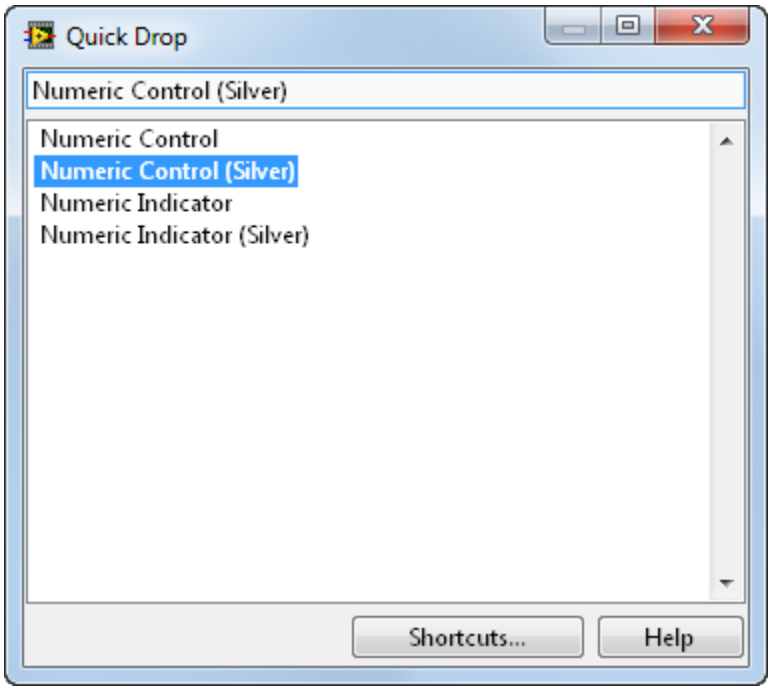
Steps

1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browsing to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the LabVIEW for Instrumentation Hands-On.lvprj file.
2. Create a new VI in the LED Test folder by right–clicking on the folder and selecting **New»VI**.
3. Select **File»Save** and name the VI **LED Voltage Sweep**.
4. Tile the front panel and block diagram using <Ctrl-T>.
5. Add a **Numeric Control** to the front panel to adjust the Start Voltage.
    a. Right-click on the front panel.
    b. Navigate to **Silver»Numeric»Numeric Control**.
    c. Drag and drop the Numeric Control onto the front panel.
    d. Name the Numeric Control **Start Voltage**.

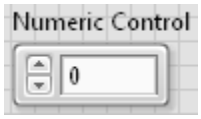


6. Add a **Numeric Control** to the front panel to adjust the End Voltage.
    a. Use <Ctrl-Space> to bring up the Controls Quick Drop window.
    b. Type `Numeric Control` into the Quick Drop search window.

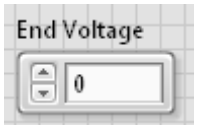
c. Double-click **Numeric Control (Silver)**.



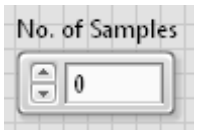
d. Drop the Numeric Control onto the front panel.



e. Name the Numeric Control **End Voltage**.

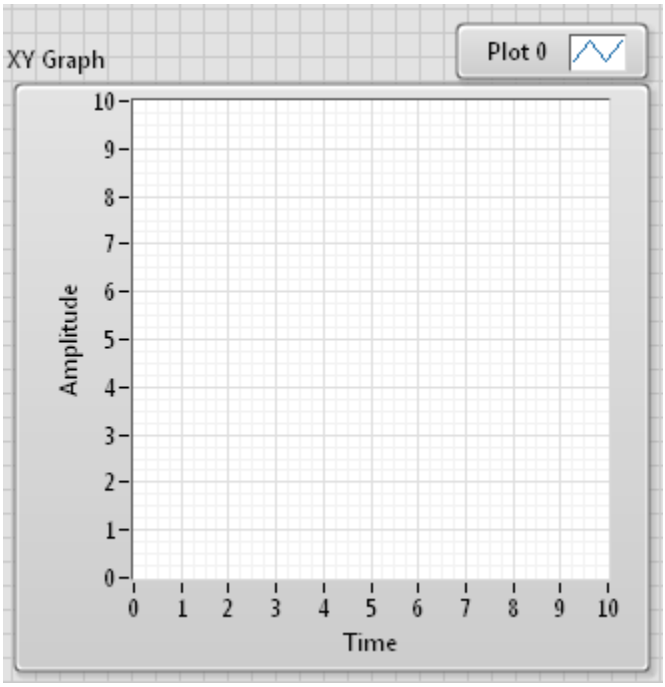


7. Add a **Numeric Control** to the front panel to adjust the Number of Samples.
   a. Right-click on the front panel.
   b. Navigate to **Silver»Numeric»Numeric Control**.
   c. Drag and drop the Numeric Control onto the front panel.
   d. Name the Numeric Control **No. of Samples**.

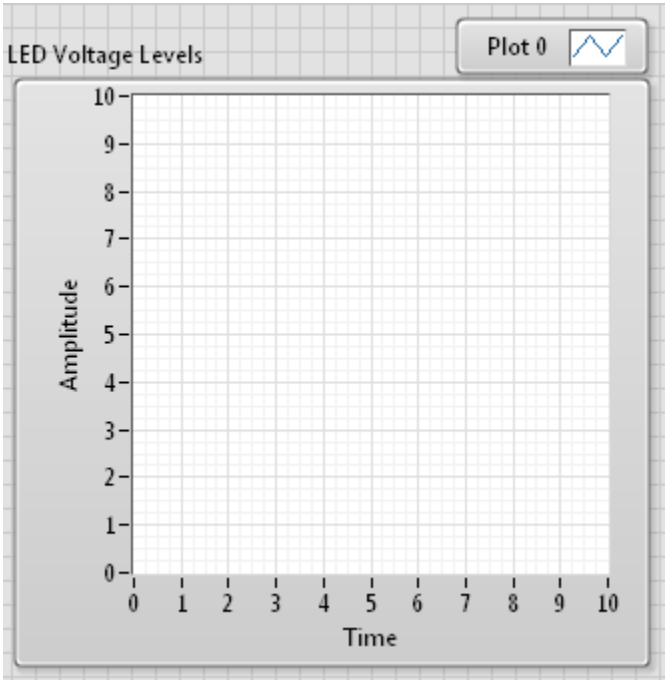


Tip: You can make copies of front panel items using <Ctrl-C> and <Ctrl-V> or by selecting a front panel item and holding <Ctrl> while dragging it.

8. Add a **XY Graph** to display the Voltage Sweep.
    a. Right-click on the front panel.
    b. Navigate to **Silver»Graph»XY Graph**.
    c. Drag and drop the **XY Graph** onto the front panel**.**



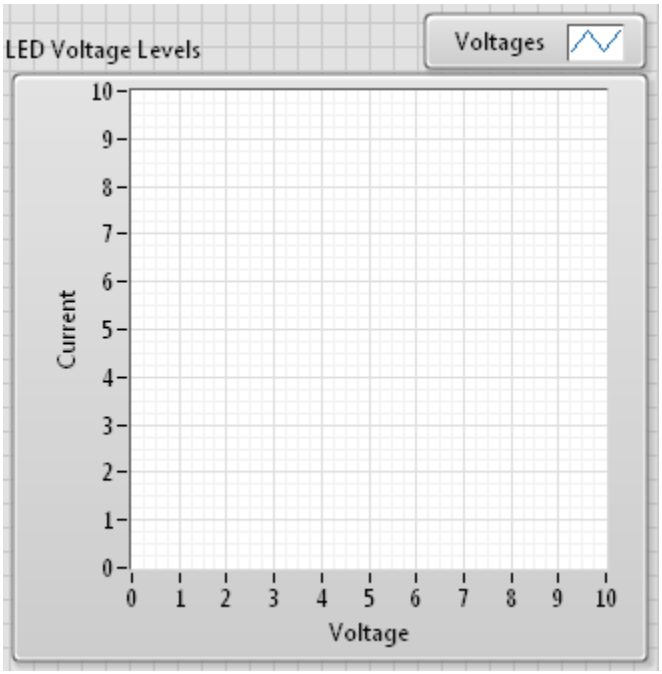
9. Modify the **XY Graph** as follows:
    a. Rename the **XY Graph "LED Voltage Levels"**



b. Rename the **X-Axis "Voltage"**

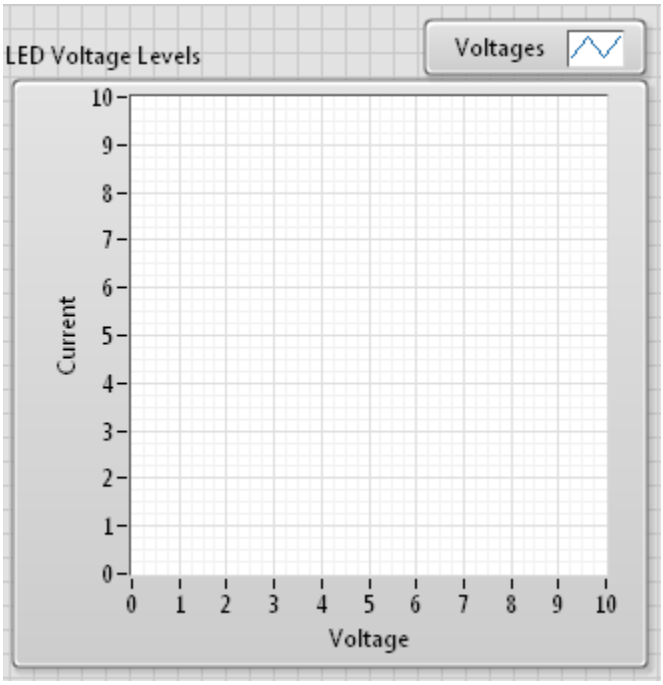
c. Rename the **Y-Axis "Current"**

d. Rename **Plot 0** in the Legend **"Voltages"**

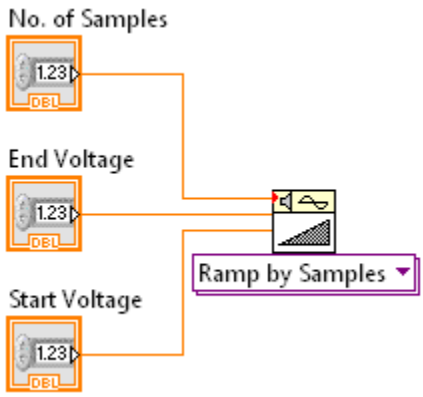


e. Change the scale of the **X-Axis** to **[0, 5]**

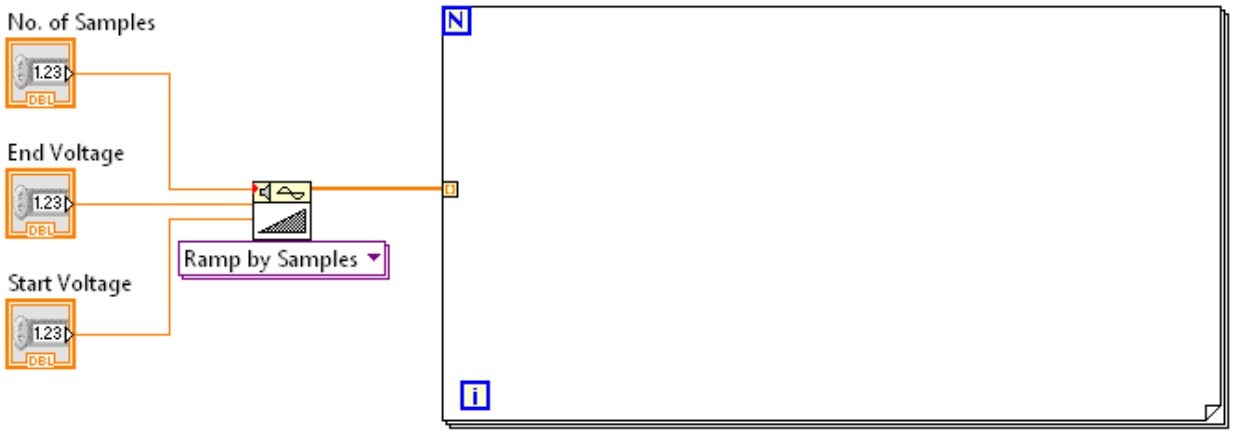
f. Change the **X-Axis** to **not auto scale** by right-click on the **X-Axis** and deselecting **AutoScale X**



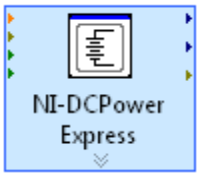
10. Toggle to the block diagram using <Ctrl-E>.
11. Add a **Ramp Pattern** function to the block diagram.
    a. Right-click the block diagram.
    b. Navigate to **Signal Processing»Sig Generation»Ramp Pattern**.
    c. Drag and drop the Ramp Pattern function onto the block diagram.
12. Wire the following controls into the Ramp Pattern function.
    a. Wire the **No. of Samples** control into the **samples** input terminal of the **Ramp Pattern** function.
    b. Wire the **Start Frequency** control into the **start** input terminal of the **Ramp Pattern** function.
    c. Wire the **End Frequency** control into the **end** input terminal of the **Ramp Pattern** function.



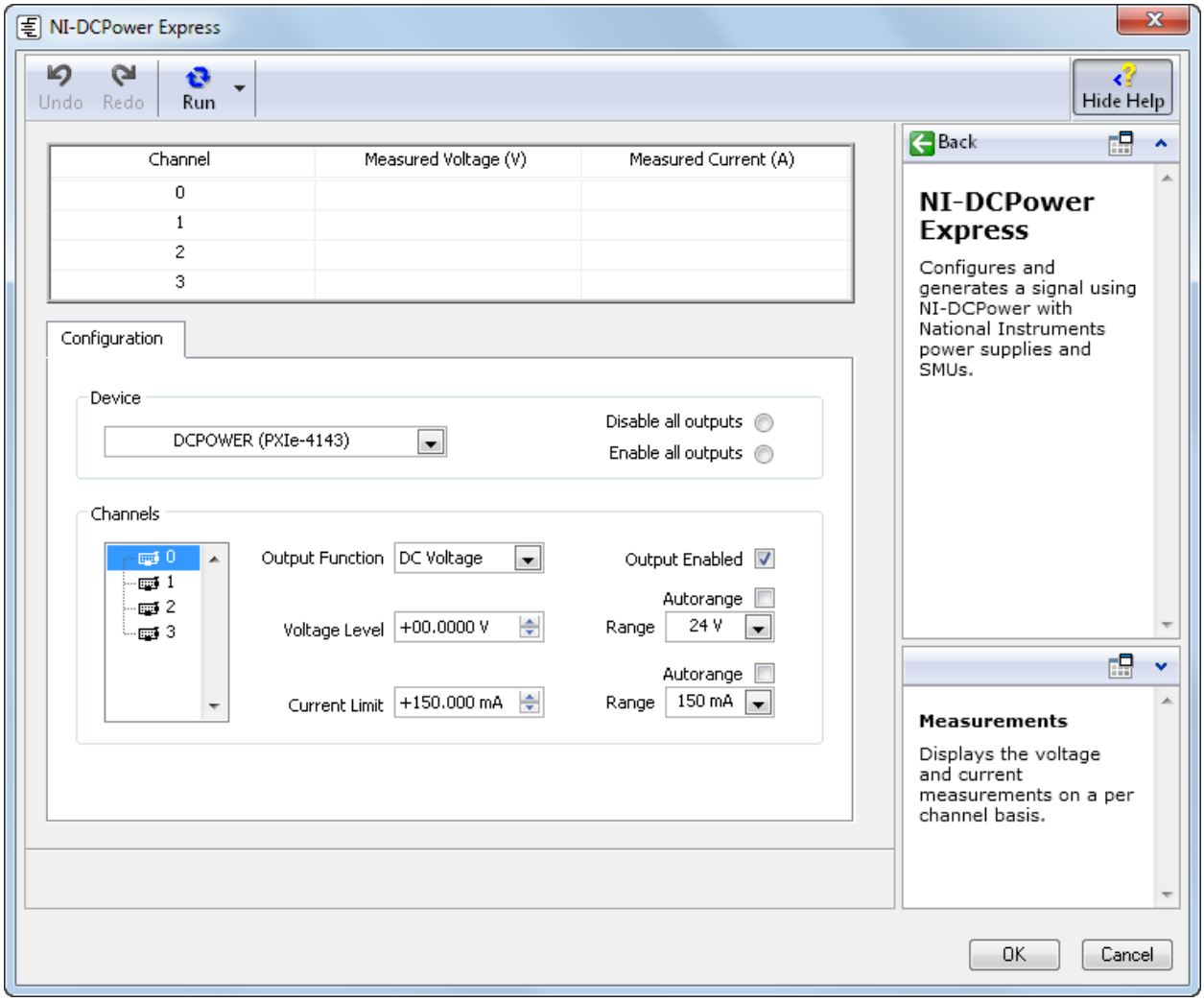
13. Add a **For Loop** to the block diagram.
    a. Right-click the block diagram.
    b. Navigate to **Programming»Structures»For Loop**.
    c. Drag and draw the For Loop on to the block diagram.
14. Wire the **Ramp Pattern** output terminal of the **Ramp Pattern** function to the side of the **For Loop**.



15. Add a **DCPower Express VI** inside the For Loop.
    a. Right-click on the block diagram.
    b. Navigate to **Measurement IO»NI-DCPower»NI-DCPower Express**.
    c. Drag and drop the ***DCPower Express VI*** inside the For Loop.

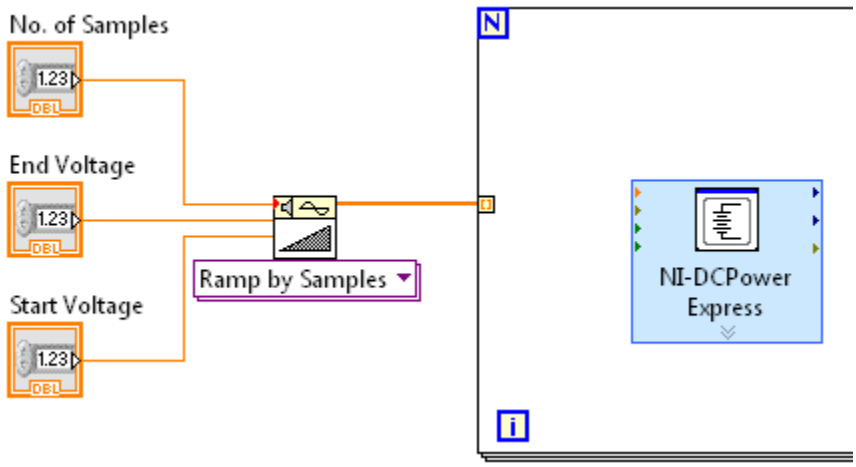


16. In the DCPower Express configuration window,
    a. Select **Channel 0**.
    b. Check the **Output Enabled** checkbox.

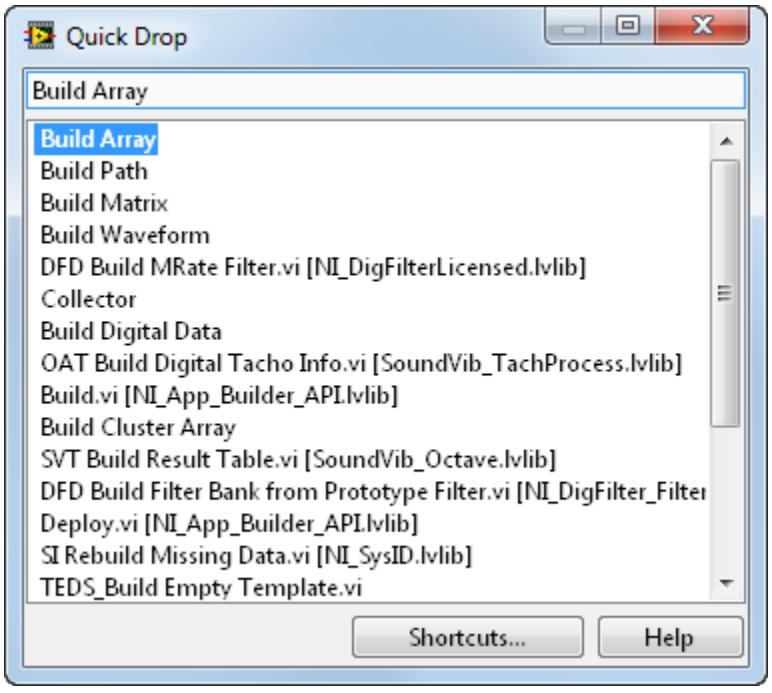


    c. Click **OK**.

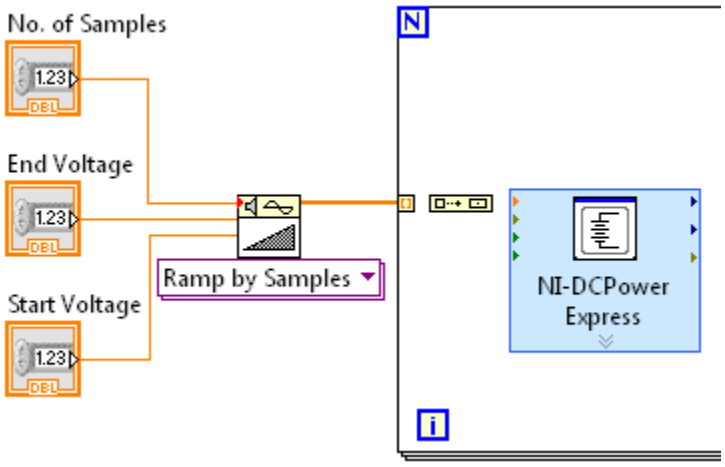
17. Your code should look similar to the following.



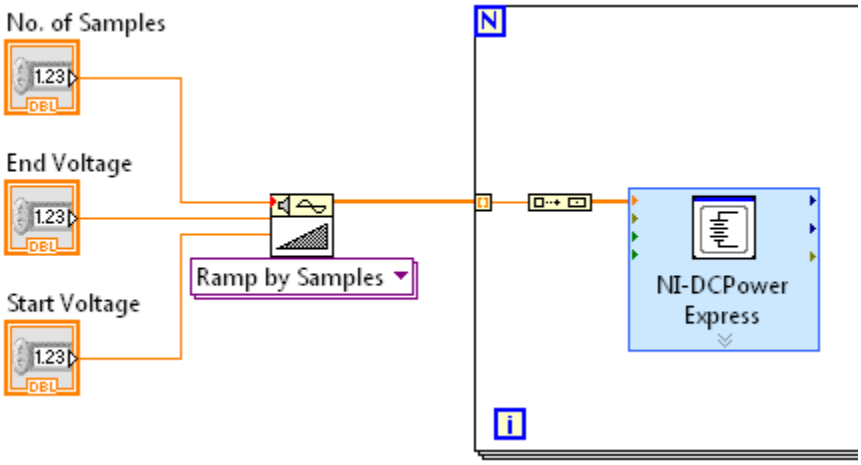
18. Add a **Build Array** function to the For Loop.
    a. Use <Ctrl-Space> to bring up the Functions Quick Drop window.
    b. Type `Build Array` into the Quick Drop search window.
    c. Double-click **Build Array**.



d. Drop the Build Array function inside the For Loop.

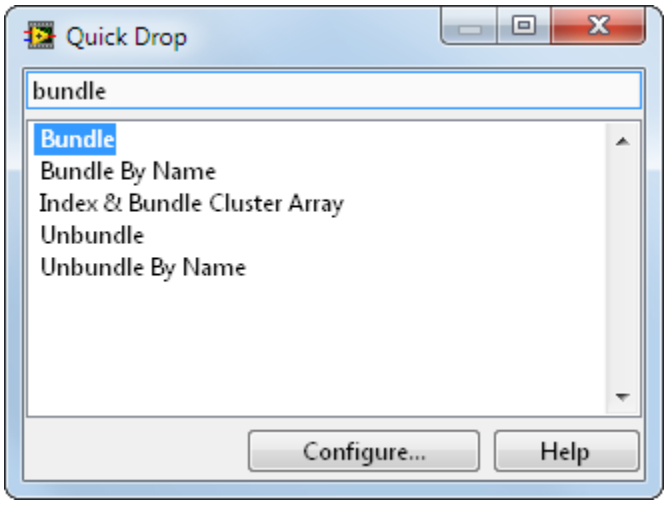


19. Wire the **Ramp Pattern Indexed Array** to the **Build Array** function.
20. Wire the **Build Array appended array** output terminal to the **voltage/current level** input terminal of the **DCPower Express VI**.



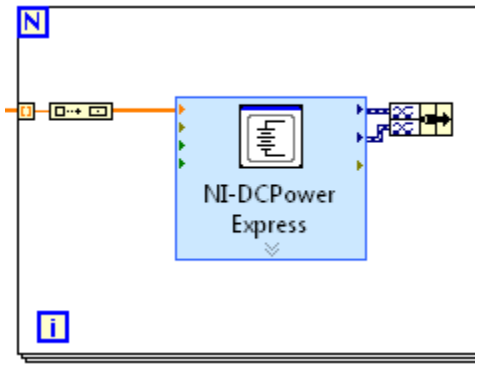
21. Add a Bundle function to the For Loop
    a. Use **CTRL + SPACE** to bring up the **Functions Quick Drop window**
    b. Type **"Bundle"** into the **Quick Drop search window**

c. Double-click **Bundle**

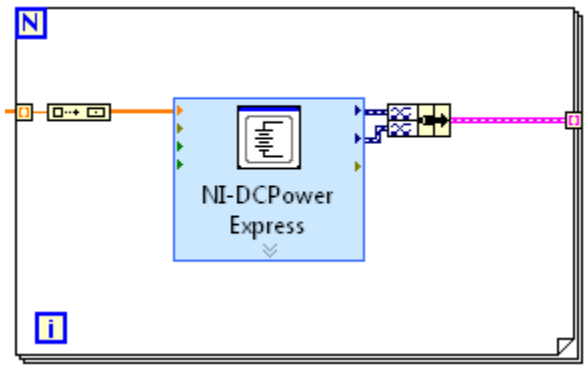


d. Drop the **Bundle** function inside the **For Loop**

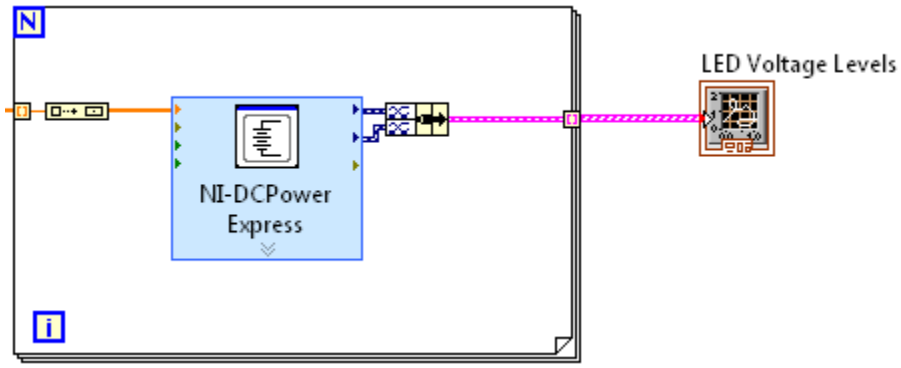
22. Wire the **measured voltage** output terminal of the **DCPower Express VI** to the top input terminal of the Bundle function
23. Wire the **measured current** output terminal of the **DCPower Express VI** to the bottom input terminal of the Bundle function



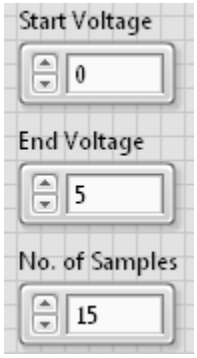
24. Wire the **output cluster** from the **Bundle** function to the right side of the **For Loop**



25. Wire the **output cluster** from the right side of the Fop Loop to the **LED Voltage Levels** XY Graph

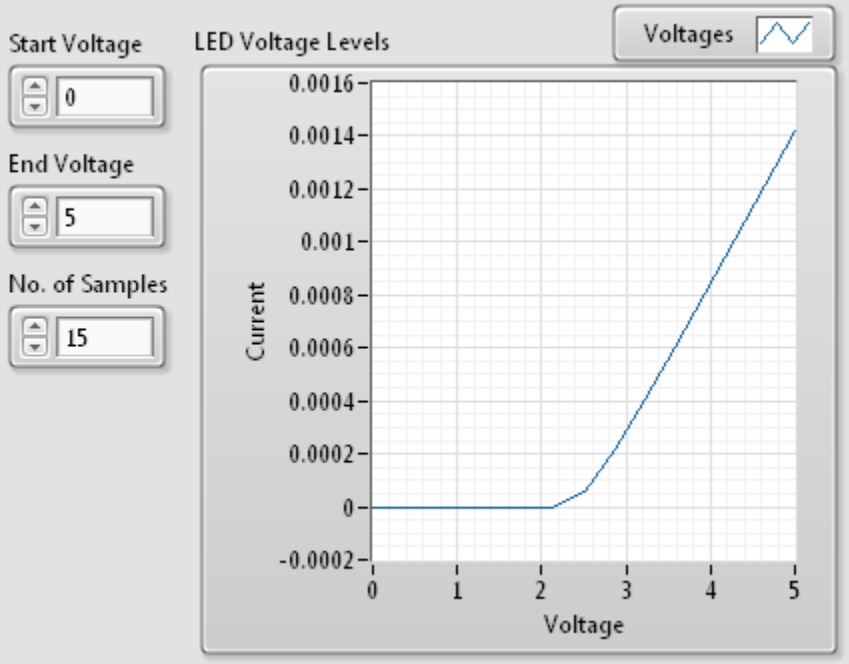


26. Toggle to the front panel using <Ctrl-E>.
27. On the front panel, set the following values.
    a. **Start Voltage = 0**
    b. **End Voltage = 5**
    c. **No. of Samples = 15**



28. **Run** the application and observe that the LED on the Socket 0 UUT increases in light intensity after the voltage exceeds 2.5 V.

    Warning! Do not supply more than 5 V.



29. **Save** and **Close** the application.

EXERCISE 6.4 – POWER SUPPLY VOLTAGE SWEEP WITH STANDARD VIS

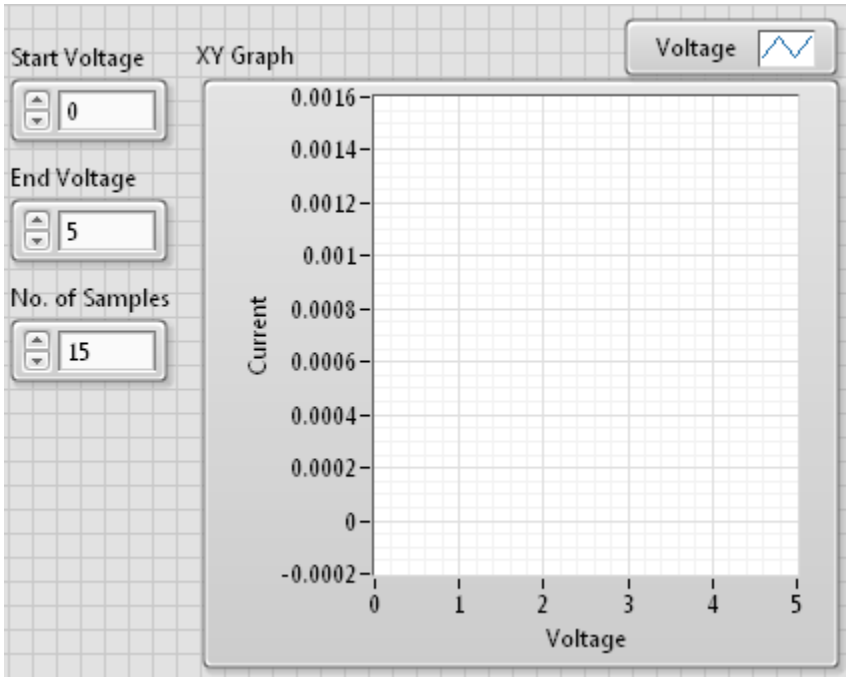
Goal

The goal of this exercise is to open and run a pre-built example that shows how you could implement a Power Supply Voltage Sweep using Standard VIs, rather than Express VIs.

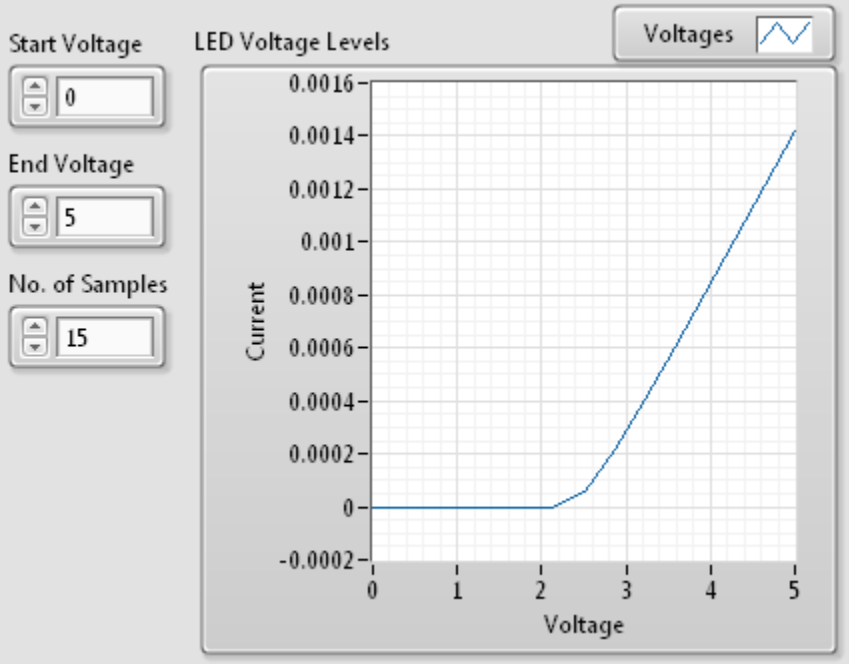
Steps

1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browsing to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the LabVIEW for Instrumentation Hands-On.lvprj file.
2. In the Power Supply Test folder, open the Power Supply Voltage Sweep – Standard VIs.
3. Open the block diagram and observe that the source code is implemented using Standard VIs from the DCPower instrument driver.

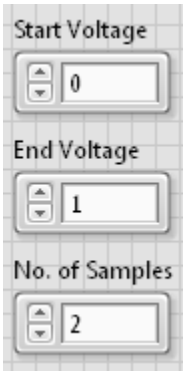
4. On the front panel, set the following parameters.
    a. **Start Voltage = 0**
    b. **End Voltage = 5**
    c. **No. of Samples = 15**



5. **Run** the application and observe that the LED on the Socket 0 UUT increases in light intensity after the voltage exceeds 2.5 V**.**



6. Set the following parameters on the front panel.
    a. **Start Voltage = 0**
    b. **End Voltage = 1**
    c. **No. of Samples = 1**



7. **Run** the application once more.

    Note: This will power the LED off.

8. **Close** the application and *do not save*.

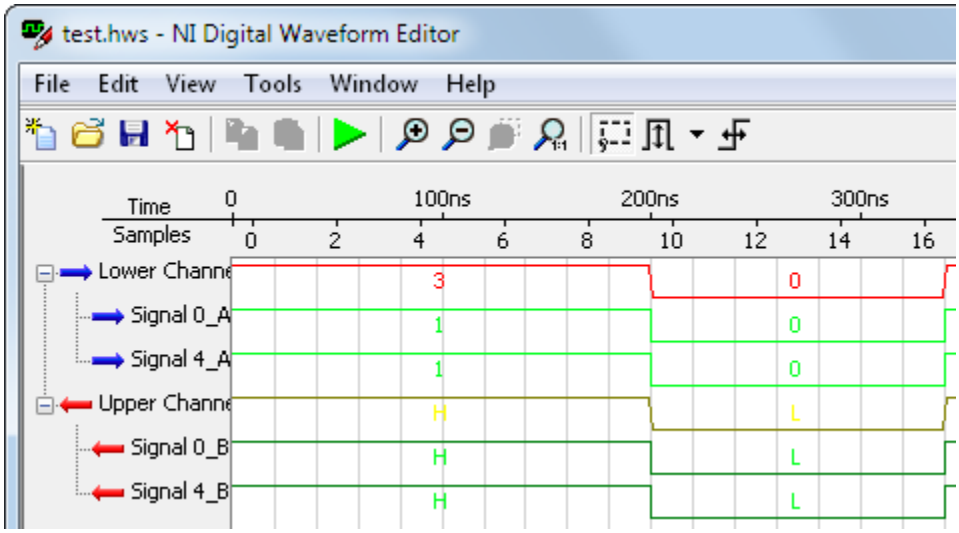
EXERCISE 7.1 – DIGITAL WAVEFORM EDITOR

Goal

The goal of this exercise is to open up and observe a digital waveform file that will be used in the HSDIO exercises.

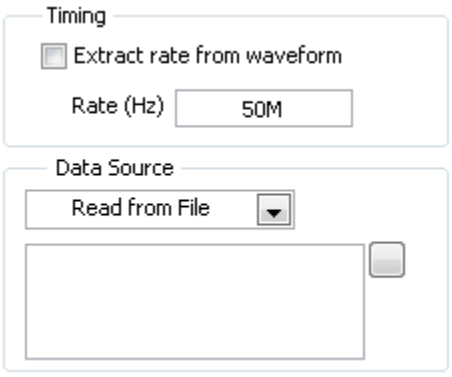
Steps

1. Open the Digital Waveform Editor by browsing to **Start»All Programs»National Instruments»Digital Waveform Editor»Digital Waveform Editor**.
2. Open the digital waveform file.
    a. Select **File»Open Waveform**.
    b. Brose to `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Bit Error Rate Test`.
    c. Select the **test.hws** waveform file.
    d. Click **OK**.
3. In this waveform file you will see two digital signals that will be applied to the lower channel and two digital signals that are expected by the upper channel.

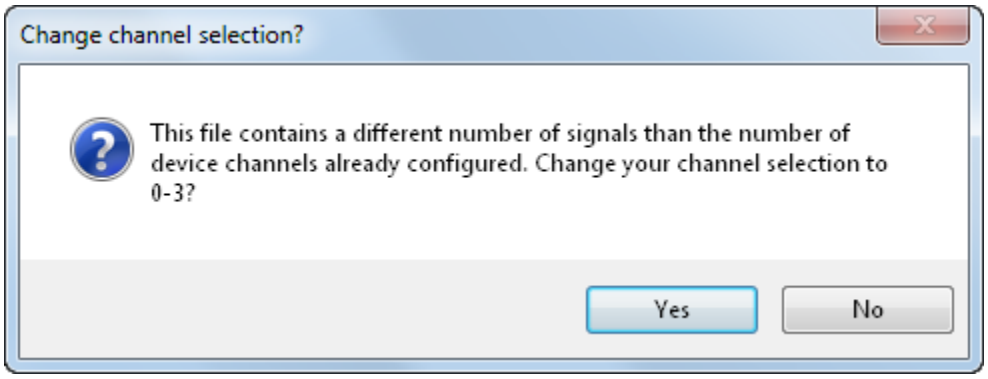


4. Keep the Digital Waveform Editor open**.**
5. Open a LabVIEW blank VI by selecting **File»New VI** from the Project Explorer**.**
6. Add a HSDIO Express (Stimulus & Response) VI to the block diagram.
    a. Right-click on the block diagram.
    b. Navigate to **Measurement I/O»NI-HSDIO»HSDIO Express (Stimulus & Response) VI.**
    c. Drag and Drop the HSDIO Express (Stimulus & Response) VI onto the block diagram.

7. Configure the HSDIO (Stimulus & Response) VI Configuration tab as follows.
    a. In the Data Source drop down, select **Read from File**.

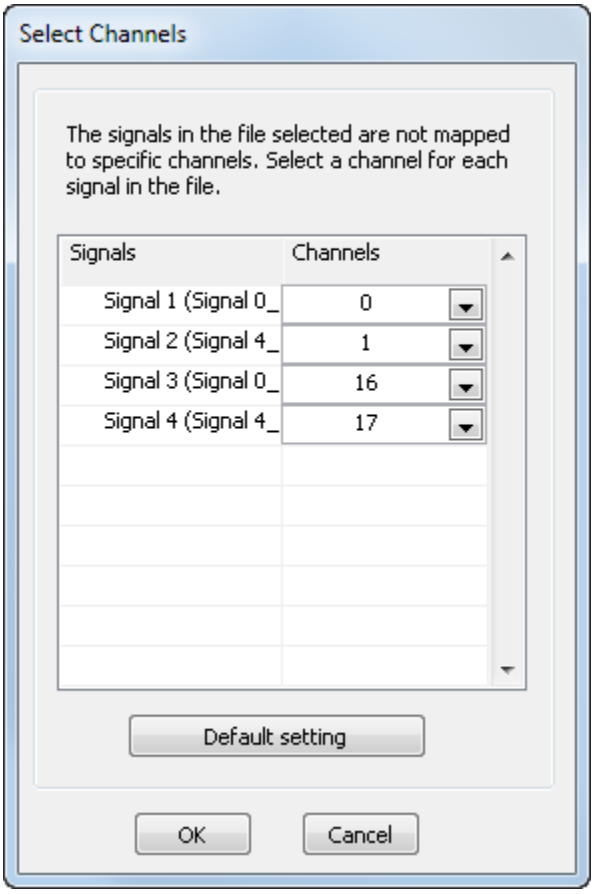


    b. Brose to `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Bit Error Rate Test`.
    c. Select the **test.hws** waveform file.
    d. Click **OK**.
    e. Click **Yes** in the Change channel Selection dialog box.



8. Select **Modify** to change the channels.

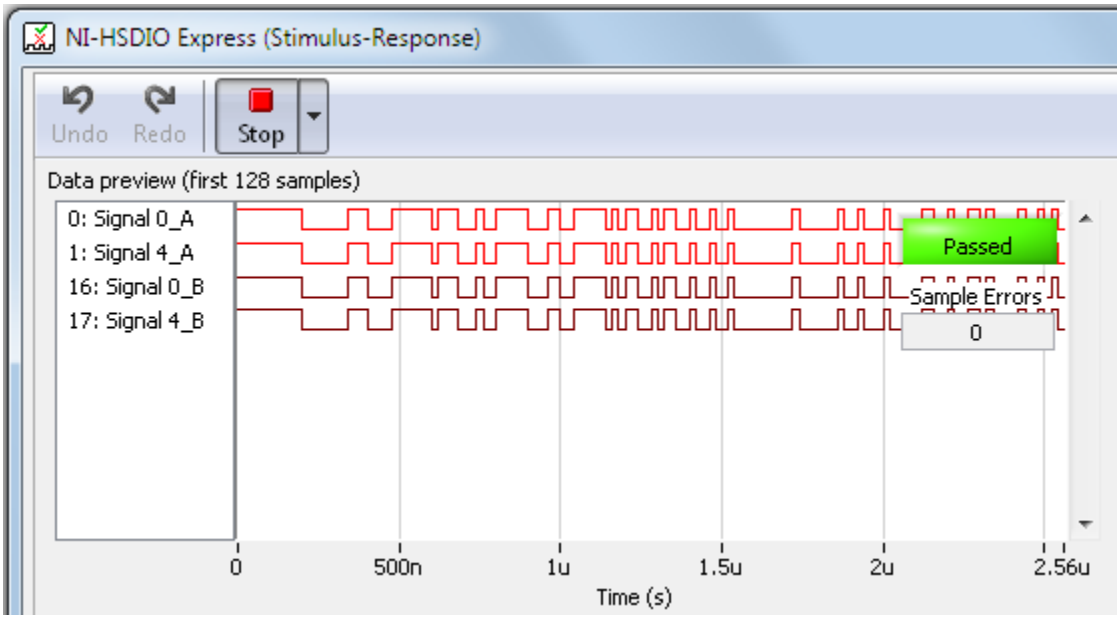
9. Specify the following channel connections.
    a. **Signal 1 = Channel 0**
    b. **Signal 2 = Channel 1**
    c. **Signal 3 = Channel 16**
    d. **Signal 4 = Channel17**



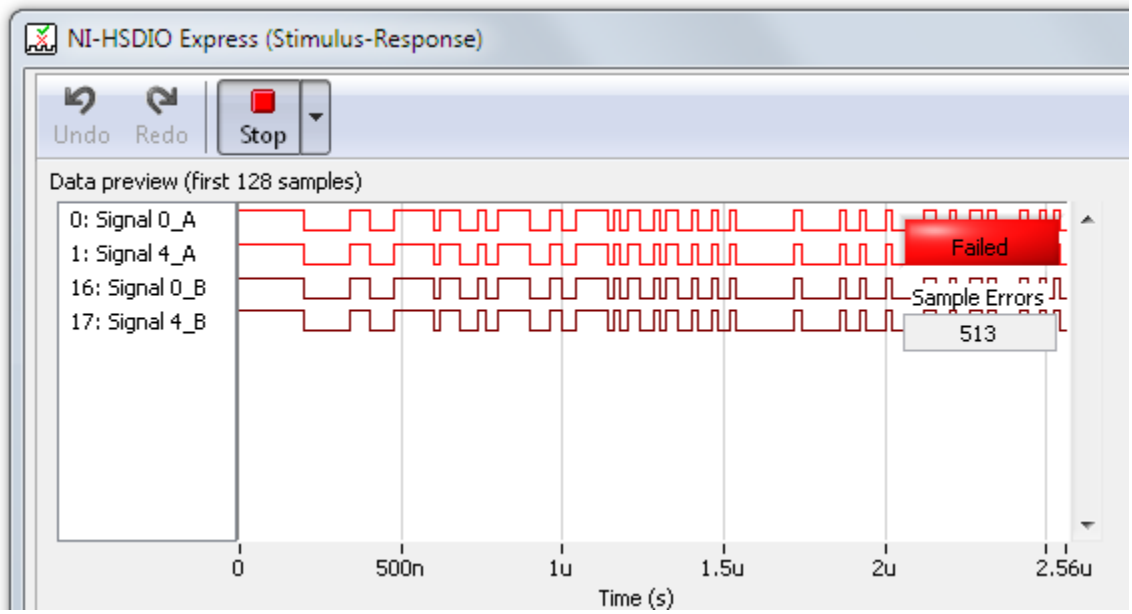
    e. Select **OK**.

10. On the Socket 0 UUT BERT, place the toggle switch in the Pass position.
11. Run a test of the digital signals by selecting the **Run** button in the HSDIO Express VI configuration window.

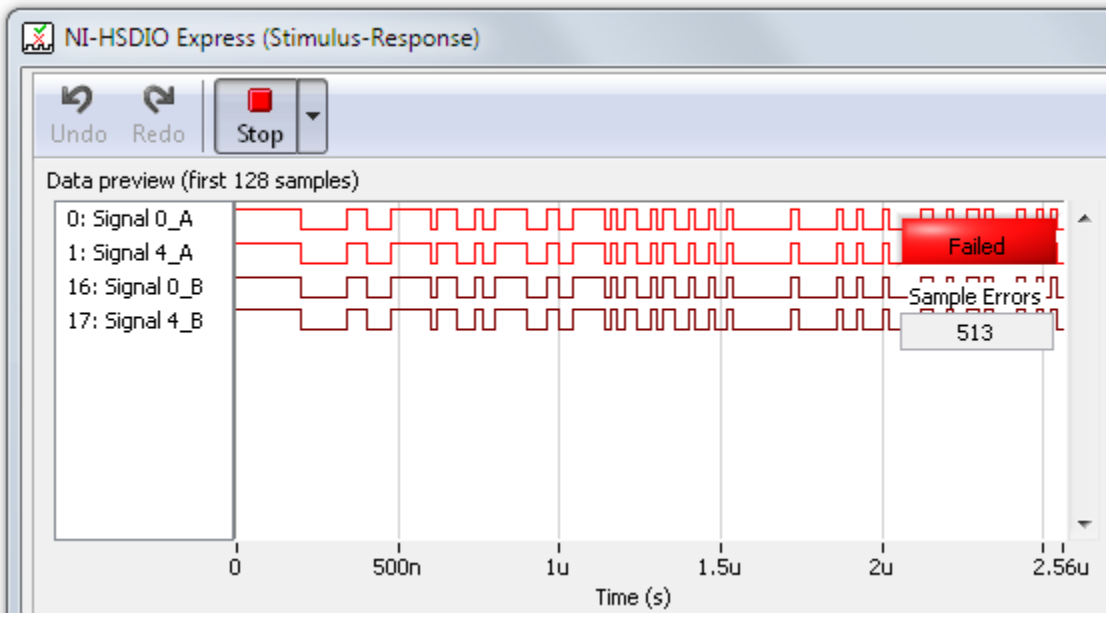
12. Observe that there are no errors and the status of the test is **Passed**.



13. While the test is running, press the small push button on the Socket 0 UUT to inject errors into the digital lines that are being generated and observe that the status of the test will flash Failed.

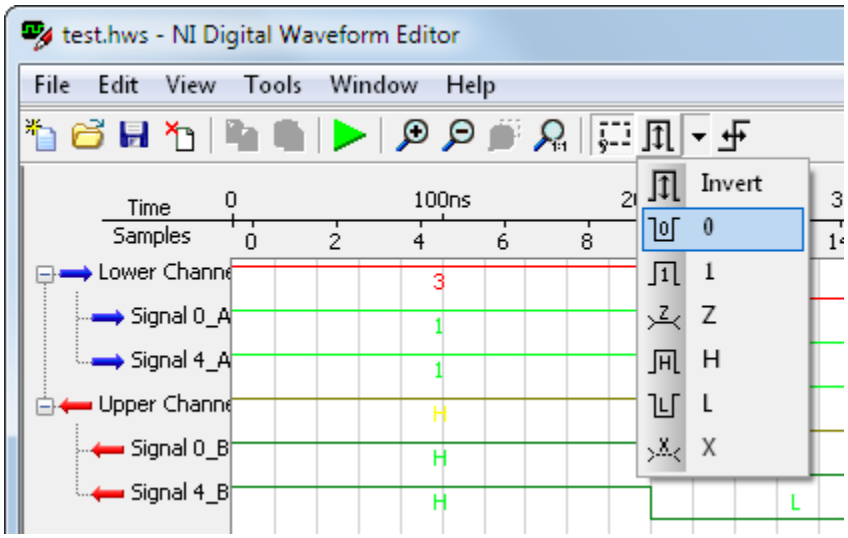


14. Now place the Socket 0 UUT BERT toggle switch in the Fail position to disconnect the line and observe that the status of the test will hold a constant Failed.

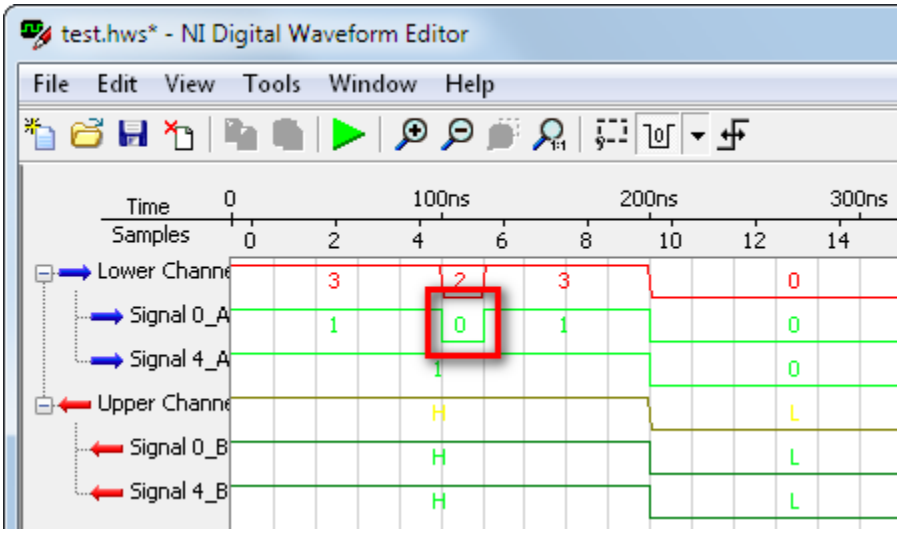


15. On the Socket 0 UUT BERT, place the toggle switch in the Pass position.
16. Select **Stop** to stop the test.

17. Return to the Digital Waveform Editor and change the waveform file to have an error.
    a. Click the Fill bits drop down.
    b. Select **0**.

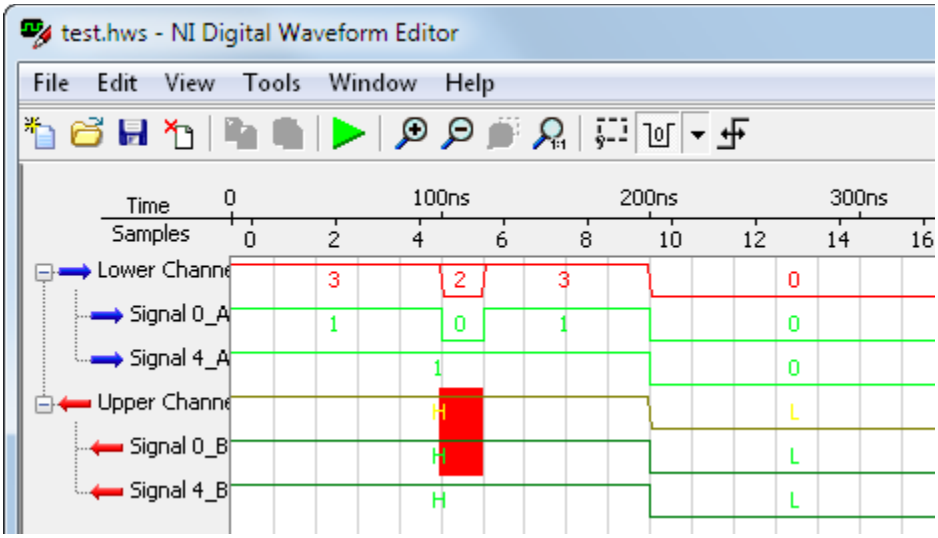


18. Click on the first instance the **1** in the Lower Channel Signal 0 to inject an error in the waveform**.**



19. Save the digital waveform file by selecting **File»Save** in the Digital waveform Editor.
20. Return to the HSDIO Express (Stimulus-Response) VI configuration window and select **Run** again.

21. Observe that the test fails and the failure points show up in the Digital Waveform Editor.



22. Select **Stop** in the HSDIO Express (Stimulus-Response) VI configuration window.
23. Select **OK**.
24. **Close** the blank VI and select **Don't Save**.
25. **Close** the Digital Waveform Editor.

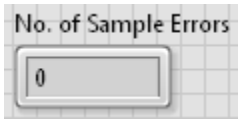
EXERCISE 7.2 – BIT ERROR RATE TEST

Goal

The goal of this exercise is to create a LabVIEW application that will use the HSDIO to supply two know digital signals and checks to see if the returned signals from the UUT match the expected waveform.

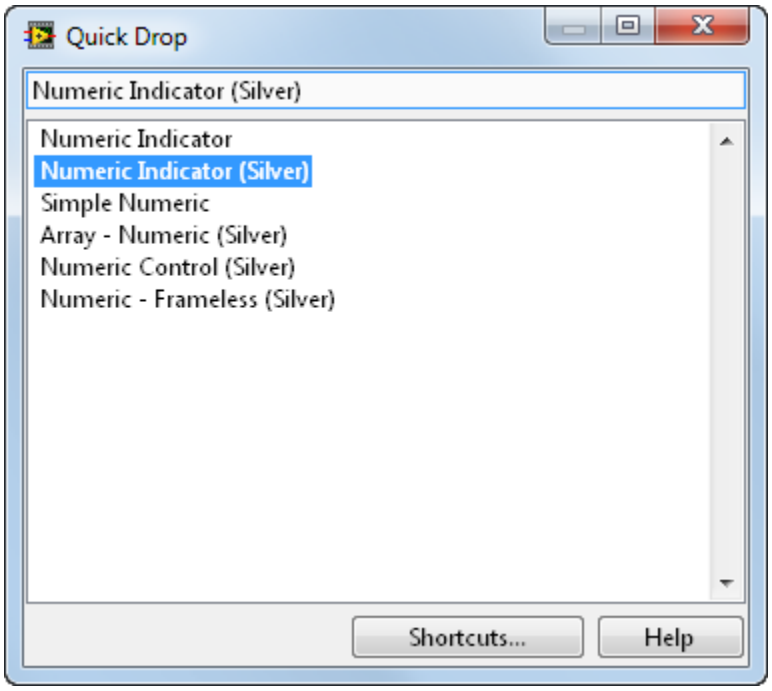
Steps

1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browsing to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the LabVIEW for Instrumentation Hands-On.lvprj file.
2. Create a new VI in the Bit Error Rate Test folder by right–clicking on the folder and selecting **New»VI**.
3. Select **File»Save** and name the VI **Bit Error Rate Test**.
4. Tile the front panel and block diagram using <Ctrl-T>.
5. Add a **Numeric Indicator** to the front panel to display the number of sample errors.
    a. Right-click on the front panel.
    b. Navigate to **Silver»Numeric»Numeric Indicator**.
    c. Drag and drop the Numeric Indicator onto the front panel.
    d. Name the Numeric Indicator **No. of Sample Errors**.

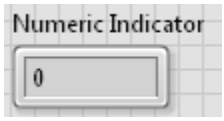


6. Add a **Numeric Indicator** to the front panel to display the bit error rate percentage.
    a. Use <Ctrl-Space> to bring up the Controls Quick Drop window.
    b. Type `Numeric Indicator` into the Quick Drop search window.

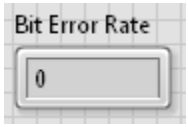
c. Double-click **Numeric Indicator (Silver)**.



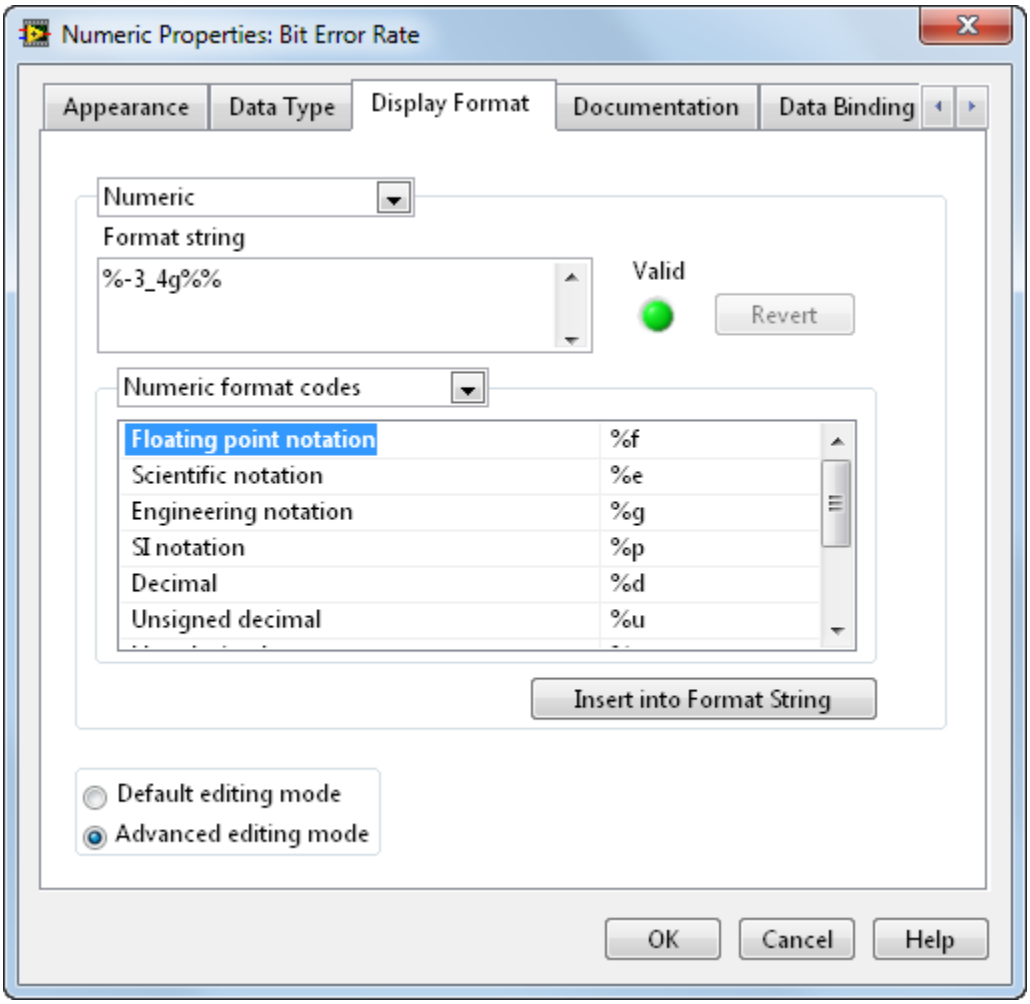
d. Drop the Numeric Indicator on to the front panel.



e. Name the Numeric Indicator **Bit Error Rate**.

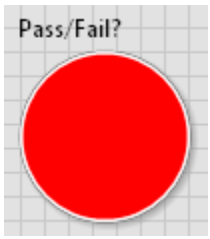


7. Change the representation of the Bit Error Rate Numeric Indicator to show a percentage.
    a. Right-click the Bit Error Rate Numeric Indicator and select **Properties**.
    b. Select the **Display Format** tab.
    c. Check the **Advanced editing mode** radio button.
    d. Enter `%-3_4g%%` into the Format string box.

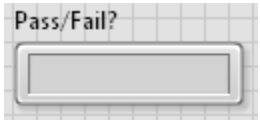


    e. Click **OK**.
8. Add an LED to the front panel to display a pass/fail LED for the test.
    a. Right-click on the front panel.
    b. Navigate to **Silver»Boolean»LED**.
    c. Drag and drop the LED on to the front panel.
    d. Name the LED **Pass/Fail?**.

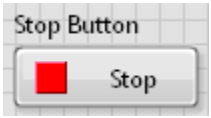
9. Change the Off color of the LED to **Red**.
    a. Right-click the LED and select **Properties**.
    b. Change the Off color to **Red**.
    c. Select **OK**.



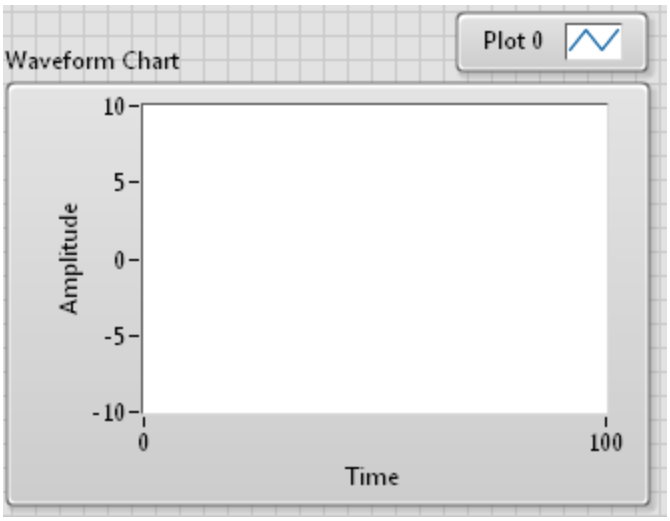
10. Add a **String Indicator** to the front panel to display a pass/fail string for the test.
    a. Right-click on the front panel.
    b. Navigate to **Silver»String»String Indicator**.
    c. Drag and drop the String Indicator onto the front panel.
    d. Name the String Indicator **Pass/Fail?**.



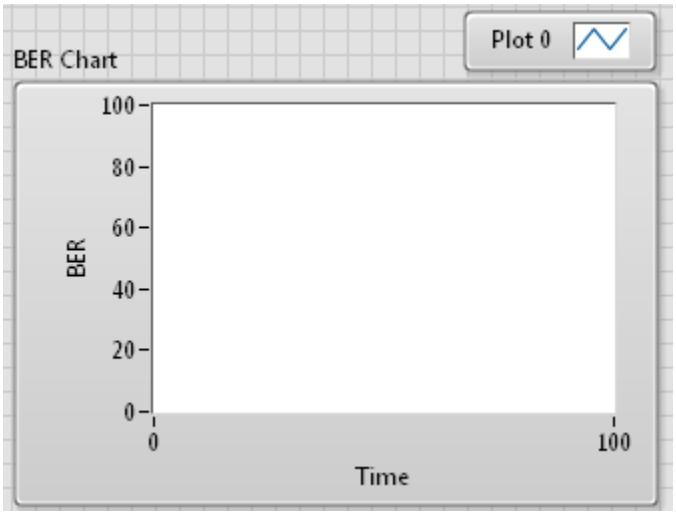
11. Add a **Stop Button** to the Front Panel
    a. Right-click on the **Front Panel**
    b. Navigate to **Silver»Boolean»Stop Button (Silver)**
    c. Drag and drop the **Stop Button** on to the **Front Panel**



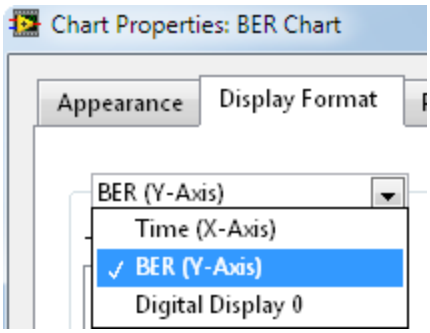
12. Add a **Waveform Chart** to the Front Panel
    a. Right-click on the **Front Panel**
    b. Navigate to **Silver»Graph»Waveform Chart (Silver)**
    c. Drag and drop the **Waveform Chart** on to the **Front Panel**



13. Modify the **Waveform Chart** as follows
    a. Name the **Waveform Chart "BER Chart"**
    b. Change the **Y-Axis** name to **"BER"**
    c. Change the **Y-Axis** to not auto scale by right-clicking on the **Y-Axis** and deselecting **AutoScale Y**
    d. Change the **scale** of the **Y-Axis** to **[0, 100]**



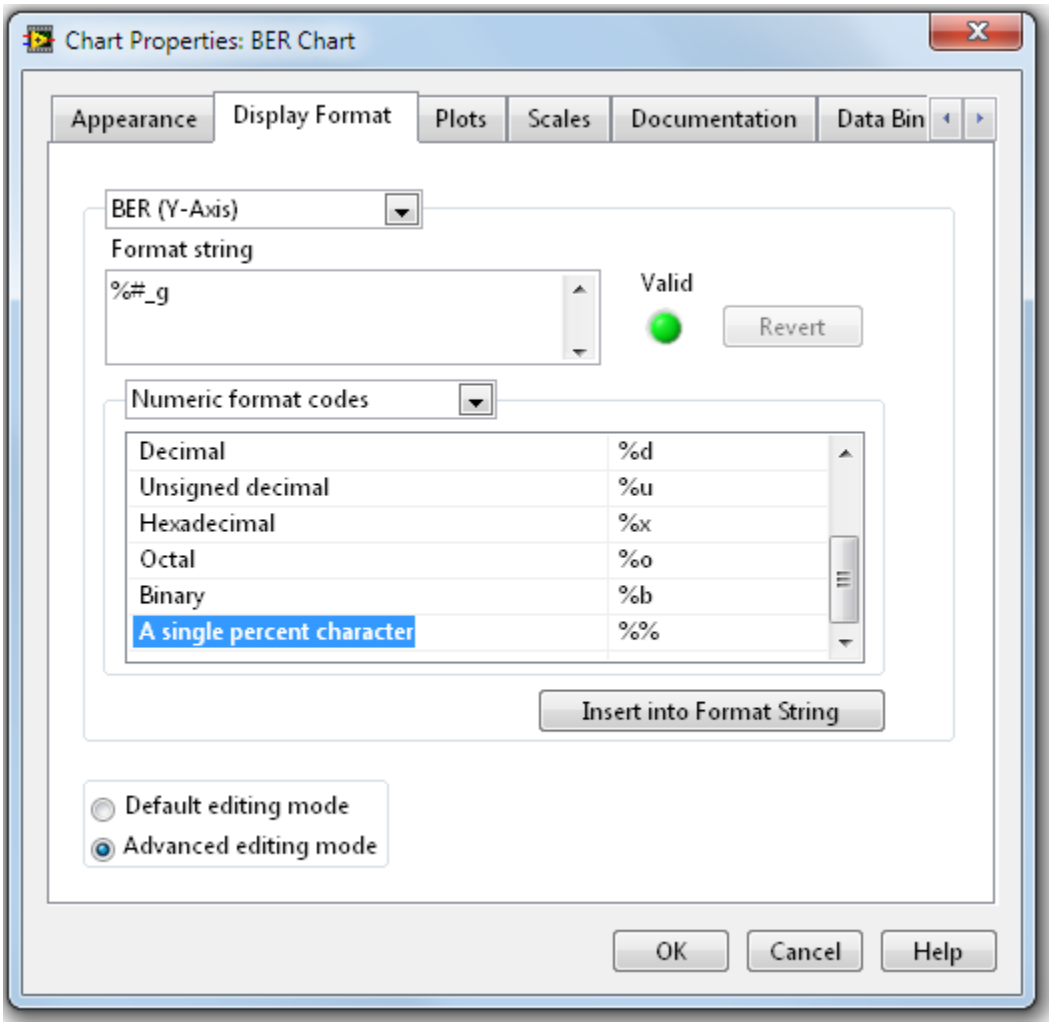
    e. Change the representation of the **Y-Axis** to **percentage**
        i. Right-clicking on the **Y-Axis** and select **Properties**
        ii. Select the **Display Format** tab
        iii. Select **BER (Y-Axis)** from the drop down



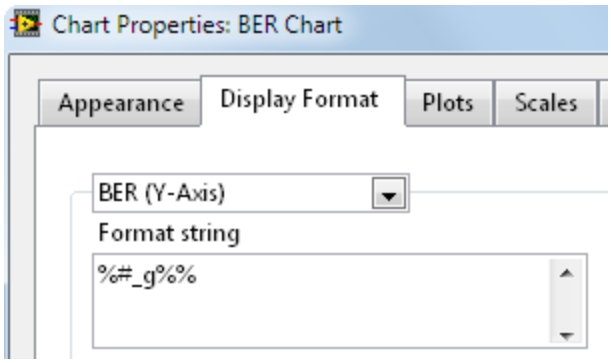
iv. Click the **Advanced editing mode** radio button

v. Place the cursor after the "g" in the Format string window.

vi. Select **A single percent character** in the **Numeric format codes** window

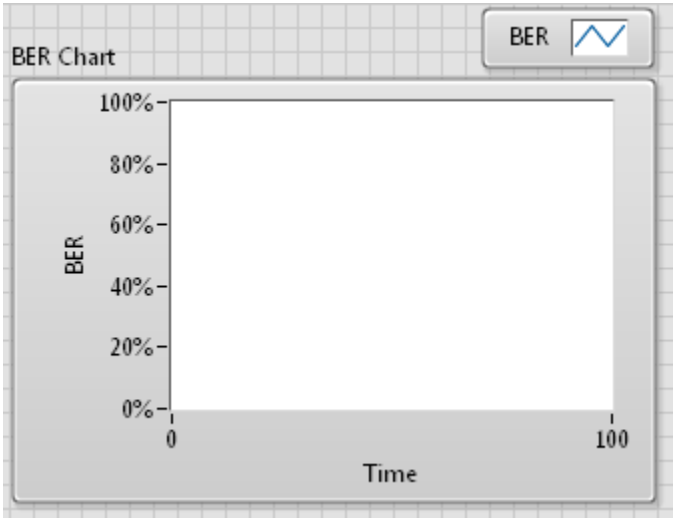


vii. Click **Insert into Format String** - the notation should be **"%#_g%%"**

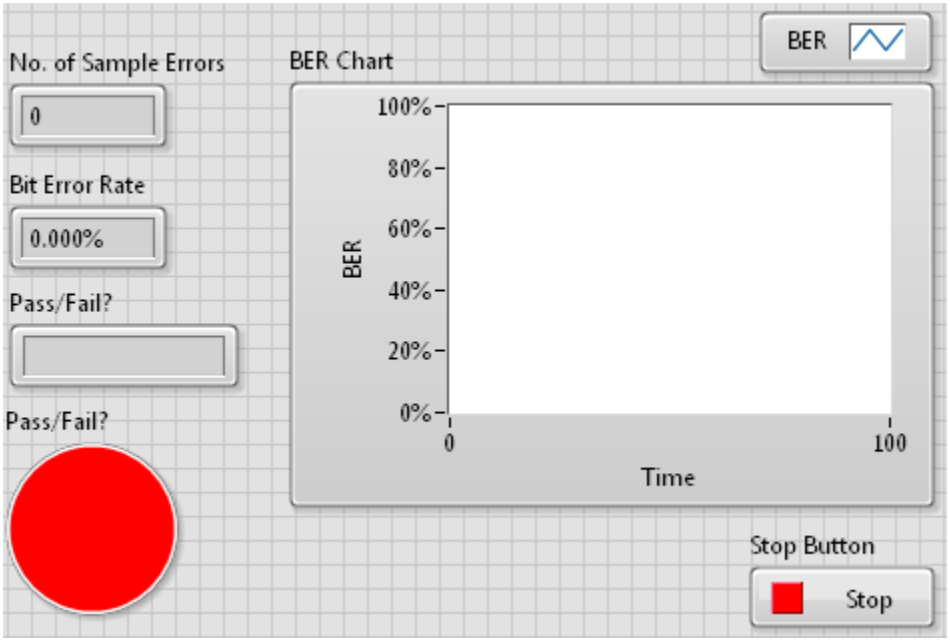


f. Click OK.

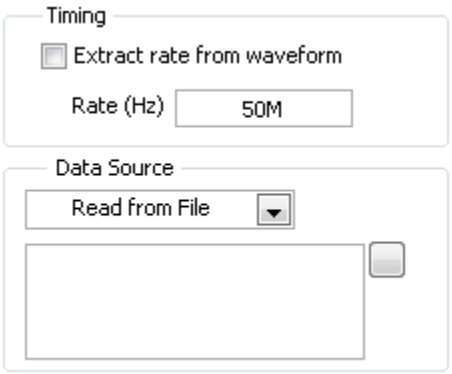
g. Change **Plot 0** in the **Legend** to **"BER"**



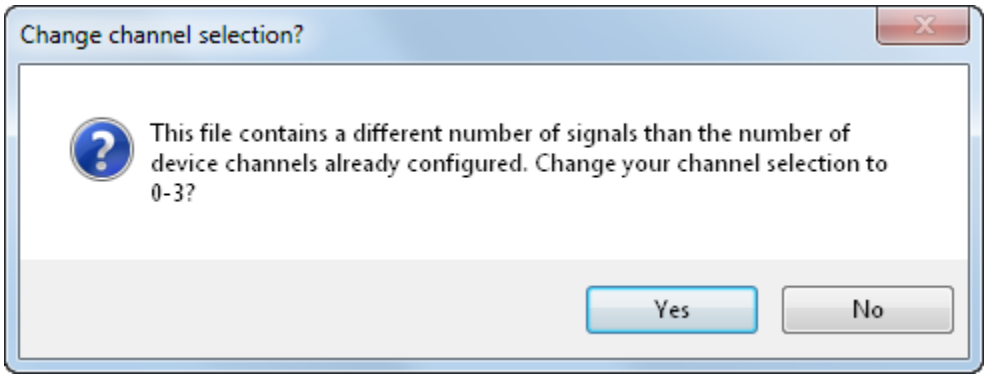
14. Your front panel should look similar to the following user interface.



15. Toggle to the block diagram using <Ctrl-E>.
16. Add a **HSDIO Express (Stimulus & Response) VI** to the block diagram.
    a. Right-click on the block diagram.
    b. Navigate to **Measurement I/O»NI-HSDIO»HSDIO Express (Stimulus & Response) VI.**
    c. Drag and Drop the HSDIO Express (Stimulus & Response) VI onto the block diagram.
17. Configure the HSDIO (Stimulus & Response) VI Configuration tab as follows.
    a. In the Data Source drop down, select **Read from File** .

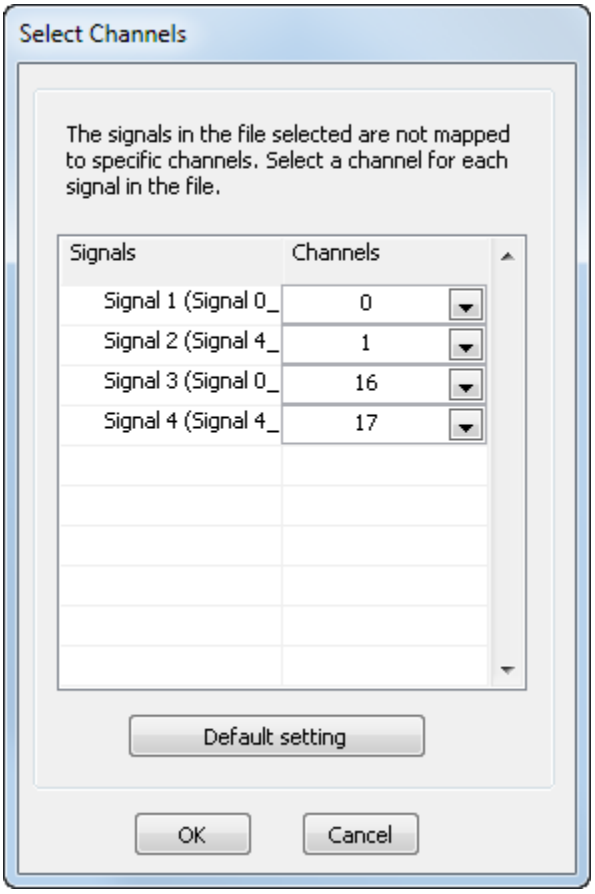


    b. Brose to `C:\Seminars\LabVIEW for Instrumentation Hands-On\Exercises\Bit Error Rate Test`.
    c. Select the **bidirectionalWfm_4bitPseudorandom.hws** waveform file.
    d. Click **OK**.
    e. Click **Yes** in the Change channel Selection dialog box.



18. Select **Modify** to change the channels.

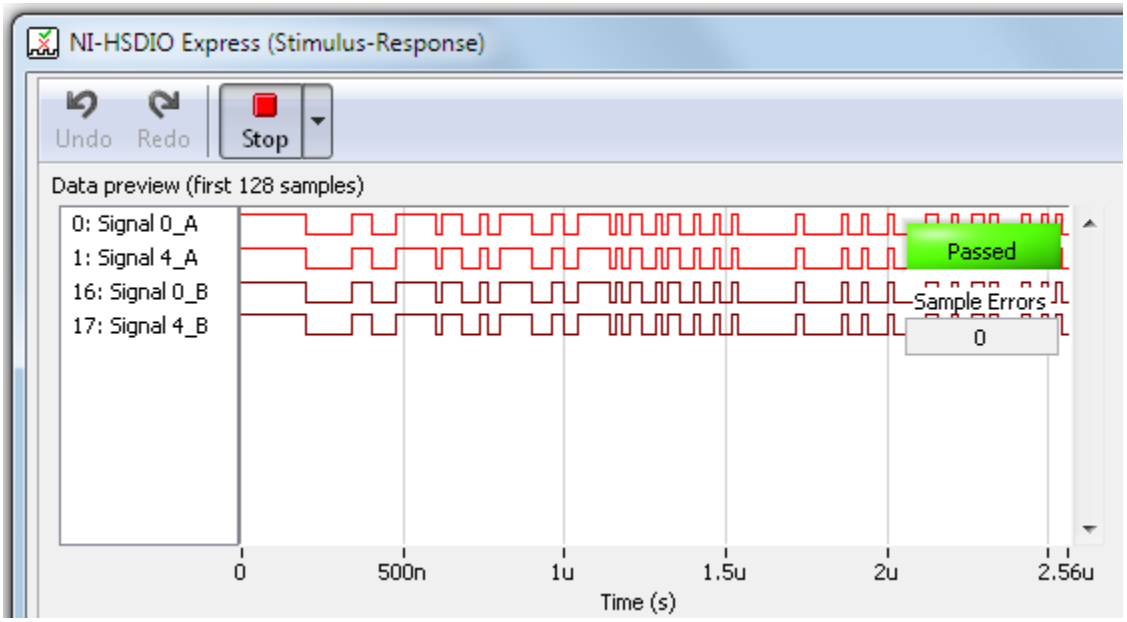
19. Specify the following channel connections.
    a. **Signal 1 = Channel 0**
    b. **Signal 2 = Channel 1**
    c. **Signal 3 = Channel 16**
    d. **Signal 4 = Channel17**



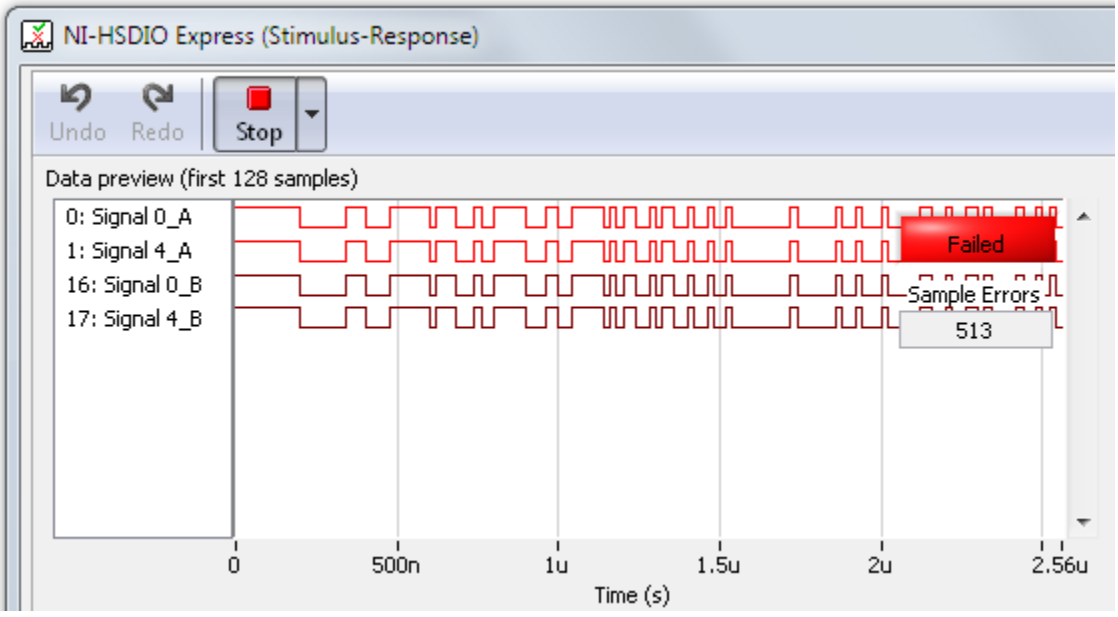
    e. Select **OK**.

20. On the Socket 0 UUT BERT, place the toggle switch in the Pass position.
21. Run test of the digital signals by selecting the **Run** button in the HSDIO Express VI configuration window.

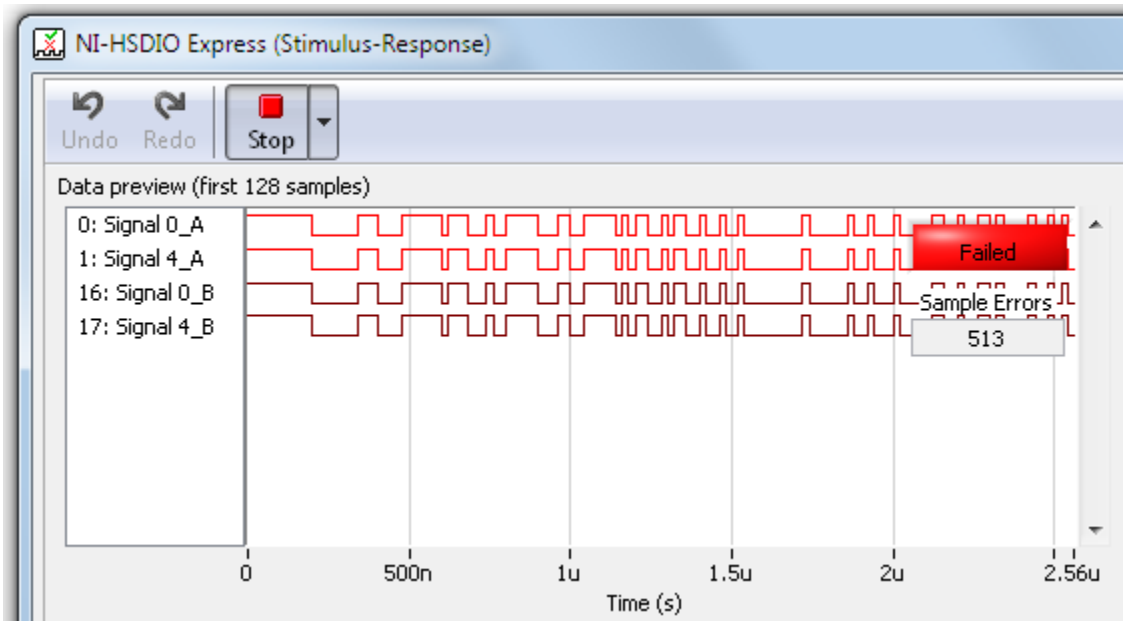
22. Observe that there are no errors and the status of the test is Passed.



23. While the test is running, press the small push button on the Socket 0 UUT BERT to inject errors into the digital lines that are being generated and observe that the status of the test will flash Failed.



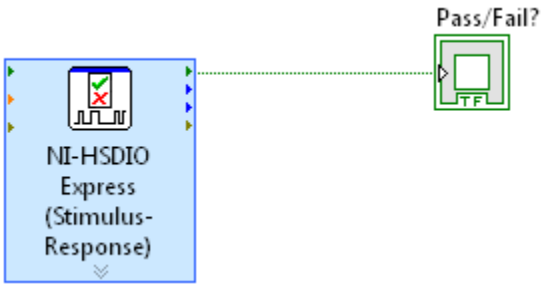
24. Now place the Socket 0 UUT BERT toggle switch in the Fail position to disconnect the line and observe that the status of the test will hold a constant Failed.



25. Select **Stop** to stop the test.

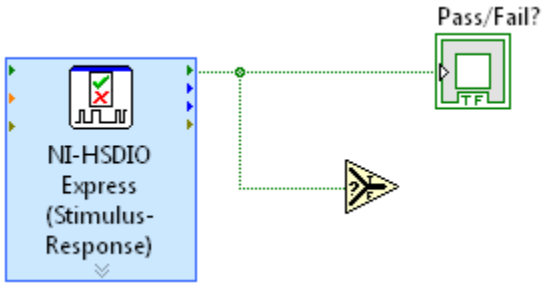
26. Select **OK** to close the configuration window.

27. Wire the **passed** output terminal of the **HSDIO Express (Stimulus & Response) VI** to the **Pass/Fail?** LED.

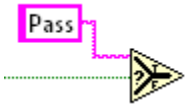


28. Add a **Select** function to the block diagram.
    a. Right-click the block diagram.
    b. Navigate to **Programming»Comparison**.
    c. Drag and drop the **Select** function on to the block diagram.

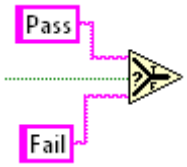
29. Wire the **passed?** output terminal of the **HSDIO Express VI** to the **s** input terminal of the **Select** function .



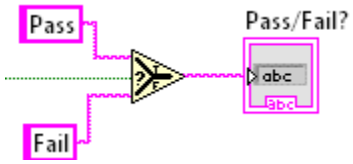
30. Add a **String Constant** to the block diagram.
    a. Right click the block diagram.
    b. Navigate to **Programming»String»String Constant**.
    c. Drag and drop a String Constant onto the block diagram.
    d. Type `Pass` into the String Constant.
    e. Wire the **Pass** String Constant to the **t** input terminal of the **Select** function.



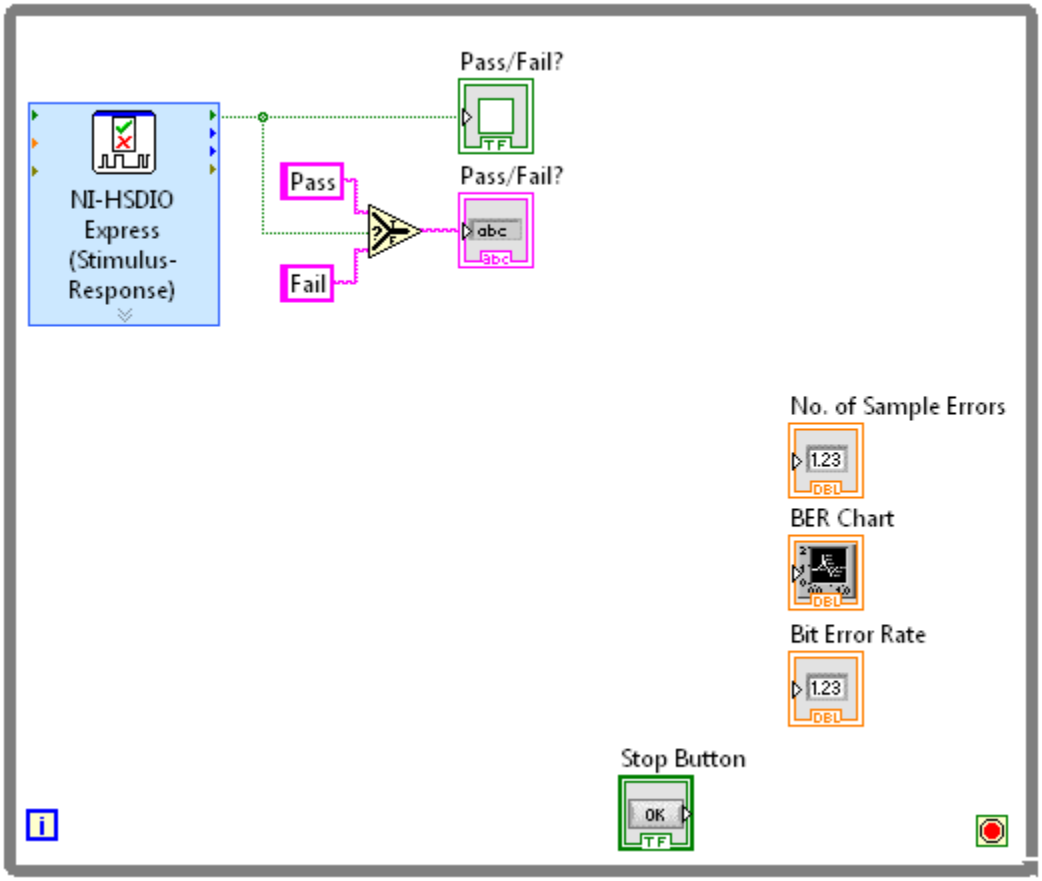
31. Make a copy of the String Constant you just added.
    a. Select the **String Constant** .
    b. Hold <Ctrl> while dragging the mouse.
    c. Change the value of the new String Constant to **Fail**.
    d. Wire the **Fail** String Constant to the **f** input terminal of the **Select** function.



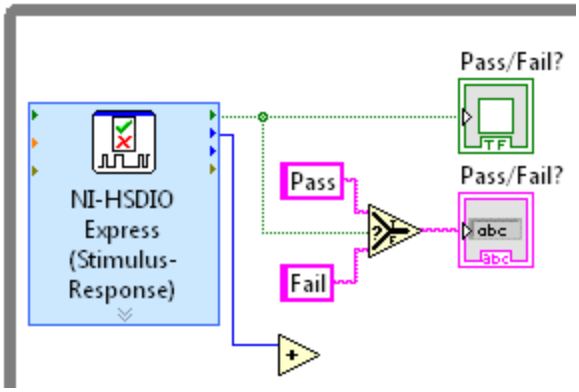
32. Wire the **s?t:f** output terminal of the **Select** function to the **Pass/Fail?** String Indicator.



33. Add a **While Loop** around all of your code
    a. Right-click on the **Block Diagram**
    b. Navigate to **Programming»Structures»While Loop**
    c. Draw a **While Loop** around all of your code



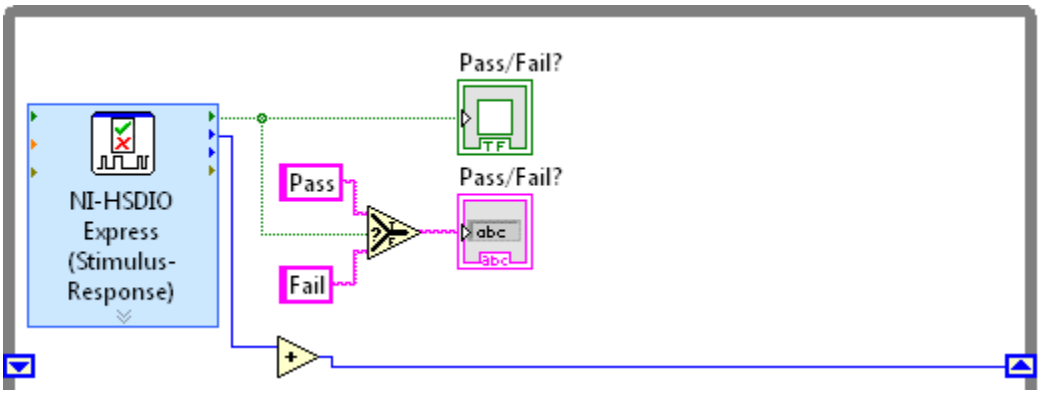
34. Add an **Add** function to the **While Loop**
    a. Right-click the **Block Diagram**
    b. Navigate to **Programming»Numeric»Add**
    c. Drag and drop the **Add** function to the **While Loop**
35. Wire the **number of sample errors** output terminal from the N**I-HSDIO Express (Stimulus-Response Express VI** to the **x** input terminal of the **Add** function



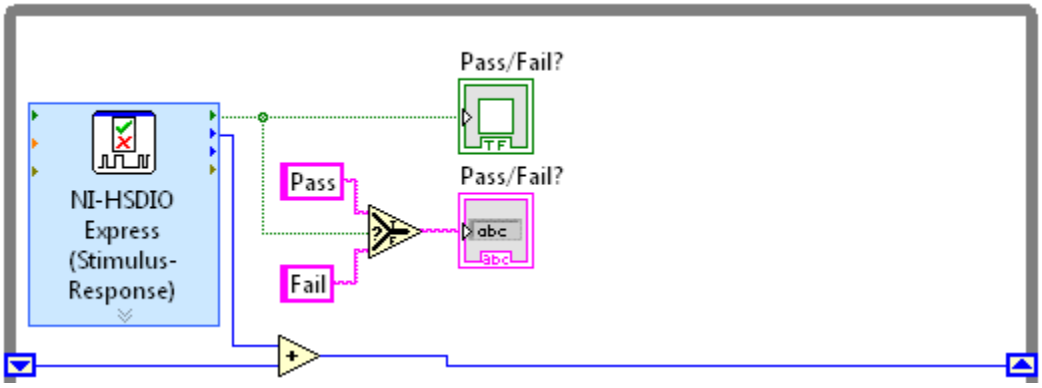
36. Wire the **x+y** output terminal of the **Add** function to the right side of the **For Loop**

37. Right-click on the terminal and select **Replace with Shift Register**

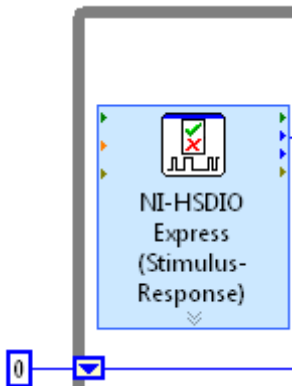
38. Click anywhere on the **left side** of the **For Loop** to provide a corresponding **input Shift Register**



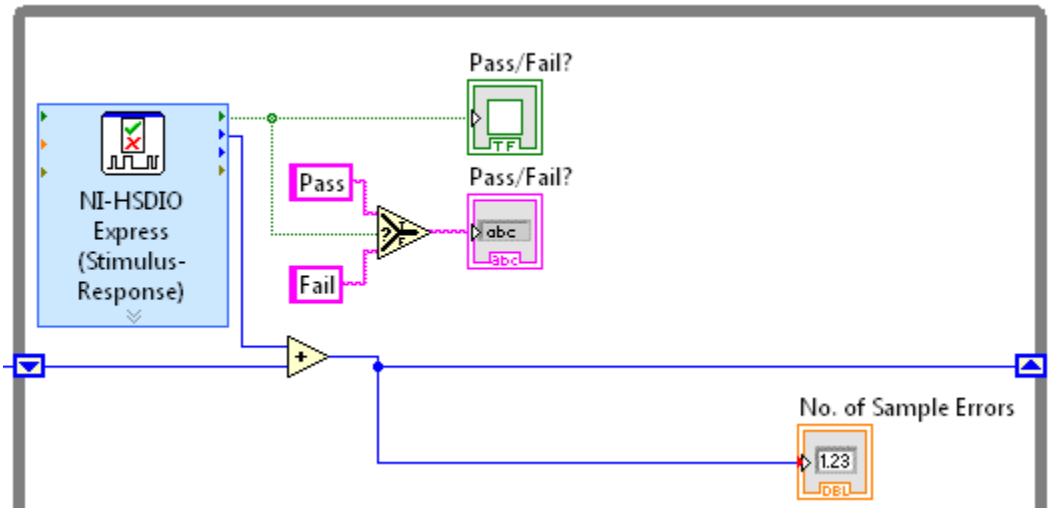
39. Wire the **Shift Register** from the left side of the **While Loop** to the **y** input terminal of the **Add** function



40. **Initialize** the **Shift Register**
    a. Right-click on the **Shift Register** on the left side of the While Loop and select **Create Constant**
    b. Enter **0** as the constant value



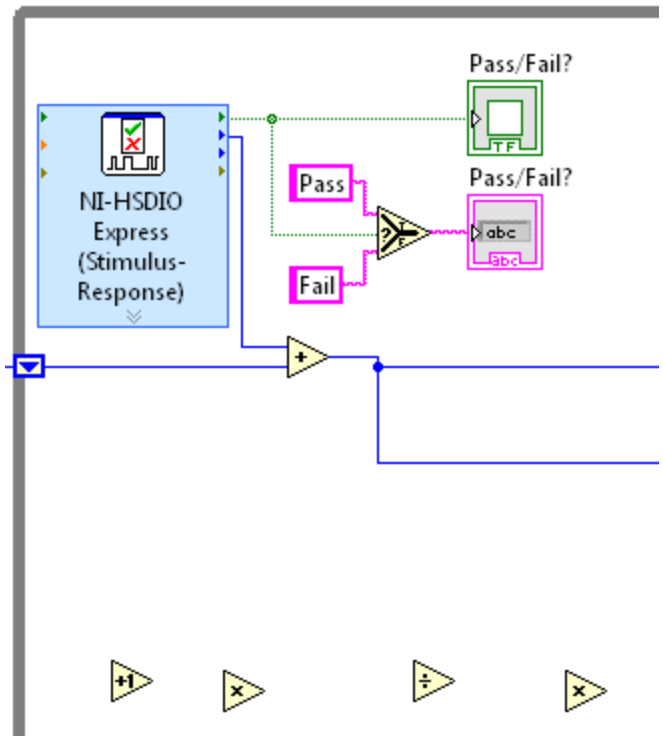
41. Wire the **x+y** output terminal of the **Add** function to the **No. Of Samples** Numeric Indicator



42. Add the following functions to the **While Loop** using the Quick Drop window

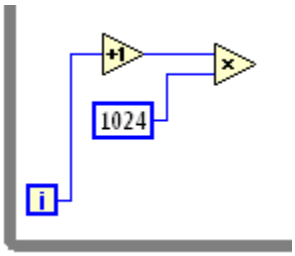
    Tip: You can bring up the Quick Drop window using CTRL + SPACE

    a. **Increment**
    b. **Multiply**
    c. **Divide**
    d. **Multiply**

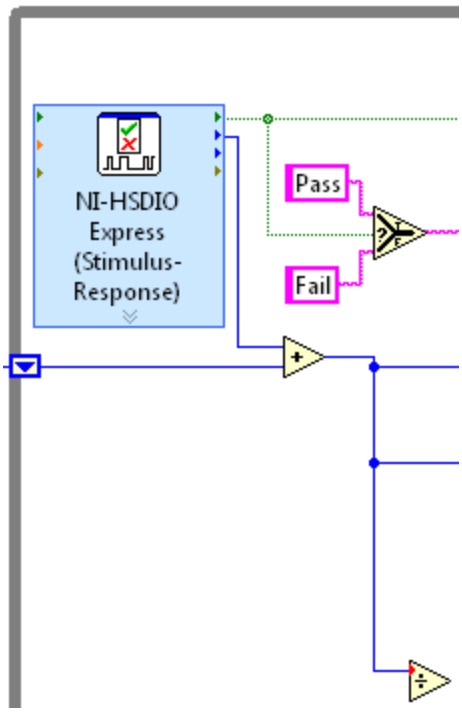


43. Wire the **While Loop index** to the **Increment** function
44. Wire the **x+1** output terminal of the **Increment** function to the **x** input terminal of the **Multiply** function
45. Wire a **Constant** value of **1024** in to the **y** input terminal of the **Multiply** function

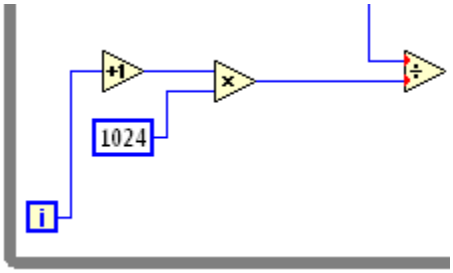
    Tip: You can create constants by right-clicking on an input terminal and selecting Create»Constant



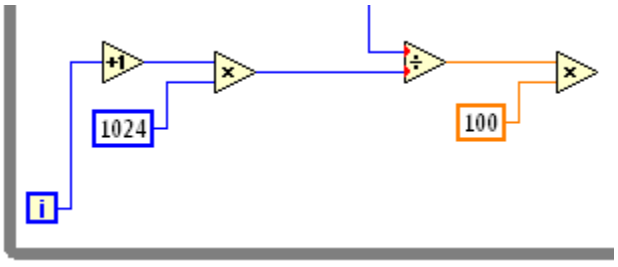
46. Wire the **x+y** output terminal of the **Add** function to the **x** input of the **Divide** function



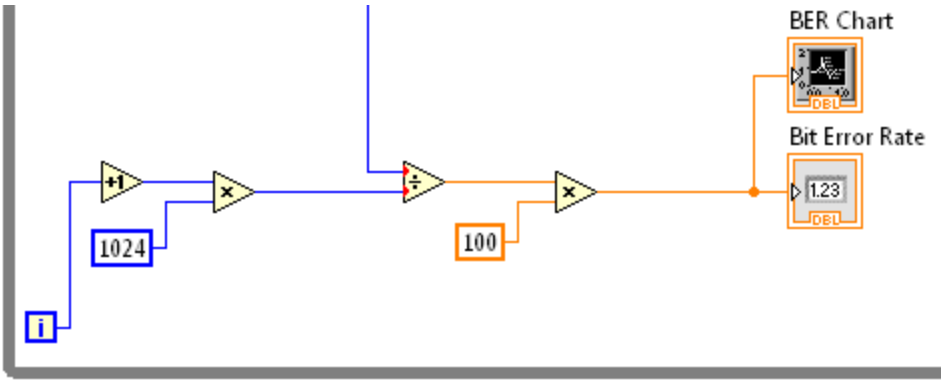
47. Wire the **x*y** output terminal of the **Multiply** function to the **y** input terminal of the **Divide** function



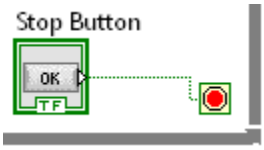
48. Wire the **x/y** output terminal of the **Divide** function to the **x** input of the second **Multiply** function
49. Wire a **Constant** value of **100** in to the **y** input terminal of the second **Multiply** function



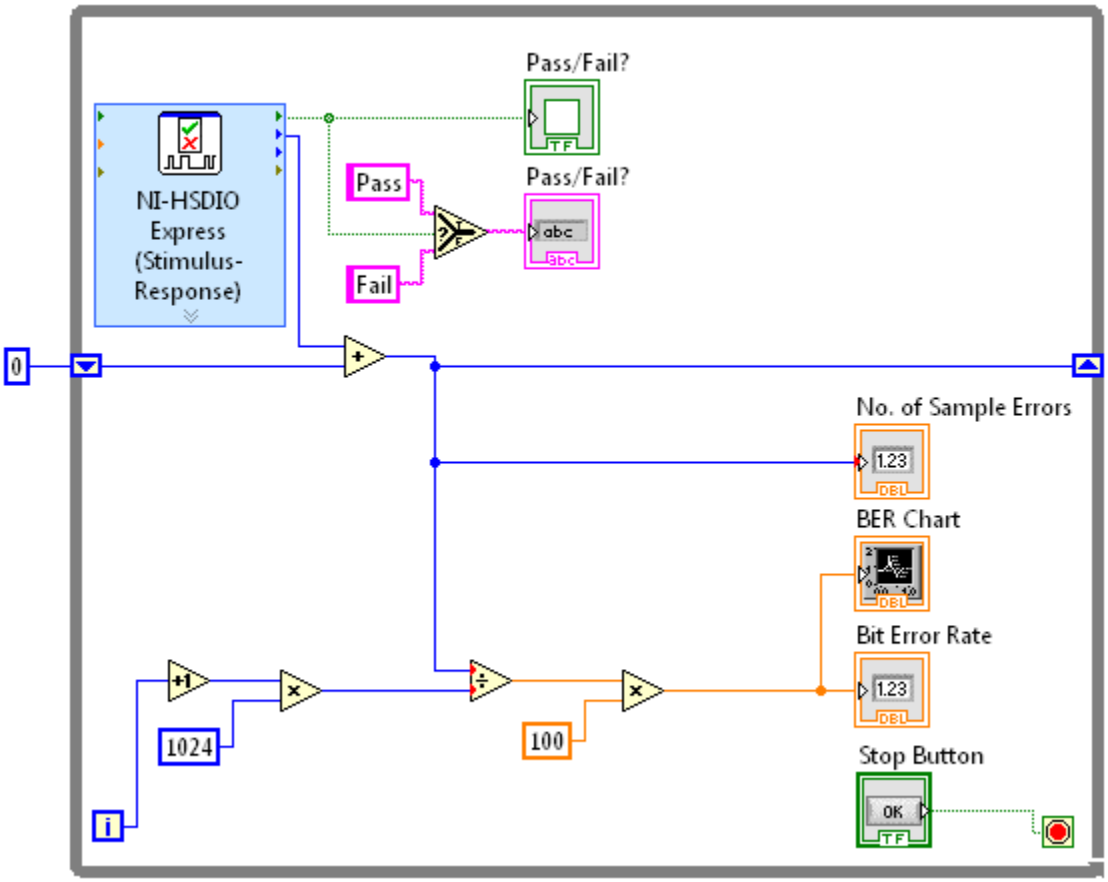
50. Wire the **x*y** output terminal of the second **Multiply** function to the **BER Waveform Chart** and the **Bit Error Rate Numeric Indicator**



51. Wire the **Stop Button** to the While Loop Condition



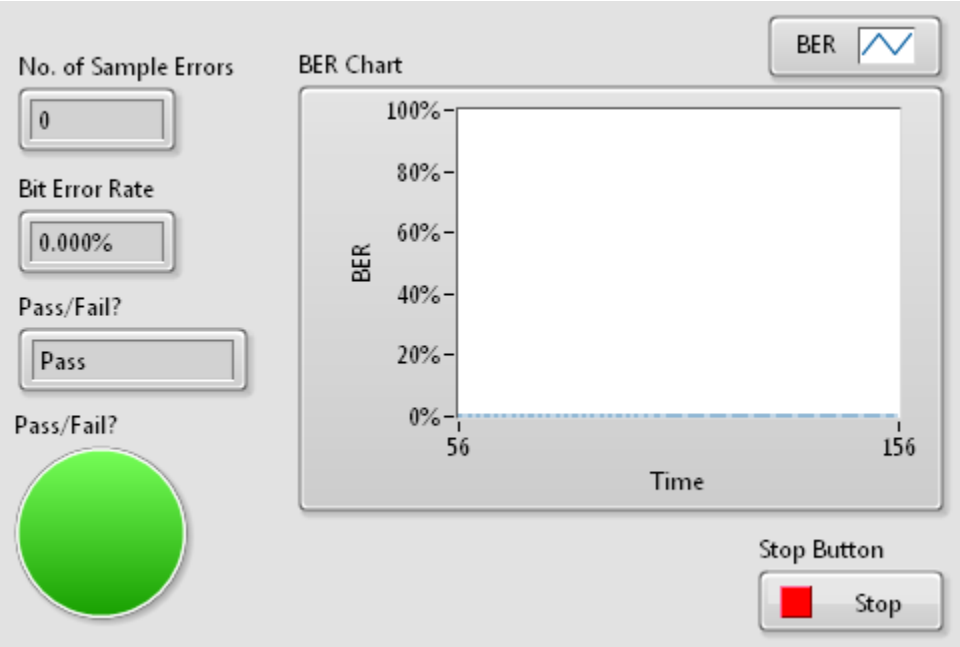
52. Your **code** should look similar to the following **Block Diagram**



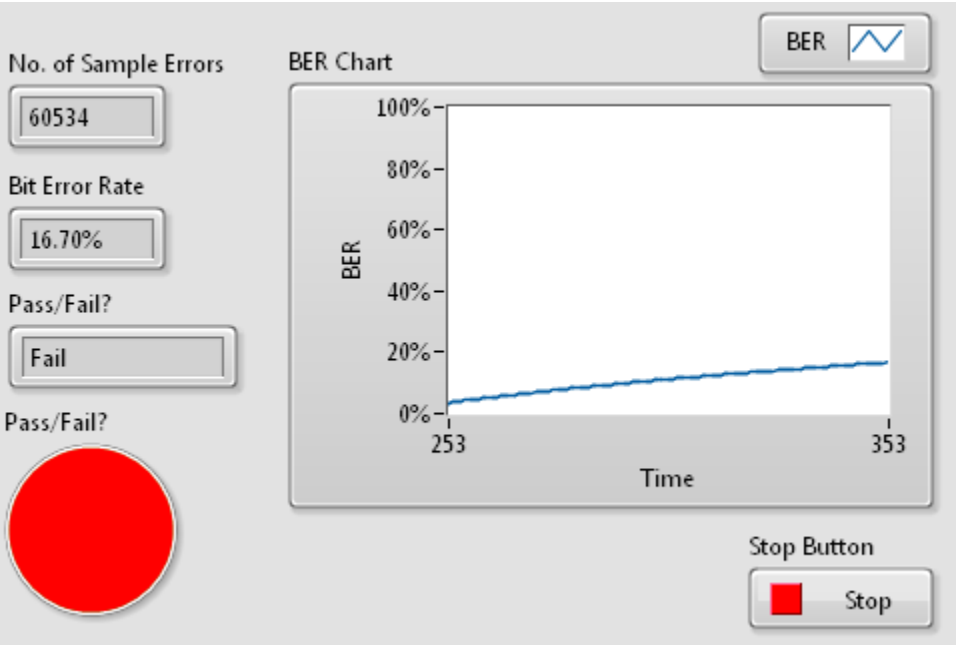
53. On the Socket 0 UUT BERT, place the toggle switch in the Pass position.

54. **Run** the application.

55. Observe that there are no errors and the status of the test is **Passed**.



56. Now place the Socket 0 UUT BERT toggle switch in the Fail position to disconnect the line.
57. Observe that errors now occur and the status of the test is Failed.



58. On the Socket 0 UUT BERT, place the toggle switch in the Pass position.
59. Press the small push button on the **Socket 0 UUT BERT** to inject errors into the digital lines that are being generated and observe that the status of the test will flash **Failed**
60. **Save** and **close** the application

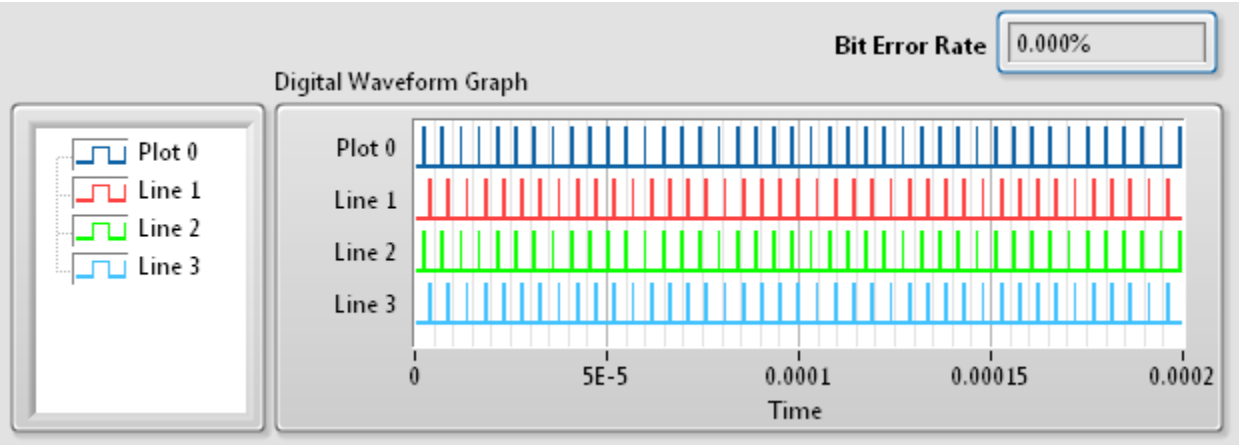
EXERCISE 7.3 – BIT ERROR RATE TEST WITH STANDARD VIS

Goal

The goal of this exercise is to open and run a pre-built example that shows how you could implement a Bit Error Rate using Standard VIs, rather than Express VIs.

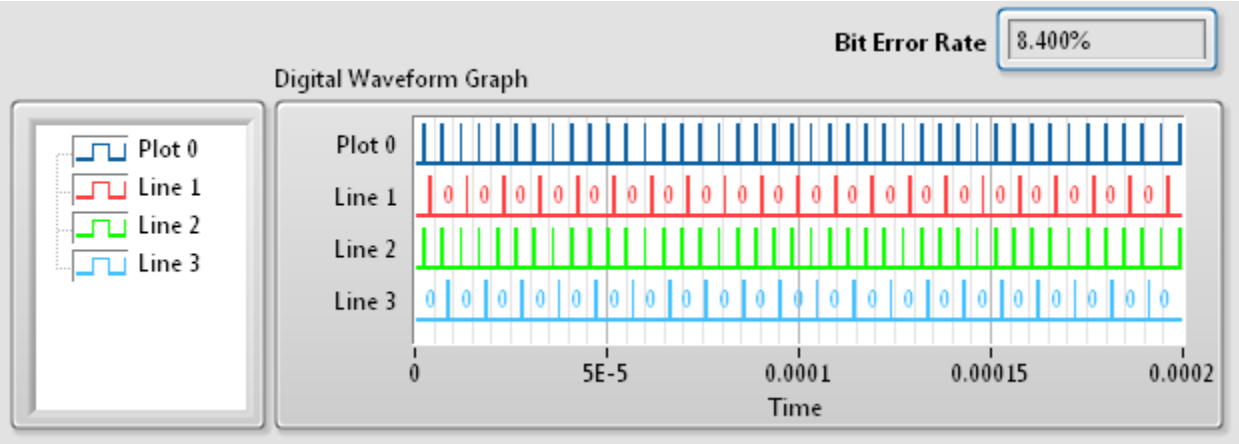
Steps

1. The LabVIEW for Instrumentation Hands-On Project should already be open, but if it is not, browsing to `C:\Seminars\LabVIEW for Instrumentation Hands-On` and double-click the LabVIEW for Instrumentation Hands-On.lvprj file.
2. In the Bit Error Rate Test folder, open the Bit Error Rate Test – Standard VIs.
3. Open the block diagram and observe that the source code is implemented using Standard VIs from the HSDIO instrument driver.
4. On the Socket 0 UUT BERT, place the toggle switch in the Pass position.
5. **Run** the application and observe that there are no errors.



6. Now place the Socket 0 UUT BERT toggle switch in the Fail position.

7. Running the application again will disconnect the line and cause errors to occur.



8. On the Socket 0 UUT BERT, place the toggle switch in the Pass position.
9. Select **Run Continuously**.

10. While the application is running press the small push button on the Socket 0 UUT BERT to inject errors into the digital lines that are being generated and observe that the status of the test will flash Failed.
11. Deselect **Run Continuously**.
12. **Close** and *do not save* the application.