Al Imam Mohammad Ibn Saud Islamic University

College of Engineering

Department of Electrical Engineering

LabVIEW Manual

Prepared By: Mohammad Obaidullah Khan

LABVIEW

Experiment No: 1. INTRODUCTION TO LABVIEW

Objective: To learn the LABVIEW Basics and get familiar with LABVIEW commands.

Equipments Required: LABVIEW (software)

Overview of LABVIEW

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is a platform and development environment for a visual programming language from National Instruments. The graphical language is named "G". LabVIEW is commonly used for data acquisition, instrument control, and industrial automation on a variety of platforms including Microsoft Windows, various flavors of UNIX, Linux, and Mac OS.

One benefit of LabVIEW over other development environments is the extensive support for accessing instrumentation hardware. Drivers and abstraction layers for many different types of instruments and buses are included or are available for inclusion. These present themselves as graphical nodes. The abstraction layers offer standard software interfaces to communicate with hardware devices. The provided driver interfaces save program development time. Many libraries with a large number of functions for data acquisition, signal generation, mathematics, statistics, signal conditioning, analysis, etc.,

along with numerous graphical interface elements are provided in several LabVIEW package options. Another benefit of the LabVIEW environment is the platform independent nature of the G-code (LabVIEW programming language), which is (with the exception of a few platform-specific

functions) portable between the different LabVIEW systems for different operating systems (Windows, MacOSX and Linux).

Dataflow

The programming language used in LabVIEW, called "G", is a dataflow language. Execution is determined by the structure of a graphical block diagram (the LV-source code) on which the programmer connects different function-nodes by drawing wires. These wires propagate variables and any node can execute as soon as all its input data become available. Since this might be the case for multiple nodes simultaneously, G is inherently capable of parallel execution.

Screenshot of a simple LabVIEW program that generates, synthesizes, analyzes and displays waveforms, showing the block diagram and front panel. Each symbol on the block diagram represents a LabVIEW subroutine (subVI) which can be another LabVIEW program or a LV library function.

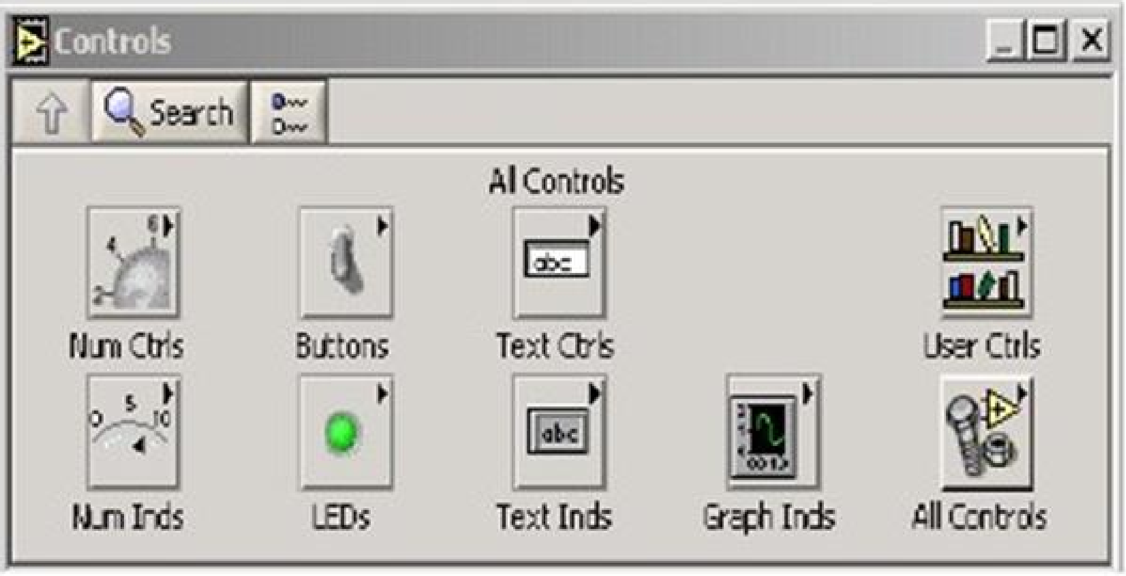
Graphical programming LabVIEW ties the creation of user interfaces (called front panels) into the de velopment cycle. LabVIEW programs/subroutines are called virtual instruments (VIs). Each VI h as three components: Each VI contains three main parts:

1. Front Panel - How the user interacts with the VI.
2. Block Diagram - The code that controls the program.
3. Icon/Connector - Means of connecting a VI to other VIs.

The Front Panel is the user interface of the VI, to interact with the user when the program is running. Users can control the program, change inputs, and see data updated in real time. You build the front p anel with controls and indicators (available on the Control Palette), which are the interactive in put and output terminals of the VI, respectively. Controls are knobs, pushbuttons, dials, and other input devices.
Indicators are graphs, LEDs, and other displays.

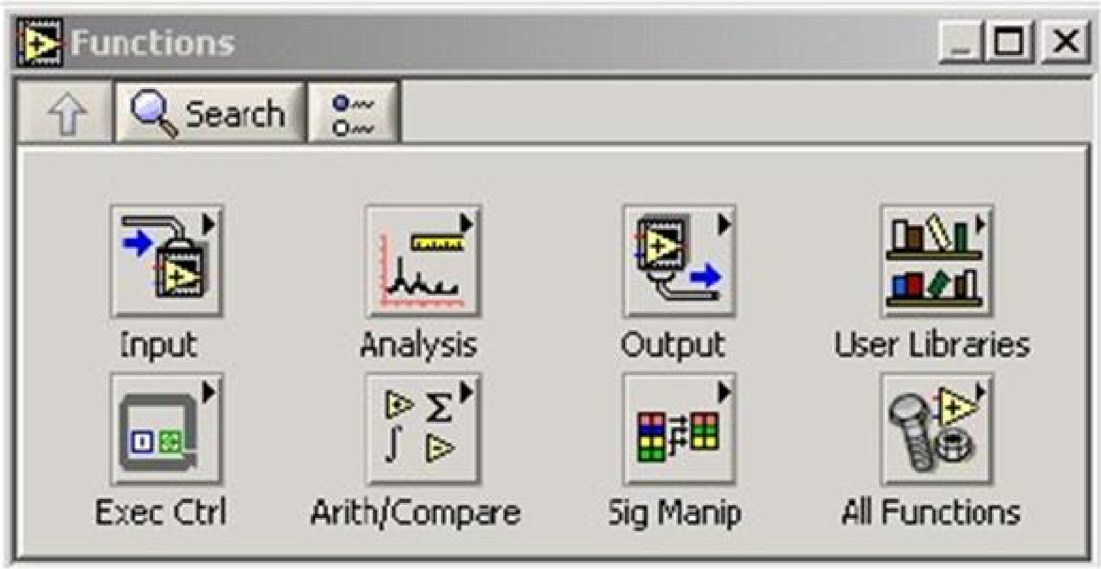
Controls simulate instrument input devices and supply values to the block diagram of the VI.
Indicators simulate instrument output devices and display values the block diagram acquires or

generates. These may include data, program states, and other information. Every front panel co ntrol or indicator has a corresponding terminal on the block diagram. When a VI is run, values from controls flow through the block diagram, where they are used in the functions on the diagram , and the results are passed into other
functions or indicators. The Controls palette is available only on the front panel. Go to Windo w»Show Controls Palette or right-click the front panel workspace to display the Controls palette.



The Block Diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram. Additionally, the block diagram contains functions and structures from built-in

LabVIEW VI libraries. These can be accessed from the Functions Palette. Wires connect each of the nodes on the block diagram, including control andindicator terminals, functions, and structures.

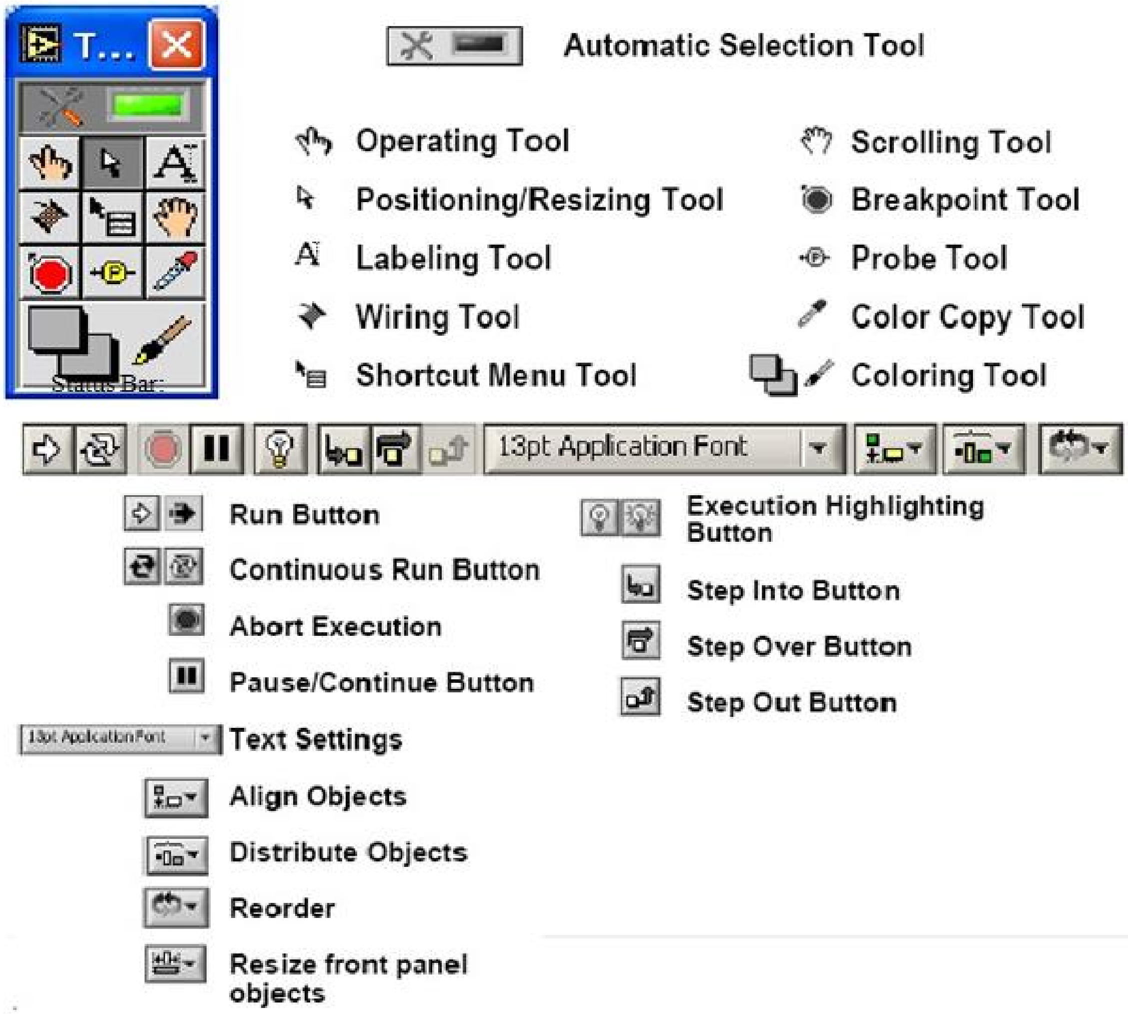


The Functions palette is available only on the block diagram. Select Window»Show Functions Palette or right-click the block diagram workspace to display the Functions palette The Icon/Connector may represent the VI as a subVI in block diagrams of calling VIs. Controls and indicators on the front panel allow an operator to input data into or extract data from a running virtual instrument.

However, the front panel can also serve as a programmatic interface. Thus a virtual instrument can either be run as a program, with the front panel serving as a user interface, or, when droppedas a node onto the block diagram, the front panel defines the inputs and outputs for the given node through the connector pane. This implies each VI can be easily tested before being embedded as a subroutine into a larger program.
The graphical approach also allows non-programmers to build programs by simply dragging and dropping virtual representations of the lab equipment with which they are already familiar.

The LabVIEW programming environment, with the included examples and the documentation, makes it easy to create small applications. For complex algorithms or large-scale code it is important

that the programmer possess an extensive knowledge of the special LabVIEW syntax and the topology of its memory management. The most advanced LabVIEW development systems offer the possibility of building stand alone application.The other features that you will need include the Tools Palette which is a Floating Palette that is used to operate and modify front panel and block diagram objects and the Status Toolbar. They co ntain the following tools and buttons respectively.



Exercise 1 - Open and Run a Virtual Instrument

Examine the signal feneration processing VI and run it. Change the frequency and types of the input signals and notice how the display on the graph changes. Change the signal processing window and filter options. After you have changed the VI and other options, you can change , stop the VI processing by pressing stop button .

1. Select Start»Programs»National Instruments»LabVIEW 7.0»LabVIEW to launch LabVIEW.

The LabVIEW dialog box appears.

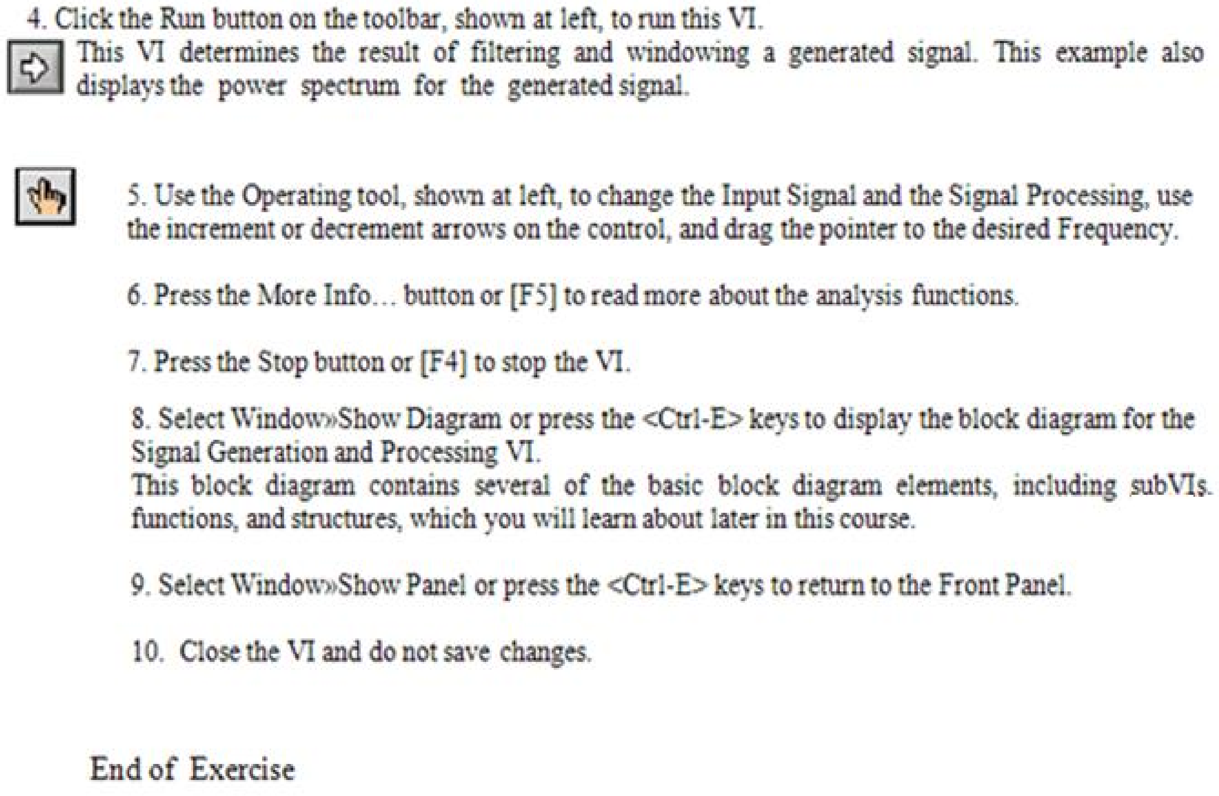
2. Select Help»Find Examples. The dialog box that appears lists and links to all avail able

LabVIEW example VIs.

3. On the Browse Tab, select browse according to task. Choose Analyzing and Proces sing

Signals, then Signal Processing, then Signal Generation and Processing.vi.

This will open the Signal Generation and Processing VI Front Panel.



4. Click the Run button on the toolbar, shown at left, to run this VI.

This VI determines the result of filtering and windowing a generated signal. This example also displays the power spectrum for the generated signal.

5. Use the Operating tool, shown at left, to change the Input Signal and the Signal Processing, use the increment or decrement arrows on the control, and drag the pointer to the desired Frequency.

6. Press the More Info… button or [F5] to read more about the analysis functions.

7. Press the Stop button or [F4] to stop the VI.

8. Select Window»Show Diagram or press the <Ctrl-E> keys to display the block diagram for the Signal Generation and Processing VI.

This block diagram contains several of the basic block diagram elements, including subVIs. functions, and structures, which you will learn about later in this course.

9. Select Window»Show Panel or press the <Ctrl-E> keys to return to the Front Panel.

10. Close the VI and do not save changes.

End of Exercise

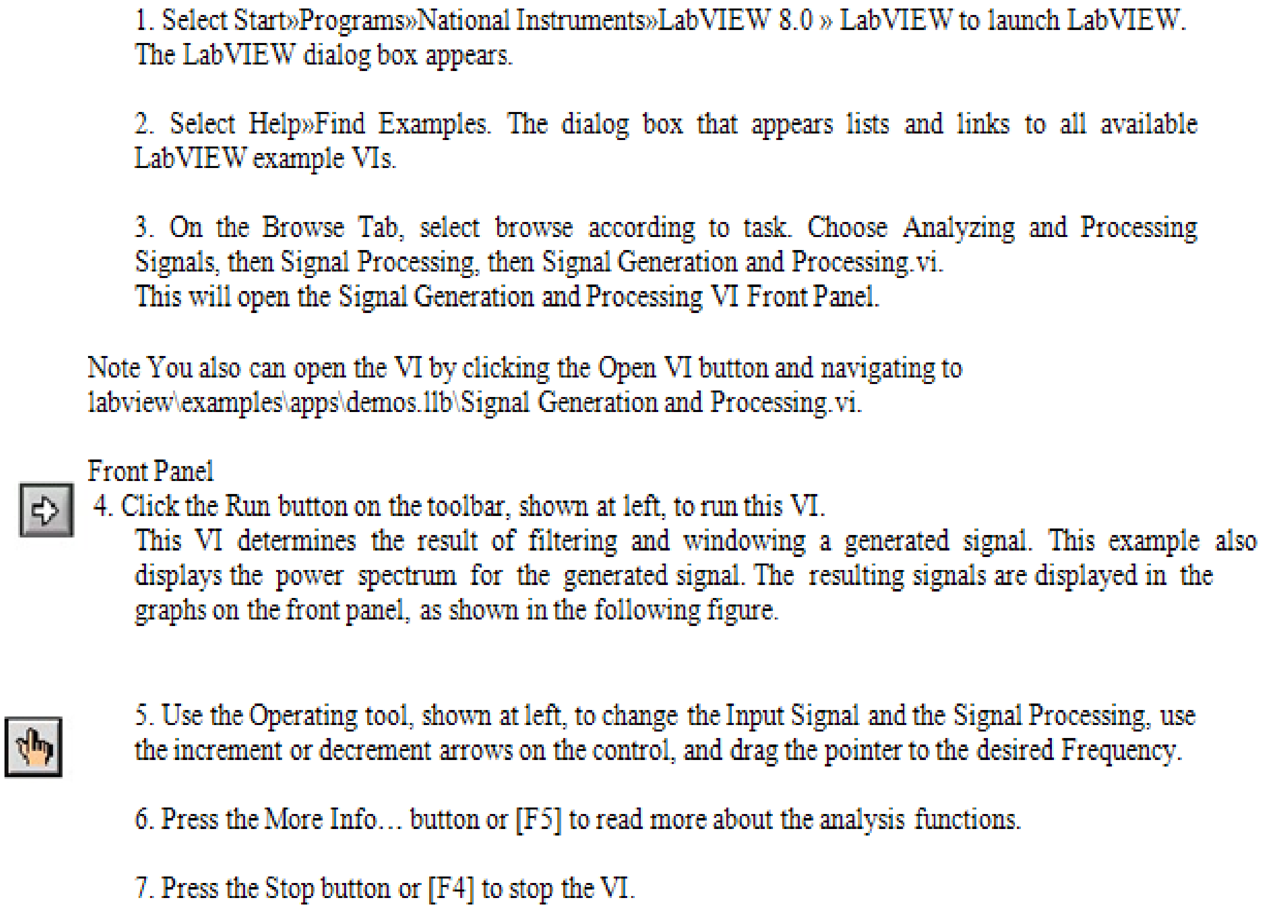
Experiment No: 2. Building a VI & Modifying a Signal

Objective: To learn the LABVIEW Basics and get familiar with LABVIEW commands.

Equipments Required: LABVIEW (software)

Building a VI

Beforebuilding our own VI we will first examine the Signal Generation and Processing VI and run it. Change the frequencies and types of the input signals and notice how the display on the graph changes.
Change the Signal Processing Window and Filter options. After you have examined the VI and the different options you can change, stop the VI by pressing the Stop button.



1. Select Start»Programs»National Instruments»LabVIEW 8.0 » LabVIEW to launch LabVIEW. The LabVIEW dialog box appears.

2. Select Help»Find Examples. The dialog box that appears lists and links to all available LabVIEW example VIs.

3. On the Browse Tab, select browse according to task. Choose Analyzing and Processing Signals, then Signal Processing, then Signal Generation and Processing.vi.
This will open the Signal Generation and Processing VI Front Panel.

Note You also can open the VI by clicking the Open VI button and navigating to labview\examples\apps\demos.llb\Signal Generation and Processing.vi.

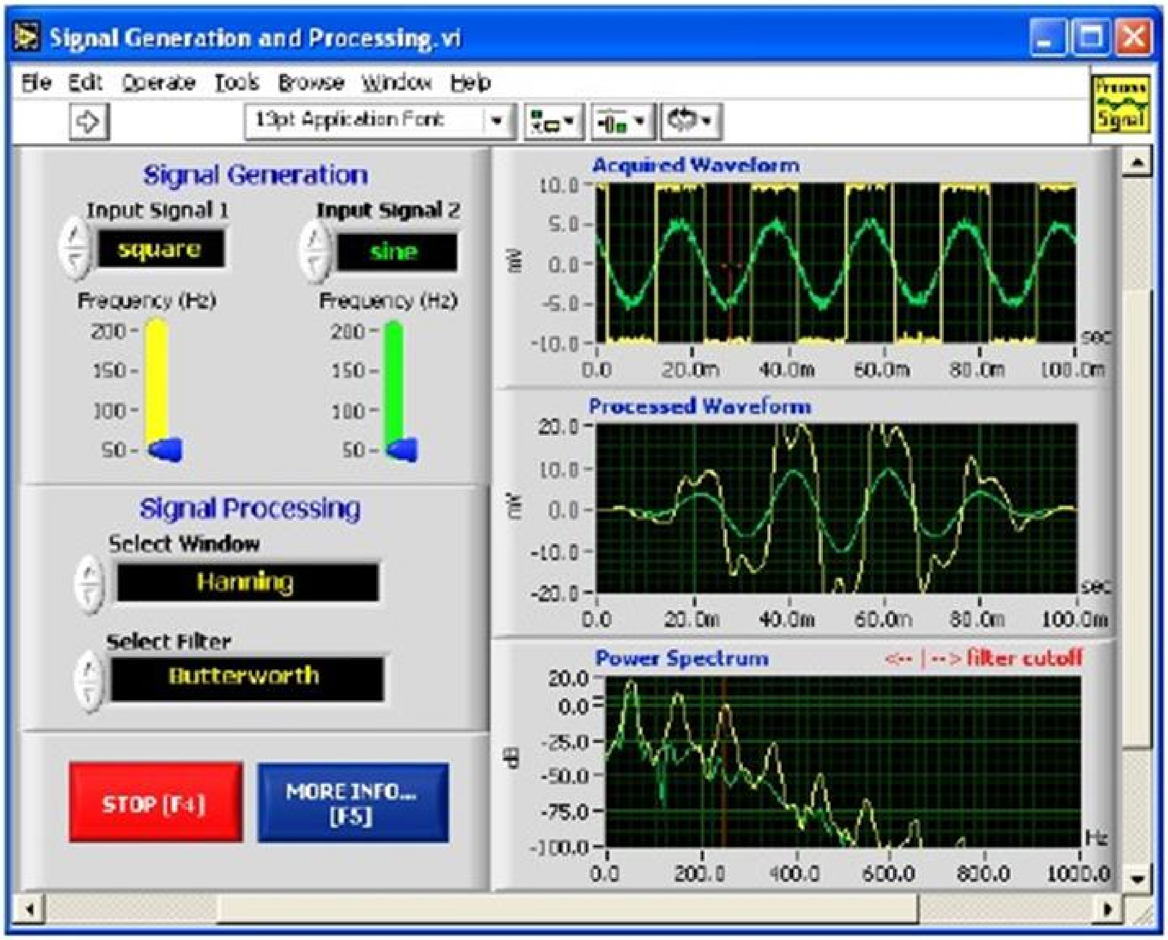
Front Panel

4. Click the Run button on the toolbar, shown at left, to run this VI.
This VI determines the result of filtering and windowing a generated signal. This example also displays the power spectrum for the generated signal. The resulting signals are displayed in the graphs on the front panel, as shown in the following figure.

5. Use the Operating tool, shown at left, to change the Input Signal and the Signal Processing, use the increment or decrement arrows on the control, and drag the pointer to the desired Frequency.

6. Press the More Info… button or [F5] to read more about the analysis functions.

7. Press the Stop button or [F4] to stop the VI.



Input Signal 1: square

Input Signal 2: sine

Select Window: Hanning

Select Filter: Butterworth

Block Diagram

8. Select Window»Show Diagram or press the <Ctrl-E> keys to display the block diagram for the Signal Generation and Processing VI.

This block diagram contains several of the basic block diagram elements, including su bVIs, functions, and structures, which you will learn about later in this course.

9. Select Window»Show Panel or press the <Ctrl-E> keys to return to the Front Panel.

10. Close the VI and do not save changes.

Creating Your Own VI:

When you create an object on the Front Panel, a terminal will be created on the Block Diagram. These terminals give you access to the Front Panel objects from the Block Diagram code. Each terminal contains useful information about the Front Panel object it corresponds to. For example, the colour and symbols provide the data type. Double-precision, floating point numbers are represented with orange terminals and the letters DBL, boolean terminals are green with TF lettering. In general, orange terminals should wire to orange terminals, green to green, and so on. This is not a hard-and-fast rule; LabVIEW will allow a user to connect a blue terminal (integer value) to an orange terminal (fractional value), for example.

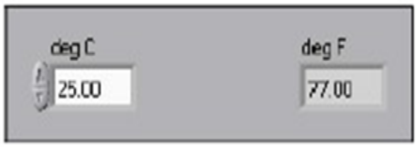
For more help with the terminals, right-click on the function and select Visible Items»Terminals. The functions picture will be pulled back to reveal the connection terminals. For additional help, select Help»Show Context Help, or press <Ctrl-H>. As you move your mouse over the function, this window will show you the function, terminals, and a brief help description in the context help window. For organization you can, right-click on the particular wire in question and choose Clean Up Wire to automatically re-route that wire.

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

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

Front Panel

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



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

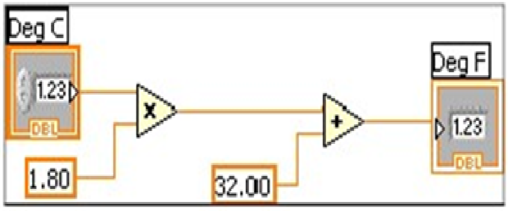
a. Select the digital indicator on the Controls»Numeric Indicators palette.
b. Move the indicator to the front panel and click to place the indicator.
c. Type deg F inside the label and click outside the label or click the Enter button.

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

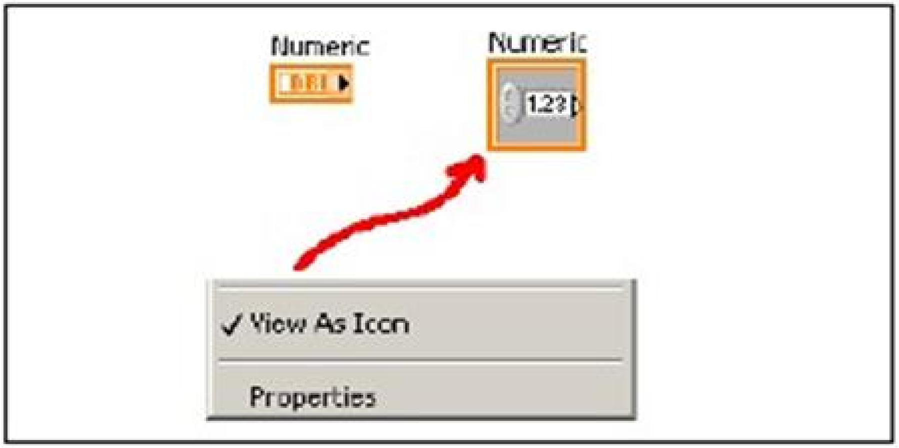
Note Control terminals have a thicker border than indicator terminals.

Block Diagram

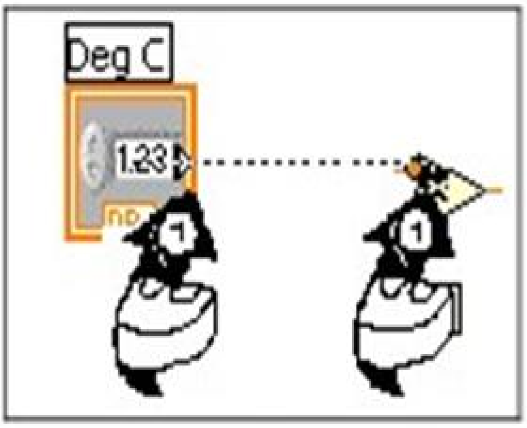
5. Display the block diagram by clicking it or by selecting Window»Show Diagram.



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



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



You can bend a wire by clicking to tack the wire down and moving the cursor in a perpendicular direction. Press the spacebar to toggle the wire direction.

- To identify terminals on the nodes, right-click the Multiply and Add functions and select Visible Items»Terminals from the shortcut menu to display the connector pane. Return to the icons after wiring by right-clicking the functions and selecting Visible Items»
- Terminals from the shortcut menu to remove the checkmark.
- When you move the Wiring tool over a terminal, the terminal area blinks, indicating that clicking will connect the wire to that terminal and a tip strip appears, listing the name of the terminal.

You can bend a wire by clicking to tack the wire down and moving the cursor in a perpendicular direction. Press the spacebar to toggle the wire direction.

- To identify terminals on the nodes, right-click the Multiply and Add functions and select Visible Items»Terminals from the shortcut menu to display the connector pane. Return to the icons after wiring by right-clicking the functions and selecting Visible Items»
- Terminals from the shortcut menu to remove the checkmark.
- When you move the Wiring tool over a terminal, the terminal area blinks, indicating that clicking will connect the wire to that terminal and a tip strip appears, listing the name of the terminal.
- To cancel a wire you started, press the <Esc> key, right-click, or click the source terminal.

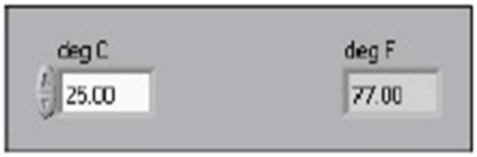
10. Display the front panel by clicking it or by selecting Window»Show Panel.
11. Save the VI for later use (have a folder by your own name).
12. Enter a number in the digital control and run the VI.
    a. Use the Operating tool or the Labeling tool to double-click the digital control and type a new number.
    b. Click the Run button to run the VI.
    c. Try several different numbers and run the VI again.

Exercise 3 – Create a SubVI

Front Panel

1. Select File»Open and navigate to your folder to open the C to F VI.

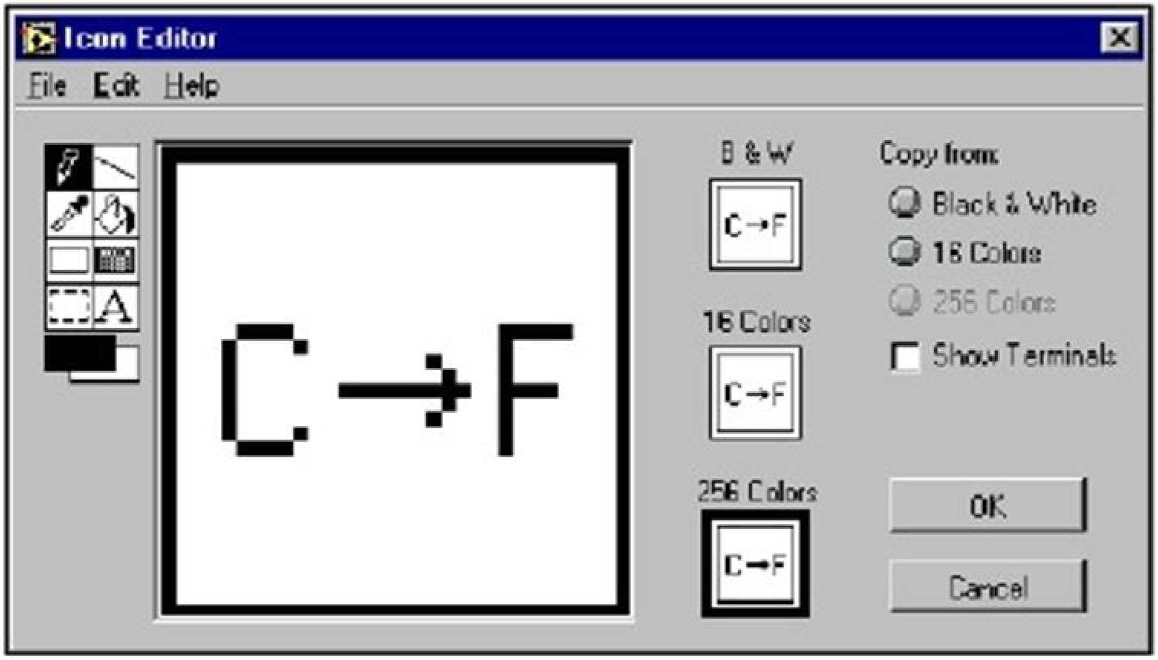
The following front panel appears.



deg C 25.00

deg F 77.00

2. Right-click the icon in the upper right corner of the front panel and select Edit Icon from the shortcut menu. The Icon Editor dialog box appears.
3. Double-click the Select tool on the left side of the Icon Editor dialog box to select the default icon.
4. Press the <Delete> key to remove the default icon.
5. Double-click the Rectangle tool to redraw the border.
6. Create the following icon.



a. Use the Text tool to click the editing area.
b. Type C and F.
c. Double-click the Text tool and change the font to Small Fonts.
d. Use the Pencil tool to create the arrow. (To draw horizontal or vertical straight lines press the <Shift> key while you use the Pencil tool to drag the cursor).
e. Use the Select tool and the arrow keys to move the text and arrow you created.
f. Select the B&W icon and select 256 Colors in the Copy from field to create a black and white icon, which LabVIEW uses for printing unless you have a color printer.
g. When the icon is complete, click the OK button to close the Icon Editor dialog box. The icon appears in the upper right corner of the front panel and block diagram.

7. Right-click the icon on the front panel and select Show Connector from the shortcut menu to define the connector pane terminal pattern.
LabVIEW selects a connector pane pattern based on the number of controls and indicators on the front panel. For example, this front panel has two terminals, deg C and deg F, so LabVIEW selects a connector pane pattern with two terminals.

8. Assign the terminals to the digital control and digital indicator.
   a. Select Help»Show Context Help to display the Context Help window. View each connection in the Context Help window as you make it.

b. Click the left terminal in the connector pane. The tool automatically changes to the Wiring tool, and the terminal turns black.

c. Click the deg C control. The left terminal turns orange, and a marquee highlights the control.

d. Click an open area of the front panel. The marquee disappears, and the terminal changes to the data type color of the control to indicate that you connected the terminal.

e. Click the right terminal in the connector pane and click the deg F indicator. The right terminal turns orange.

f. Click an open area on the front panel. Both terminals are orange.

g. Move the cursor over the connector pane. The Context Help window shows that both terminals are connected to floating-point values.

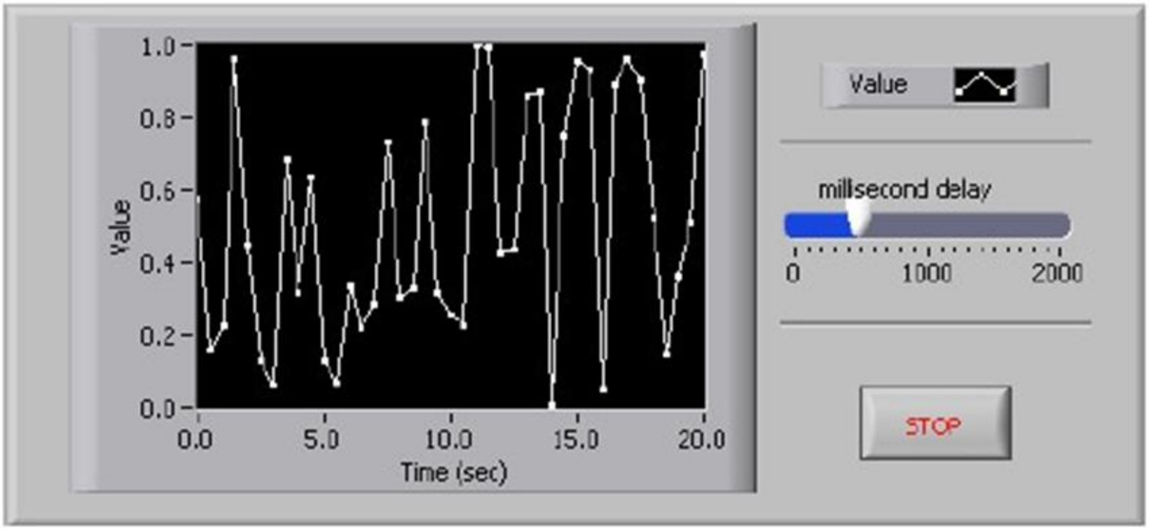
9. Select File»Save to save the VI.

Exercise 4 ï Using Loops

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

Front Panel

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



a. Select the horizontal pointer slide on the Controls»Numeric Controls palette and place it on the front panel. You will use the slide to change the software timing.

b. Type millisecond delay inside the label and click outside the label or click the Enter button on the toolbar, shown at left.

c. Place a Stop Button from the Controls»Buttons palette.

d. Select a waveform chart on the Controls»Graph Indicators palette and place it on the front panel. The waveform chart will display the data in real time.

e. Type Value History inside the label and click outside the label or click the Enter button.

f. The waveform chart legend labels the plot Plot 0. Use the Labeling tool to triple-click Plot 0 in the chart legend, type Value, and click outside the label or click the Enter button to relabel the legend.

g. The random number generator generates numbers between 0 and 1, in a classroom setting you could replace this with a data acquisition VI. Use the Labeling tool to double-click 10.0 in the y-axis, type 1, and click outside the label or click the Enter button to rescale the chart.

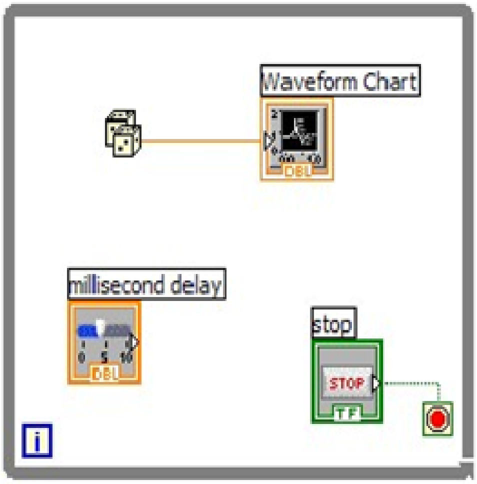
h. Change –10.0 in the y-axis to 0.

i. Label the y-axis Value and the x-axis Time (sec).

Block Diagram

3. Select Window»Show Diagram to display the block diagram.

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



   a. Select the While Loop on the Functions»Execution Control palette.
   b. Click and drag a selection rectangle around the two terminals.
   c. Use the Positioning tool to resize the loop, if necessary.

5. Select the Random Number (0-1) on the Functions»Arithmetic and Comparison»Numeric palette. Alternatively you could use a VI that is gathering data from an external sensor.
6. Wire the block diagram objects as shown in the previous block diagram.
7. Save the VI as Use a Loop.vi in your folder
8. Display the front panel by clicking it or by selecting Window»Show Panel.
9. Run the VI. The section of the block diagram within the While Loop border executes until the specified condition is TRUE. For example, while the STOP button is not pressed, the VI returns a new number and displays it on the waveform chart.
10. Click the STOP button to stop the acquisition. The condition is FALSE, and the loop stops executing.
11. Format and customize the X and Y scales of the waveform chart.
    a. Right-click the chart and select Properties from the shortcut menu. The following dialog box appears.
    b. Click the Scale tab and select different styles for the y-axis. You also can select different mapping modes, grid options, scaling factors, and formats and precisions. Notice that these will update interactively on the waveform chart
 c. Select the options you desire and click the OK button.

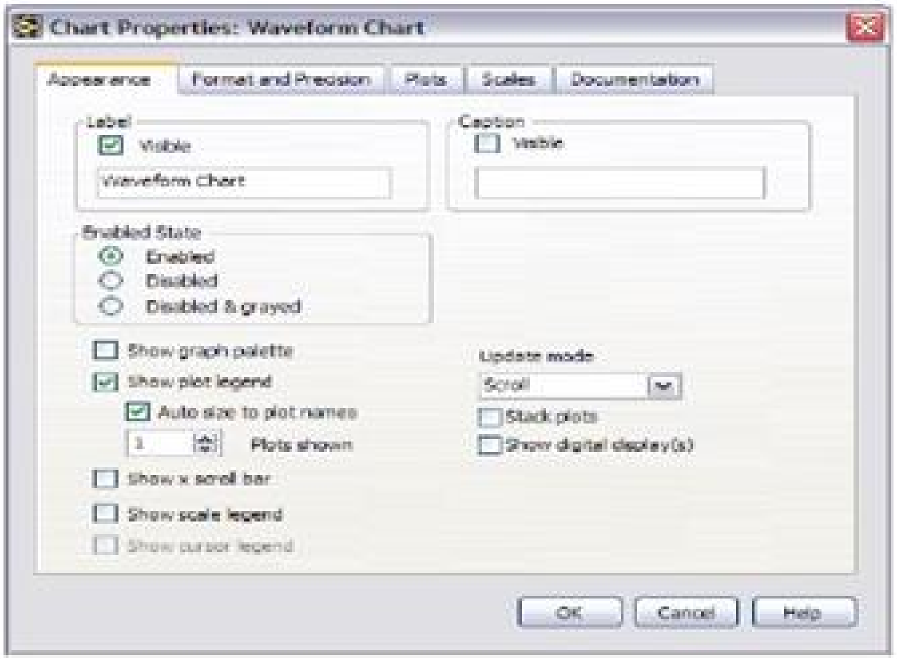


Chart Properties: Waveform Chart

Appearance | Format and Precision | Plots | Scales | Documentation

Label
☑ Visible
Waveform Chart

Caption
□ Visible

Enabled State
● Enabled
○ Disabled
○ Disabled & grayed

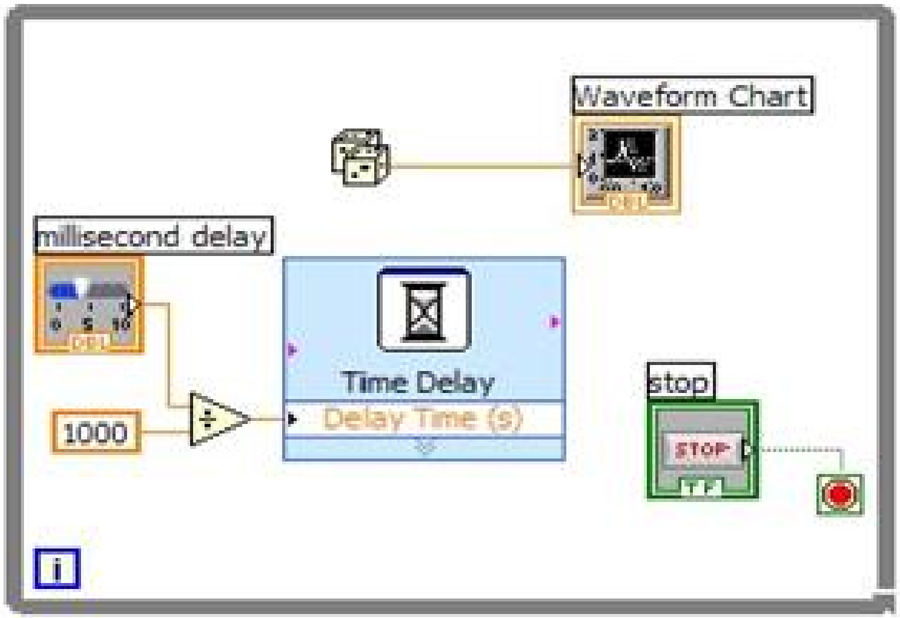
□ Show graph palette
☑ Show plot legend
☑ Auto size to plot names
1 Plots shown
□ Show x scroll bar
□ Show scale legend
□ Show cursor legend

Update mode
Scroll
□ Stack plots
□ Show digital display(s)

OK | Cancel | Help

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

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



a. Place the Time Delay Express VI located on the Functions»Execution Control palette. In the dialog box that appears, insert 0.5. This function would make sure that each iteration occers every half-second(500ms).

b. Divide the millisecond delay by 1000 to get time in seconds. Connect the output of the divide function to the Delay Time (s) input of the Time Delay Express VI. This will allow you to adjust the speed of the execution from the pointer slide on the front panel.

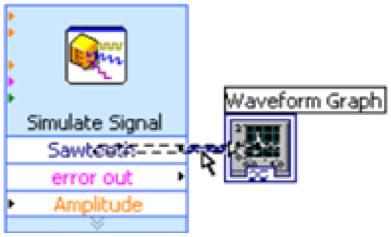
13. Save the VI, because you will use this VI later in the course.

14. Run the VI.

15. Try different values for the millisecond delay and run the VI again. Notice how this effects the speed of the number generation and display.

2. Modifying a Signal

Complete the following steps to scale the signal by 10 and display the results in the graph on the front panel.

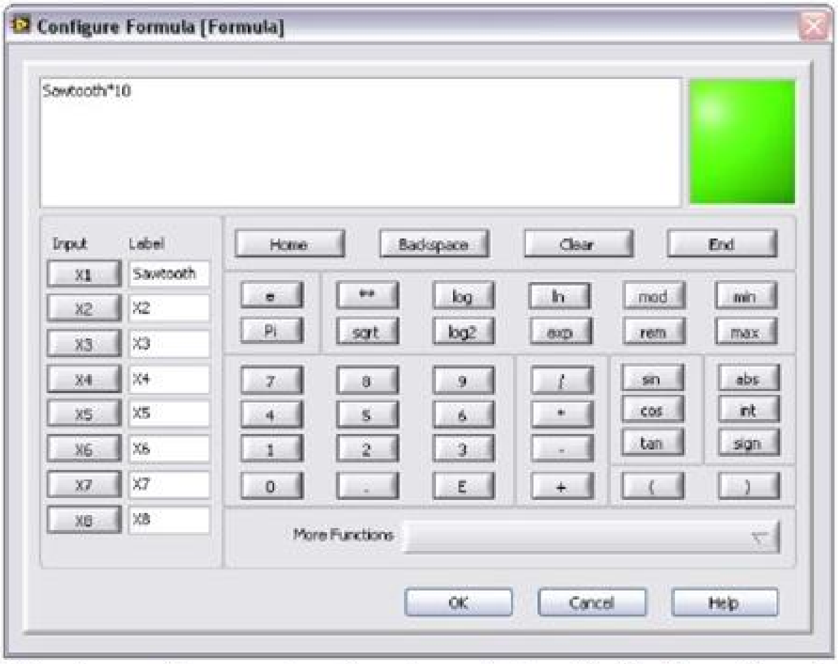


1. The Functions palette opens with the Express subpalette visible by default. If you have selected another subpalette, you can return to the Express subpalette by clicking Express on the Functions palette.
2. On the Arithmetic & Comparison palette, select the Formula Express VI, shown, and place it on the block diagram inside the loop between the Simulate Signal Express VI and the Waveform Graph terminal. You can move the Waveform Graph terminal to the right to make more room between the Express VI and the terminal. The Configure Formula dialog box appears when you place the Express VI on the block diagram. When you place an Express VI on the block diagram, the configuration dialog box for that VI always appears automatically.

3. Click the Help button, in the bottom right corner of the Configure Formula dialog box to display
the LabVIEW Help topic for this Express VI. The Formula help topic describes the Express VI, the
configuration dialog box options, and the inputs and outputs of the Express VI. Each Express VI
4. has a corresponding help topic you can access by clicking the Help button in the configuration
dialog box or by right-clicking the Express VI and selecting Help from the shortcut menu.
5. In the Formula topic, find the dialog box option whose description indicates that it enters a
6. variable into the formula.
Minimize the LabVIEW Help to return to the Configure Formula dialog box.
Change the text in the Label text box of the dialog box option you read about from X1 to Sawtooth
to indicate the input value to the Formula Express VI. When you click in the String text box at the
7. top of the Configure Formula dialog box, the text changes to match the label you entered.
Define the value of the scaling factor by entering *10 after Saw tooth in the String text box. You
can use the Input buttons in the configuration dialog box or you can use the *, 1, and 0 keyboard

buttons to enter the scaling factor. If you use the Input buttons in the configuration dialog box, LabVIEW places the formula input after the Sawtooth input in the String text box. If you use the keyboard, click in the String text box after Sawtooth and enter the formula you want to appear in the text box.

The Configure Formula dialog box should appear similar to

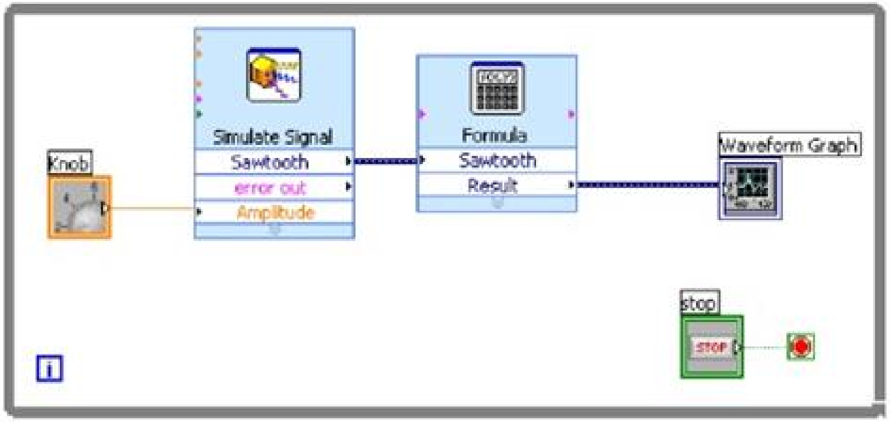


Configure Formula [Formula]

Sawtooth*10

Input	Label
X1	Sawtooth
X2	X2
X3	X3
X4	X4
X5	X5
X6	X6
X7	X7
X8	X8

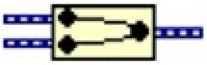
8. Click the OK button to save the current configuration and close the Configure Formula dialog box.
9. Move the cursor over the arrow on the Sawtooth output of the Simulate Signal Express VI.
10. When the Wiring tool appears, click the arrow on the Sawtooth output and then click the arrow on the Sawtooth input of the Formula Express VI, shown at left, to wire the two objects together.
11. Use the Wiring tool to wire the Result output of the Formula Express VI to the Waveform Graph terminal.



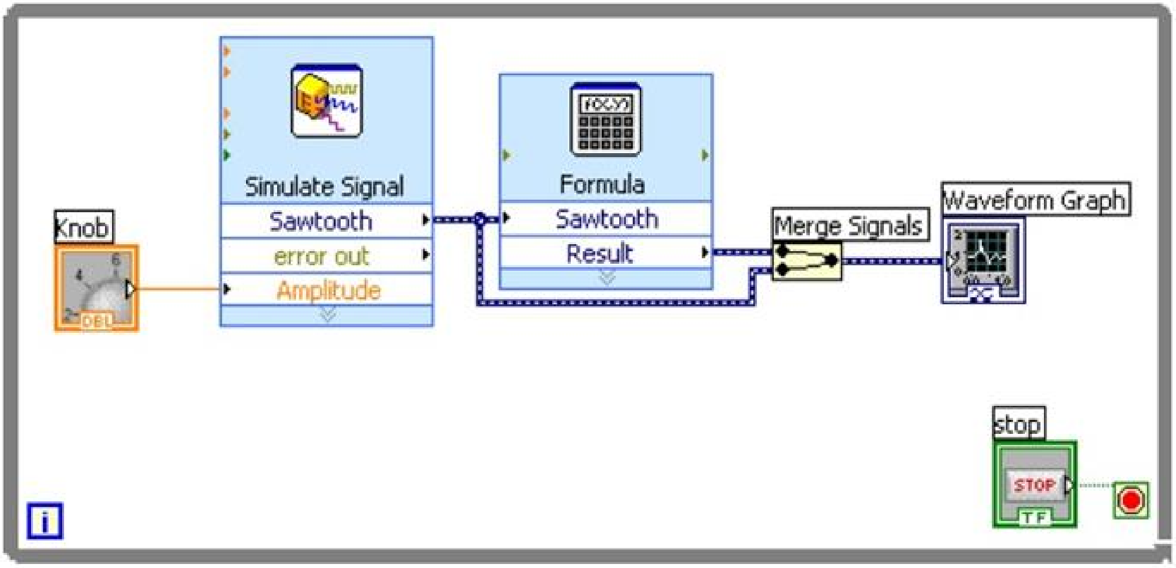
Displaying Two Signals on a Graph

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

1. On the block diagram, move the cursor over the arrow on the Saw tooth output of the Simulate Signal Express VI.
2. Use the Wiring tool to wire the Saw tooth output to the Waveform Graph terminal. The Merge Signals function, shown, appears where the two wires connect. A function is a built-in execution element, comparable to an operator, function, or statement in a text-based programming language. The Merge Signals function takes the two separate signals and combines them so that both can display on the same graph.



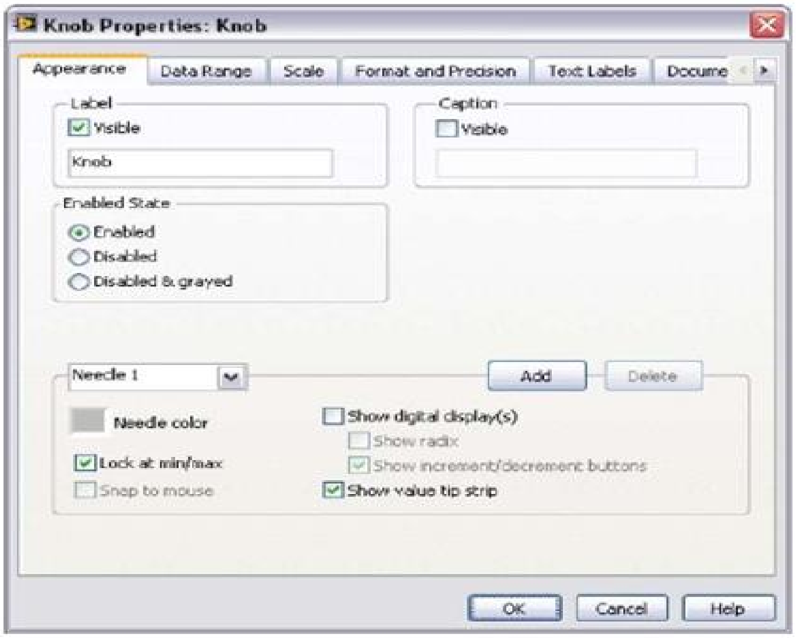
3. Press the <Ctrl-S> keys or select File» Save to save the VI.
4. Return to the front panel, run the VI, and turn the knob control. The graph plots the saw tooth wave and the scaled signal. The maximum value on the y-axis automatically changes to be 10 times the knob value. This scaling occurs because you configured the Formula Express VI to generate a slope of 10.
5. Click the STOP button to stop the VI.



Customizing a Knob Control

The knob control changes the amplitude of the saw tooth wave, so labeling it Amplitude accurately describes the behavior of the knob. Complete the following steps to customize the appearance of the knob.

1. On the front panel, right-click the knob and select Properties from the shortcut menu to display the Knob Properties dialog box.
2. In the Label section on the Appearance page, delete the label Knob, and enter Amplitude in the text box.



Label
☑ Visible
Knob

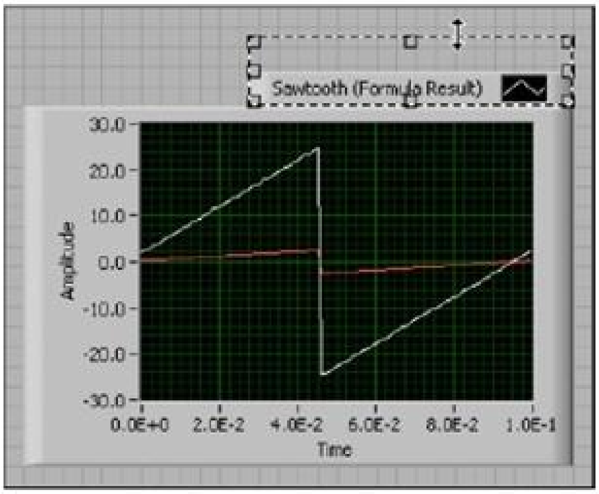
Caption
□ Visible

Enabled State
● Enabled
○ Disabled
○ Disabled & grayed

Needle 1
Needle color
☑ Lock at min/max
□ Snap to mouse
□ Show digital display(s)
□ Show radix
☑ Show increment/decrement buttons
☑ Show value tip strip

3. Click the Scale tab and in the Scale Style section, place a checkmark in the Show colour ramp checkbox. The knob on the front panel updates to reflect these changes.
4. Click the OK button to save the current configuration and close the Knob Properties dialog box.
5. Save the VI.
6. Reopen the Knob Properties dialog box and experiment with other properties of the knob. For example, on the Scale page, try changing the colors for the Marker text color by clicking the color box.
7. Click the Cancel button to avoid applying any changes you made while experimenting. If you want to keep the changes you made, click the OK button.

4. On the Plots page, select Sawtooth from the pull-down menu. In the Colors section, click the Line colour box to display the color picker. Select a new line colour.
5. Select Saw tooth (Formula Result) from the pull-down menu.
6. Place a checkmark in the Do not use waveform names for plot names checkbox.
7. In the Name text box, delete the current label and change the name of this plot to Scaled Sawtooth.
8. Click the OK button to save the current configuration and close the Waveform Graph Properties dialog box. The plot colour on the front panel changes.
9. Reopen the Waveform Graph Properties dialog box and experiment with other properties of the graph. For example, on the Scales page, try disabling automatic scaling and changing the minimum and maximum value of the y-axis.
10. Click the Cancel button to avoid applying any changes you made while experimenting. If you want to keep the changes you made, click the OK button.
11. Save and close the VI.

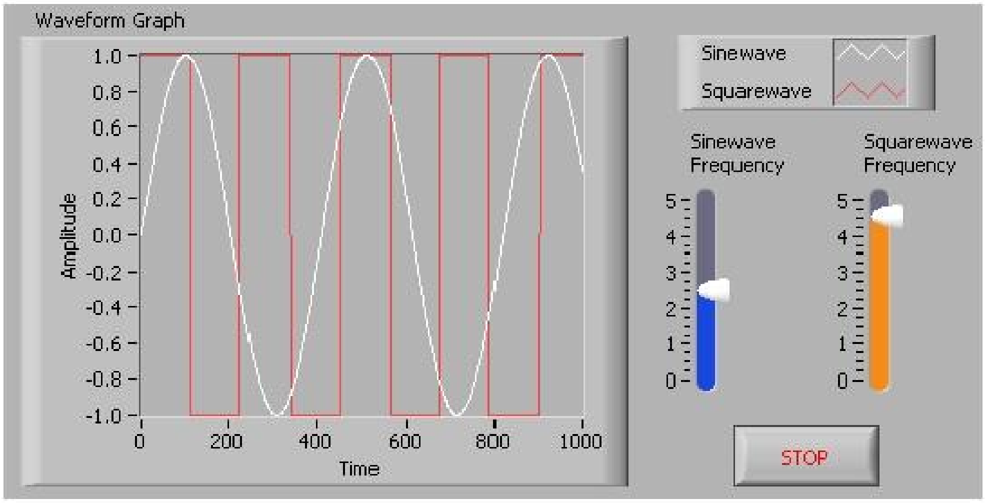

Sawtooth (Formula Result)
30.0
20.0
10.0
0.0
-10.0
-20.0
-30.0
Amplitude
0.0E+0
2.0E-2
4.0E-2
6.0E-2
8.0E-2
1.0E-1
Time

Experiment No. 3: Exercises in Lab VIEW

Exercise 1 - Using Waveform Graphs

Front Panel

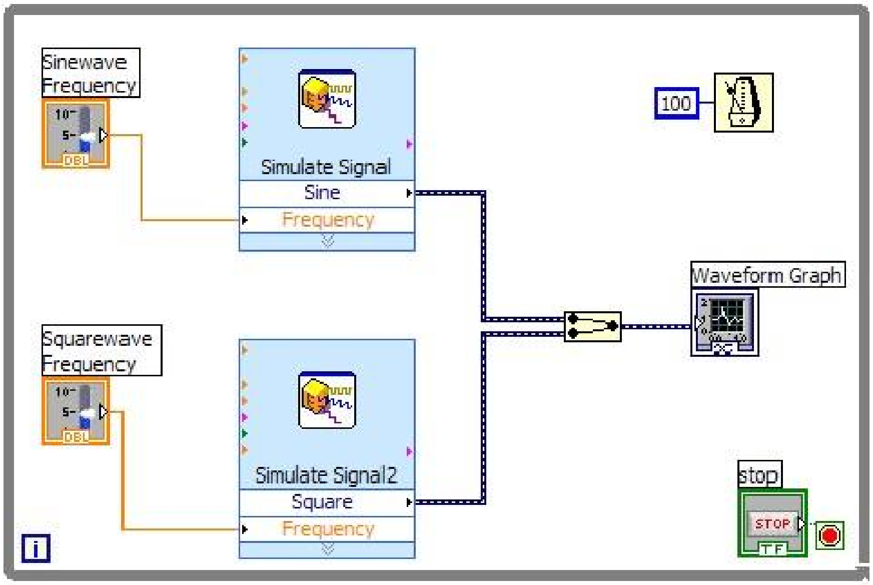
1. Open a new VI and build the following front panel using the following tips.



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

Block Diagram

2. Build the following block diagram.



a. Place a While Loop from Functions» Execution Control palette.
b. Place a Wait Until Next ms Multiple from Functions»All Functions »Time & Dialog and create a constant with a value of 100.
c. Place two Simulate Signal Express VIs from the Functions»Input and leave the Signal type as Sine for the first Simulate Signal VI and change the Signal Type to Square for the second VI. Wire both of the outputs into the waveform graph. A Merge Signals function will automatically be inserted.
d. Expand the Simulate Signal Express VIs to show another Input/Output. By default, error out should appear. Change this to Frequency by clicking on error out and choosing Frequency.

3. Save the VI as Multi-plot Graph.vi.

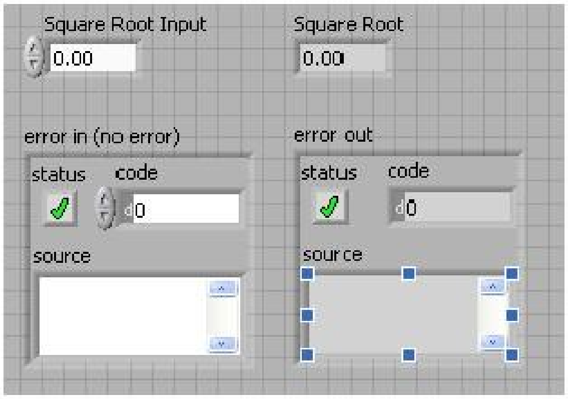
4. Display the front panel and run the VI.

5. Save and close the VI.

Exercise 2 - Error Clusters & Handling

Front Panel

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



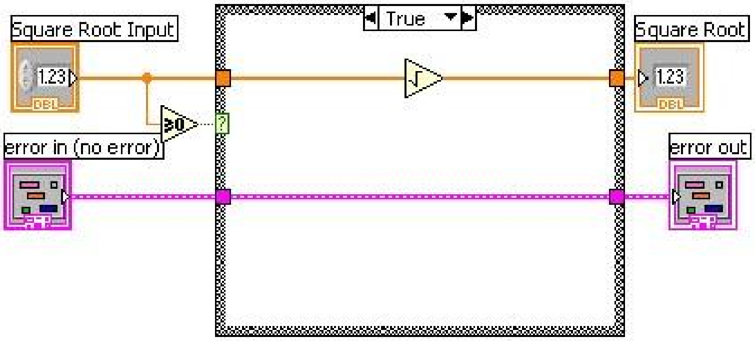
a. Create a numeric control and change the Label to Square Root Input. Create a numeric indicator for Square Root.

b. Place Error In 3D.ctl from Controls»All Controls»Arrays & Clusters.

c. Place Error Out 3D.ctl from Controls» All Controls»Arrays & Clusters.

Block Diagram

2. Build the following block diagram.



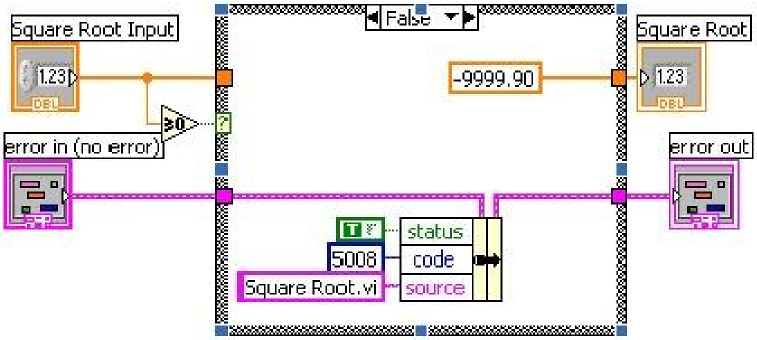
a. Place a Case Structure from Functions»Execution Control palette.

b. Place a Greater or Equal to 0? from the Functions»Arithmetic and Comparison»Comparison palette and wire it to the condition terminal of the case structure.

In the True Case:

c. Place the Square Root function from Functions»Arithmetic and Comparison»Numeric palette.

In the False Case:



d. Create a numeric constant from Functions»Arithmetic and Comparison»Numeric palette and type   9999.90.

e. Place the Bundle By Name from Functions»All Functions»Arrays & Clusters palette. Wire from Error in to the center terminal of Bundle by Name to make status show up. Create constants. Wire from the Error Out indicator to the output of Bundle By name.

3. Save the VI as Square Root.vi.
4. Display the front panel and run the VI.
5. Save and close the VI.

End of Exercise

Exercise 3- Simple State Machine

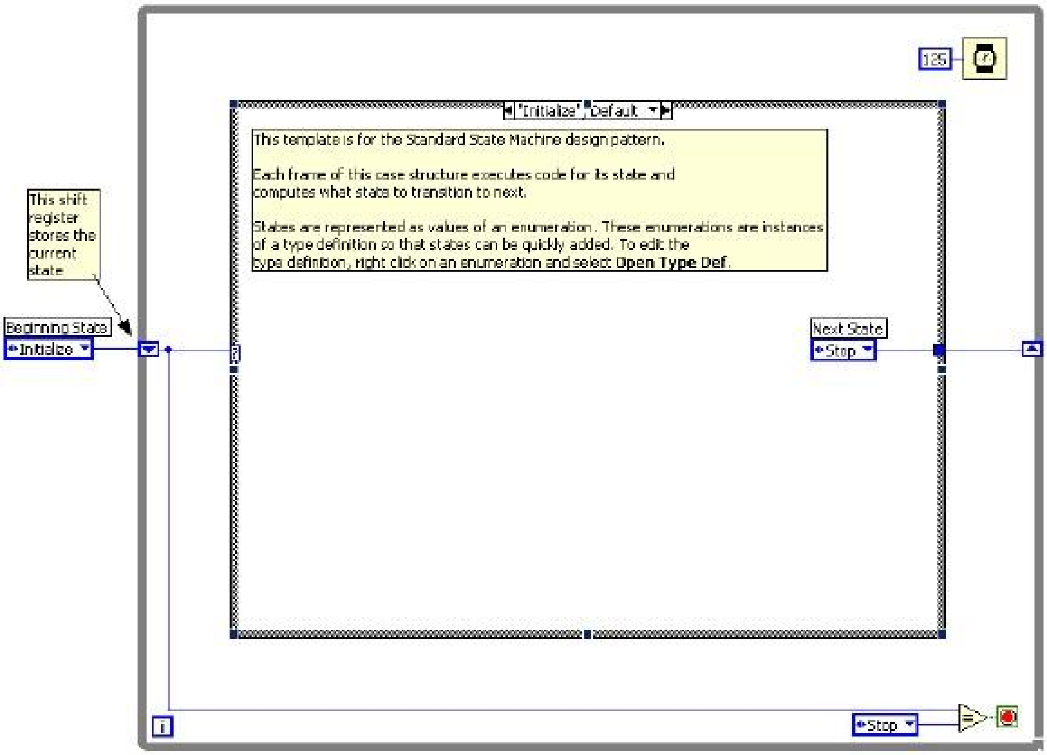
Create a VI using state machine architecture that simulates a simple test sequence. The VI will have an initial state, where it will display a pop   up message indicating that it is starting the test. Then it will proceed to the next case and then to the final state where it will ask the user whether to start over or end the test.

Front Panel

Rather than start from scratch, we will use a VI template to create our state machine.

1. From the initial LabVIEW screen click on Newé , and choose Standard State Machine, which is located under the VI from Template » Frameworks » Design Patterns heading.
2. Examine the template, and then save it in another directory before you begin working on it.

Block Diagram



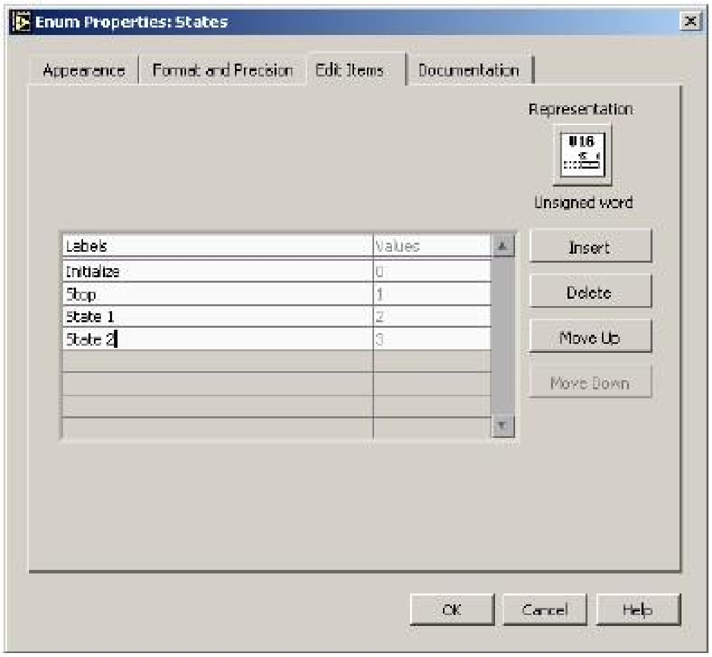
This shift register stores the current state

This template is for the Standard State Machine design pattern.

Each frame of this case structure executes code for its state and computes what state to transition to next.

States are represented as values of an enumeration. These enumerations are instances of a type definition so that states can be quickly added. To edit the type definition, right click on an enumeration and select **Open Type Def**.

3. Right click on the enum constant labeled Next State and select Open Type Def.
4. On the front panel of the StateMachinesStates.ctl Type Def VI, right click on the States enum control and choose Edit Items.
5. Add two more states. Call them ñState 1ò and ñState 2ò



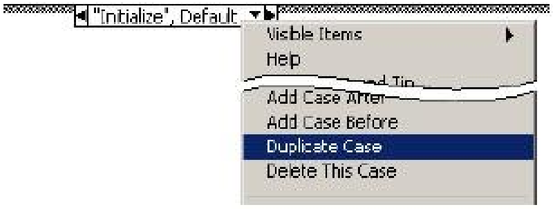
Enum Properties: States

Appearance | Format and Precision | Edit Items | Documentation

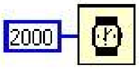
Representation: U16, Unsigned word

Labels	Values
Initialize	0
Stop	1
State 1	2
State 2	3

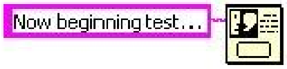
6. Close the State Machines.ctl Type Def Front panel and save the control with the default name when prompted.
7. Right click on the Case Selector Label of the case structure and choose Duplicate case. Do this one additional time so that there are four cases: Initialize, State 1, State 2, and Stop.



8. Change the value connected to the Wait function to 2000.

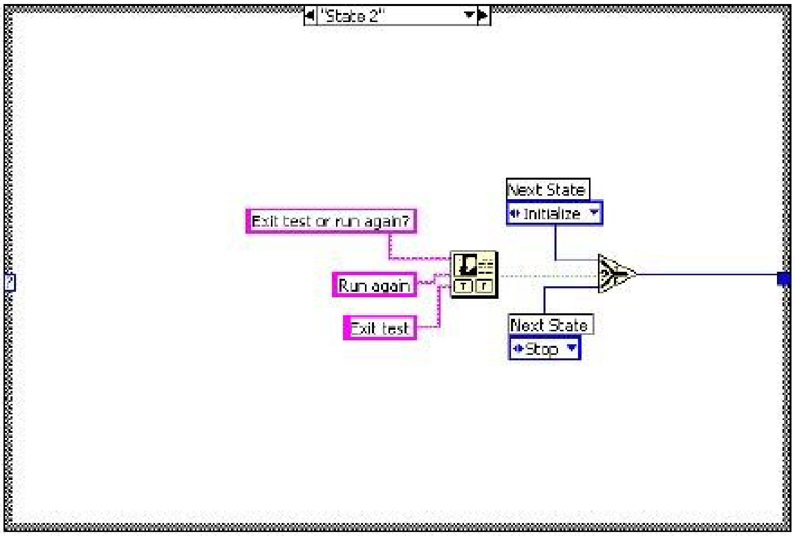


9. Right click on the shift register on the left side of the while loop and create an indicator. Change itõs name to ñCurrent Stateò.
10. In the ñInitializeò, Default case place a One Button Dialog function and wire a string constant into the Message input. Type ñNow beginning testé ò into the string constant.



11. Change the enum constant labled Next State to ñState 1ò.
12. Change to the next state in the case structure (ñState 1ò) and change the enum constant labled Next State to ñState 2ò.
13. Change to the next state (ñState2ò) and add the following code.
    a. Place a Select function and connect two enum constants
    (Tip: Copy the enum constants from one of the previous cases)

b. Place a Two Button Dialog and wire create the constants as illustrated below.



14. Run the VI.
15. Save and close the VI.

End of Exercise

Experiment No.4: Data Acquisition

Objective: To learn the data acquisition technique using LABVIEW.

Equipments Required: LABVIEW (software)

Data Acquisition

The NI USB 6009 multifunction I/O is a device that can be used to connect LabVIEW to external devices for Data Acquisition. The NI USB 6009 is a USB based data acquisition (DAQ) and control device with analog input and output and digital input and output.

The main features of NI USB 6009 are as follows:

ÅAnalog input (AI): 8 inputs with referenced single ended signal coupling or 4 inputs with differential signal coupling. Software configurable voltage ranges: ±20V, ±10V, ±5V, ±4V, ±2.5V, ±2V, ±1.25V, ±1V. Max sampling rate is 48kS/s (48000 samples per second). 14 bits AD converter (USB 6008: 12 bits).

ÅAnalog output (AO): 2 outputs. Voltage range is 0 to 5V (fixed). Output rate is 150Hz (samples/second). 12 bits DA converter.

ÅDigital input (DI) and digital output (DO): 12 channels which can be used as either DI or DO (configured individually). These 12 channels are organized in ports, with Port 0 having lines 0, .., 7, and Port 1 having lines 0, .., 3. Input low is between 0.3V and +0.8V. Input high is between 2.0V and +5.8V. Output low is below 0.8V. Output high is above 2V (with open drain and push pull as options).

ÅCounter: 32 bits: Counting on falling edge.

ÅOn-board voltage sources (available at individual terminals): 2.5V and 5.0V

ÅPower: USB 6009 is powered via the USB cable.

ÅConfiguring and testing: USB6009 can be configured and tested using MAX (Measurement and Automation Explorer) 4.0, which is installed with NI DAQ 8.0.

ÅApplication software: LabVIEW, C, or Visual Studio.

ÅPlatforms: Windows, Mac, Linux.

Installing the DAQ

After inserting the CD, the NI DAQmx 7.x installer should open automatically. If not, select Start»Run. Enter x:\setup.exe, where x is the letter of the CD drive. Complete the instructions in the installer, including rebooting the computer if necessary. For troubleshooting instructions, refer to the Hardware Installation/Configuration Troubleshooter at ni.com/support/install.

Note: Install your driver software before installing new hardware devices, so Windows can detect your device.

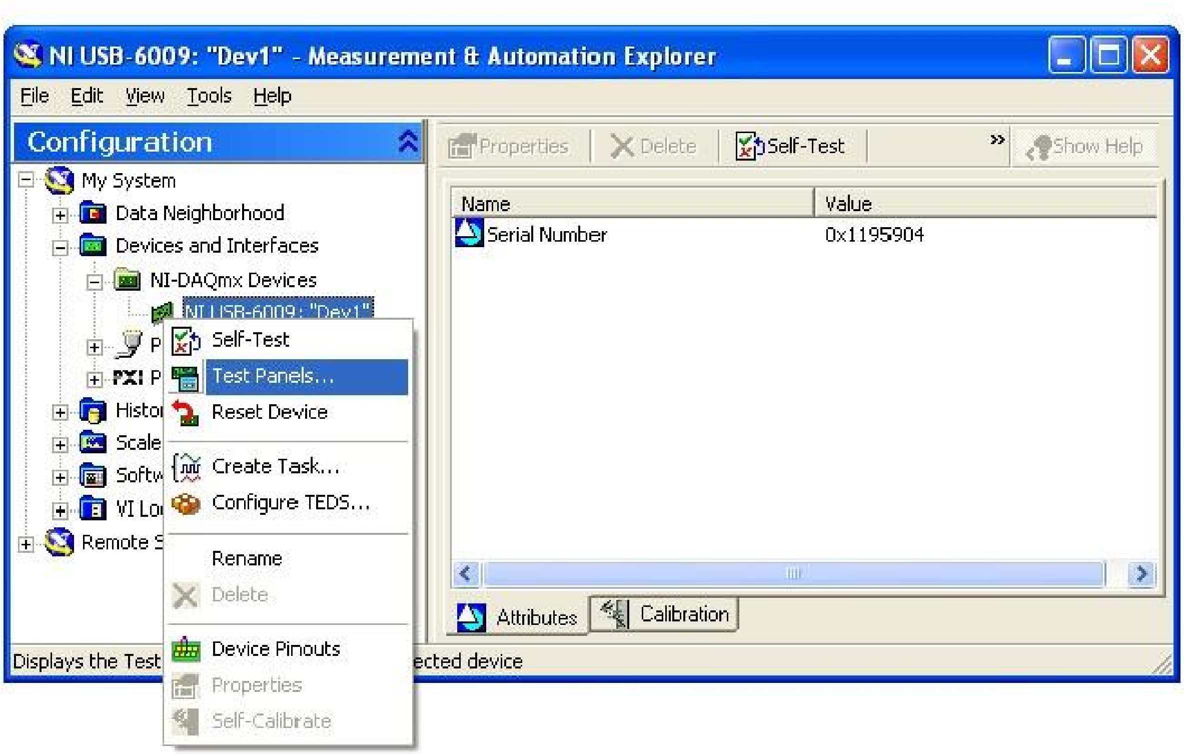
After installing the software, the actual device should be installed. This is done quite easily by simply plugging in the USB cable into PC and device, and then click Next on any dialog that appears. Next click Finish.

Note USB-6008/6009 Onlyð If this is the first time that a USB 6008/6009 device is installed on your computer, you might be prompted to install a USB 6xxx Firmware Loader. Please complete the Windows Hardware Wizard prompts to completely install this device.

Device Confirmation

Complete the following steps:

1. Double click the Measurement & Automation icon on the desktop to open MAX.
2. Expand Devices and Interfaces, then expand NI-DAQmx Devices.



Name	Value
Serial Number	0x1195904

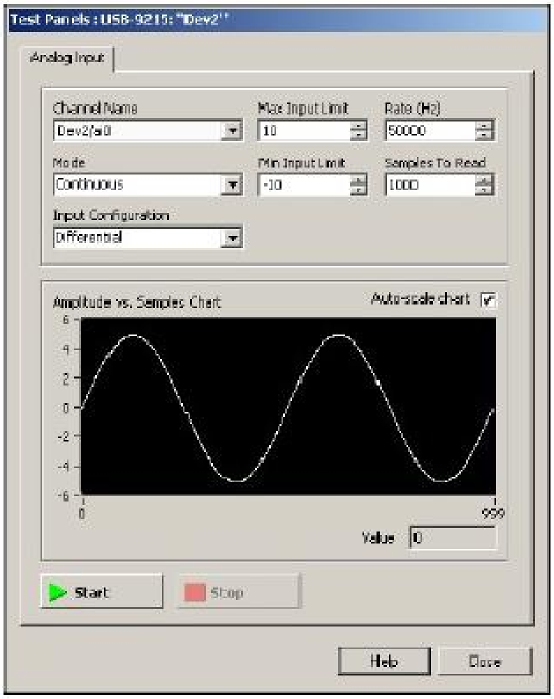
Displays the Test ected device

3. Check that your device appears under Devices and Interfaces. If your device does not appear, press <F5> to refresh the view in MAX. If the device is still not recognized, refer to ni.com/support/install for troubleshooting information.
4. Right click your device and select Self-Test.

If you need help during the self test, select Help»HelpTopics»NI-DAQmx and click Max Help for NI-DAQmx. When the self test finishes, a message indicates successful verification or if an error occurred.

You can also use a test panel for testing specific device functionality, such as the ability to acquire and generate signals. It can be done as follows:

1. In MAX, expand Devices and Interfaces»NI-DAQmx Devices.
2. Right click the device to test.
3. Select Test Panels to open a test panel for the selected device.



Test Panels : USB-9215: "Dev2"

Channel Name: Dev2/ai0

Max Input Limit: 10

Rate (Hz): 50000

Mode: Continuous

Min Input Limit: -10

Samples To Read: 1000

Input Configuration: Differential

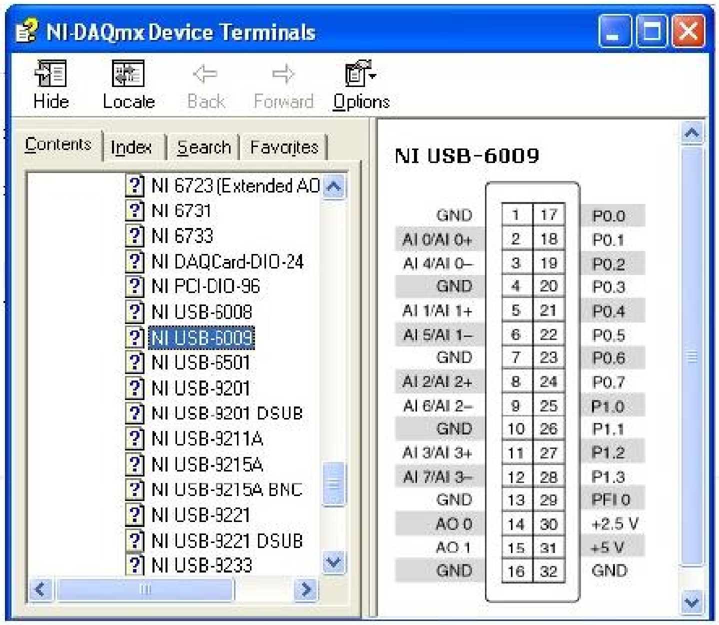
Auto-scale chart ☑

Value: 0

4. Click Start to test different functions of the device. Click Help for instructions on operating the test panels.
5. The test panel displays a message indicating whether an error occurred.

To see the terminals of the USB 6009, select Device Pinouts in the menu shown below. The terminals or pins are shown in the NI DAQmx
Device Terminals window.

The DAQ is now ready to connect
external devices to LabVIEW.



NI USB-6009

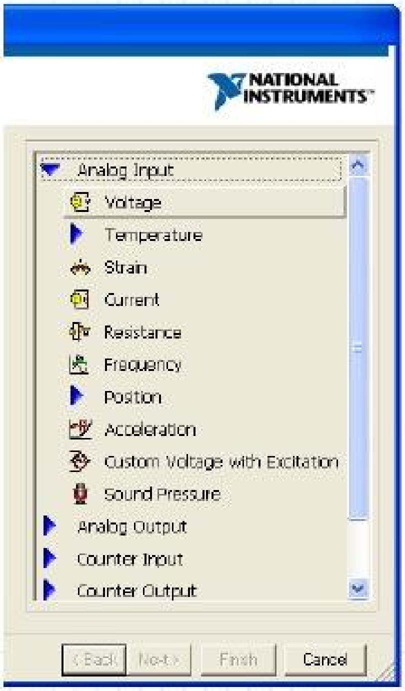
GND	1	17	P0.0
AI 0/AI 0+	2	18	P0.1
AI 4/AI 0–	3	19	P0.2
GND	4	20	P0.3
AI 1/AI 1+	5	21	P0.4
AI 5/AI 1–	6	22	P0.5
GND	7	23	P0.6
AI 2/AI 2+	8	24	P0.7
AI 6/AI 2–	9	25	P1.0
GND	10	26	P1.1
AI 3/AI 3+	11	27	P1.2
AI 7/AI 3–	12	28	P1.3
GND	13	29	PFI 0
AO 0	14	30	+2.5 V
AO 1	15	31	+5 V
GND	16	32	GND

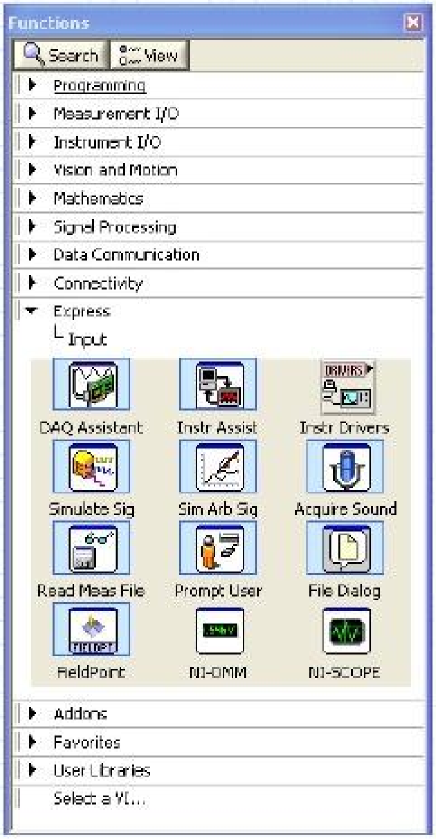
Exersise :

Reading a analog signal from a signal generator and displaying in Labview.

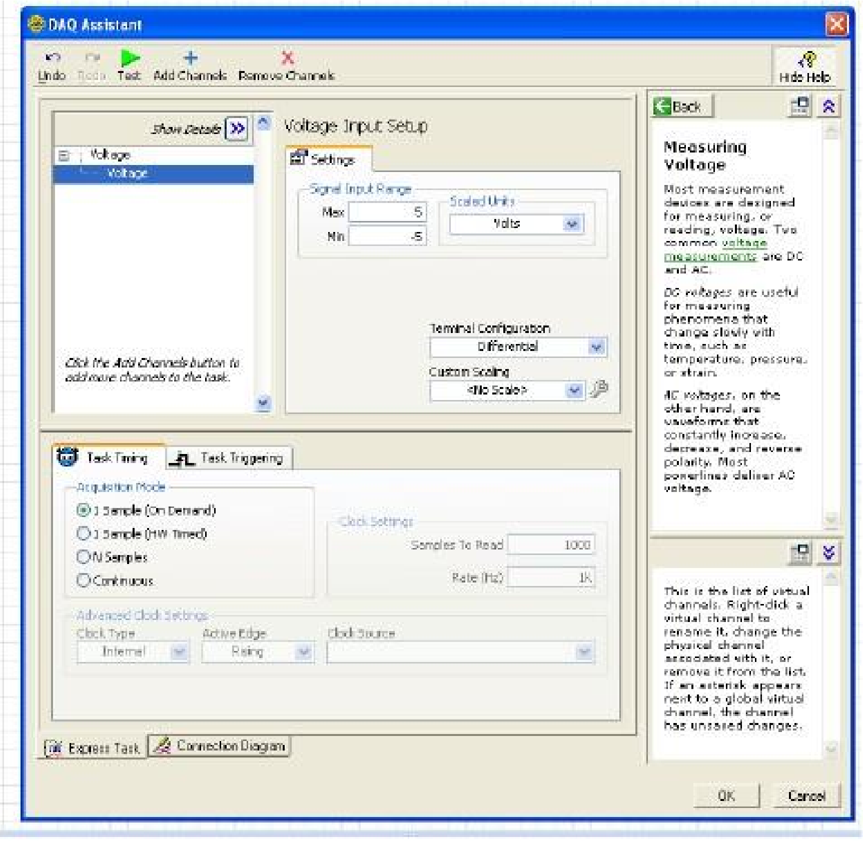
The purpose of this exersice is to teach you how to build a LabVIEW application to retrieve voltage input values using the NI USB 6009 data acquisition device.

1. Click ñNew VIò button to create a new blank LabVIEW program
2. Select Front Panel and enable Controls Palette to choose a ñWaveform Chartò indicator in ñGraphò group, add to front panel and name it ñVoltageò.
3. Switch to Block Diagram to verify that a new data terminal was created (named ñVoltageò).
4. In the Block diagram, open the Functions Palette to get the DAQ Assistant. The path goes as Functions>>Express>>Input>>DAQ Assistant.
   Note: If you have any trouble in finding a block use the Search function available just below the Title Bar of the Functions Palette.
5. When the DAQ Assistant is placed in the Block diagram window, a window with ñCretae Newé ò in the title bar appears. Select Analog Input and then Voltage.





6. The Physical Tab will appear showing the devices attached and the corresponding pins. Select the required pin and click Finish.
7. Another DAQ Assistant window will appear. Under the Voltage Input setup, Settings Tab select the appropriate maximum & minimum input signal range and units.(preferably, 5 to 5 volts). Also, in the Task Timing Tab under Acquisition mode select 1 Sample (On Demand).Click OK.



Click the Add Channels button to add more channels to the task.

Signal Input Range: Max 5, Min -5

Scaled Units: Volts

Terminal Configuration: Differential

Custom Scaling: <No Scale>

Acquisition Mode:
● 1 Sample (On Demand)
○ 1 Sample (HW Timed)
○ N Samples
○ Continuous

Clock Settings: Samples To Read 1000; Rate (Hz) 1k

Advanced Clock Settings: Clock Type Internal; Active Edge Rising; Clock Source

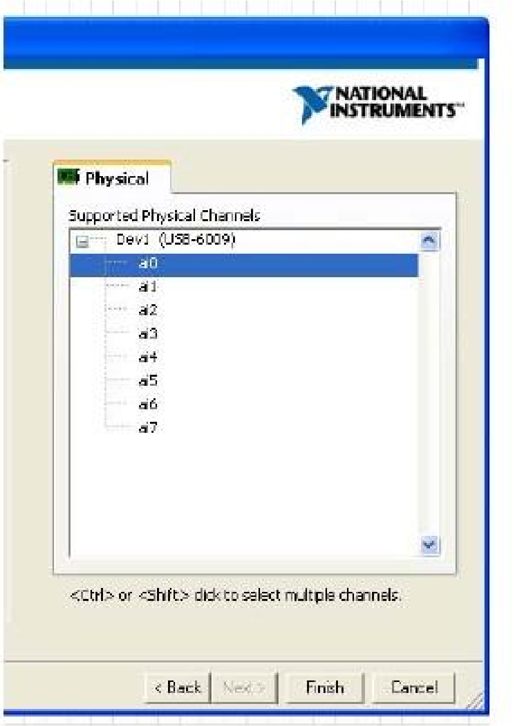
Measuring Voltage

Most measurement devices are designed for measuring, or reading, voltage. Two common voltage measurements are DC and AC.

DC voltages are useful for measuring phenomena that change slowly with time, such as temperature, pressure, or strain.

AC voltages, on the other hand, are waveforms that constantly increase, decrease, and reverse polarity. Most powerlines deliver AC voltage.

This is the list of virtual channels. Right-click a virtual channel to rename it, change the physical channel associated with it, or remove it from the list. If an asterisk appears next to a global virtual channel, the channel has unsaved changes.



Supported Physical Channels

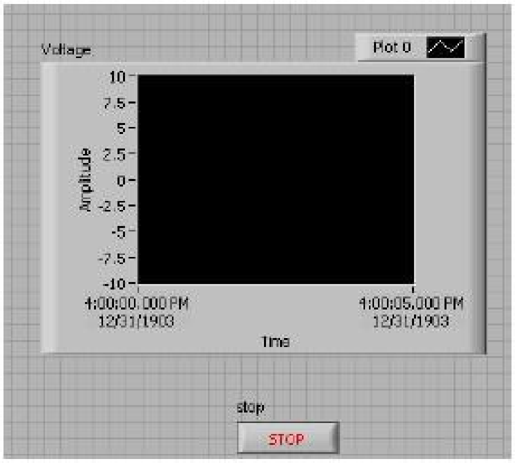
Dev1 (USB-6009)
- ai0
- ai1
- ai2
- ai3
- ai4
- ai5
- ai6
- ai7

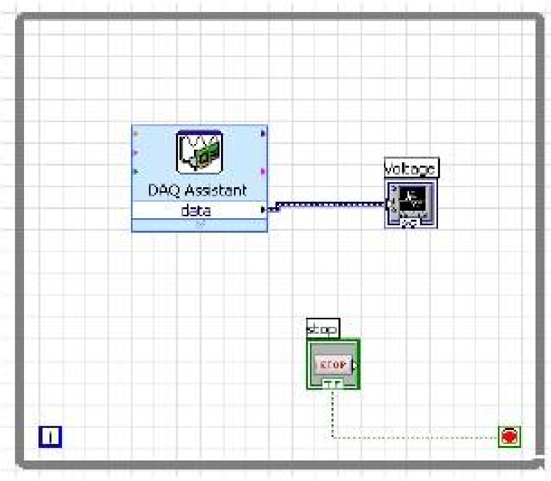
<Ctrl> or <Shift> click to select multiple channels.

8. Connect the data output from the DAQ Assistant to the Waveform chart terminal, using the wiring tool. If the wiring tool does not appear automatically on moving the cursor over the Data output of DAQ, go to View>>Tools Palette and select the wiring tool. To make wiring easier, you can invoke the Help window to support it. The Help window shows input and output parameters of each selected VI. (The way to display the Help window is to click on the Help menu selection and select the ñShow Context Helpò option)
9. You can change the window to the Front Panel window and click the ñRunò (or ñRun Continuouslyò) button to execute the VI.
10. If your waveform chart does not show the whole graph, you can change the scale of the Yaxis. To change the Y axis scale on the waveform chart, move the cursor to the graphic origin, click the left mouse button, and type in the desired value
11. The ñRun Continuouslyò button executes the VI continuously. You can also add a ñWhile Loopò to enhance your program. First, move mouse to the Block Diagram. Open the Functions Palette and select the Structures group and then the While Loop. Add the While Loop to the Block Diagram

and enlarge it to include the DAQ Assistant and the Voltage Waveform data terminal. On the front panel add a Stop button (from Controls>>Boolean>>Stop Button) to control the execution of the While Loop. In the block diagram, wire the stop button of the front panel to the stop button of the while loop.

12. After wiring, switch to Front Panel and press the RUN button to execute the VI.





Make sure that all your external wiring is connected properly and all devices have a common ground.

Exersise :

Sending an analog signal from the NI USB 6009 and displaying in a Oscilloscope.

The purpose of this exersice is to teach you how to build a LabVIEW application to send voltage outputs values from NI USB 6009 data acquisition cards.

1. Click ñNew VIò button to create a new blank LabVIEW program
2. Select Front Panel and enable Controls Palette to choose a ñWaveform Chartò indicator in ñGraphò group, add to front panel and name it ñVoltageò.
3. Switch to Block Diagram to verify that a new data terminal was created (named ñVoltageò).
4. In the Block diagram, open the Functions Palette to get the DAQ Assistant. The path goes as Functions>>Express>>Input>> DAQ Assistant.
   Note: If you have any trouble in finding a block use the Search function available just below the Title Bar of the Functions Palette.
5. When the DAQ Assistant is placed in the Block diagram window, a window with ñCretae Newé ò in the title bar appears. Select Analog Output and then Voltage.
6. The Physical Tab will appear showing the devices attached and the corresponding pins. Select the required pin and click Finish.
7. Another DAQ Assistant window will appear. Under the Voltage Output setup, Settings Tab select the appropriate maximum & minimum input signal range and units.(preferably, 0 to 10 volts). Also, in the Task Timing Tab under Acquisition mode select 1 Sample (On Demand).Click OK.
8. Generate a sinusoidal signal using the express VI using the ñSimulate sigò express VI. The path to access these ñFunctionsò >> Express>> ñInputsò >> ñSimulate sigò. Select a type of signal, its

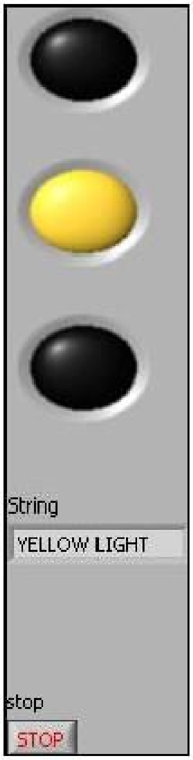
amplitude and frequency. Select a number of ñsamples per secondò as 1000 and uncheck the automatic for number of samples and select ñ1ò as number of samples. Now connect the output of ñSimulate sigò to the voltage input of the DAQ Assistant.

9. Also connect the Voltage waveform to the output of ñSimulate sigò.
10. Complete your wiring and run to see the result in your graph as well as the oscilloscope.
11. The ñRun Continuouslyò button executes the VI continuously. You can also add a ñWhile Loopò to enhance your program. First, move mouse to the Block Diagram. Open the Functions Palette and select the Structures group and then the While Loop. Add the While Loop to the Block Diagram and enlarge it to include the DAQ Assistant and the Voltage Waveform data terminal. On the front panel add a Stop button (from Controls>>Boolean>>Stop Button) to control the execution of the While Loop. In the block diagram, wire the stop button of the front panel to the stop button of the while loop.
12. After wiring, switch to Front Panel and press the RUN button to execute the VI.

Make sure that all your external wiring is connected properly and all devices have a common ground.

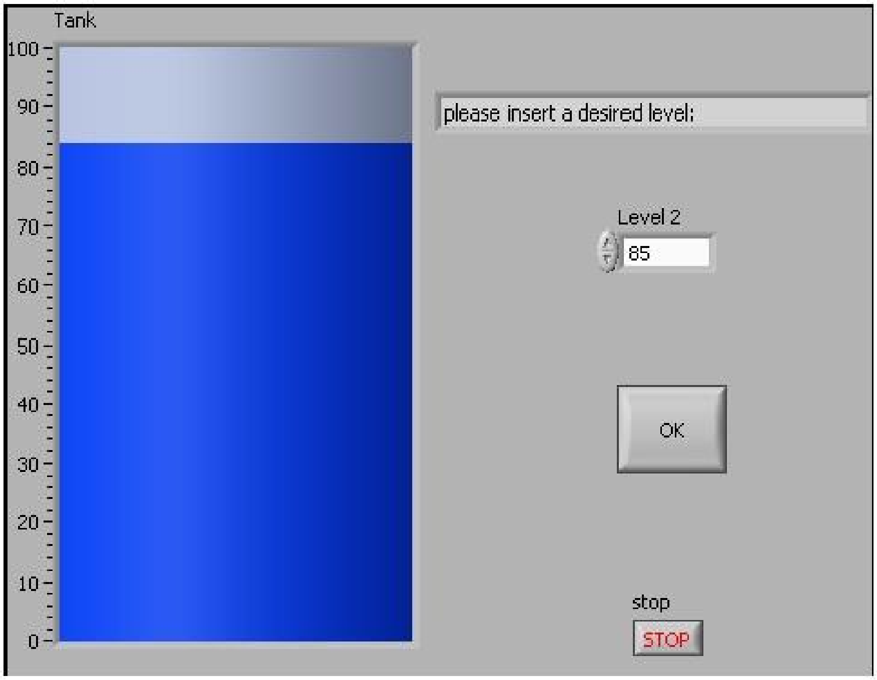
Exercise: Traffic Light

Construct a VI that will display a traffic light in the Front Panel. See figure below. Light transitions are timed. Let red take 2s, yellow 1s and green 3s. Try to utilize your knowledge of LabVIEW in constructing this VI.



Exercise: Filling and emptying a Tank

Build a VI that display a Front Panel similar to which shown below. The front panel should have a Numeric Control to set a desired tank level. The filling and emptying tasks should be displayed gradually. Your VI should decide by itself if it is to fill or to empty the tank. An OK button is to start the task.



please insert a desired level:

Level 2: 85

Experiment No. 5: Digital Input / Output

Objective: To learn sending and receive Digital Signals using LABVIEW commands.

Equipments Required: LABVIEW (software)

Digital signals are digital representations of discretetime signals, which are often derived from analog signals. An analog signal is a datum that changes over timeð say, the temperature at a given location; the depth of a certain point in a pond; or the amplitude of the voltage at some node in a circuitð that can be represented as a mathematical function, with time as the free variable (abscissa) and the signal itself as the dependent variable (ordinate). A discrete time signal is a sampled version of an analog signal: the value of the datum is noted at fixed intervals (for example, every microsecond) rather than continuously.

If individual time values of the discrete time signal, instead of being measured precisely (which would require an infinite number of digits), are approximated to a certain precisionð which, therefore, only requires a specific number of digitsð then the resultant data stream is termed a digital signal. The process of approximating the precise value within a fixed number of digits, or bits, is called quantization.
In conceptual summary, a digital signal is a quantized discrete time signal; a discrete time signal is a sampled analog signal.

In most applications, digital signals are represented as binary numbers, so their precision of quantization is measured in bits. Suppose, for example, that we wish to measure a signal to two significant decimal digits. Since seven bits, or binary digits, can record 128 discrete values (viz., from 0 to 127), those seven bits are more than sufficient to express a range of one hundred values.

Exercise: Sending and receiving digital signals

The purpose of this exercise is to teach you how to build a LabVIEW application to send and receive digital values from NI USB 6009 data acquisition cards.

ÅPart 1: Sending

1. Click ñNew VIò button to create a new blank LabVIEW program
2. Select Front Panel and enable Controls Palette to choose a ñPush Buttonò indicator in ñBooleanò group (you can also use a toggle or slide switch, etc. as well), add to front panel and name it ñStatusò.
3. Switch to Block Diagram to verify that a new data terminal was created (named ñStatusò).
4. In the Block diagram, open the Functions Palette to get the DAQ Assistant. The path goes as Functions>>Express>>Input>> DAQ Assistant.
5. When the DAQ Assistant is placed in the Block diagram window, a window with ñCreate Newé ò in the title bar appears. Select Digital I/O and then Line Output.
6. The Physical Tab will appear showing the devices attached and the corresponding pins. Select the required pin(s) and click Finish.
7. Another DAQ Assistant window will appear. Under the Task Timing Tab under Acquisition mode select 1 Sample (On Demand).Click OK.

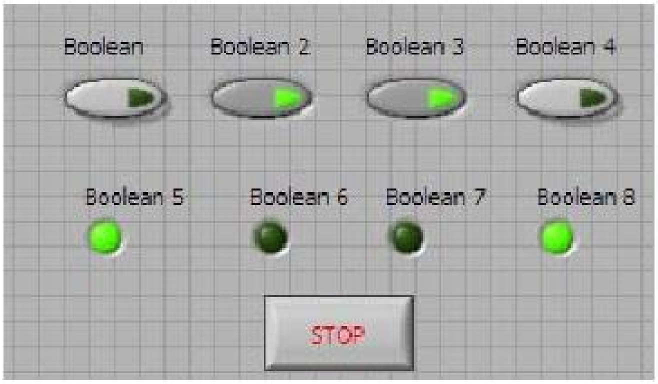
8. If you want to send more than 1 signal at a time, then first put more Boolean switches/buttons on the front panel. In the Block diagram, connect all the Boolean terminals into one using ñFunctionsò >> Programming>> ñArraysò >> ñBuild arraysò. Extend the size of the array according to the number of signals to send.
9. Next connect the array output to the data pin of the DAQ assistant. Note: Make sure that the correct pin numbers are assigned to the Boolean switches in the DAQ Assistant.
10. You can also add LEDs (from Controls>>Boolean) and connect them to the switches in the block diagram if you want to view the result on the front panel as well.
11. The ñRun Continuouslyò button executes the VI continuously. You can also add a ñWhile Loopò to enhance your program.
12. After wiring, switch to Front Panel and press the RUN button to execute the VI.

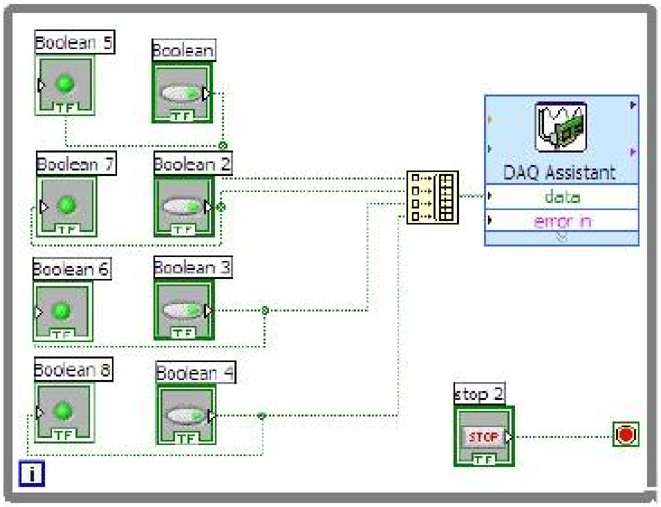
Part 2: Receiving

For testing Digital input,

- Select Digital I/O>>Line Input in the DAQ assistant
- And instead of Boolean switches use LEDs.
- Also Index array can be used.

Note: In hard wire, real LEDs and switches can be used for demonstration. Make sure that all your external wiring is connected properly and all devices have a common ground.





Good Luck!

Prepared By: Mohammad Obaidullah Khan, Lecturer in Electrical Engineering, Al Imam Mohammad Bin Saud Islamic University, Riyadh , KSA