Introduction to LabVIEW

1 Introduction

1.1 Aims

The following material is a short introduction to LabVIEW and it aims for you to:

- familiarise with the LabVIEW programming environment including front panel and block diagram;
- familiarise with the programming structures and data types that exist in LabVIEW;
- understand the data flow in LabVIEW;
- and be able to navigate the help utilities.

It is estimated that this introduction session takes between 1.5 and 2 hours.

Have Fun,

Mohammed El-Hajjar
Rob Maunder
Michael Ng

1.2 Installing LabVIEW

All university students are covered by a LabVIEW campus licence. You can download the latest LabVIEW and install it on your own windows machines by navigating to the iSolutions Software Distribution Service: https://www.software.soton.ac.uk/.

1.3 Navigating LabVIEW

LabVIEW programs are called Virtual Instruments and are normally referred to as VIs. A LabVIEW VI contains a front panel and a block diagram.

The front panel window is the user interface for the VI. Figure 1 shows an example of a front panel window. The front panel is built from controls and indicators, which correspond to the interactive input and output of the VI, respectively.

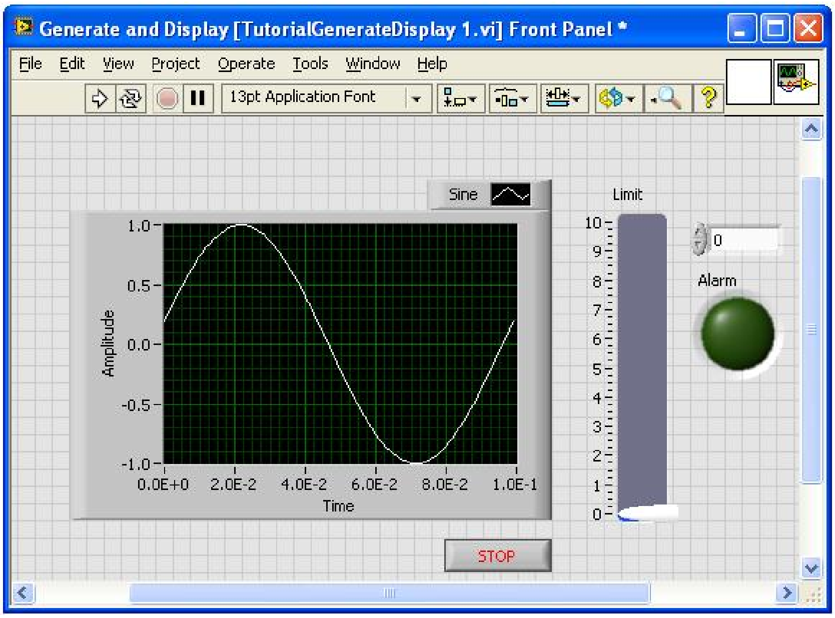
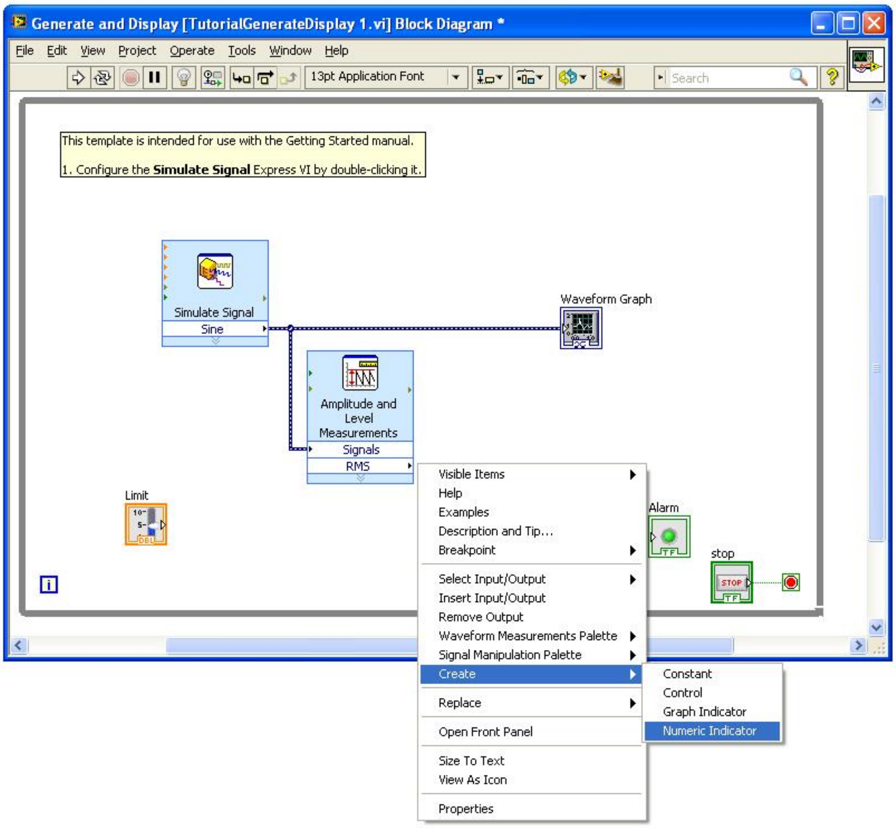


Figure 1 VI front panel.

The block diagram is formed of the graphical representation of the code for the functions in the front panel. Figure 2 shows an example block diagram window.



This template is intended for use with the Getting Started manual.

1. Configure the **Simulate Signal** Express VI by double-clicking it.

Figure 2 VI block diagram.

1.4 Dataflow in LabVIEW

LabVIEW follows a dataflow model for running VIs. A block diagram node executes when it receives all required inputs. When a node executes, it produces output data and passes the data to the next node in the dataflow path. A node with multiple outputs produces all the outputs at the same time after the node finishes execution. The movement of the data through the node determines the execution order of the VIs and functions in the block diagram.

To elaborate the idea of dataflow programming further, consider the example block diagram shown in Figure 3. In Figure 3, the block diagram executes from left to right, not because the objects are placed in that order, but because the "subtract" function cannot execute until the "add" function finishes execution and passes its output data as an input to the "subtract" function.

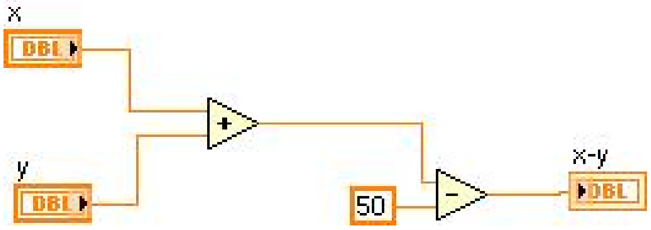


Figure 3 Dataflow programming example.

Figure 4 shows a different example with two code segments. In Figure 4, it is not possible to determine which of the "add", "random number" and "divide" functions execute first. The input to the "add" and "divide" functions are available at the same time and the "random number" function has no input.

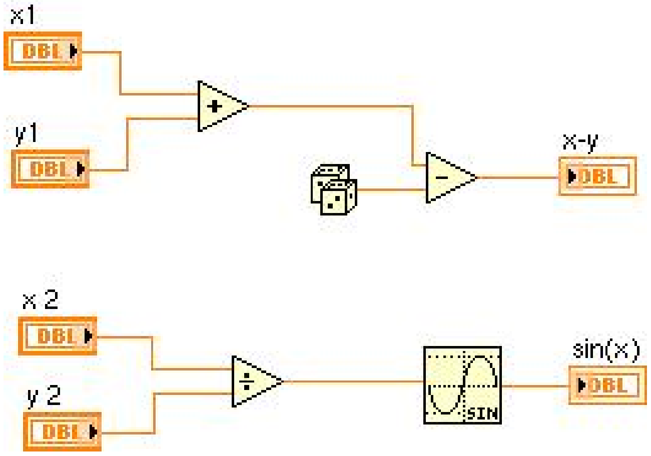
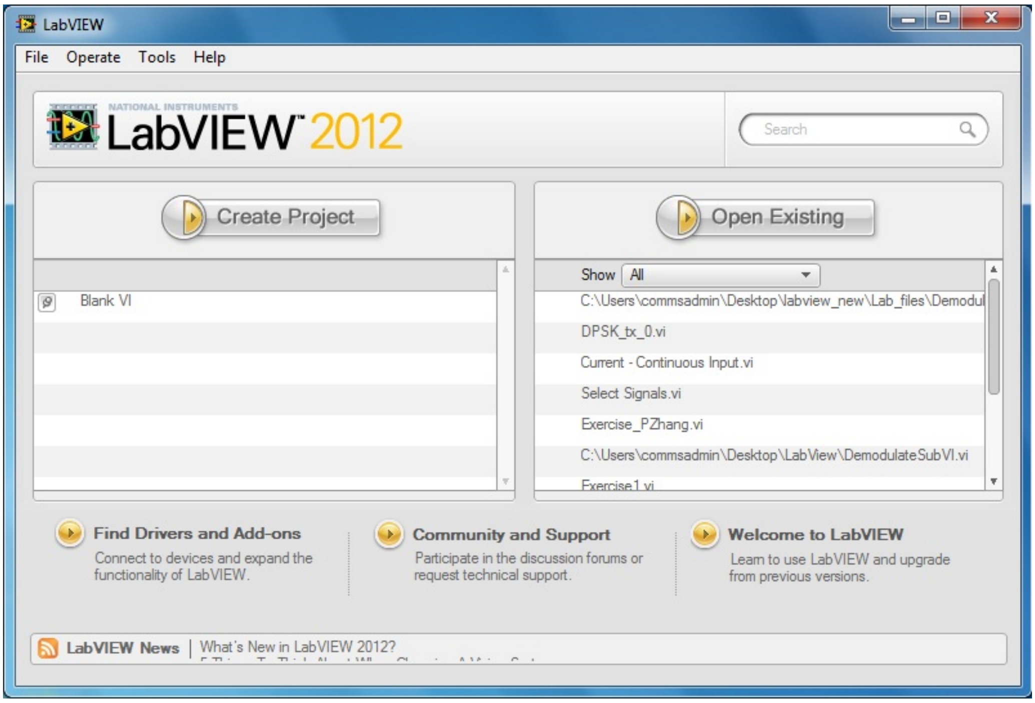


Figure 4 Dataflow example of multiple code segments.

2 Exercise 1

In this exercise you will create a VI that simulates an analogue signal and plots it on a waveform graph. The VI will test the input values against a user-specified limit and light an LED if the input value exceeds that limit.

Launch LabVIEW; you will get a splash screen like the one shown in Figure 5.


LabVIEW
File Operate Tools Help
NATIONAL INSTRUMENTS LabVIEW 2012
Search
Create Project
Open Existing
Blank VI
Show All
C:\Users\commsadmin\Desktop\labview_new\Lab_files\Demodul
DPSK_tx_0.vi
Current - Continuous Input.vi
Select Signals.vi
Exercise_PZhang.vi
C:\Users\commsadmin\Desktop\LabView\DemodulateSubVI.vi
Exercise1.vi
Find Drivers and Add-ons
Community and Support
Welcome to LabVIEW
LabVIEW News | What's New in LabVIEW 2012?

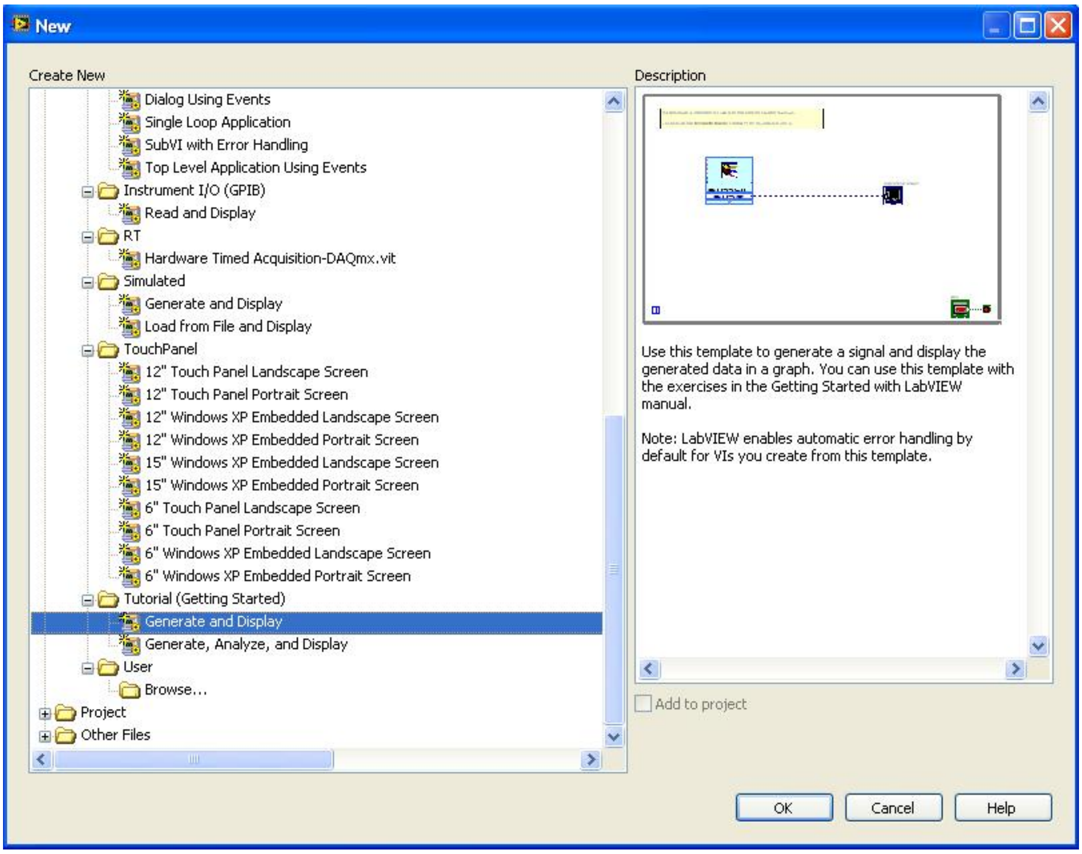
Connect to devices and expand the functionality of LabVIEW.

Participate in the discussion forums or request technical support.

Learn to use LabVIEW and upgrade from previous versions.

Figure 5 LabVIEW Getting Started window.

Navigate to and Select **File>>New>>From Template>>Tutorial (Getting Started) >>Generate and Display** and click OK as shown in Figure 6.


New
Create New
Dialog Using Events
Single Loop Application
SubVI with Error Handling
Top Level Application Using Events
Instrument I/O (GPIB)
Read and Display
RT
Hardware Timed Acquisition-DAQmx.vit
Simulated
Generate and Display
Load from File and Display
TouchPanel
12" Touch Panel Landscape Screen
12" Touch Panel Portrait Screen
12" Windows XP Embedded Landscape Screen
12" Windows XP Embedded Portrait Screen
15" Windows XP Embedded Landscape Screen
15" Windows XP Embedded Portrait Screen
6" Touch Panel Landscape Screen
6" Touch Panel Portrait Screen
6" Windows XP Embedded Landscape Screen
6" Windows XP Embedded Portrait Screen
Tutorial (Getting Started)
Generate and Display
Generate, Analyze, and Display
User
Browse...
Project
Other Files
Description
□ Add to project
OK
Cancel
Help

Use this template to generate a signal and display the generated data in a graph. You can use this template with the exercises in the Getting Started with LabVIEW manual.

Note: LabVIEW enables automatic error handling by default for VIs you create from this template.

Figure 6 Opening a VI from template.

Two windows appear: the grey window is the Front Panel and the white one is the Block Diagram. The Front Panel contains the parts of your VI used for the User Interface and presenting information, whereas the Block Diagram contains the code that controls the functionality of the VI. You can toggle between the two windows by selecting **Window>>Show Block Diagram** or **Window>>Show Front Panel**. You can also switch between the windows by pressing <Ctrl+E> on your keyboard.

Examine the Front Panel and Block Diagram of this Template VI. The Front Panel contains a Waveform Chart and a STOP button as shown in the Figure 7.

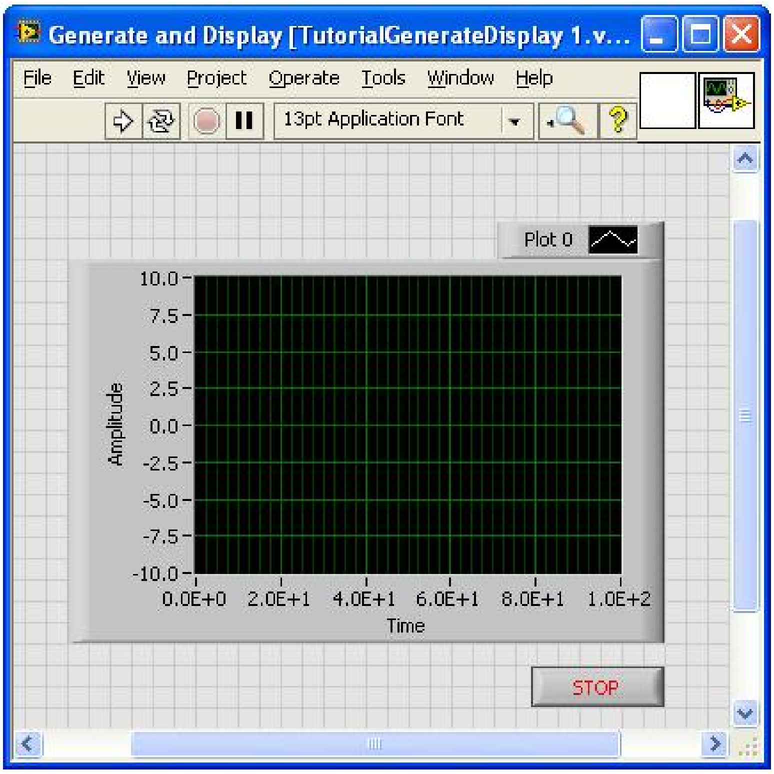
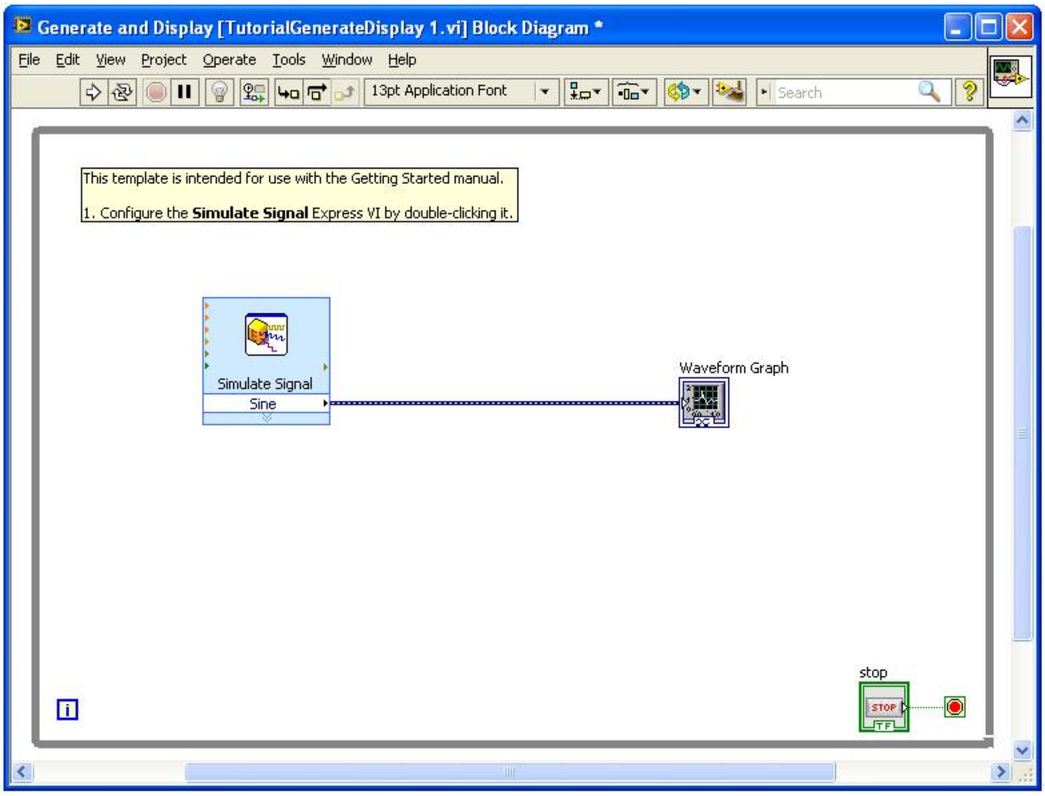


Figure 7 Front Panel Window

The Block Diagram in Figure 8 contains a Simulate Signal VI, which is currently configured to simulate a sine wave and plot it to the chart.


Generate and Display [TutorialGenerateDisplay 1.vi] Block Diagram *
File Edit View Project Operate Tools Window Help
13pt Application Font
Search
Simulate Signal
Sine
Waveform Graph
stop
STOP
TF

This template is intended for use with the Getting Started manual.

1. Configure the **Simulate Signal** Express VI by double-clicking it.

Figure 8 Block Diagram Window.

Switch back to the Front Panel by pressing <Ctrl+E>. Since the Run button (the white arrow at the top left corner) is solid, you can run this VI as it is. Click the Run button and examine the operation of the VI. When you are finished, click the STOP button on the Front Panel to stop running the VI.

Note: As you will see later in the exercise, when the Run button in the upper left corner of both the Front Panel and the Block Diagram changes from a solid white arrow to a broken grey arrow, this new icon indicates that the VI is currently not executable. Pressing it will reveal the programmatic errors that are keeping it from executing.

Now we can add some functionality to this basic VI. We will modify the VI to flash an alarm whenever the signal value is above a certain level. Open the Controls palette (if it is not open already) by **right-clicking** the Front Panel window. A snapshot of the controls palette is shown in Figure 9.

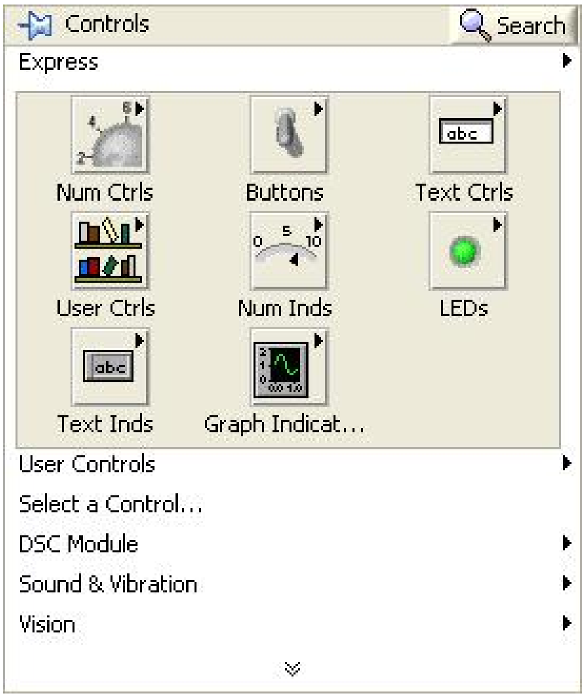


Figure 9 Controls Palette Snapshot.

Navigate to the **Express** palette and click on the **Numeric Controls** sub-palette. Select a **Vertical Pointer Slide** to be placed on the Front Panel. To do this, click the **Vertical Pointer Slide** and drag it onto the Front Panel. A snapshot of the Numeric Controls sub-palette is shown in Figure 10.

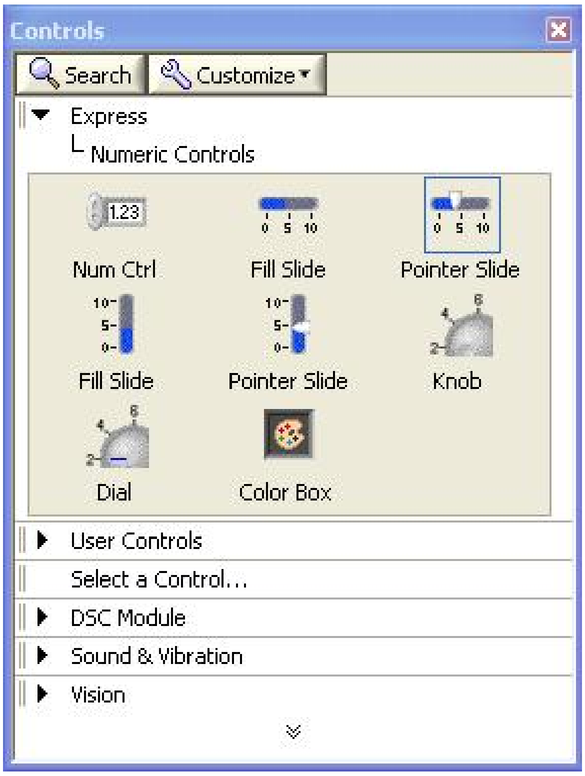


Figure 10 Numeric Controls Sub-Palette.

Click the Express menu item on the Controls palette to return to the Express Controls palette.Then click the **LEDs** sub-palette shown in Figure 11, and place a **Round LED** on the Front Panel.

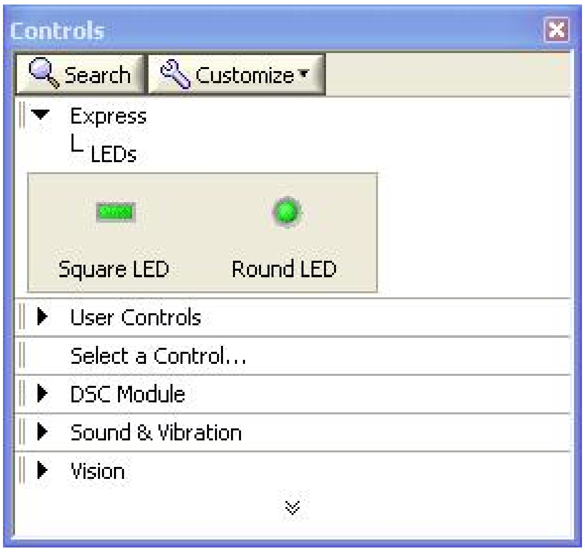
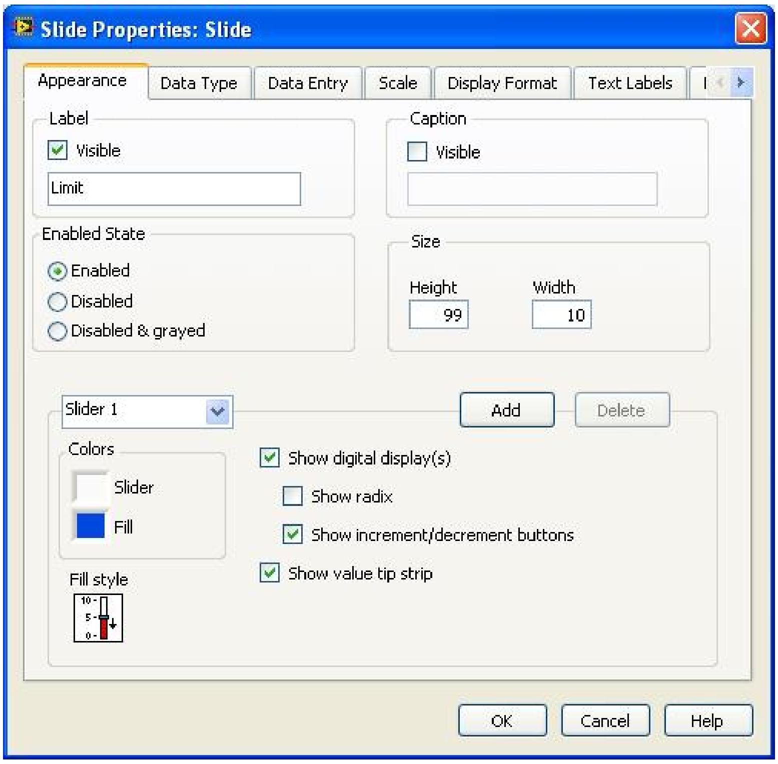


Figure 11 LEDs Sub-Palette.

Right-click the **Vertical Pointer Slide** and select **Properties**. A property page will appear. Make the following changes on the **Appearance** tab and click OK to apply the changes. A snapshot of the properties window is shown in Figure 12.

Label: Limit

Slider 1: Check Show digital display(s)



Slide Properties: Slide

Appearance | Data Type | Data Entry | Scale | Display Format | Text Labels

Label
☑ Visible
Limit

Caption
□ Visible

Enabled State
● Enabled
○ Disabled
○ Disabled & grayed

Size
Height 99
Width 10

Slider 1
Add Delete

Colors
Slider
Fill

Fill style

☑ Show digital display(s)
□ Show radix
☑ Show increment/decrement buttons
☑ Show value tip strip

OK Cancel Help

Figure 12 Vertical Pointer Slide Properties.

Right-click the Round LED labelled **Boolean** and select **Properties**. Examine the different properties that can be modified. On the **Appearance** tab, change the label from Boolean to **Alarm**. Click **OK** to apply your change. Move the objects on the Front Panel so it resembles Figure 13.

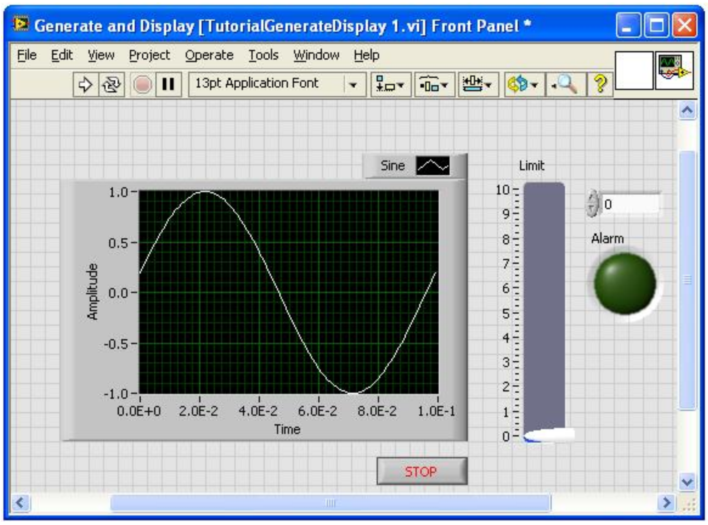

Generate and Display [TutorialGenerateDisplay 1.vi] Front Panel *
File Edit View Project Operate Tools Window Help
13pt Application Font
Sine
Amplitude
1.0
0.5
0.0
-0.5
-1.0
0.0E+0 2.0E-2 4.0E-2 6.0E-2 8.0E-2 1.0E-1
Time
Limit
10 9 8 7 6 5 4 3 2 1 0
0
Alarm
STOP

Figure 13 Example Front Panel.

Switch to the Block Diagram by pressing <Ctrl+E>. Double-click the **Simulate Signal** Express VI to bring up its properties window. Examine the different properties you can modify. Change the **Amplitude** of the signal to **10**. Click **OK** to apply this change and to close the properties window.

Bring up the **Functions** palette by **right-clicking** on the Block Diagram. Select **Express>>Signal Analysis** as shown in Figure 14 and place the **Amp&Level** Express VI on the Block Diagram by dragging and dropping as before.

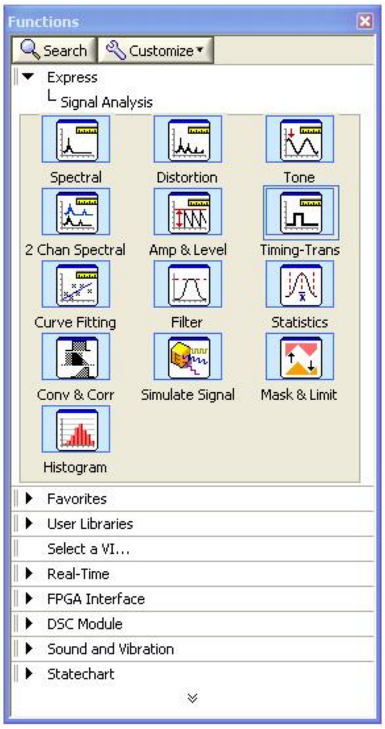
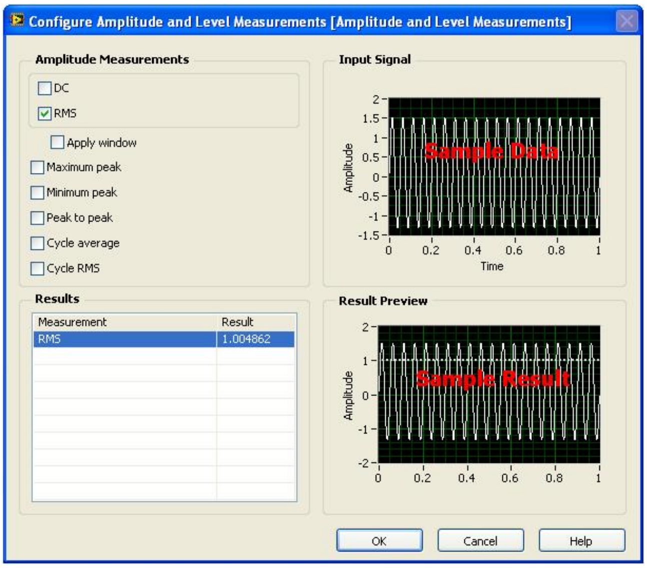


Figure 14 Signal Analysis Palette.

When you place an Express VI on the Block Diagram a dialog box appears so that you can configure the function as per your needs. For this function select **RMS** as shown in Figure 15.



Configure Amplitude and Level Measurements [Amplitude and Level Measurements]

Amplitude Measurements

- □ DC
- ☑ RMS
  - □ Apply window
- □ Maximum peak
- □ Minimum peak
- □ Peak to peak
- □ Cycle average
- □ Cycle RMS

Results

Measurement	Result
RMS	1.004862

Figure 15 Configure the Amp&Level Express VI.

You can connect Controls, Functions, and Indicators on the Block Diagram by pointing to an object and clicking on its terminal when the cursor changes to a spool of wire. You can then move the cursor to the terminal of the object you want to connect it to and click again. Following this, connect the **Limit** control to the **Alarm** indicator as shown in Figure 16.

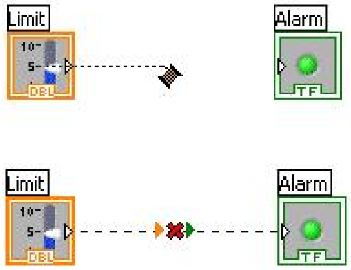


Figure 16

Note: The Run button in the upper left corner of both the Front Panel and the Block Diagram has changed from a solid white arrow, to a broken grey arrow. This new icon indicates that the VI is currently not executable. If you click the Run button when it is solid and white, it runs the VI. Clicking it when it is broken and grey brings up a dialog box that will help you debug the VI.

Click the Run button now. The resulting dialog box shows that, in this case, the error results from connecting terminals of two different types. Since the **Limit** control is a Numeric type and the **Alarm** indicator is a Boolean type, we cannot wire these two terminals together. Highlight the error by clicking it, and then click **Show Error**. LabVIEW will highlight the location of the error.

Notice that the wire between **Limit** and **Alarm** is dashed and a red "X" is displayed on it as shown in Figure 17.

To delete this broken wire, press <Ctrl+B>. This keyboard shortcut removes all broken wires from the Block Diagram.

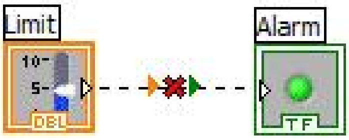
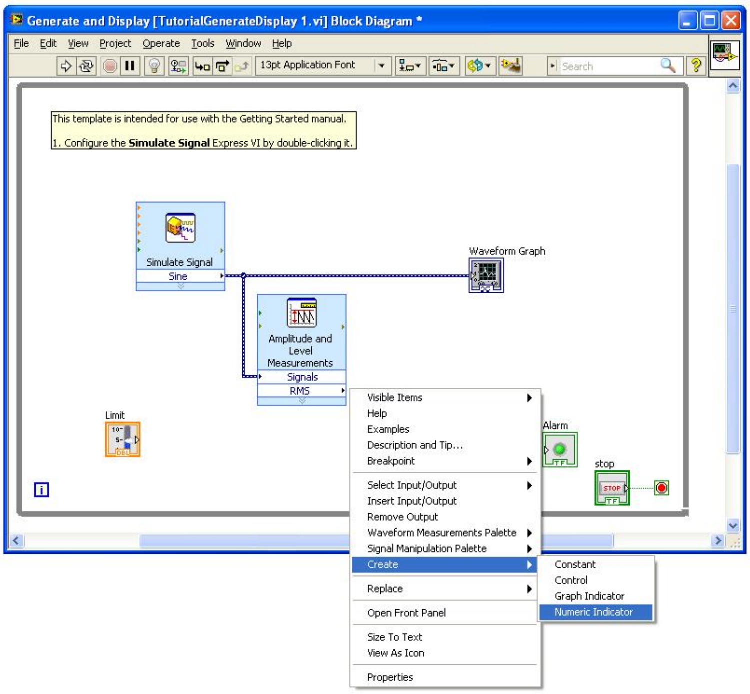


Figure 17

Wire the **output** of the **Simulate Signal** VI to the **Signals** input on the **Amplitude and Level Measurements** VI. Right-click on the **RMS** output and select **Create>>Numeric Indicator** from the list as shown in Figure 18.



This template is intended for use with the Getting Started manual.

1. Configure the **Simulate Signal** Express VI by double-clicking it.

Figure 18 Example Block Diagram.

Bring up the functions palette by right-clicking the Block Diagram. Select **Arithmetic & Comparison >> Comparison** and place the **Comparison** Express VI shown in Figure 19 on the diagram.

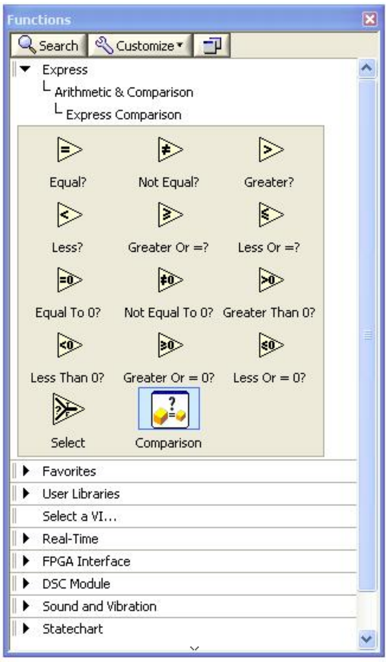
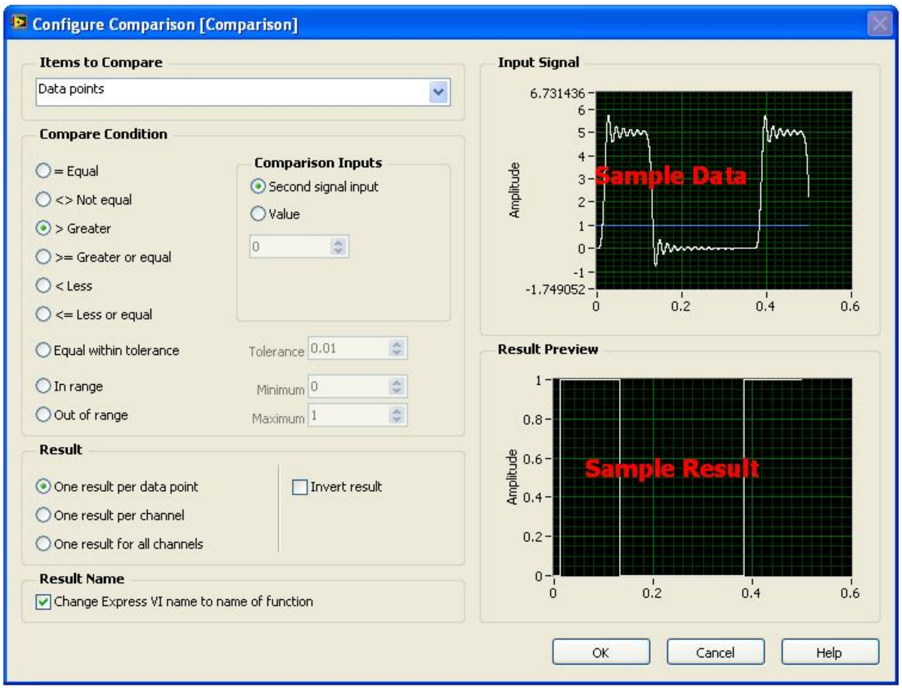


Figure 19 Comparison Express VI.

When you place the Comparison Express VI on the Block Diagram, a dialog box appears that lets you configure what type of comparison you will be doing. Make the following selections as shown in Figure 20, then click OK to apply these changes and to close the dialog box.

Compare Condition: Greater

Comparison Inputs: Second signal input


Configure Comparison [Comparison]

Items to Compare

Data points

Compare Condition

○ = Equal
○ <> Not equal
● > Greater
○ >= Greater or equal
○ < Less
○ <= Less or equal

Comparison Inputs

● Second signal input
○ Value
0

○ Equal within tolerance Tolerance 0.01

○ In range Minimum 0

○ Out of range Maximum 1

Result

● One result per data point
○ One result per channel
○ One result for all channels

□ Invert result

Result Name

☑ Change Express VI name to name of function

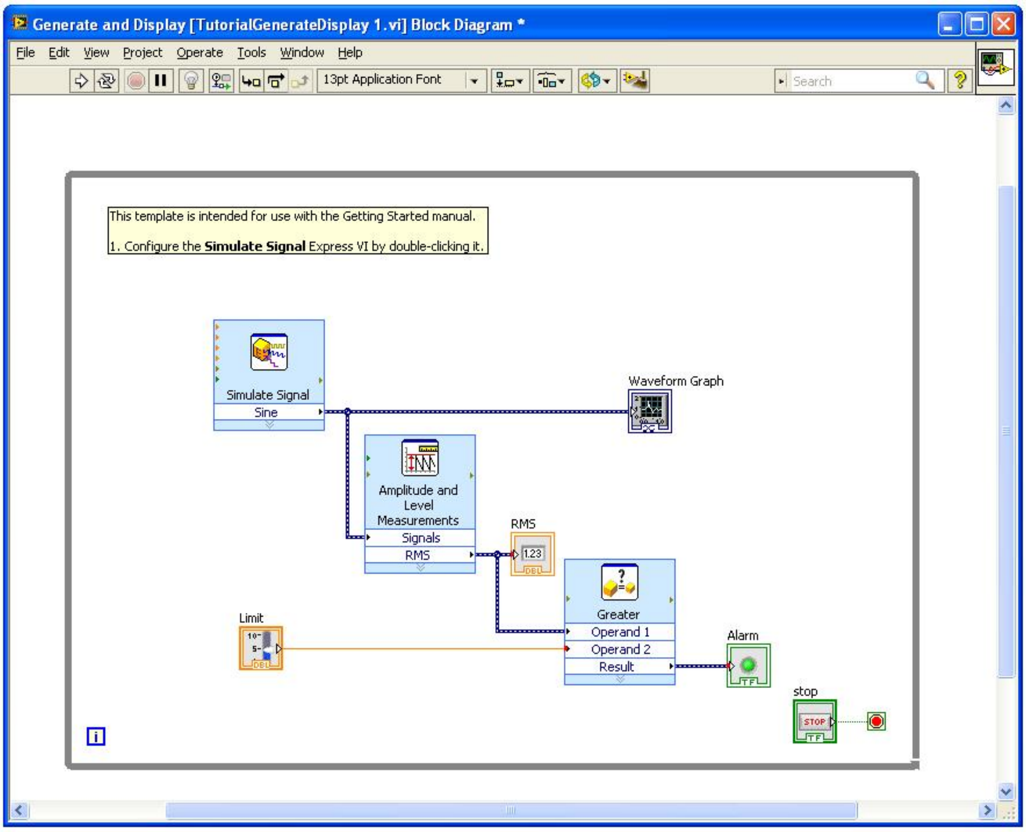
Figure 20 Comparison Express VI configuration.

Make your Block Diagram resemble Figure 21 by completing these steps:

1. Wire the Limit control to the Operand 2 input of the Comparison function.

2. Connect the RMS output of the Amplitude and Lever Measurements block to the Operand 1 input of the Comparison block.

3. Wire the Result output of the Comparison block to the Alarm indicator.


Generate and Display [TutorialGenerateDisplay 1.vi] Block Diagram *
File Edit View Project Operate Tools Window Help
13pt Application Font
Search
Simulate Signal
Sine
Waveform Graph
Amplitude and Level Measurements
Signals
RMS
RMS
1.23
DBL
Greater
Operand 1
Operand 2
Result
Limit
DBL
Alarm
TF
stop
STOP
TF

This template is intended for use with the Getting Started manual.

1. Configure the **Simulate Signal** Express VI by double-clicking it.

Figure 21 Block Diagram Snapshot.

- Switch to the Front Panel by pressing <Ctrl+E>.

- Save the VI using the File menu and name it Exercise1.vi.

- Run the VI.

While the VI is running you can change the **Limit** value. Also notice that when a data point received from the **Simulate Signal** VI is greater than the **Limit** value, the **Alarm** indicator lights up.

While the VI is still running, switch to the Block Diagram by pressing <Ctrl+E>. Enable **Highlight Execution** by clicking on the light bulb on the tool bar shown in Figure 22. This will allow you to see the flow of data through your program.

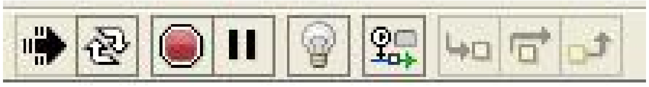


Figure 22

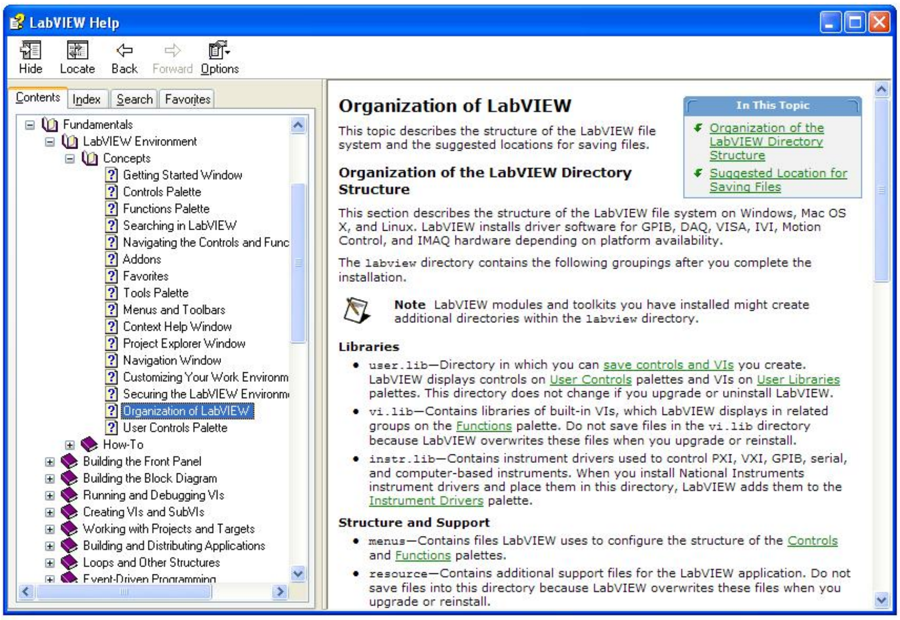
When you are finished, stop the VI by clicking the STOP button on the Front Panel.

3 Exercise 2: The LabVIEW Help

The LabVIEW help system is a great place to learn about LabVIEW and to go when you have questions. This exercise will introduce you to the rich source of information that is available for you to take advantage of.

Go back to the VI you just created, and press <F1> on the keyboard to start the help system

Expand **Fundamentals>>LabVIEW Environment** and explore the information available here, click around and get a feel for how it is organised. Figure 23 shows a snapshot of the LabVIEW help window.



Organization of LabVIEW

This topic describes the structure of the LabVIEW file system and the suggested locations for saving files.

Organization of the LabVIEW Directory Structure

This section describes the structure of the LabVIEW file system on Windows, Mac OS X, and Linux. LabVIEW installs driver software for GPIB, DAQ, VISA, IVI, Motion Control, and IMAQ hardware depending on platform availability.

The labview directory contains the following groupings after you complete the installation.

Note LabVIEW modules and toolkits you have installed might create additional directories within the labview directory.

Libraries

- user.lib—Directory in which you can save controls and VIs you create. LabVIEW displays controls on User Controls palettes and VIs on User Libraries palettes. This directory does not change if you upgrade or uninstall LabVIEW.
- vi.lib—Contains libraries of built-in VIs, which LabVIEW displays in related groups on the Functions palette. Do not save files in the vi.lib directory because LabVIEW overwrites these files when you upgrade or reinstall.
- instr.lib—Contains instrument drivers used to control PXI, VXI, GPIB, serial, and computer-based instruments. When you install National Instruments instrument drivers and place them in this directory, LabVIEW adds them to the Instrument Drivers palette.

Structure and Support

- menus—Contains files LabVIEW uses to configure the structure of the Controls and Functions palettes.
- resource—Contains additional support files for the LabVIEW application. Do not save files into this directory because LabVIEW overwrites these files when you upgrade or reinstall.

Figure 23 LabVIEW Help.

Take a few minutes to explore other topics in the help system. Click on the **Search** tab and try searching for analysis functions with features you might need in your work applications.

4 Exercise 3: LabVIEW Example Finder

In addition to the help system, LabVIEW includes a set of examples that in many cases are the building blocks for your applications. This exercise will introduce you to what examples are available.

Go back to the VI you created in Exercise 1 and from the **Help** menu select **Find Examples** as shown in Figure 24.

Switch to the **Search** tab, type in ”select” and hit Enter. From the results, select and double-click on **Select Signals.vi**.

Run the VI and explore the Block Diagram. Take a few minutes to look around at the other example programs available, search on terms that you are familiar with, switch back to the **Browse** tab and navigate through the different example categories and try out few examples.

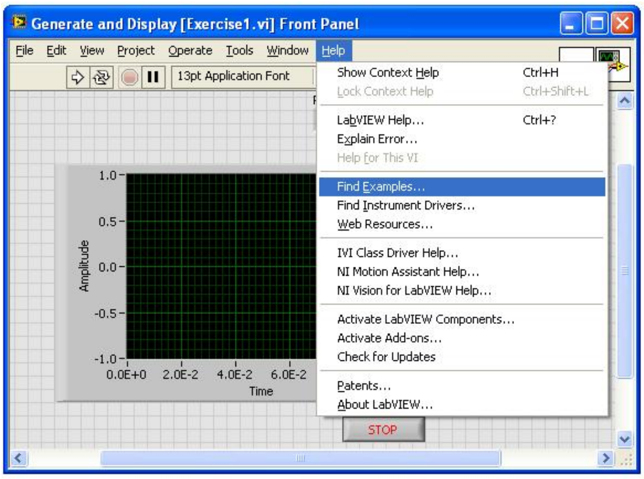


Figure 24

5 Exercise 4: Context Help

The **Context Help** window displays basic information about LabVIEW objects when you move the cursor over each object. To toggle display of **Context Help** window, select **Help** >> **Show Context Help** or press <Control-H> keys or click the “Show Context Help” button on the toolbar.

Go back to the VI you created in Exercise 1. Show the context help and move the cursor over the front panel and block diagram objects and read the available help. Take few minutes to look at the context help of the objects in the front panel and the block diagram.