Introduction to LabView

This lab introduces some fundamental concepts in National Instruments LabVIEW, through a simple programming example. LabVIEW stands for **Lab**oratory **V**irtual **I**nstrument **E**ngineering **W**orkbench.

The objective of this lab exercise is to

- Gain familiarity with the LabVIEW environment and programming fundamentals.
- Learn to create and debug VI files.
- Build intuitive Front Panel to view the results.

LabView is a graphical programming language and has the ability to interface with a wide range of external devices. LabVIEW programs are called Virtual Instruments (VIs). Each VI has three main parts: the block diagram, the front panel and the icon/connector. Each VI in turn can contain sub VIs and other structures. Controls and indicators on the front panel allow the user to input data into or extract data from a running virtual environment.

Basic Concepts

LabVIEW terms

Each VI contains three main parts

- *Front Panel* – Determines how the user interacts with the UI.
- *Block Diagram* – Code that controls the program.
- *Icon/Connector* – Means of connecting a VI to other VIs.

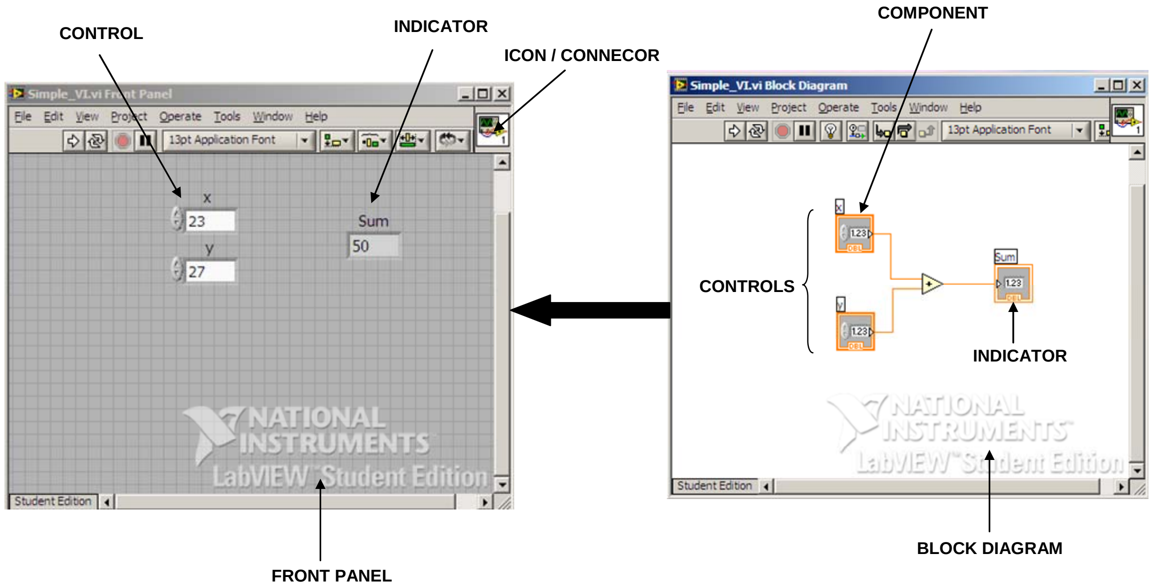


Figure 1. Illustration of components of a VI

The ***Front Panel*** is used to interact with the user when the program is running. Users can control the program, change inputs and see data updated in real-time. ***Controls*** are used as inputs and ***Indicators*** are used as outputs. Block diagram is the accompanying program for the front panel. A block diagram is created by wiring the LabVIEW components together. When a VI runs, 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.

An example of a VI is shown in Figure 1. Lets get started with a very simple LabView exercise to get some familiarity.

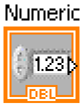
NOTE: At any point you need help on an object hit **CTRL+H** and a small context help will appear on the top. You can **move your mouse over any component** and the help for that component will be displayed in that window.

Example :

- Open LabView 8.5
- Open a Blank VI
- You should be able to see two windows open. a) Front Panel & b) Block Diagram. In case you do not see the block diagram, you can press **CTRL+E**.
- In the ***Front Panel***, when you right click, you should be able to see ***Controls*** menu.

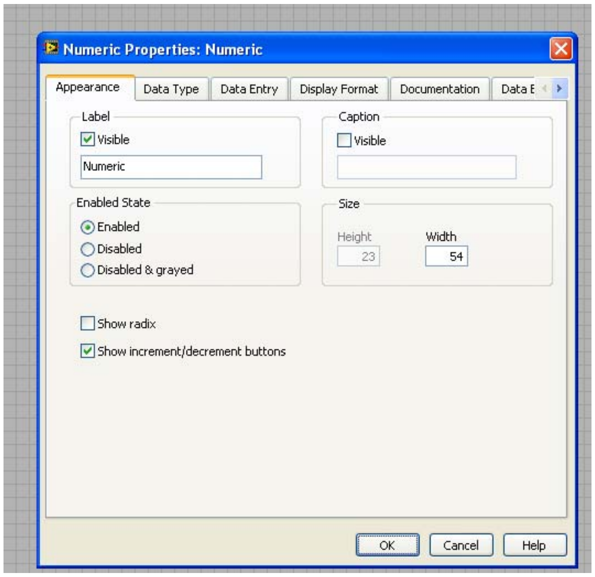
- In the Controls menu: click on ***Modern*** >> ***Numeric*** >> ***Numeric Control*** and place it on the Front Panel. This will be used to give input to provide the sum.

- Now in the ***Block Diagram*** you will see a ***component*** has appeared with title *Numeric*



- Right click on this component and click on ***properties***. Change the label to say *X*.

- Under the tab ***Data entry***, uncheck ***Use Default Limits***. You can now give a range of minimum and maximum values of the input. Let us for example choose minimum to be 0 and maximum to be 10. Select the increment as 1 for this example and click *ok*.



Label
☑ Visible
Numeric

Caption
□ Visible

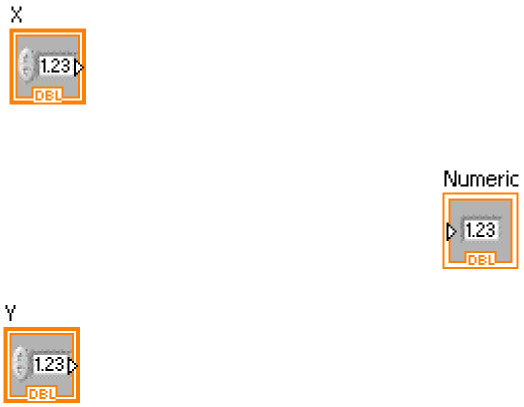
Enabled State
● Enabled
○ Disabled
○ Disabled & grayed

Size
Height 23
Width 54

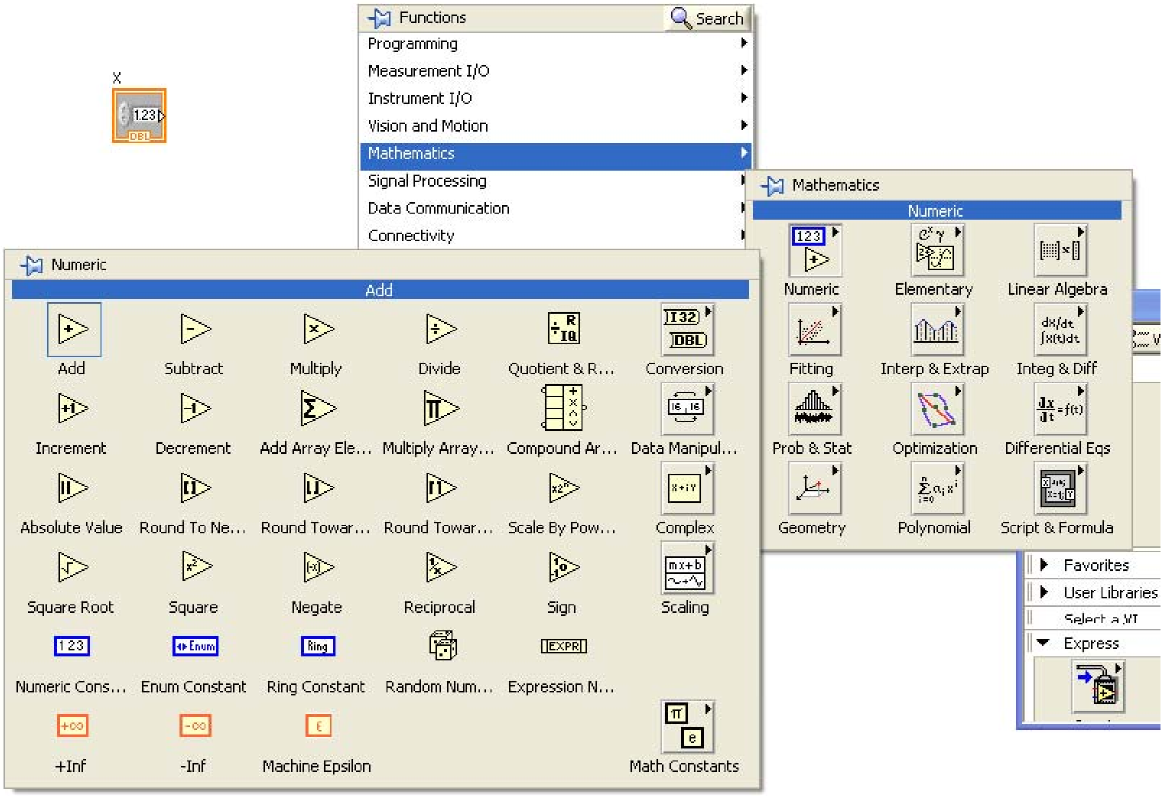
□ Show radix
☑ Show increment/decrement buttons

- Go to ***Front Panel*** and create another Numeric Control and label it as say *Y*.

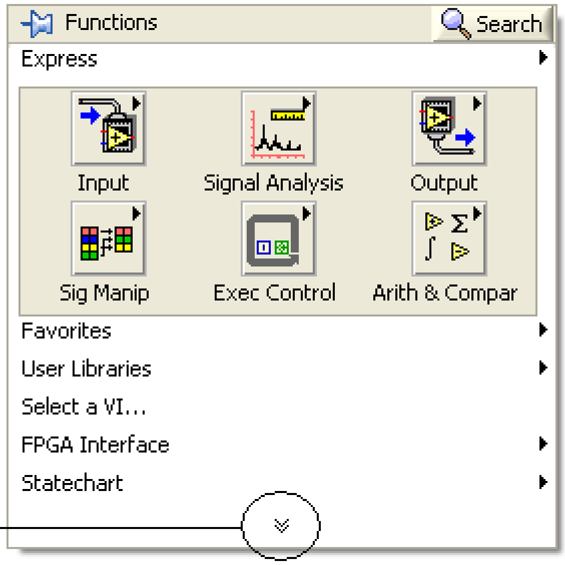
- In the Front Panel, right click and go to ***Modern*** >> ***Numeric*** >> ***Numeric Indicator*** and place it on the Front Panel. This will be used to display the sum of two numbers which were given as input using ***Numeric Control***. You will see that a component would have been created in the Block Diagram with label Numeric 2. You can change the label to say SUM by right clicking on the component and going to properties. label.



- In the ***Block Diagram***, right click anywhere and you will see ***Function Palette or Function Menu***. Go to ***Mathematics*** >> ***Numeric*** >> ***Add***. Place this component between the ***Numeric Controls (X & Y component) and Numeric Indicator (SUM component).*** (In case you don't see ***Mathematics*** option see **NOTE** below***)***

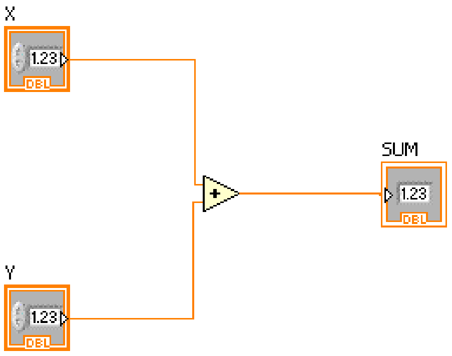


Note: In case you do not see Mathematics option in Function Palette, then do as shown in the figure below.

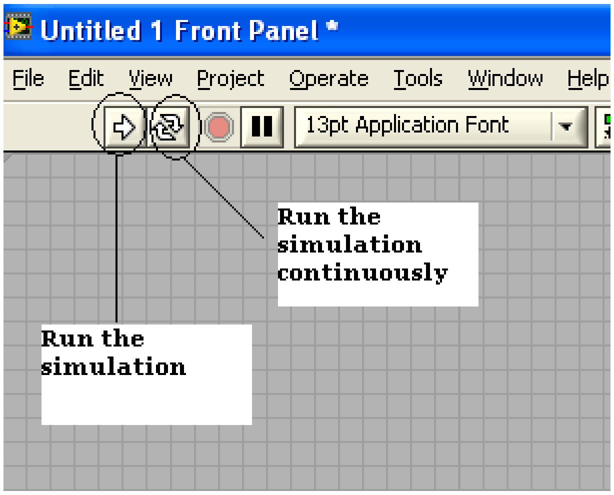


Click on this to expand the Function Palette Options

- Place you cursor on the X component in the ***Block Diagram***. You will see an orange dot being highlight. Drag this dot to the ***add*** component input. Do the same for the Y component. Connect the output of the add component to the input of the SUM component as shown below.



- Go to the ***Front Panel***. At the top left corner of the screen, you will see the following two icons.


Untitled 1 Front Panel *
File Edit View Project Operate Tools Window Help
13pt Application Font
Run the simulation continuously
Run the simulation

- By clicking on the arrow, the simulation runs once. Whatever values are there in X and Y, the sum will be shown in SUM. By clicking on the continuous run icon, the simulation will be running continuously and as and when you change the values of X and/or Y, the SUM changes. In the next labs you will see ways to use the while/for loops and you can avoid using thee Run Continuously button.

The *Controls Palette* can be used to place controls and indicators on the front panel. Select **Window>>Show Controls Palette** or right-click the front panel workspace to display the palette. Similarly, the *Functions Palette* can be used to build the block diagram. It contains the LabVIEW's native components listed under different categories. Select **Window>>Show Functions Palette** or right-click on the block diagram workspace to display the palette. *Tools palette* contains the set of tools that can be used to edit and build the block diagram. If automatic tool selection is enabled and we move the cursor over the objects on the front panel or block diagram, LabView automatically selects the corresponding tool from the palette. The VI can be executed by clicking the **Run** button in the status toolbar.