

Subject: Modeling & Simulation

Batch: 12BME

Year: 3rd Term: 6th

Lab Handout – 01

Name: _____	Roll No: _____	
Date: _____	Signature of Tutor: _____	Score: _____

DEVELOPING VIRTUAL INSTRUMENTS (VI'S) IN LAB VIEW

Lab Objectives:

- To understand Virtual Instrumentation and its applications.
- To Introduce Lab VIEW and common Lab VIEW functions.
- To understand front panels, block diagrams.
- To differentiate between indicators and controls and how they can be used.
- To Use built-in Lab VIEW functions.

Virtual Instrumentation:

- A Virtual Instrumentation is a software based system that is used by the user to develop a computerized test and measurement system, for controlling an external measurement hardware device from a desktop computer, and for displaying test or measurement data on panels in the computer screen. The test and measurement data are collected by the external device interfaced with the desktop computer.
- Thus, the basic idea of virtual instrumentation in general can be “any computer can simulate any other if we simply load it with software simulating the other computer.” This universality introduces user to wider range of applications using virtual instruments.

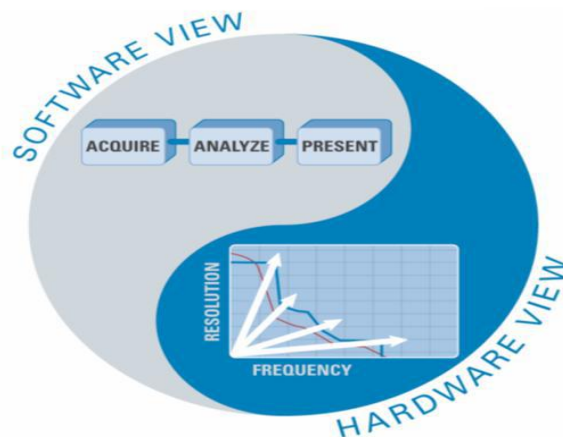


Figure 1- The Virtual Instrumentation Approach

- **Applications:** In virtual instrumentation, engineers use graphical programming software to create user-defined solutions that meet their specific needs. Some of the applications are listed in following table:

Table 1- Applications of Virtual Instrumentation

Design	Control	Measurements
<ul style="list-style-type: none"> • Signal and image processing • Embedded System Programming(FPGA, DSP, microcontroller, etc) • Simulation and Prototyping • And more.. 	<ul style="list-style-type: none"> • Automatic controls and dynamic systems • Mechatronics and dynamic systems • And more.. 	<ul style="list-style-type: none"> • Circuits and Electronics • Measurements and instrumentation • And more..

Creating Virtual Instruments Using LabVIEW:

- LabView is an integral part of virtual instrumentation because it provides an easy-to-use application development environment designed specifically with the needs of engineers and scientists in mind.
- With labView the user can design custom virtual instruments by creating a graphical user interface on the computer screen through which one can:
 - Operate the instrumentation program
 - Control selected hardware
 - Analyze acquired data
 - Display results

What Exactly Is LabVIEW?

LabVIEW, short for **Laboratory Virtual Instrument Engineering Workbench**, is a programming environment in which you create programs using a graphical notation (connecting functional nodes via wires through which data flows).

Features:

- It is an interactive program development and execution system designed for people, like scientists and engineers, who need to program as part of their jobs.
- Other programming systems like C, C++, or Java, use text-based languages to create lines of code, while LabVIEW uses a graphical programming language, G, to create programs in block diagram form.
- Programs that take weeks or months to write using conventional programming languages can be completed in hours using LabVIEW because it is specifically designed to take measurements, analyze data, and present results to the user.
- And because LabVIEW has such a versatile graphical user interface and is so easy to program with, it is also ideal for simulations, presentation of ideas, general programming, or even teaching basic programming concepts.

Virtual Instruments (VI's):

LabVIEW Programs are Called Virtual Instruments (VIs). A VI contains two main parts: a front panel & a block diagram.

1. **Front Panel:** The front panel is the interactive user interface of a VI. The front panel is where one can see program's output, and one can input data using a mouse and keyboard, and then view the results produced by your program on the screen. Users can control the program, change inputs, and see data updated in real time.

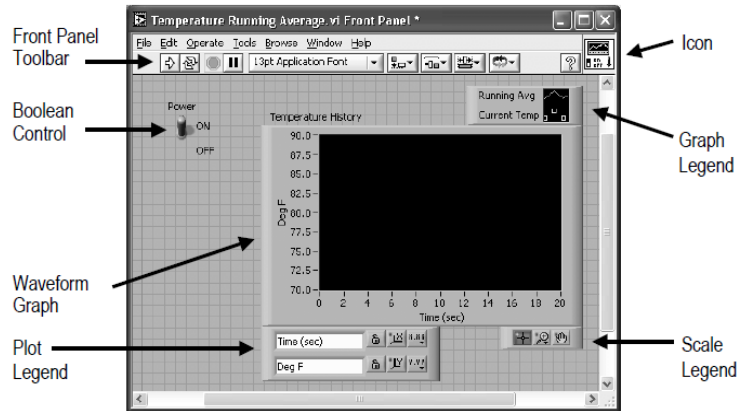


Figure 2- Front Panel

- Front Panel gives you the Control Palette, which consists of:
 - **Controls** = Inputs, Controls are knobs, pushbuttons, dials, and other input devices.
 - **Indicators** = Outputs, Indicators are graphs, LEDs, and other displays.

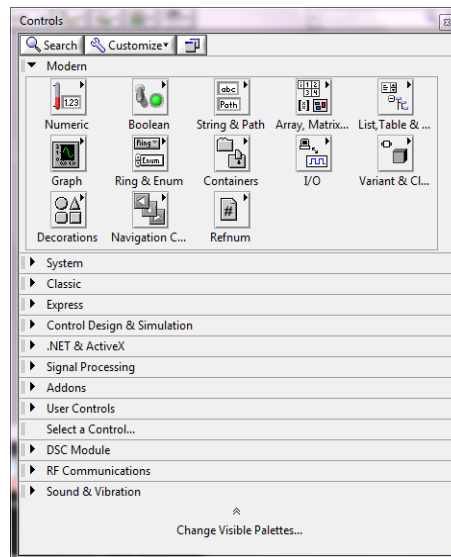


Figure 3 - Front Panel Controls Palette

2. **Block Diagram:** The block diagram is the VI's source code, constructed in Lab View's graphical programming language; G. The block diagram is the actual executable program. The components of a block diagram are lower-level VIs, built-in functions, constants, and program execution control structures.

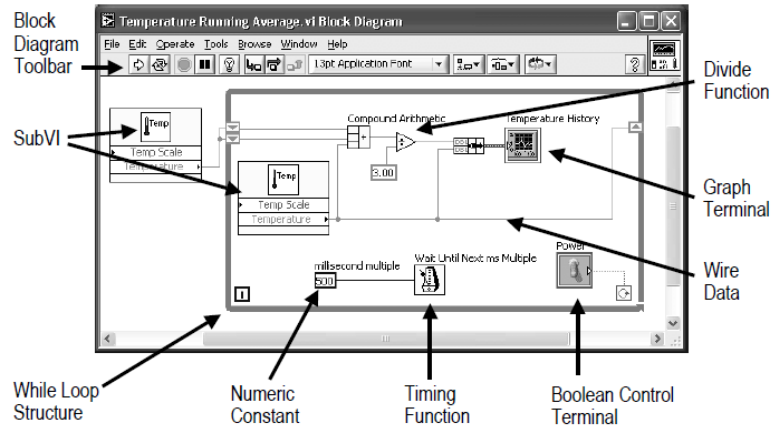


Figure 4 - Block Diagram

- The various components of a **block diagram** are **terminals, nodes** and **wires**:
 - Front panel objects have corresponding **terminals** on the block diagram so data can pass from the user to the program and back to the user.
 - A **node** is just a fancy word for program execution element. The Add and Subtract functions represent one type of node. A structure is another type of node.
 - A LabView VI is held together by **wires** connecting nodes and terminals. A wire can only have one data source, but it can have multiple data sinks. Each wire has a different style or color, depending on the data type that flows through the wire. The following table shows a few wires and corresponding types:

Table 2 – Data types and their representation

Wire Type	Scalar	1D Array	2D Array	Color
Numeric				Orange (floating-point)
				Blue (integer)
Boolean				Green
String				Pink

- Block diagram gives you Functions Palette, as shown in figure:

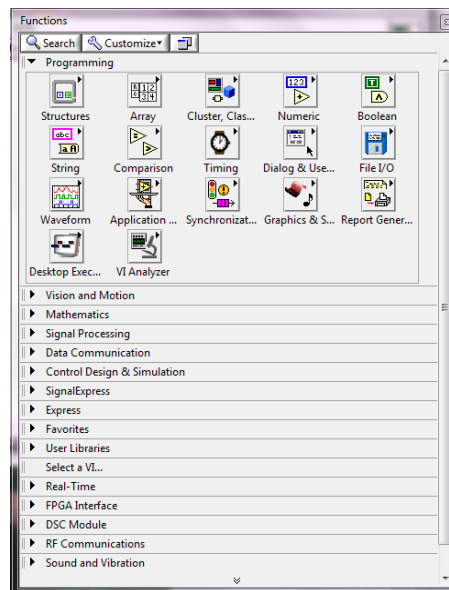


Figure 5 - Block Diagram Functions Palette

Express VIs:

- Express VIs is interactive VIs with configurable dialog page. They differ from normal LabView functions because they will allow you to define the behavior of the function using a wizard or configuration dialog.
- Whenever you put an express VI on your block diagram, LabView immediately and automatically builds lower-level LabView code for you, “behind the scenes”. One can find many Express Vis in the Express category of the Functions palette, as shown in figure:

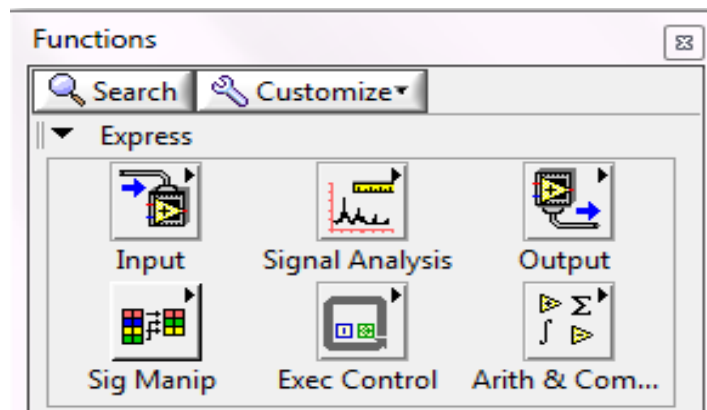


Figure 6 - Express VIs

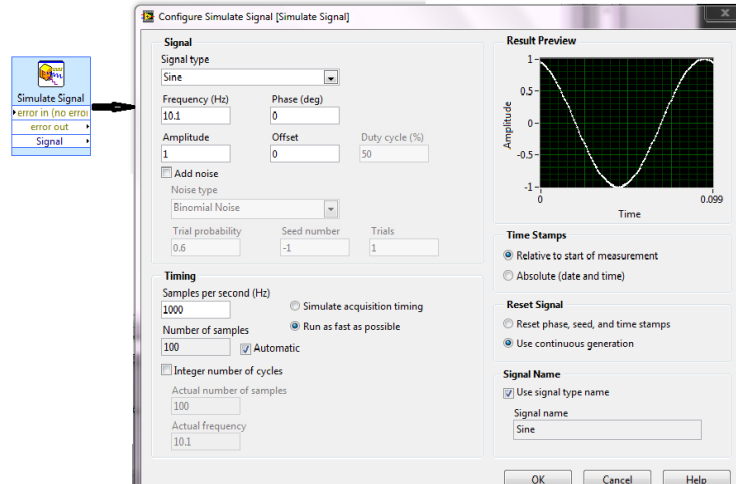


Figure 7 - Express VIs with configurable dialog box

Status Toolbar:

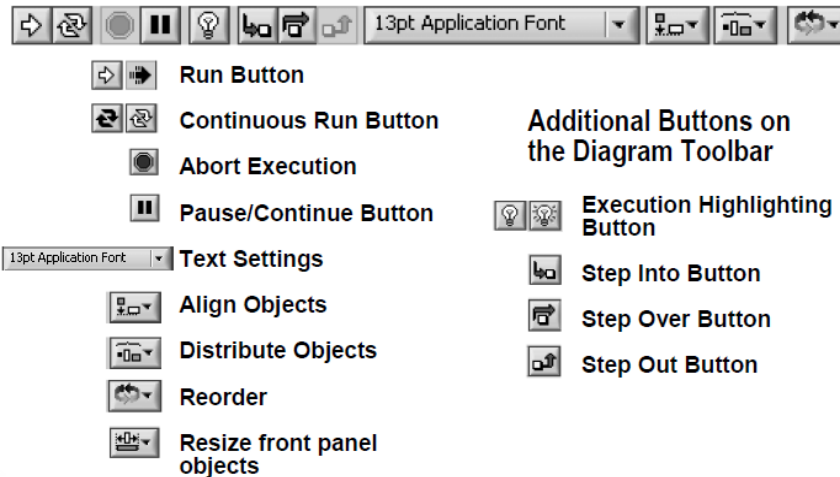


Figure 8 – Status Toolbar

Context Help:

- One can find help for any function by pressing ctrl+H. This will give information of each input and output node of the function, as shown in figure:

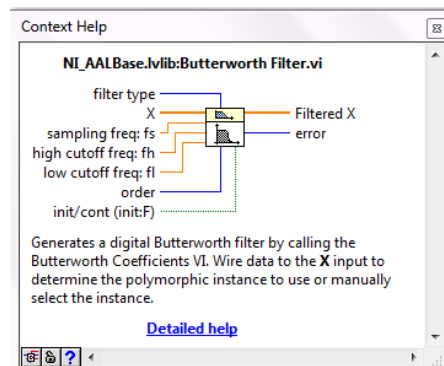


Figure 9 - Help window in VI environment

Building Your Own VI:

- Launch LabVIEW, you will see this getting started window.

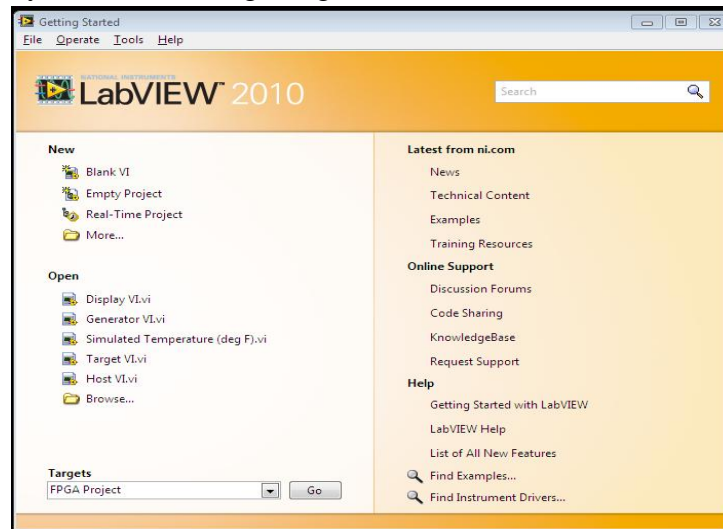


Figure 10 - Getting started window

- Click Blank VI
- Then you will get front panel and block diagram windows, as shown in fig:11

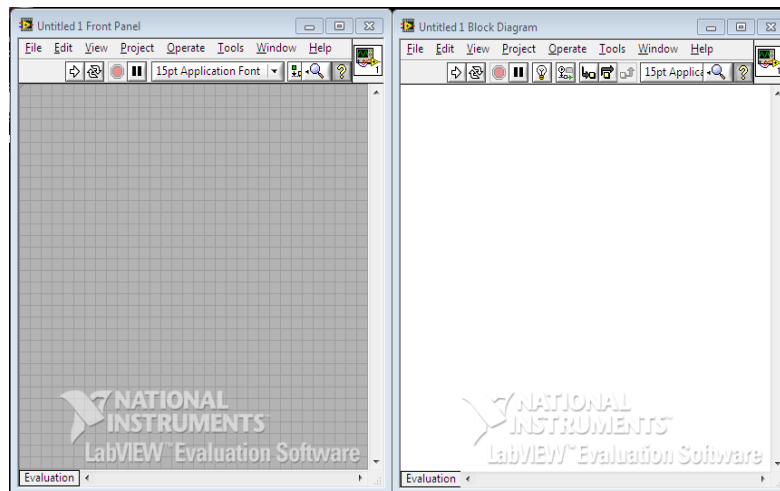


Figure 11 - VI front panel and block diagram

- Now, one can put controls and indicators on front panel and their corresponding terminals will appear on block diagram.
- Then wire those terminals on block diagram accordingly and click on run button to see your result on front panel.

LabVIEW Keystroke shortcuts:

Ctrl+H	Activate/ Deactivate context help window.
Ctrl+N	Opens a new VI.
Ctrl+W	Close the VI.
Ctrl+S	Saves the VI.
Ctrl+E	Toggle between front panel & block diagram.
Ctrl+L	Display error list window.
Ctrl+B	Remove broken wires from block diagram.
Ctrl+Z	Undo last action.
Ctrl+Q	Quits Labview.
Ctrl+T	Tiles front panel & block diagram window.

Review Questions:

- 1. What is G programming and how it differs from other programming languages?**

- 2. What is LabView?**
