LabVIEW and MatLab

E80 Teaching Team

February 5, 2008

Objectives of this lecture

- Learn LabVIEW and LabVIEW's functions
- Understand, design, modify and use Virtual Instruments (VIs)
- Construct (modify) and use data acquisition applications for acquiring and processing digital and analog signals supplied by sensors, transmitters, …

Outline

- Basics of LabVIEW
- Mathscript and LabVIEW
- Data Acquisition with LabVIEW
- MATLAB

What is LabVIEW?

Laboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench

A Graphical Program Development Environment

Used in some of the most advanced R&D labs (JPL, Siemens Medical, …)

Has been around since 1980

The best way to learn LabVIEW is to PRACTICE, PRACTICE, PRACTICE

Highly Addictive!

What is the use of LabVIEW in E80?

We will use LabVIEW to….

- Monitor and connect to sensors and measurement devices in an experiment
- Retrieve signals using data acquisition platforms controlled by LabVIEW
- Process data and represent them in a meaningful, efficient way
- Consolidate all the data obtained in the experiment to perform analysis

Video: LabVIEW and Rubik Cube!

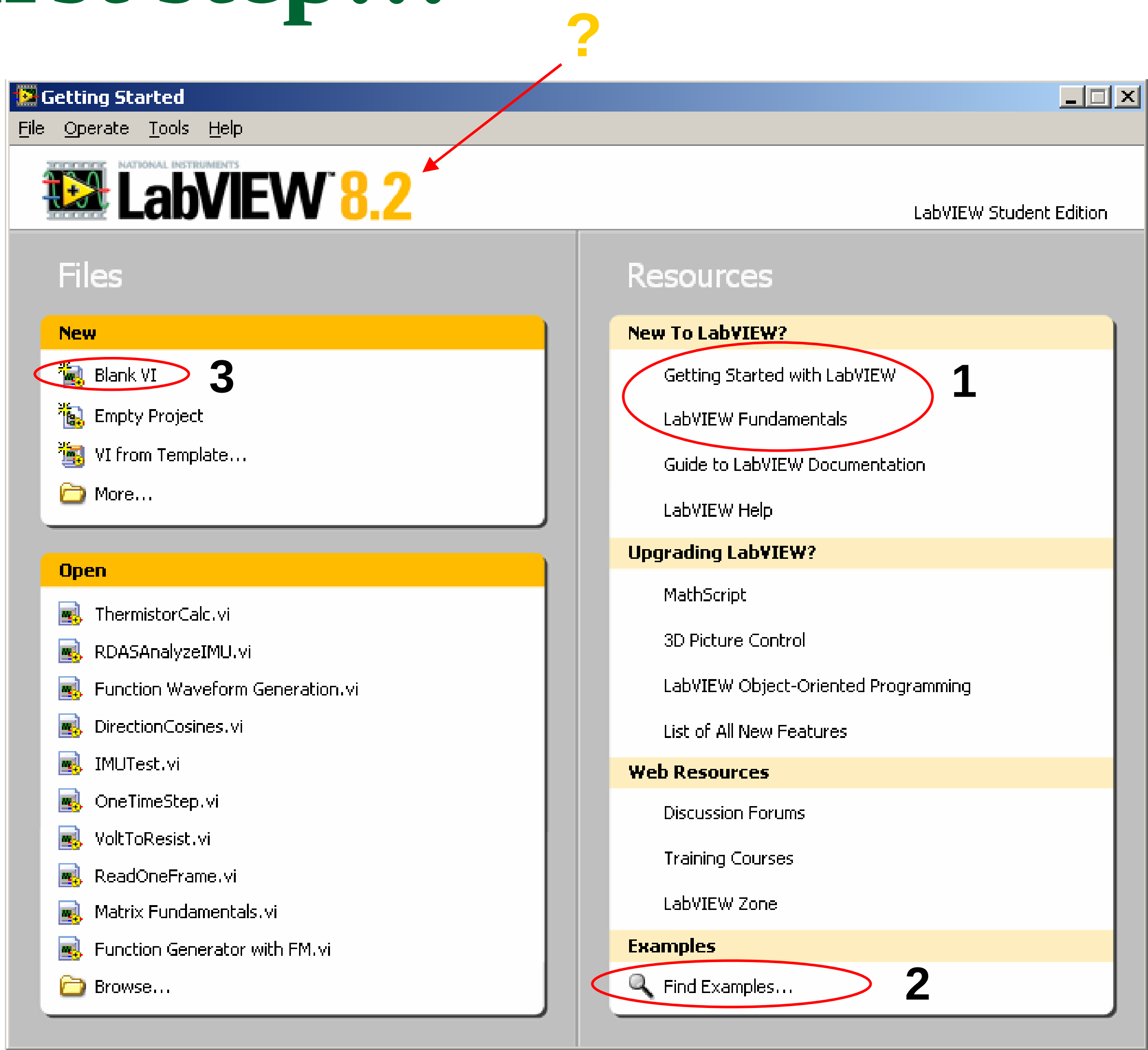
Where can I find LabVIEW?

- Your E80 laptops will have LabVIEW 8.2.1 installed on them
- You can install LabVIEW on your PC using the CD that comes with your you textbook
- There are lots of information about LabVIEW that you can find on the web and on National Instrument's web page:

www.ni.com/labview/

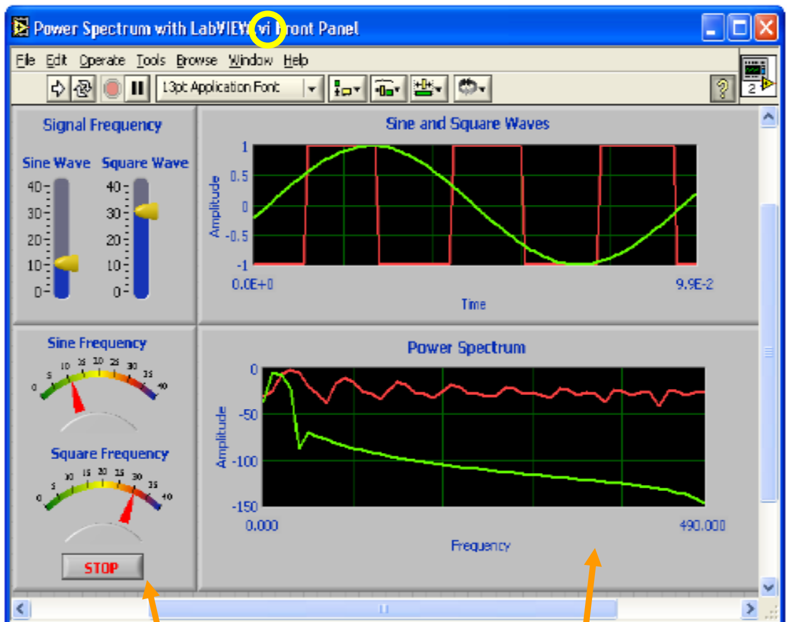
The very first step…

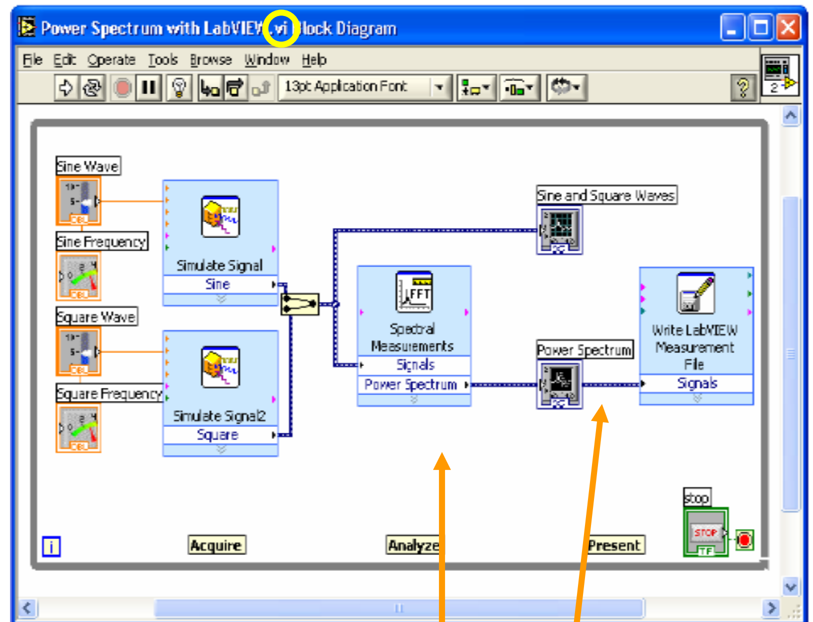
This is what you see
 if you run LabVIEW
On your PC



Basics

LabVIEW programs are called Virtual Instruments (VIs)





Front Panel
Controls = Inputs
Indicators = Outputs

Tip:

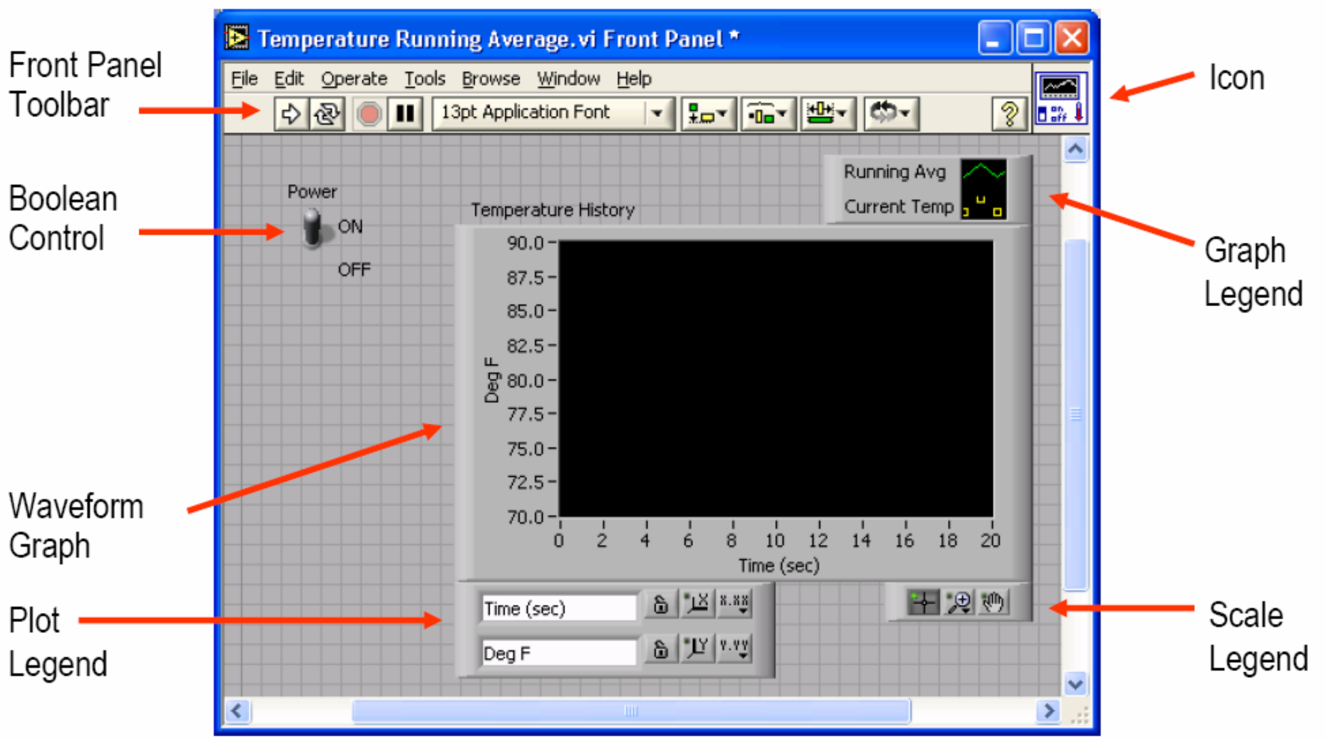
Every LabVIEW vi has a Front Panel and a Block Diagram

Block Diagram
Behind the scene
Components wired

ni.com

The Front Panel

VI Front Panel

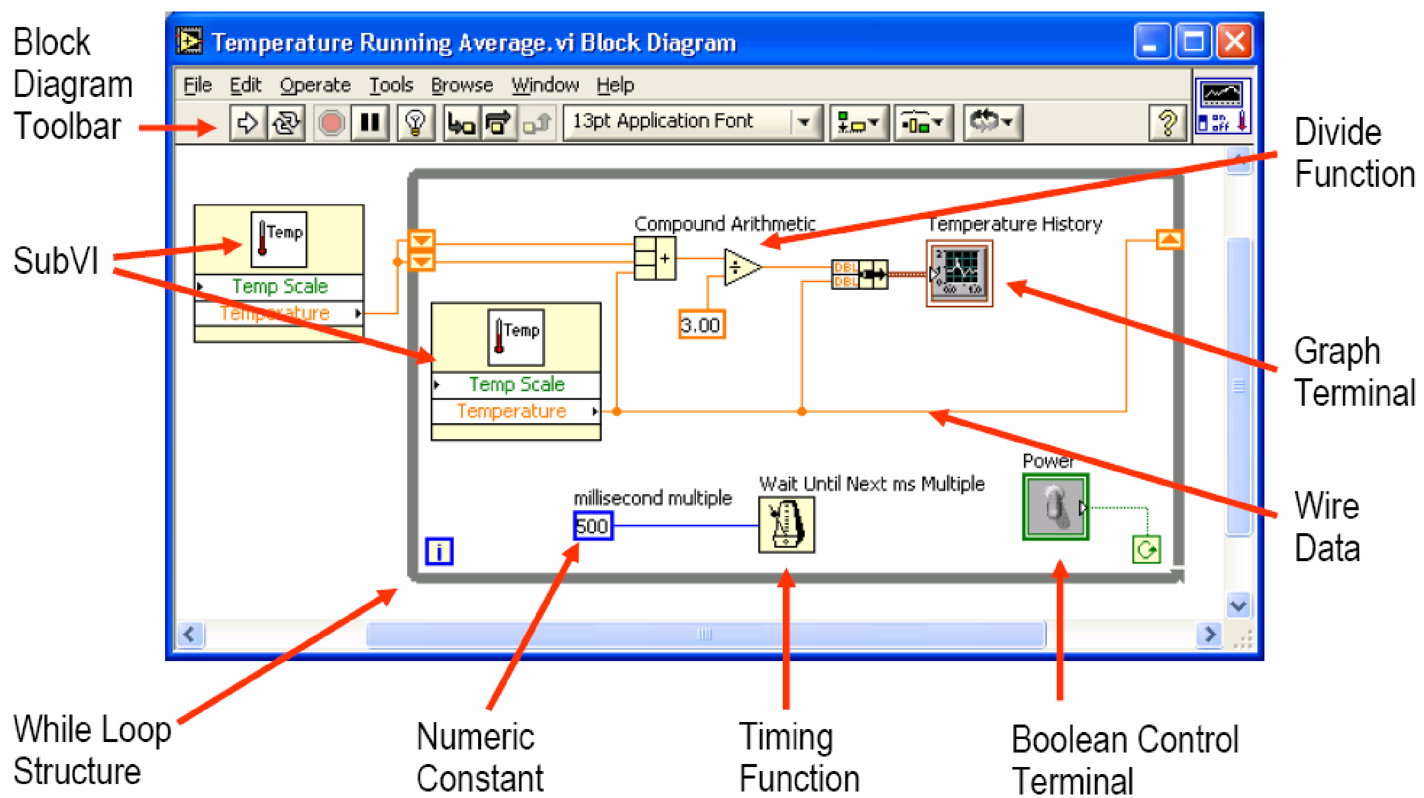


ni.com

NATIONAL INSTRUMENTS

The Block diagram

VI Block Diagram



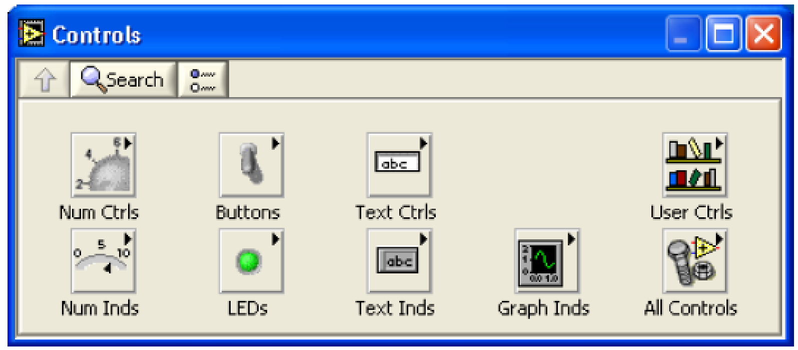
Tip: Use **Ctrl-E** to switch between front panel and block diagram

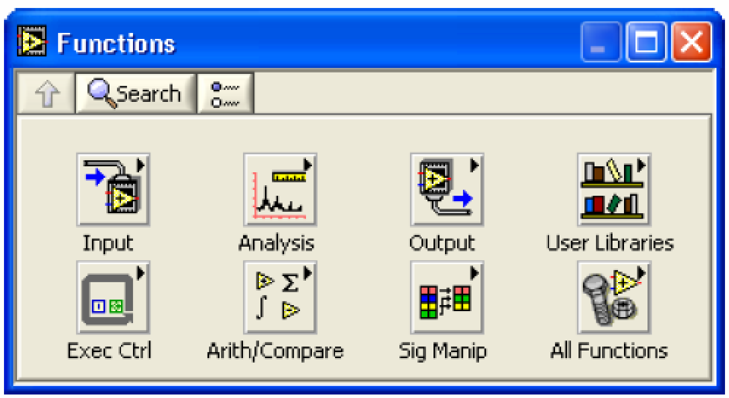
ni.com
NATIONAL INSTRUMENTS

Basic Functions

Controls and Functions Palettes

Controls Palette
(Front Panel Window)





Functions Palette
(Block Diagram Window)

ni.com

Tools Palette

Tools Palette

- Floating Palette
- Used to operate and modify front panel and block diagram objects.

Automatic Selection Tool

Operating Tool

Positioning/Resizing Tool

Labeling Tool

Wiring Tool

Shortcut Menu Tool

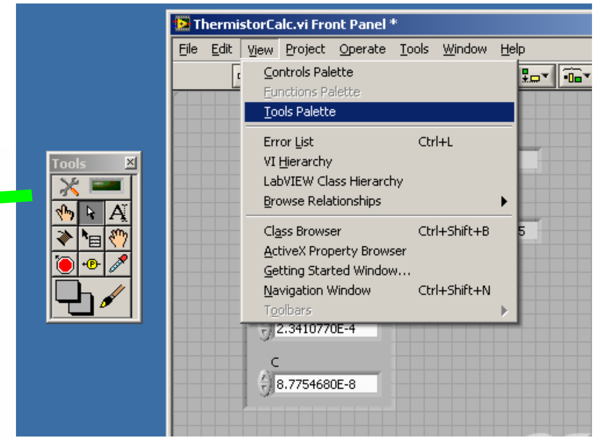
Scrolling Tool

Breakpoint Tool

Probe Tool

Color Copy Tool

Coloring Tool



Activating Tools Palette

ni.com

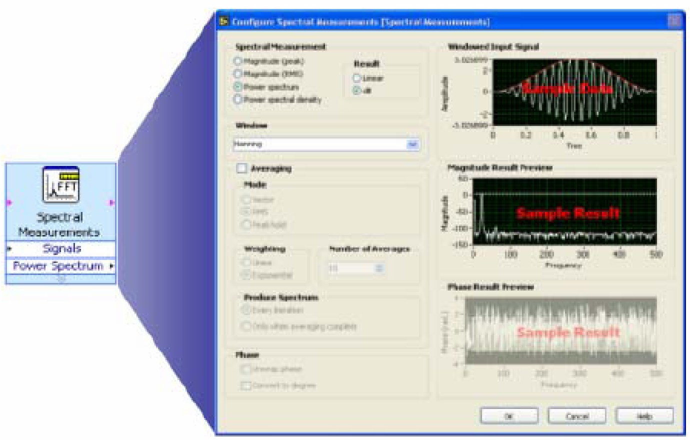
NATIONAL INSTRUMENTS

VIs and Functions

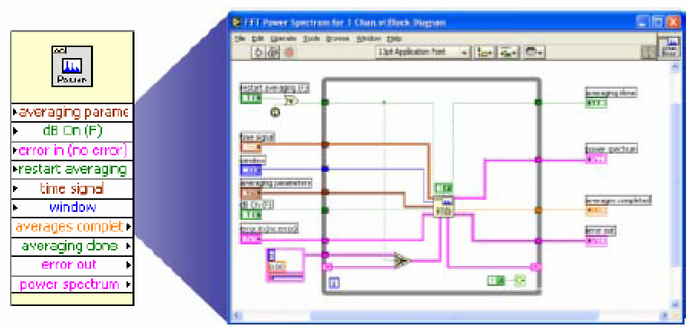
Express VIs, VIs and Functions

- Express VIs: interactive VIs with configurable dialog page
- Standard VIs: modularized VIs customized by wiring
- Functions: fundamental operating elements of LabVIEW; no front panel or block diagram

Function



Express VI



Standard VI

ni.com

NATIONAL INSTRUMENTS

Example 1 : Simple Conversion

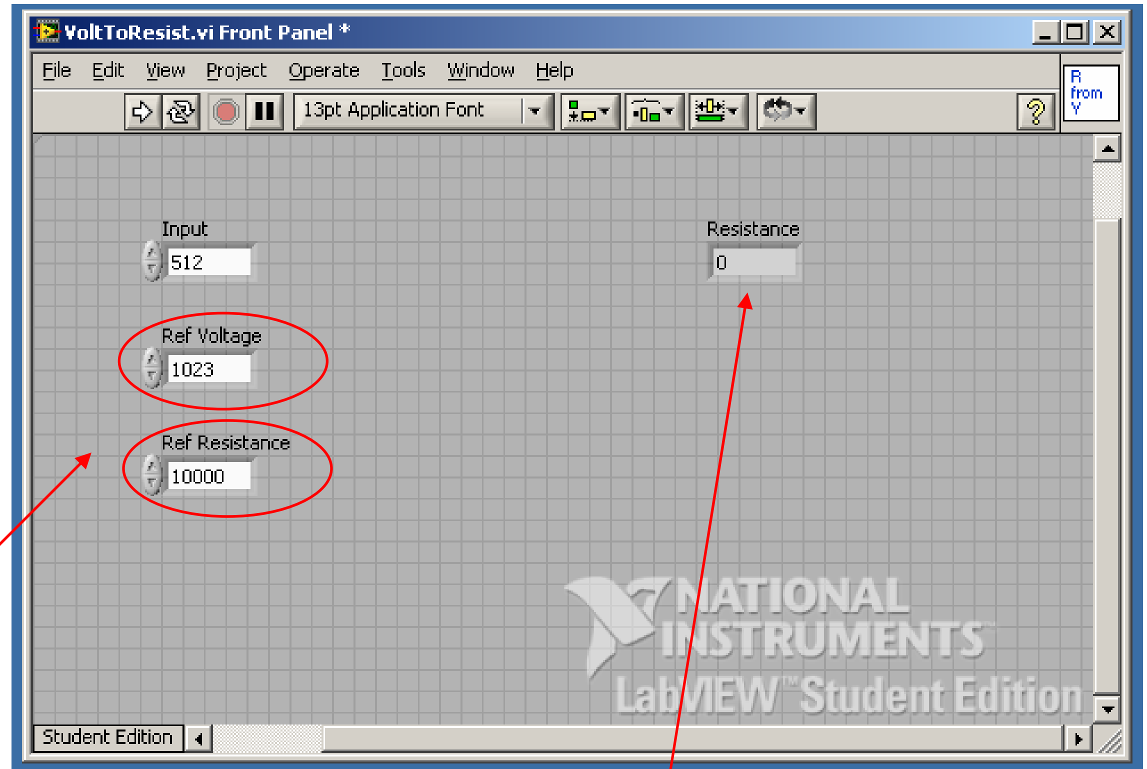
$$R = \frac{V - V_{ref}}{V \cdot R_{ref}}$$

Goal: Convert voltage to resistance

1) Have an input signal in volts coming from a thermistor

2) Know the conversion equation between the voltage received and the resistance desired

3) Need an interactive vi to show us the resistance for input value of voltage



Input and Refs.

Display

Example 1 : Simple Conversion

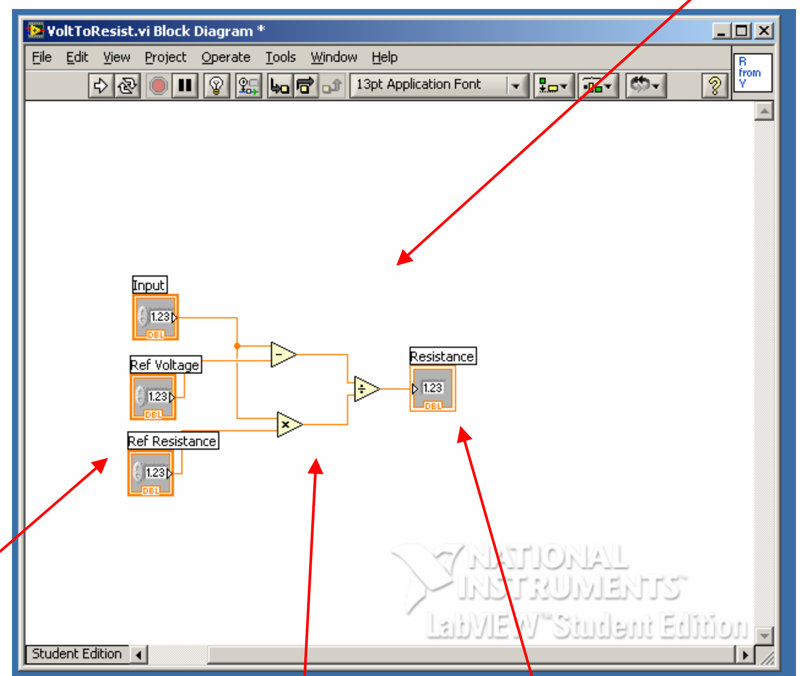
$$R = \frac{V - V_{ref}}{V \cdot R_{ref}}$$

Goal: Convert voltage to resistance

1) Have an input signal in volts coming from a thermistor

2) Know the conversion equation for the voltage read, and resistance

3) Need an interactive vi to show us the resistance for input value of voltage



Input and Refs.

Operations

Output

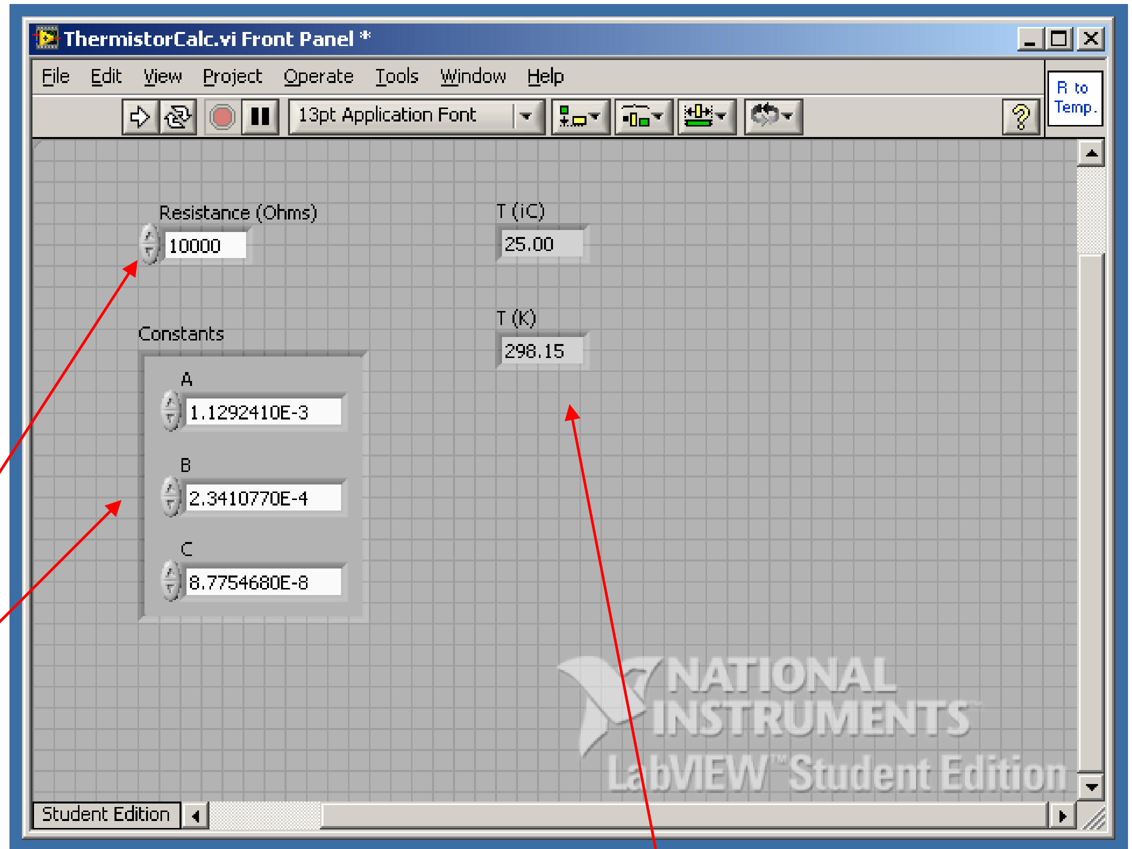
Tip: Use **Ctrl-E** to switch between front panel and block diagram

Demonstration I

Example 2 : Thermistor Calculation

Goal: Calculate temperature from measured resistance…

1) Know the conversion equation for the resistance read
2) Have constants in the equation
3) Calculate temperature in ˚C
4) Convert temperature to ˚K



Input

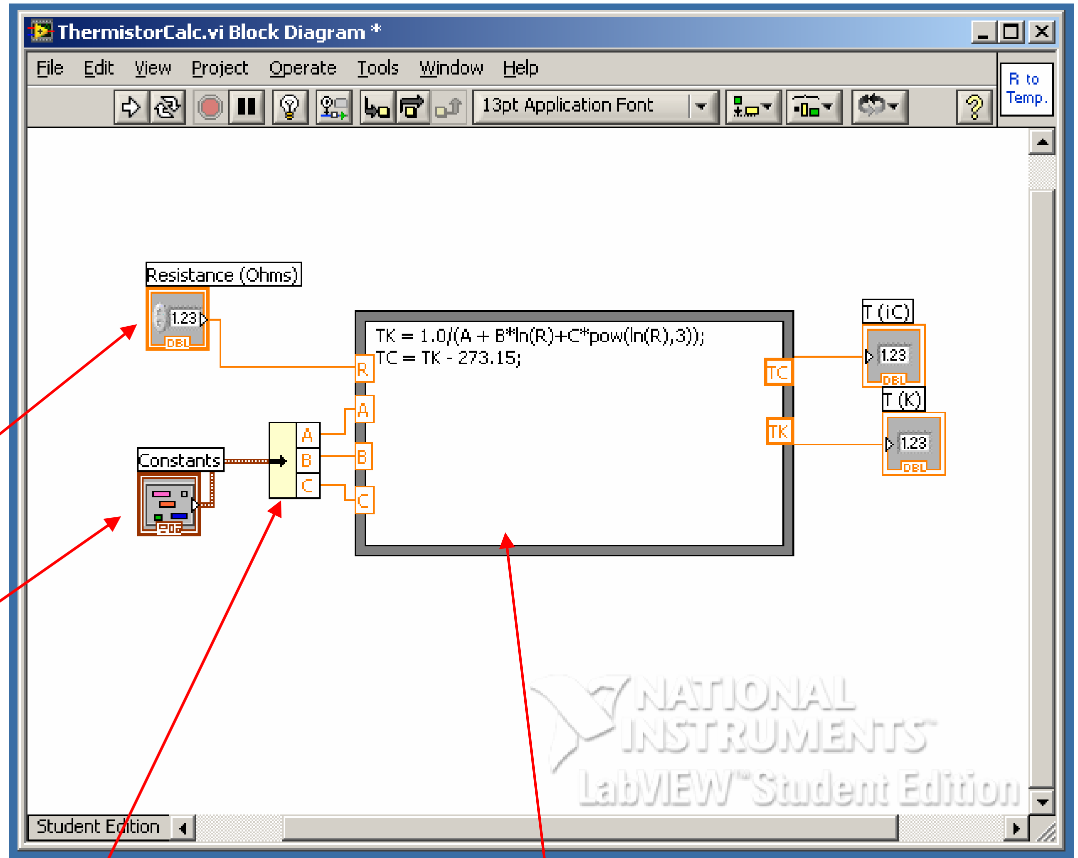
Constants

Results

Example 2 : Thermistor Calculation

$$T^{\circ K} = \frac{1}{A + B\ln(R) + C[\ln(R)]^3}$$

$$T^{\circ K} = T^{\circ C} + 273.15$$



Input

Constants

Unbundled (see Ch. 7)

Mathscript (similar to MATLAB)

Demonstration II

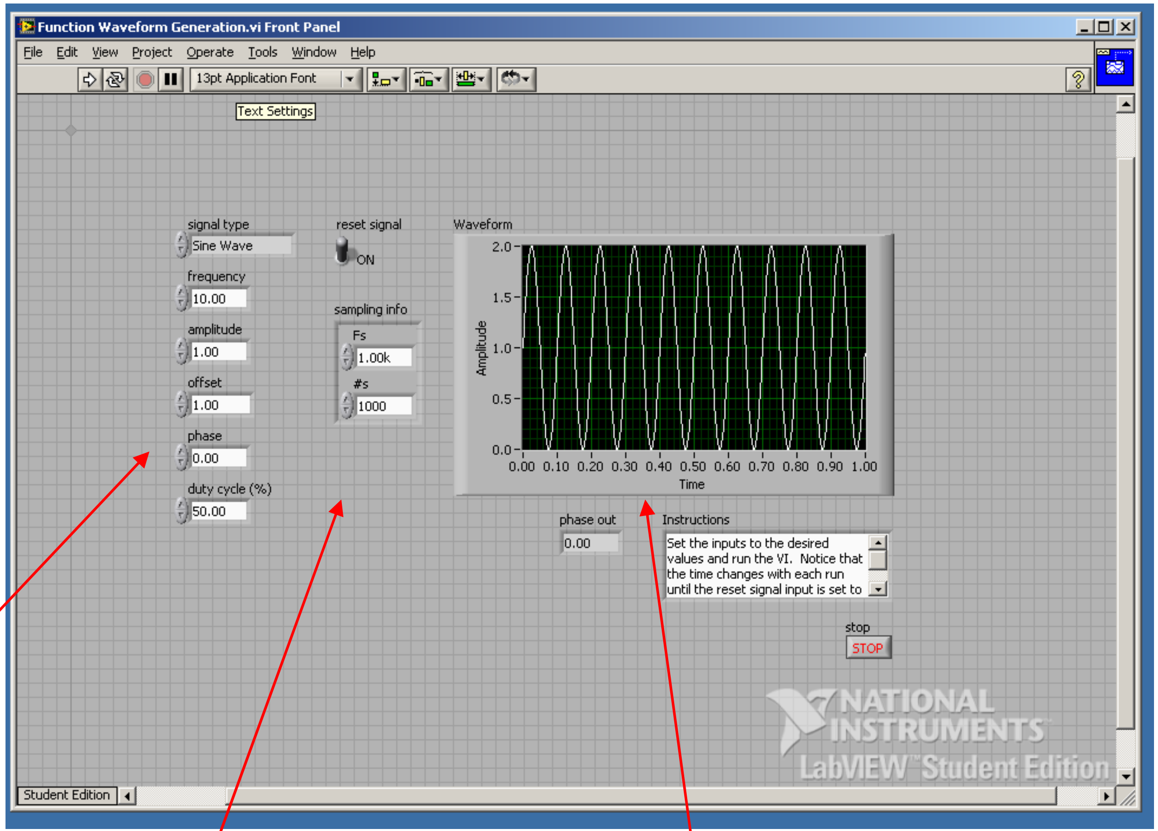
Example 3 : Function Generator

Goal: Make a function generator that…

1) Allows choosing signal type, varying frequency, amplitude, offset, phase, …

2) Displays the signal graphically

3) Addresses sampling rate and tracking of the signal



Controls

Controls

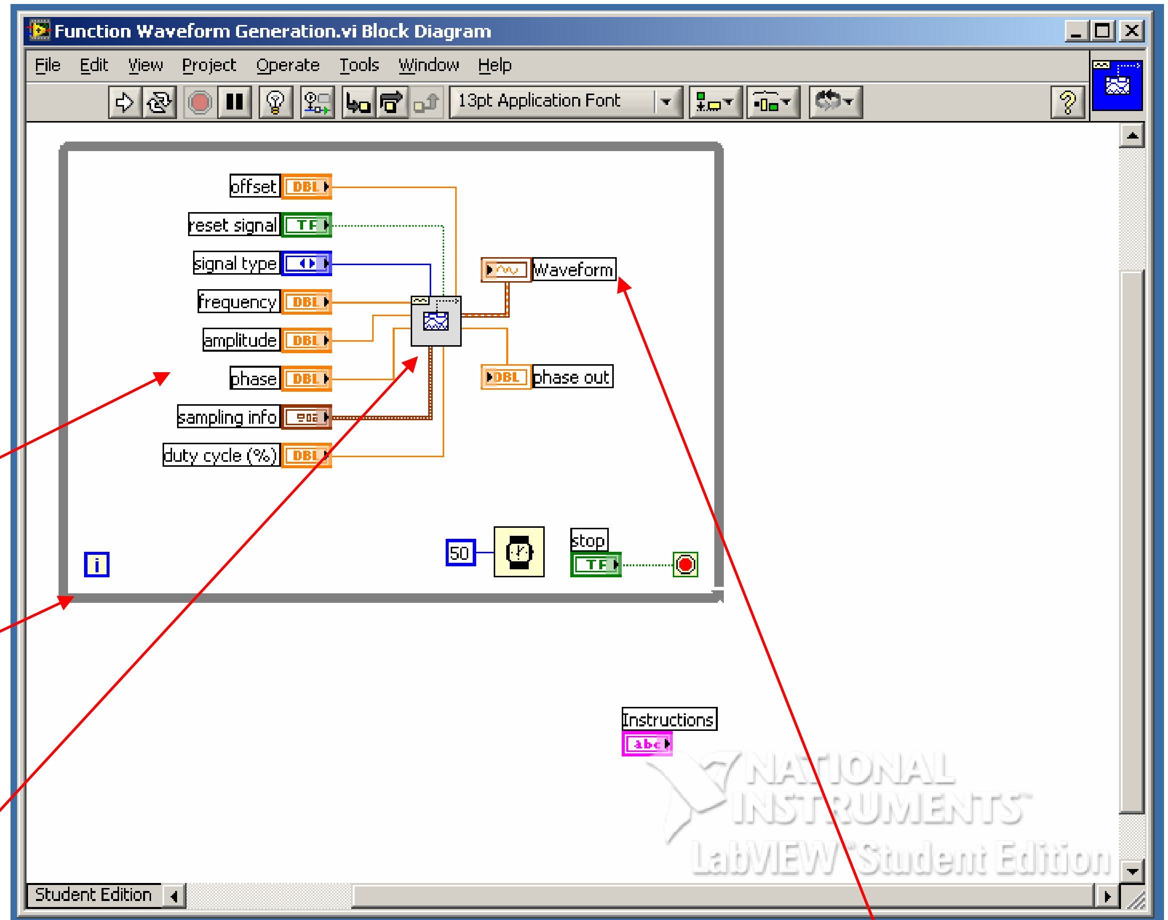
Display

Example 3: Function Generator

Method:

Use available modules as “building blocks” and “wires” as connection tools for flow of data/commands to…

Tie elements to each other in an interactive, repetitive platform.



Numerical Data

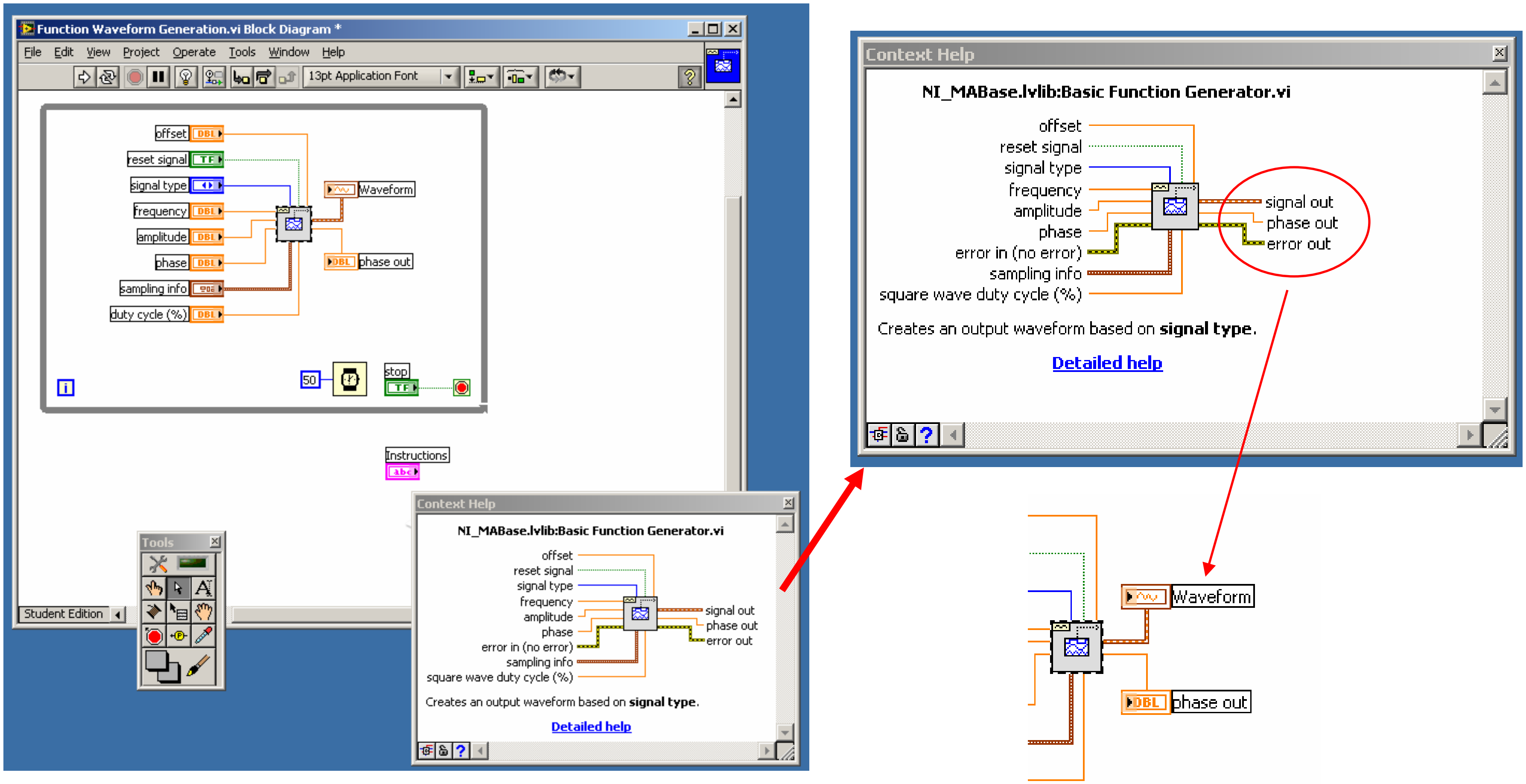
While Loop (Ch. 6)

Function Generator VI (Express)

Display

Tip: Use **Ctrl-H** and point on a component for a pop-up help window

Example 3: Function Generator



Example 3: Numeric Data Types

Numeric Data Types Table

The following table displays the numeric data types available in LabVIEW.

Terminal	Numeric Data Type	Bits of Storage on Disk	Approximate Number of Decimal Digits	Approximate Range on Disk
SGL	Single-precision, floating-point	32	6	Minimum positive number: 1.40e-45 Maximum positive number: 3.40e+38 Minimum negative number: -1.40e-45 Maximum negative number: -3.40e+38
DBL	Double-precision, floating-point	64	15	Minimum positive number: 4.94e-324 Maximum positive number: 1.79e+308 Minimum negative number: -4.94e-324 Maximum negative number: -1.79e+308
EXT	Extended-precision, floating-point	128	varies from 15 to 33 by platform; refer to the LabVIEW Data Storage Application Note for more information about using numeric data types in LabVIEW	Minimum positive number: 6.48e-4966 Maximum positive number: 1.19e+4932 Minimum negative number: -6.48e-4966 Maximum negative number: -1.19e+4932
CSG	Complex single-precision, floating-point	64	6	Same as single-precision, floating-point for each (real and imaginary) part
CDB	Complex double-precision, floating-point	128	15	Same as double-precision, floating-point for each (real and imaginary) part
CXT	Complex extended-precision, floating-point	256	varies from 15 to 33 by platform; refer to the LabVIEW Data Storage Application Note for more information about using numeric data types in LabVIEW	Same as extended-precision, floating-point for each (real and imaginary) part
I8	Byte signed integer	8	2	-128 to 127
I16	Word signed integer	16	4	-32,768 to 32,767
I32	Long signed integer	32	9	-2,147,483,648 to 2,147,483,647
U8	Byte unsigned integer	8	2	0 to 255
U16	Word unsigned integer	16	4	0 to 65,535
U32	Long unsigned integer	32	9	0 to 4,294,967,295
	128-bit time stamp	<64.64>	15; refer to the LabVIEW Data Storage Application Note for more information about using the time stamp data type in LabVIEW	Minimum time (in seconds): 5.4210108624275221700372640043497e-20 Maximum time (in seconds): 9,223,372,036,854,775,808

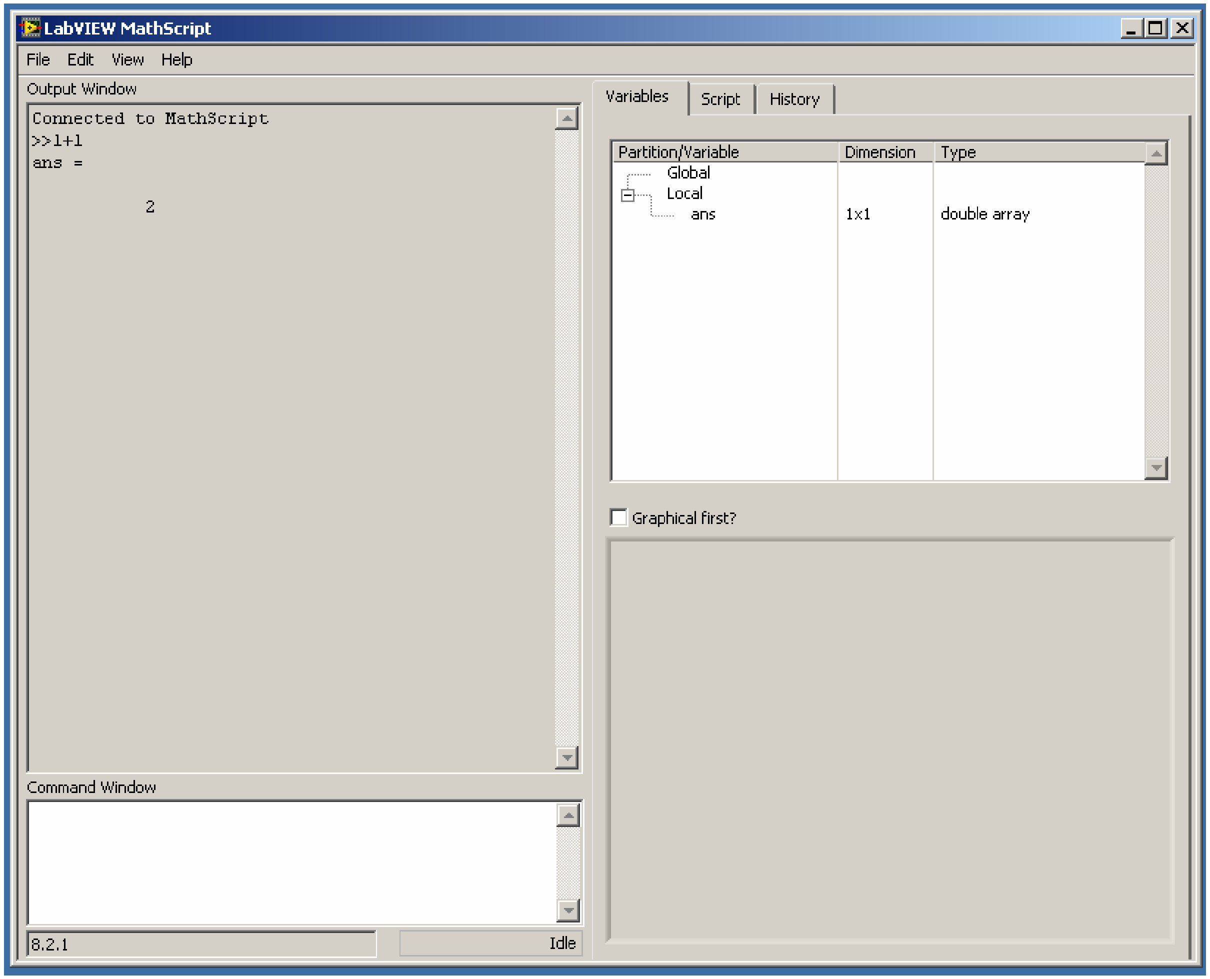
Demonstration III

LabVIEW Mathscript

- A LabVIEW tool for executing textual mathematical commands

  - Matrix and vector based calculations (linear algebra)
  - Visualization of data in plots
  - Running scripts containing a number of commands written in a file
  - A large number of mathematical functions. An overview is given later in this document.
  - MathScript command are equal to MATLAB commands (some MATLAB commands may not be implemented).

LabVIEW Mathscript

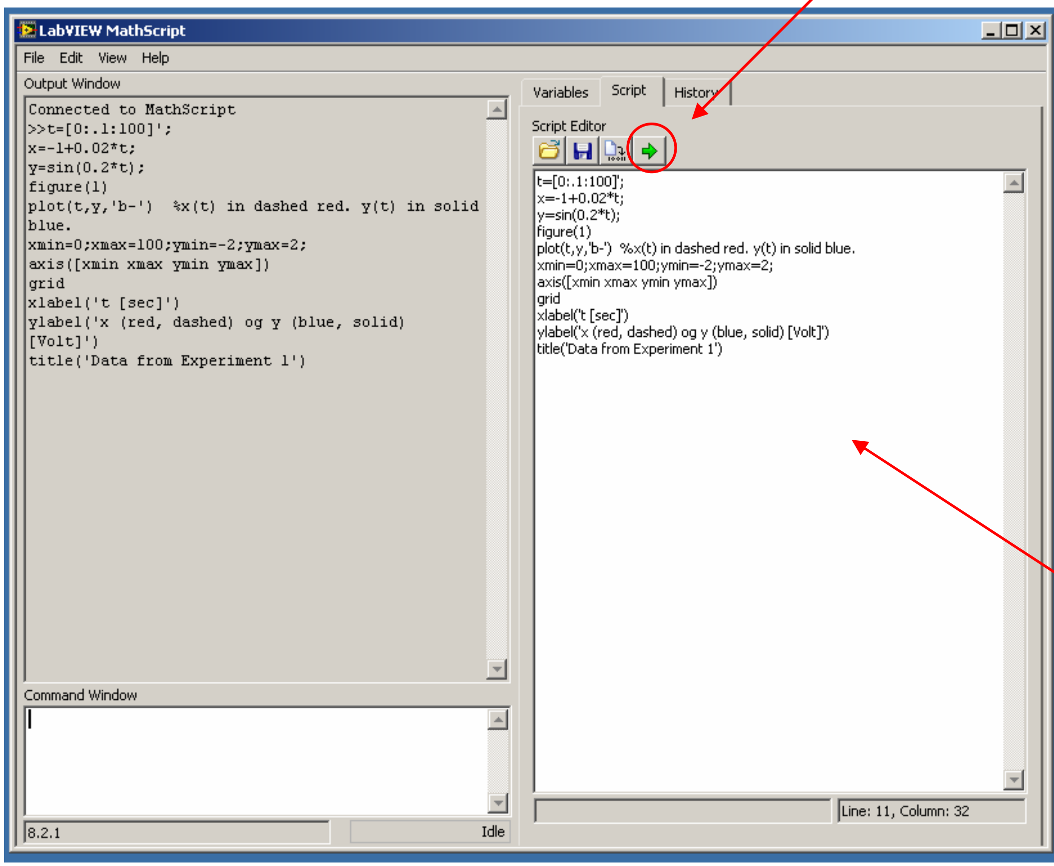


How do I use Mathscript?

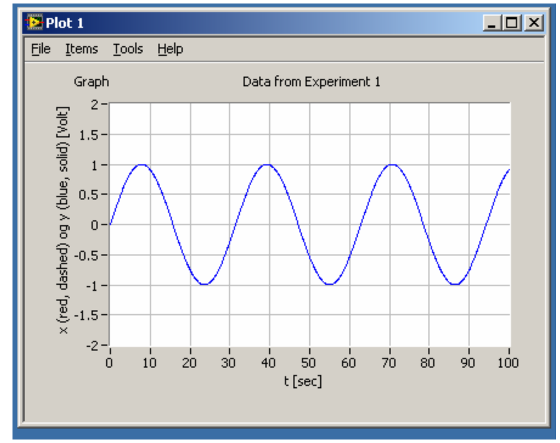
- MathScript can be used in two ways
  - In a MathScript window as a desktop mathematical tool independent of LabVIEW
  - In a MathScript node which appears as a frame inside the Block diagram of a VI (available on the Functions / Mathematics / Scripts & Formulas palette.)

Example 4: Plotting a sine wave

Run



Script Editor

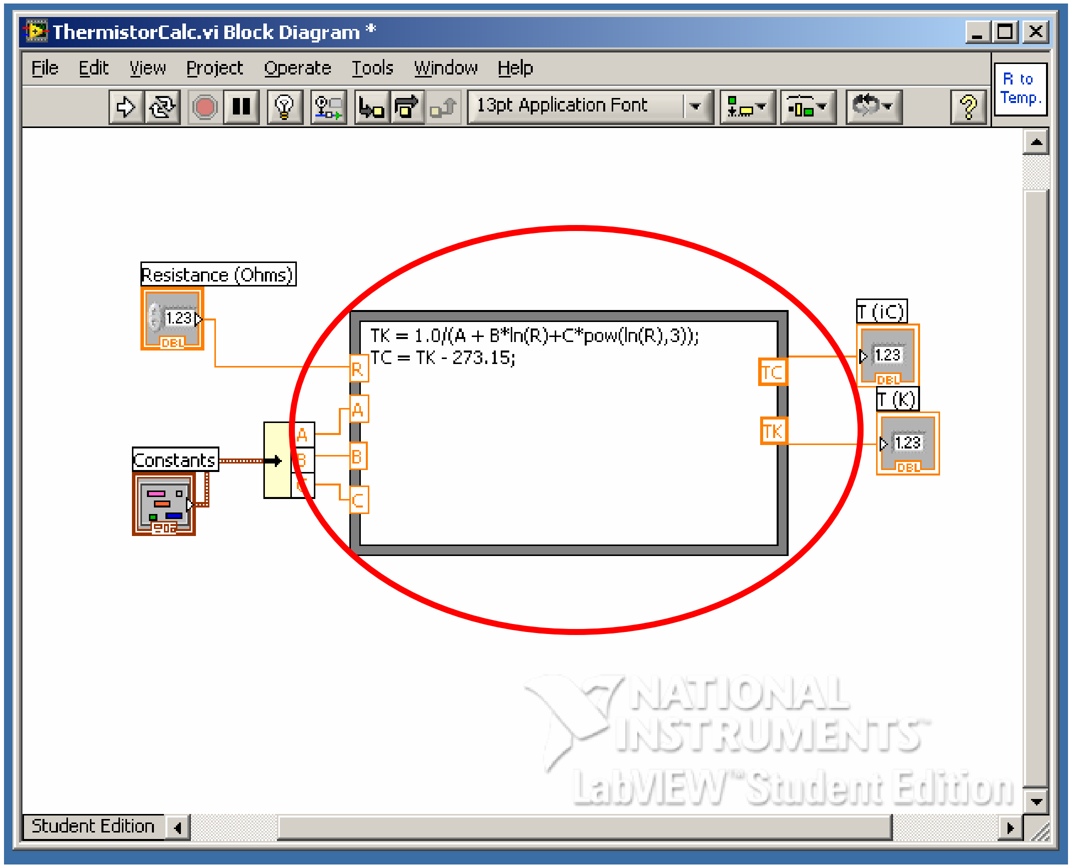


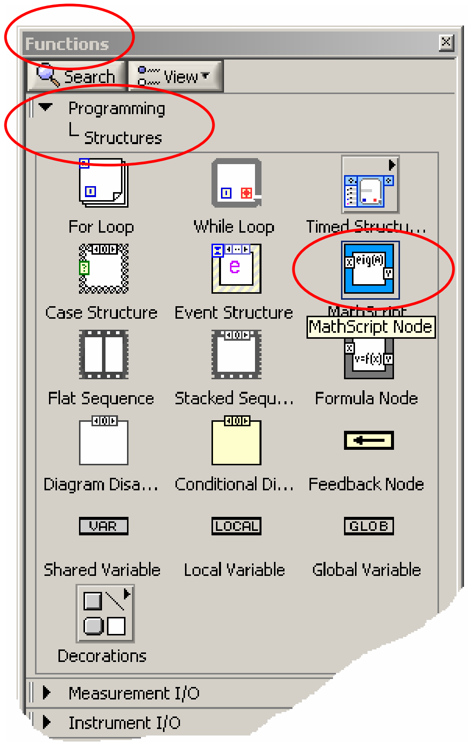
Plot

```
t=[0:.1:100]';
x=-1+0.02*t;
y=sin(0.2*t);
figure(1)
plot(t,y,'b-')  %x(t) in dashed red.
y(t) in solid blue.
xmin=0;xmax=100;ymin=-2;ymax=2;
axis([xmin xmax ymin ymax])
grid
xlabel('t [sec]')
ylabel('x (red, dashed) og y (blue,
solid) [Volt]')
title('Data from Experiment 1')
```

Demonstration IV

Example 5: Embedded Mathscript





Tip: Get MathScript module from Functions tools, under Programming/Structures

$$T^{\circ K} = \frac{1}{A + B\ln(R) + C[\ln(R)]^3} \qquad T^{\circ C} = T^{\circ K} + 273.15$$

Data Acquisition (DAQ) with LabVIEW

- This is where E59 and E80 merge!
- You saw sampling, aliasing, discrete and continuous signals, Bode plot and…E59
- Now you will acquire those signals in real experiments
- LabVIEW helps you as a tool collecting and displaying data

What is the use of data acquisition?

<u>Sensors</u> or <u>transducers</u> as our "sensing" tools convert physical signal to an electrical signal.

1) Need DAQ devices to grab those signals and hand them to computer for display and processing

2) May need to control the flow of data from our transducers (triggering)

3) Will convert continuous time signals to digital which is suitable for computers

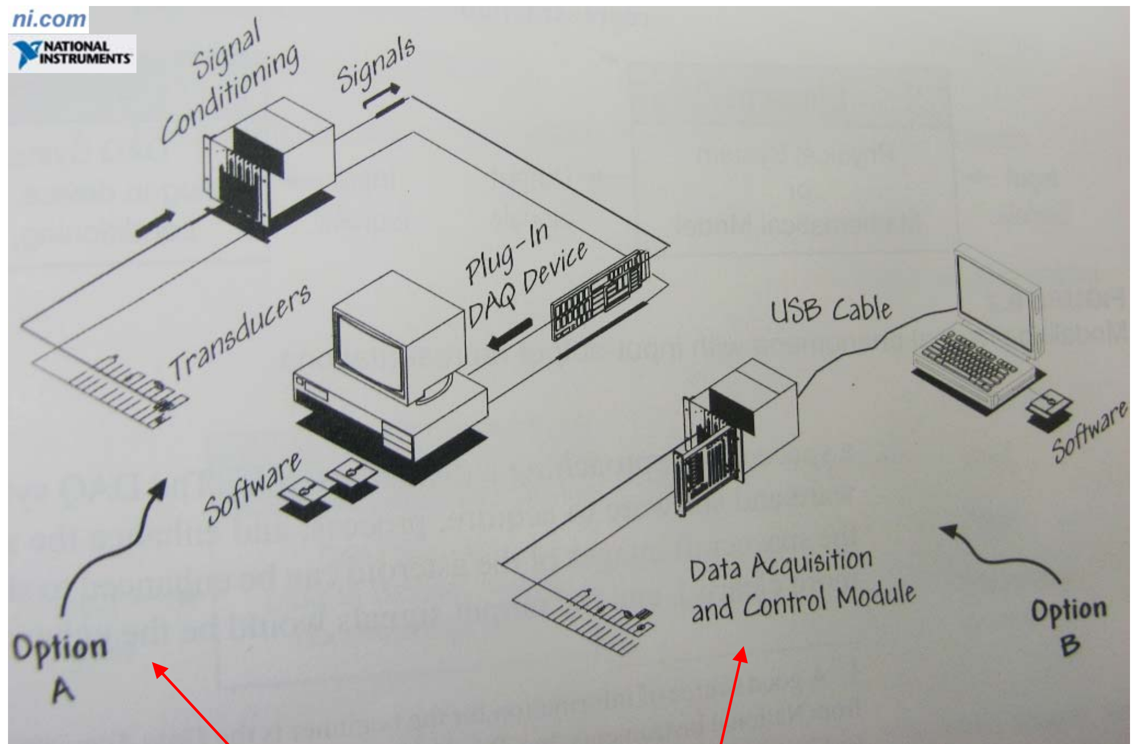


Fig 9.1

DAQ devices can be <u>internal</u> (PCI cards) **or** <u>external</u> (USB)

Types Of Signals

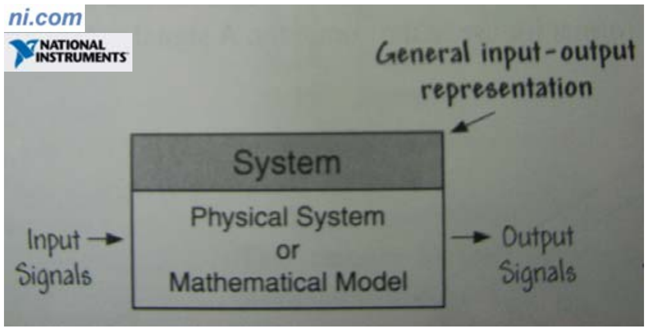


Fig 9.2

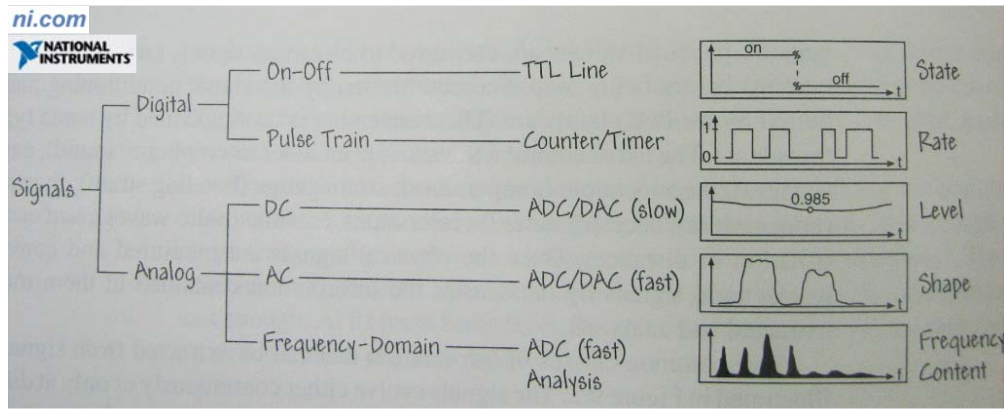


Fig 9.4

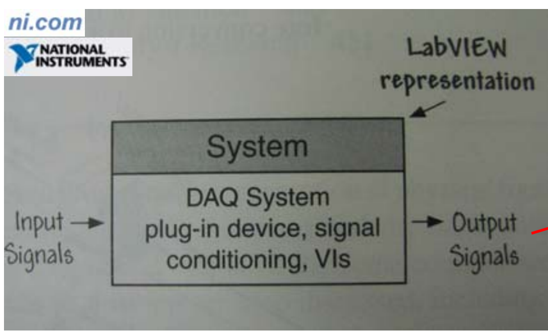


Fig 9.2

- State
- Rate
- Level
- Shape
- Frequency Content

Signal Conditioning

Electrical signal from a transducer may not be very suitable for Analog-to-Digital converters.

↓ ?

Signal conditioning (filtering, amplifying, …)

↓

→ LabVIEW

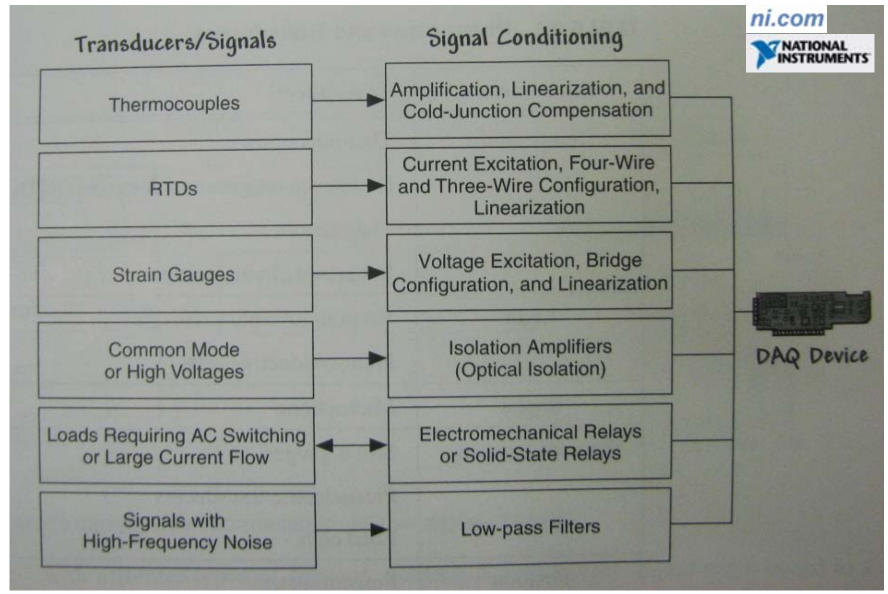
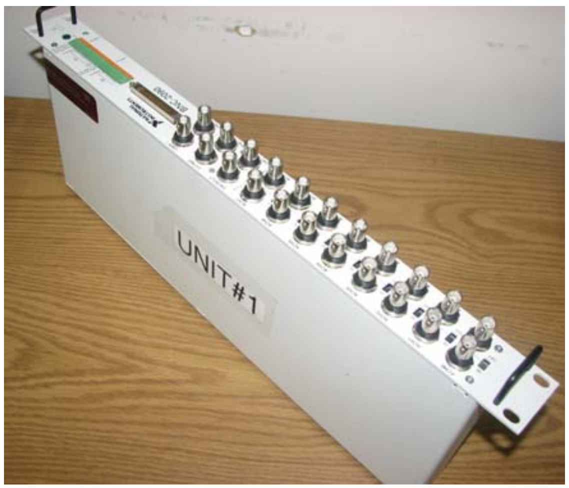


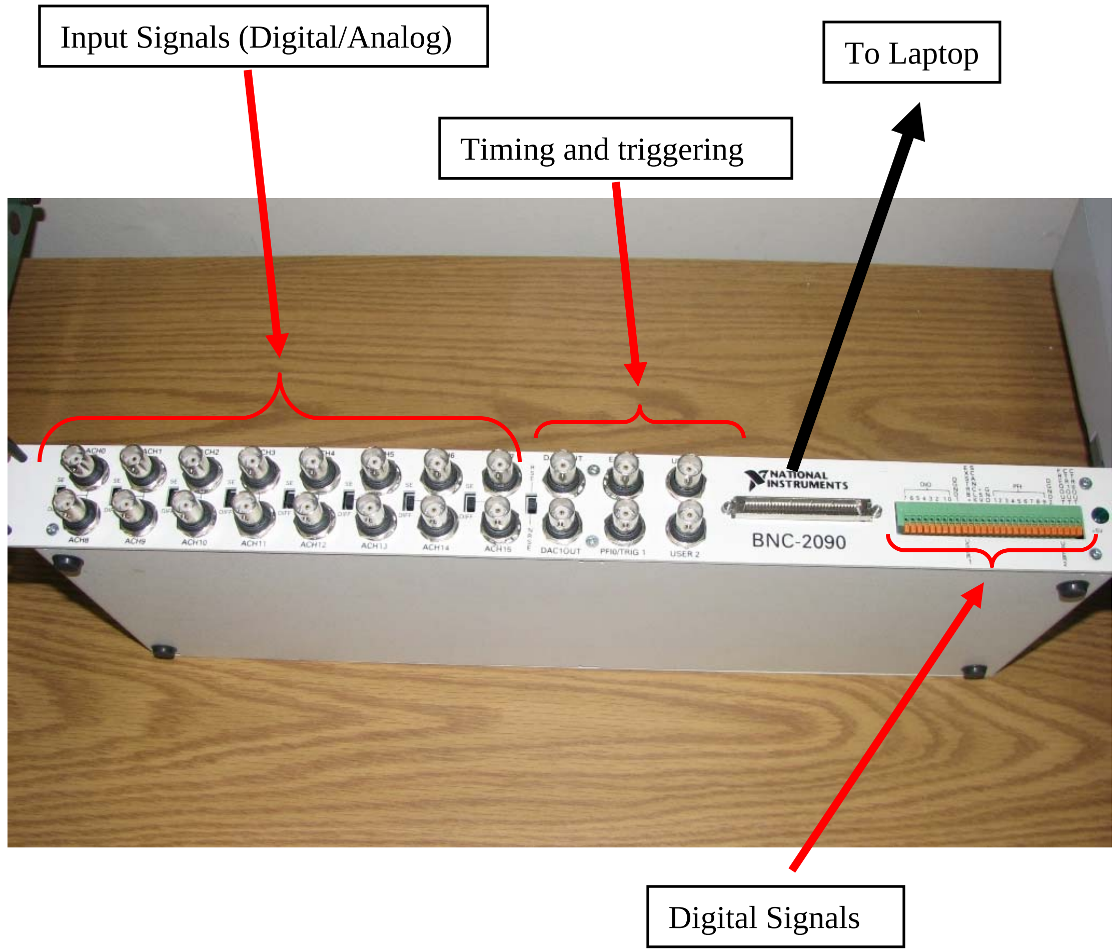
Fig 9.4

Note: Your DAQ may include built-in signal conditioning

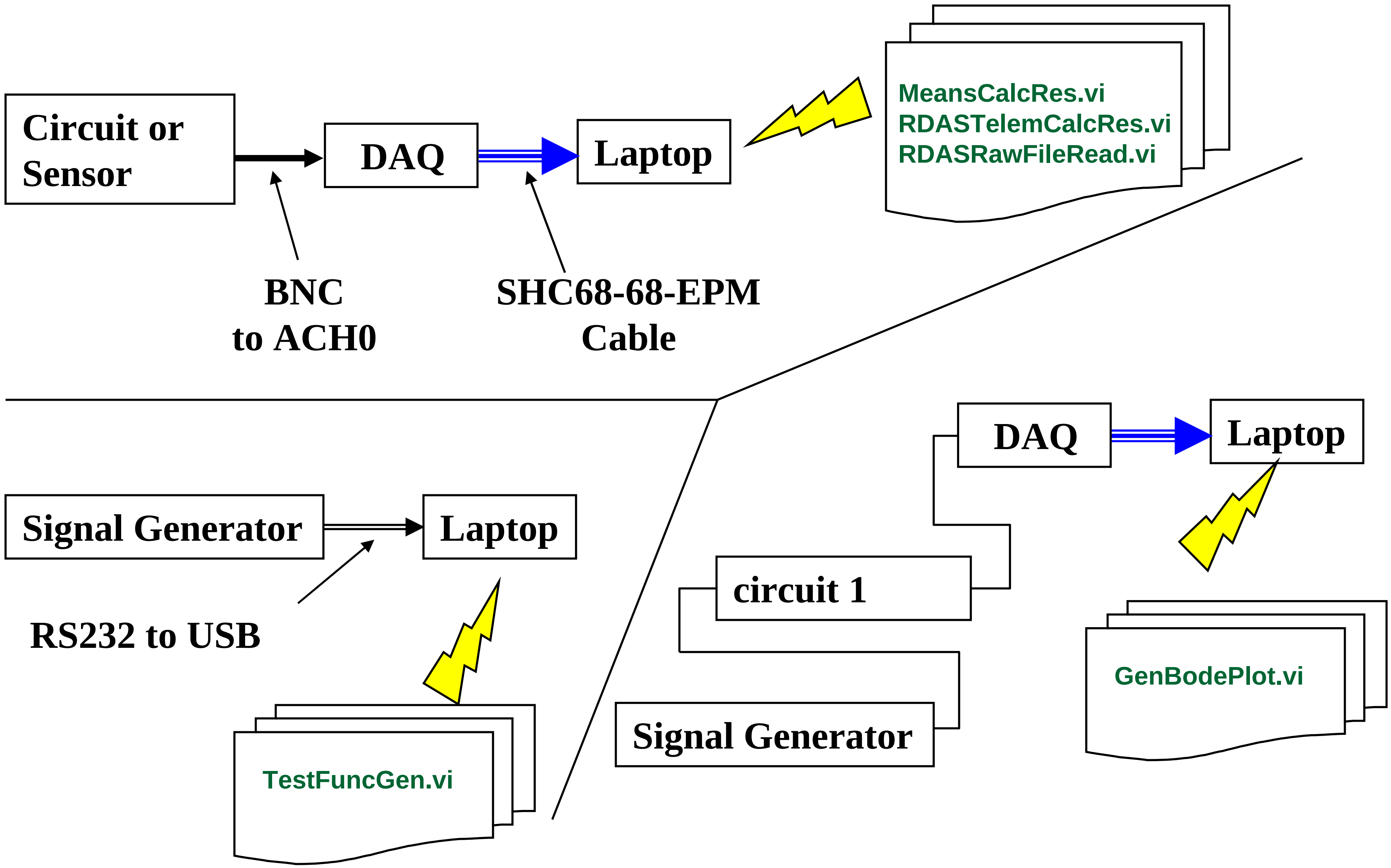
BNC-2090 DAQ at Mudd



BNC-2090 DAQ



LabView and DAQ in the lab



Matlab

Most of the concepts discussed for LabVIEW are valid for MATLAB

Key differences:

1) MATLAB has it's own language and commands
2) Unlike LabVIEW, mainly commands and scripts are needed to run the code
3) To interface the DAQ and other instruments with MATLAB need MATLAB drivers

Summary

Remember the following tools/skills/knowledge …

- Knowing basic electrical measurement techniques
- Understanding the concept of sensors/transducers
- Acquiring data from sensors instrumentation
- Communicating with the PC using DAQ
- Analyzing and presenting the data