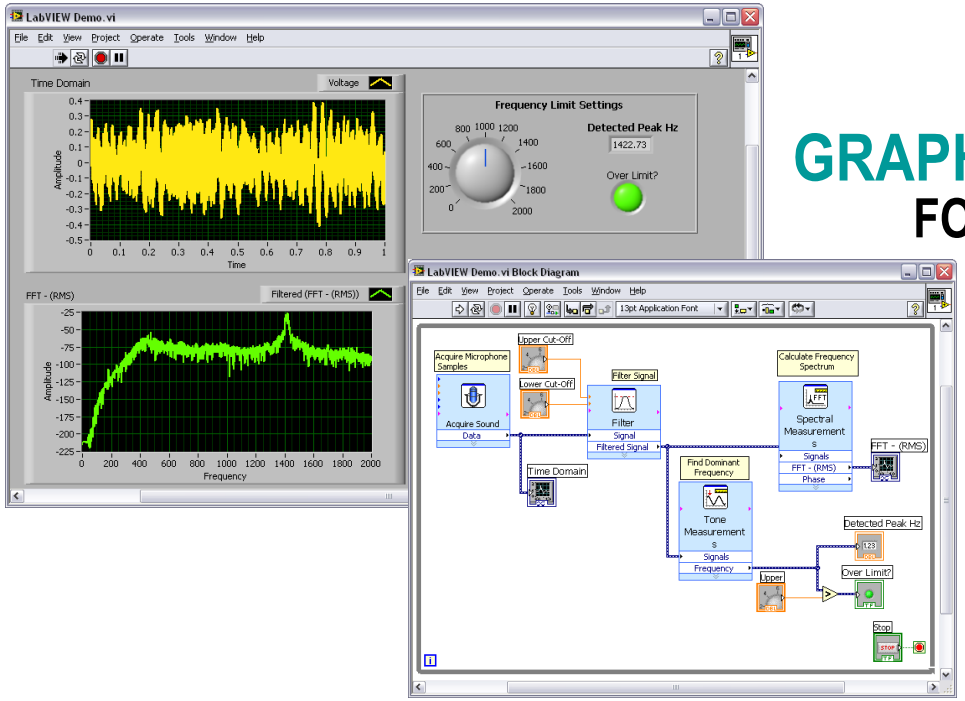
Introduction to LabVIEW

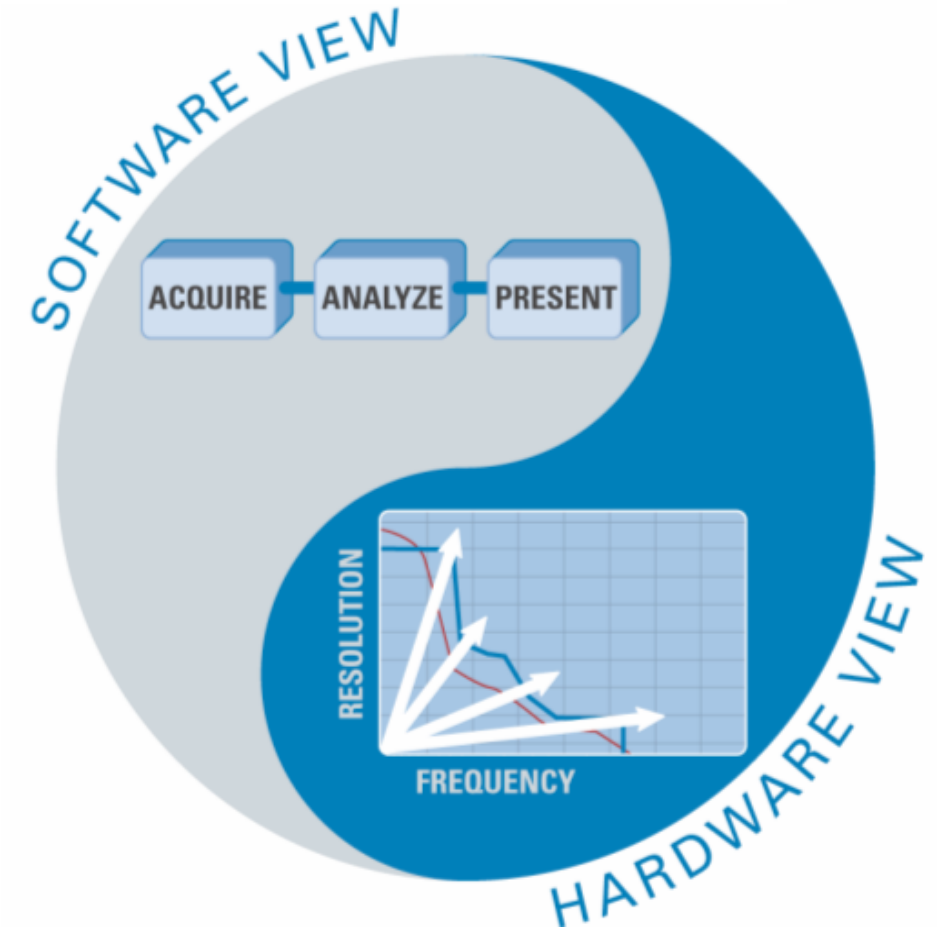


GRAPHICAL PROGRAMMING
FOR ENGINEERS AND SCIENTISTS

LabVIEW™ 8.6

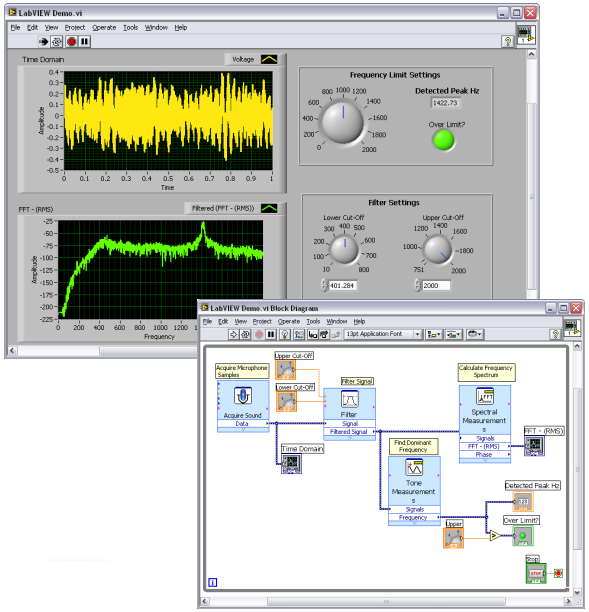
6-Hour Hands-On

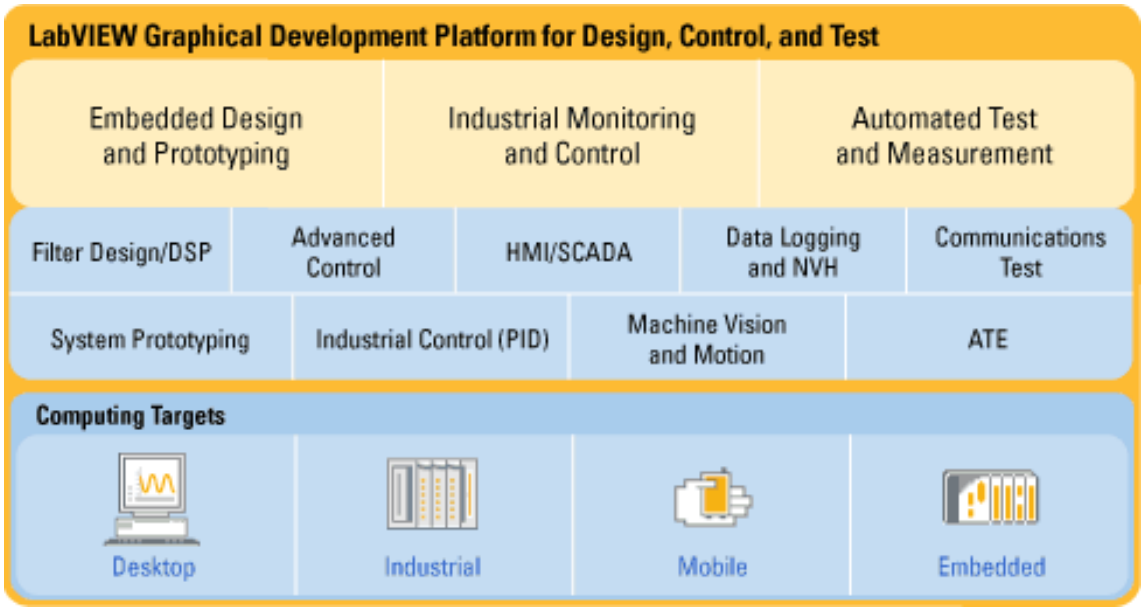
The Virtual Instrumentation Approach



LabVIEW Graphical Development System

- Graphical programming environment
- Compile code for multiple OS and devices
- Useful in a broad range of applications

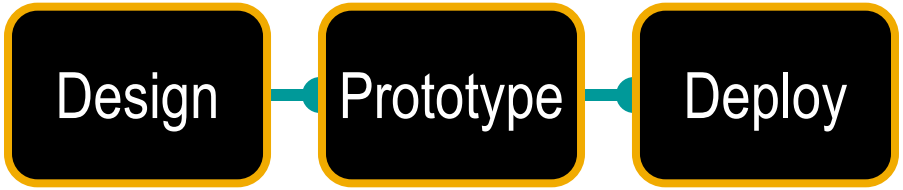




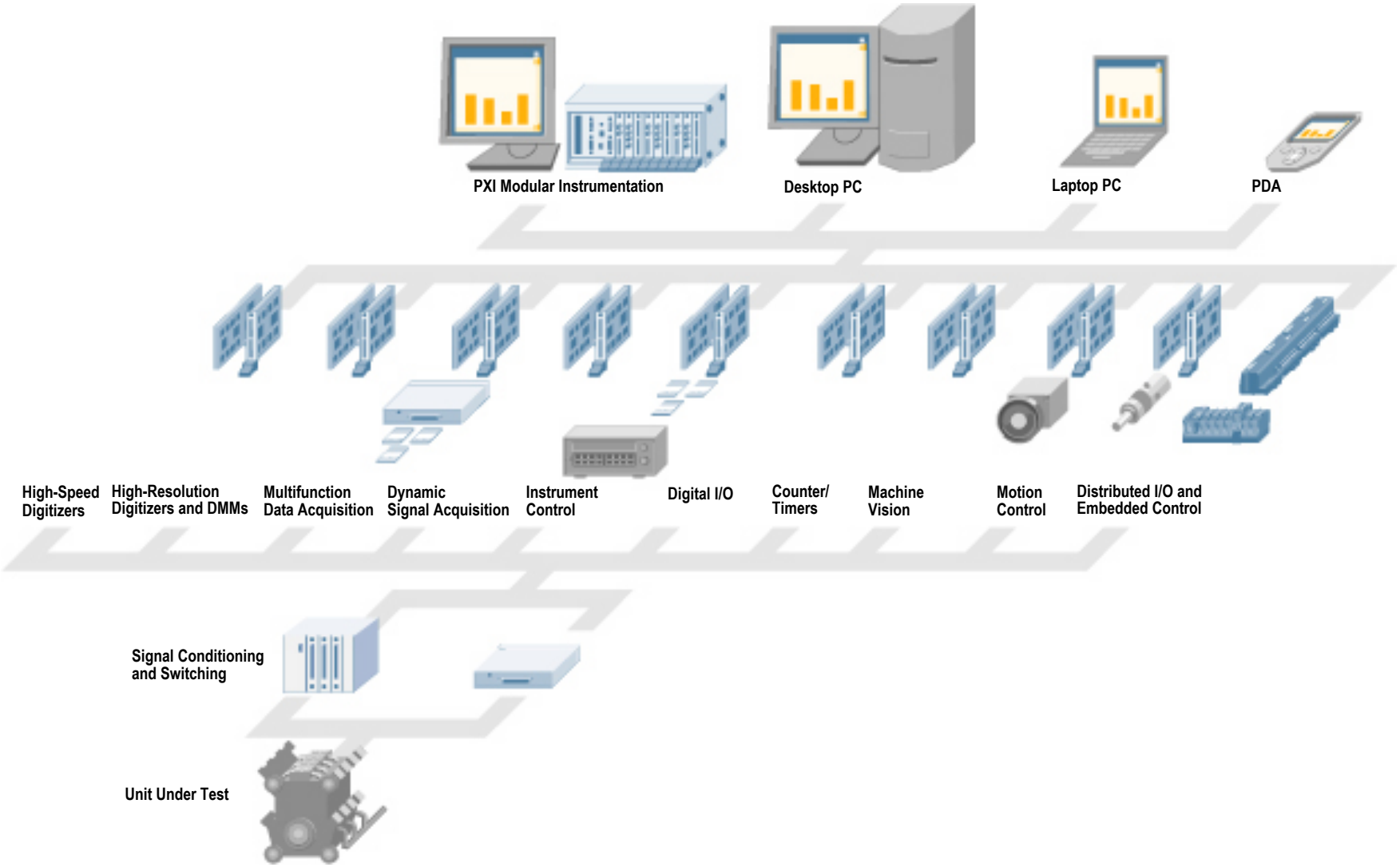
Virtual Instrumentation Applications

- **Design**
  - Signal and image processing
  - Embedded system programming
    - (PC, DSP, FPGA, microcontroller)
  - Simulation and Prototyping
  - And more …
- **Control**
  - Automatic controls and dynamic systems
  - Mechatronics and robotics
  - And more …
- **Measurements**
  - Circuits and electronics
  - Measurements and instrumentation
  - And more …

A single graphical development platform



The NI Approach – Integrated Hardware Platforms



Section I – LabVIEW Environment

A. Getting Data into Your Computer
- Data Acquisition Devices
  - NI-DAQmx
  - Simulated data acquisition
  - Sound card

B. LabVIEW Environment
- Front Panel/Block Diagram
- Toolbar/Tools Palette

C. Components of a LabVIEW Application
- Creating a VI
- Dataflow Execution

D. Additional Help
- Finding Functions
- Tips for Working in LabVIEW

A. Setting Up Your Hardware

- Data Acquisition Device (DAQ) Track A
  - Actual USB, PCI, or PXI Device
  - Configured in Measurement and Automation Explorer(MAX)

- Simulated Data Acquisition Device (DAQ) Track B
  - Software simulated at the driver level
  - Configured in MAX

- Sound Card
  - Built into most computers Track C

What Type of Device Should I Use?





	Sound Card*	NI USB DAQ	NI PCI DAQ	Instruments*
AI Bandwidth	8 to 44 kS/s	10 kS/s to 1.25 MS/s	20 kS/s to 10 MS/s	100 S/s to 2 GS/s
Accuracy	12 to 16 bits	12 to 18 bits	12 to 18 bits	8 to 26 bits
Portable	√	√	—	some
AI Channels	2	8 to 80	2 to 80	1 to 80
AO Channels	2	2 to 4	2 to 8	2 to 8
AC or DC	AC	AC/DC	AC/DC	AC/DC
Triggering	—	√	√	√
Calibrated	—	√	√	√

* The above table may not be representative of all device variations that exist in each category

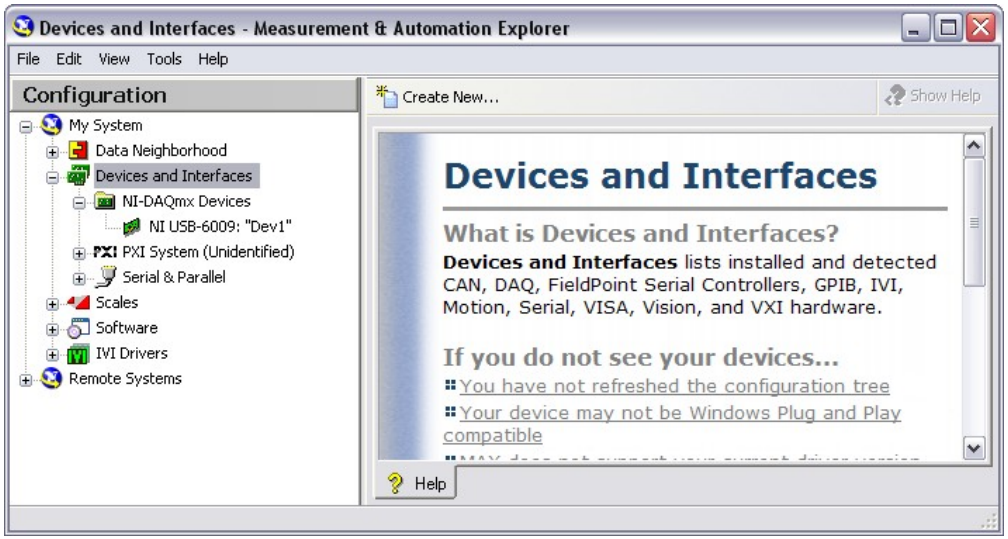
What is MAX?

- Stands for Measurement & Automation Explorer
- Configures and organizes all your National Instruments DAQ, PCI/PXI, GPIB, IMAQ, IVI, motion, VISA, and VXI devices
- Tests devices

Icon Found on

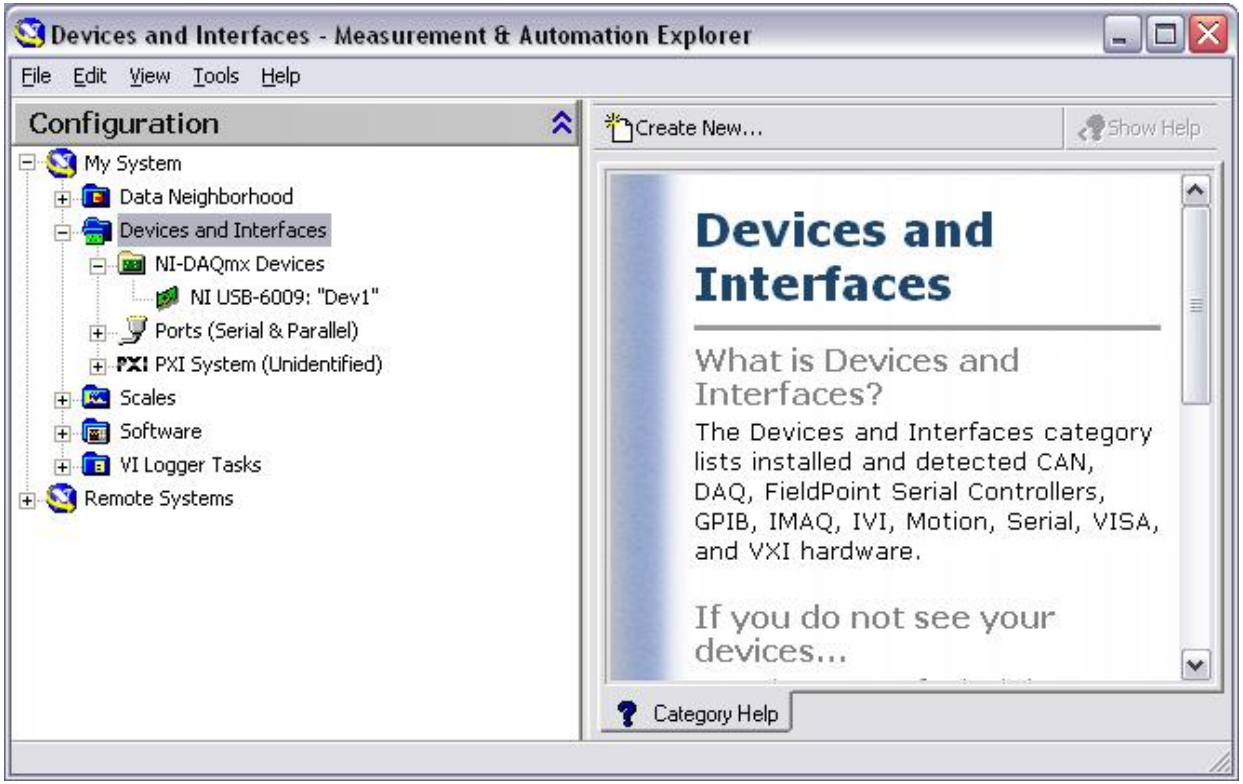
Windows Desktop

Measurement
& Automation



Exercise 1 – Setting Up Your Device

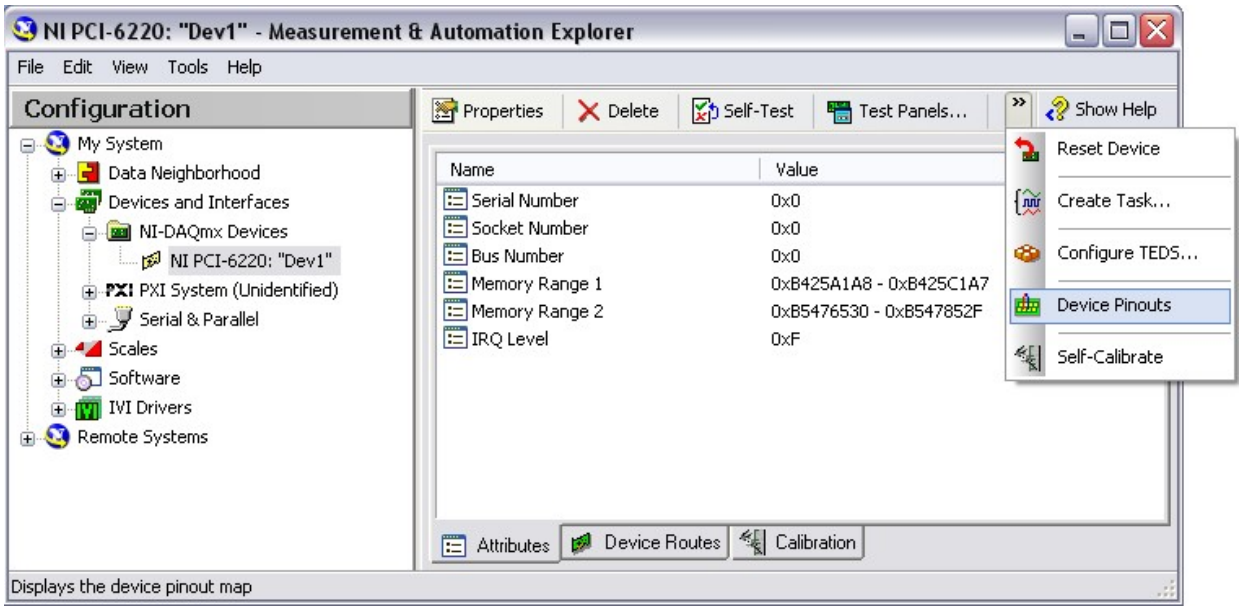
- Use Measurement and Automation Explorer (MAX) to:
  - Configure and test your Data Acquisition (DAQ) device



Track A

Exercise 1 – Setting Up Your Device

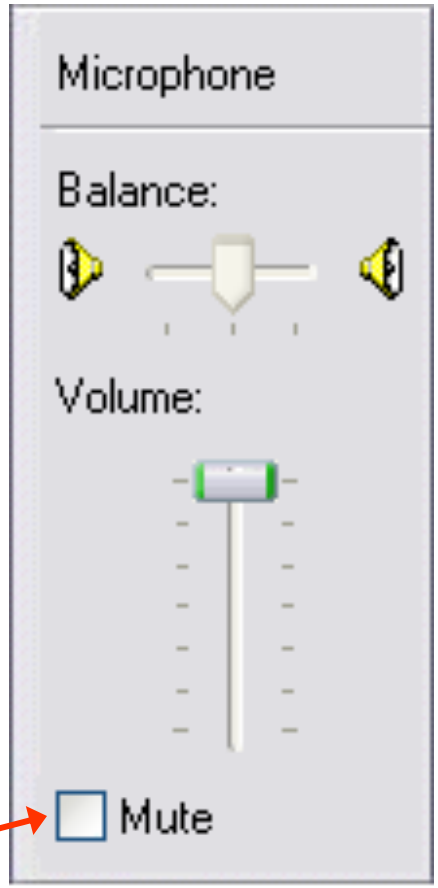
- Use Measurement and Automation Explorer (MAX) to:
  - Configure and test your Simulated Data Acquisition (DAQ) device



Track B

Exercise 1 – Setting Up Your Device

- Use Windows to:
  - Verify your Sound Card

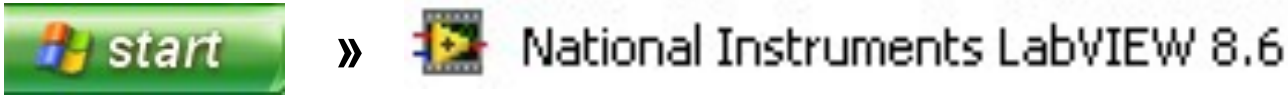


Un-Mute Microphone

Track C

Open and Run LabVIEW

Start»All Programs»National Instruments LabVIEW 8.6



Startup Screen:

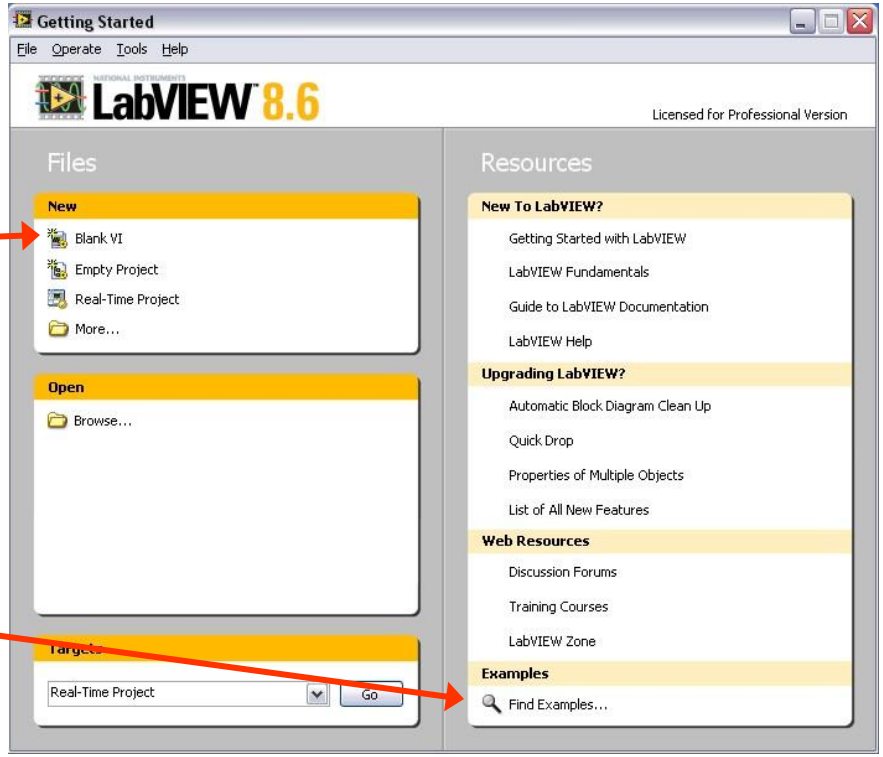
Start from a blank VI:

New»Blank VI

or

Start from an example:

Examples»Find Examples…

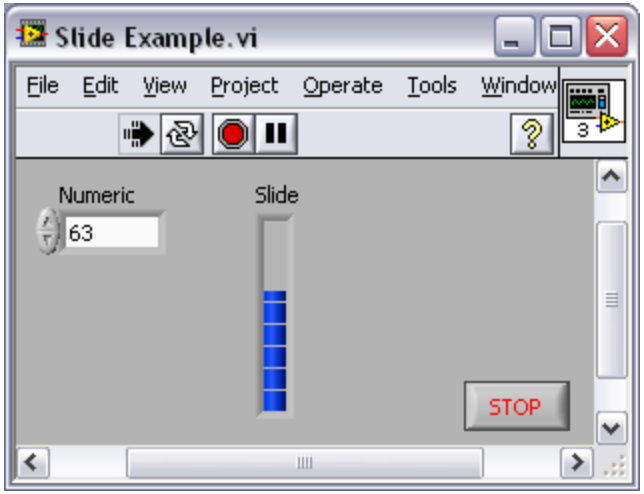


LabVIEW Programs Are Called Virtual Instruments (VIs)

Each VI has 2 windows

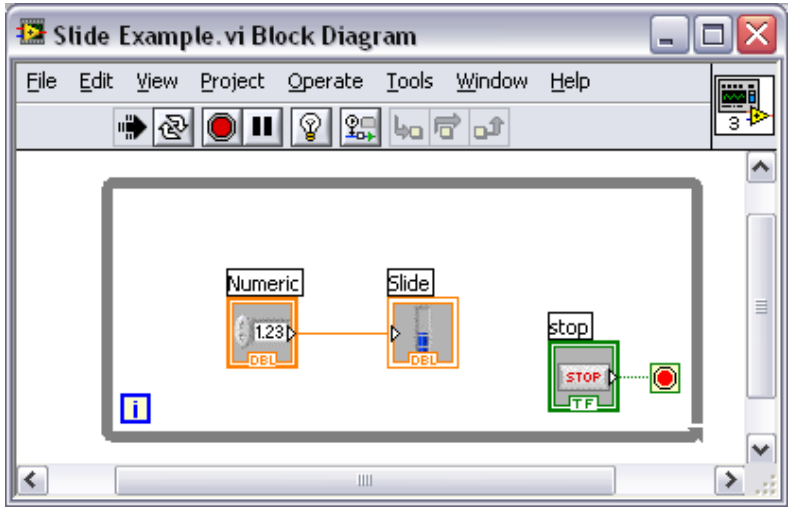
Front Panel

- User interface (UI)
  - Controls = inputs
  - Indicators = outputs



Block Diagram

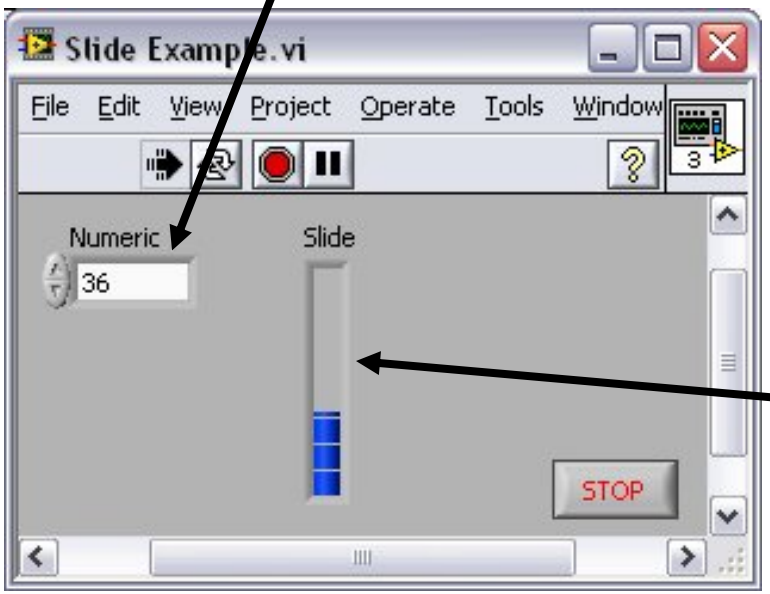
- Graphical code
  - Data travels on wires from controls through functions to indicators
  - Blocks execute by data flow

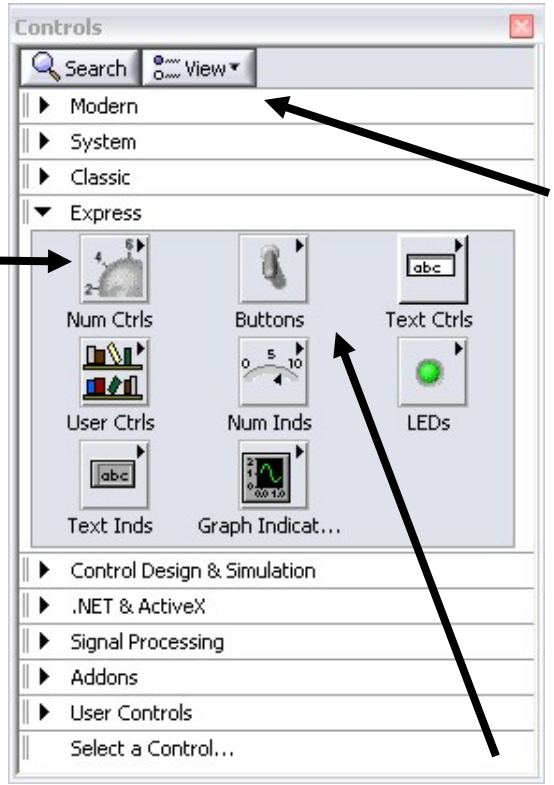


Controls Palette

(Place items on the front panel window)

Control: Numeric

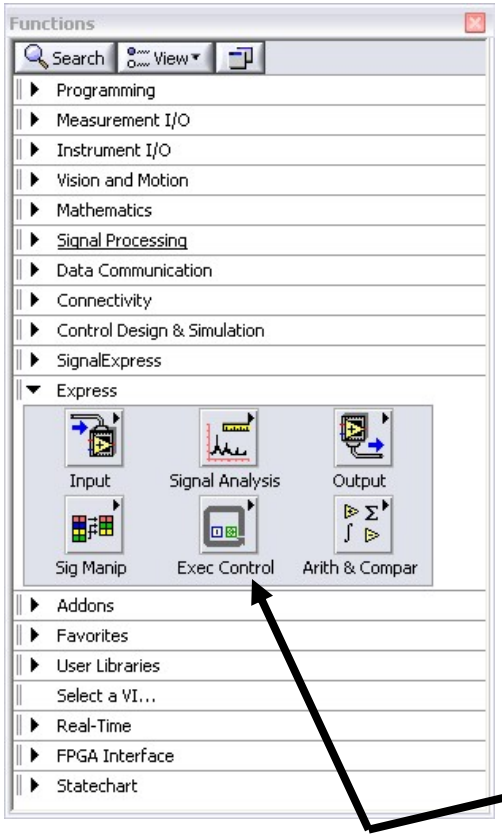




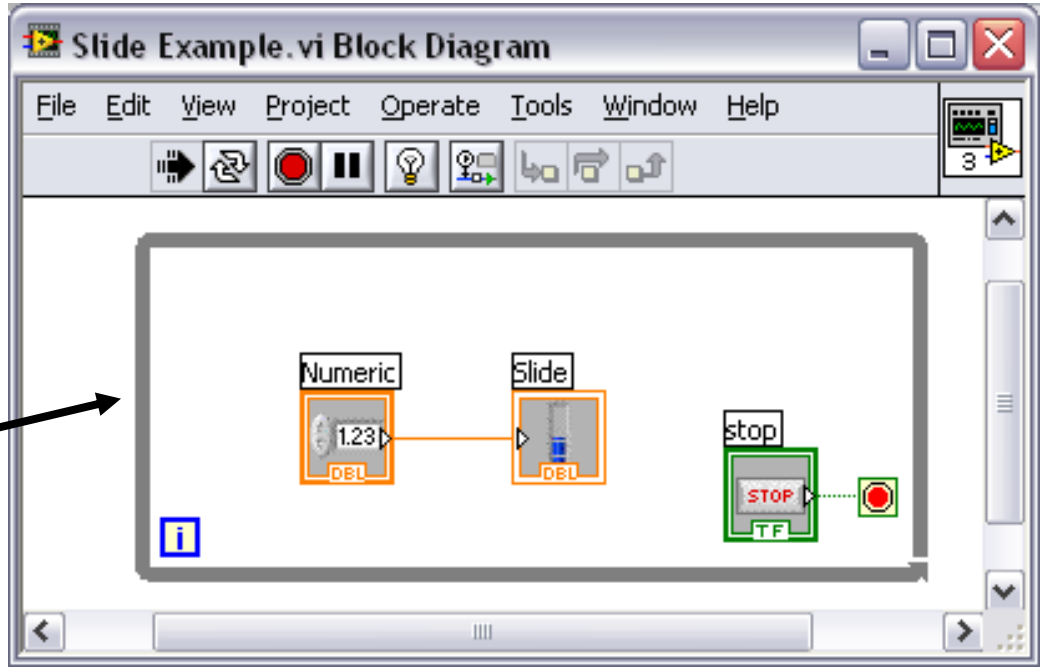
Customize Palette View

Indicator: Numeric Slide

Functions (and Structures) Palette

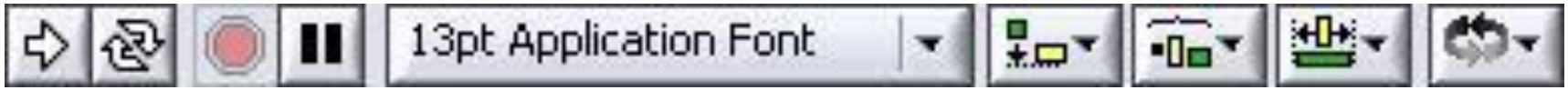


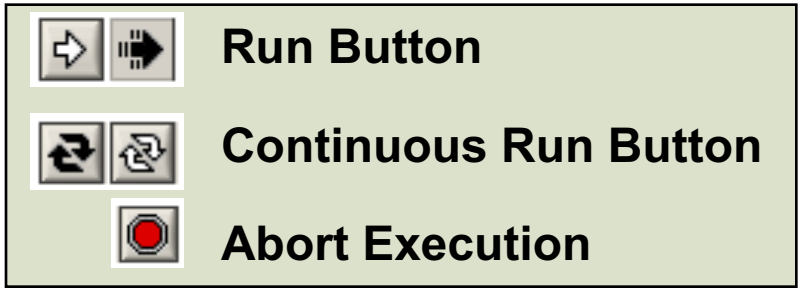
(Place items on the block diagram Window)



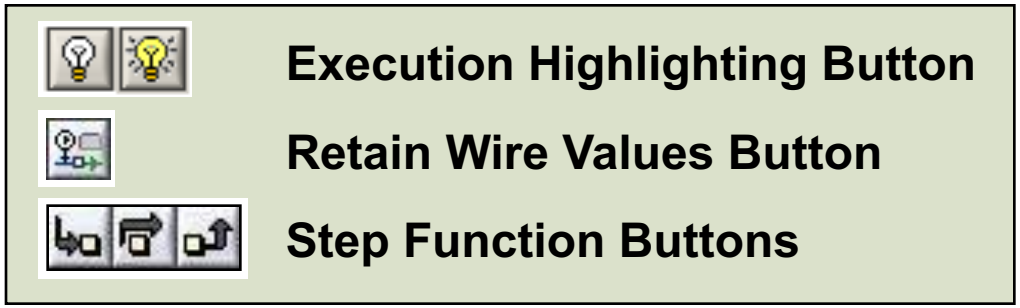
Structure: While Loop

Status Toolbar



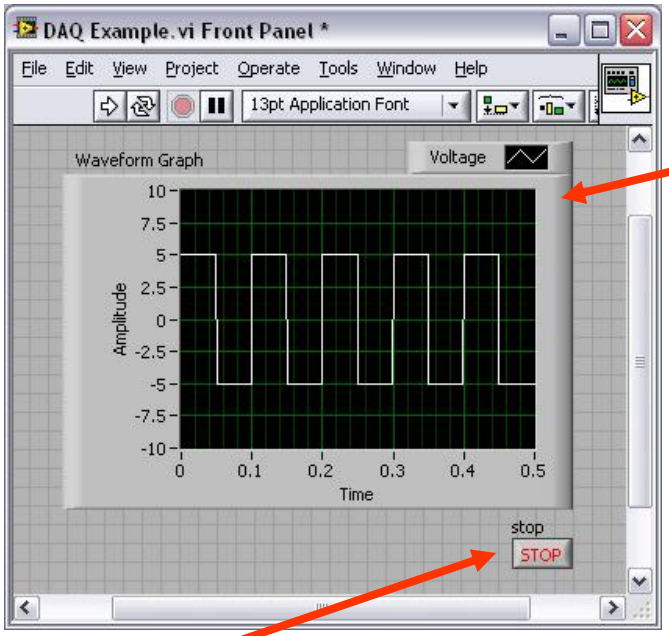


Additional Buttons on the Diagram Toolbar



Demonstration 1: Creating a VI

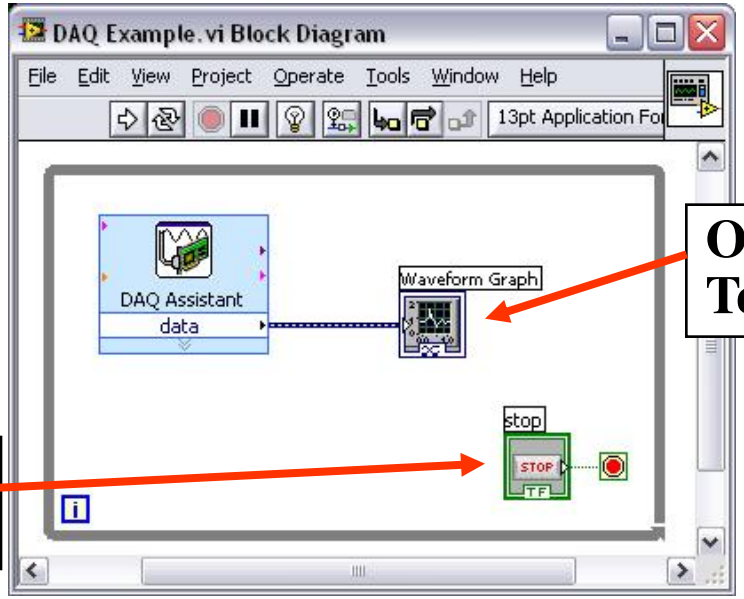
Front Panel Window



Graph Indicator

Boolean Control

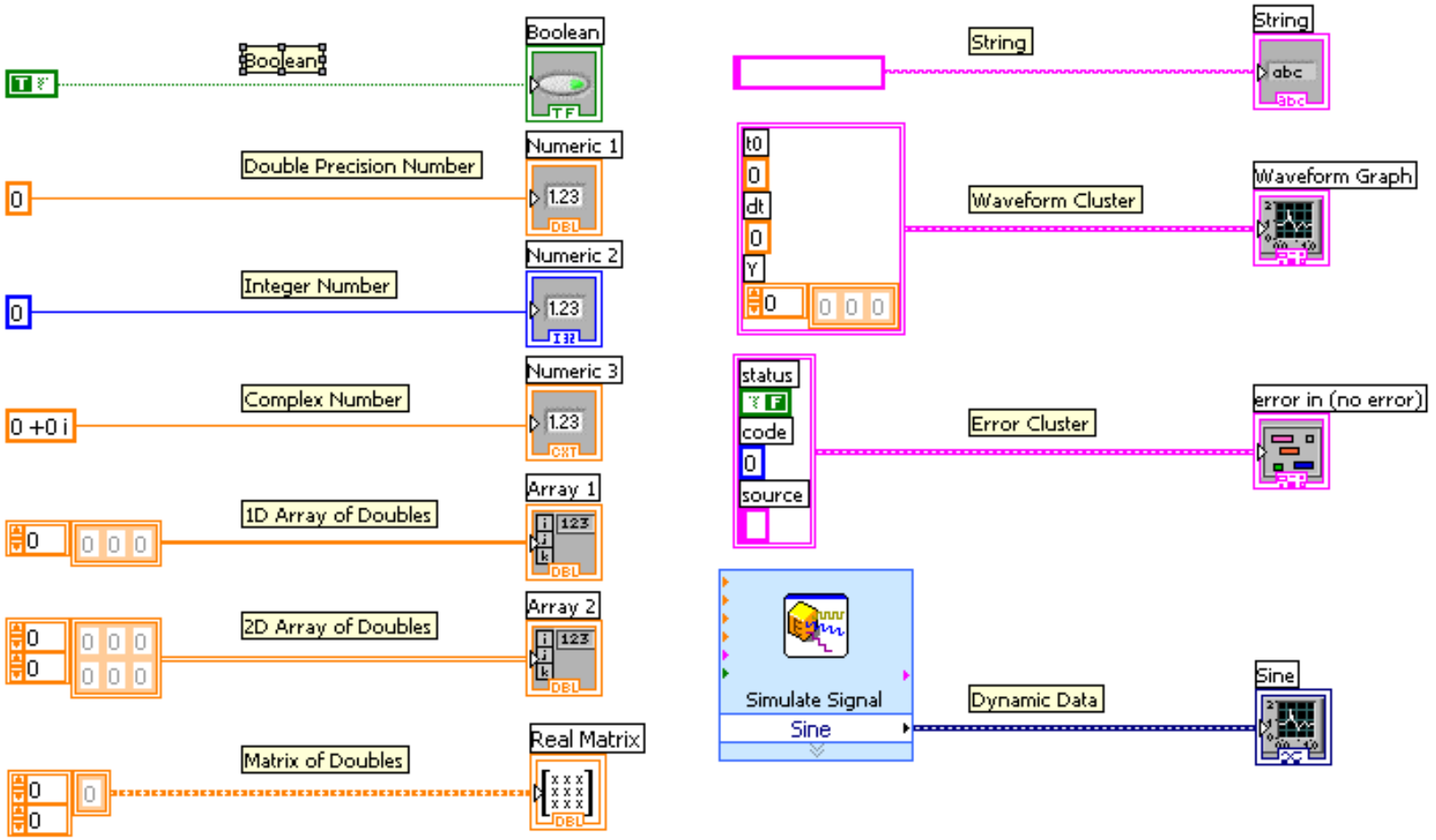
Block Diagram Window



Output Terminal

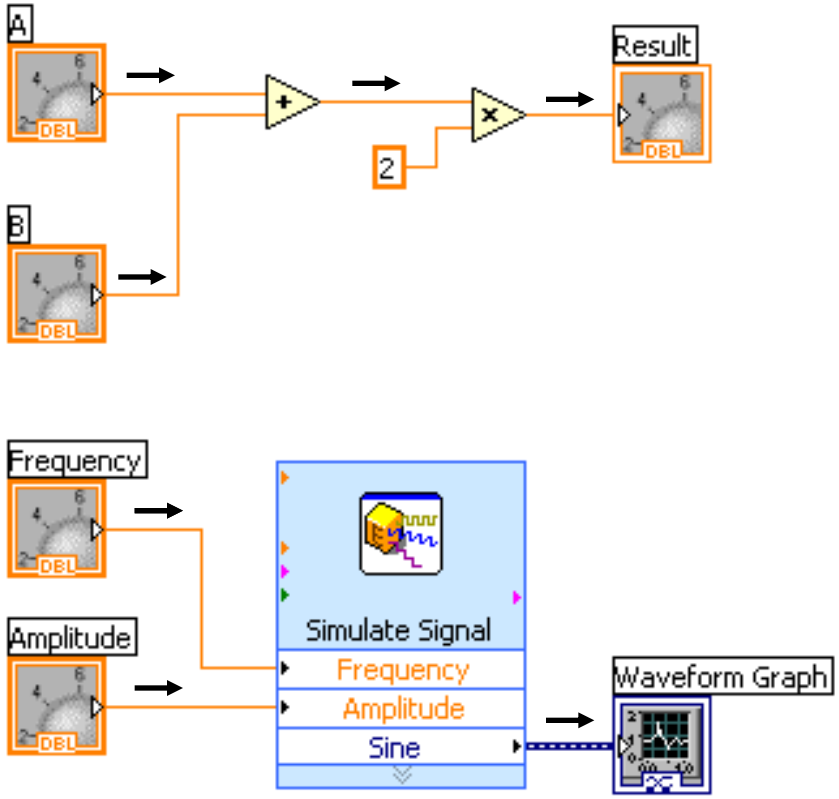
Input Terminals

Data Types Found in LabVIEW



Dataflow Programming

- Block diagram execution
  - Dependent on the flow of data
  - Block diagram does NOT execute left to right
- Node executes when data is available to ALL input terminals
- Nodes supply data to all output terminals when done



Debugging Techniques

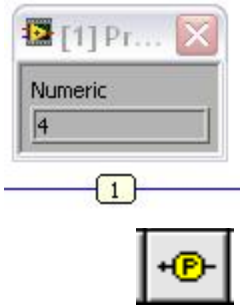
- **Finding Errors**

Click on broken **Run** button.
Window showing error appears.

- **Execution Highlighting**

Click on **Execution Highlighting** button; data flow is animated using bubbles. Values are displayed on wires.

- **Probes**

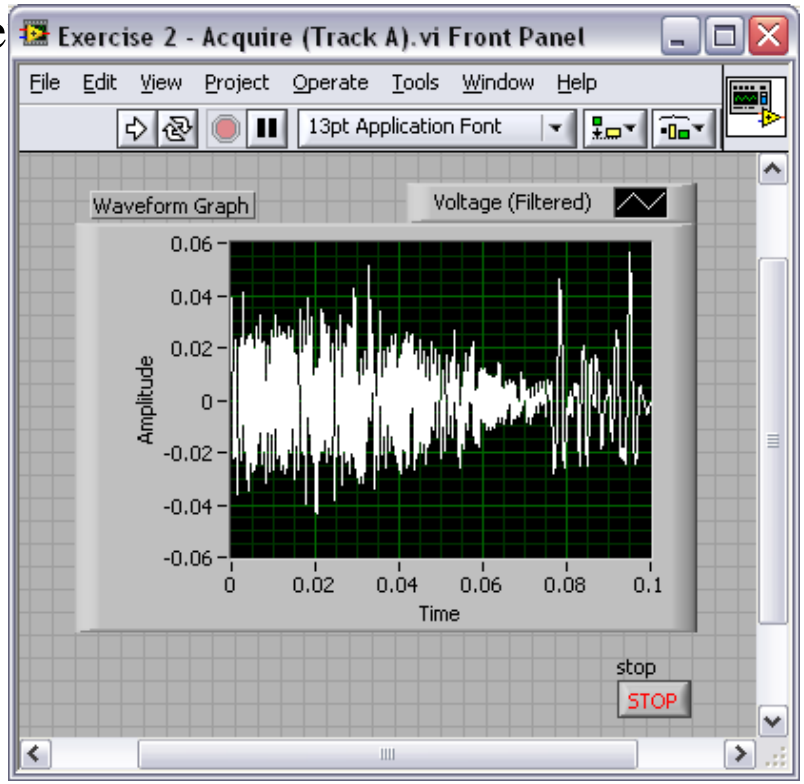


Right-click on wire to display probe; it shows data as it flows through wire segment.

You can also select Probe tool from Tools palette and click on wire.

Exercise 2 – Acquiring a Signal with DAQ

- Use a LabVIEW template to:
  - Acquire a signal from your DAQ de



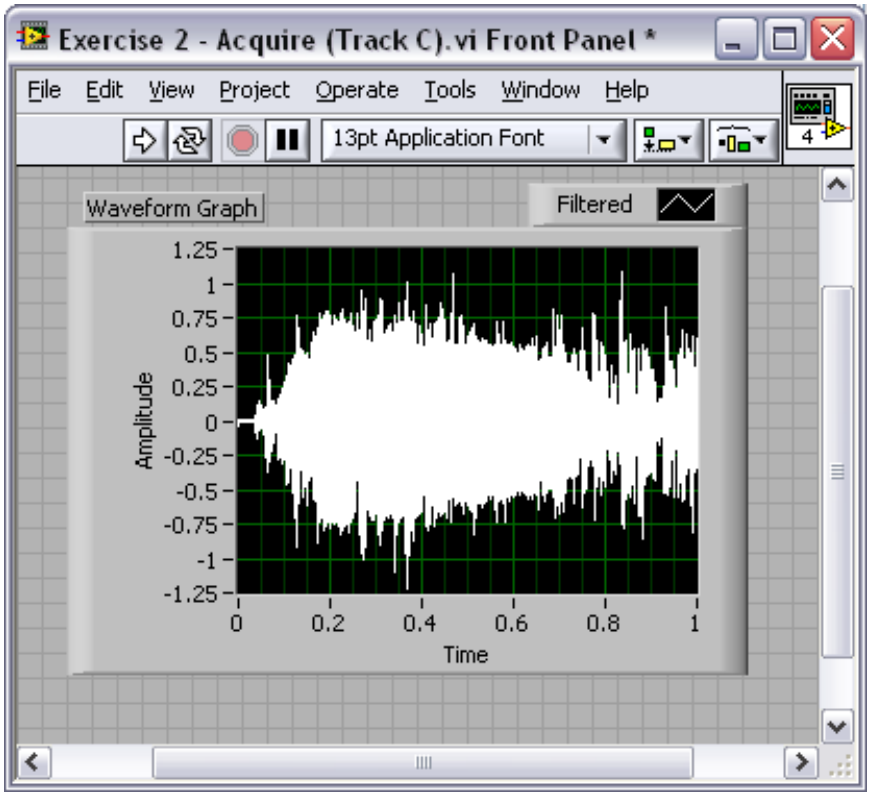
This exercise should take 15 minutes.

Track A&B

Exercise 2 – Acquiring a Signal with the Sound Card

- Use LabVIEW to:
  - Acquire a signal from your sound card

This exercise should take 15 minutes.



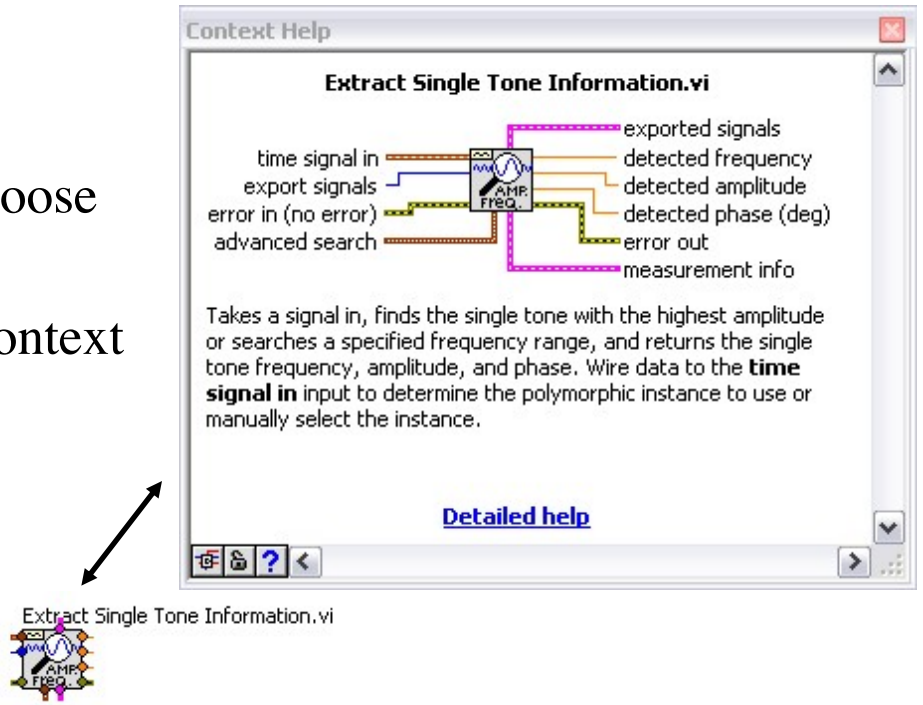
Track C

Context Help Window

- **Help»Show Context Help**, press the <Ctrl+H> keys
- Hover cursor over object to update window

Additional Help

- Right-click on the VI icon and choose **Help**, or
- Choose "**Detailed help**" on the context help window



Tips for Working in LabVIEW

- Keystroke Shortcuts
  - <Ctrl+H> – Activate/Deactivate Context Help Window
  - <Ctrl+B> – Remove Broken Wires from Block Diagram
  - <Ctrl+E> – Toggle between Front Panel and Block Diagram
  - <Ctrl+Z> – Undo (also in Edit menu)
- **Tools»Options…** – Set Preferences in LabVIEW
- **File»VI Properties** – Configure VI Appearance, Documentation, and so on

Section II – Elements of Typical Programs

A. Loops

- While Loop
- For Loop

B. Functions and SubVIs

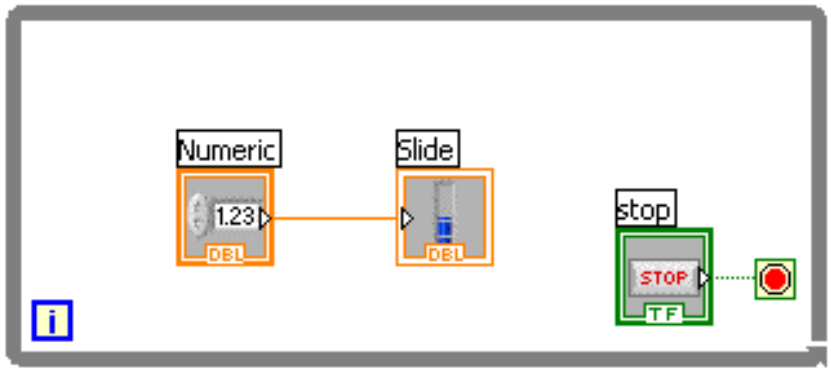
- Types of Functions
- Creating Custom Functions (subVI)
- Functions Palette and Searching

C. Decision Making and File I/O

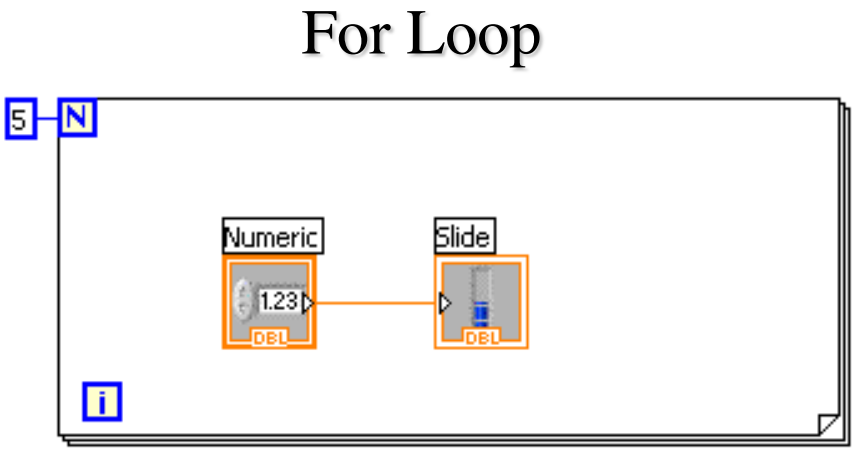
- Case Structure
- Select (simple If statement)
- File I/O

Loops

- While Loop
  - i Terminal counts iterations
  - Always runs at least once
  - Runs until stop condition is met

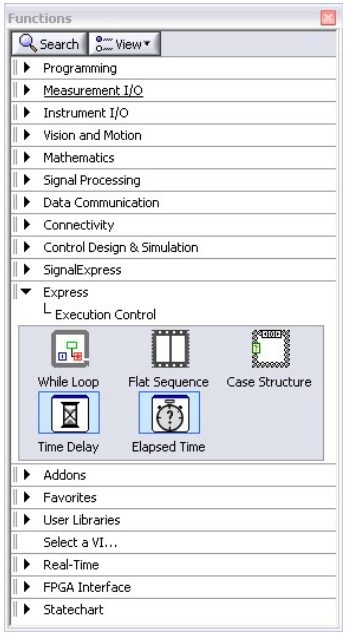


- For Loop
  - i Terminal counts iterations
  - Runs according to input **N** of count terminal N

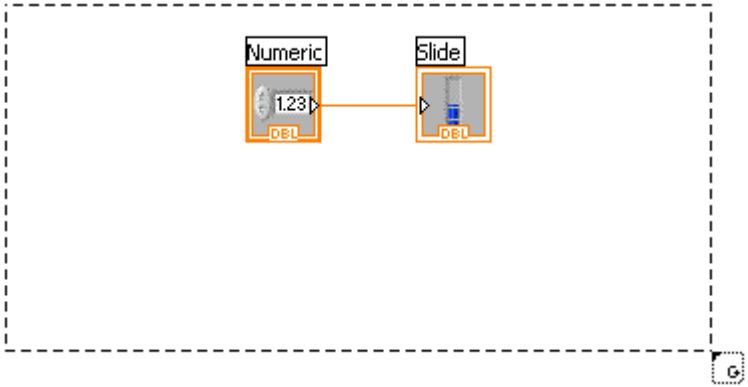


Drawing a Loop

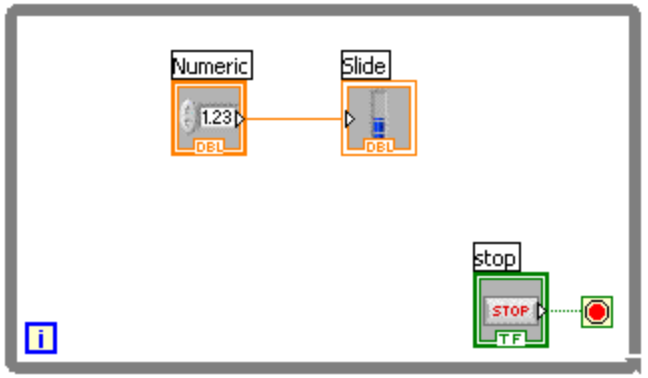
1. Select the structure



2. Enclose code to be repeated

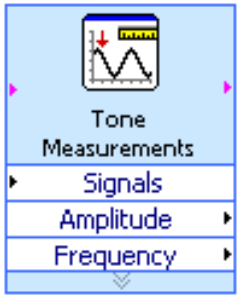


3. Drop or drag additional nodes and then wire

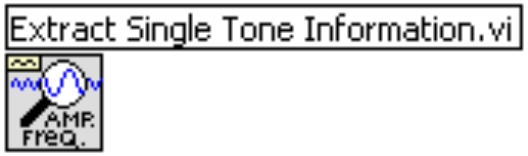


3 Types of Functions (from the Functions Palette)

Express VIs: interactive VIs with configurable dialog page **(blue border)**



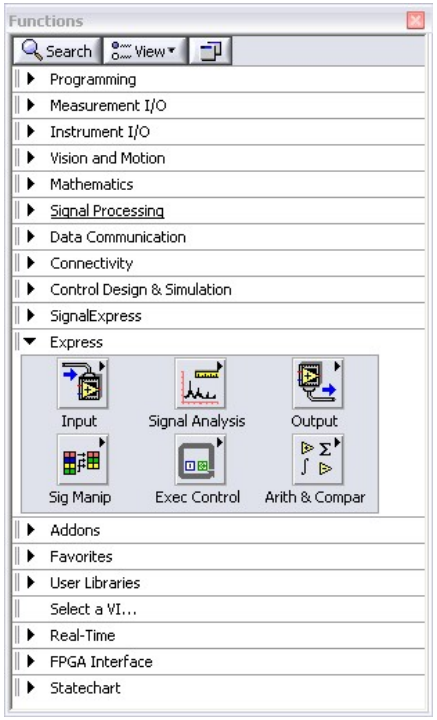
Standard VIs: modularized VIs customized by wiring **(customizable)**



Functions: fundamental operating elements of LabVIEW; no front panel or block diagram **(yellow)**

What Types of Functions Are Available?

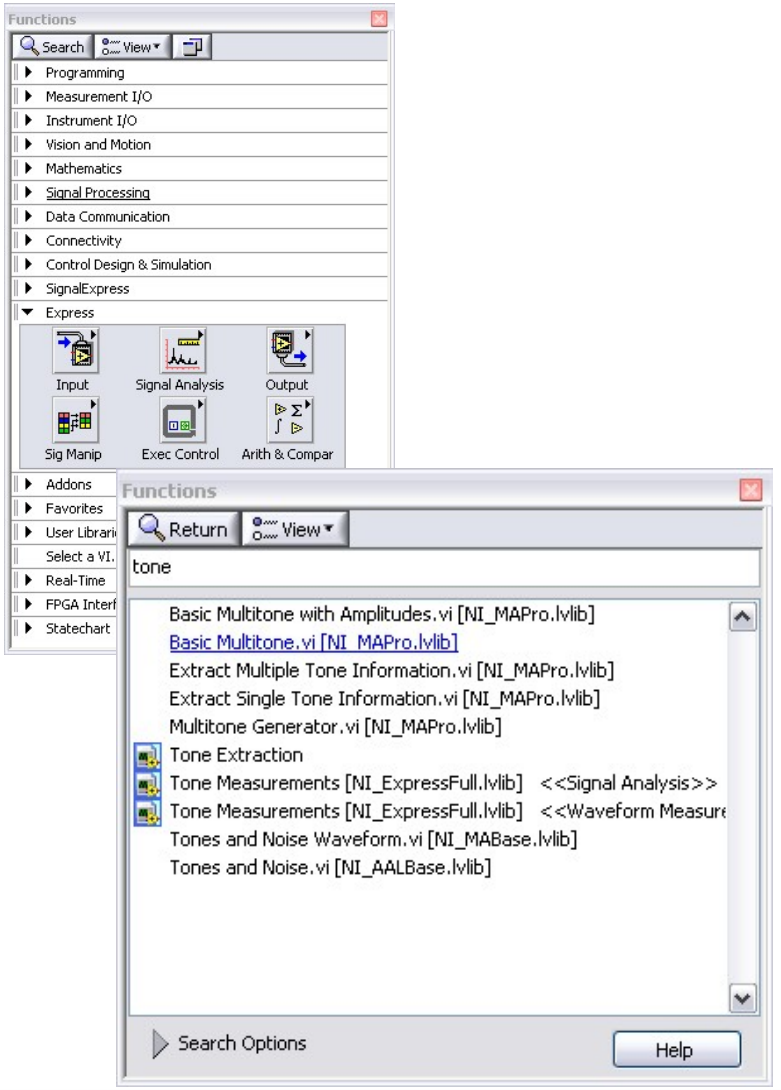
- **Input and Output**
  - Signal and data simulation
  - Real signal acquisition and generation with DAQ
  - Instrument I/O Assistant (Serial and GPIB)
  - ActiveX for communication with other programs
- **Analysis**
  - Signal processing
  - Statistics
  - Advanced math and formulas
  - Continuous time solver
- **Storage**
  - File I/O



Express Functions Palette

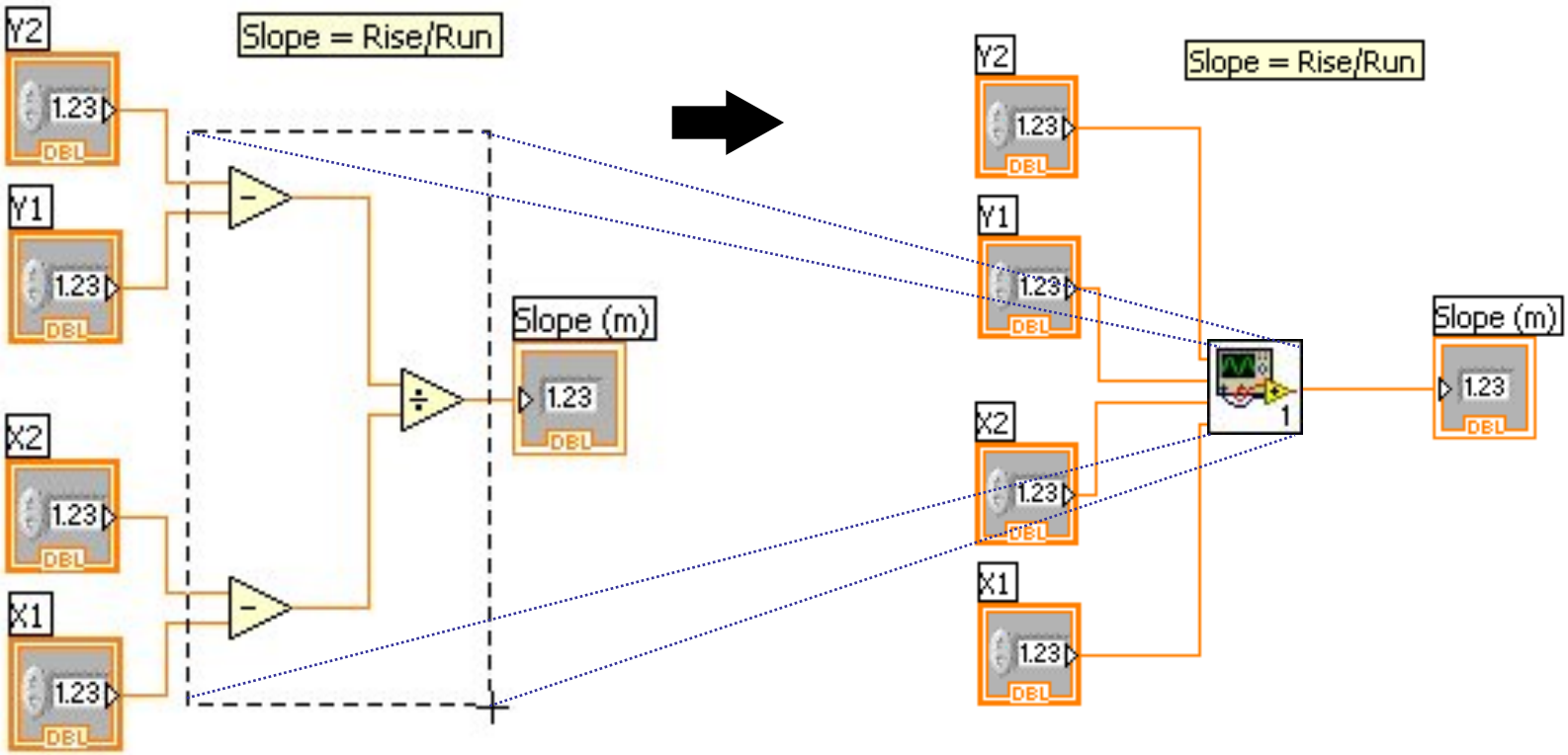
Searching for Controls, VIs, and Functions

- Palettes are filled with hundreds of VIs
- Press the search button to index all VIs for text searching
- Click and drag an item from the search window to the block diagram
- Double-click an item to open the owning palette



Create SubVI

- Enclose area to be converted into a subVI
- Select **Edit»Create SubVI** from the Edit menu



LabVIEW Functions and SubVIs Operate Like Functions in Other Languages

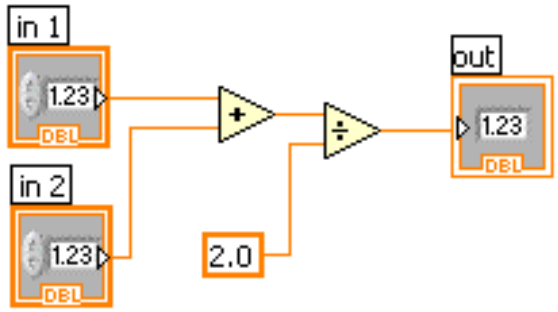
Function Pseudo Code

```
function average (in1, in2, out)
{
out = (in1 + in2)/2.0;
}
```

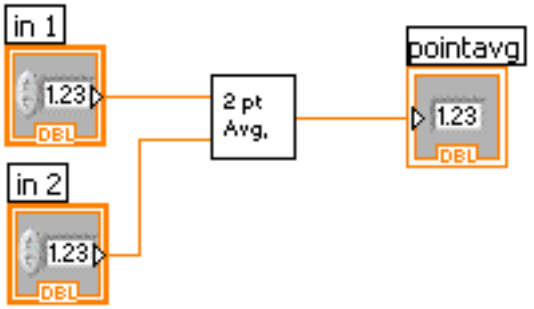
Calling Program Pseudo Code

```
main
{
average (in1, in2, pointavg)
}
```

SubVI Block Diagram



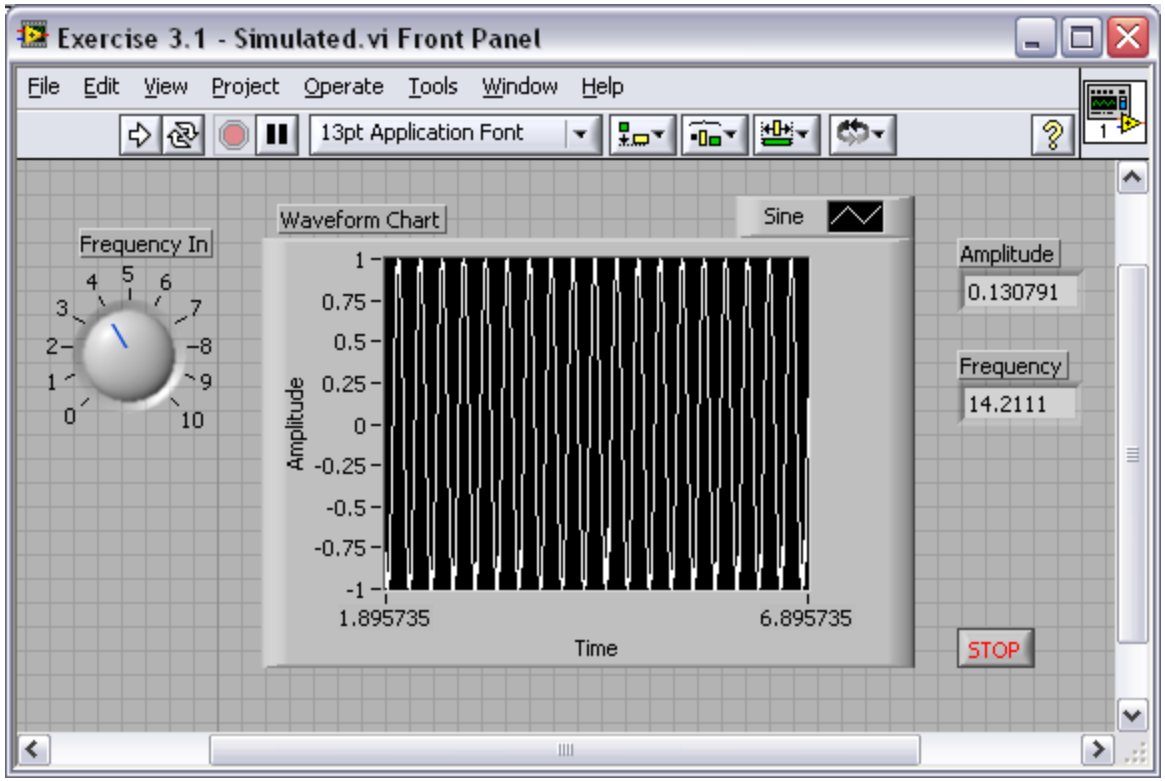
Calling VI Block Diagram



Exercise 3.1 – Analysis

- Use LabVIEW Express VIs to:
  - Simulate a signal and display its amplitude and frequency

This exercise should take 15 minutes.

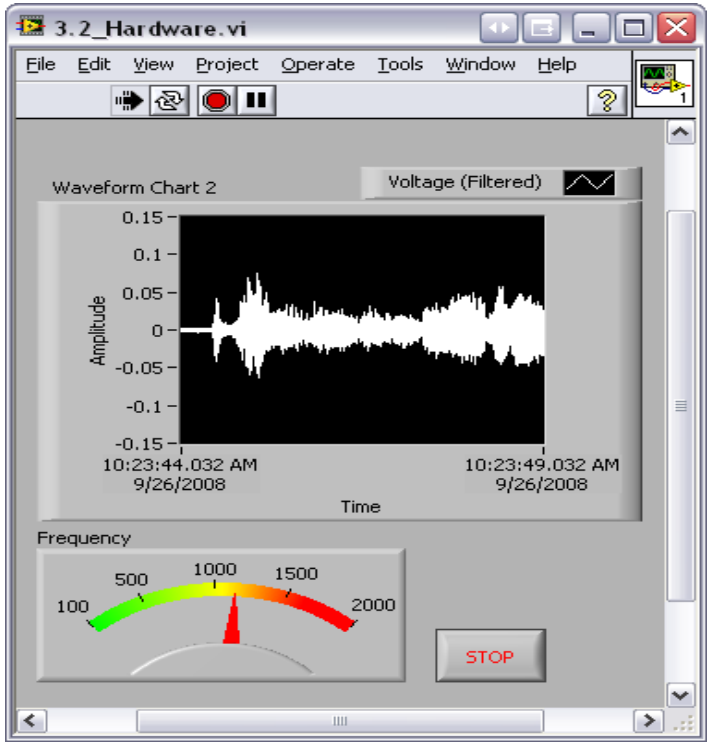


Track A,B,&C

Exercise 3.2 – Analysis

- Use LabVIEW Express VIs to:
  - Acquire a signal and display its amplitude and frequency

This exercise should take 15 minutes.

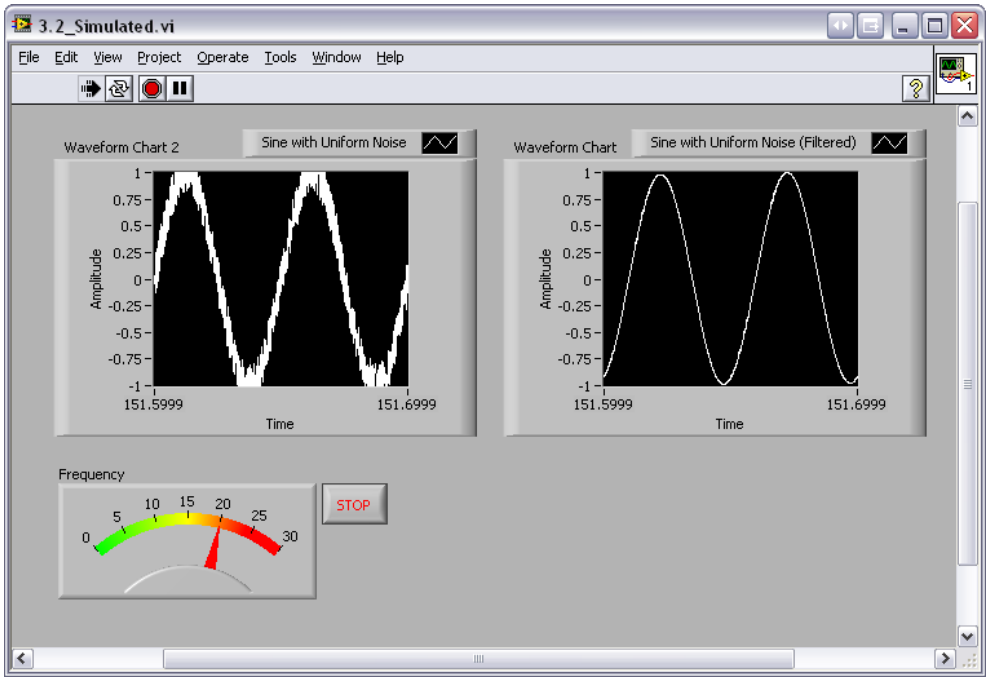


Track A

Exercise 3.2 – Analysis

- Use LabVIEW Express VIs to:
  - Acquire a signal and display its amplitude and frequency

This exercise should take 15 minutes.

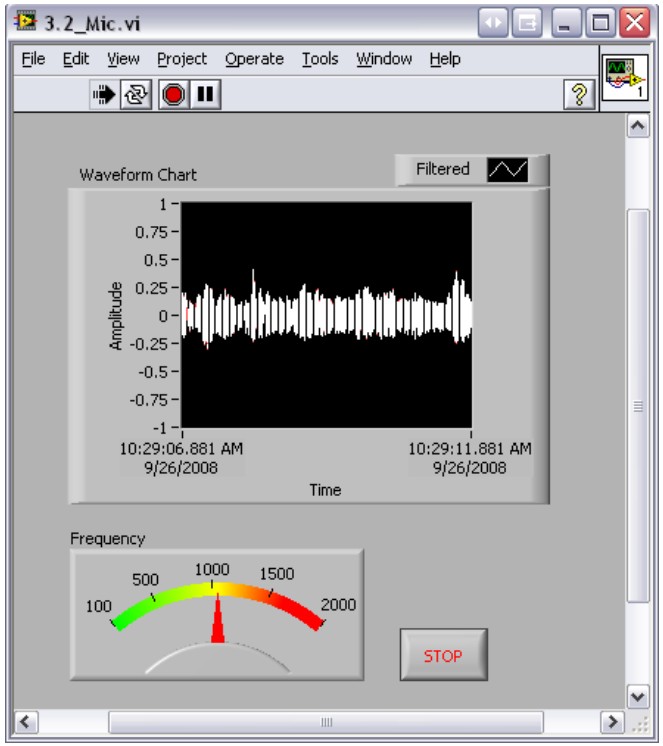


Track B

Exercise 3.2 – Analysis

- Use LabVIEW Express VIs to:
  - Acquire a signal and display its amplitude and frequency

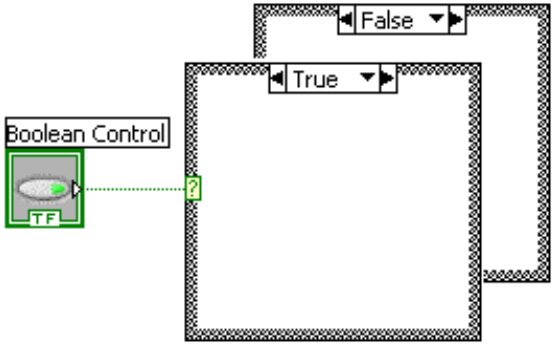
This exercise should take 15 minutes.



Track C

How Do I Make Decisions in LabVIEW?

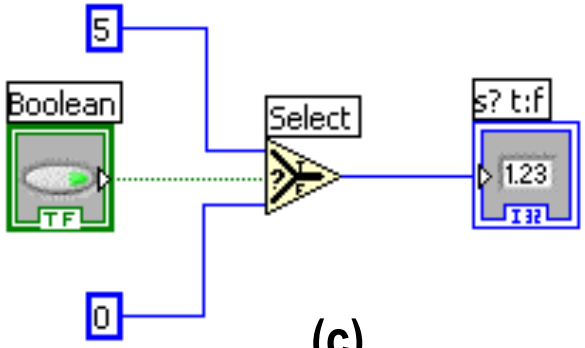
1. Case Structures



(a)

(b)

2. Select

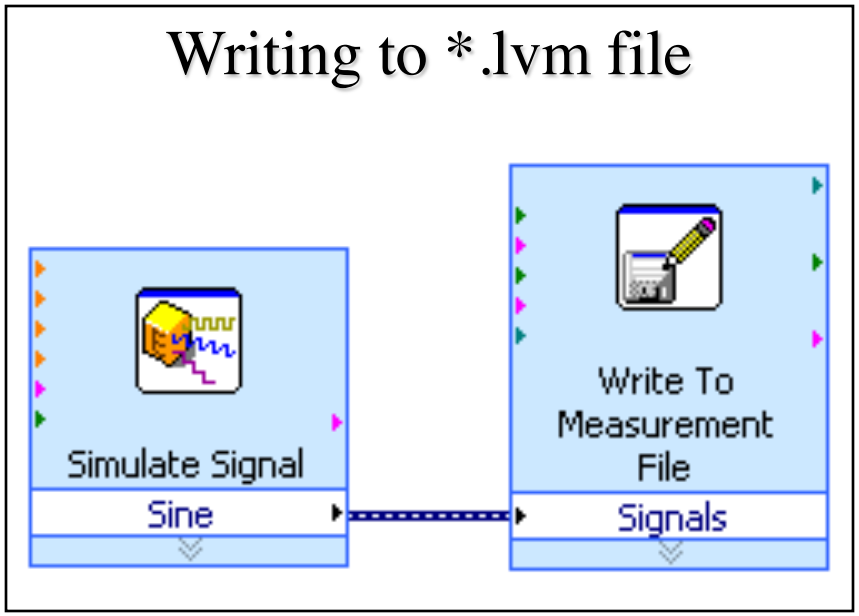


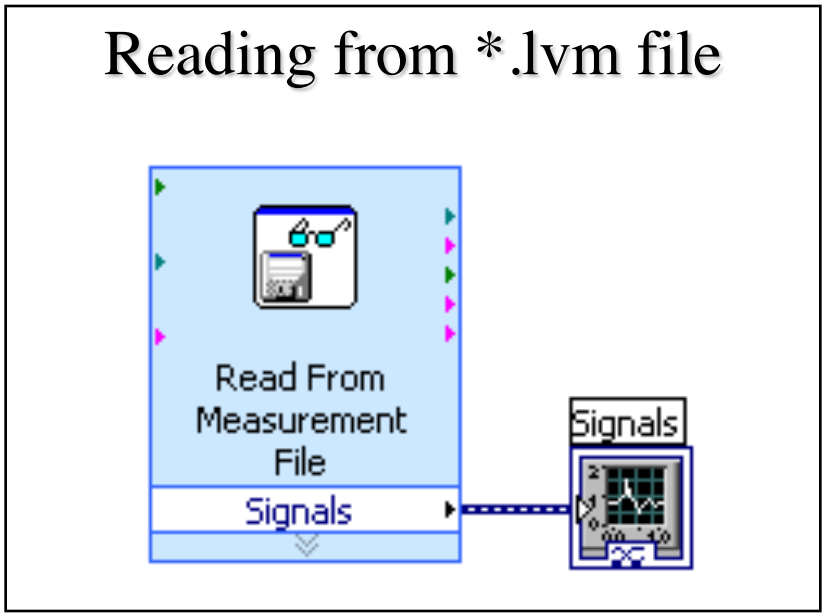
(c)

File I/O

File I/O – passing data to and from files

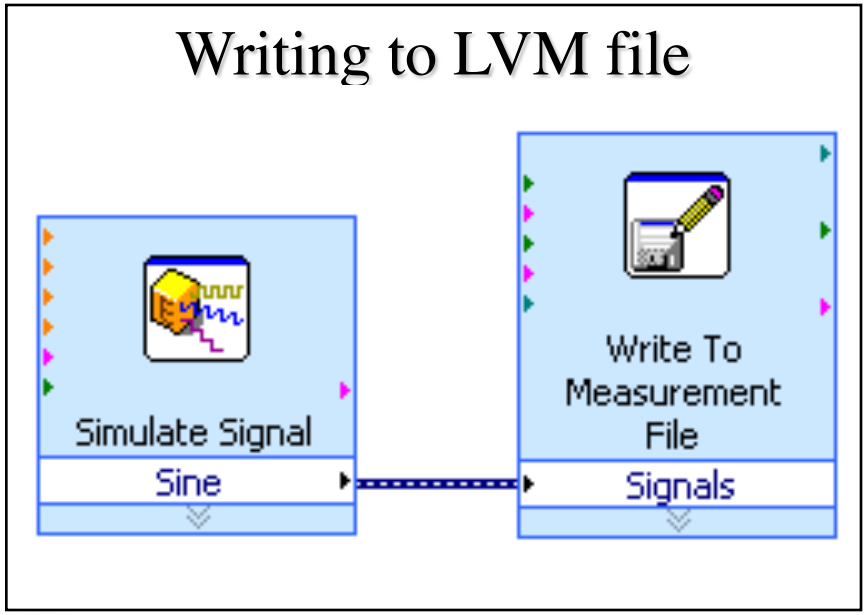
- Files can be binary, text, or spreadsheet
- Write/Read LabVIEW Measurements file (*.lvm)

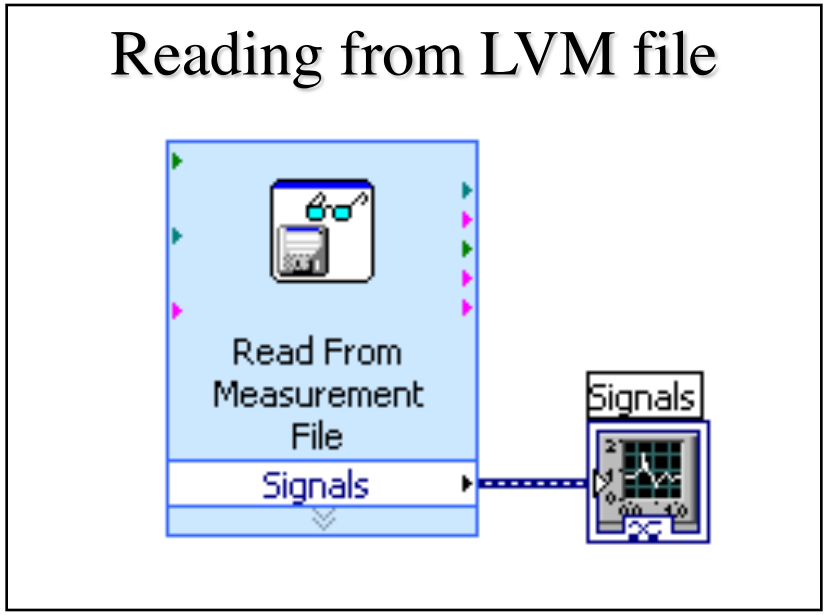




High-Level File I/O Functions

- Easy to use
- High level of abstraction

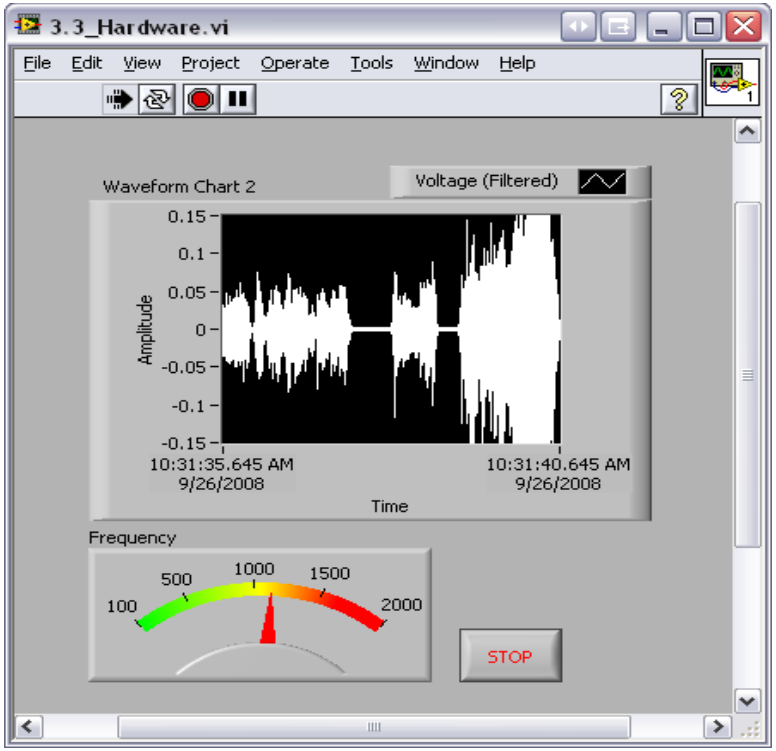




Exercise 3.3 – Decision Making and Saving Data

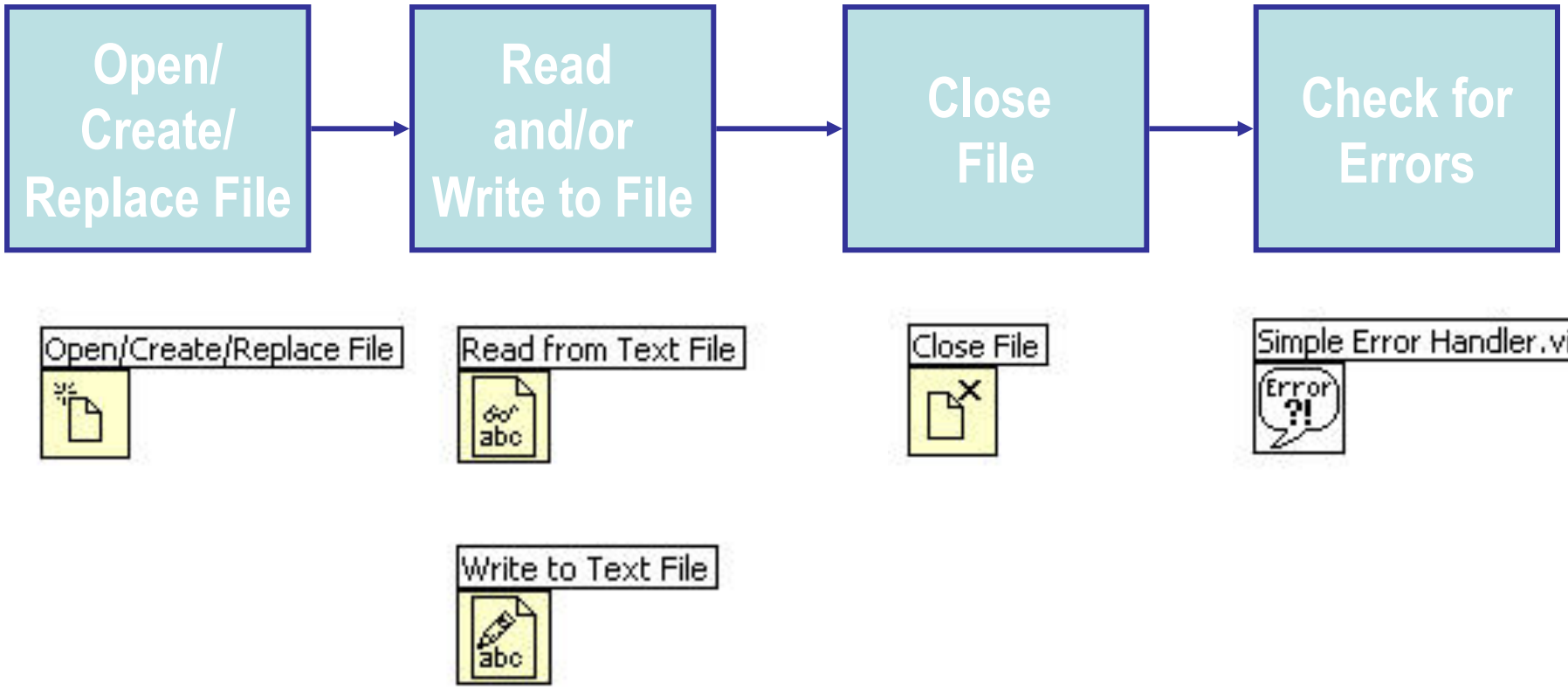
- Use a case structure to:
  - Make a VI that saves data when a condition is met

This exercise should take 15 minutes.



Track A,B,C

File I/O Programming Model – Under the Hood



Section III–Presenting Your Results

A. Displaying Data on the Front Panel

- Controls and Indicators
- Graphs and Charts
- Loop Timing

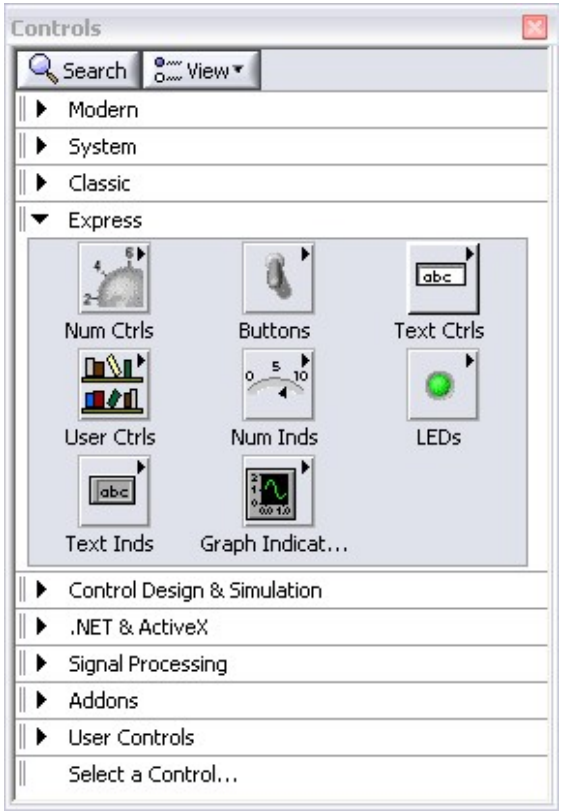
B. Signal Processing

- LabVIEW MathScript
- Arrays
- Clusters
- Waveforms

What Types of Controls and Indicators Are Available?

- **Numeric Data**
  - Number Input and Display
  - Analog Sliders, Dials, and Gauges
- **Boolean Data**
  - Buttons and LEDs
- **Array and Matrix Data**
  - Numeric Display
  - Chart
  - Graph
  - XY Graph
  - Intensity Graph
  - 3D Graph: Point, Surface, and Model
- **Decorations**
  - Tab Control
  - Arrows
- **Other**
  - Strings and Text Boxes
  - Picture/Image Display
  - ActiveX Controls

Express Controls Palette

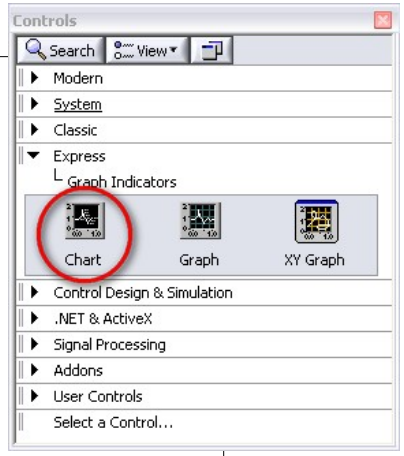


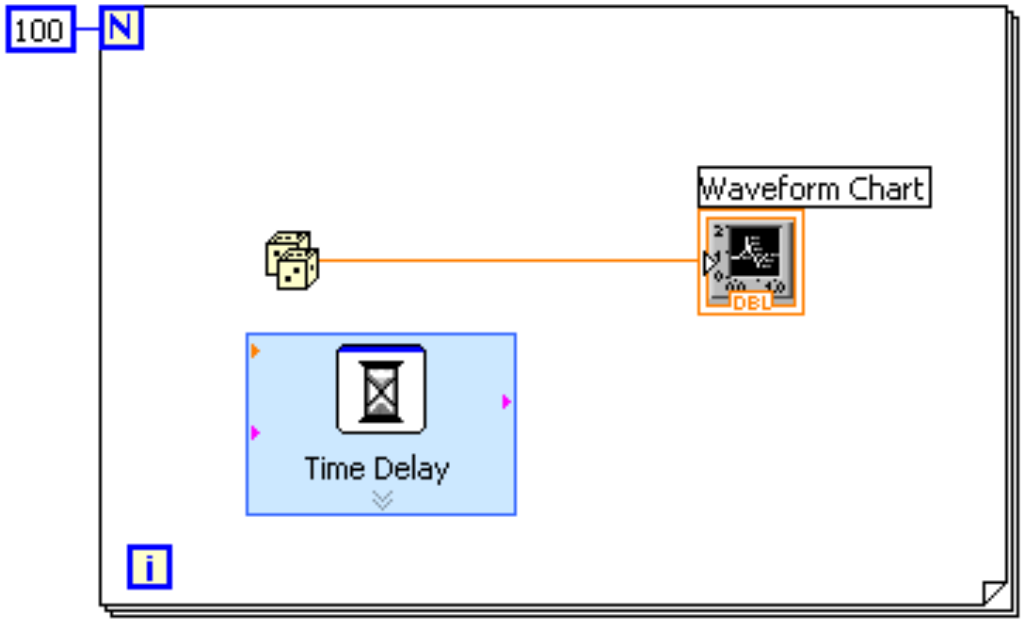
Charts – Add 1 Data Point at a Time with History

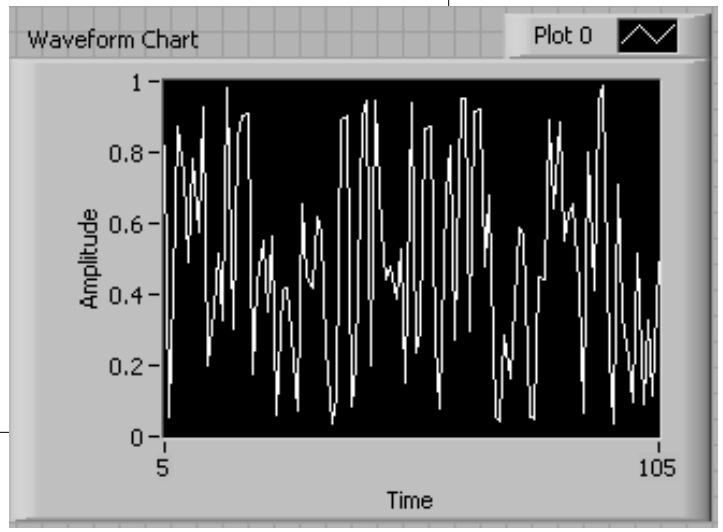
Waveform chart – special numeric indicator that can display a history of values

- Chart updates with each individual point it receives

Controls»Express»Graph Indicators»Chart



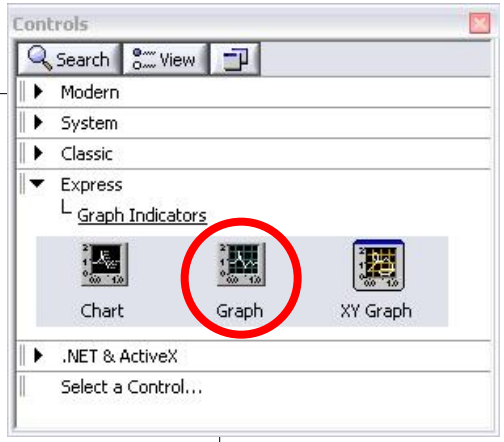




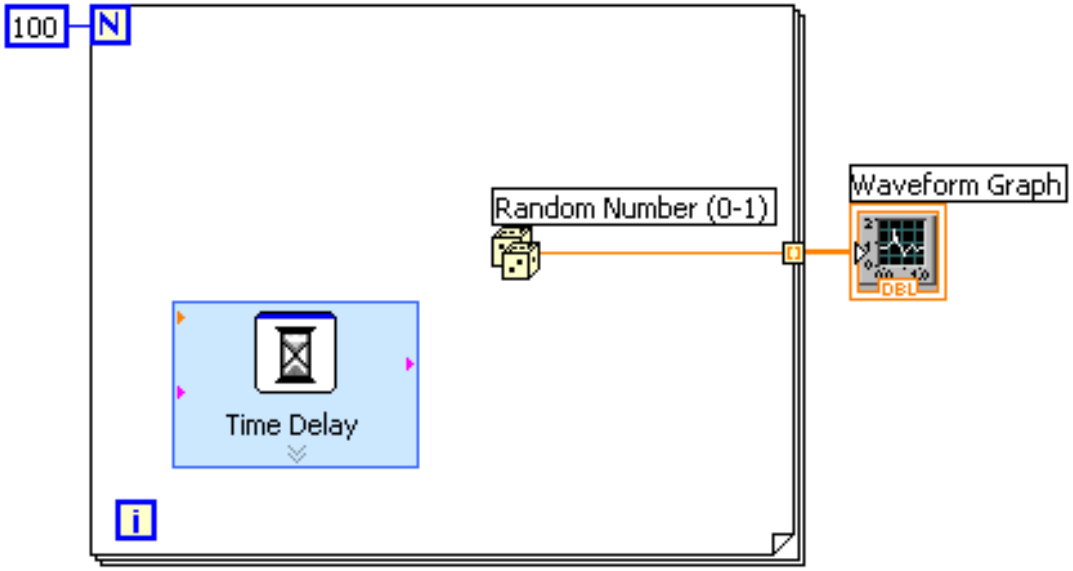
Graphs – Display Many Data Points at Once

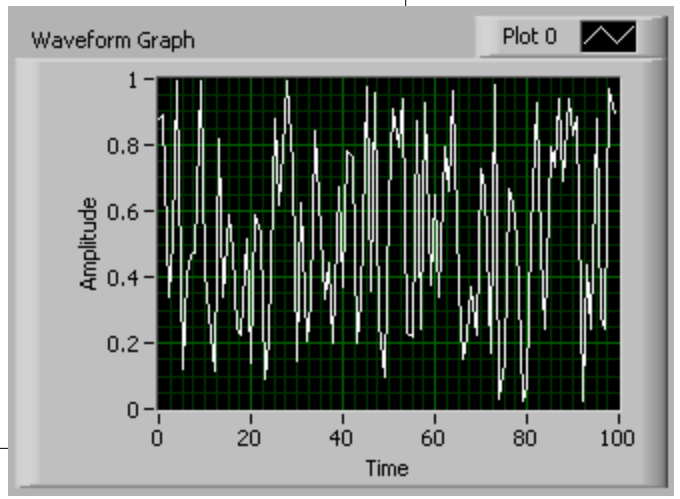
Waveform graph – special numeric indicator that displays an array of data

- Graph updates after all points have been collected
- May be used in a loop if VI collects buffers of data



Controls»Express»Graph Indicators»Graph

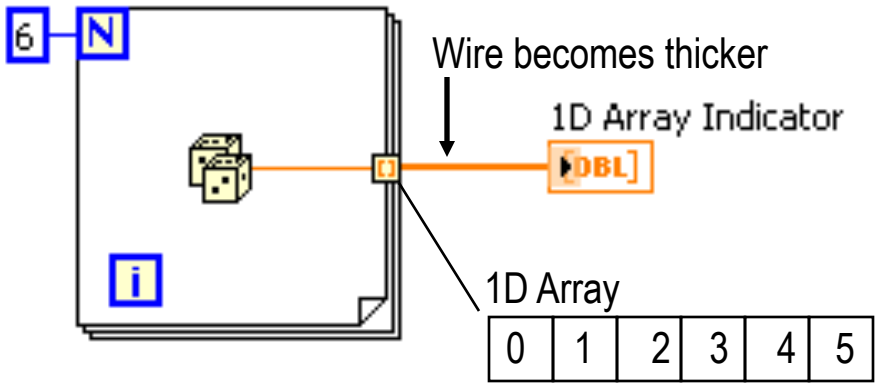




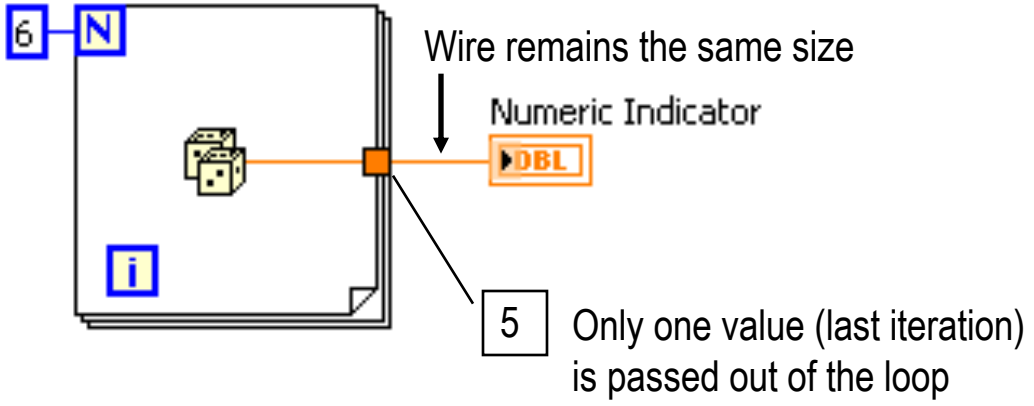
Building Arrays with Loops (Auto-Indexing)

- Loops can accumulate arrays at their boundaries with auto-indexing
- For loops auto-index by default
- While loops output only the final value by default
- Right-click tunnel and enable/disable auto-indexing

Auto-Indexing Enabled

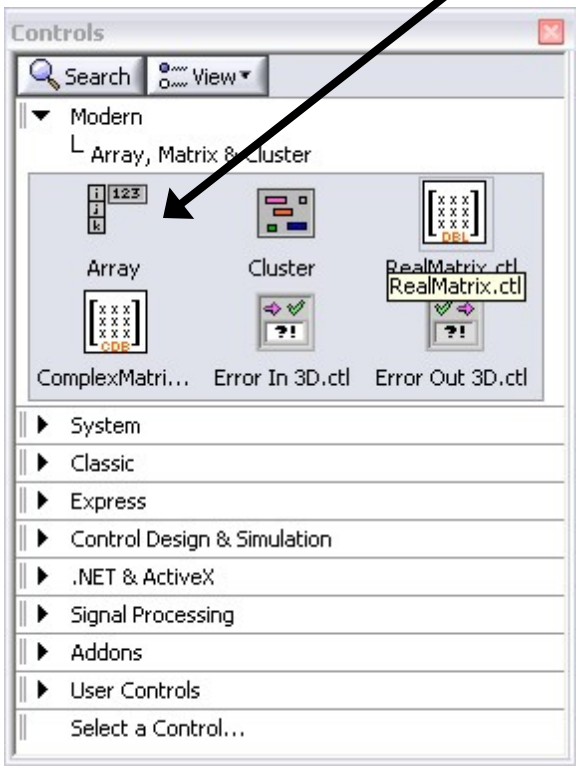


Auto-Indexing Disabled

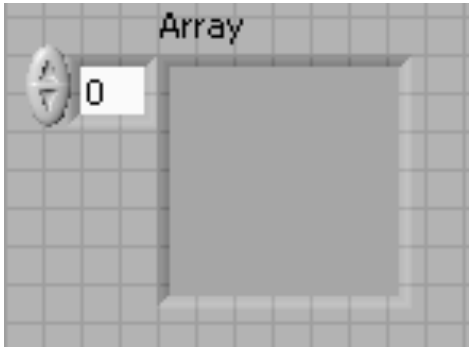


Creating an Array (Step 1 of 2)

From the **Controls»Modern»Array, Matrix, and Cluster** subpalette, select the **Array** icon.

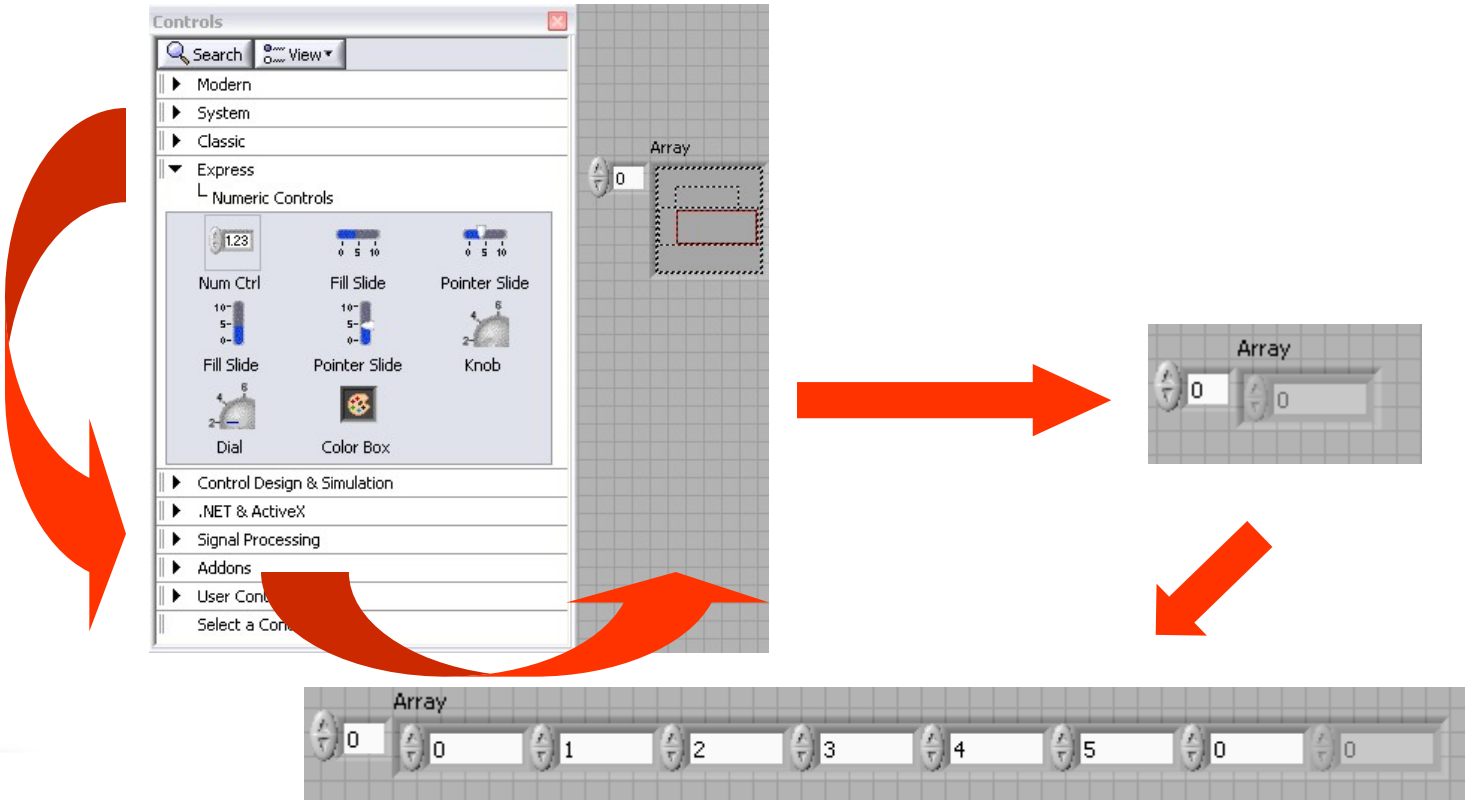


Drop it on the front panel.



Create an Array (Step 2 of 2)

1. Place an array shell.
2. Insert data type into the shell (i.e. numeric control).



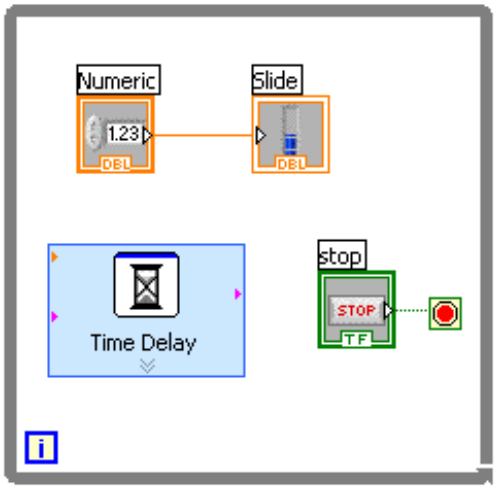
How Do I Time a Loop?

1. Loop Time Delay

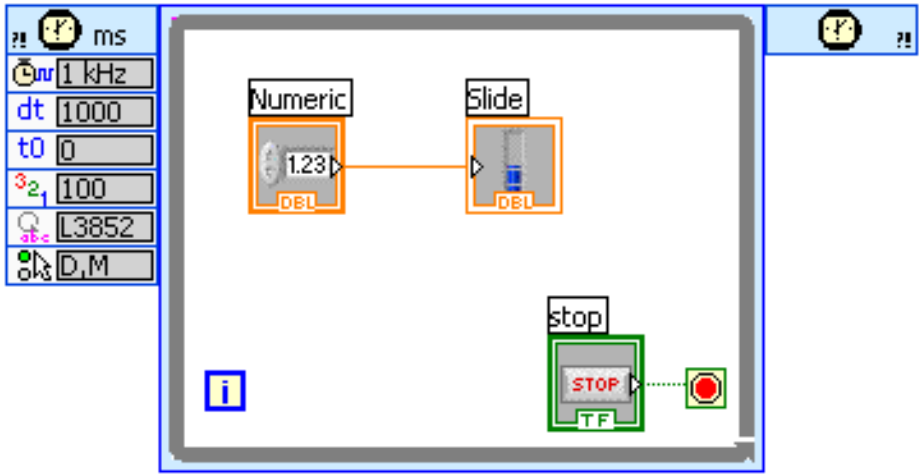
- Configure the Time Delay Express VI for seconds to wait each iteration of the loop (works on for and while loops).

2. Timed Loops

- Configure special timed while loop for desired *dt*.



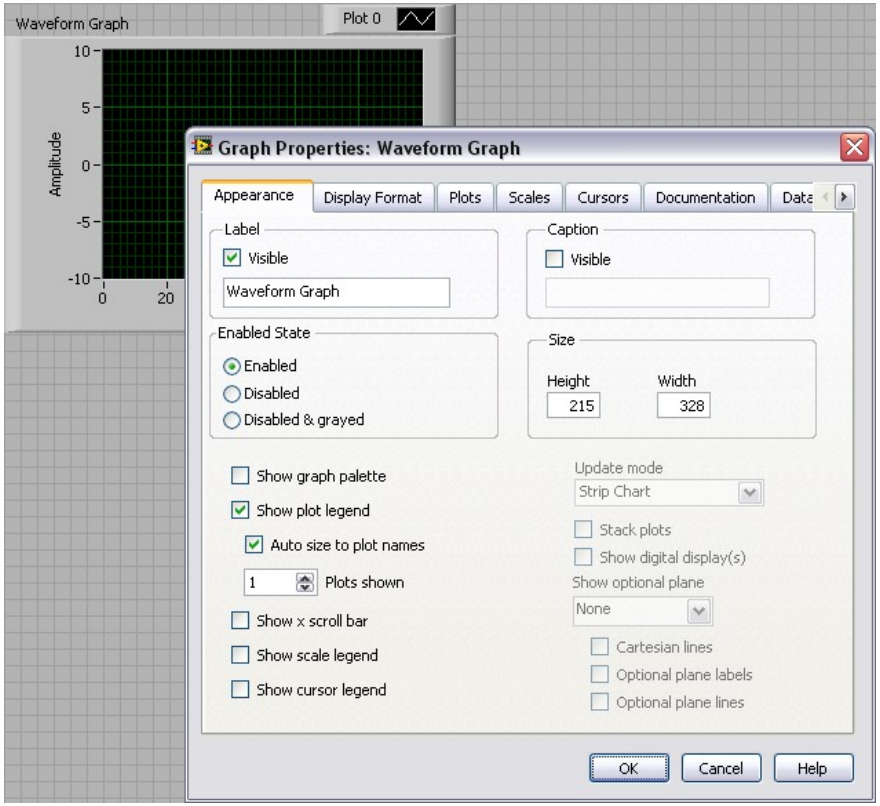
Time Delay



Timed Loop

Control and Indicator Properties

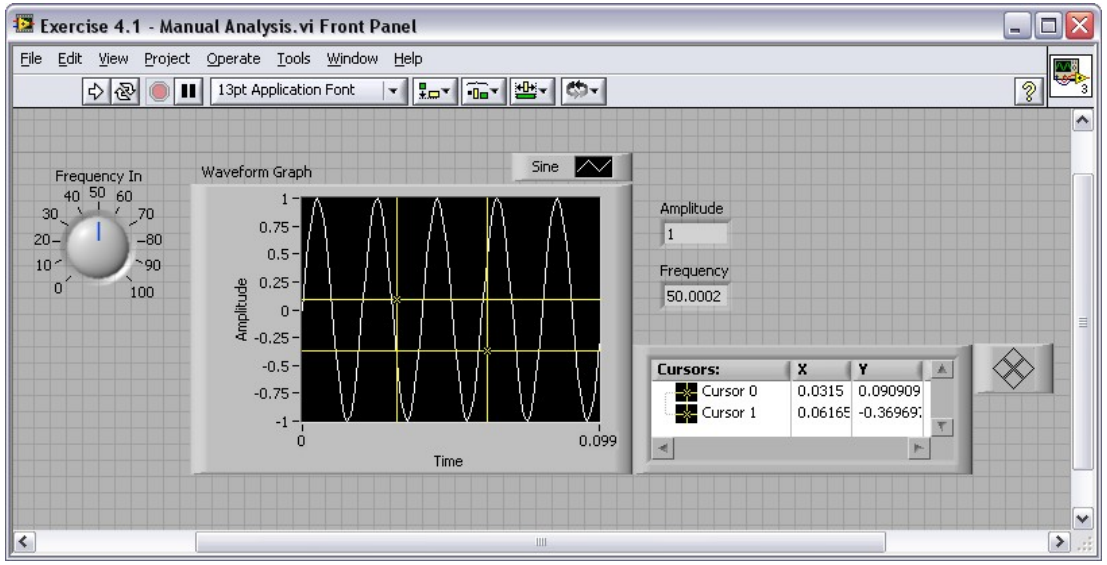
- Properties are characteristics or qualities about an object
- Properties can be found by right-clicking on a control or indicator
  - Properties include:
    - Size
    - Color
    - Plot style
    - Plot color
  - Features include:
    - Cursors
    - Scaling



Exercise 4.1 – Manual Analysis

- Use the cursor legend on a graph to:
  - Verify your frequency and amplitude measurements

This exercise should take 15 minutes.



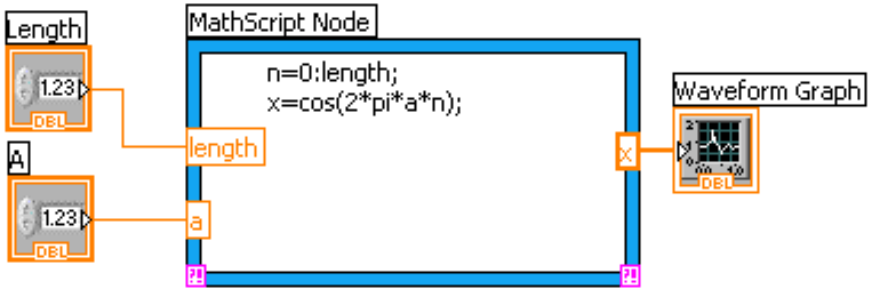
Track A,B,&C

Textual Math in LabVIEW

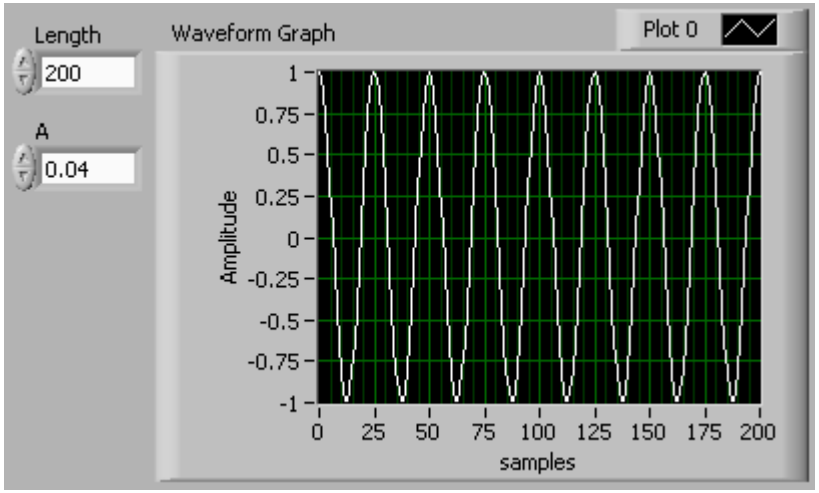
- Integrate existing scripts with LabVIEW for faster development
- Use Interactive, easy-to-use, hands-on learning environment
- Develop algorithms, explore mathematical concepts, and analyze results using a single environment
- Freedom to Choose the most effective syntax, whether graphical or textual within one VI

Math with the LabVIEW MathScript Node

- Implement equations and algorithms textually
- Input and output variables created at the border
- Generally compatible with popular .m file script language
- Terminate statements with a semicolon to disable immediate output



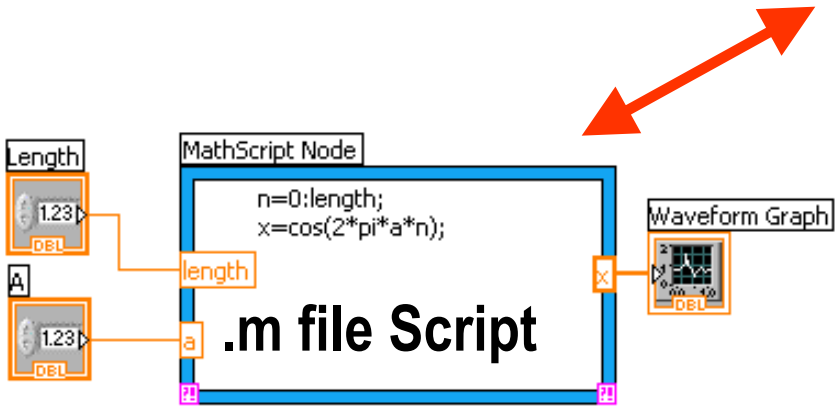
(Functions»Programming» Structures»MathScript)

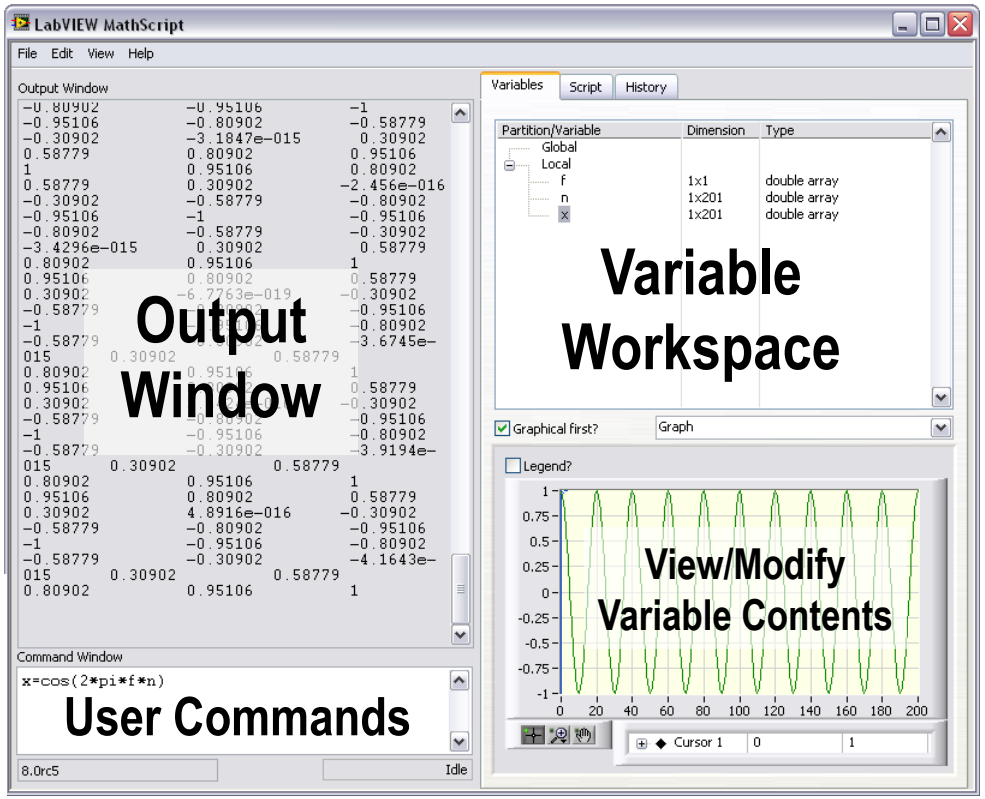


Prototype your equations in the interactive LabVIEW **MathScript Window**.

The Interactive LabVIEW MathScript Window

- Rapidly develop and test algorithms
- Share scripts and variables with the node
- View/modify variable content in 1D, 2D, and 3D



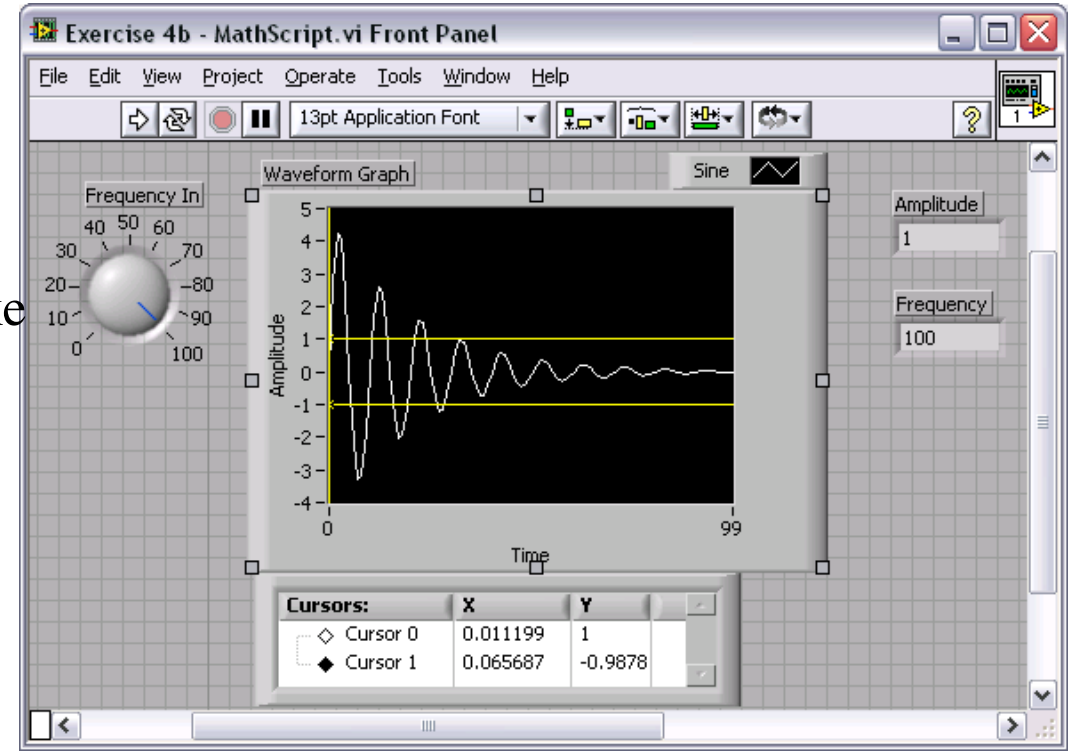


(LabVIEW»Tools»MathScript Window)

Exercise 3.2 – Using MathScript

Use the MathScript Node and Interactive Window to process the acquired signal (logarithmic decay) in the MathScript and save the script.

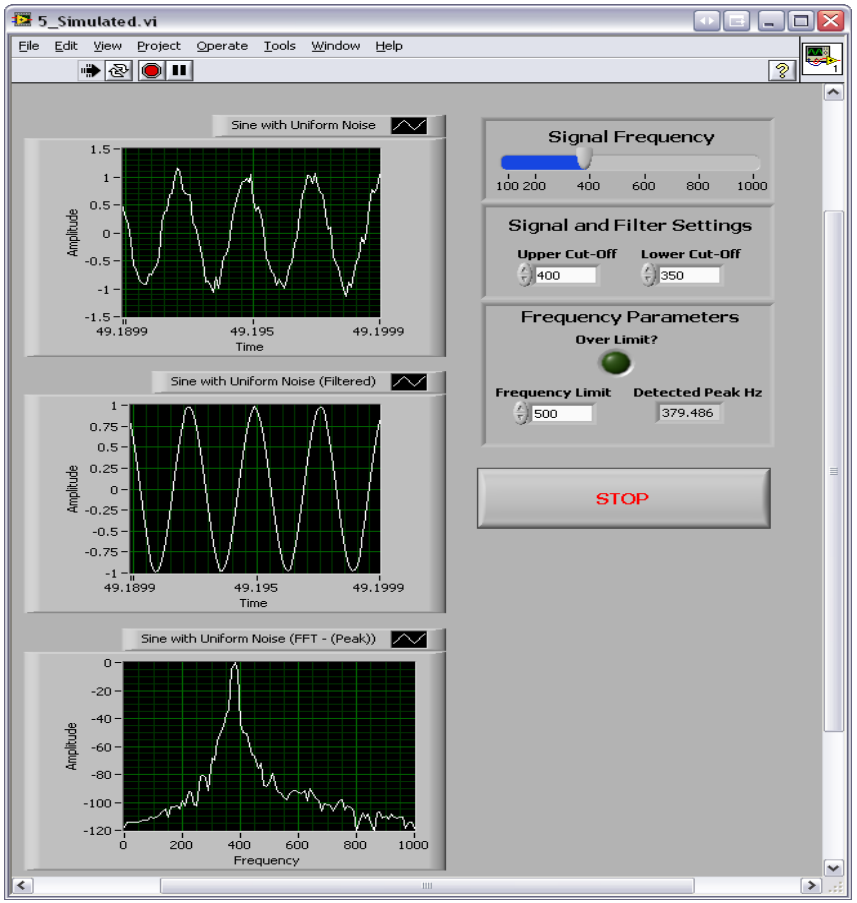
This exercise should take 25 minutes.



Track A,B,&C

Exercise 5 – Apply What You Have Learned

This exercise should take 20 minutes.



Track A,B,&C

Section IV – Advanced Data flow Topics (Optional)

A. Additional Data Types

- Cluster

B. Data flow Constructs

- Shift Register
- Local Variables

C. Large Application Development

- Navigator Window
- LabVIEW Projects

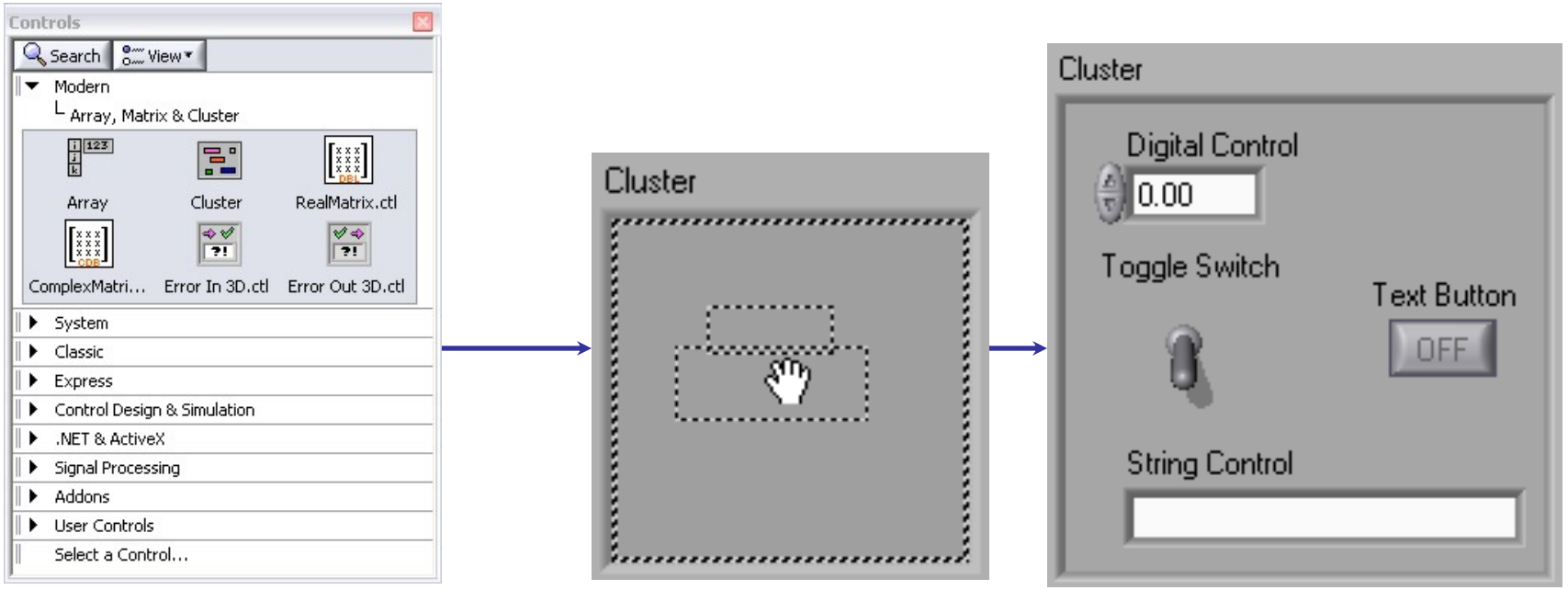
Introduction to Clusters

- Data structure that groups data together
- Data may be of different types
- Analogous to *struct* in ANSI C
- Elements must be either all controls or all indicators
- Thought of as wires bundled into a cable
- **Order is important**

Creating a Cluster

1. Select a **Cluster** shell.
2. Place objects inside the shell.

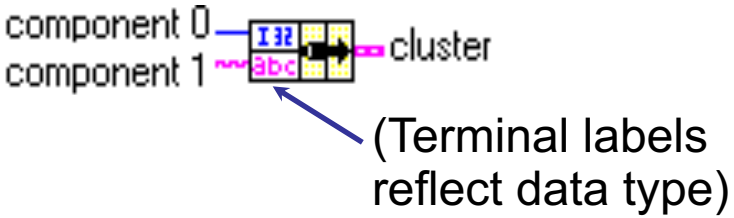
Controls»Modern»Array, Matrix & Cluster



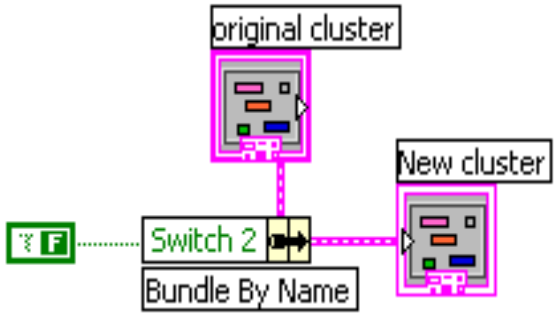
Cluster Functions

- In the **Cluster & Variant** subpalette of the **Programming** palette
- Can also be accessed by right-clicking the cluster terminal

Bundle



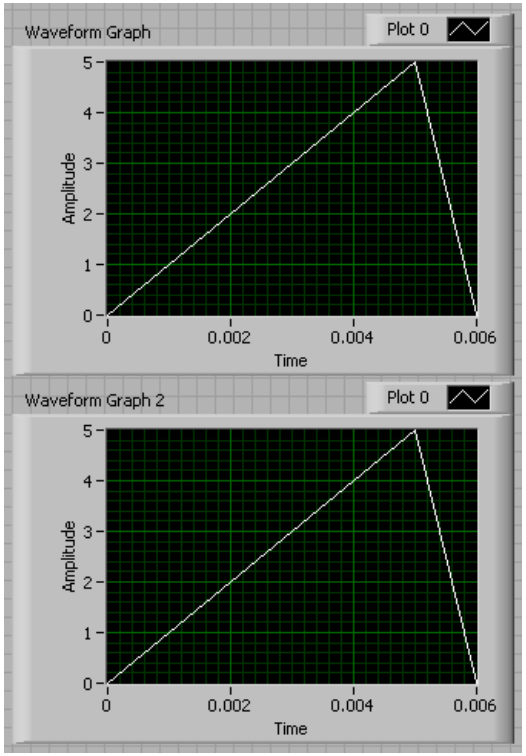
Bundle By Name



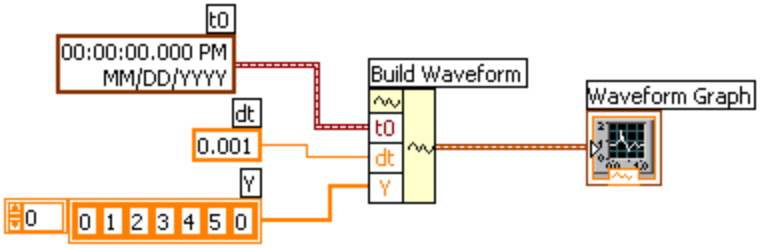
Using Arrays and Clusters with Graphs

The waveform data type contains 3 pieces of data:

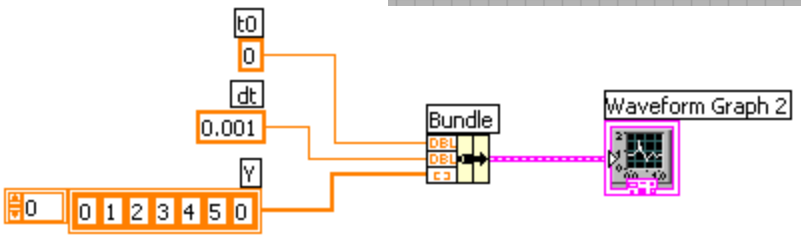
- t0 = Start time
- dt = Time between samples
- Y = Array of Y magnitudes



You can create a waveform cluster in two ways:



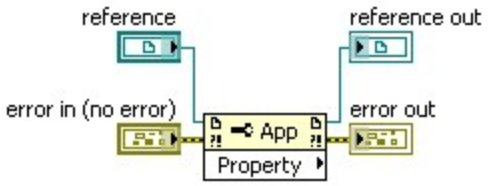
Build Waveform (absolute time)



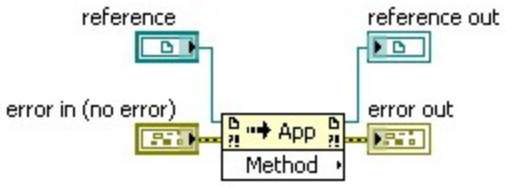
Cluster (relative time)

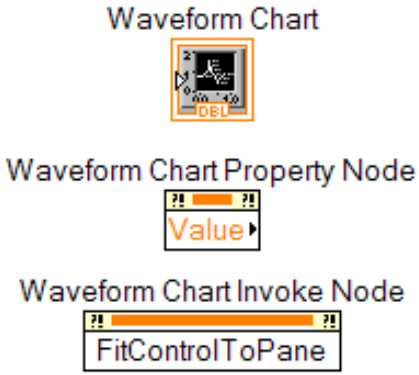
Property and Invoke Nodes

- Create Property and Invoke nodes for a VI element by right-clicking the element and selecting **Create»Property Node.**
- Use the Property Node to get and set object properties.



- Invoke object methods with the Invoke Node.





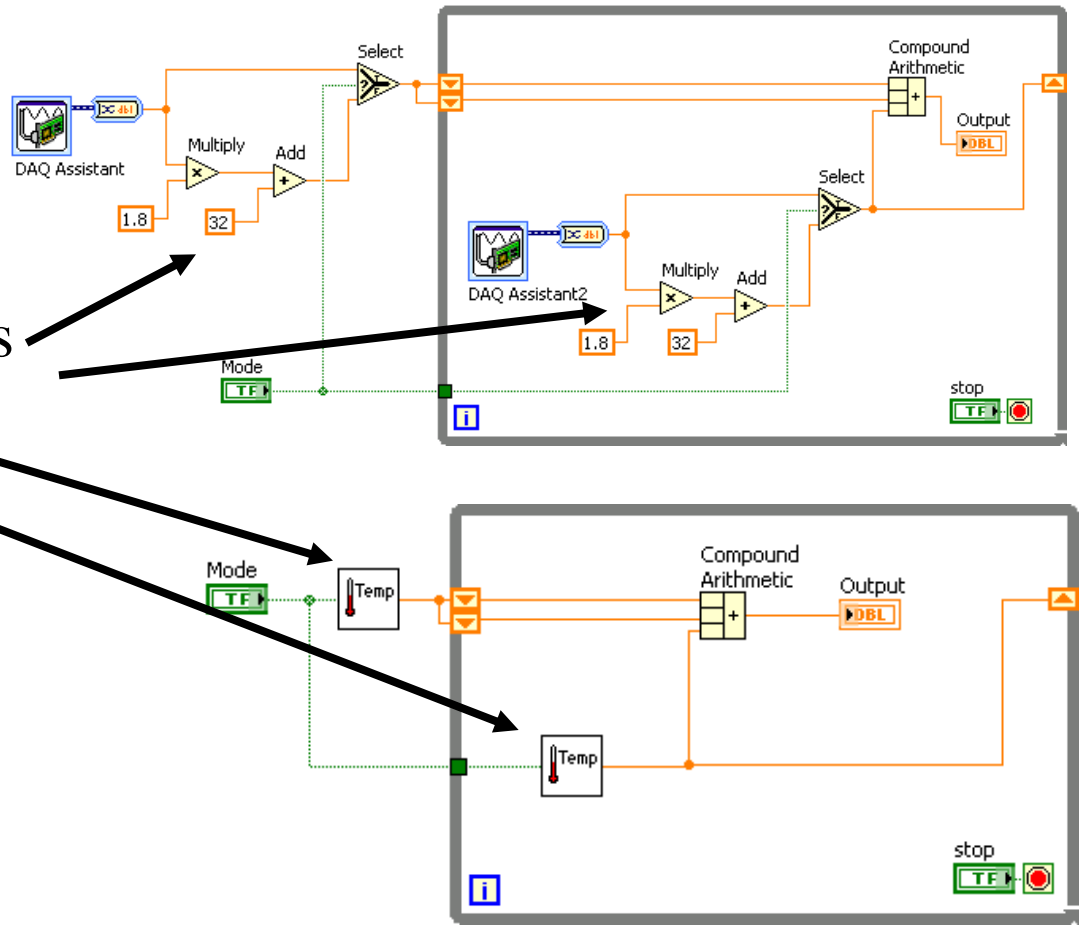
Do Not Delete
Exercise Instructions

Do Not Delete
Exercise Instructions

Do Not Delete
Exercise Instructions

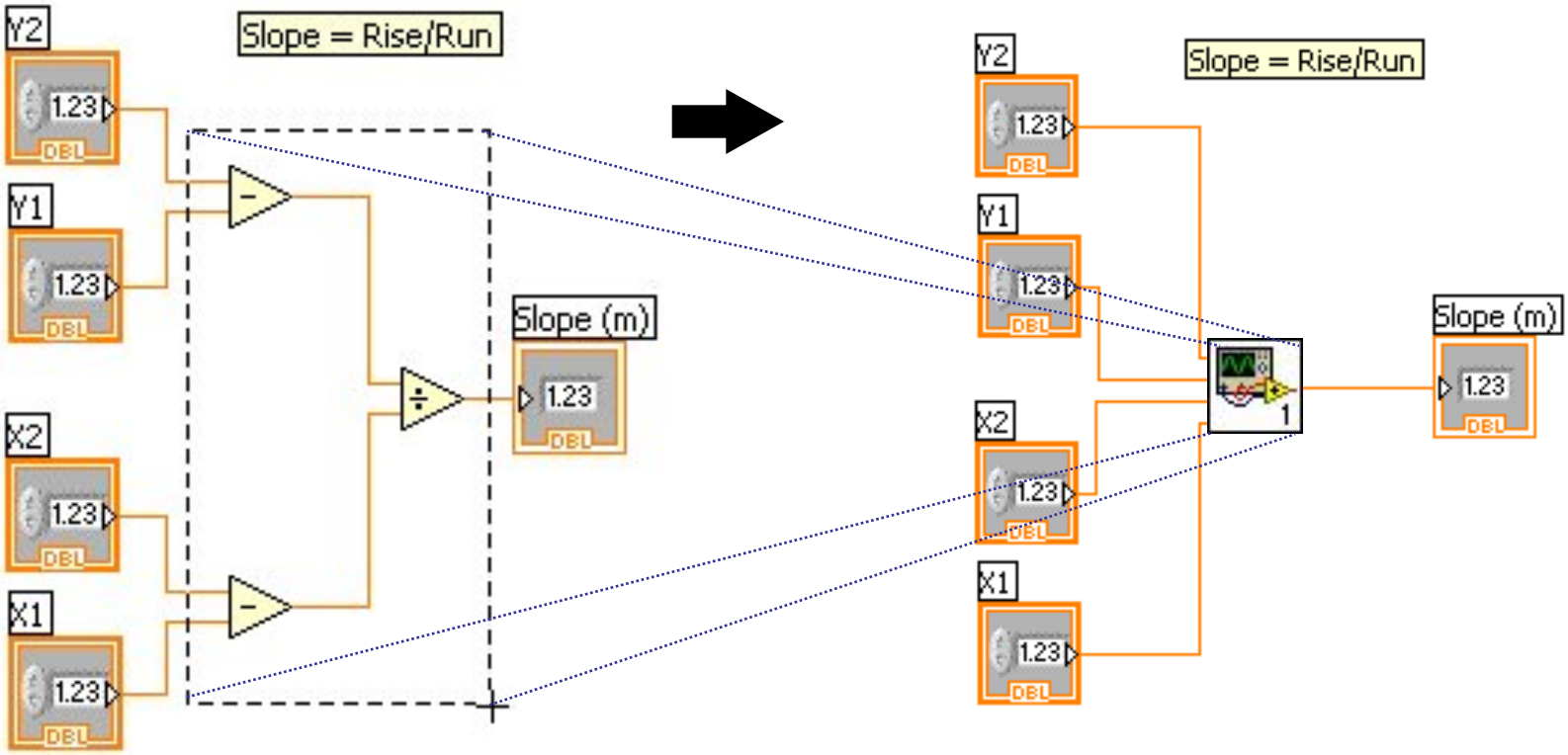
Modularity in LabVIEW – SubVIs

Convert repeated functions and VIs with a single VI



Create SubVI

- Enclose area to be converted into a subVI.
- Select **Edit»Create SubVI** from the Edit menu.



LabVIEW Functions and SubVIs operate like Functions in other languages

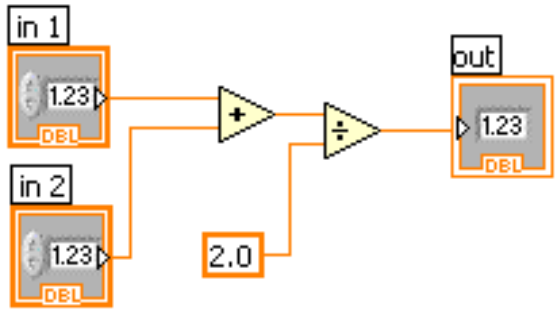
Function Pseudo Code

```
function average (in1, in2, out)
{
out = (in1 + in2)/2.0;
}
```

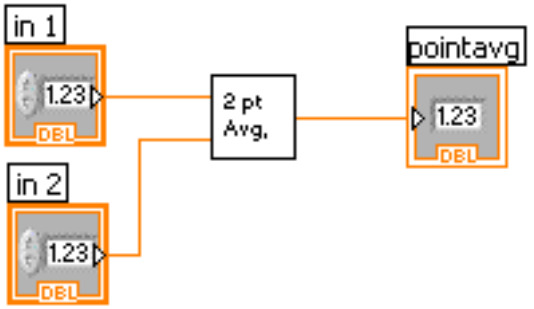
Calling Program Pseudo Code

```
main
{
average (in1, in2, pointavg)
}
```

SubVI Block Diagram

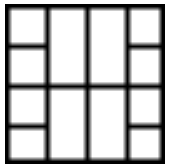


Calling VI Block Diagram

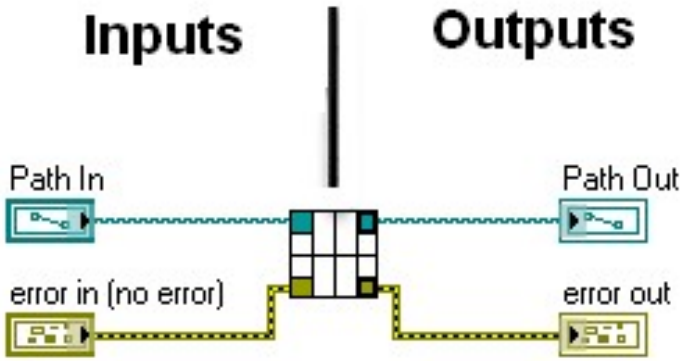


Connector Pane and Icon Viewer

- Use this connector pane layout as a standard



- Top terminals are usually reserved for file paths and references, such as a file reference
- Bottom terminals are usually reserved for error clusters



Icon Viewer – Create an Icon

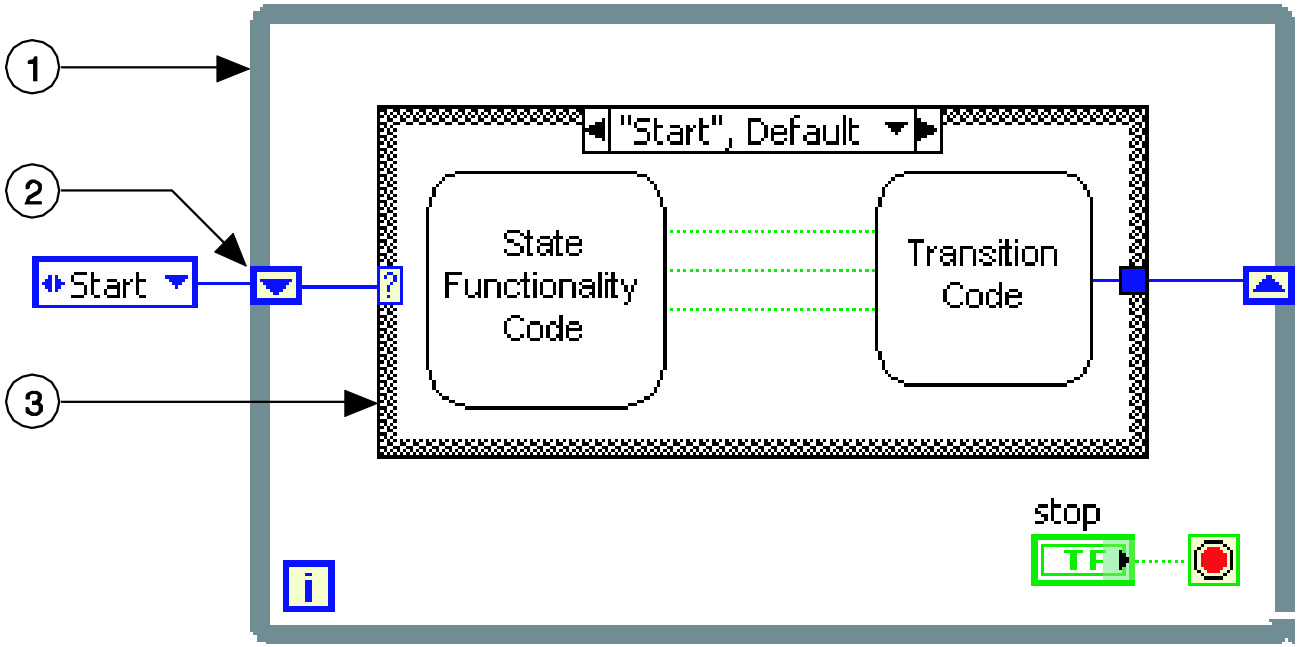
- Create custom icons by right-clicking the icon in the upper-right corner of the front panel or block diagram and selecting **Edit Icon** or by double-clicking the icon
- You also can drag a graphic from anywhere in your file system and drop it on the icon
- Refer to the Icon Art Glossary at ni.com for standard graphics to use in a VI icon

Do Not Delete
Exercise Instructions

Do Not Delete
Exercise Instructions

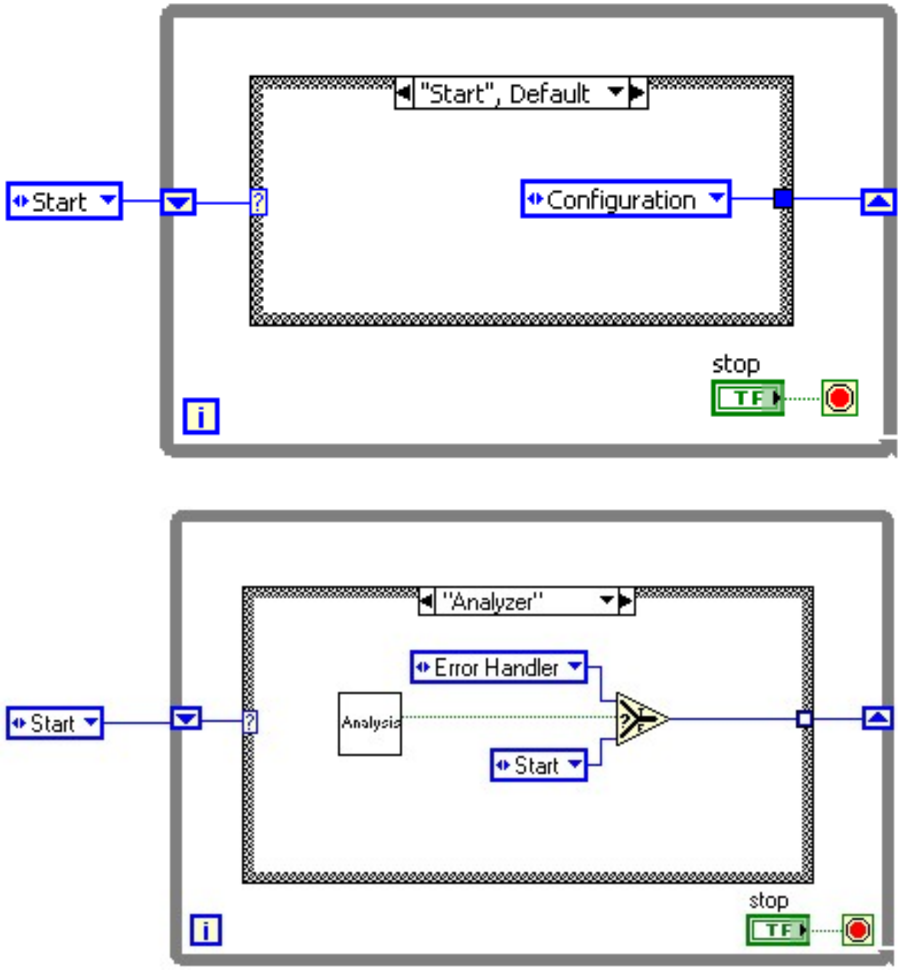
State Machines

- While Loop
- Case Structure
- Shift Register



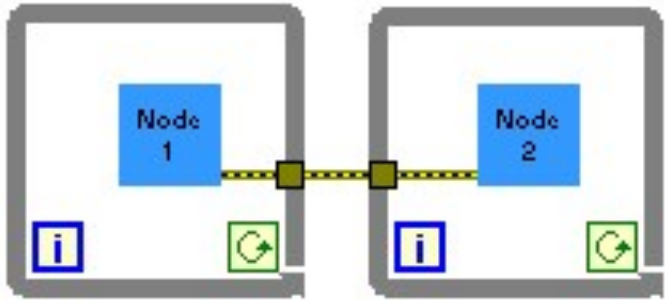
State Machines Transitions

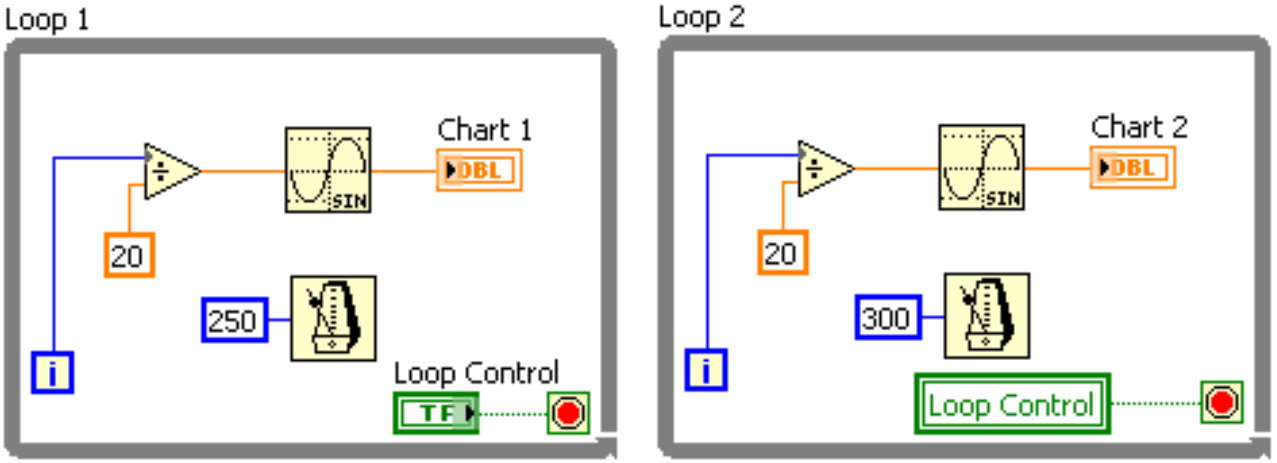
- Several programming techniques exist for transitioning from state to state in LabVIEW using state machines
- Default transition implies that after one state, another state always follows
- Transitions between two potential states can be handled by a select function



Communicating between Loops

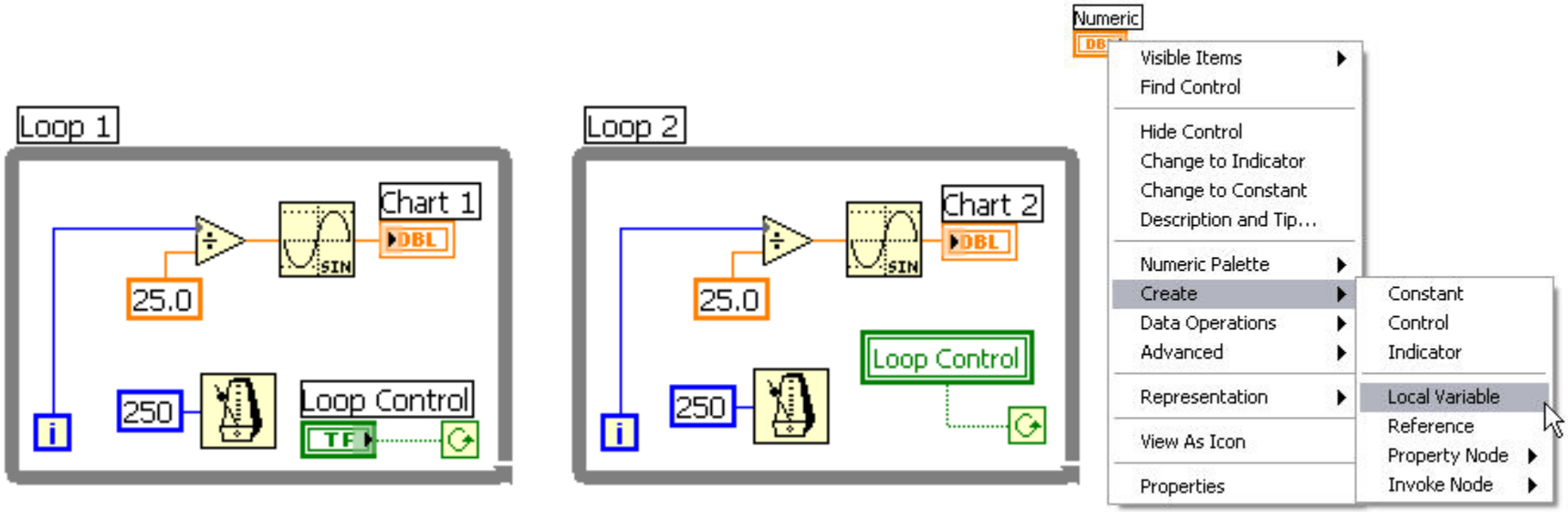
- Communicating between loops using data flow is not possible
- The left loop executes completely before the right loop
- Variables are needed when communication with wires does not give the desired behavior





Local Variables

- Local Variables allow data to be passed between parallel loops.
- A single control or indicator can be read or written to from more than one location in the program
  - Local Variables break the dataflow paradigm and should be used sparingly



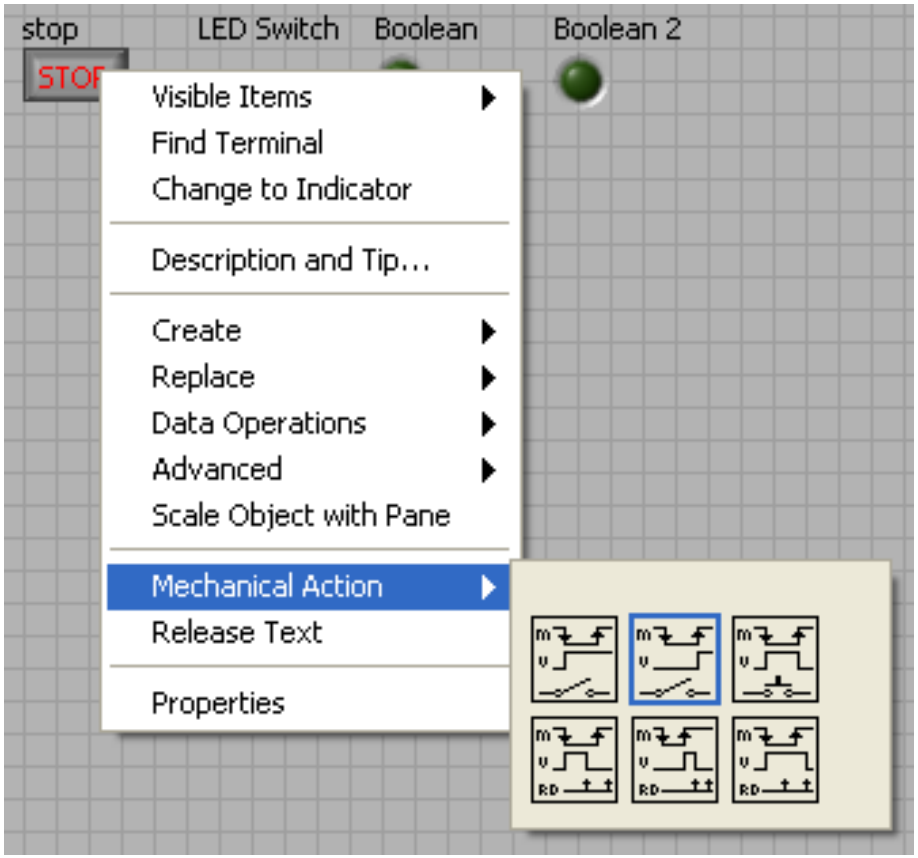
Creating a Local Variable - Walk Through

In this example we will see how Local Variables allow us to share data between parallel loops. We will create a program that toggles two separate LEDs using one switch. It also stops two loops with one button.

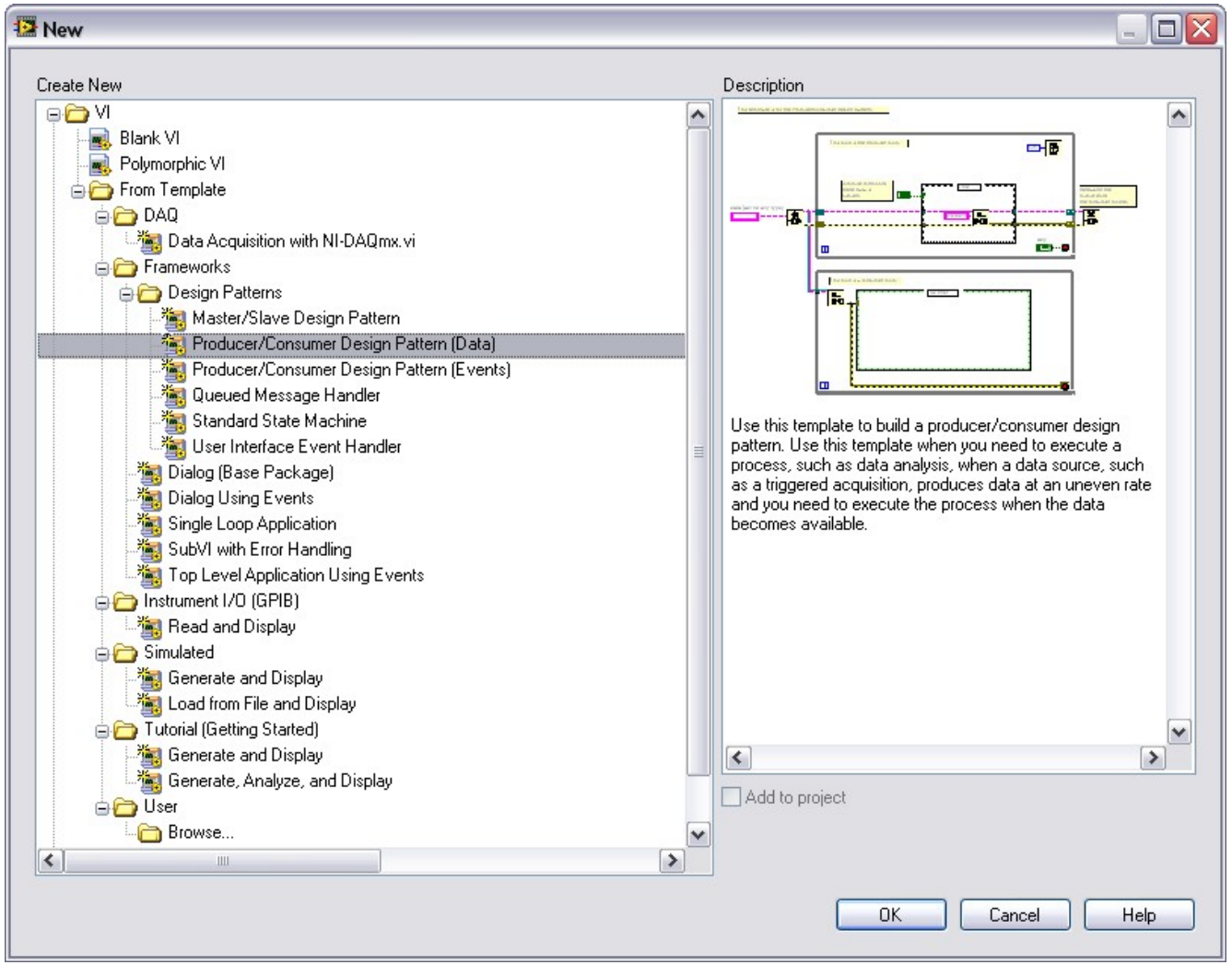
Create the block diagram and front panel show in the notes.

Creating Local Variables

Right-click on the Stop Button in the Front Panel and change the Mechanical Action to "Switch When Released" since Local Variables cannot store Latched data.



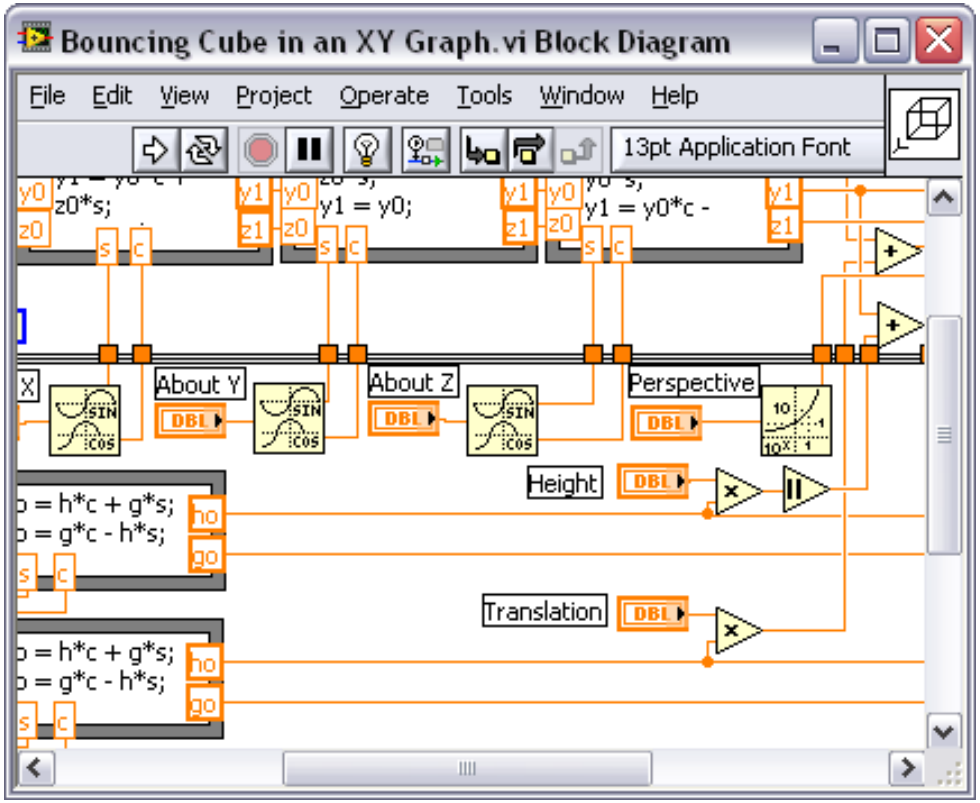
Producer/Consumer Design Pattern

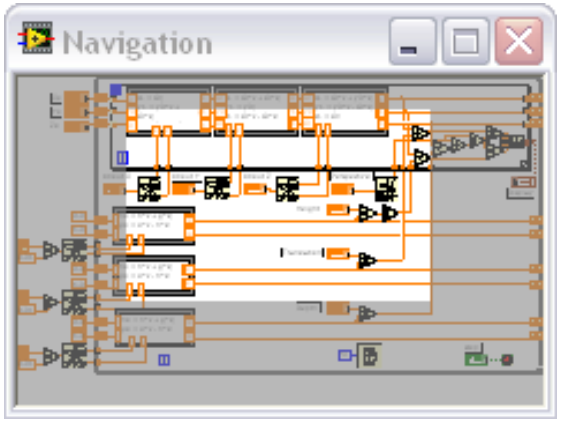


V. Large Program Development

A. Navigation Window
B. LabVIEW Project
C. Shared Variable

LabVIEW Navigation Window



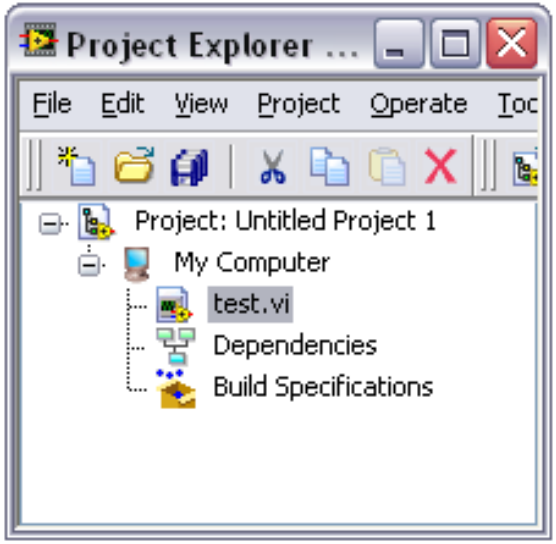


- Shows the current region of view compared to entire front panel or block diagram
- Works well for large programs

Organize and reduce program visual size with subVIs.

LabVIEW Project

- Group and organize VIs
- Manage hardware and I/O
- Manage large LabVIEW applications
- Manage VIs for multiple targets
- Build libraries and executables
- Enable version tracking and management



(LabVIEW»Project»New)

Shared Variables

- LabVIEW shared variables are used to send data between VIs
- Variable types:
  - Single process: Share the data among VIs on the local computer
  - Network-published: Communicate between VIs, remote computers, and hardware through the LabVIEW shared variable engine
- Shared variables must exist within a project library
- Shared variables must be deployed to be available to other projects and remote computers

Do Not Delete
Exercise Instructions

Do Not Delete
Exercise Instructions

Section VI – Instrument Control

A. Overview of Instrument Control
B. GPIB
C. Serial
D. Instrument I/O Assistant
E. VISA
F. Instrument Drivers and IDNet

Which Types of Instruments Can Be Controlled?

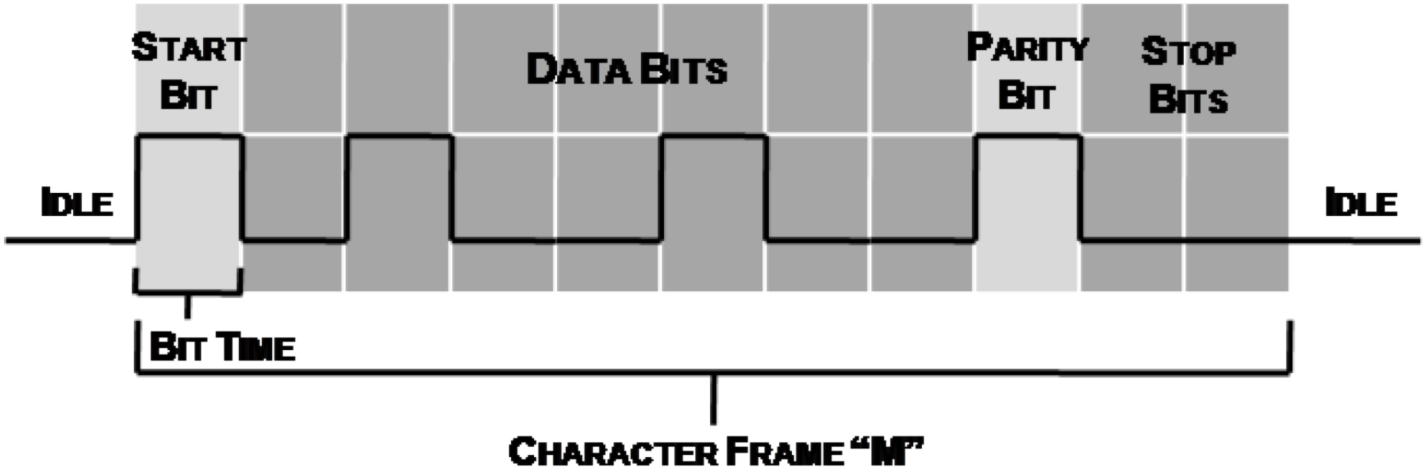
- GPIB
- Serial
- Modular Instruments
- Image Acquisition
- Motion Control
- USB
- Ethernet
- Parallel Port
- CAN

GPIB

- General Purpose Interface Bus (GPIB)
- Usually used in stand-alone benchtop instruments to control measurements and communicate data
- Features digital 8-bit parallel communication interface
- Defined by IEEE 488.1 and 488.2 standards

Serial

- Serial communication transmits one bit at a time over a transmission line
- Usually does not require external hardware
- Four parameters: baud rate, data bits, parity bit, stop bits



Instrument I/O Assistant

- LabVIEW Express VI used to communicate with message-based instruments
- Communicates with an instrument that uses a serial, Ethernet, or GPIB interface
- Use the Instrument I/O Assistant when an instrument driver is not available

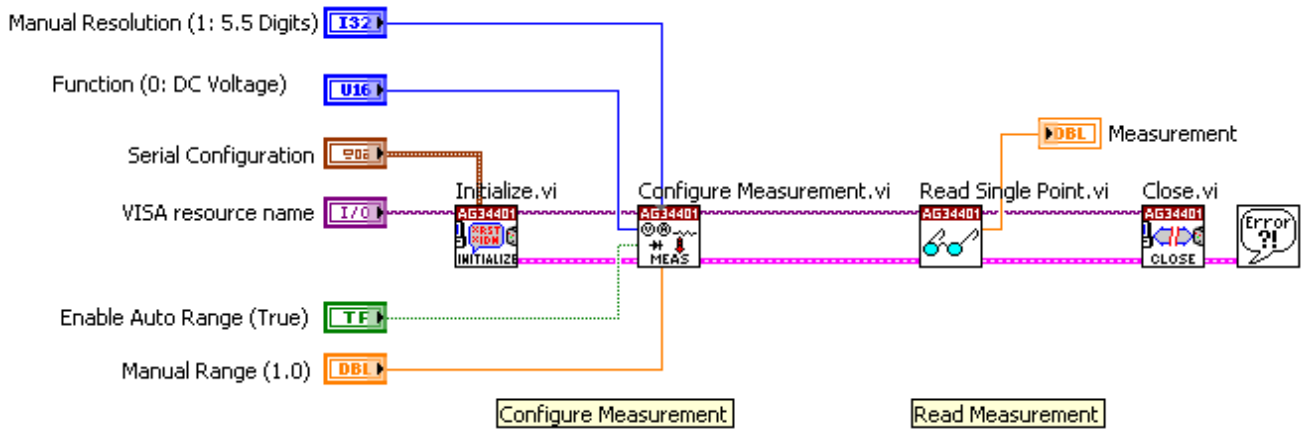
Do Not Delete
Exercise Instructions

VISA

- Virtual Instrument Software Architecture (VISA)
- High-level API that calls low-level drivers
- Can control VXI, GPIB, serial, or computer-based instruments
- Makes appropriate driver calls depending on the instrument used

Instrument Drivers

- LabVIEW Plug and Play drivers are a set of VIs that control a programmable instrument
- VIs correspond to instrument operation: configuring, triggering, and reading measurements
- Help you get started because the programming protocol for each instrument is already known



IDNet

- Instrument Driver Network (IDNet)
- Instrument Driver Finder within LabVIEW

Tools » Instrumentation » Find Instrument Drivers

Help » Find Instrument Drivers

- Can be found online at ni.com/idnet

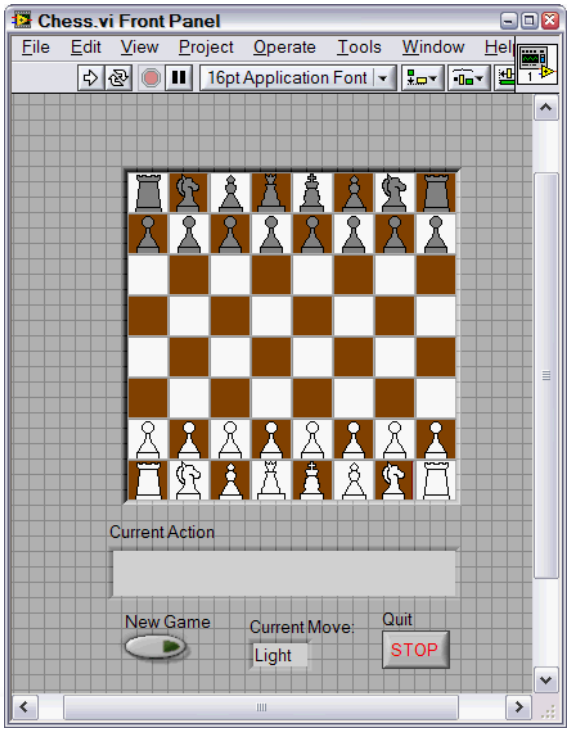
LabVIEW Case Studies

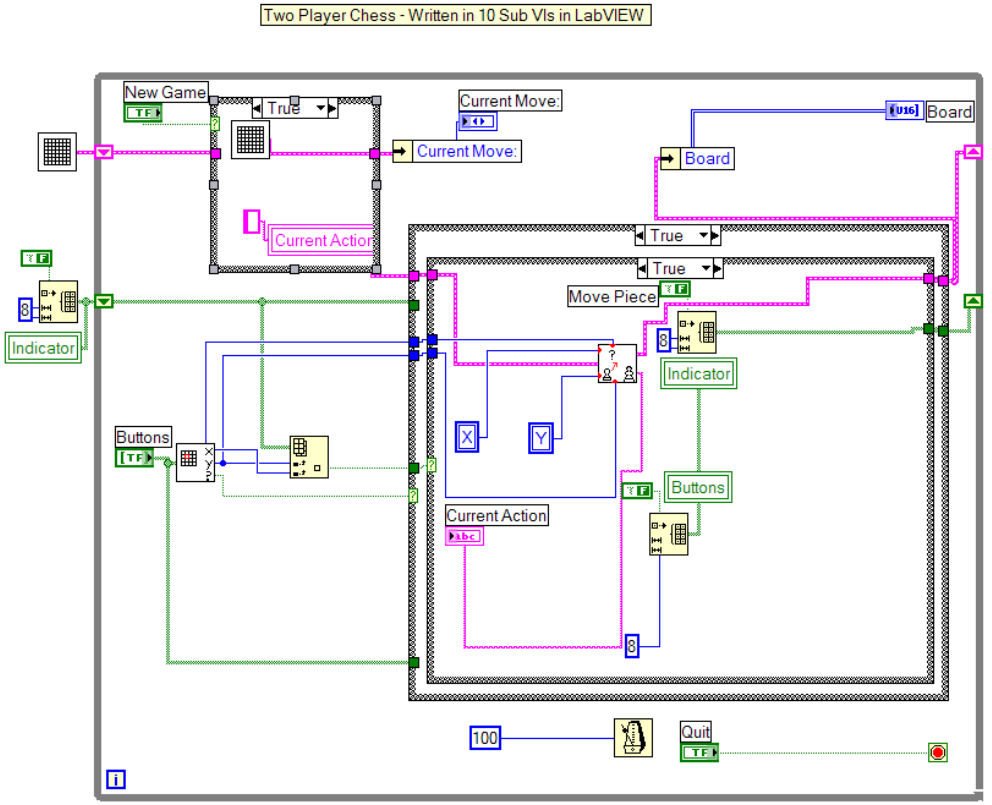
What can you do with LabVIEW? Tons!

- Two-Player Chess
- Ballistic Trajectory Calculator
- Mouse Position Calculator

Two-Player Chess

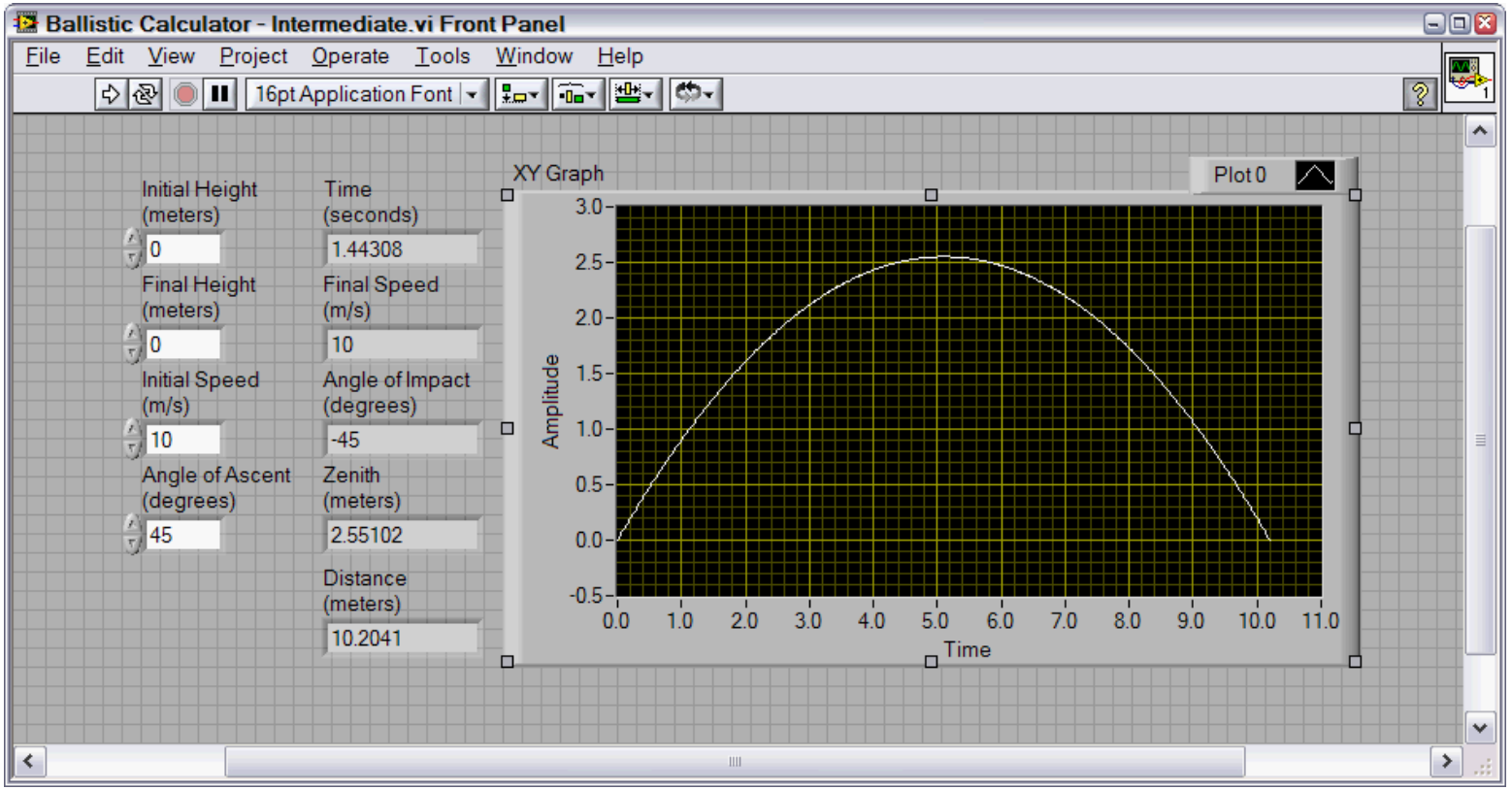
Written in LabVIEW using subVIs and custom controls





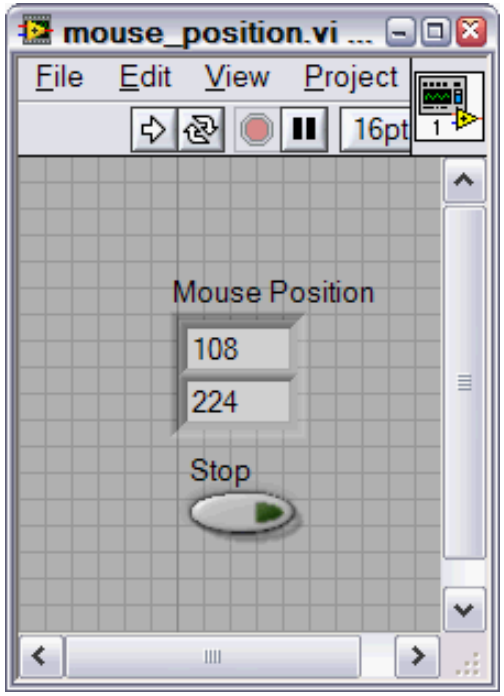
Ballistic Trajectory Calculator

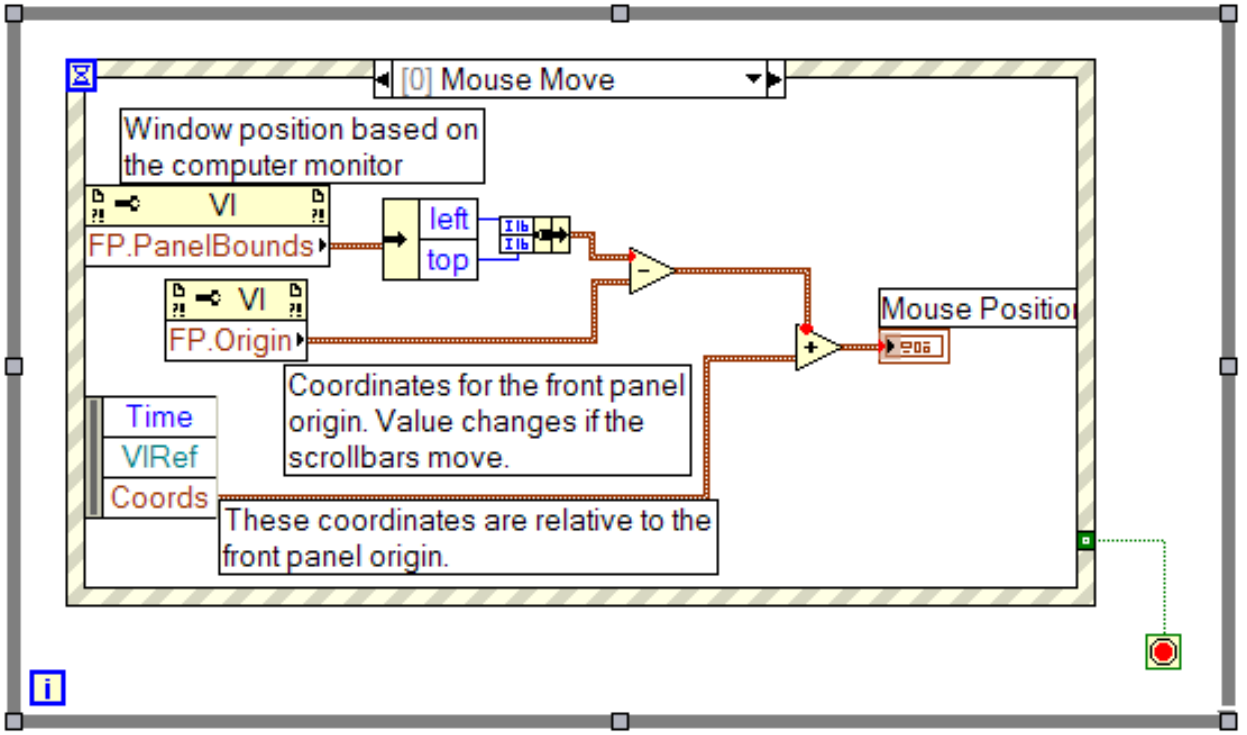
This LabVIEW VI calculates and graphs a ballistic trajectory based on the given parameters.



Mouse Position Calculator

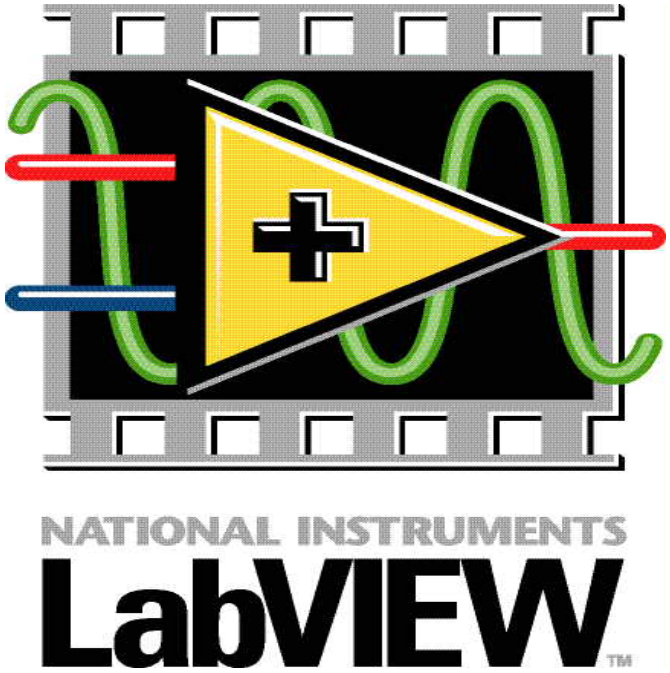
Calculate the position of your mouse cursor on the monitor using LabVIEW and event structures.





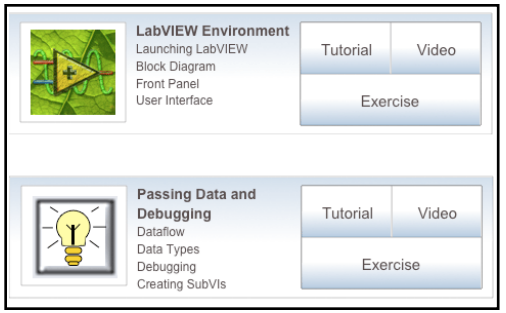
What's New in LabVIEW 8.6?

- Automatically Clean Up LabVIEW Block Diagrams
- Quick Drop: Find and Place VI Elements Faster
- Web Services: Controls Your VIs Online
- 3D Sensor Mapping: Quickly and Easily Map Data to 3D Models



Additional Resources

- NI Academic Web and Student Corner
  - ni.com/academic
  - ni.com/textbooks
  - Get your own copy of the LabVIEW Student Edition
- NI KnowledgeBase
  - ni.com/kb
- NI Developer Zone
  - ni.com/devzone
- LabVIEW Certification
  - LabVIEW Fundamentals Exam (free on ni.com/academic)
  - Certified LabVIEW Associate Developer Exam (industry-recognized certification)



The LabVIEW Certification Program

Architect

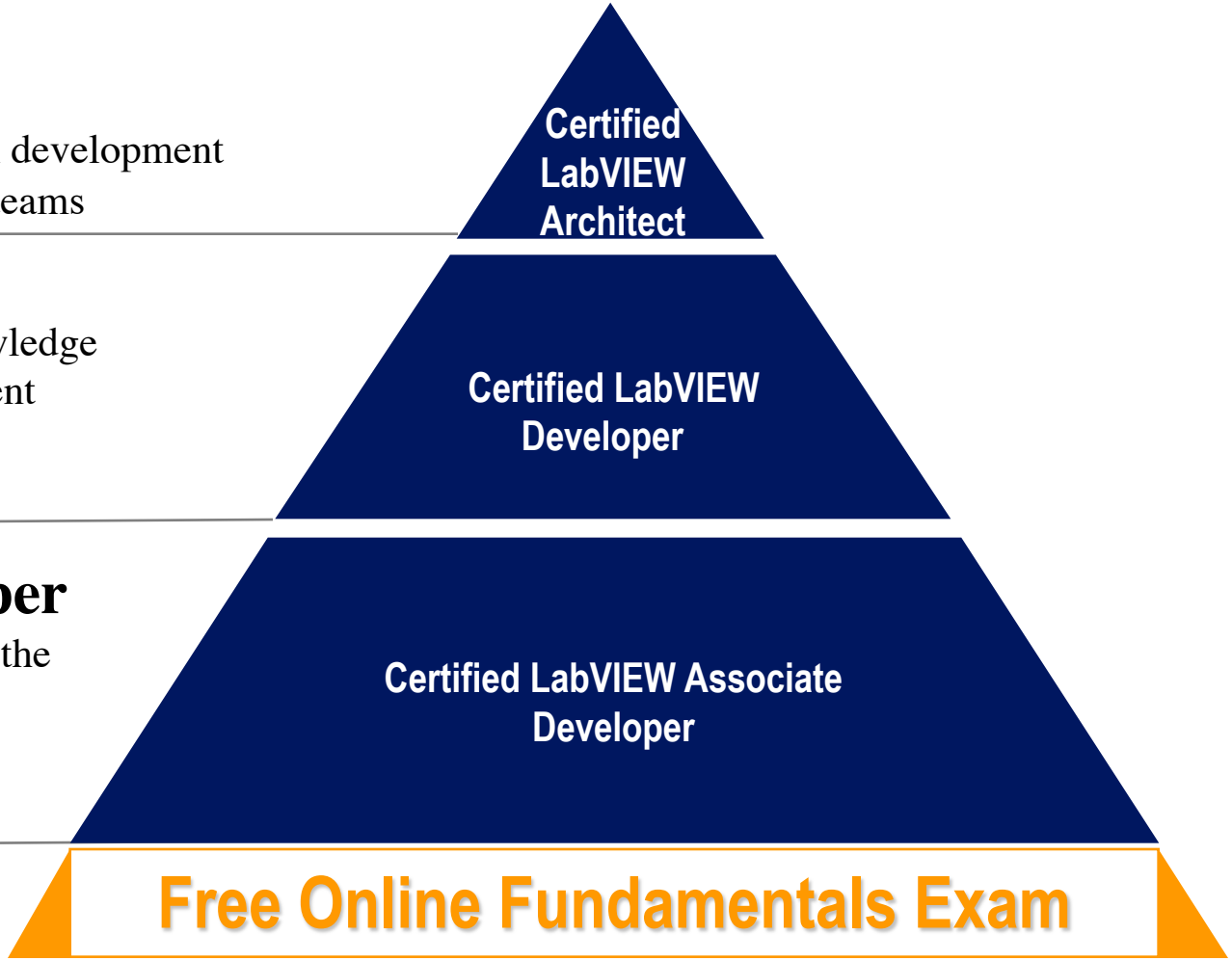
- Mastery of LabVIEW
- Expert in large application development
- Skilled in leading project teams

Developer

- Advanced LabVIEW knowledge and application development experience
- Project management skills

Associate Developer

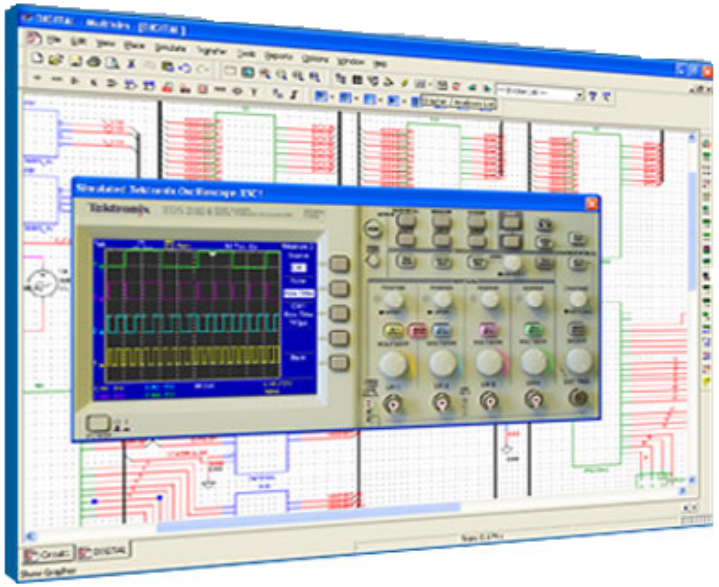
- Proficiency in navigating the LabVIEW environment
- Some application development experience



Fundamentals Exam

- Pre-certification skills test

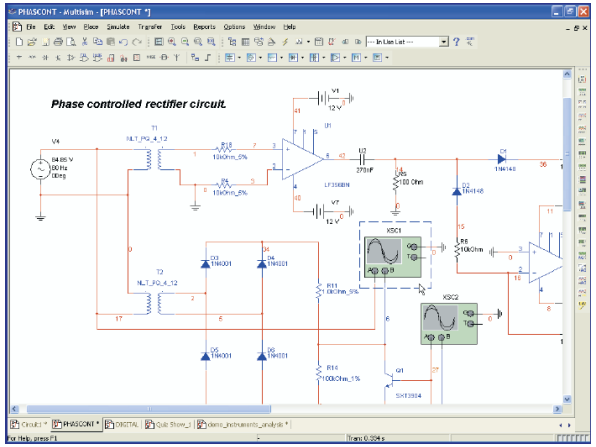
NI Multisim and Ultiboard



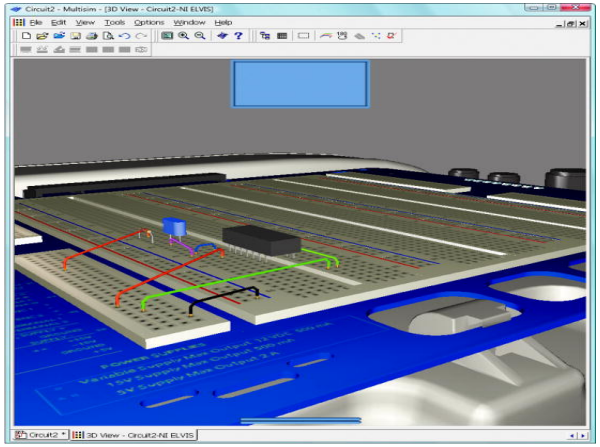
- World's most popular software for learning electronics
- 180,000 industrial and academic users
- Products include:
  - Multisim simulation and capture
  - Ultiboard PCB layout
  - Multisim MCU Module microcontroller simulation
- Low-cost student editions available
- ni.com/multisim

Multisim Integrated with LabVIEW

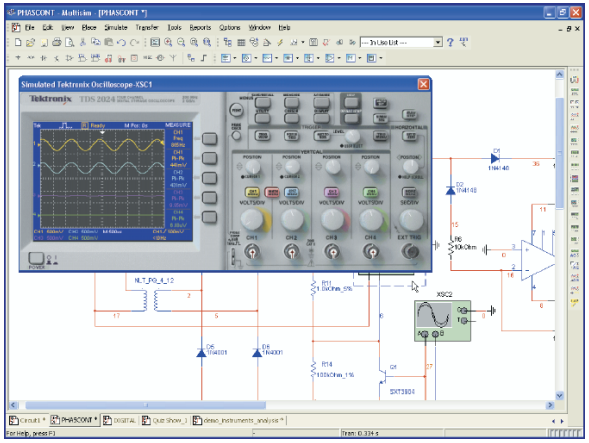
1. Create Schematic



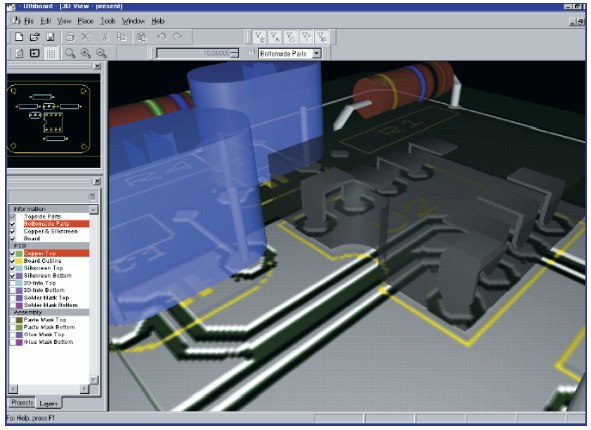
2. Virtual Breadboard



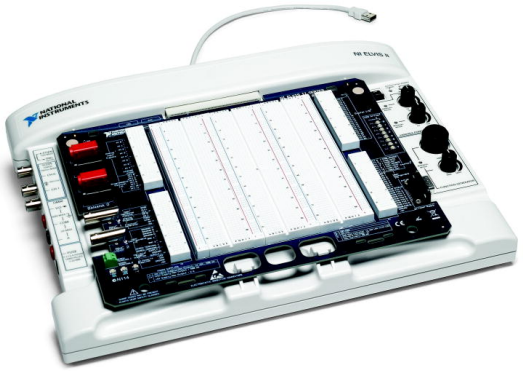
3. Simulate



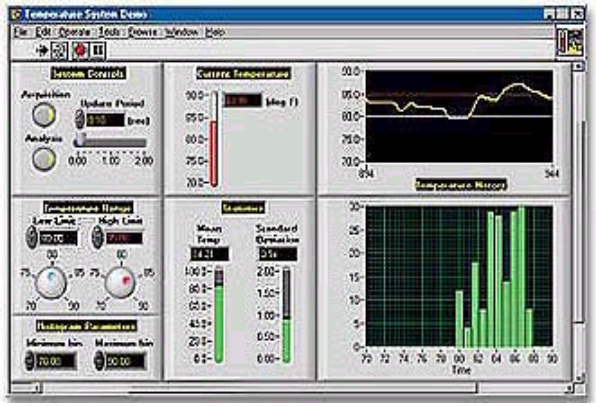
4. PCB Layout



5. Test



6. Compare



Your Next Step

Take the free LabVIEW Fundamentals Exam at
ni.com/academic

Your first step to LabVIEW certification!