Virtual Instrumentation With LabVIEW

Course Goals

- Understand the components of a Virtual Instrument
- Introduce LabVIEW and common LabVIEW functions
- Build a simple data acquisition application
- Create a subroutine in LabVIEW
- Work with Arrays, Clusters, and Structures
- Learn About Printing & Documentation Features
- Develop in Basic Programming Architectures
- Publish VIs on the Web

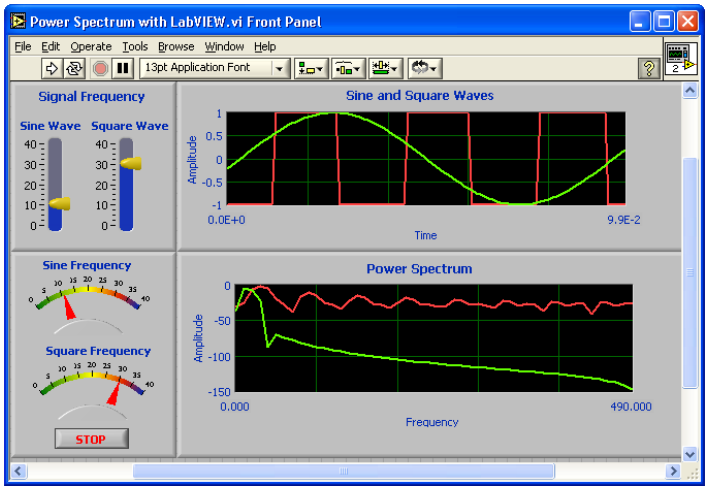
Section I

- LabVIEW terms
- Components of a LabVIEW application
- LabVIEW programming tools
- Creating an application in LabVIEW

LabVIEW Programs Are Called Virtual Instruments (VIs)

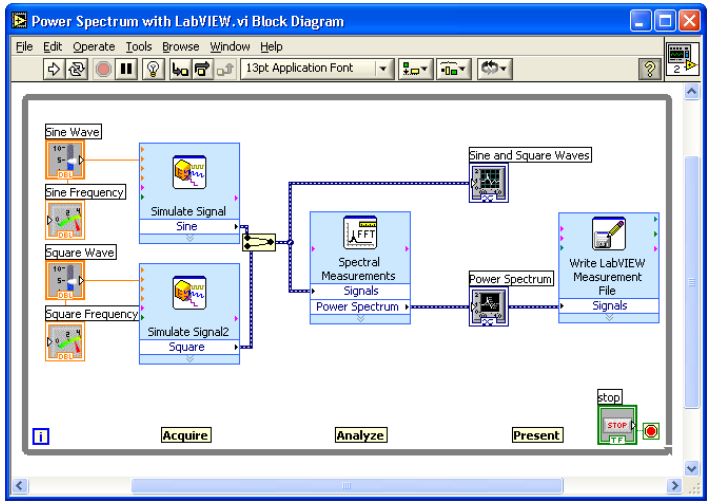
Front Panel

- Controls = Inputs
- Indicators = Outputs

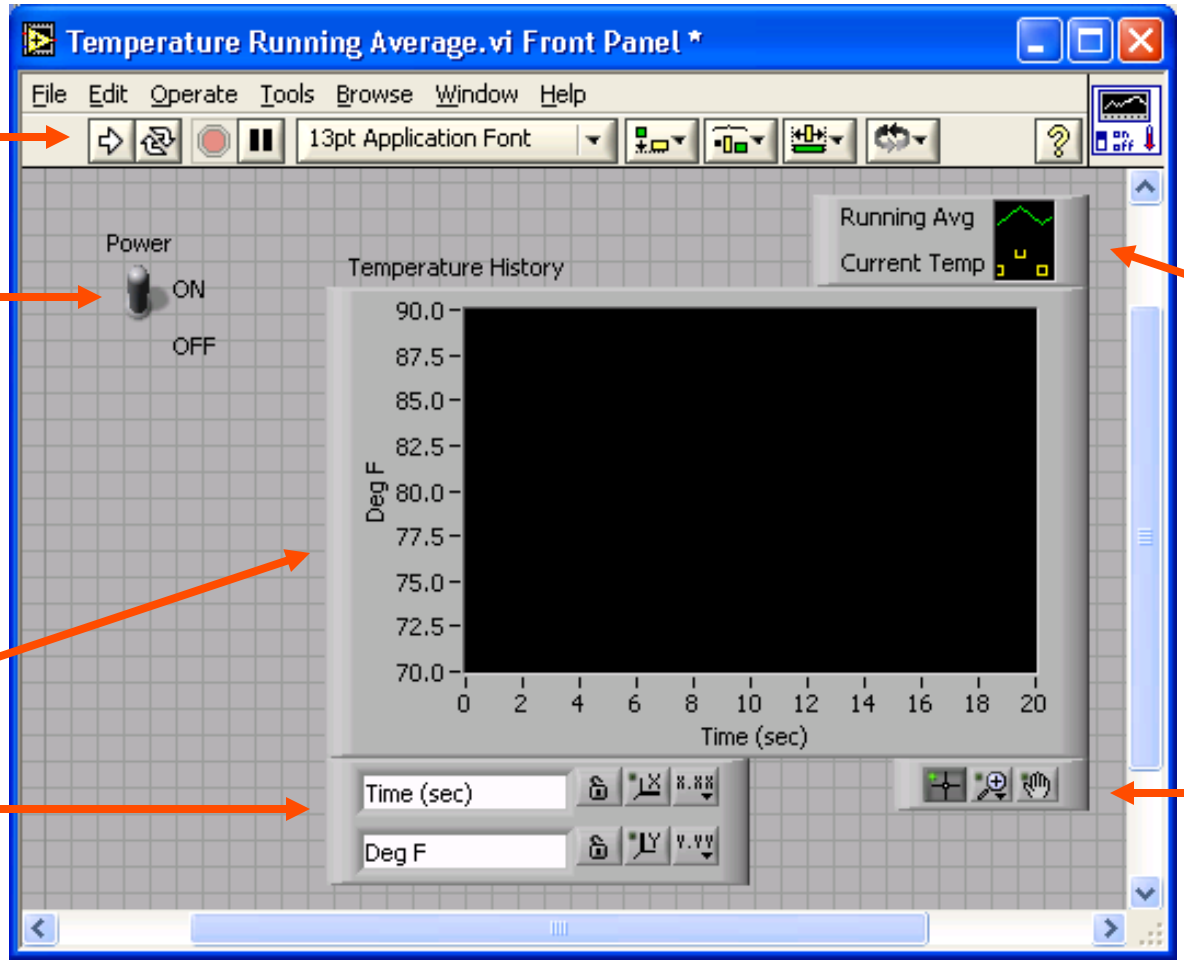


Block Diagram

- Accompanying “program” for front panel
- Components “wired” together



VI Front Panel



Front Panel Toolbar

Boolean Control

Waveform Graph

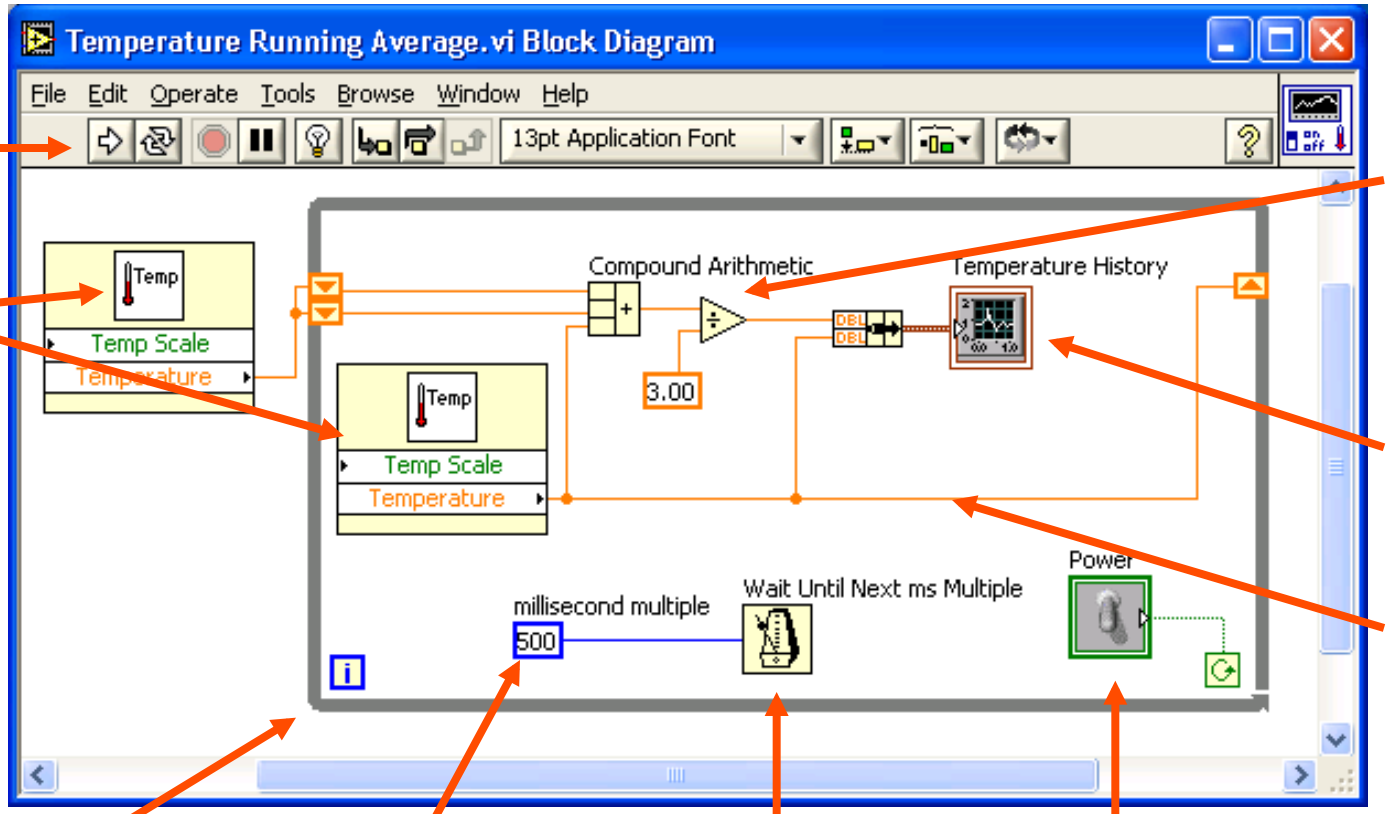
Plot Legend

Icon

Graph Legend

Scale Legend

VI Block Diagram



Block Diagram Toolbar

SubVI

While Loop Structure

Numeric Constant

Timing Function

Boolean Control Terminal

Wire Data

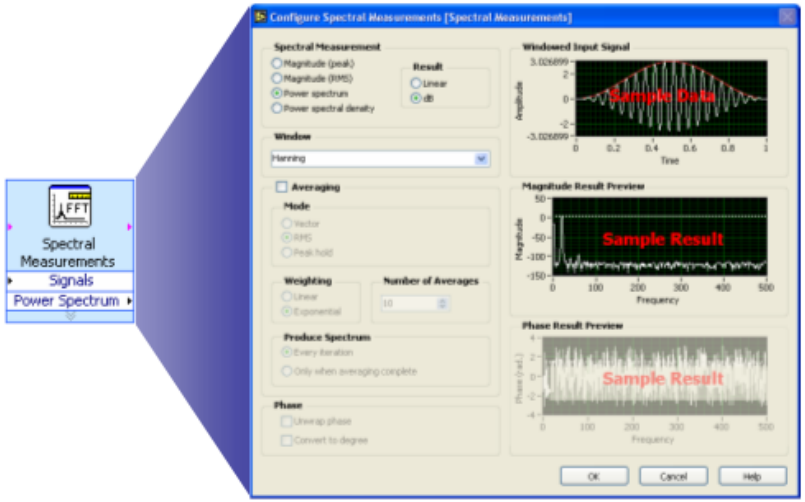
Graph Terminal

Divide Function

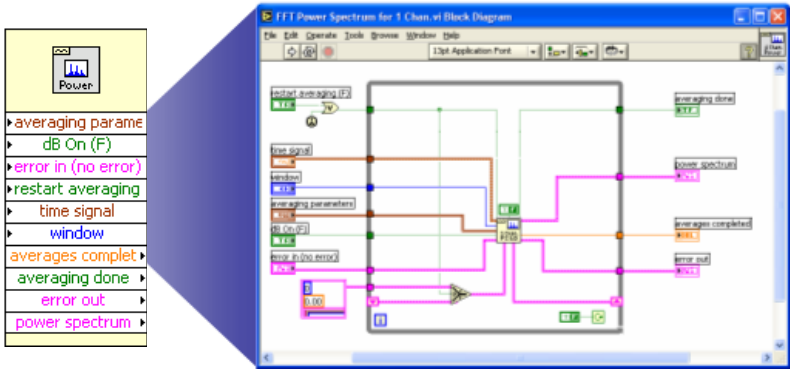
Express VIs, VIs and Functions

- 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



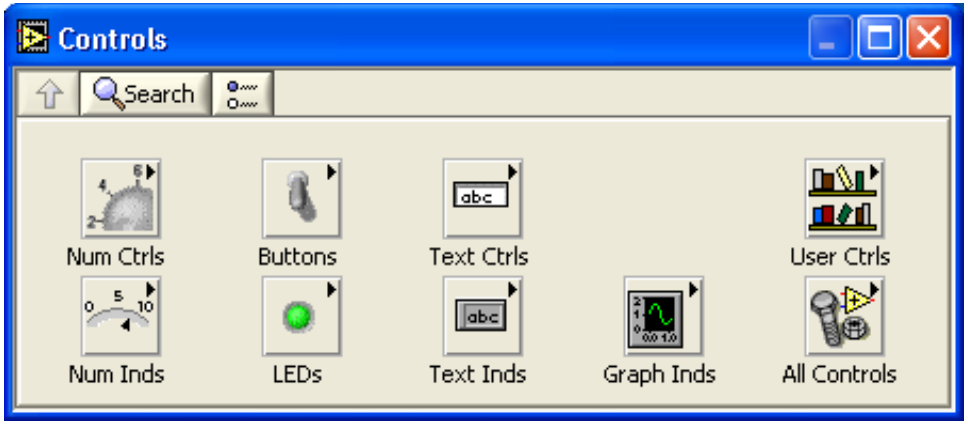
Express VI

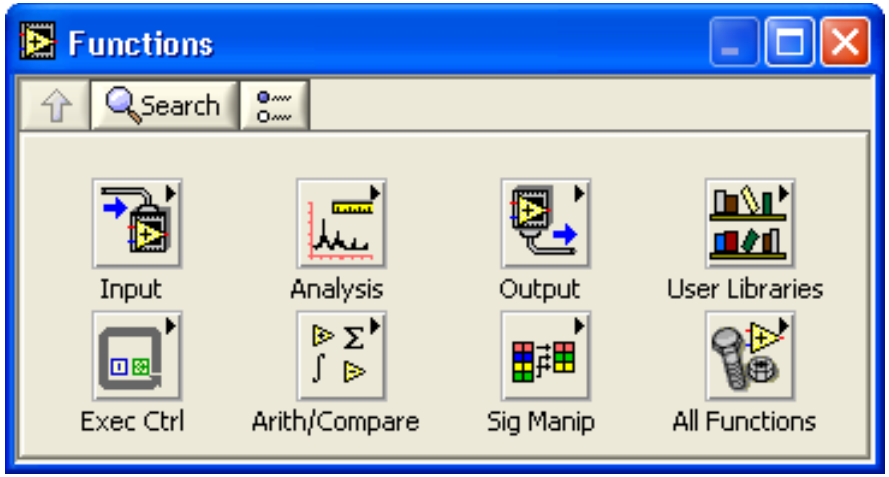


Standard VI

Controls and Functions Palettes

Controls Palette
(Front Panel Window)





Functions Palette
(Block Diagram Window)

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

Status Toolbar

Run Button

Continuous Run Button

Abort Execution

Pause/Continue Button

13pt Application Font — Text Settings

Align Objects

Distribute Objects

Reorder

Resize front panel objects

Additional Buttons on the Diagram Toolbar

Execution Highlighting Button

Step Into Button

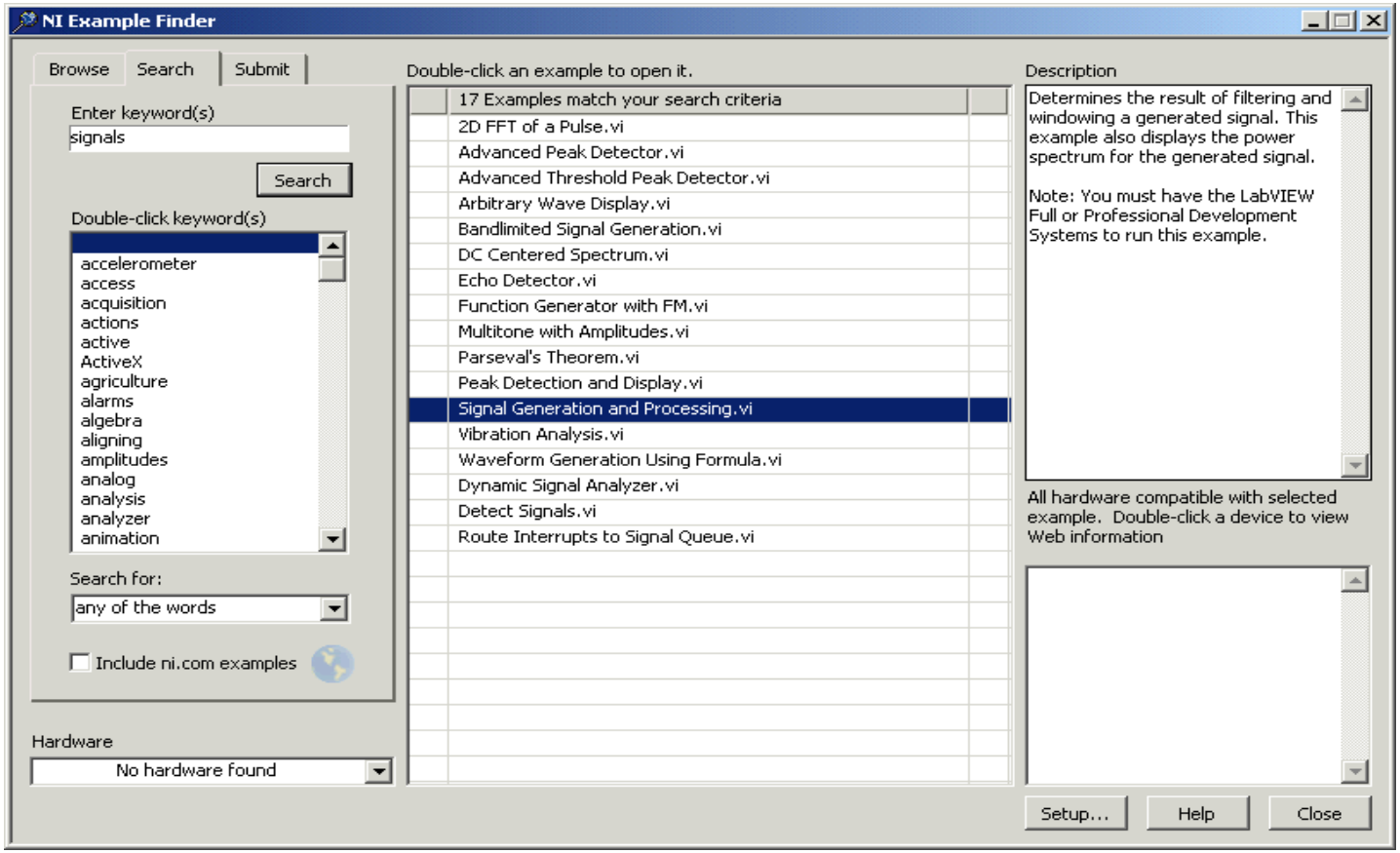
Step Over Button

Step Out Button

Do Not Delete This Slide

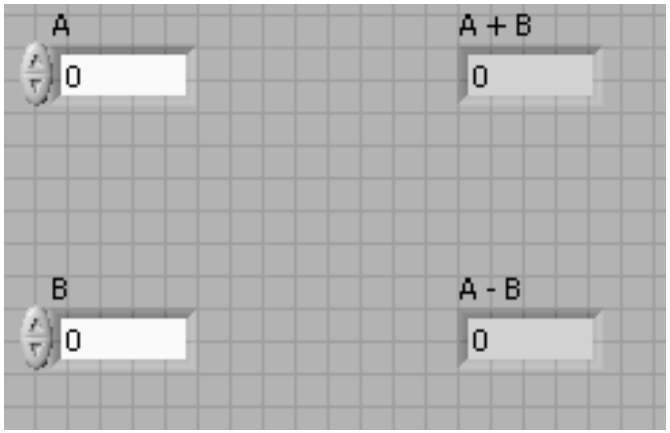
Open and Run a Virtual Instrument

Example finder

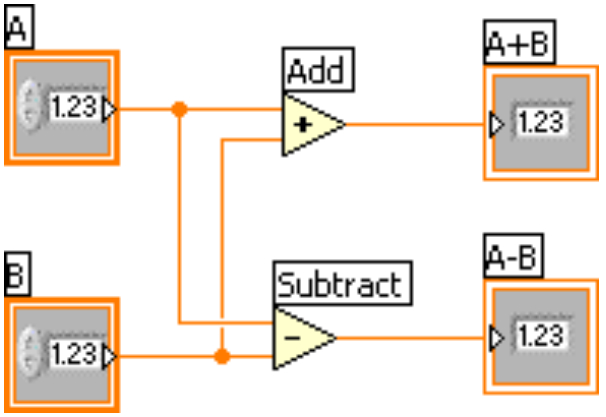


Creating a VI

Front Panel Window



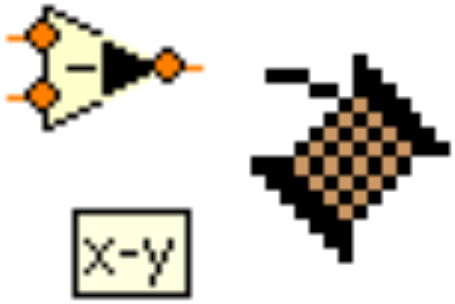
Block Diagram Window

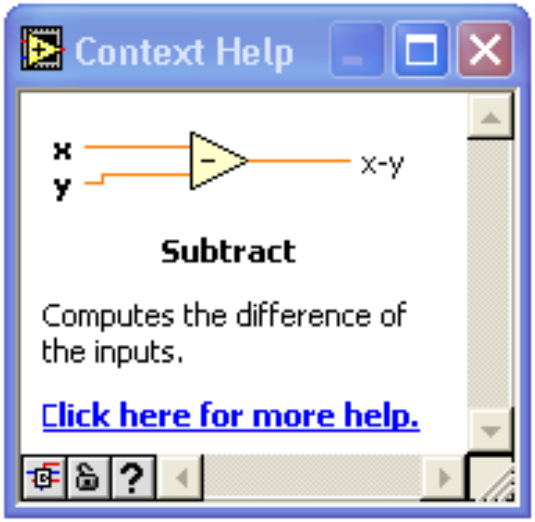


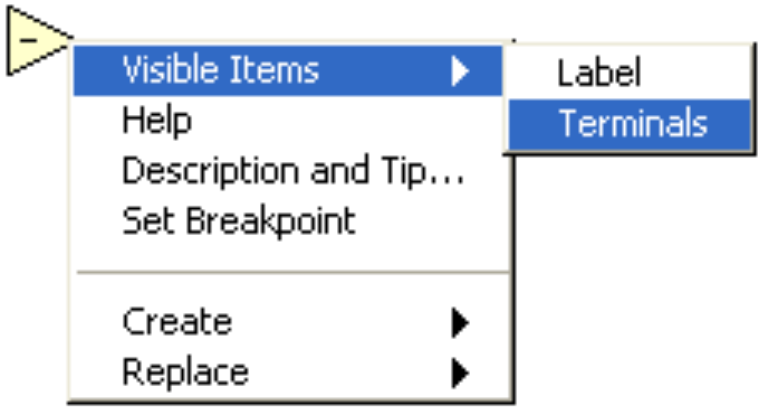
Control Terminals

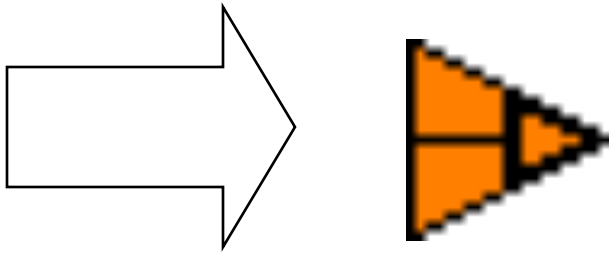
Indicator Terminals

Creating a VI – Block Diagram



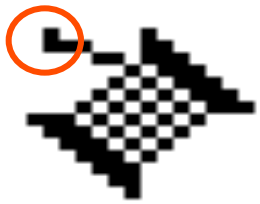




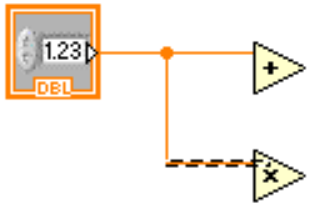


Wiring Tips – Block Diagram

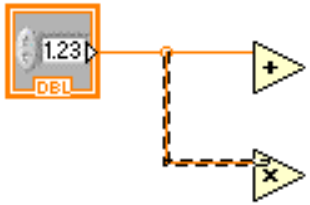
Wiring "Hot Spot"



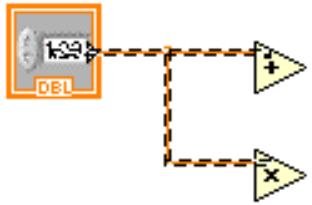
Click To Select Wires



single-click

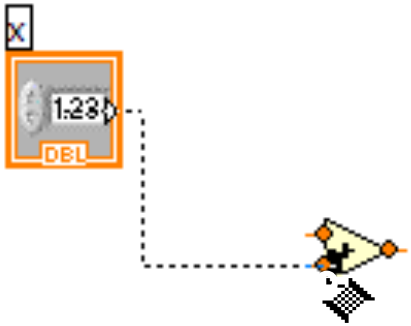


double-click

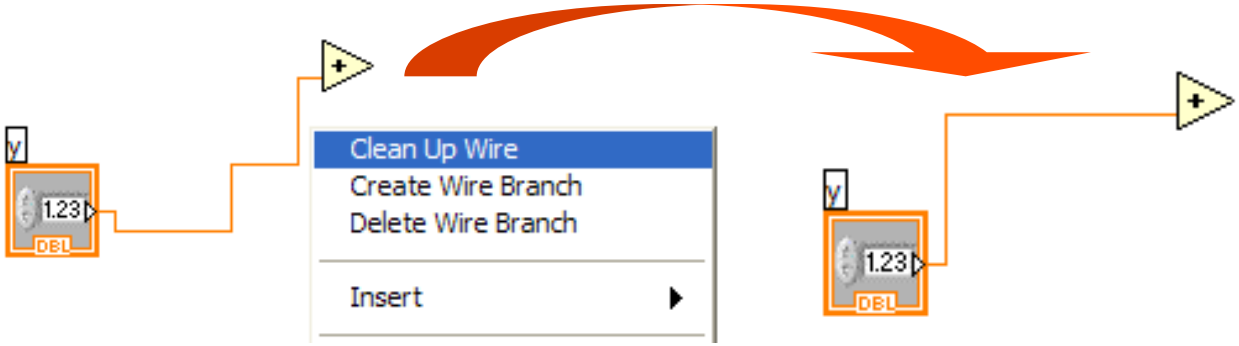


triple-click

Use Automatic Wire Routing



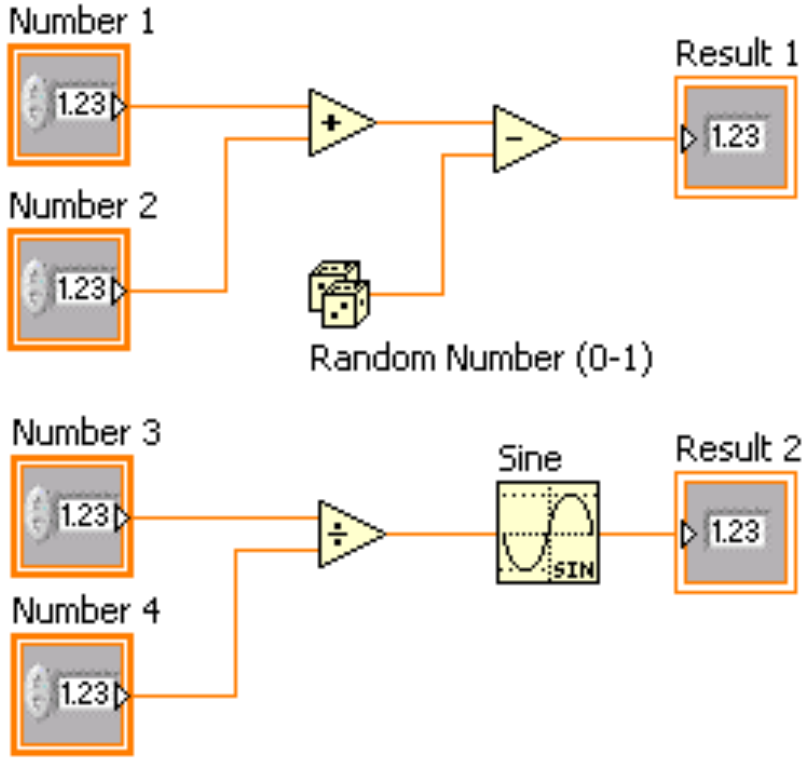
Clean Up Wiring



Do Not Delete This Slide

Dataflow Programming

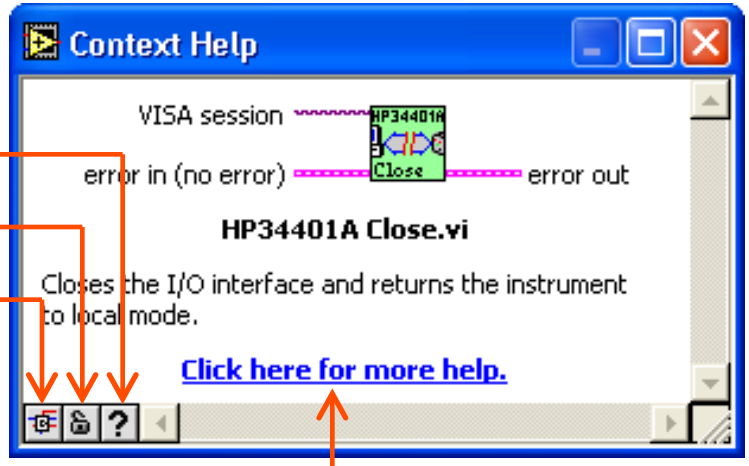
- Block diagram executes dependent on the flow of data; block diagram does NOT execute left to right
- Node executes when data is available to ALL input terminals
- Nodes supply data to all output terminals when done



Help Options

Context Help

- Online help
- Lock help
- Simple/Complex Diagram help
- Ctrl + H

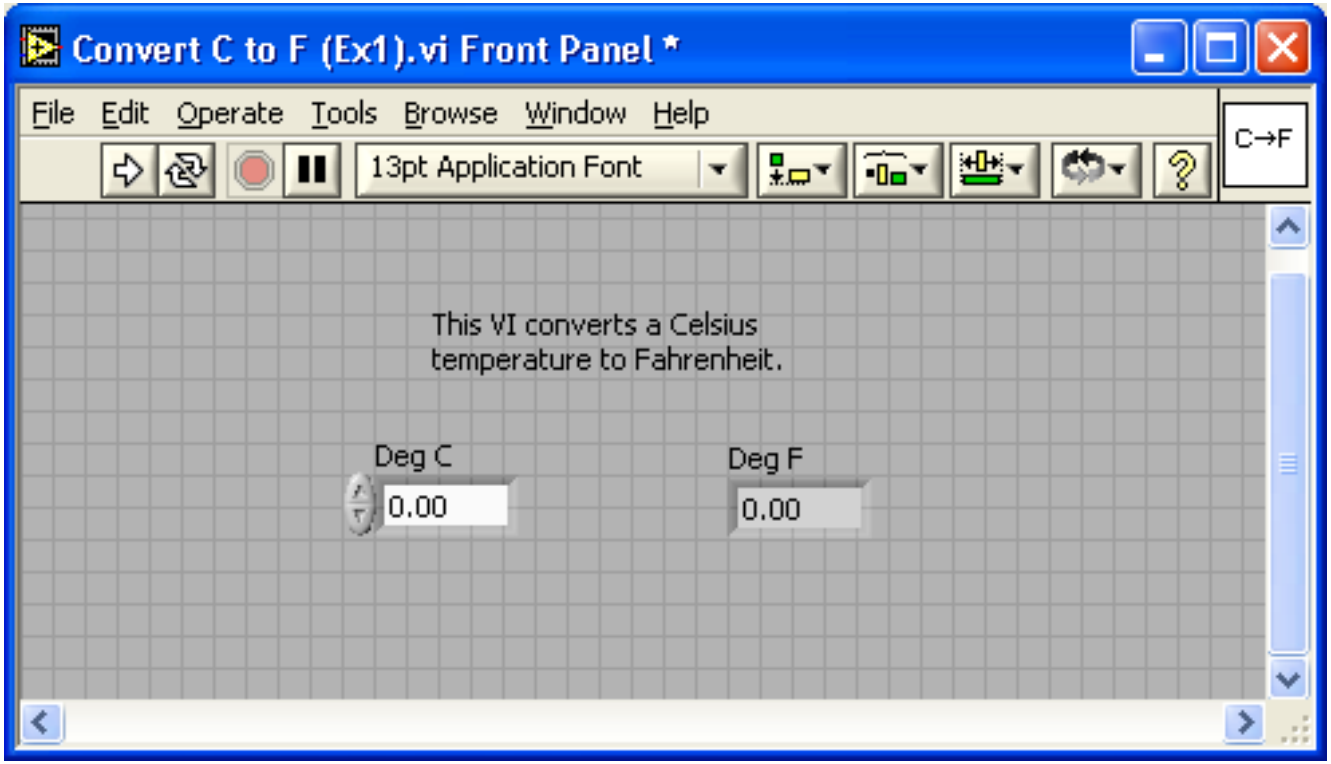


Online reference

- All menus online
- Pop up on functions in diagram to access online info directly

Do Not Delete This Slide

Exercise 1 - Convert °C to °F



Debugging Techniques

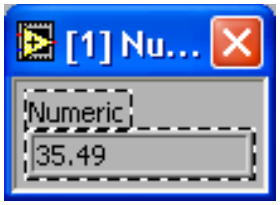
- **Finding Errors**

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.

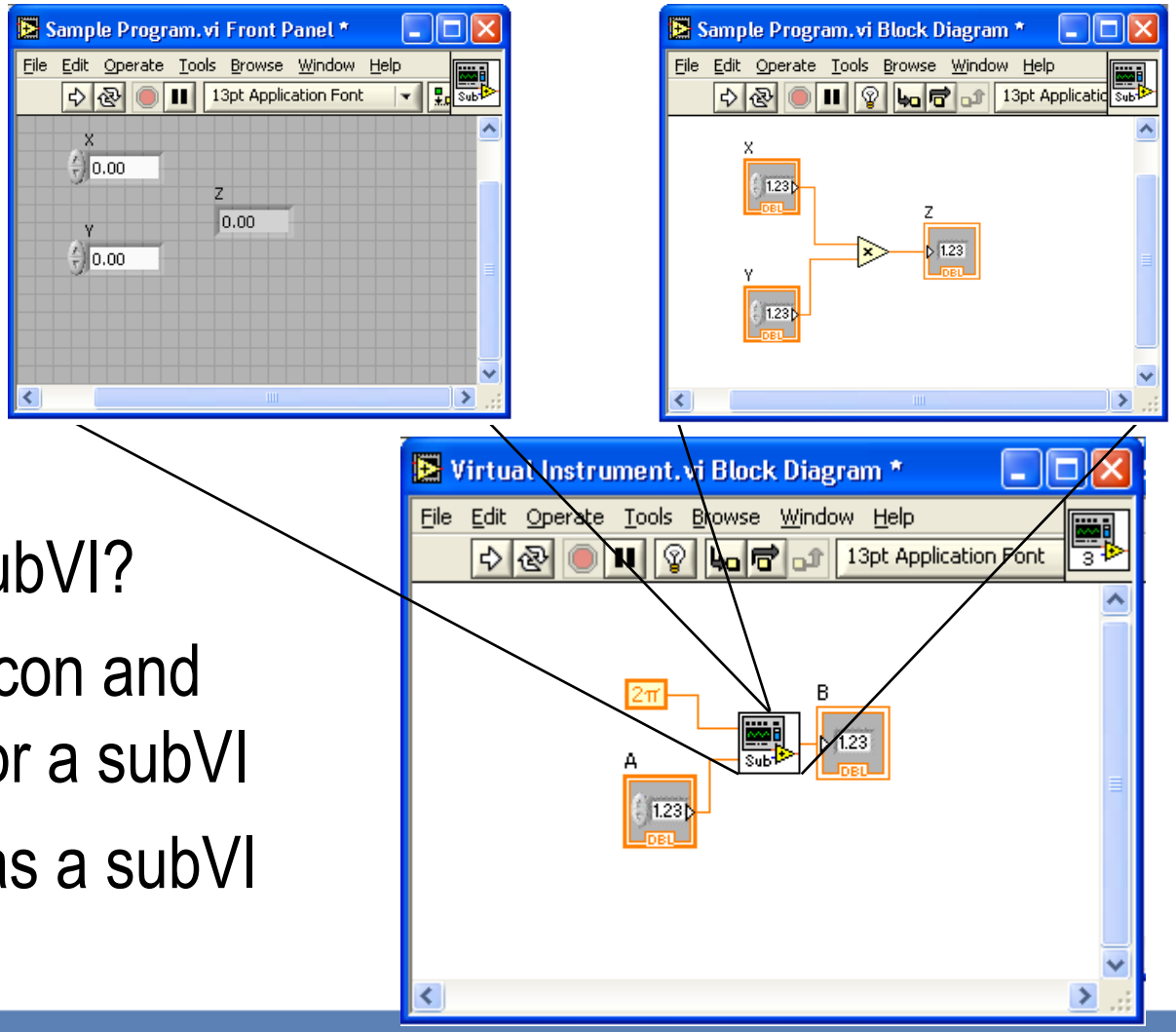
- **Probe**



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

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

Section II – SubVIs



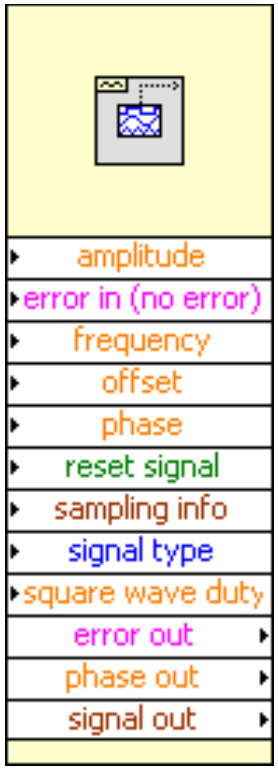
- What is a subVI?
- Making an icon and connector for a subVI
- Using a VI as a subVI

Block Diagram Nodes

Icon

Expandable Node

Expanded Node

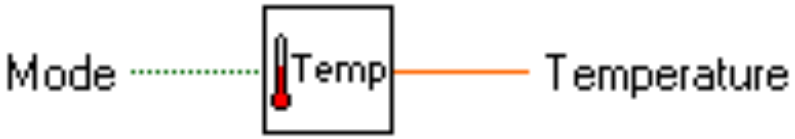


- Function Generator VI
- Same VI, viewed three different ways
- Yellow field designates a standard VI
- Blue field designates an Express VI

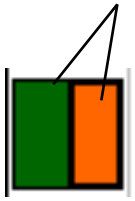
SubVIs

- A SubVI is a VI that can be used within another VI
- Similar to a subroutine
- Advantages
  - Modular
  - Easier to debug
  - Don't have to recreate code
  - Require less memory

Icon and Connector

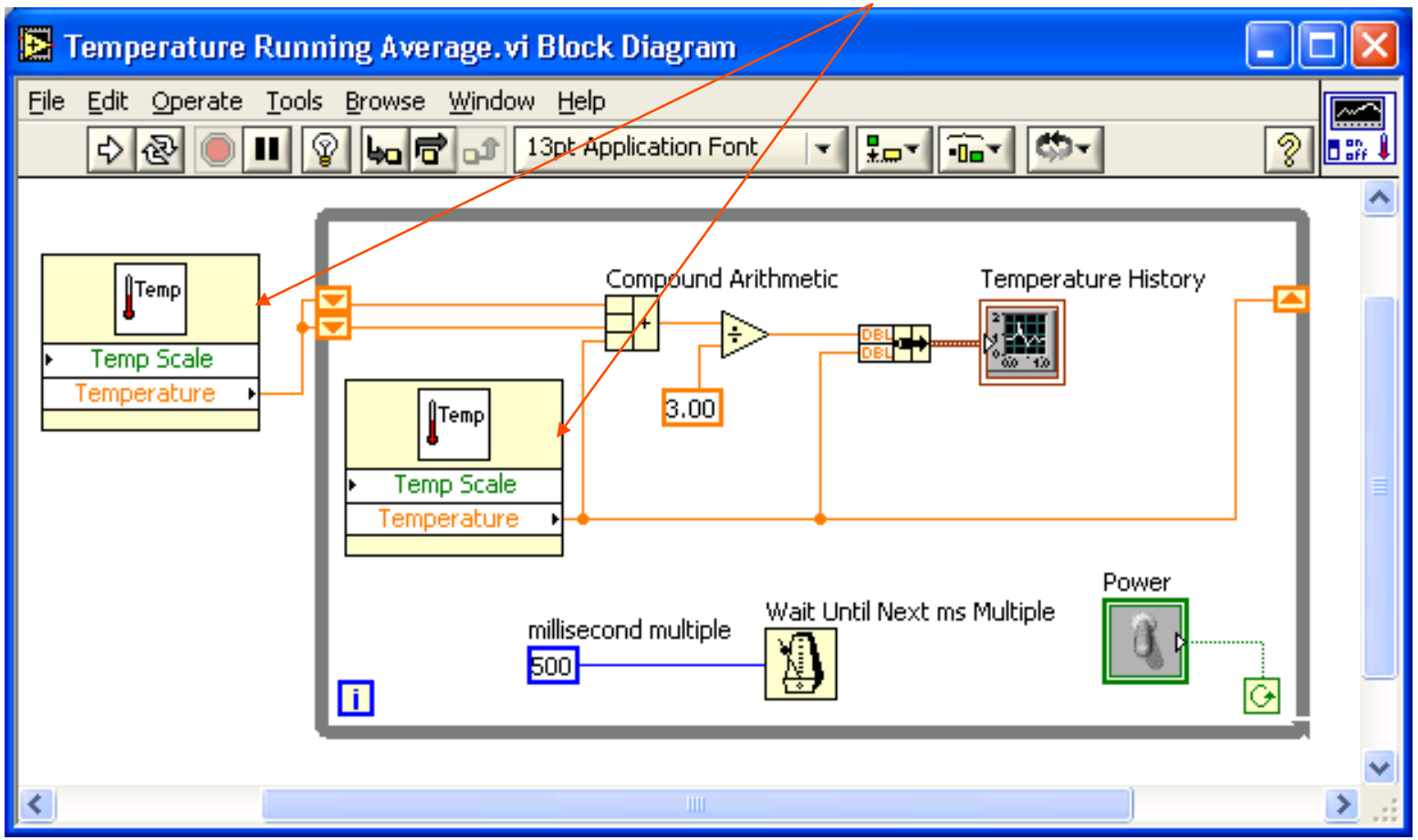


- An icon represents a VI in other block diagrams



- A connector shows available terminals for data transfer

SubVIs

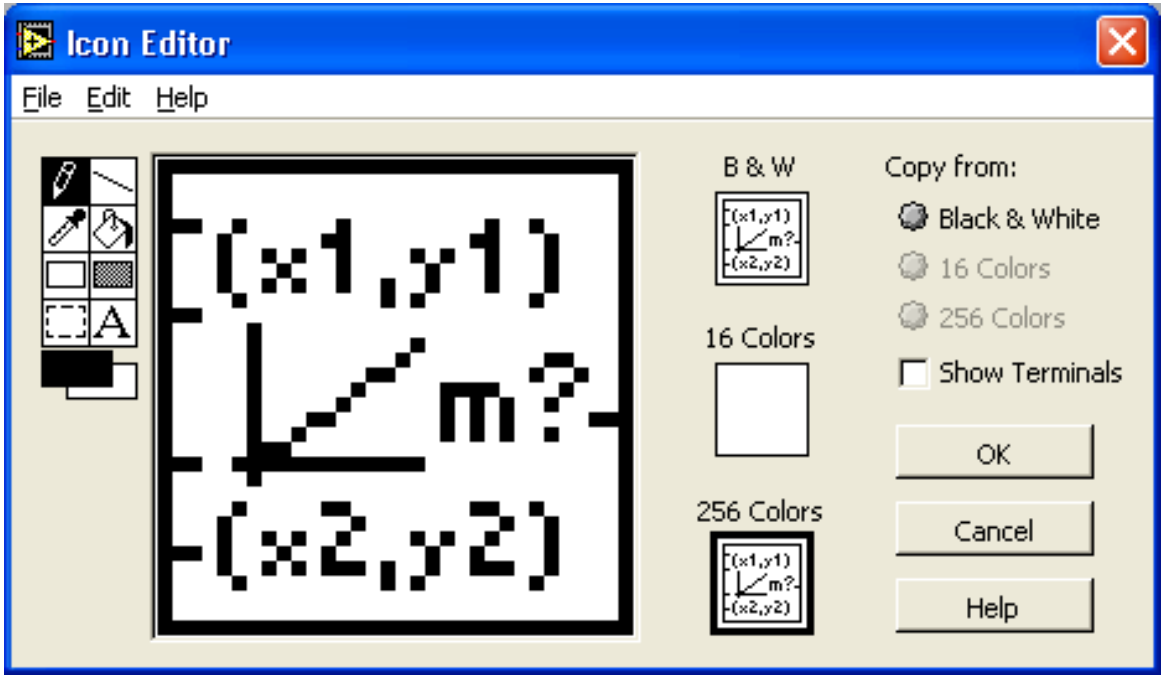

Sub VIs
Temperature Running Average.vi Block Diagram
File Edit Operate Tools Browse Window Help
13pt Application Font
Temp
Temp Scale
Temperature
Compound Arithmetic
3.00
Temperature History
DBL
DBL
Temp
Temp Scale
Temperature
millisecond multiple
500
Wait Until Next ms Multiple
Power

Steps to Create a SubVI

- Create the Icon
- Create the Connector
- Assign Terminals
- Save the VI
- Insert the VI into a Top Level VI

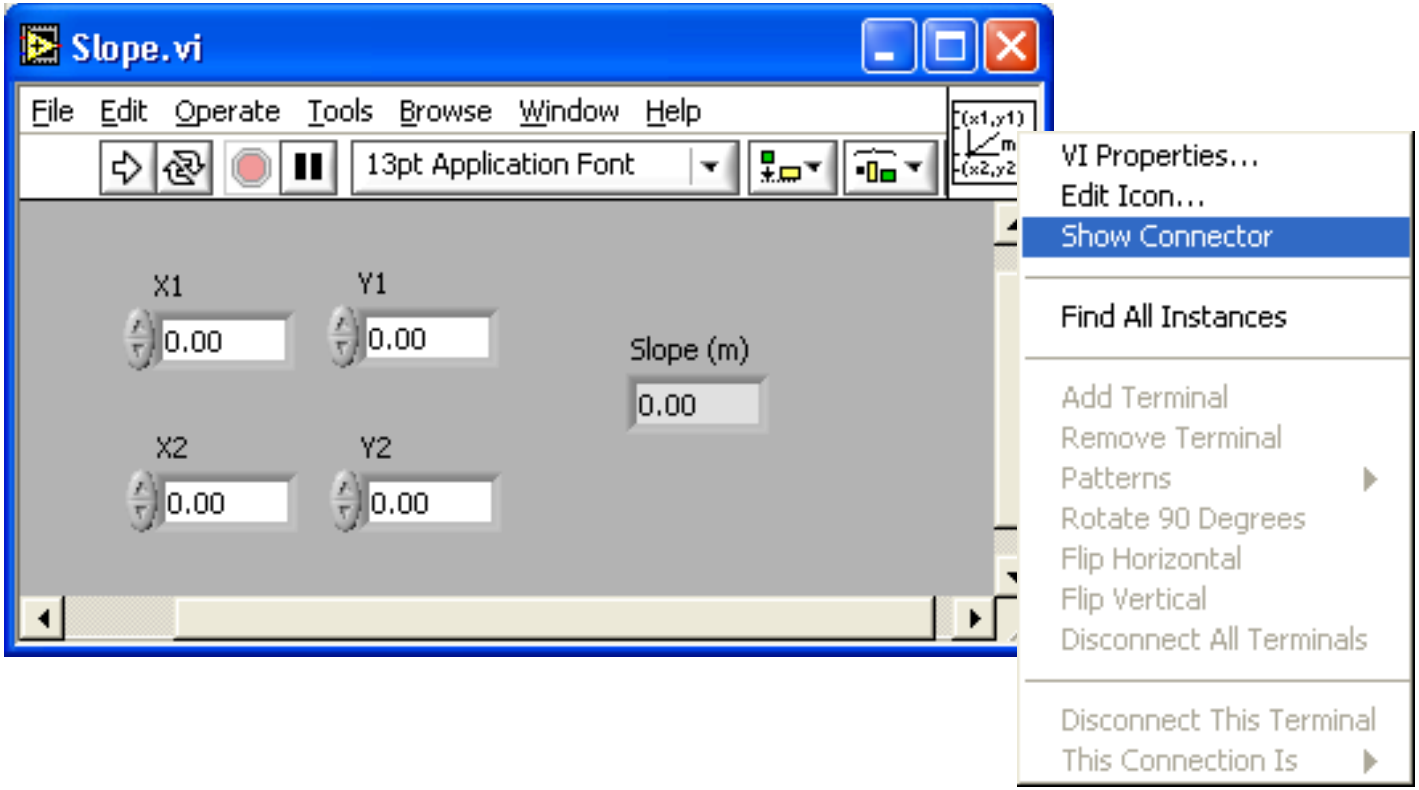
Create the Icon

- Right-click on the icon in the block diagram or front panel

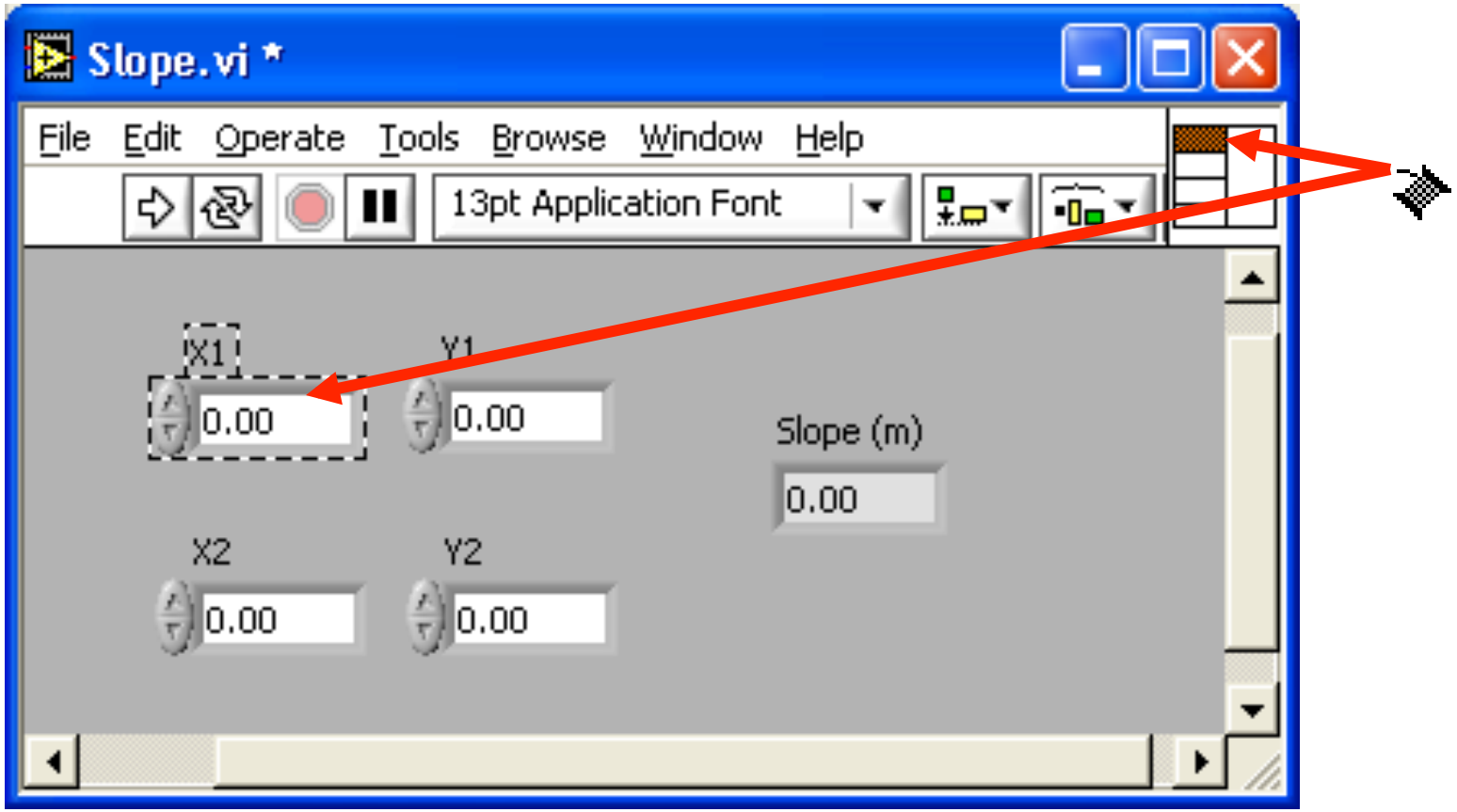


Create the Connector

Right click on the icon pane (front panel only)



Assign Terminals



Save The VI

- Choose an Easy to Remember Location
- Organize by Functionality
  - Save Similar VIs into one directory (e.g. Math Utilities)
- Organize by Application
  - Save all VIs Used for a Specific Application into one directory or library file (e.g. Lab 1 – Frequency Response)
    - Library Files (.llbs) combine many VI's into a single file, ideal for transferring entire applications across computers

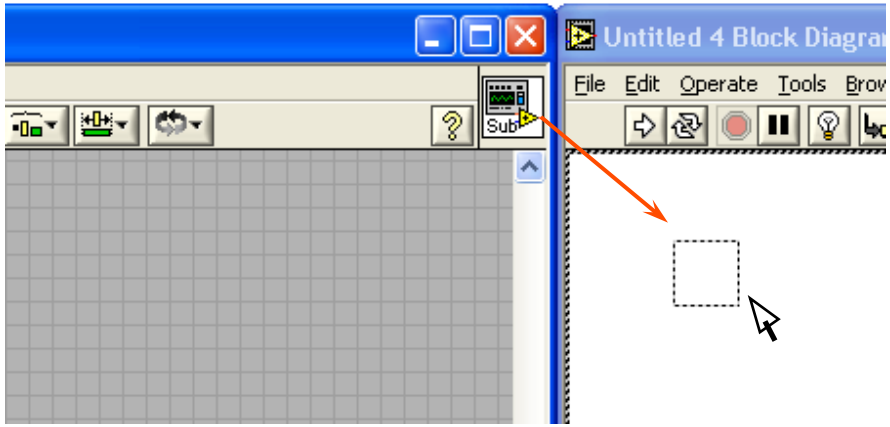
Insert the SubVI into a Top Level VI

Accessing user-made subVIs

Functions >>All Functions >> Select a VI

Or

Drag icon onto target diagram

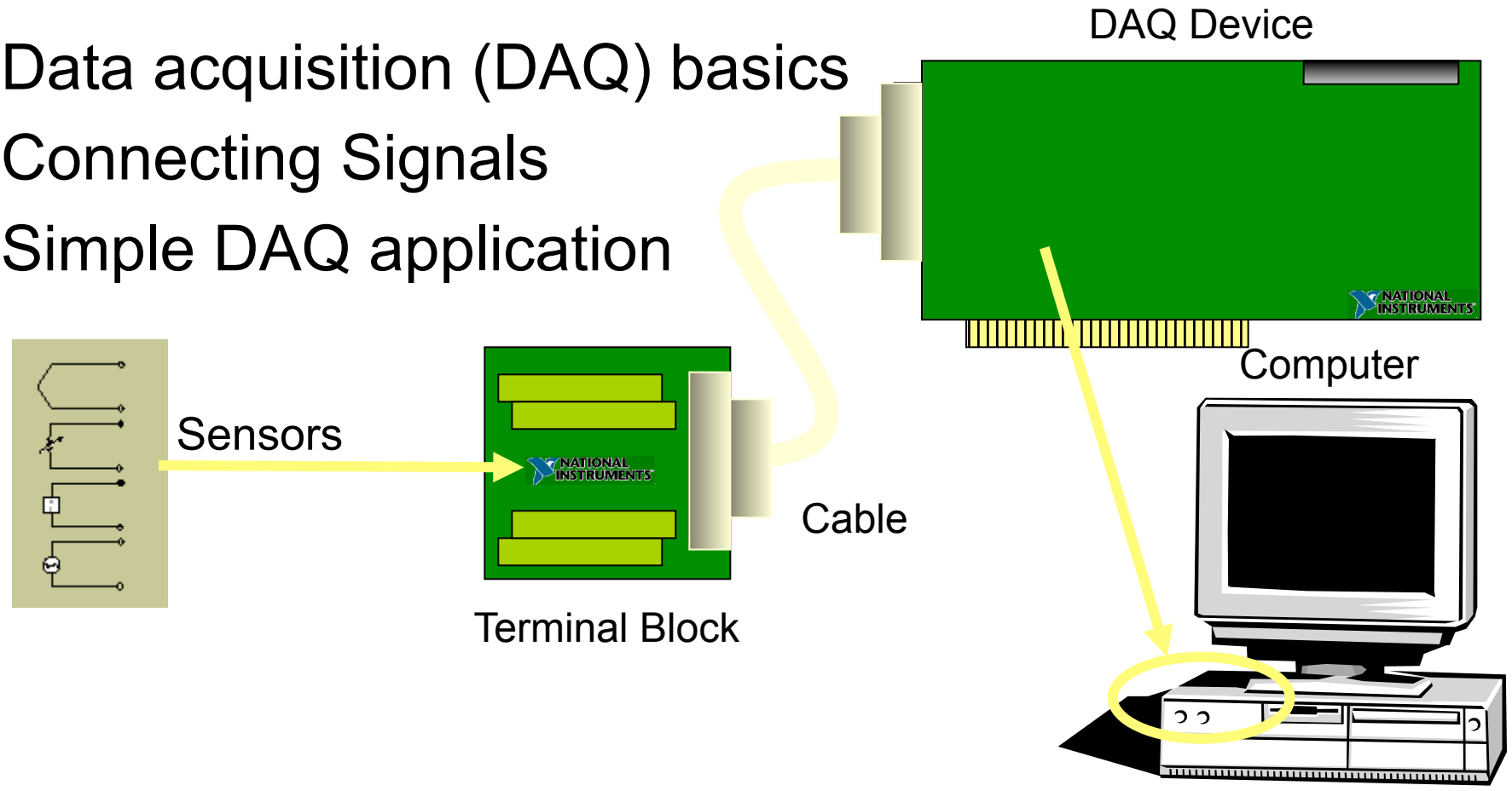


Tips for Working in LabVIEW

- 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
- VI Properties – Configure VI Appearance, Documentation, etc.

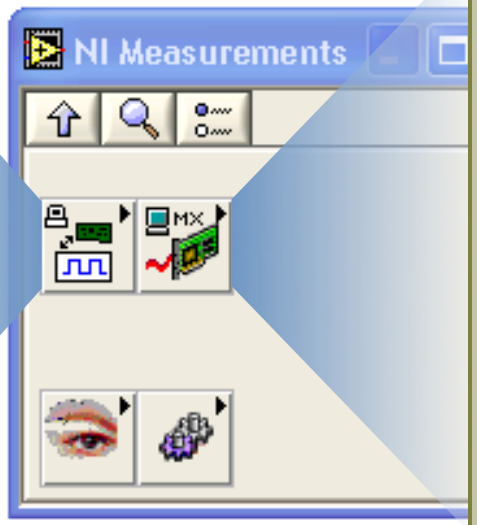
Section III – Data Acquisition

- Data acquisition (DAQ) basics
- Connecting Signals
- Simple DAQ application



Data Acquisition in LabVIEW

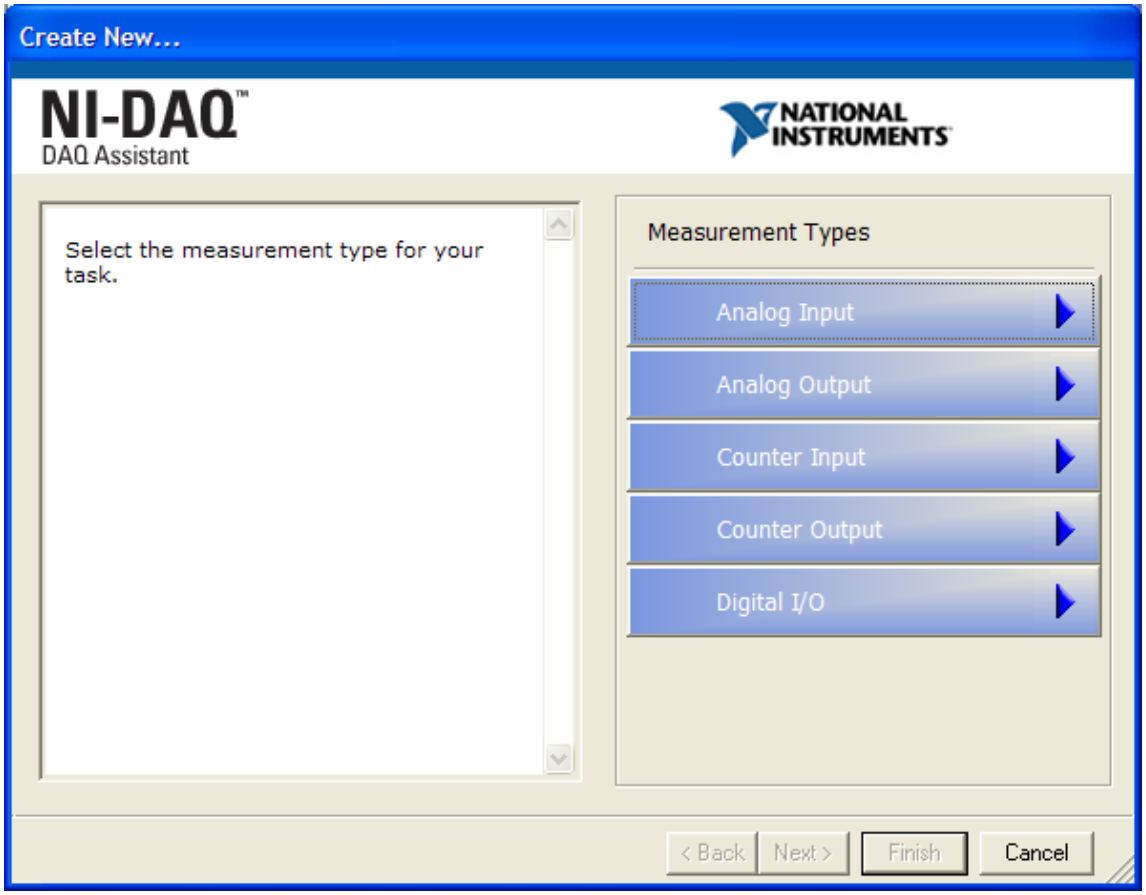
Traditional NI-DAQ
Specific VIs for performing:
- Analog Input
- Analog Output
- Digital I/O
- Counter operations



NI-DAQmx
Next generation driver:
- VIs for performing a task
- One set of VIs for all measurement types

DAQ – Data Acquisition

Temperature Acquisition using the DAQ Assistant



Data Acquisition Terminology

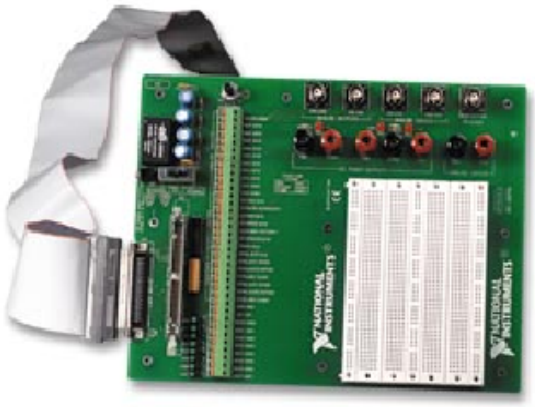
- **Resolution** - Determines How Many Different Voltage Changes Can Be Measured
  - Larger Resolution → More Precise Representation of Signal
- **Range** - Minimum and Maximum Voltages
  - Smaller range → More Precise Representation of Signal
- **Gain** - Amplifies or Attenuates Signal for Best Fit in Range

Do Not Delete This Slide

Hardware Connections

BNC-2120

SC-2075

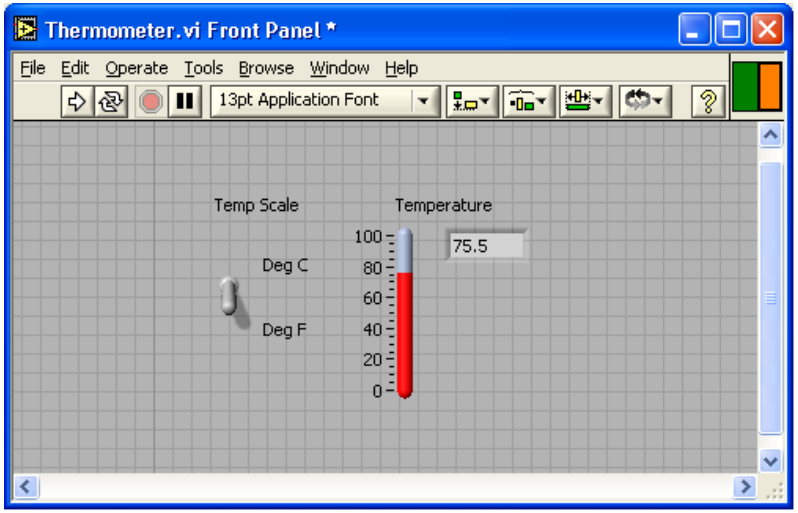


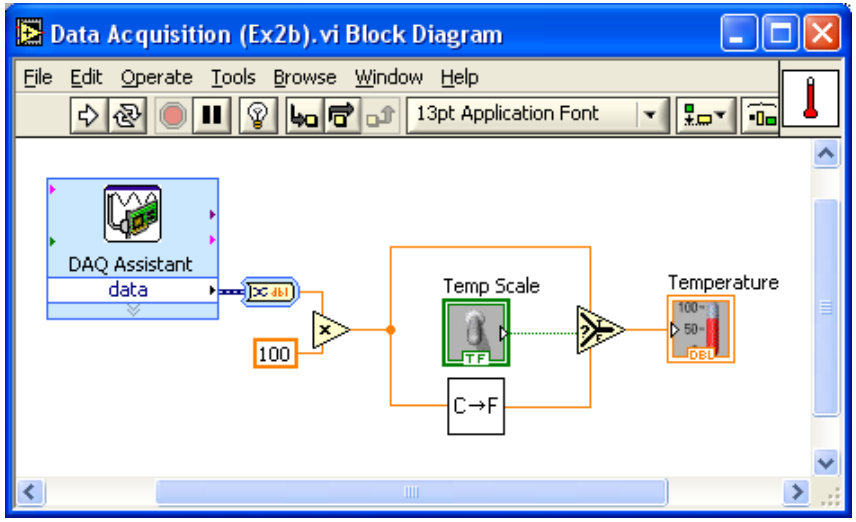
NI-ELVIS

SCB-68

Exercise 2 – Simple Data Acquisition

Complete Convert C to F.vi, then create Thermometer.vi.





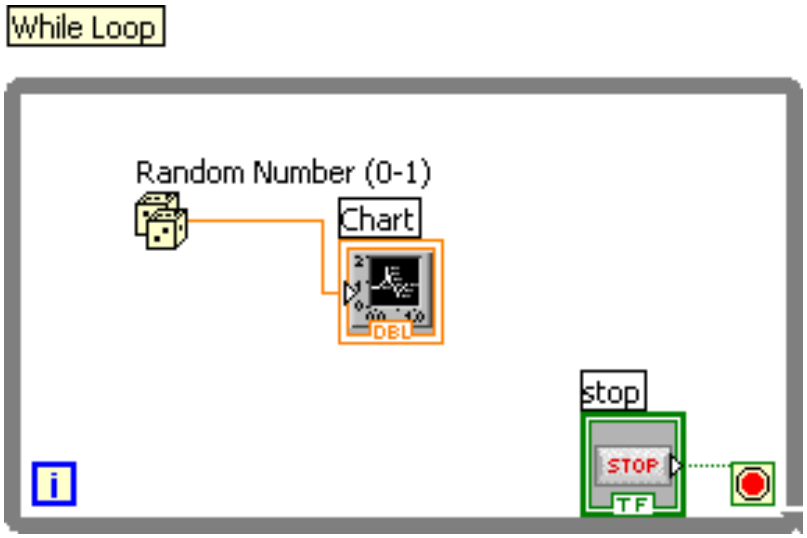
Do Not Delete This Slide

Section IV – Loops and Charts

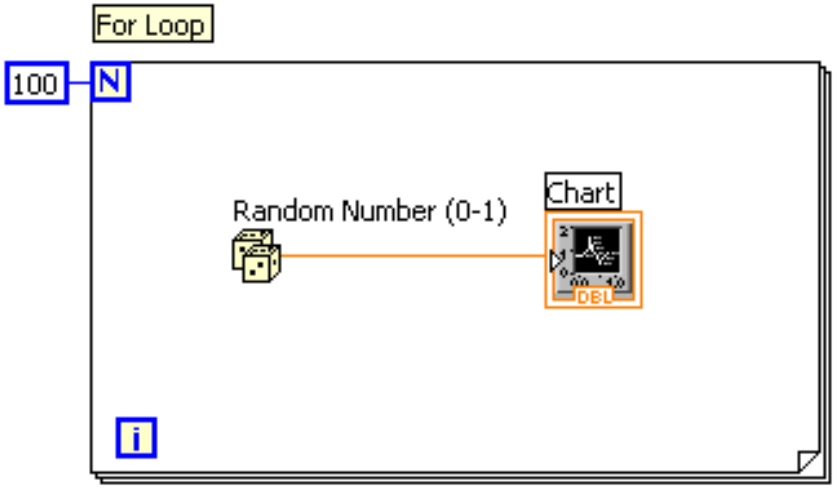
- For Loop
- While Loop
- Charts
- Multiplots

Loops

- While Loops
  - Have Iteration Terminal
  - Always Run at least Once
  - Run According to Conditional Terminal

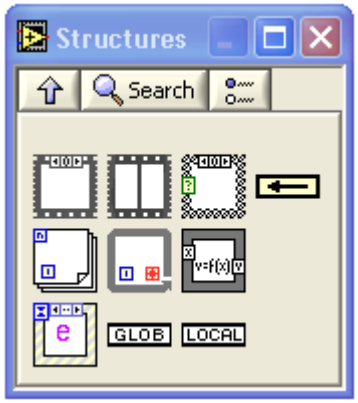


- For Loops
  - Have Iteration Terminal
  - Run According to input **N** of Count Terminal

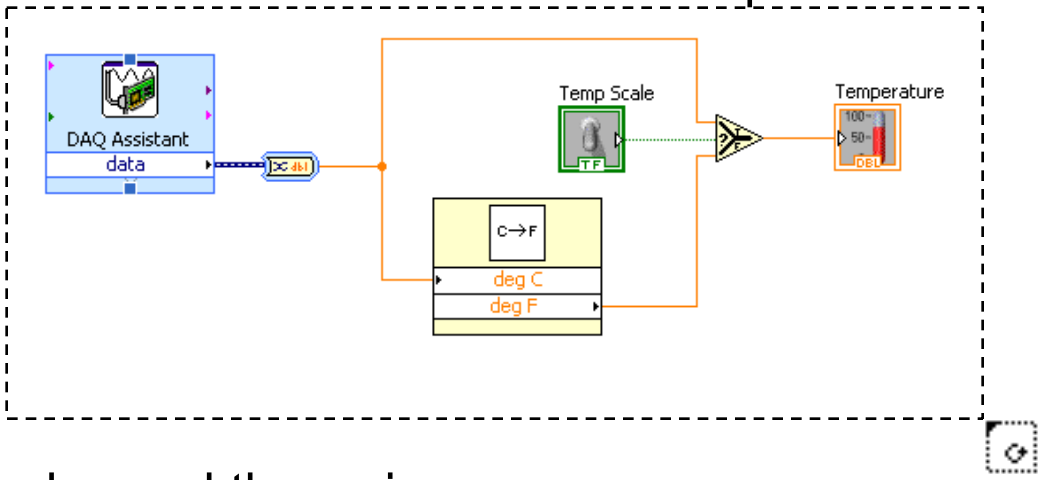


Loops (cont.)

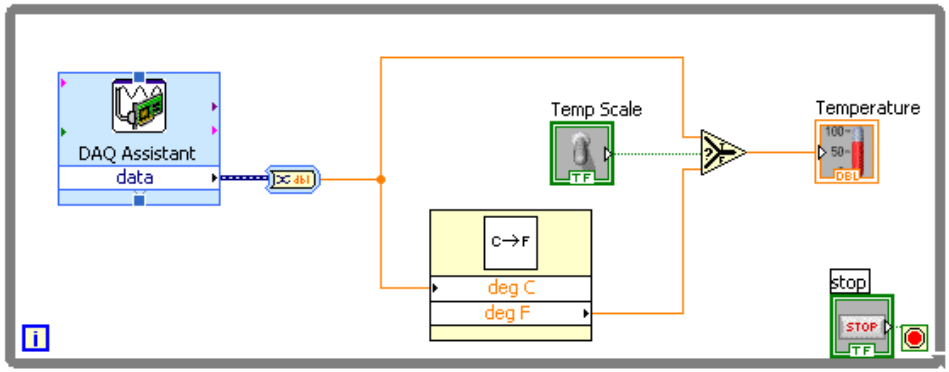
1. Select the loop



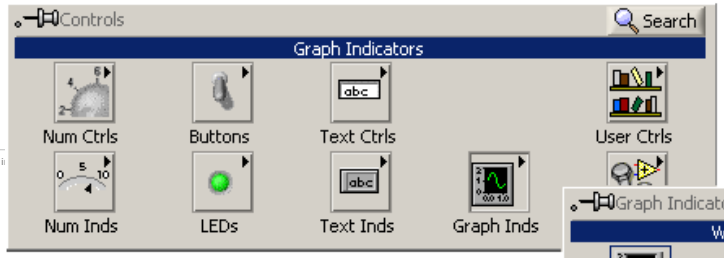
2. Enclose code to be repeated

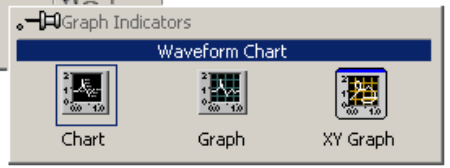


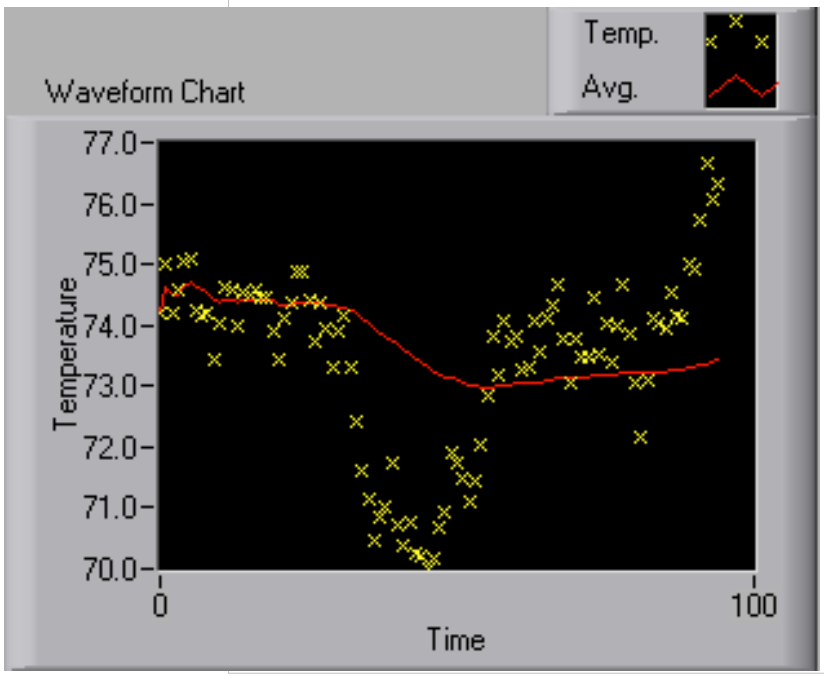
3. Drop or drag additional nodes and then wire



Charts





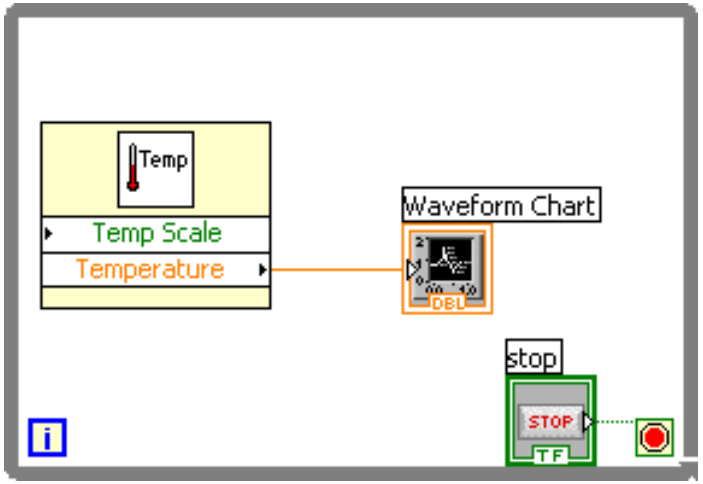


Waveform chart – special numeric indicator that can display a history of values

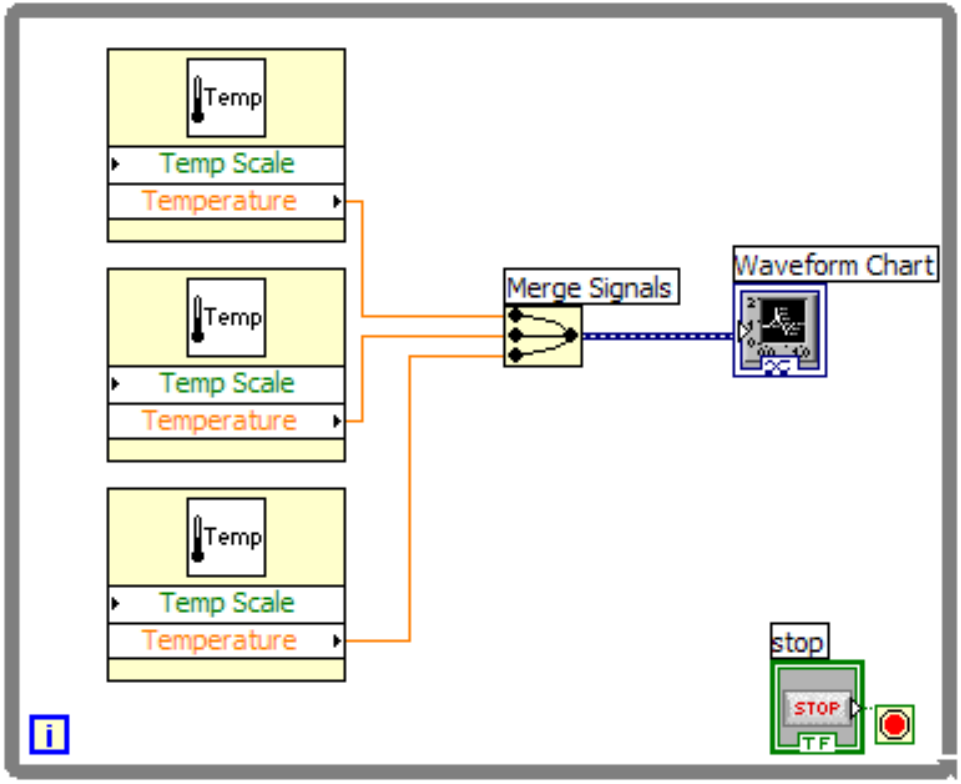
Controls >> Graph Indicators >> Waveform Chart

Wiring Data into Charts

Single Plot Charts

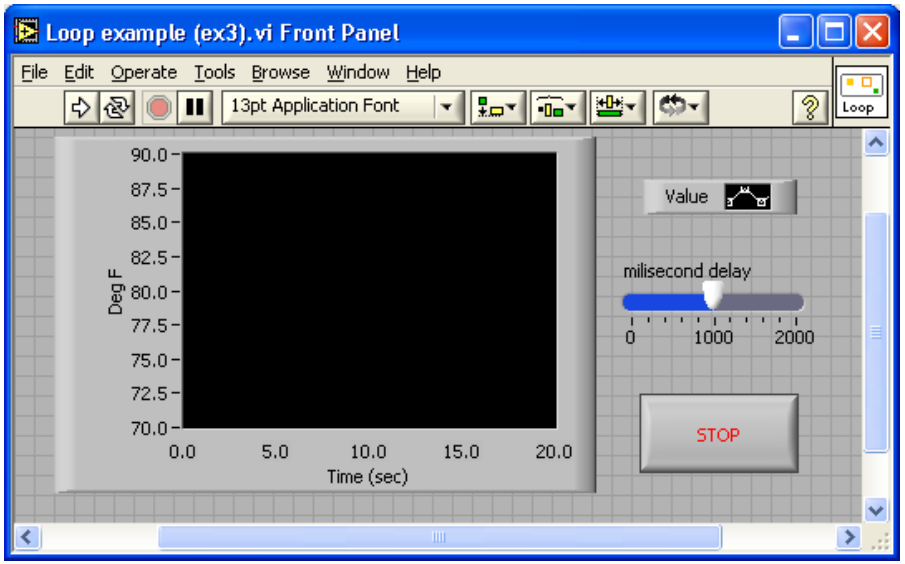


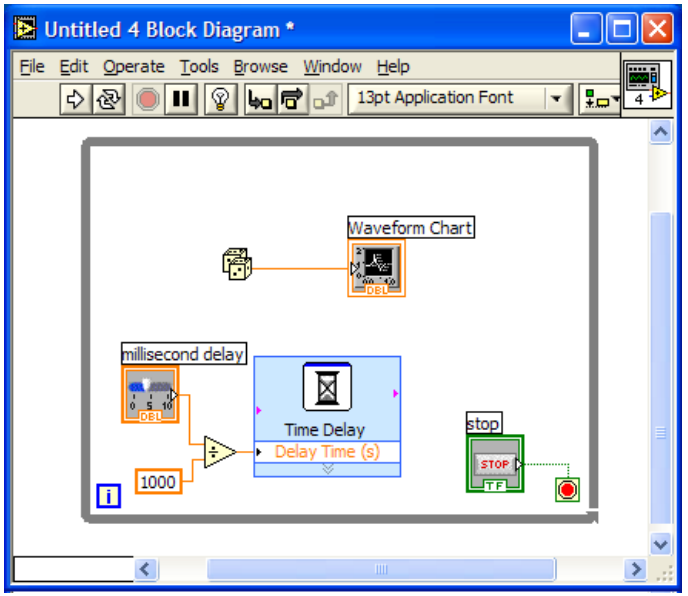
Multiplot Charts



Exercise 3 – Using loops

Students build Use a loop.vi.



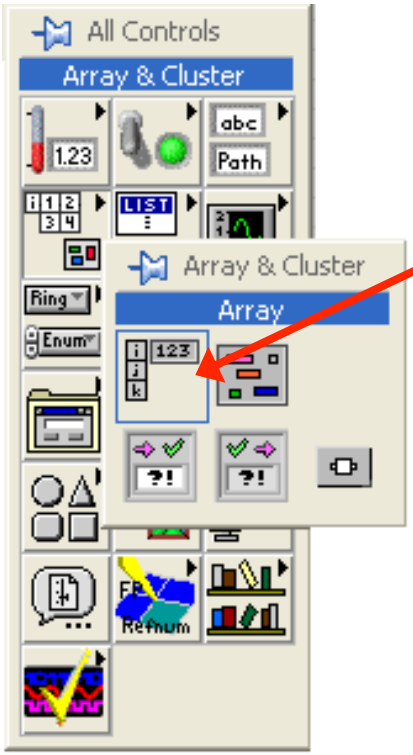


Section V – Arrays & File I/O

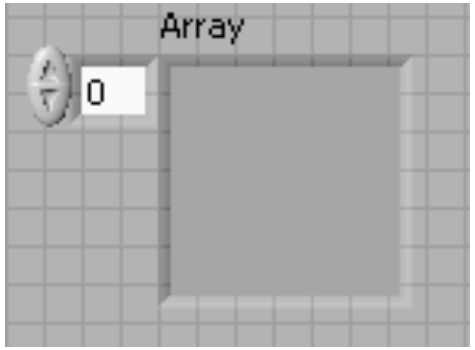
- Build arrays manually
- Have LabVIEW build arrays automatically
- Write to a spreadsheet file
- Read from a spreadsheet file

Adding an Array to the Front Panel

From the **Controls >> All Controls >> Array and Cluster** subpalette, select the **Array Shell**

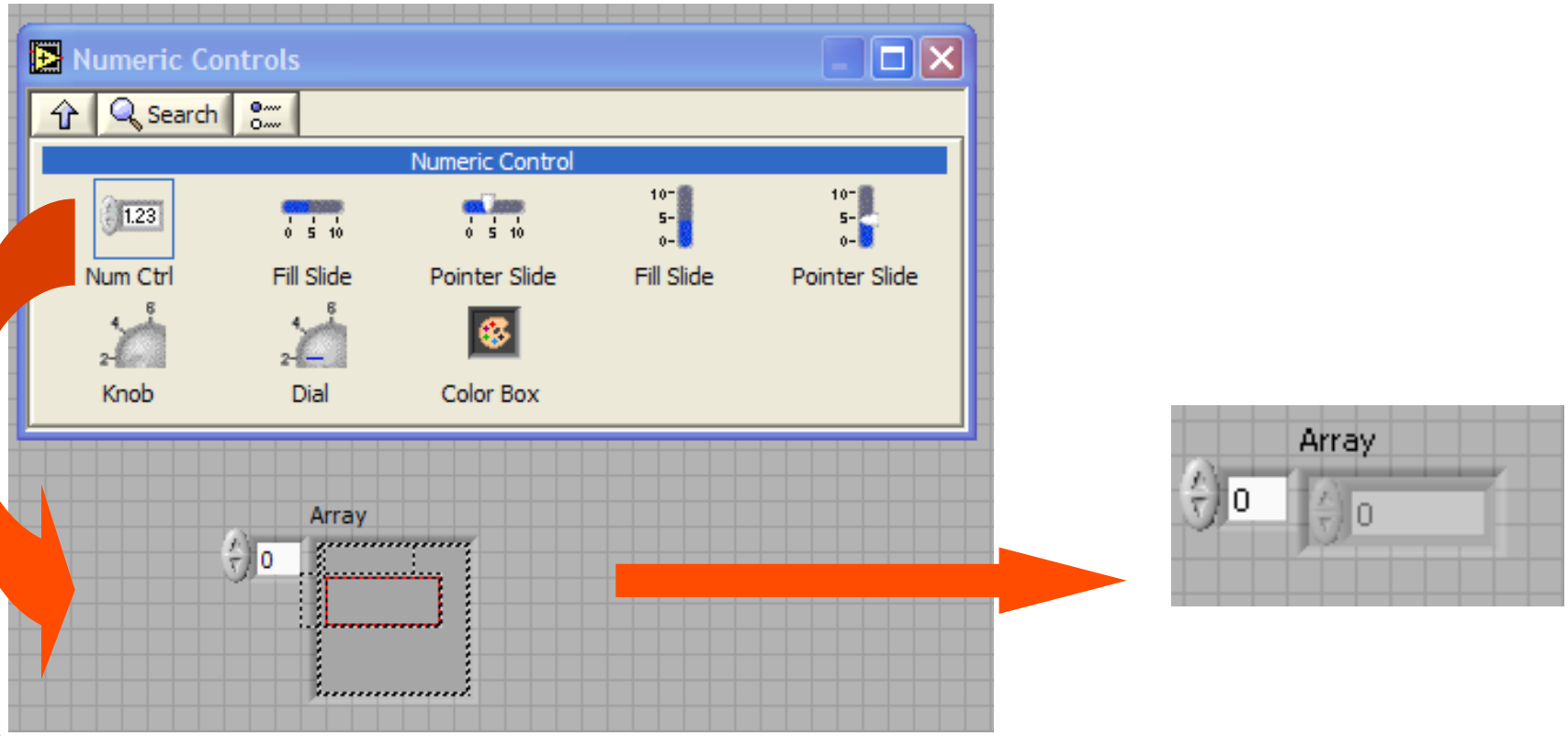


Drop it on the screen.



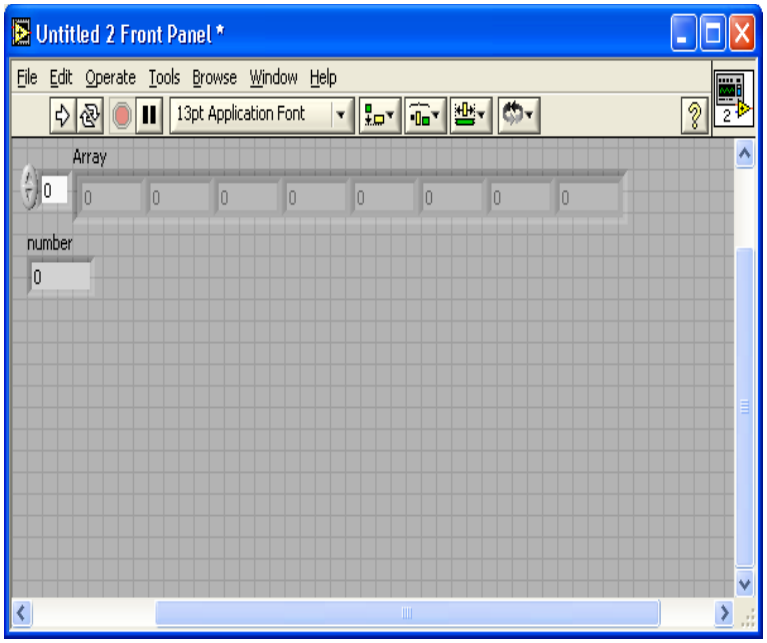
Adding an Array (cont.)

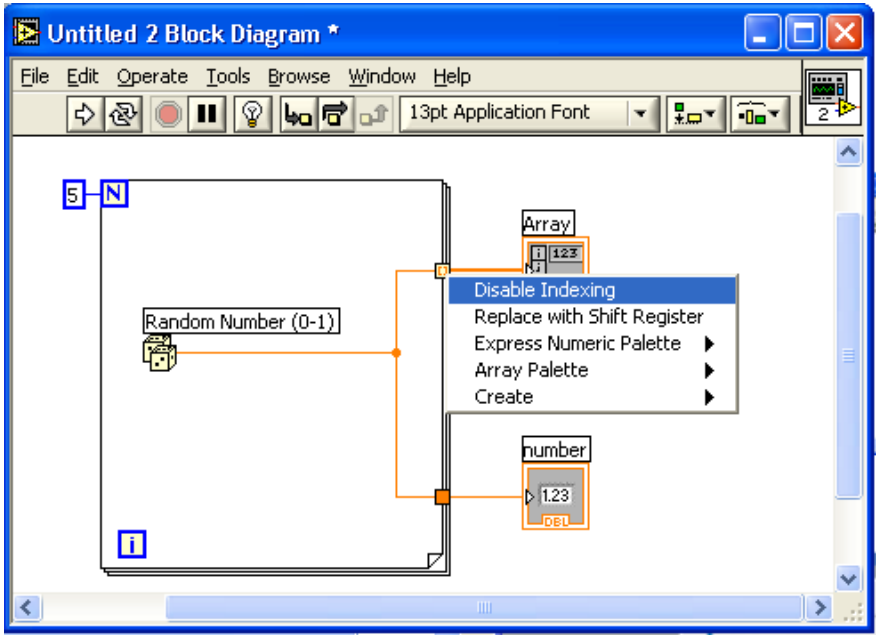
Place data object into shell (i.e. Numeric Control)



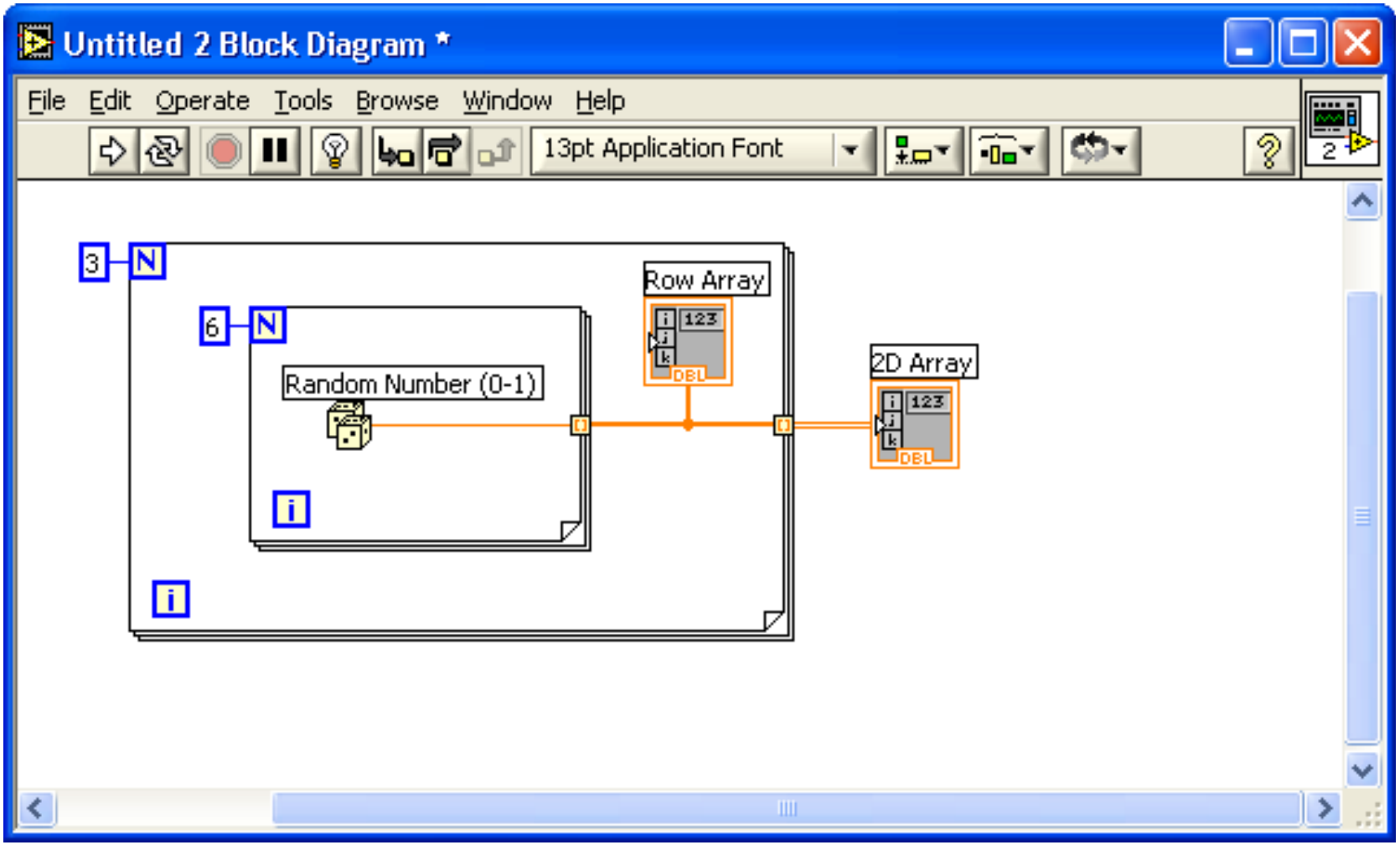
Creating an Array with a Loop

- Loops accumulate arrays at their boundaries





Creating 2D Arrays

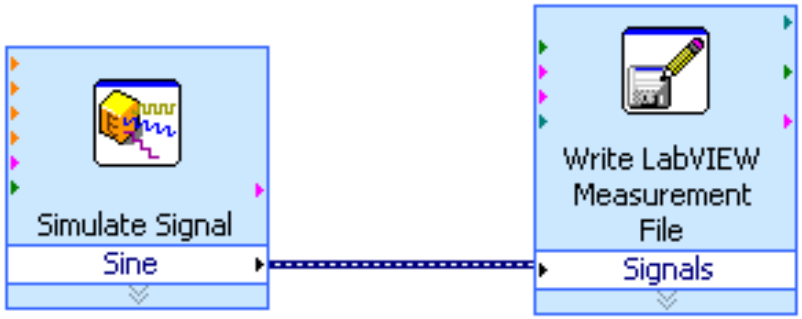


File I/O

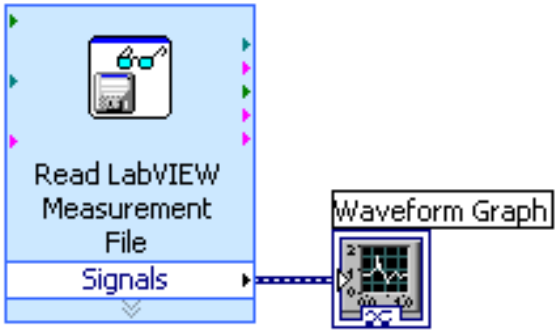
File I/O – passing data to and from files

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

Writing to LVM file

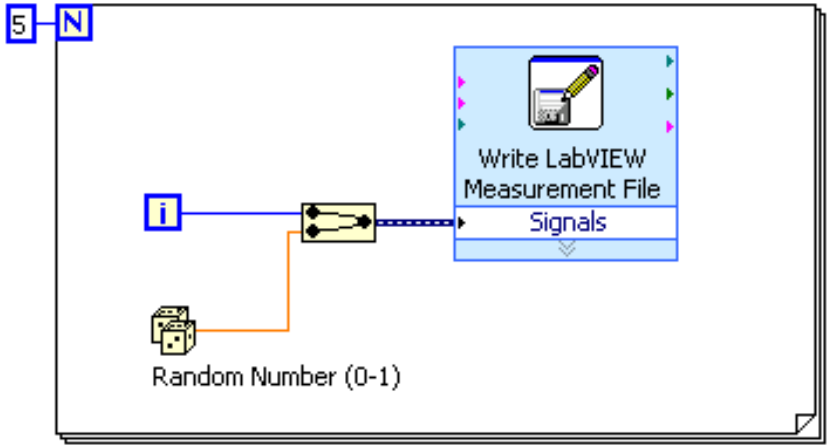


Reading from LVM file



Write LabVIEW Measurement File

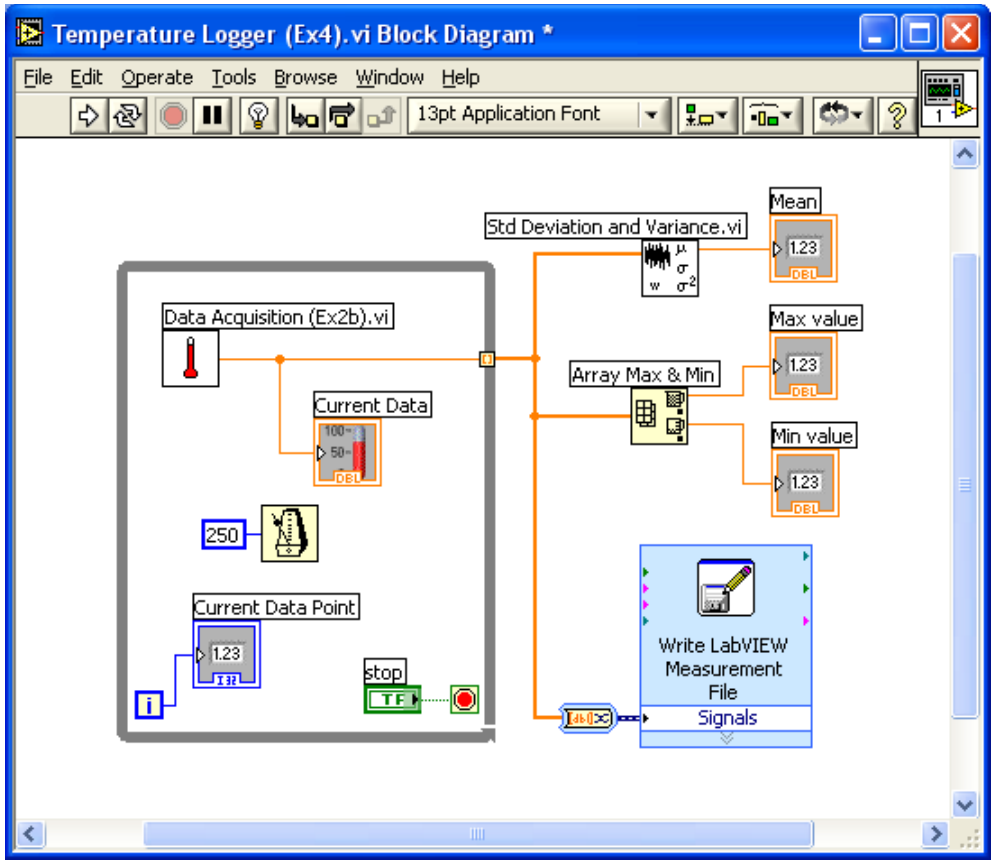
- Includes the open, write, close and error handling functions
- Handles formatting the string with either a tab or comma delimiter
- Merge Signals function is used to combine data into the dynamic data type



	A	B	C	D
1		0	0.385055	
2		1	0.23516	
3		2	0.985184	
4		3	0.177893	
5		4	0.935915	
6				
7				

Exercise 4 – Analyzing and Logging Data

Students build Temperature Logger.vi

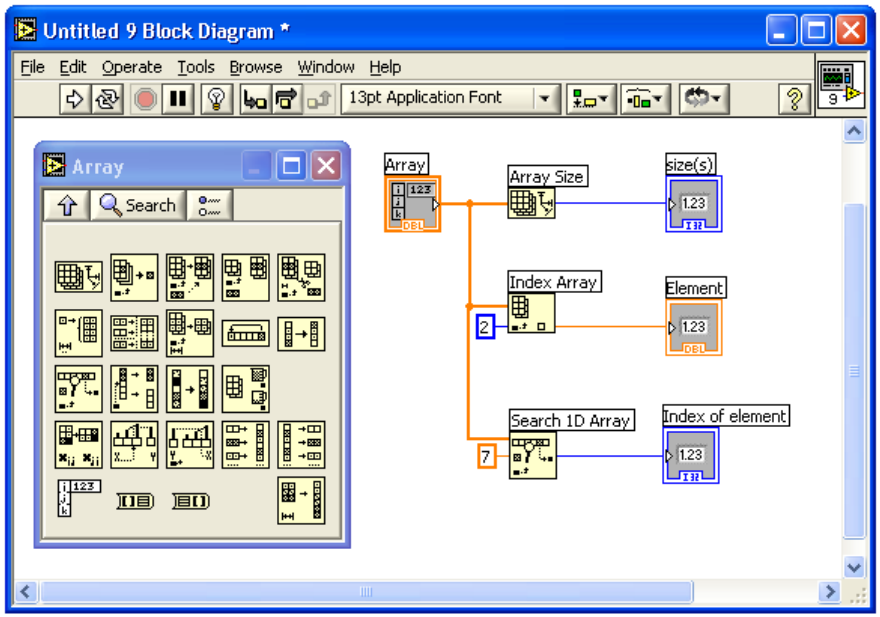


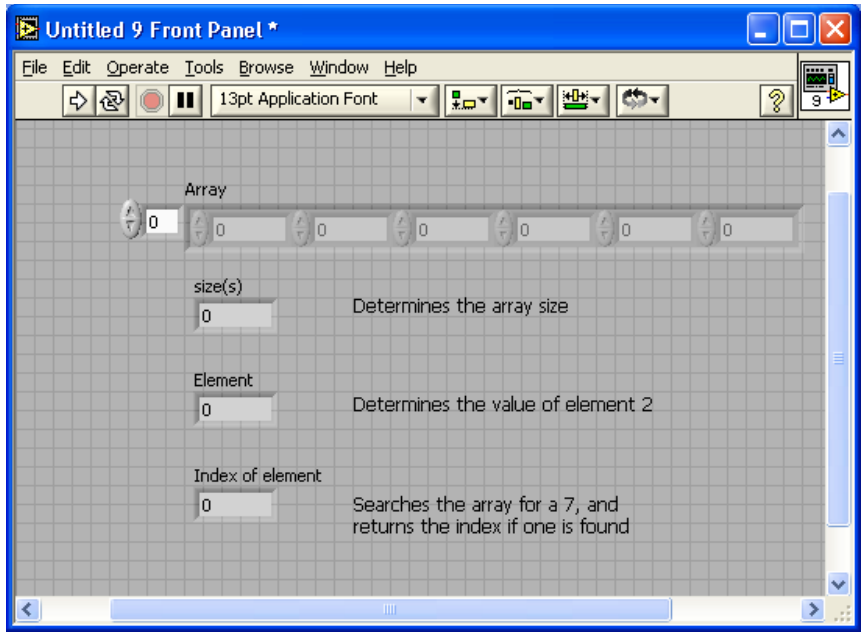
Section VI – Array Functions & Graphs

- Basic Array Functions
- Use graphs
- Create multiplots with graphs

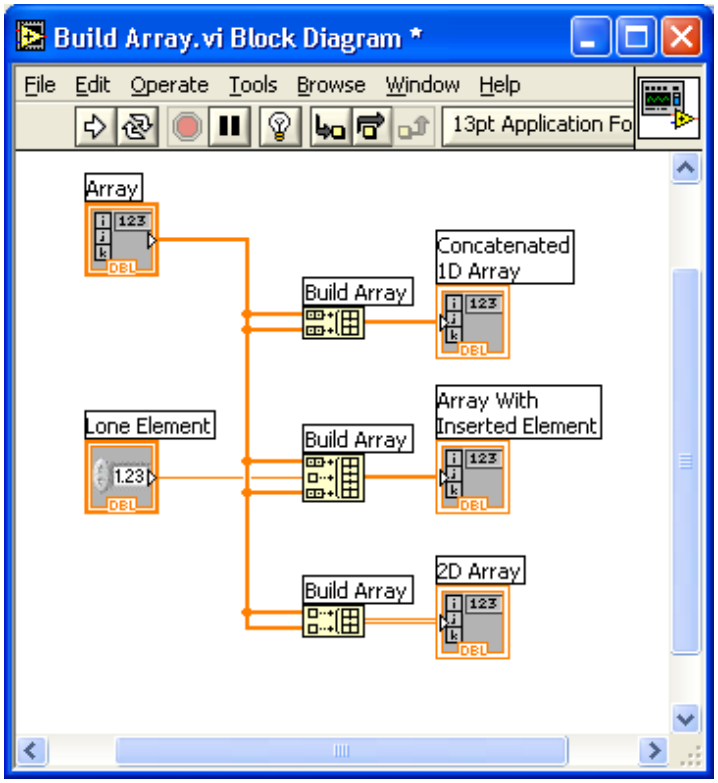
Array Functions – Basics

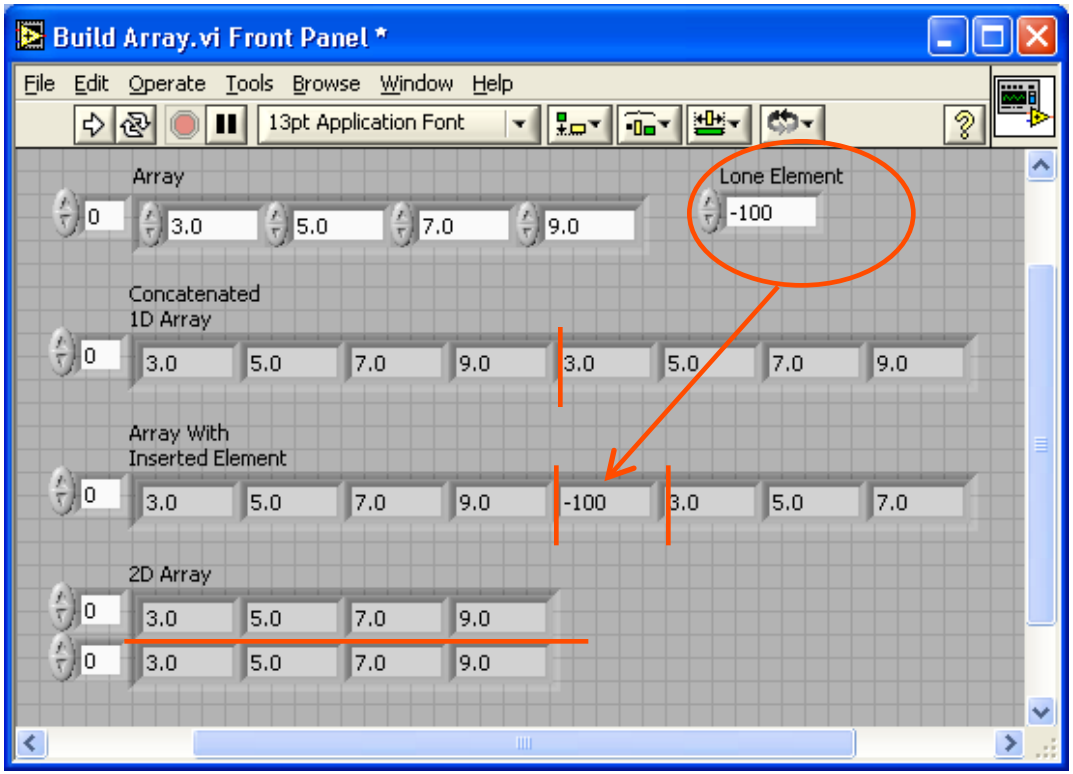
Functions >> All functions>> Array





Array Functions – Build Array





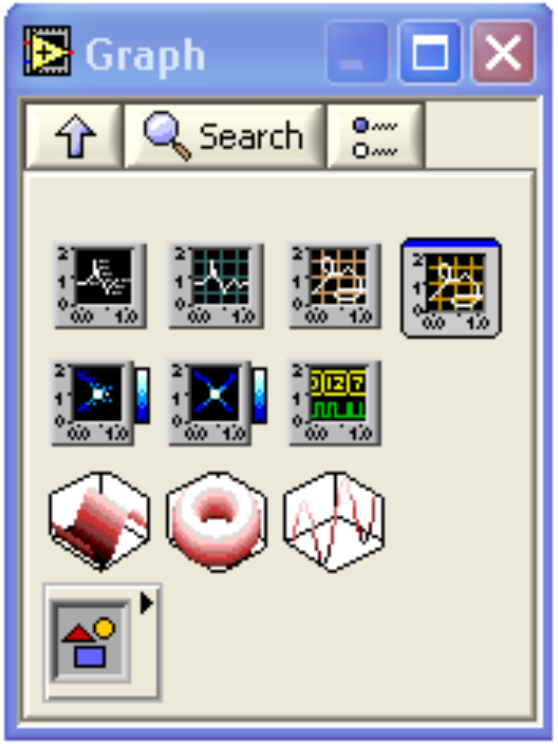
Graphs

- Selected from the Graph palette of Controls menu **Controls>>All Controls>>Graphs**

Waveform Graph – Plot an array of numbers against their indices

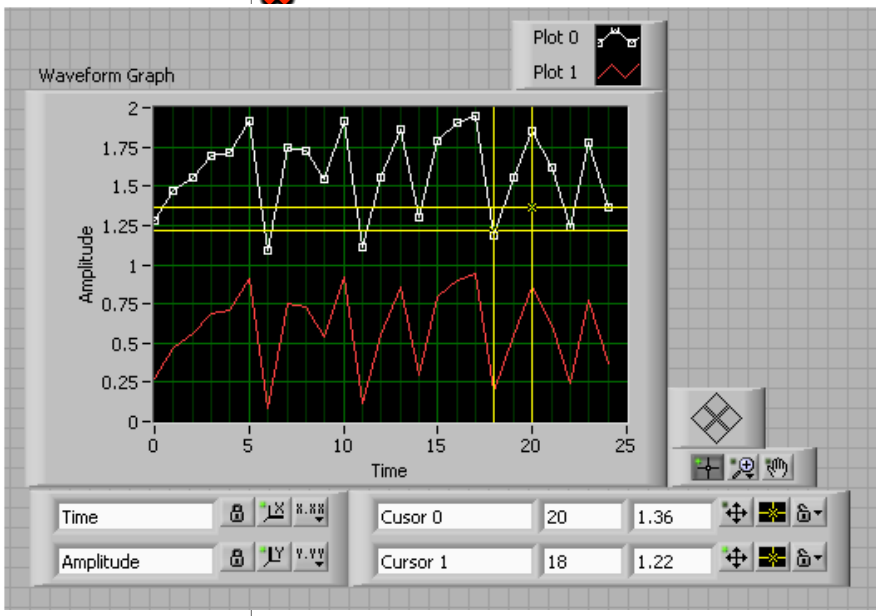
Express XY Graph – Plot one array against another

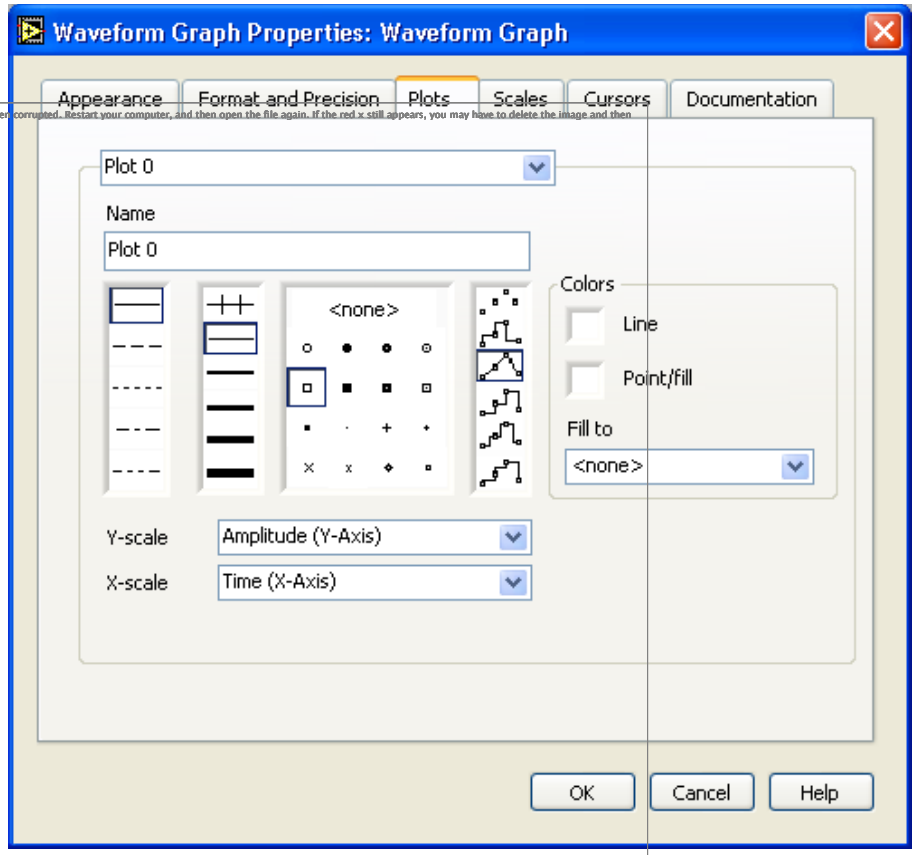
Digital Waveform Graph – Plot bits from binary data



Do Not Delete This Slide

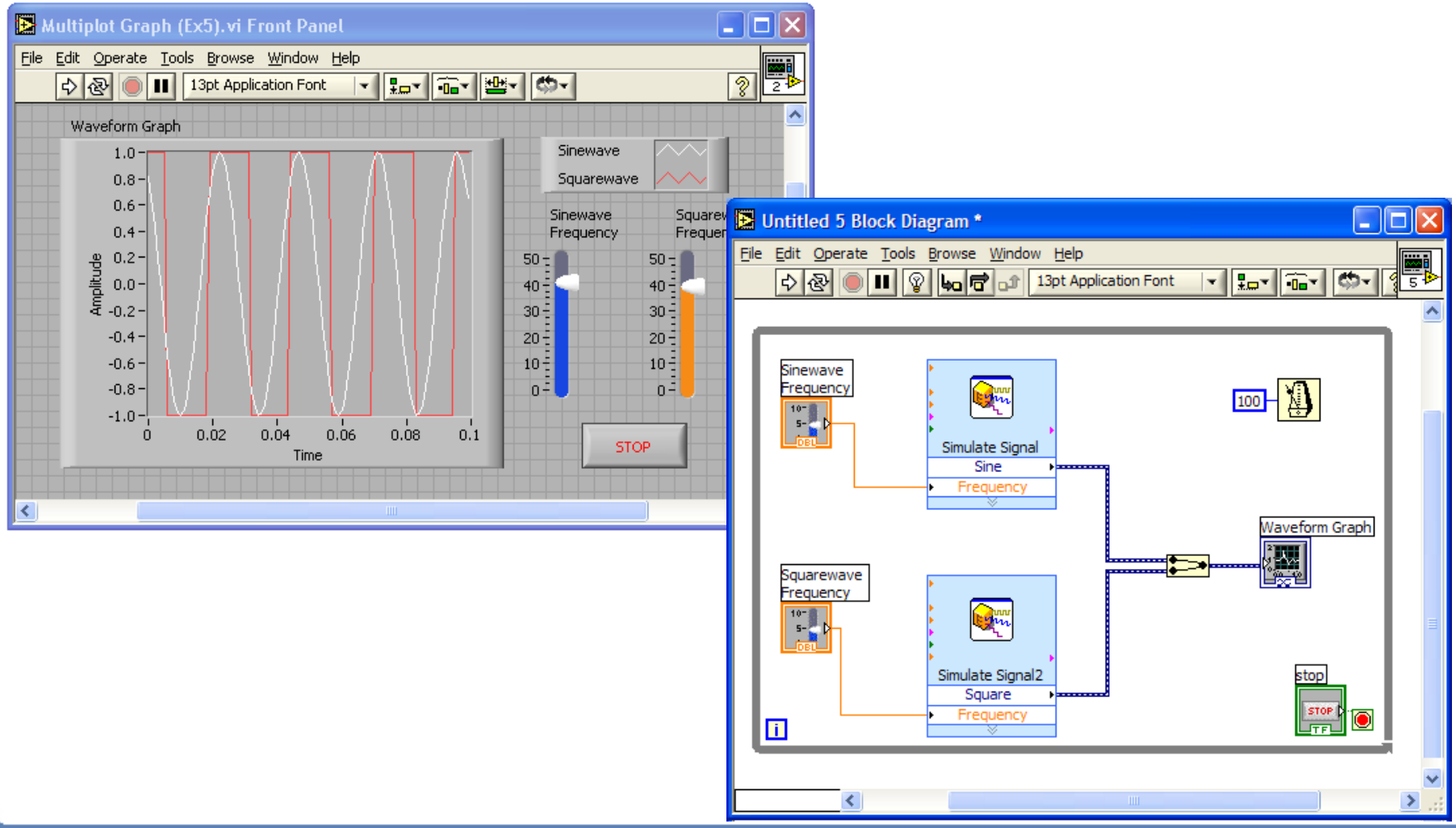
Graphs





Right-Click on the Graph and choose Properties to Interactively Customize

Exercise 5 – Using Waveform Graphs

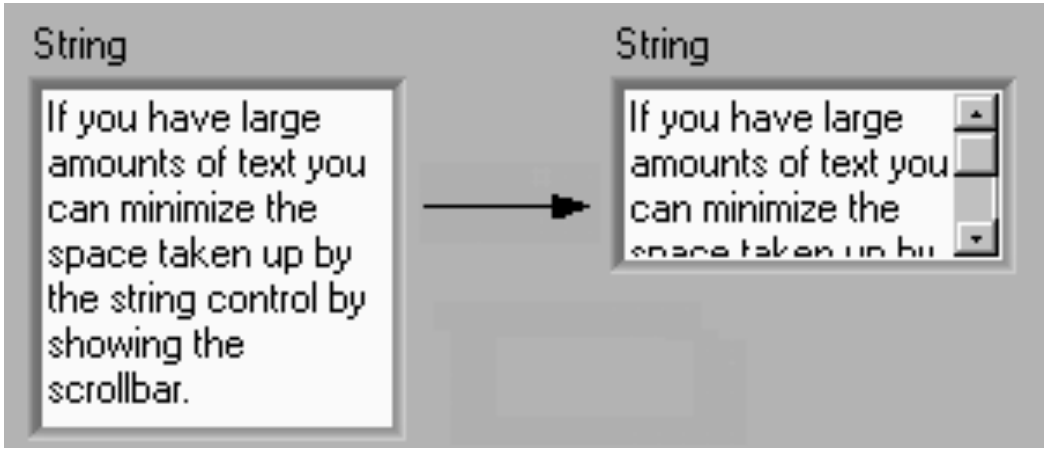


Section VII – Strings, Clusters, & Error Handling

- Strings
- Creating Clusters
- Cluster Functions
- Error I/O

Strings

- A string is a sequence of displayable or nondisplayable characters (ASCII)
- Many uses – displaying messages, instrument control, file I/O
- String control/indicator is in the **Controls »Text Control** or **Text Indicator**



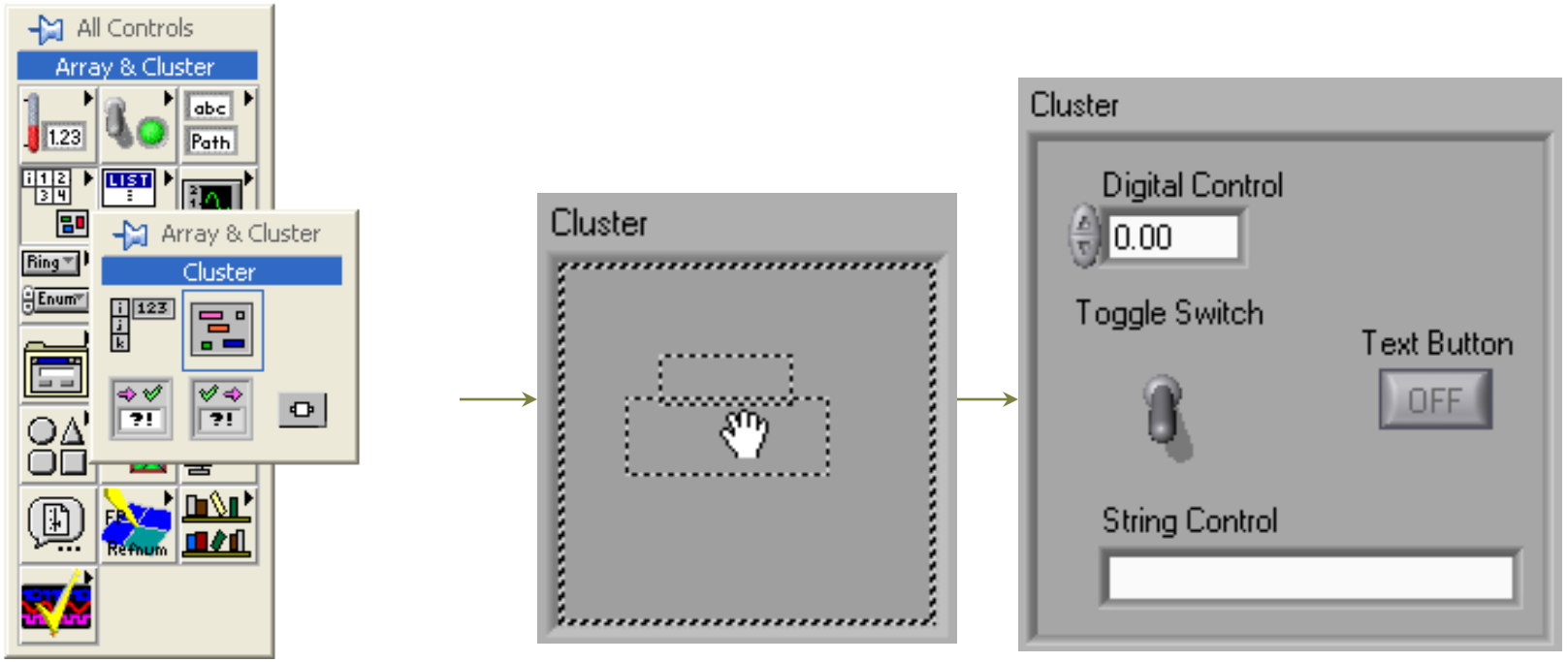
Clusters

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

Creating a Cluster

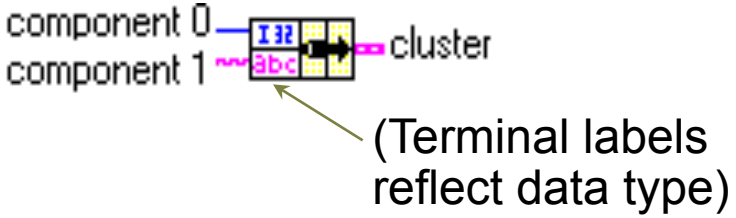
1. Select a **Cluster** shell
2. Place objects inside the shell

Controls >> All Controls >> Array & Cluster



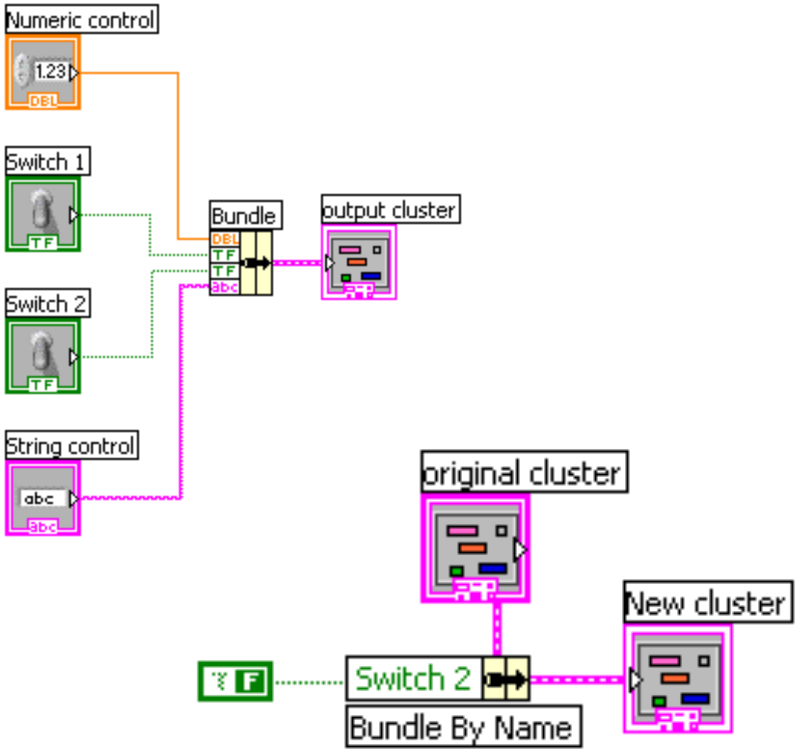
Cluster Functions

- In the **Cluster** subpalette of the **Functions>>All functions** palette
- Can also be accessed by right-clicking on the cluster terminal

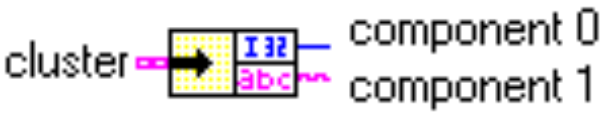


Bundle

Bundle By Name

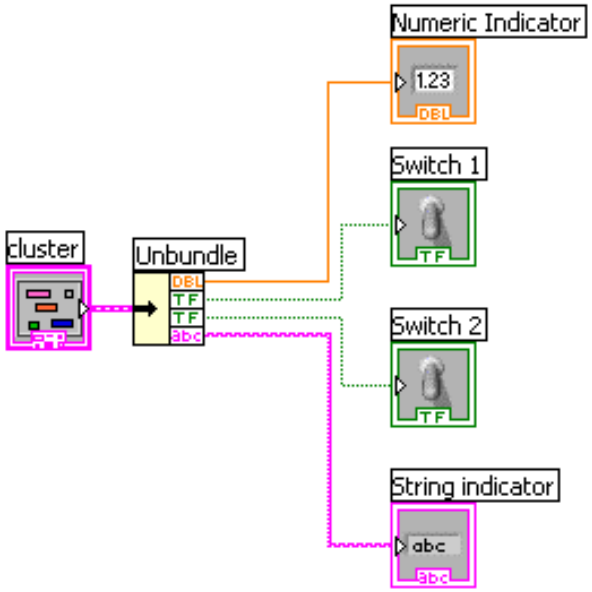


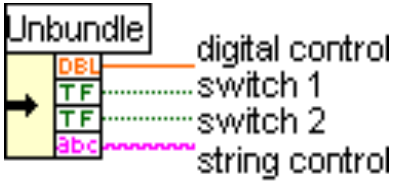
Cluster Functions

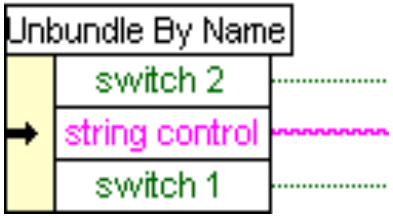


Unbundle

Unbundle By Name



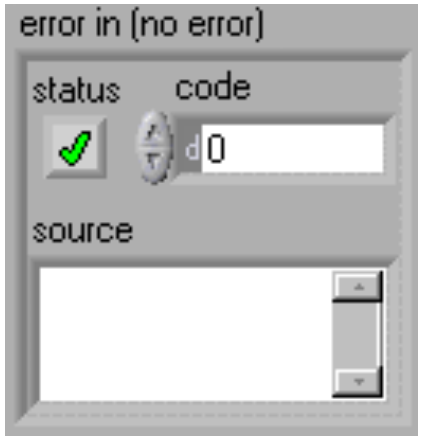


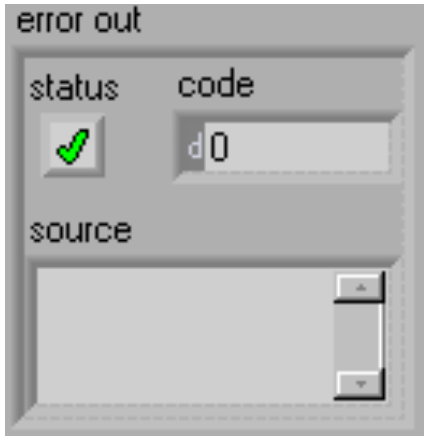


Unbundled cluster in the diagram

Error Clusters

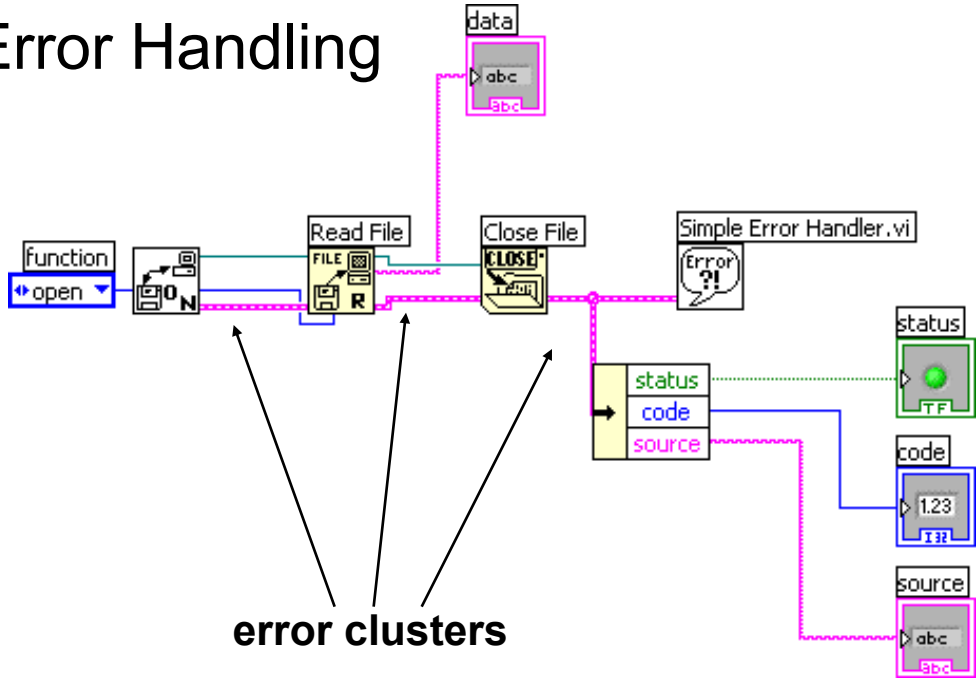
- Error cluster contains the following information:
  - Boolean to report whether error occurred
  - Integer to report a specific error code
  - String to give information about the error





Error Handling Techniques

- Error information is passed from one subVI to the next
- If an error occurs in one subVI, all subsequent subVIs are not executed in the usual manner
- Error Clusters contain all error conditions
- Automatic Error Handling

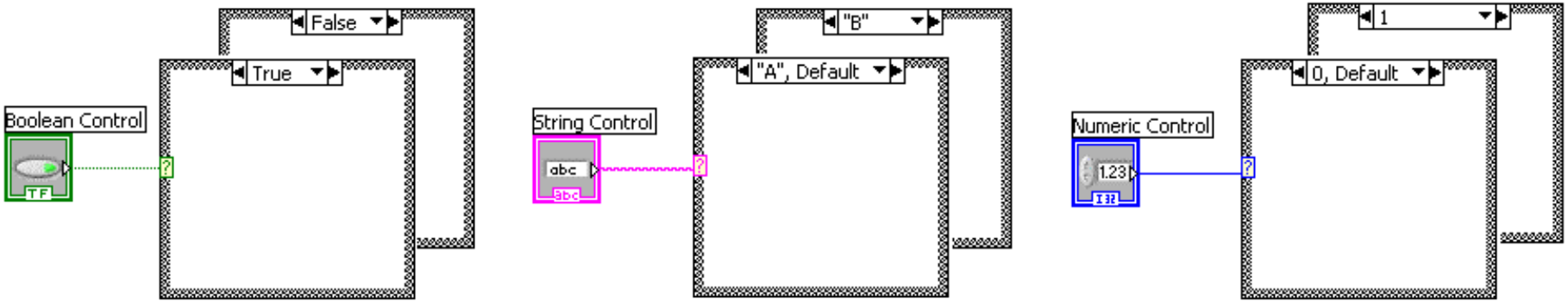


Section VIII - Case & Sequence Structures, Formula Nodes

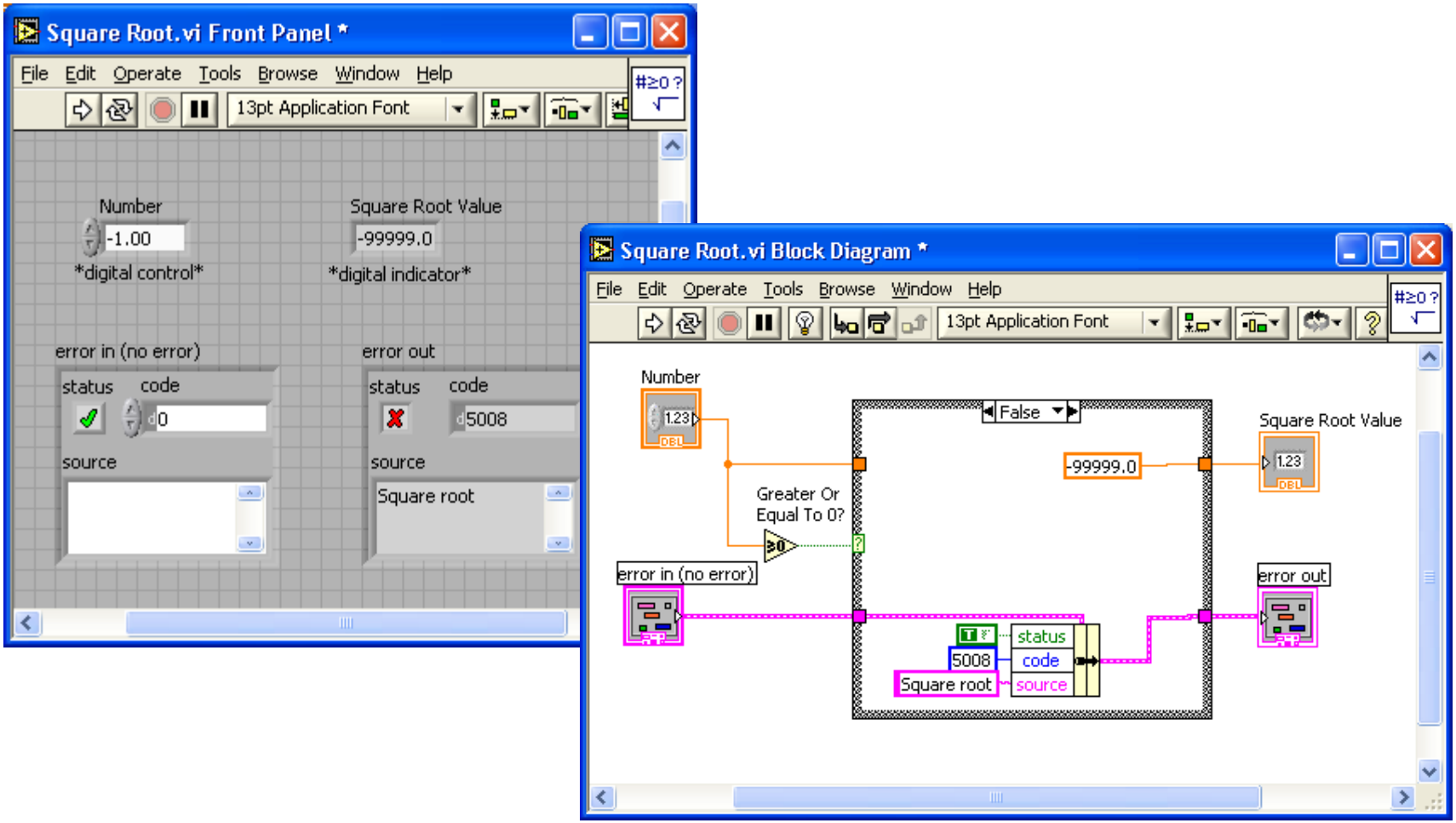
Case Structures

- In the Structures subpalette of Functions palette
- Enclose nodes or drag them inside the structure
- Stacked like a deck of cards, only one case visible

Functions >> Execution control

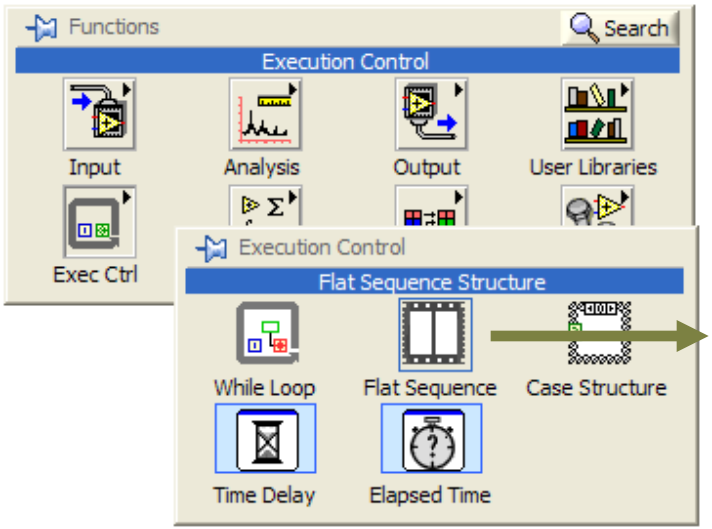


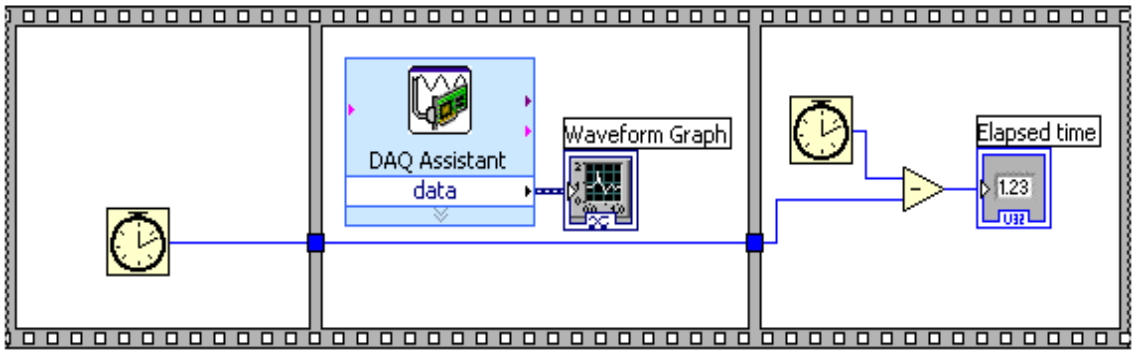
Exercise 6 – Error Clusters & Handling



Sequence Structures

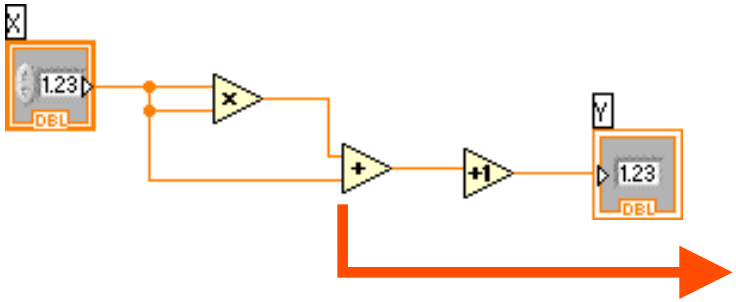
- In the **Execution Control** subpalette of Functions palette
- Executes diagrams sequentially
- Right-click to add new frame

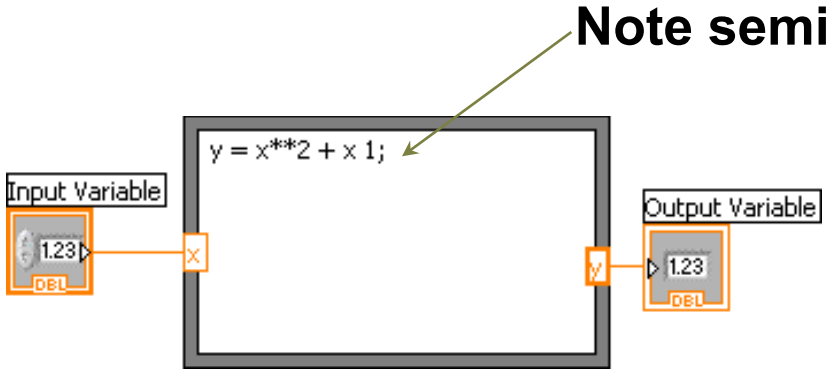




Formula Nodes

- In the Structures subpalette
- Implement complicated equations
- Variables created at border
- Variable names are case sensitive
- Each statement must terminate with a semicolon (;)
- Context Help Window shows available functions



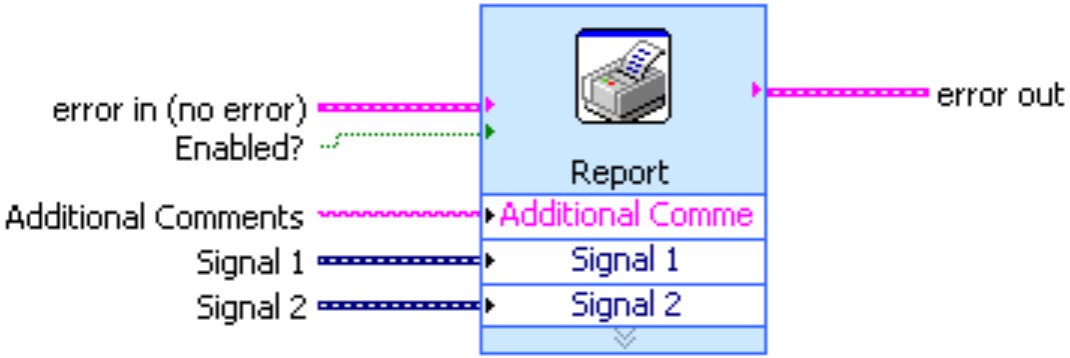


Section IX – Printing & Documentation

- Print From File Menu to Printer, HTML, Rich Text File
- Programmatically Print Graphs or Front Panel Images
- Document VIs in VI Properties » Documentation Dialog
- Add Comments Using Free Labels on Front Panel & Block Diagram

Printing

- **File » Print**… Gives Many Printing Options
  - Choose to Print Icon, Front Panel, Block Diagram, VI Hierarchy, Included SubVIs, VI History
- Print Panel.vi (Programmatically Prints a Front Panel)
  - **Functions » All Functions » Application Control**
- Generate & Print Reports (**Functions » Output » Report**)



Documenting VIs

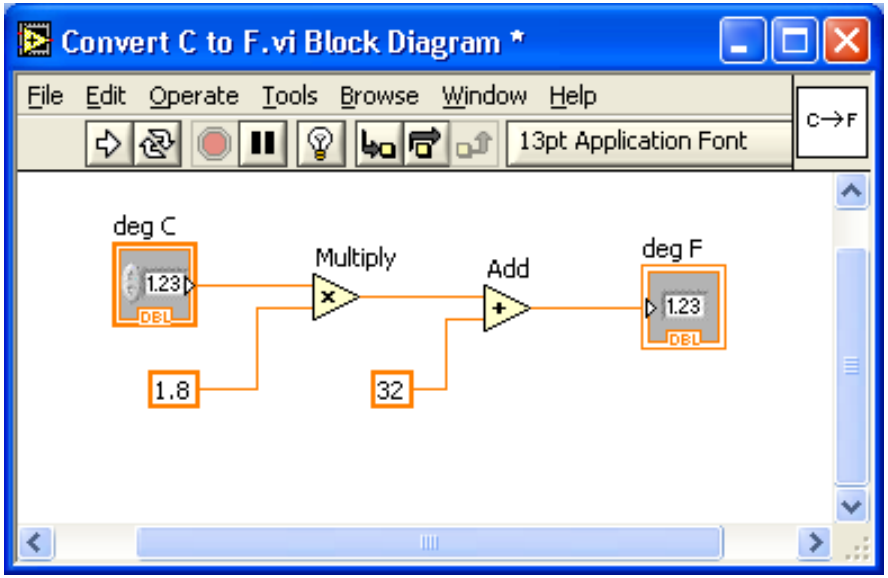
- VI Properties » Documentation
  - Provide a Description and Help Information for a VI
- VI Properties » Revision History
  - Track Changes Between Versions of a VI
- Individual Controls » Description and Tip…
  - Right Click to Provide Description and Tip Strip
- Use Labeling Tool to Document Front Panels & Block Diagrams

Section X – Basic Programming Architecture

- Simple VI Architecture
- General VI Architecture
- State Machine Architecture

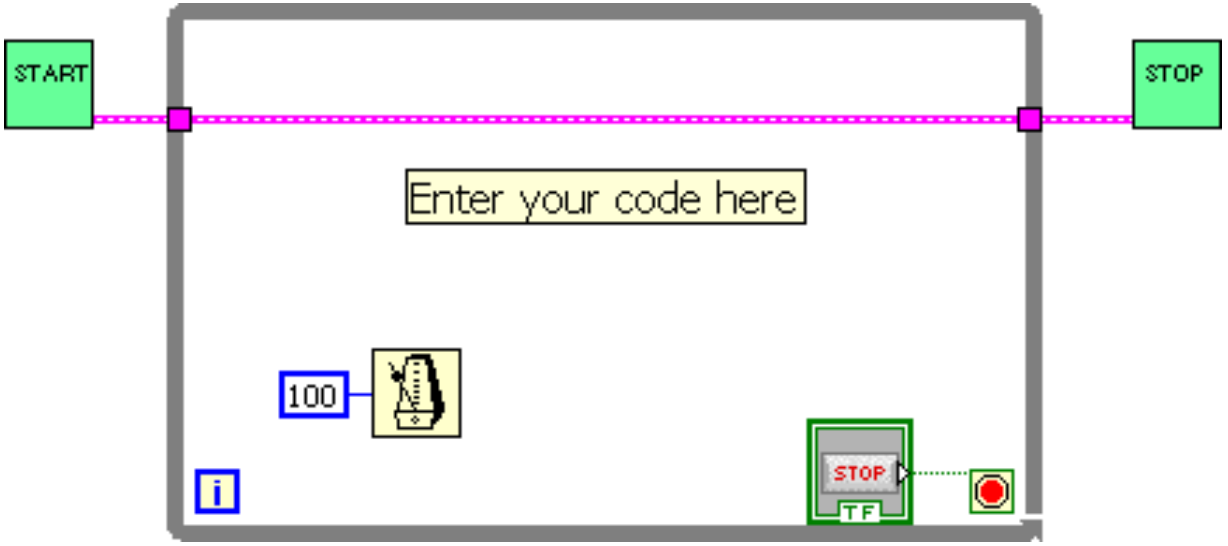
Simple VI Architecture

- Functional VI that produces results when run
  - No “start” or “stop” options
  - Suitable for lab tests, calculations
- Example: Convert C to F.vi



General VI Architecture

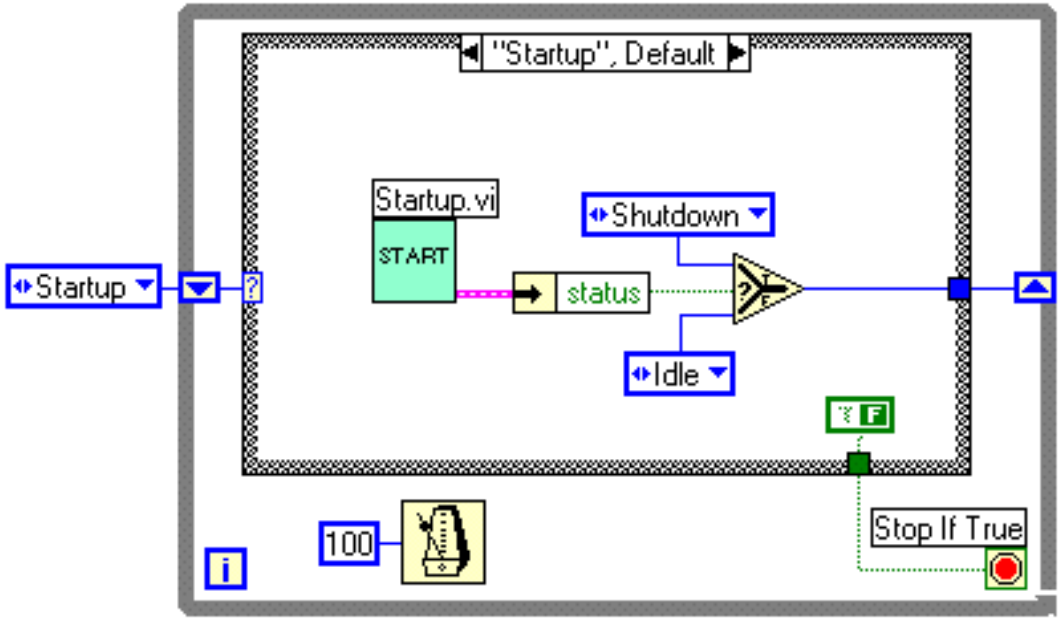
- Three Main Steps
  - Startup
  - Main Application
  - Shutdown



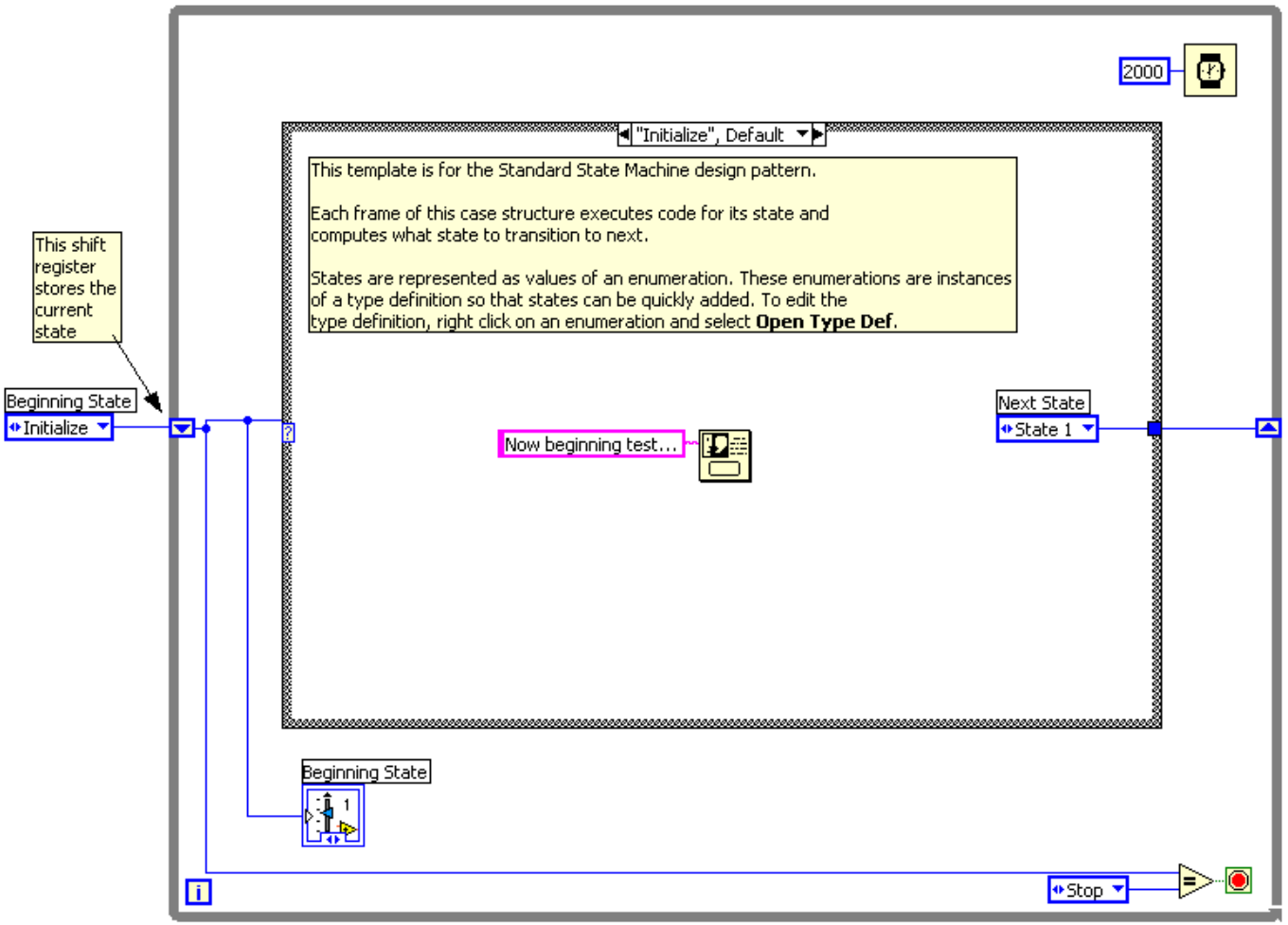
State Machine Architecture

- Advantages
  - Can go from any state from any other
  - Easy to modify and debug
- Disadvantages
  - Can lose events if two occur at the same time

States:
0: Startup
1: Idle
2: Event 1
3: Event 2
4: Shutdown



Exercise 7 – Simple State Machine



Do Not Delete This Slide

Section XI – Remote Front Panels

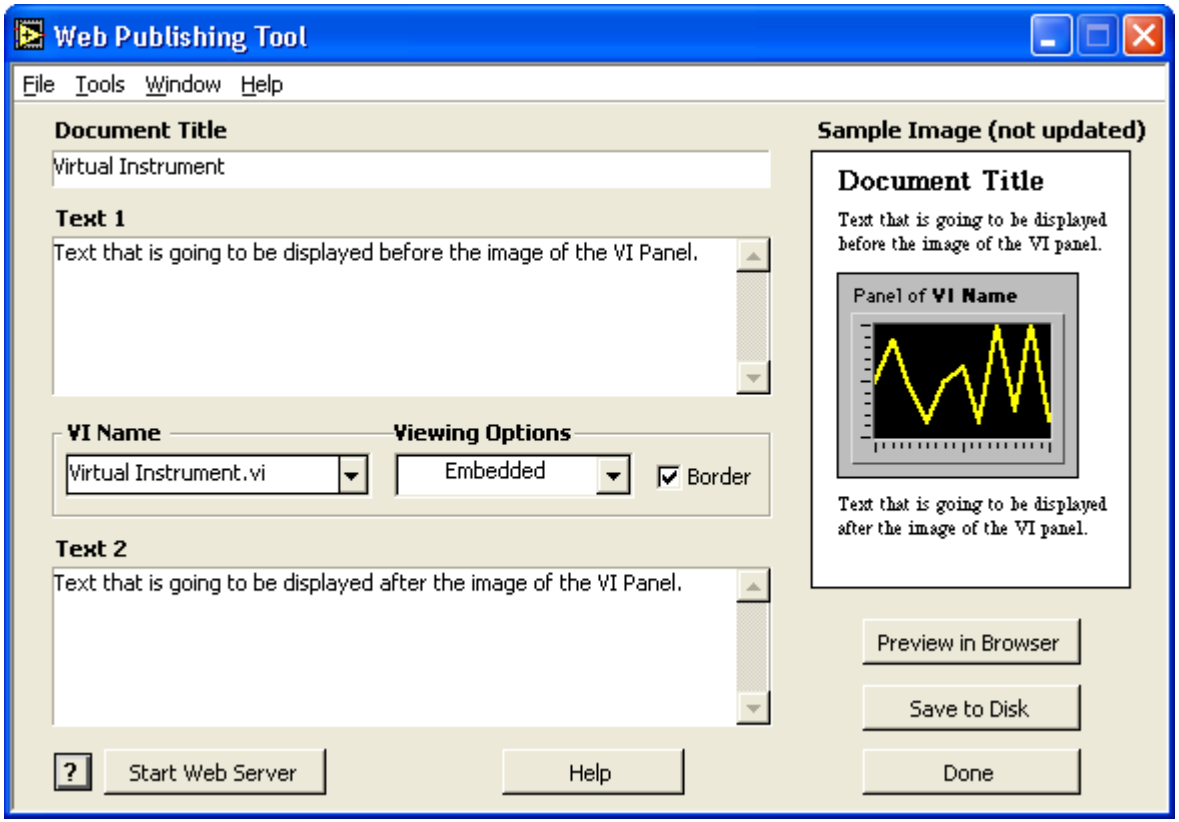
- View & Control LabVIEW Front Panels from a Web Browser
- Requires no programming
- Remote clients see “live” front panel updates
- Multiple clients can view the same panel simultaneously
- Only one client can control the front panel at a time

Remote Panel Web Publishing Tool

- **Tools » Web Publishing Tool…**

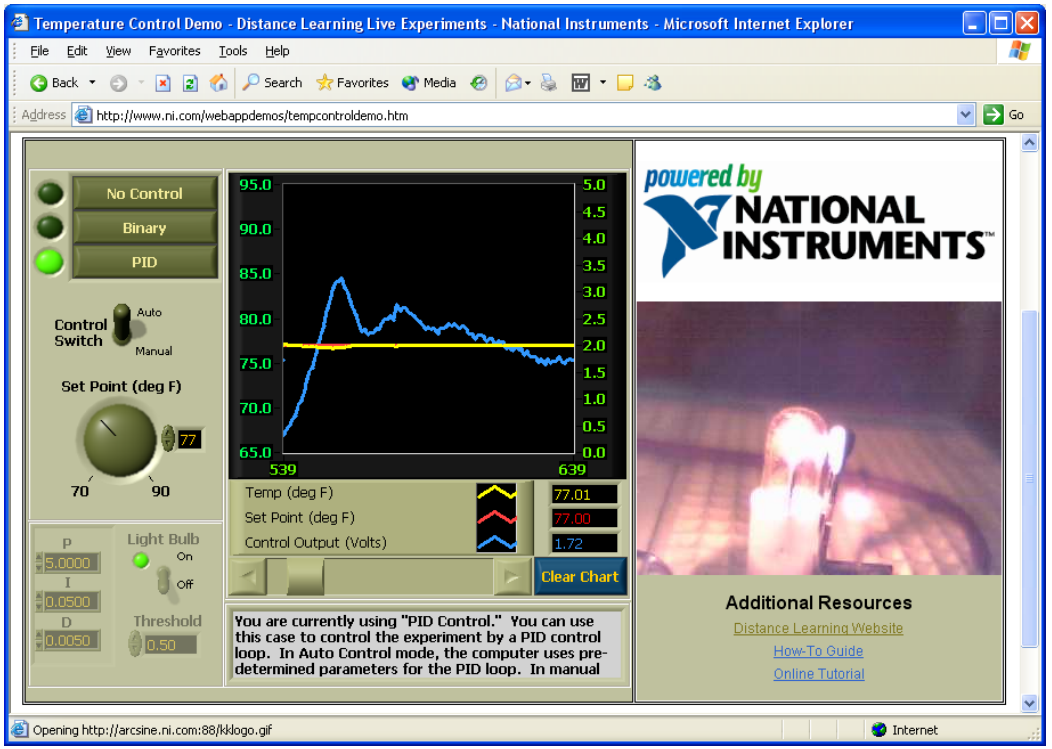
- Click Save to Disk and VI is embedded into an HTML file

- After file is saved, it can be reopened and customized in any HTML editor



Remote Front Panels - Resources

- NI Developer Zone (zone.ni.com)
  - Search for Remote Front Panel
  - Tutorials & Instructions Are Available for Download
  - Information on Incorporating Web Cameras into Remote Panel Applications



Section XII – Additional Topics

- Property Nodes
- Local Variables
- Global Variables
- DataSocket
- Binary File I/O

Where Do I Go From Here?

- Example programs (Help» Find Examples…)
- LabVIEW Student Edition (www.ni.com/labviewse)
- Web resources (ni.com)
  - NI Developer Zone (zone.ni.com)
  - Application Notes
  - Info-labview newsgroup (www.info-labview.org/)
  - Instrument Driver Library (www.ni.com/idnet)