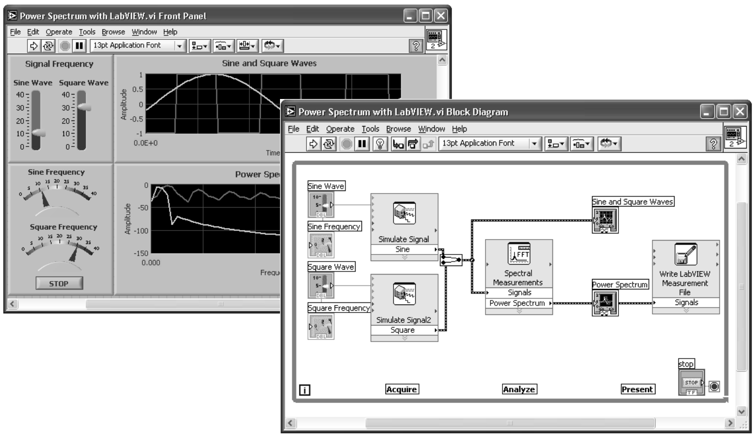
Virtual Instrumentation With LabVIEW

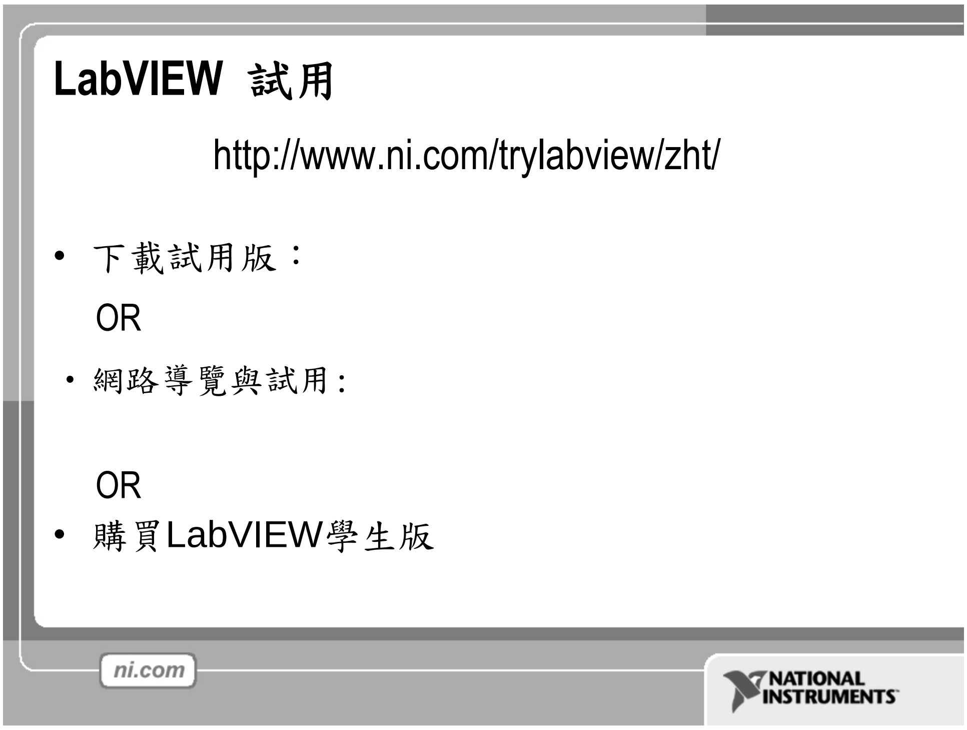


LabView programming 練習

工綜六樓計算機室電腦

- LabView 8.0（目前最新版Labview 2010）
- Interactive guide to LabView

請自行前往使用。



LabVIEW 試用

http://www.ni.com/trylabview/zht/

- 下載試用版：

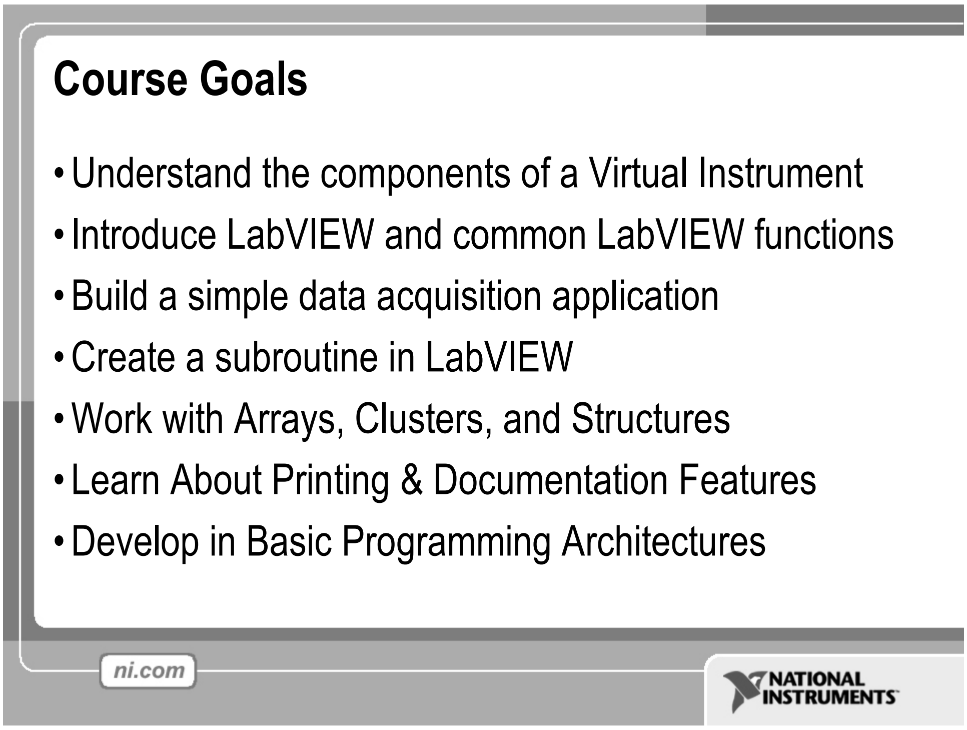
OR

- 網路導覽與試用：

OR

- 購買LabVIEW學生版

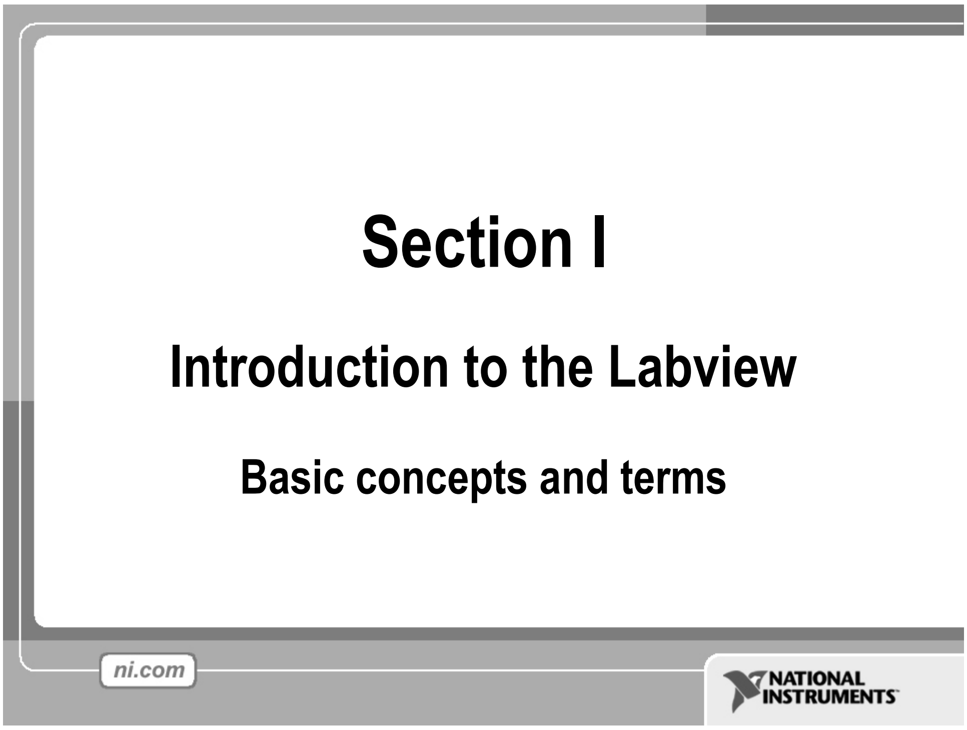
ni.com



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

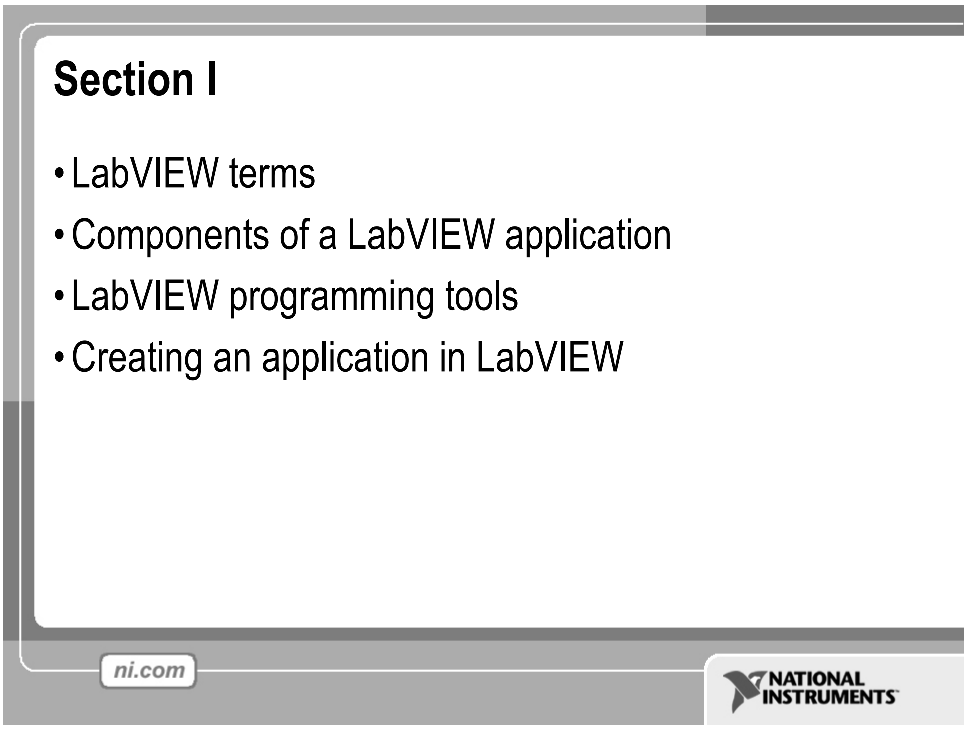
ni.com



Section I

Introduction to the Labview

Basic concepts and terms



Section I

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

Text-based programming

- Traditional computer programming involves setting down a list of tasks for the computer to execute in the given sequential order.
- Each instruction is executed in the order of appearance.
- This may not be efficient as some instruction may not need previous data.

```
1.   Add A to B
2.   Add C to the Sum of A and B
3.   Divide Sum of A,B, and C by 3
4.   Subtract A from C
```

- Figure 1. A Sequence of Instructions in “**Control flow**” programming.

Graphical programming

- LabVIEW programming consists of drawing pictures that specify the operations and the data dependencies.
- The computer can execute the instructions in any order that protects the data dependencies.
- For most people, pictures are much easier to understand than a list of instructions.

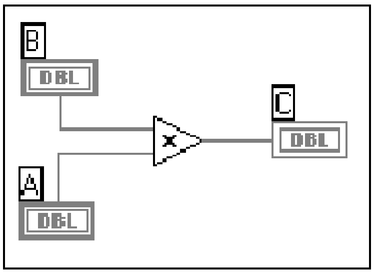


Figure 2. Program for Multiplying Two Numbers and Displaying the Result

- A LabVIEW program, called a virtual instrument (VI), is a two-window system. The code is in one window and the user interface (inputs and outputs) appears in a separate window. The program window is the block diagram window, and the user inputs and outputs are in the front panel window. Figure 2 shows a sample program that would appear in the block diagram window. The numbers are entered into the computer and displayed in the front panel window shown in Figure 3. The two boxes on the left (labeled **A** and **B**) are controls, and the box on the right (labeled **C**) is the output or indicator. (The **X** and **=** are only displays showing the operation of the VI and not inputs or outputs.) The three boxes are associated with like labeled boxes in the diagram window shown in Figure 2.

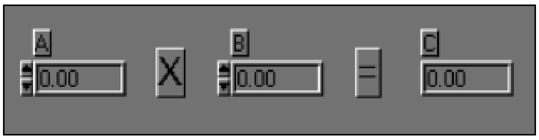
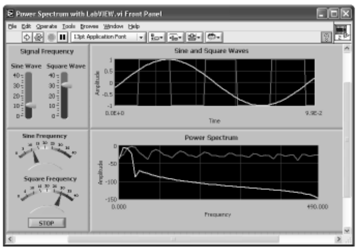


Figure 3. Front Panel for a Two-Number Multiplication Program

LabVIEW Programs Are Called Virtual Instruments (VIs)

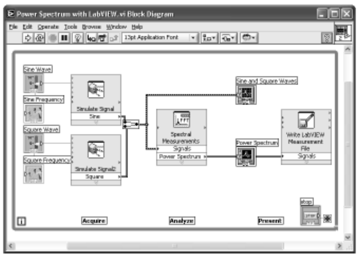


Front Panel (user interface, UI)
- **Controls = Inputs**
- **Indicators = Outputs**

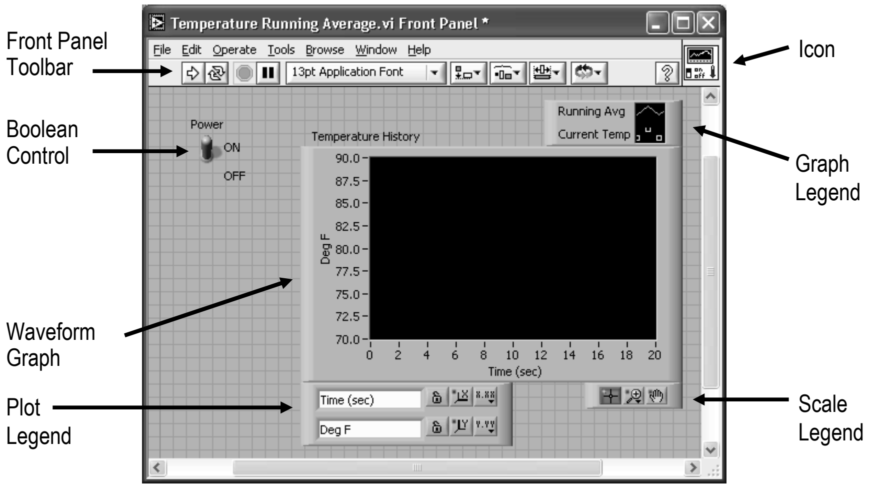
Click Window to expose

Block Diagram (graphical code)

- **Accompanying "program" for front panel**
- **Components "wired" together**
- **Data travel on wires from controls through functions to indicators.**
- **Blocks execute by Dataflow.**



VI Front Panel



Front Panel Toolbar

Boolean Control

Waveform Graph

Plot Legend

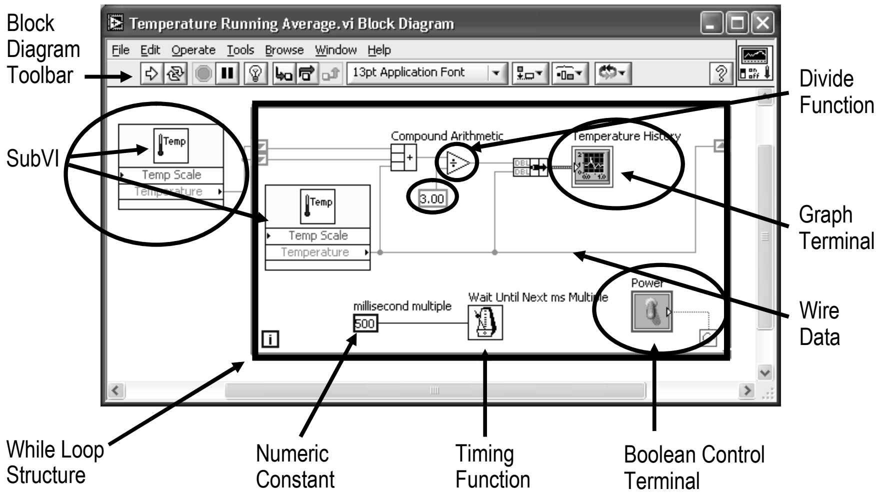
Icon

Graph Legend

Scale Legend

VI Block Diagram

Nodes



Block Diagram Toolbar

SubVI

While Loop Structure

Numeric Constant

Timing Function

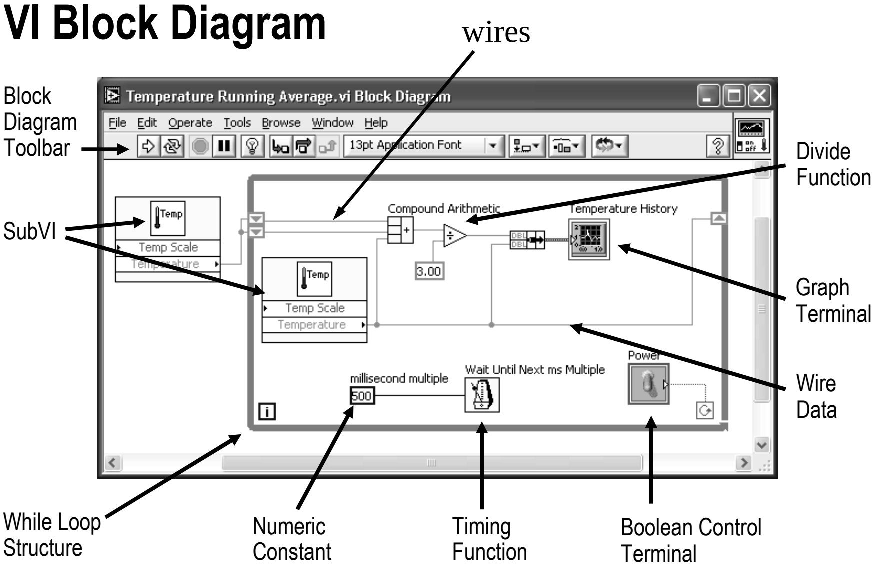
Boolean Control Terminal

Divide Function

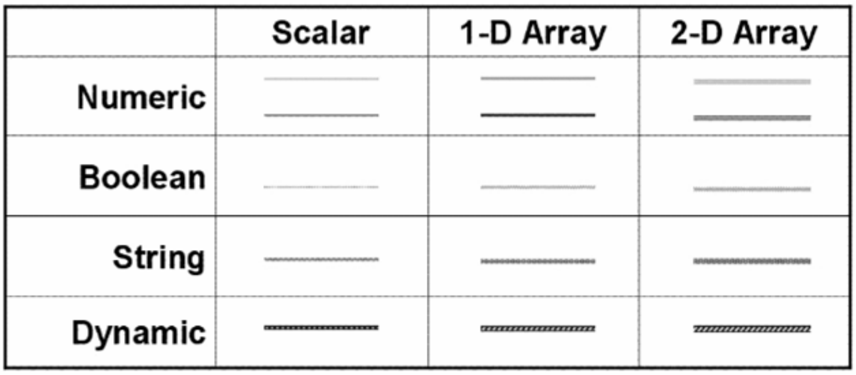
Graph Terminal

Wire Data

VI Block Diagram



Connection lines represent data types



	Scalar	1-D Array	2-D Array
Numeric	― ―	― ―	═ ═
Boolean	―	―	═
String	―	―	═
Dynamic	▬	▬	▬

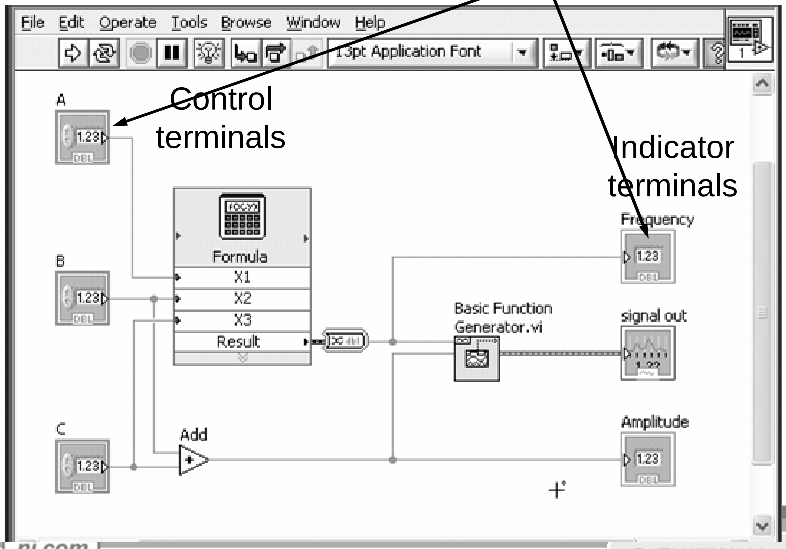
Data types

Control	Indicator	Data Type
I8	I8	Signed 8-bit integer
I16	I16	Signed 16-bit integer
I32	I32	Signed 32-bit integer
U8	U8	Unsigned 8-bit integer
U16	U16	Unsigned 16-bit integer
U32	U32	Unsigned 32-bit integer
SGL	SGL	Single-precision floating-point number
DBL	DBL	Double-precision floating-point number
EXT	EXT	Extended-precision floating-point number
abc	abc	String
TF	TF	Boolean
[I32]	[I32]	Array of signed 32-bit integers
[I32]	[I32]	2D Array of signed 32-bit integers
▤	▤	Cluster
🗋	🗋	File Refnum

ni.com

VI Block Diagram

Terminals: links between block diagram and the front panel



Control terminals

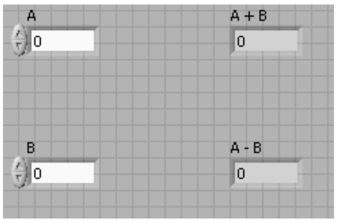
Indicator terminals

ni.com

Creating a VI

Terminals: links between block diagram and the front panel

Front Panel Window



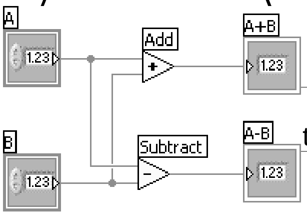
A 0

A + B 0

B 0

A - B 0

Block Diagram Window



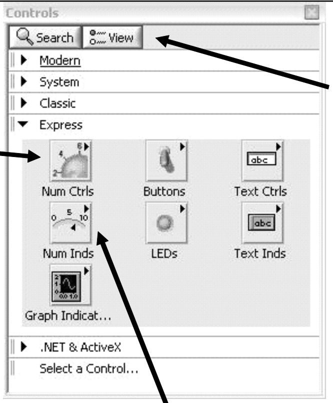
Control terminals

Indicator terminals

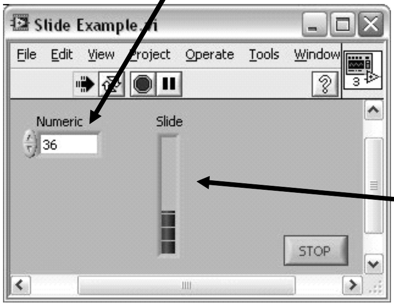
Controls Palette (Controls & Indicators)

In the Front Panel (right click)

Control: Numeric



Customize Palette View



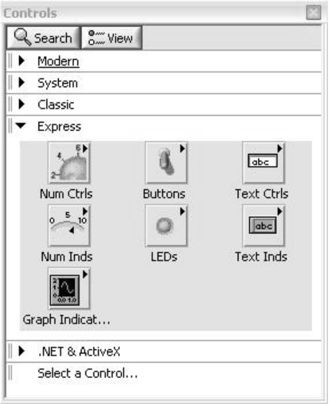
Numeric 36

Indicator: Numeric Slide

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 & 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

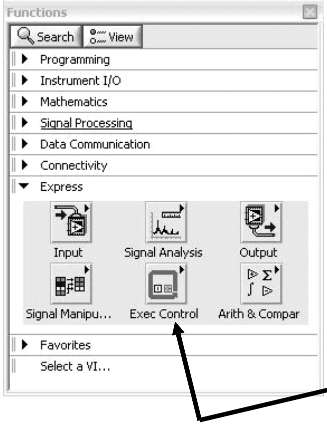


ni.com

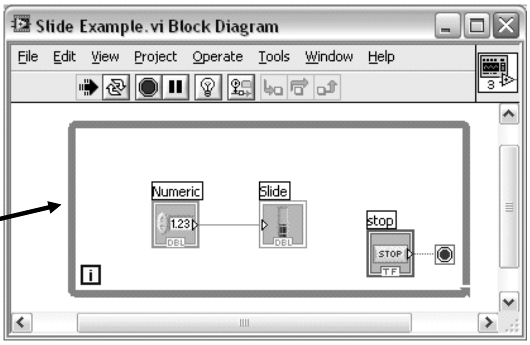
Functions (and Structures) Palette

Right click or View_Function

In the block diagram

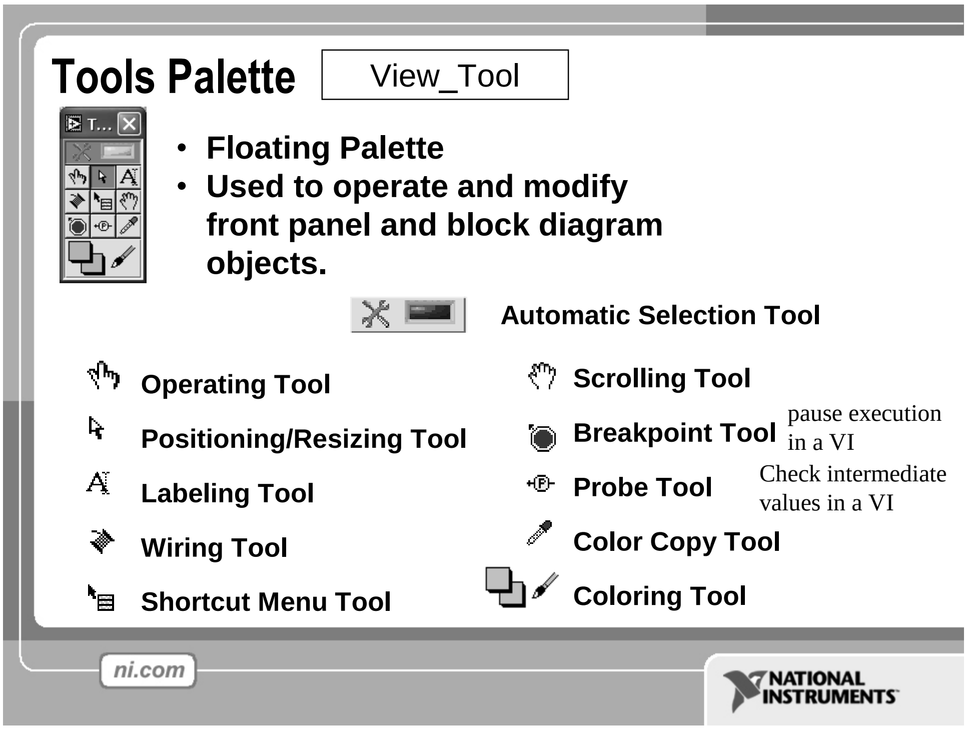


(Place items on the Block Diagram Window)



Structure: While Loop

ni.com



Tools Palette

View_Tool

- 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

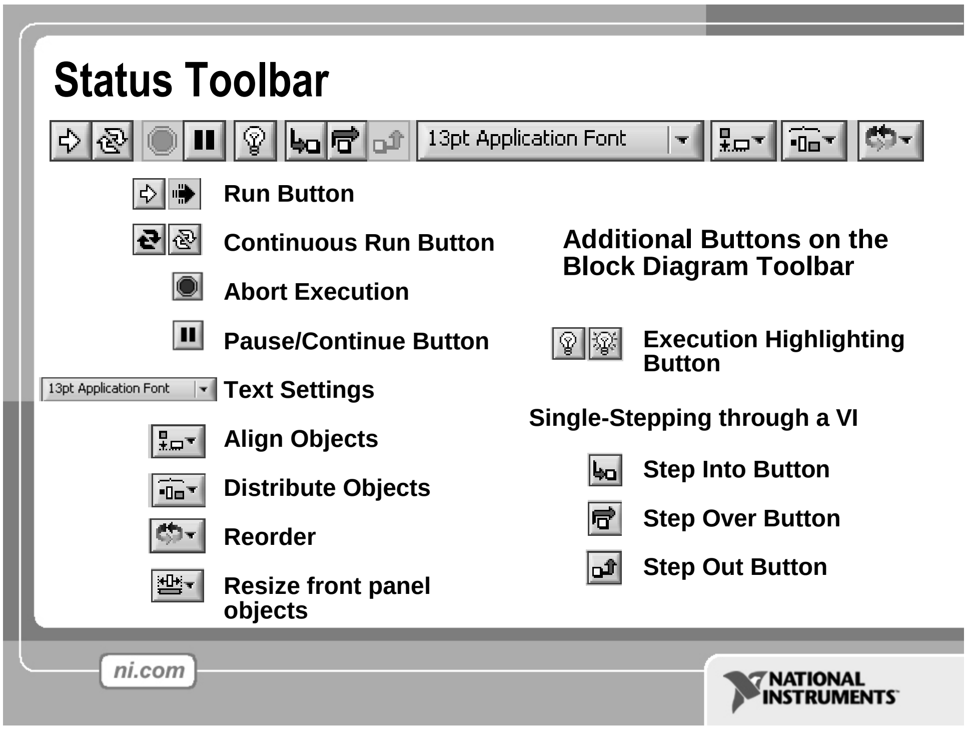
Breakpoint Tool pause execution in a VI

Probe Tool Check intermediate values in a VI

Color Copy Tool

Coloring Tool

ni.com



Status Toolbar

13pt Application Font

Run Button

Continuous Run Button

Abort Execution

Pause/Continue Button

Text Settings

Align Objects

Distribute Objects

Reorder

Resize front panel objects

Additional Buttons on the Block Diagram Toolbar

Execution Highlighting Button

Single-Stepping through a VI

Step Into Button

Step Over Button

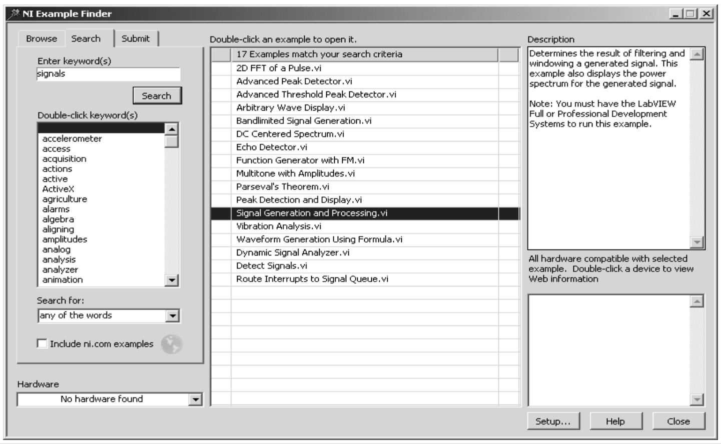
Step Out Button

ni.com

Finding an example VI

Help_Find Examples

Example finder



NI Example Finder

Browse | Search | Submit

Enter keyword(s)
signals

Search

Double-click keyword(s)

accelerometer
access
acquisition
actions
active
ActiveX
agriculture
alarms
algebra
aligning
amplitudes
analog
analysis
analyzer
animation

Search for:
any of the words

□ Include ni.com examples

Hardware
No hardware found

Double-click an example to open it.

17 Examples match your search criteria
2D FFT of a Pulse.vi
Advanced Peak Detector.vi
Advanced Threshold Peak Detector.vi
Arbitrary Wave Display.vi
Bandlimited Signal Generation.vi
DC Centered Spectrum.vi
Echo Detector.vi
Function Generator with FM.vi
Multitone with Amplitudes.vi
Parseval's Theorem.vi
Peak Detection and Display.vi
Signal Generation and Processing.vi
Vibration Analysis.vi
Waveform Generation Using Formula.vi
Dynamic Signal Analyzer.vi
Detect Signals.vi
Route Interrupts to Signal Queue.vi

Description

Determines the result of filtering and windowing a generated signal. This example also displays the power spectrum for the generated signal.

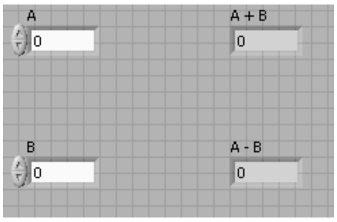
Note: You must have the LabVIEW Full or Professional Development Systems to run this example.

All hardware compatible with selected example. Double-click a device to view Web information

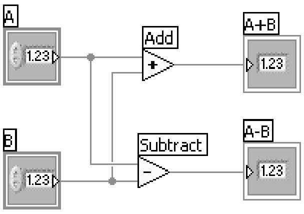
Setup... | Help | Close

Creating a VI

Front Panel Window



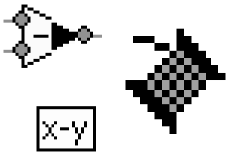
Block Diagram Window

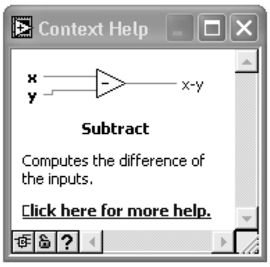


Control Terminals

Indicator Terminals

Creating a VI – Block Diagram



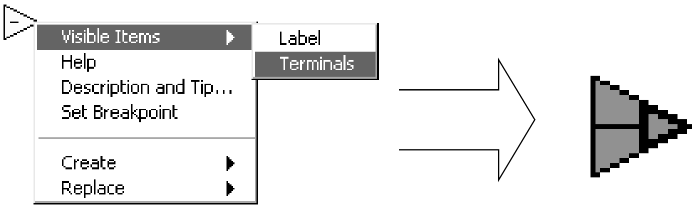


x
y
x-y

Subtract

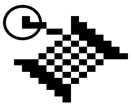
Computes the difference of the inputs.

Click here for more help.

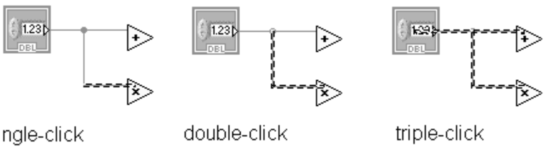


Wiring Tips – Block Diagram

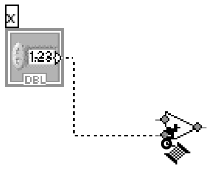
Wiring “Hot Spot”



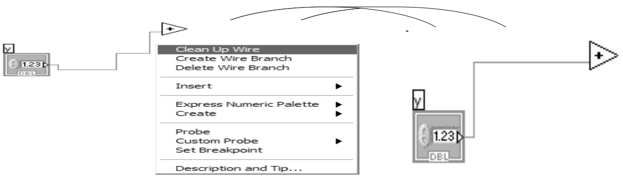
Click To Select Wires



Use Automatic Wire Routing

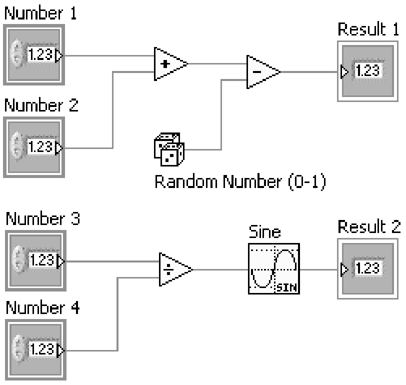


Clean Up Wiring



Dataflow Programming

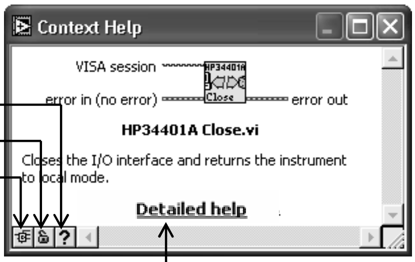
- Block diagram executes dependent on the flow of data; block diagram does NOT execute left to right
- Node executes when data is available to ALL input terminals
- Nodes supply data to all output terminals when done



Help Options

Context Help

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



HP34401A Close.vi

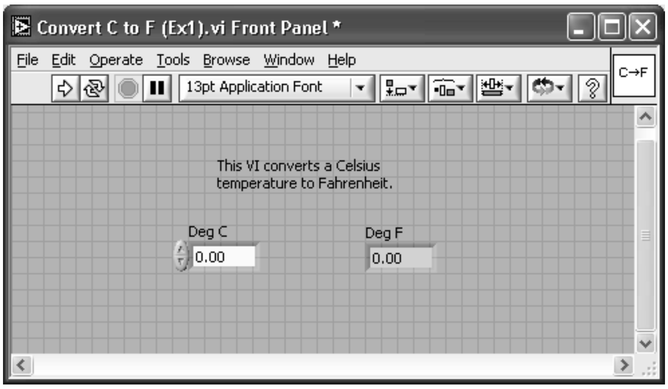
Closes the I/O interface and returns the instrument to local mode.

Detailed help

Additional Help

- Right-Click on the VI icon and choose **Help**
- Choose "**Detailed Help**." on the context help window

Exercise 1 - Convert °C to °F


Convert C to F (Ex1).vi Front Panel *
File Edit Operate Tools Browse Window Help
13pt Application Font
C→F

This VI converts a Celsius temperature to Fahrenheit.

Deg C 0.00

Deg F 0.00

1

NATIONAL INSTRUMENTS

Debugging Techniques

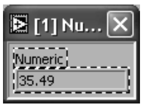
- Finding Errors

  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



Numeric 35.49

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

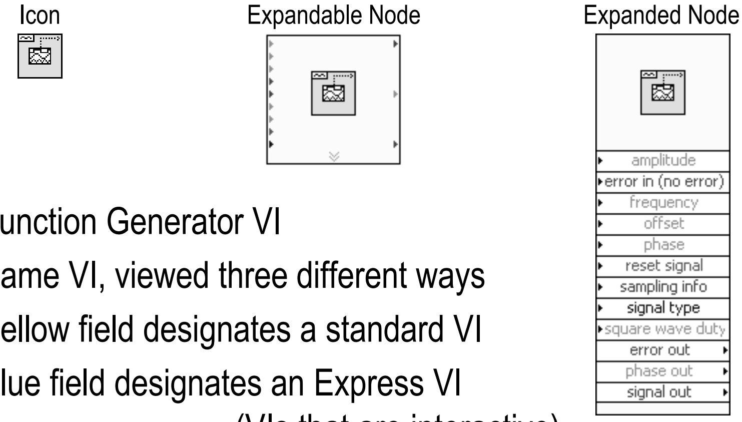
  You can also select Probe tool from Tools palette and click on wire

NATIONAL INSTRUMENTS

Section II

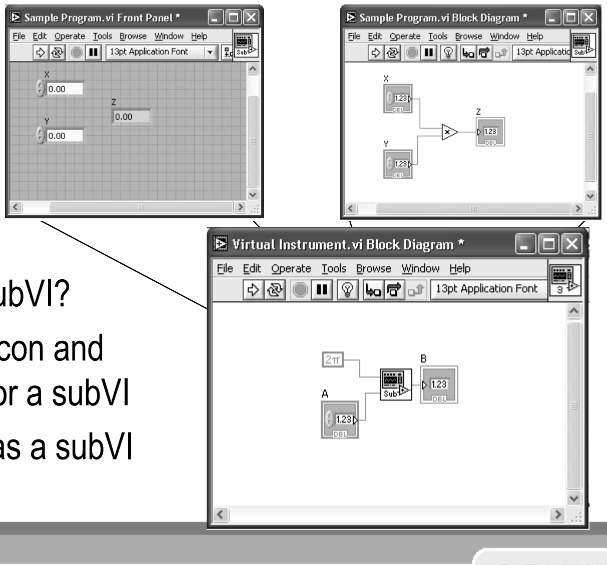
Sub VI

Block Diagram Nodes



- Function Generator VI
- Same VI, viewed three different ways
- Yellow field designates a standard VI
- Blue field designates an Express VI
  ________________(VIs that are interactive)

Section II – SubVIs

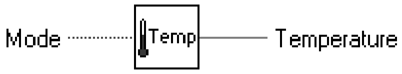

Sample Program.vi Front Panel *
File Edit Operate Tools Browse Window Help
13pt Application Font
X
0.00
Y
0.00
Z
0.00
Sample Program.vi Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Applicatio
X
Y
Z
123
DBL
Virtual Instrument.vi Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
2π
A
B
1.23
DBL
Sub

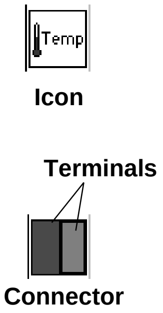
- What is a subVI?
- Making an icon and connector for a subVI
- Using a VI as a subVI

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

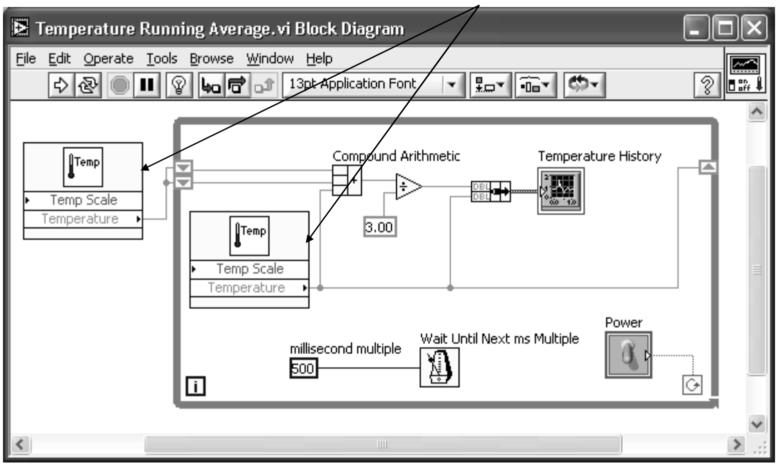




- An icon represents a VI in other block diagrams
- A connector shows available terminals for data transfer

SubVIs

Sub VIs

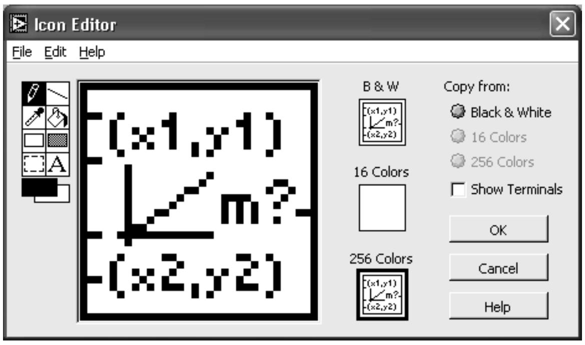

Temperature Running Average.vi Block Diagram

Steps to Create a SubVI

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

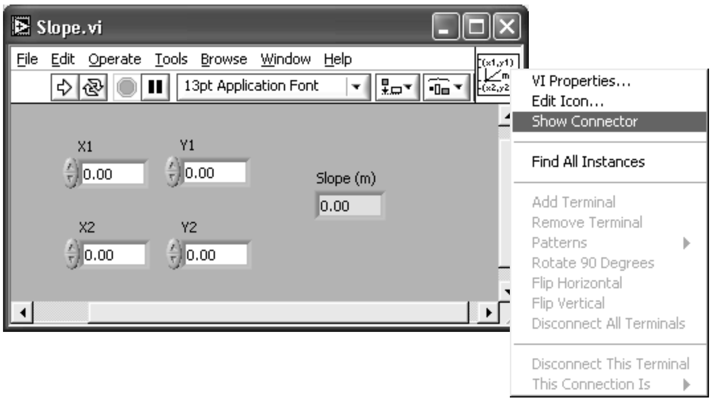
Create the Icon

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

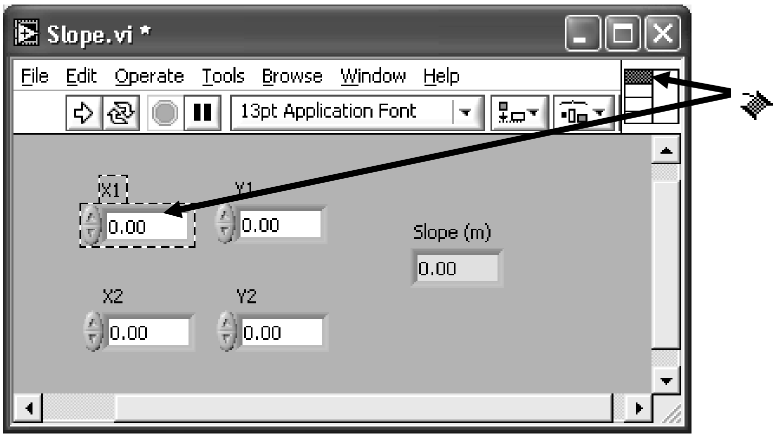


Create the Connector

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

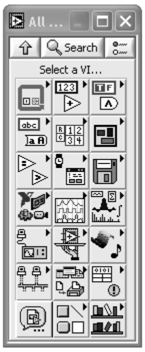
Insert the SubVI into a Top Level VI

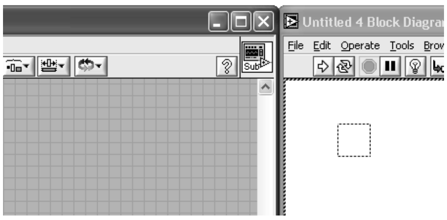
Accessing user-made subVIs

Functions >>All Functions >> Select a VI

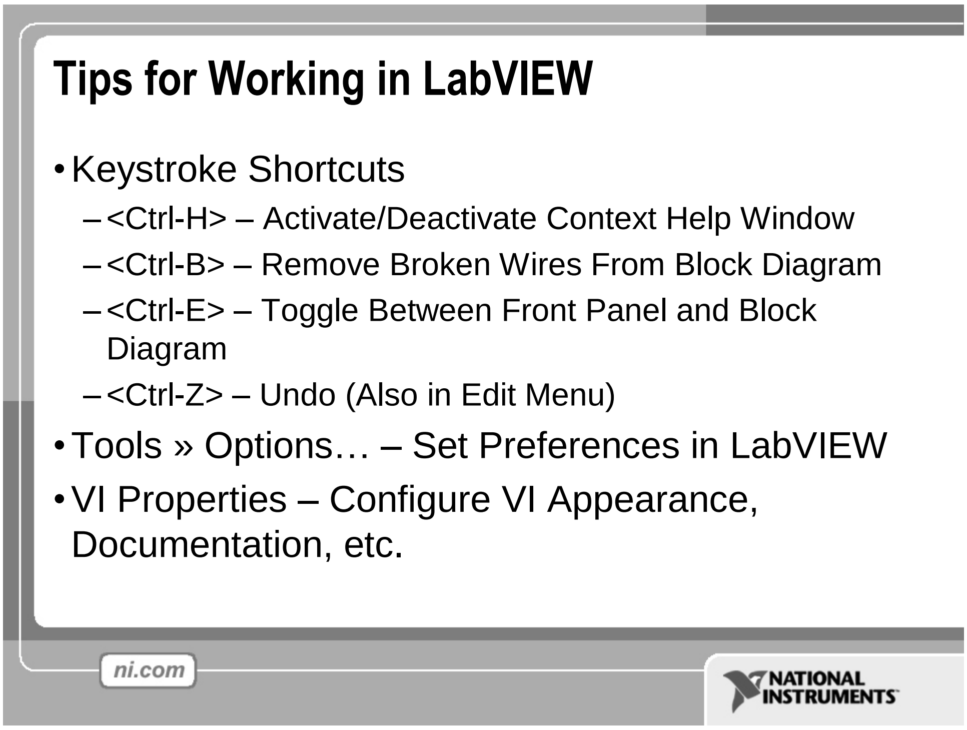
Or

Drag icon onto target diagram





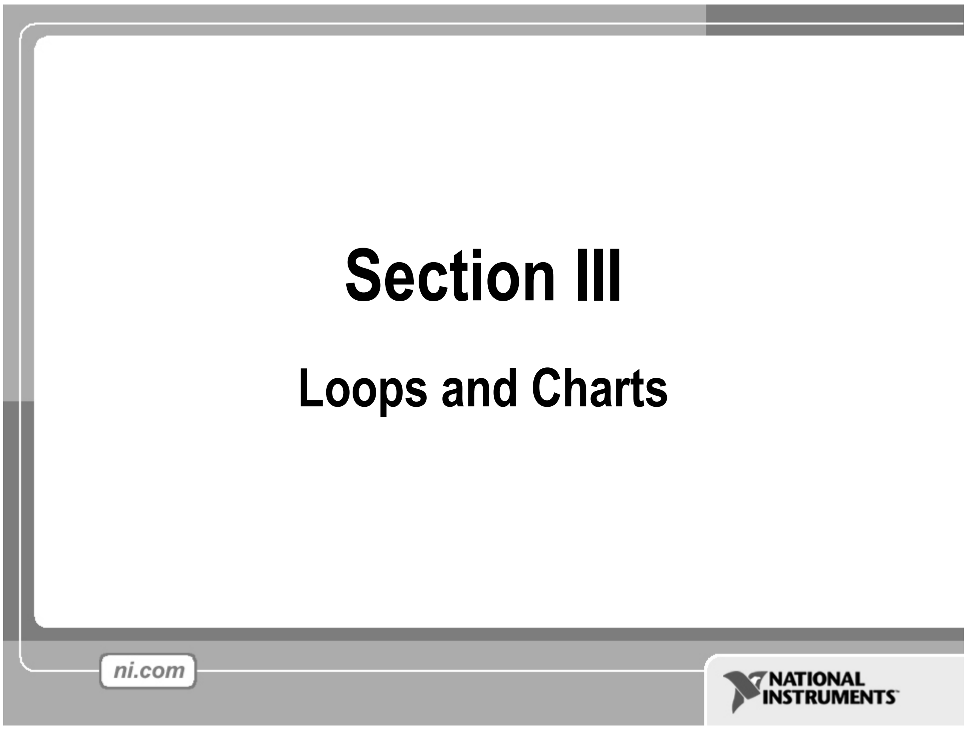
3



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.

ni.com



Section III

Loops and Charts

ni.com

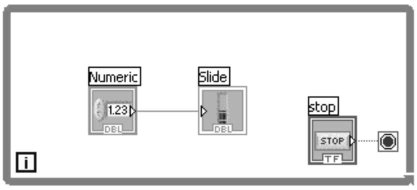
Section III – Loops and Charts

- While Loop
- For Loop
- Charts
- Multiplots

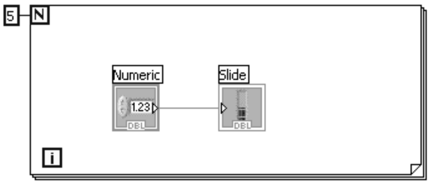
Loops

- While Loops
  - $\boxed{i}$ terminal counts iteration
  - Always runs at least once
  - Runs until stop condition is met ◉
- For Loops
  - $\boxed{i}$ terminal counts iterations
  - Run according to input $\boxed{N}$ of count terminal

While Loop

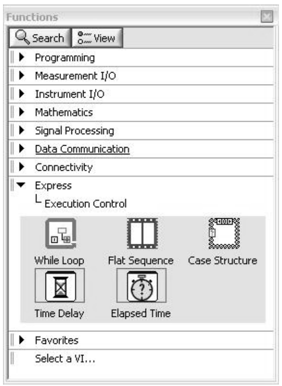


For Loop

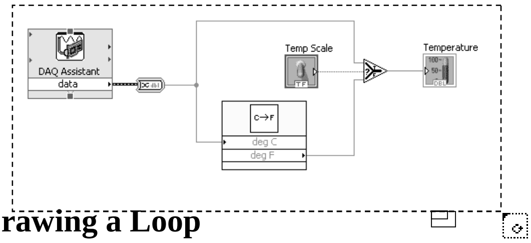


Drawing a Loop

1. Select the loop structure

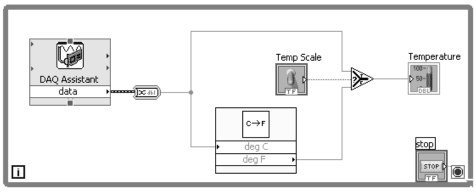


2. Enclose code to be repeated



Drawing a Loop

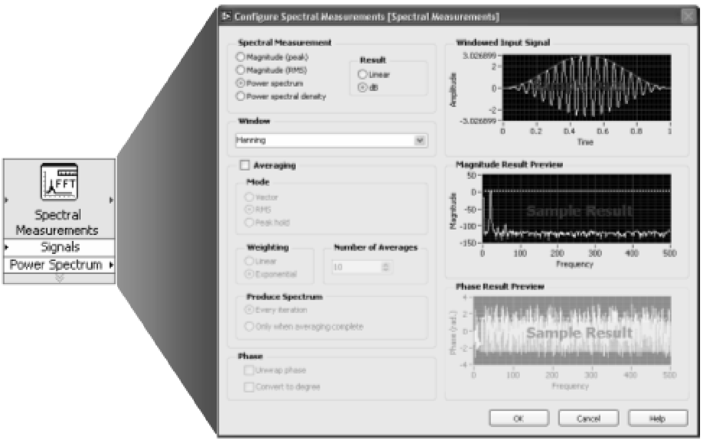
3. Drop or drag additional nodes and then wire

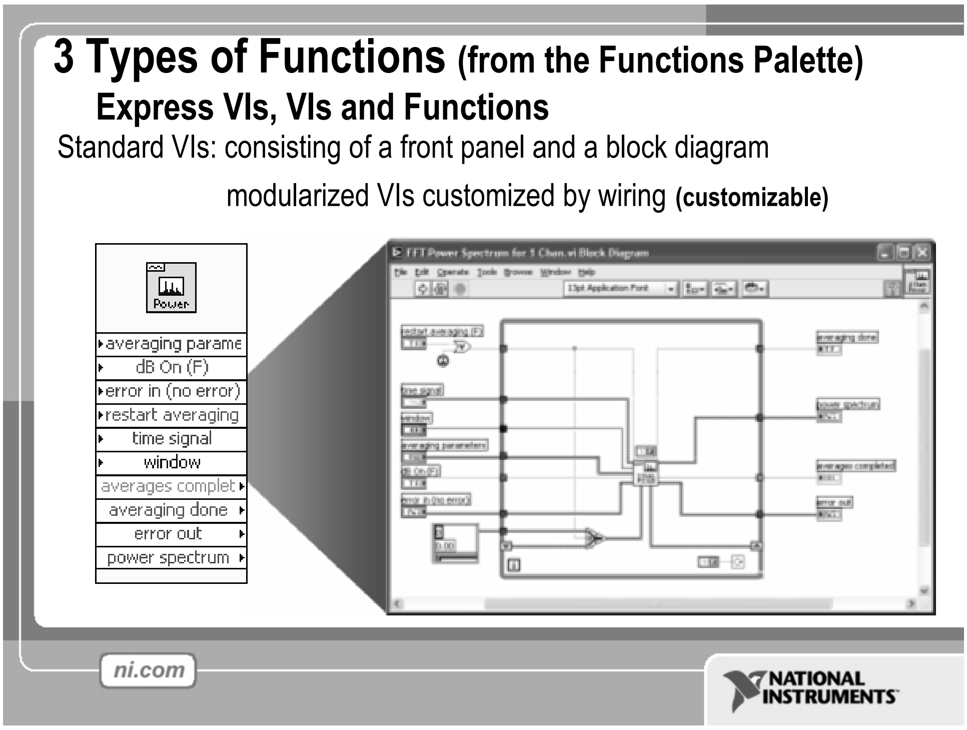


3 Types of Functions (from the Functions Palette)

Express VIs, VIs and Functions

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





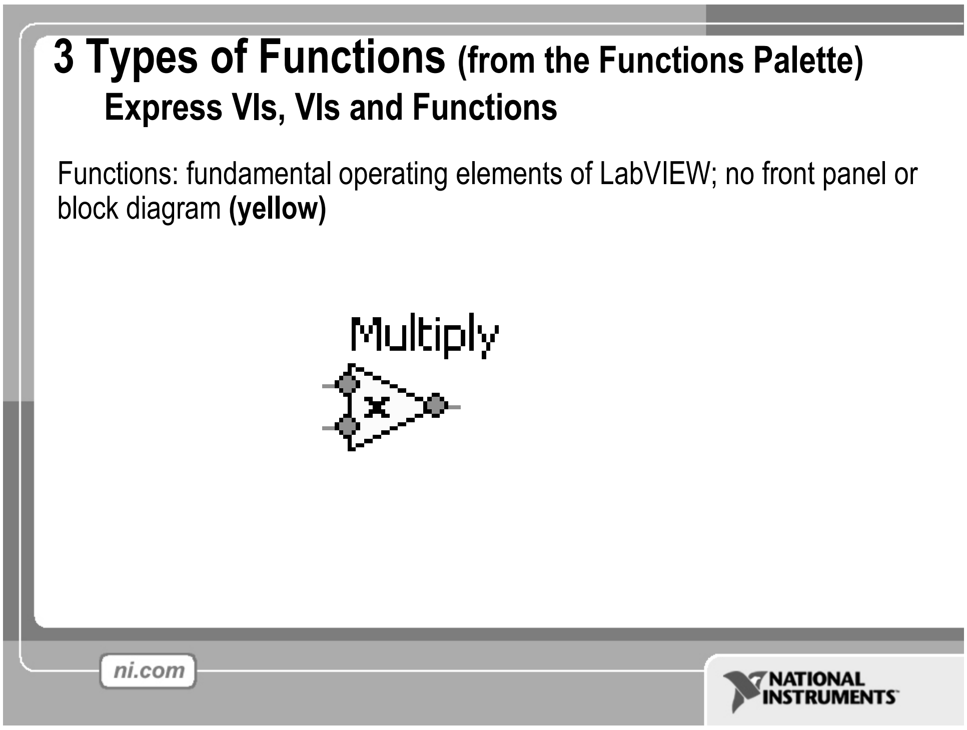
3 Types of Functions (from the Functions Palette)

Express VIs, VIs and Functions

Standard VIs: consisting of a front panel and a block diagram

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

ni.com



3 Types of Functions (from the Functions Palette)

Express VIs, VIs and Functions

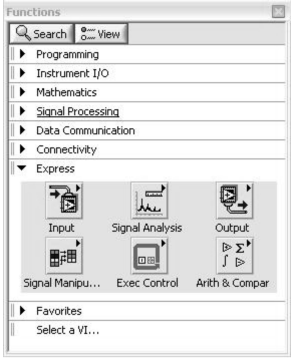
Functions: fundamental operating elements of LabVIEW; no front panel or block diagram **(yellow)**

ni.com

What Types of Functions are Available?

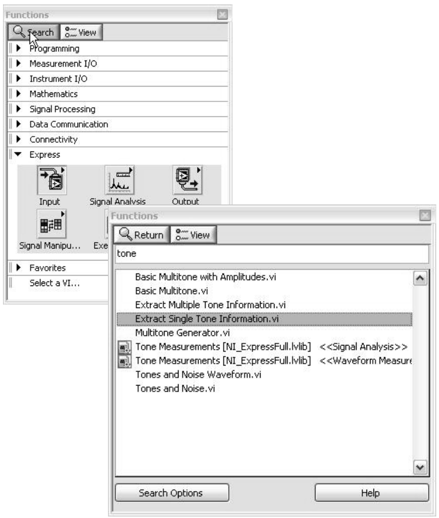
- **Input and Output**
  - Signal and Data Simulation
  - Acquire and Generate Real Signals with DAQ
  - Instrument I/O Assistant (Serial & GPIB)
  - ActiveX for communication with other programs
- **Analysis**
  - Signal Processing
  - Statistics
  - Advanced Math and Formulas
  - Continuous Time Solver
- **Storage**
  - File I/O

Express Functions Palette



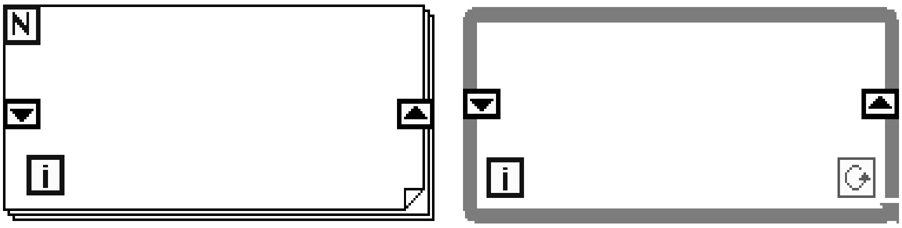
Searching for Controls, VIs, and Functions

- Palettes are filled with hundreds of VIs
- Press the search button to index the 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



Shift Registers in Loops

right click on loop border
Add Shift register



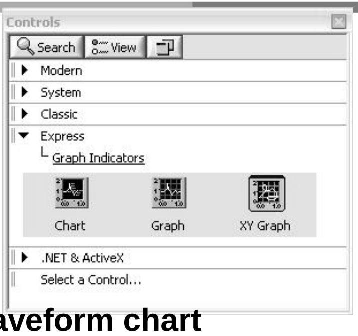
Use shift registers with For Loops and While Loops to transfer values from one loop iteration to the next. Shift registers are similar to static variables in text-based programming languages.

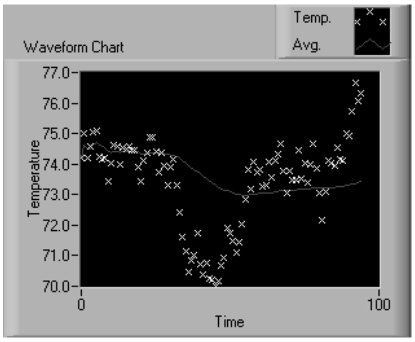
ni.com

5

NATIONAL INSTRUMENTS

Charts





Waveform chart

- special numeric indicator that can display a history of values
- Chart updates with each individual point it receives

Front panel»

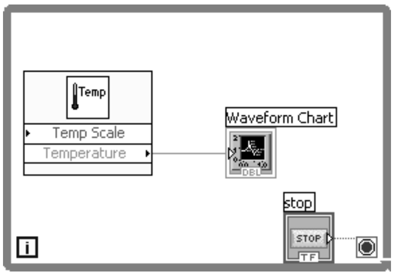
Functions»Express»Graph Indicators»Chart

ni.com

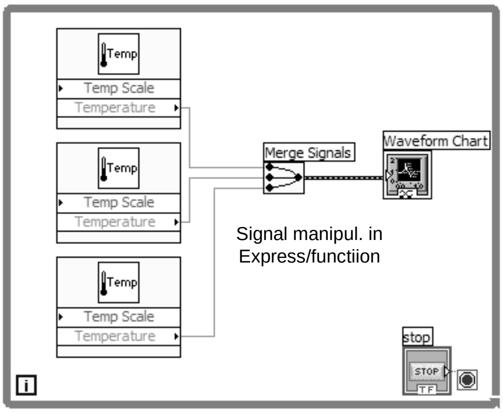
NATIONAL INSTRUMENTS

Wiring Data into Charts

Single Plot Charts

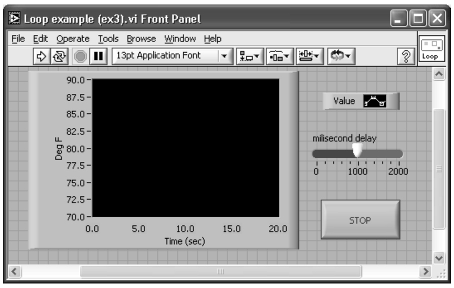


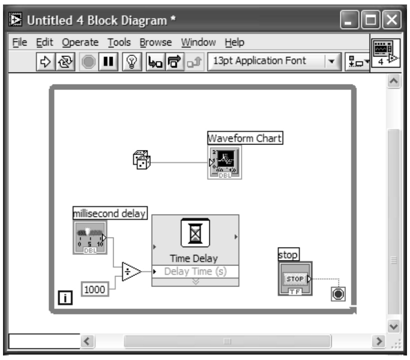
Multiplot Charts



Signal manipul. in Express/functiion

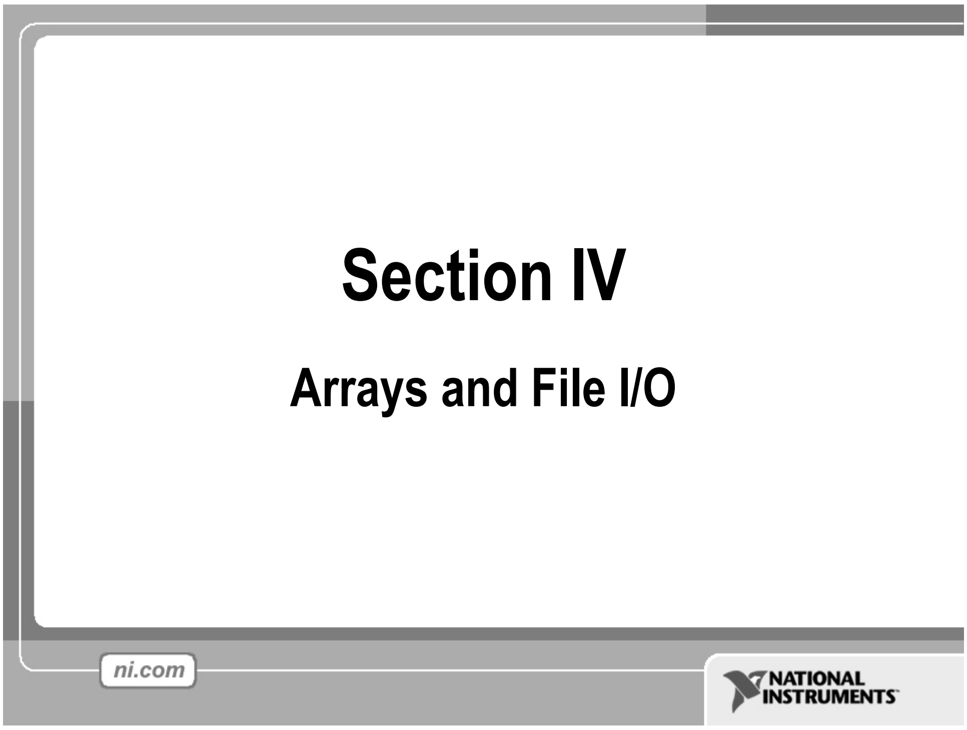
Exercise 3 – Using Charts





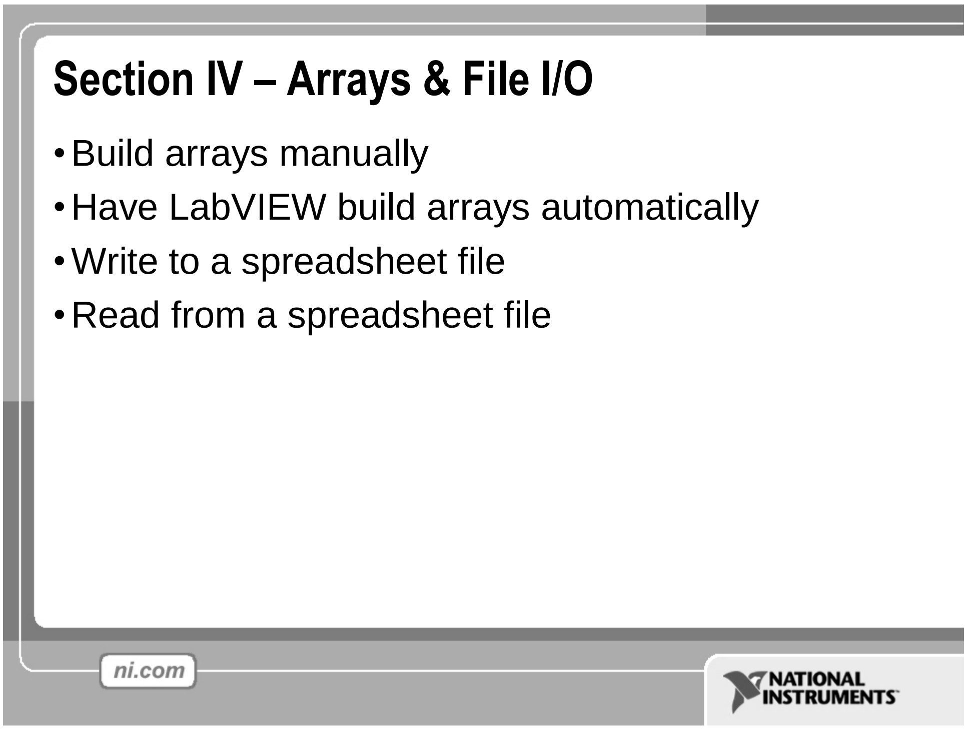
6

7



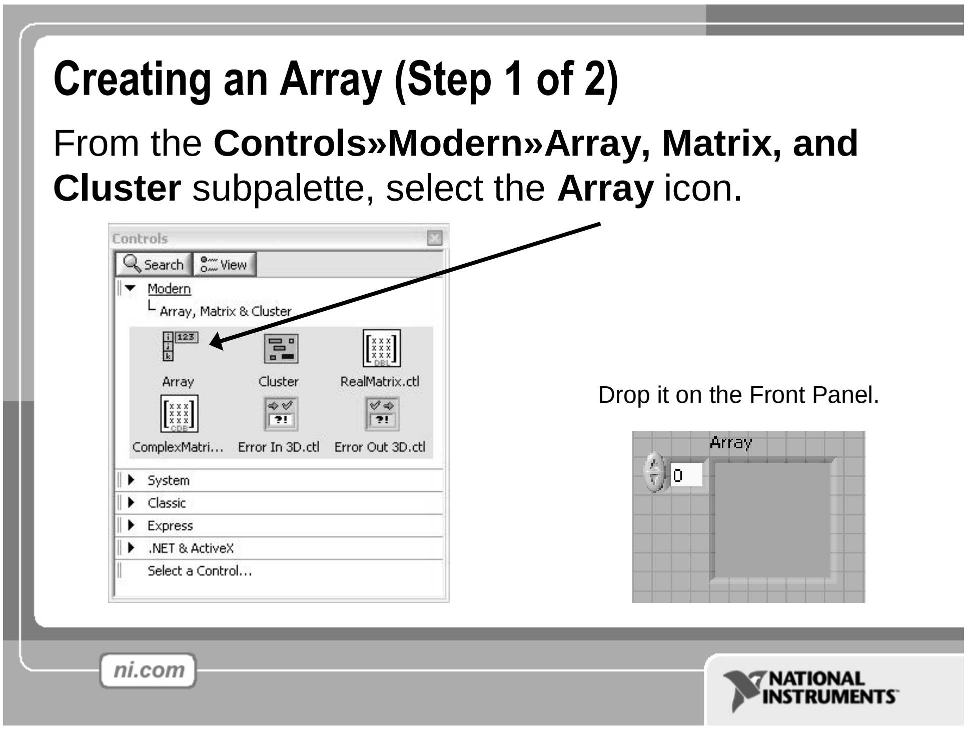
Section IV

Arrays and File I/O



Section IV – Arrays & File I/O

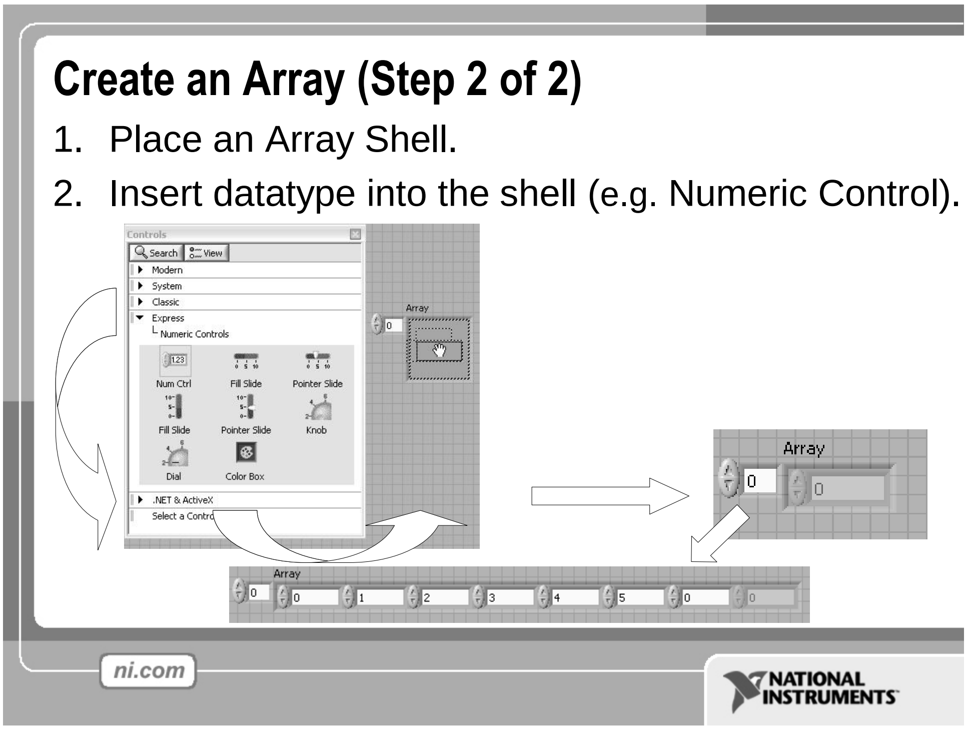
- Build arrays manually
- Have LabVIEW build arrays automatically
- Write to a spreadsheet file
- Read from a spreadsheet file



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.

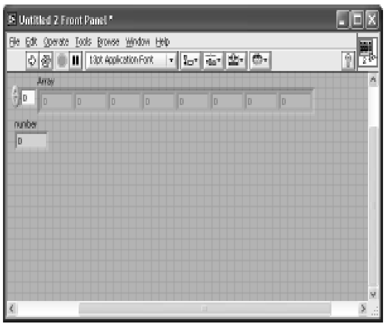


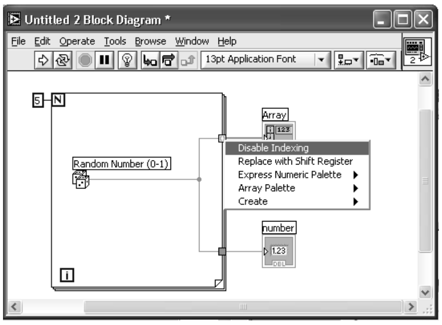
Create an Array (Step 2 of 2)

1. Place an Array Shell.
2. Insert datatype into the shell (e.g. Numeric Control).

Creating an Array with a Loop

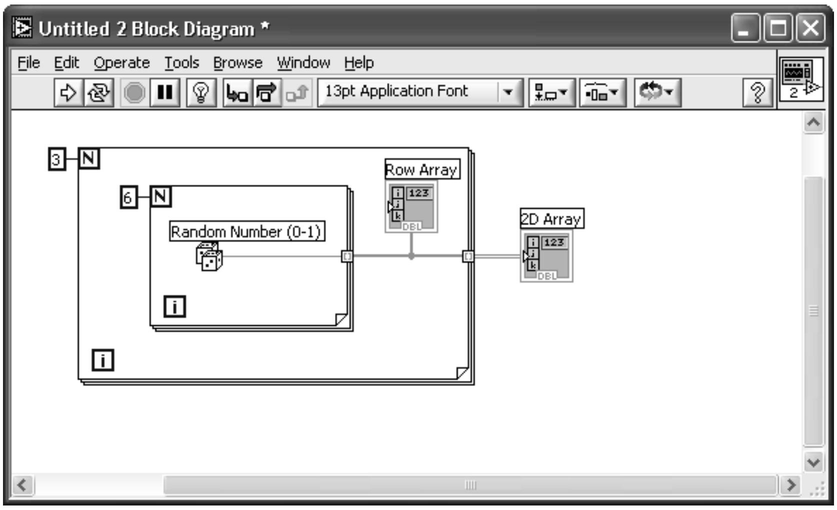
- Loops accumulate arrays at their boundaries
- Pulling a wire through a loop boundary creates an input tunnel





While Loop default no autoindexing
For Loop default with autoindexing

Creating 2D Arrays

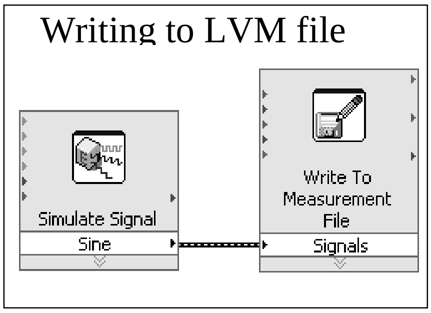

Untitled 2 Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
3
N
6
N
Random Number (0-1)
Row Array
2D Array
i
i

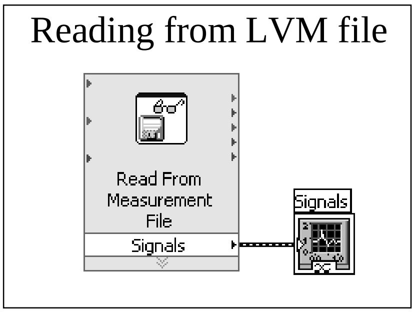
File I/O

- File I/O – Allows recording or reading data in a file.
- LabVIEW creates or uses the following file formats:
  - –Binary: underlying file format of all other file formats
  - –ASCII: regular text files
  - –LVM: LabVIEW measurement data file
  - –TDM: created for National Instruments products

High Level File I/O Functions

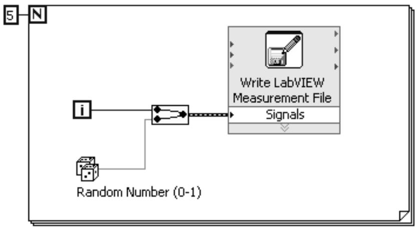
- **Easy to use**
- **High Level of abstraction**

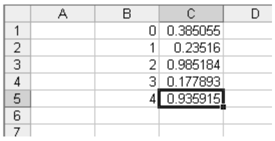

Writing to LVM file


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

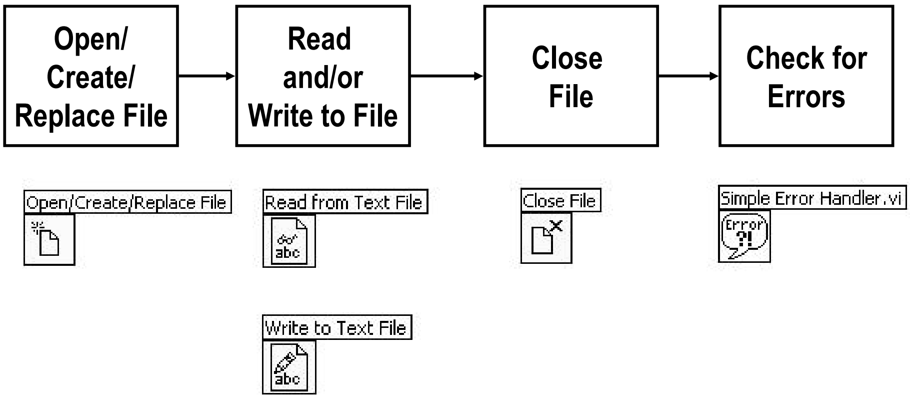




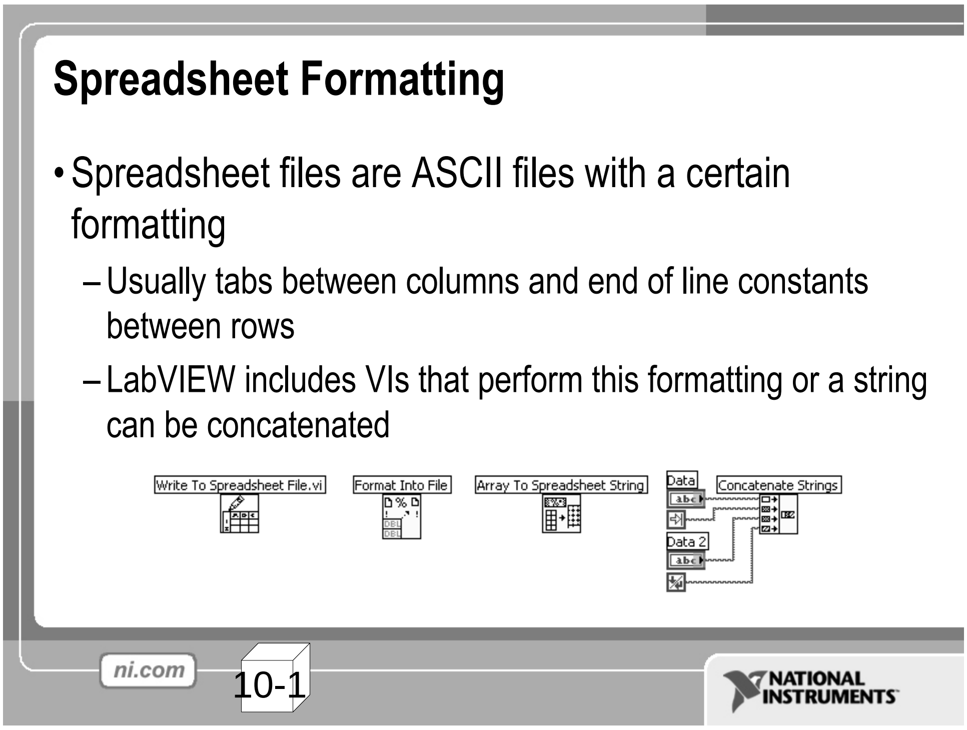
	A	B	C	D
1		0	0.385055	
2		1	0.23516	
3		2	0.985184	
4		3	0.177893	
5		4	0.935915	
6				
7				

10

File I/O Programming Model – Under the hood

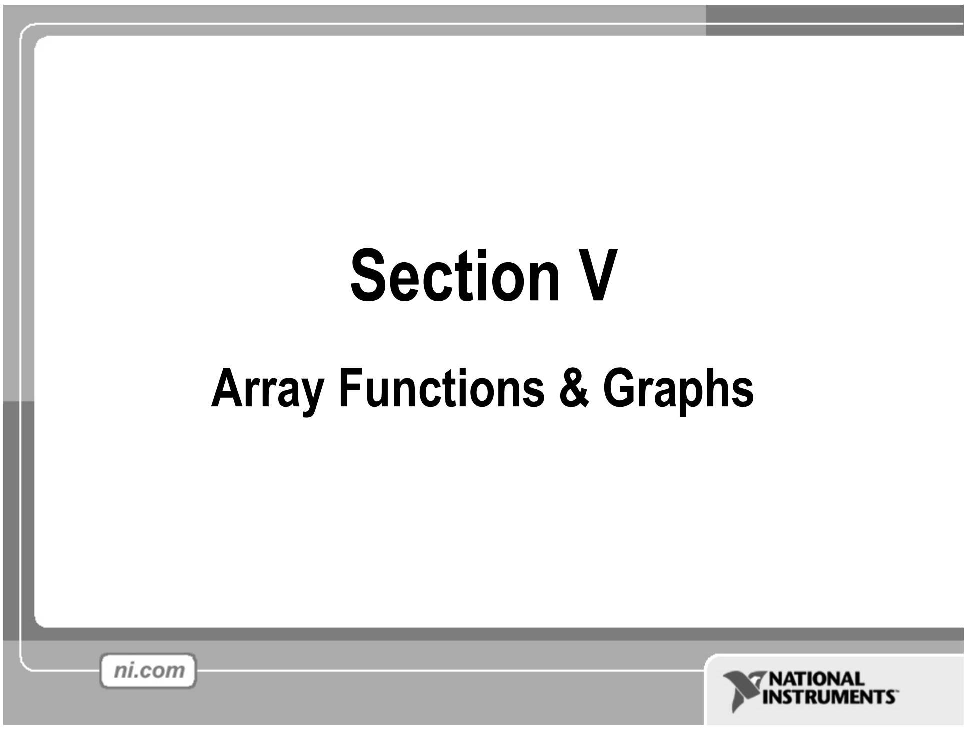


Open/ Create/ Replace File → Read and/or Write to File → Close File → Check for Errors



Spreadsheet Formatting

- Spreadsheet files are ASCII files with a certain formatting
  - Usually tabs between columns and end of line constants between rows
  - LabVIEW includes VIs that perform this formatting or a string can be concatenated



Section V

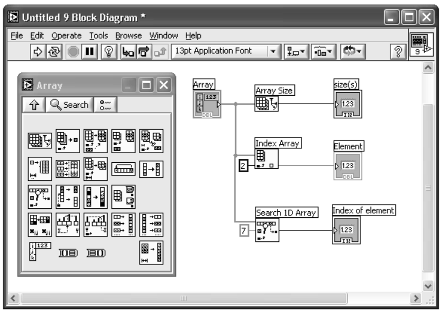
Array Functions & Graphs

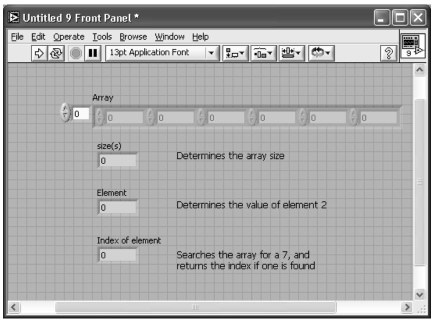
Section V – Array Functions & Graphs

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

Array Functions – Basics

Functions >> All functions>> Array



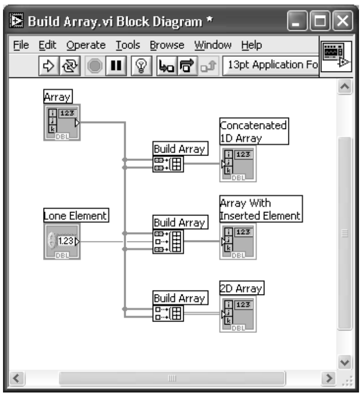


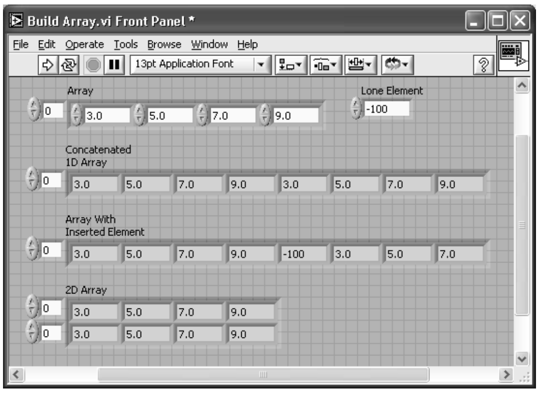
size(s)
0
Determines the array size

Element
0
Determines the value of element 2

Index of element
0
Searches the array for a 7, and returns the index if one is found

Array Functions – Build Array


Build Array.vi Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Fo
Array
Lone Element
Build Array
Build Array
Build Array
Concatenated 1D Array
Array With Inserted Element
2D Array


Build Array.vi Front Panel *
File Edit Operate Tools Browse Window Help
13pt Application Font

Array: 0 | 3.0 | 5.0 | 7.0 | 9.0

Lone Element: -100

Concatenated 1D Array: 0 | 3.0 | 5.0 | 7.0 | 9.0 | 3.0 | 5.0 | 7.0 | 9.0

Array With Inserted Element: 0 | 3.0 | 5.0 | 7.0 | 9.0 | -100 | 3.0 | 5.0 | 7.0

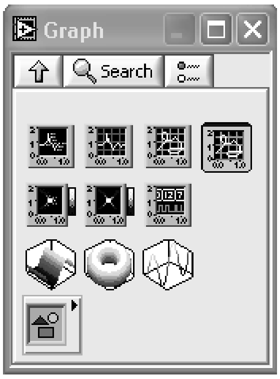
2D Array: 0 0 | 3.0 | 5.0 | 7.0 | 9.0 / 3.0 | 5.0 | 7.0 | 9.0

Graphs

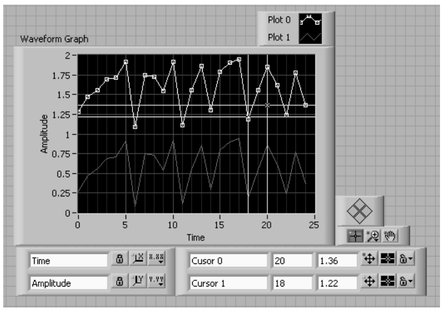
- Selected from the Graph palette of Controls menu

Controls>>All Controls>>Graphs

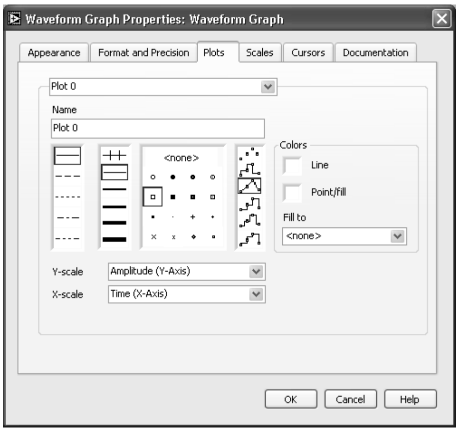
Waveform Graph – Plot an array of numbers against their indices
Express XY Graph – Plot one array against another
Digital Waveform Graph – Plot bits from binary data



Graphs



Time
Amplitude
Cursor 0 20 1.36
Cursor 1 18 1.22



Plot 0

Name Plot 0

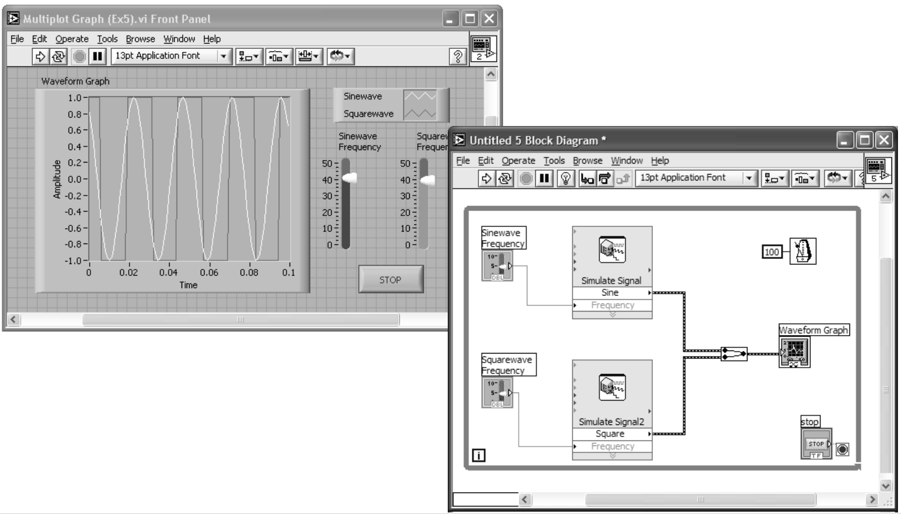
Fill to <none>

Y-scale Amplitude (Y-Axis)

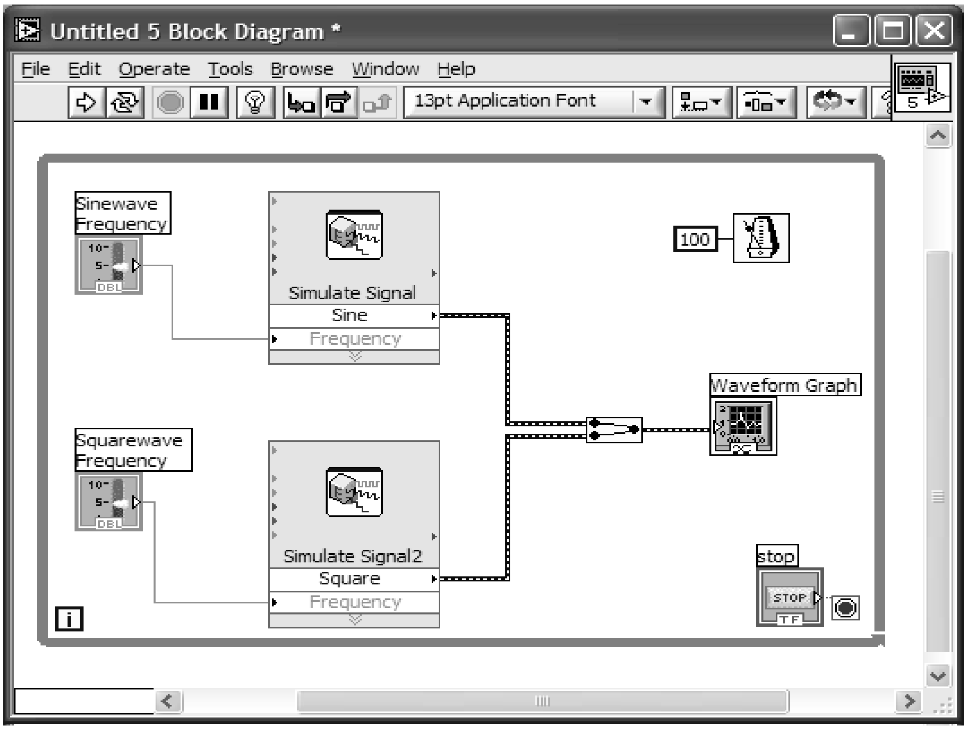
X-scale Time (X-Axis)

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

Exercise 5 – Using Waveform Graphs



11



Section VI

Strings, Clusters & Error Handling

ni.com

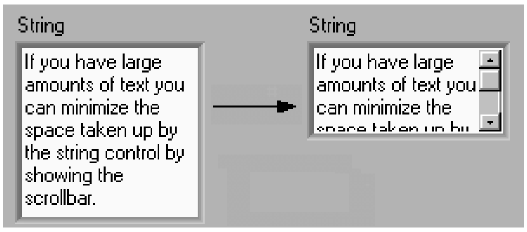
NATIONAL INSTRUMENTS

Section VI – 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**

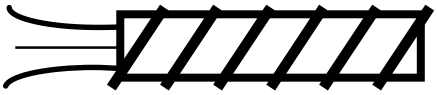


If you have large amounts of text you can minimize the space taken up by the string control by showing the scrollbar.

If you have large amounts of text you can minimize the space taken up by

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

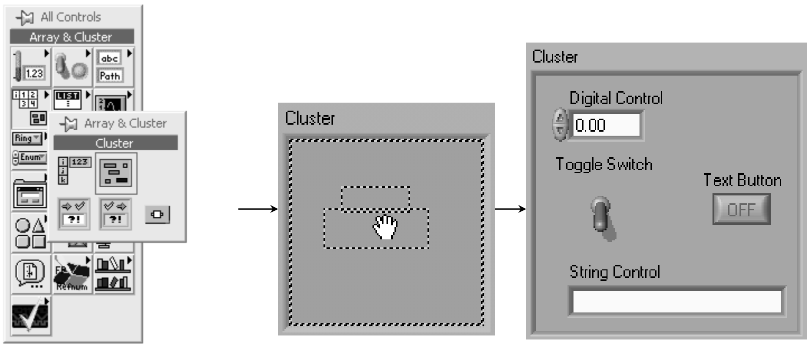


Creating a Cluster

1. Select a **Cluster** shell

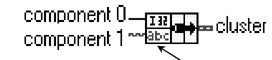
2. Place objects inside the shell

Controls >> All Controls >> Array & Cluster



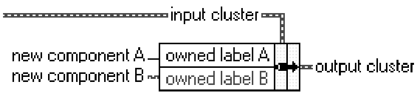
Cluster Functions

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

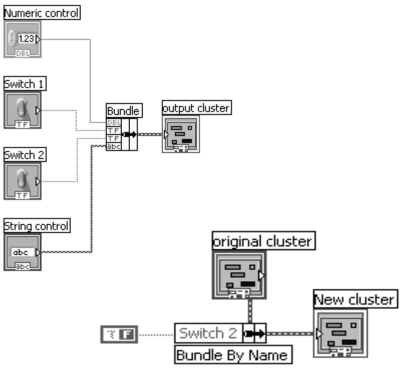


(Terminal labels reflect data type)

Bundle

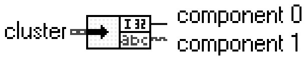


Bundle By Name

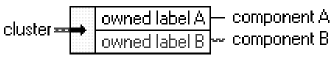


ni.com

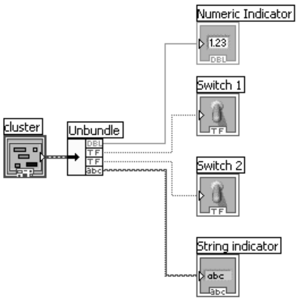
Cluster Functions

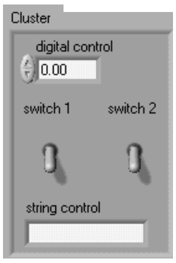


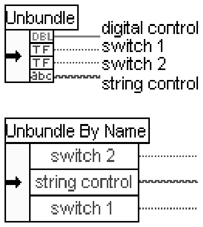
Unbundle



Unbundle By Name





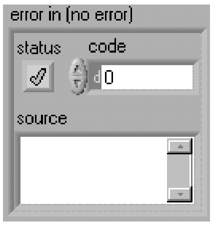


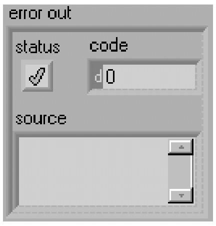
Unbundled cluster in the diagram

ni.com

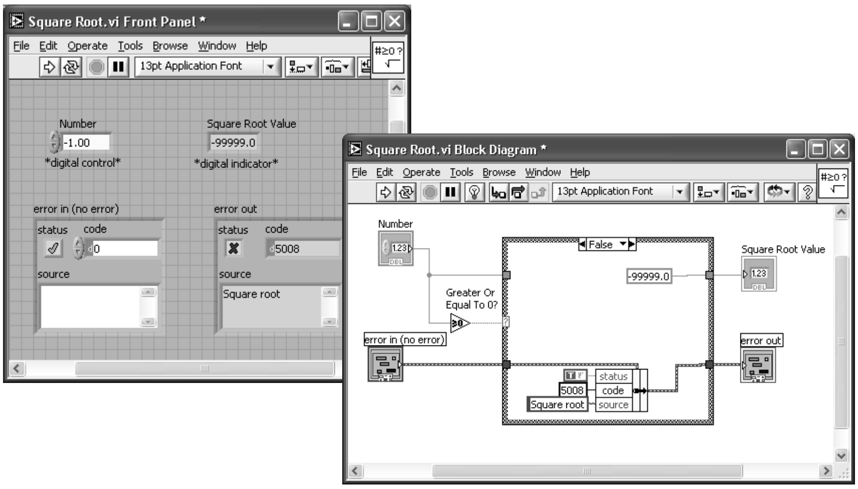
Error Clusters

- 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



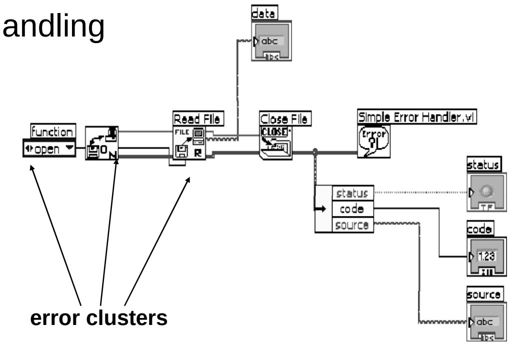


Exercise 6 – Error Clusters & Handling



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



error clusters

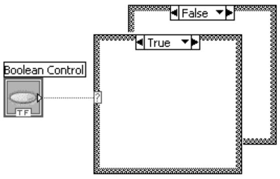
Section VII

Case & Sequence Structure

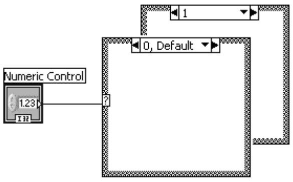
Formula Nodes

How Do I Make Decisions in LabVIEW?

1. Case Structures (/Programming/Structure)

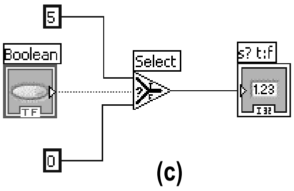


(a)



(b)

2. Select(/Programming/comparison)

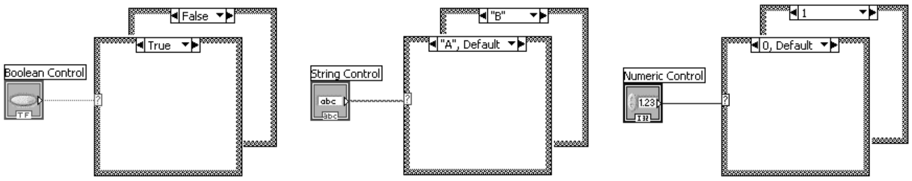


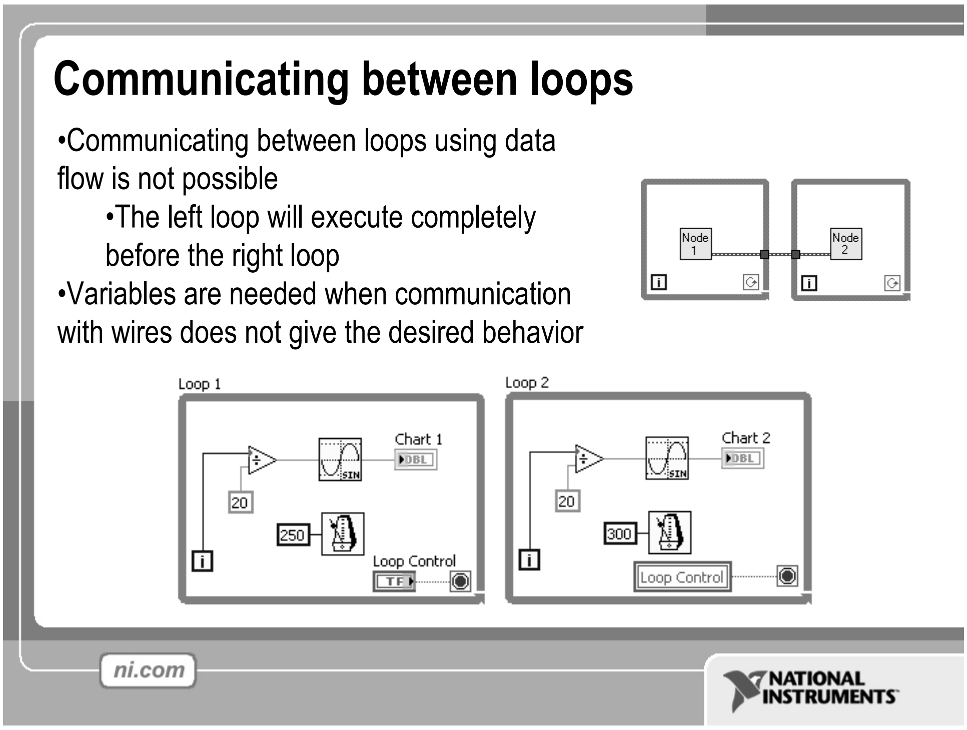
(c)

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

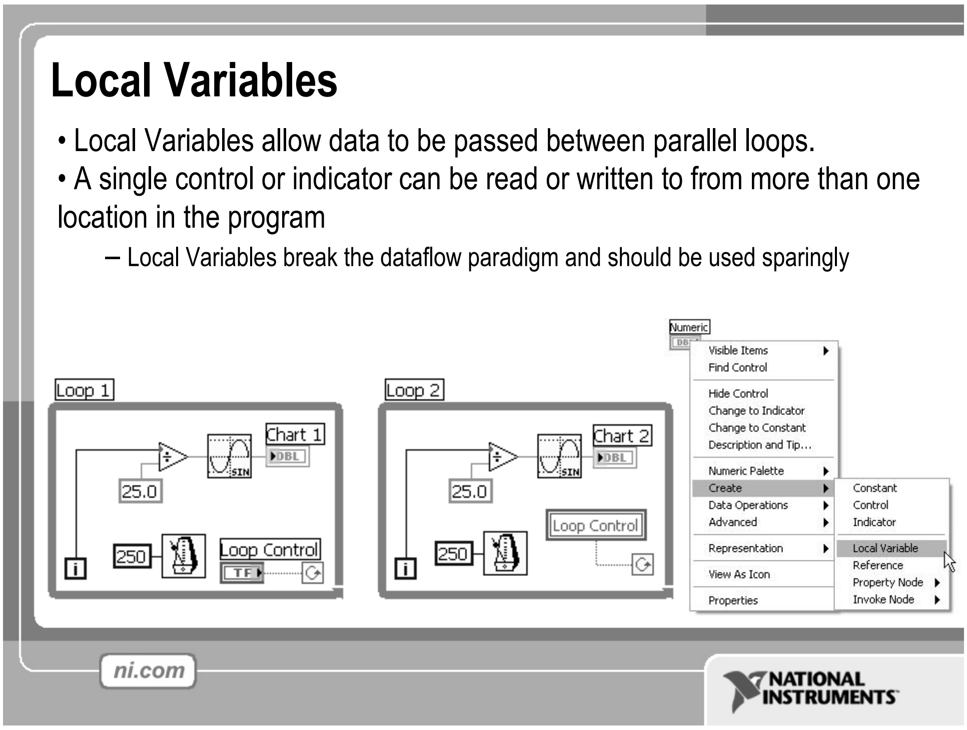




Communicating between loops

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

ni.com



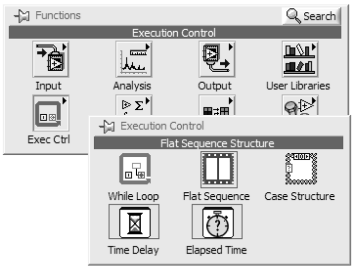
Local Variables

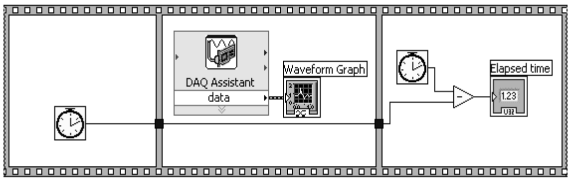
- • Local Variables allow data to be passed between parallel loops.
- • A single control or indicator can be read or written to from more than one location in the program
  - – Local Variables break the dataflow paradigm and should be used sparingly

ni.com

Sequence Structures

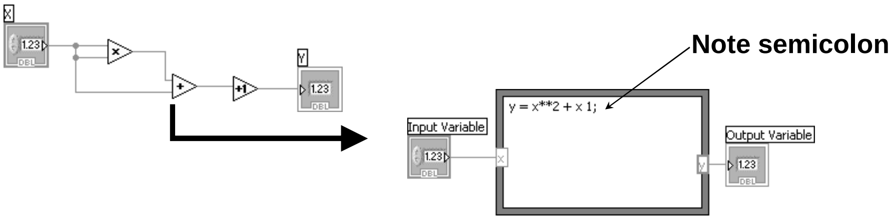
- In the **Execution Control** subpalette of Functions palette
- Executes diagrams sequentially
- Right-click to add new frame

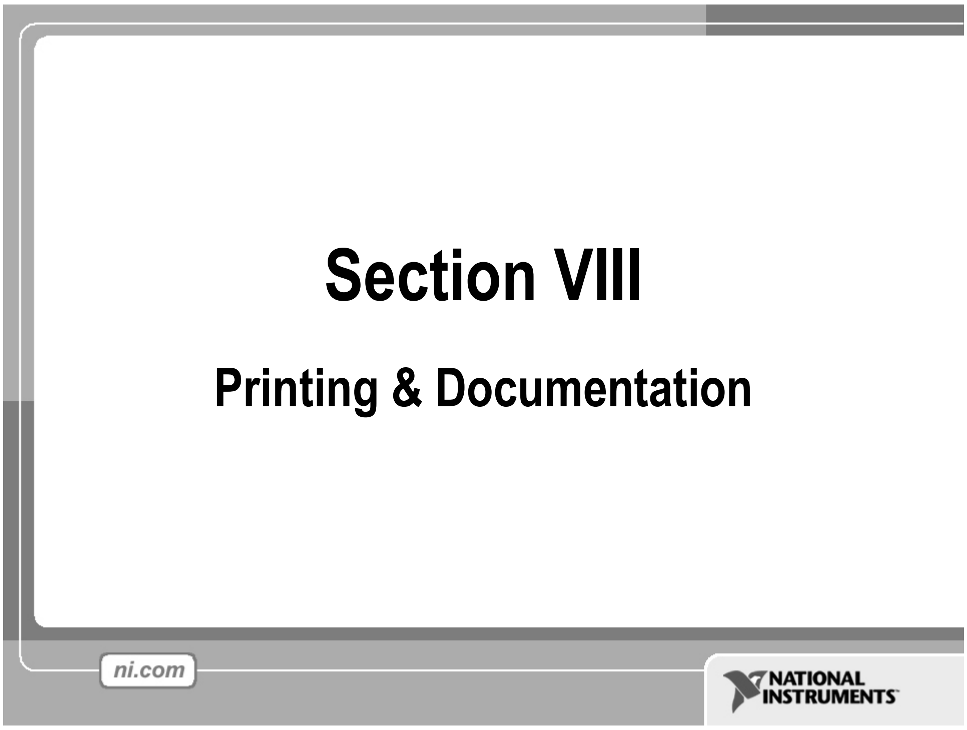




Formula Nodes

- In the Structures subpalette
- Implement complicated equations
- Variables created at border by “add input” or “add output”
- **every assigned variable (left of equation) must have an output terminal on the formula node.**
- Variable names are case sensitive
- Each statement must terminate with a semicolon (;)

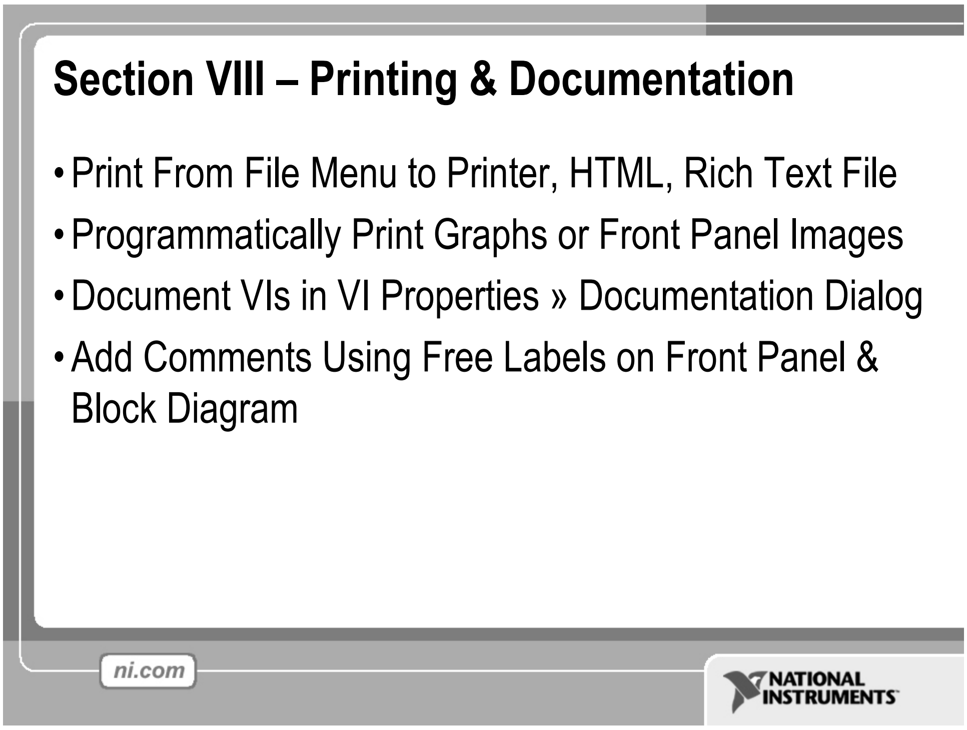




Section VIII

Printing & Documentation

ni.com



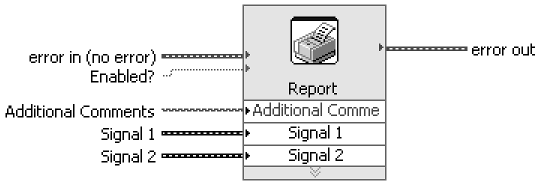
Section VIII – Printing & Documentation

- 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

ni.com

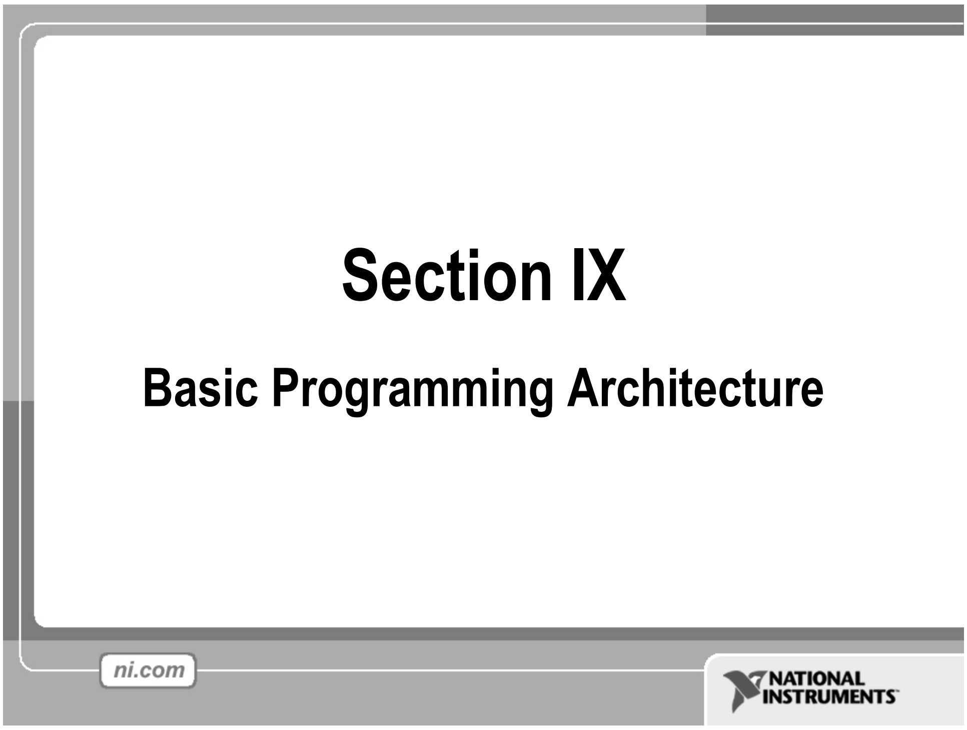
Printing

- **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

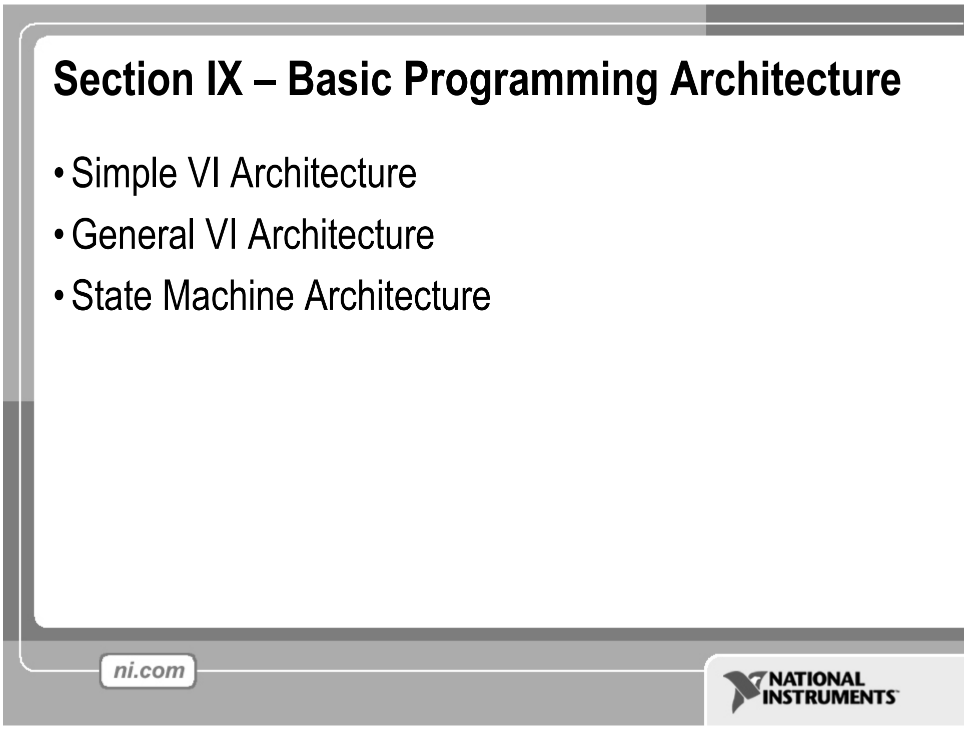


Section IX

Basic Programming Architecture

ni.com

NATIONAL INSTRUMENTS



Section IX – Basic Programming Architecture

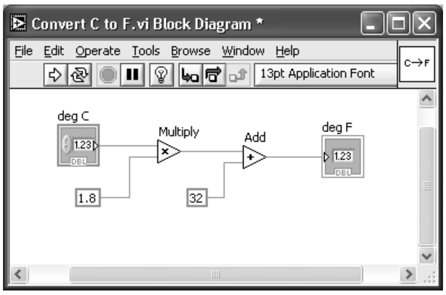
- Simple VI Architecture
- General VI Architecture
- State Machine Architecture

ni.com

NATIONAL INSTRUMENTS

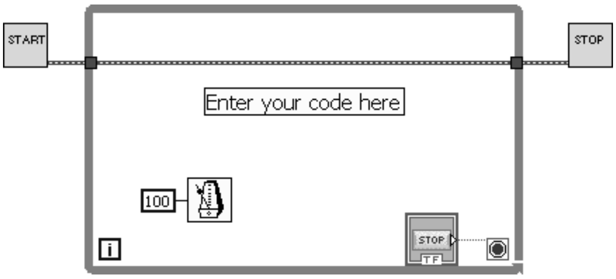
Simple VI Architecture

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


Convert C to F.vi Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
C→F
deg C
1.23
DBL
Multiply
Add
deg F
1.8
32

General VI Architecture

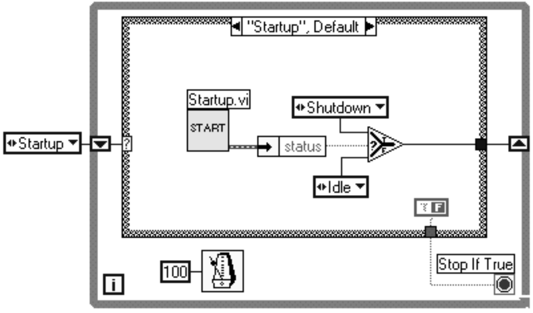
- Three Main Steps
  - Startup
  - Main Application
  - Shutdown



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

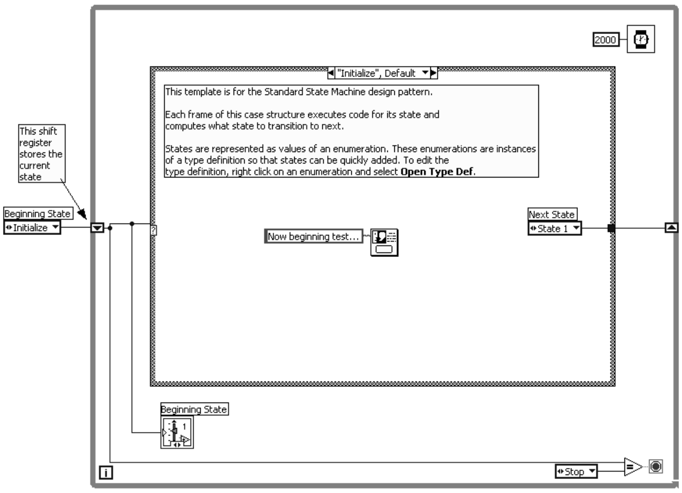
States:
0: Startup
1: Idle
2: Event 1
3: Event 2
4: Shutdown



ni.com

NATIONAL INSTRUMENTS

Exercise 7 – Simple State Machine



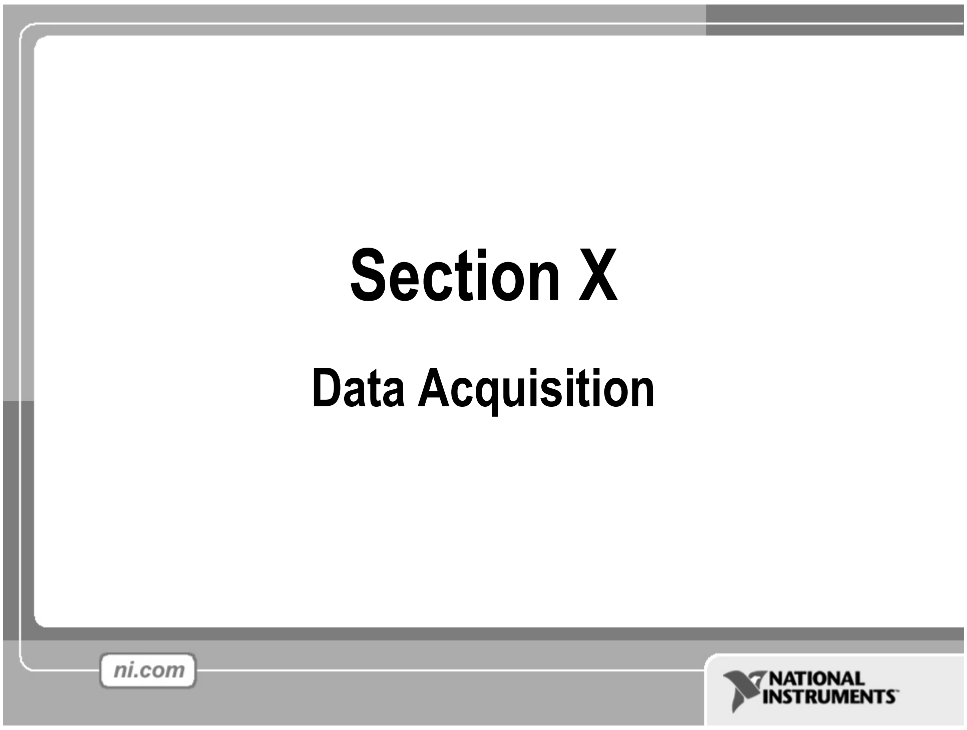
This template is for the Standard State Machine design pattern.

Each frame of this case structure executes code for its state and computes what state to transition to next.

States are represented as values of an enumeration. These enumerations are instances of a type definition so that states can be quickly added. To edit the type definition, right click on an enumeration and select **Open Type Def**.

ni.com

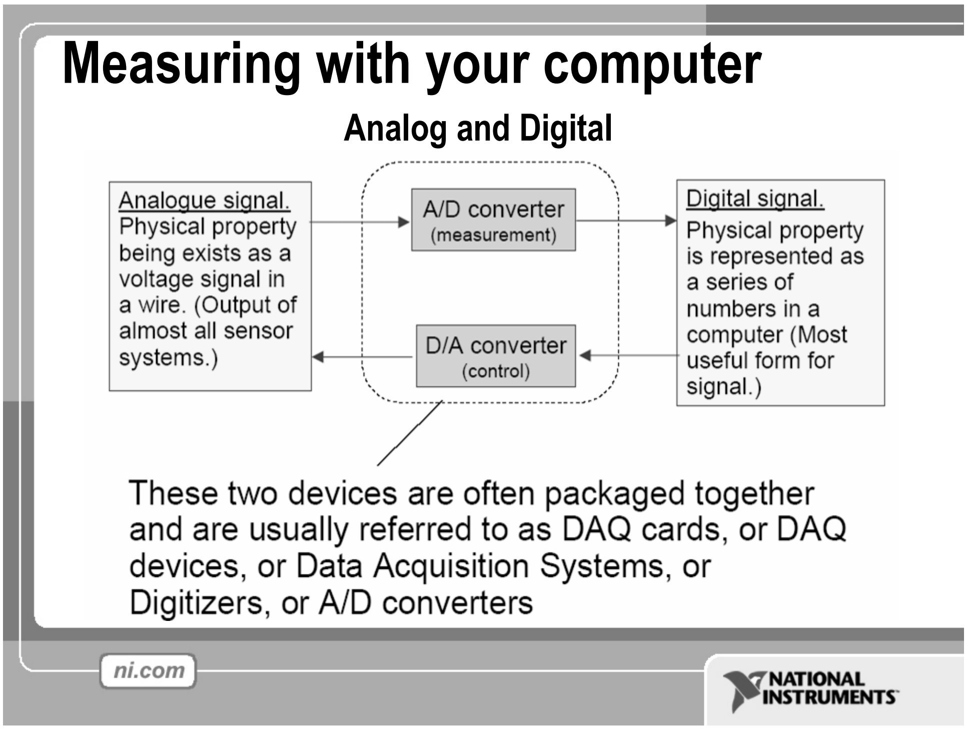
NATIONAL INSTRUMENTS



Section X

Data Acquisition

ni.com
NATIONAL INSTRUMENTS



Measuring with your computer

Analog and Digital

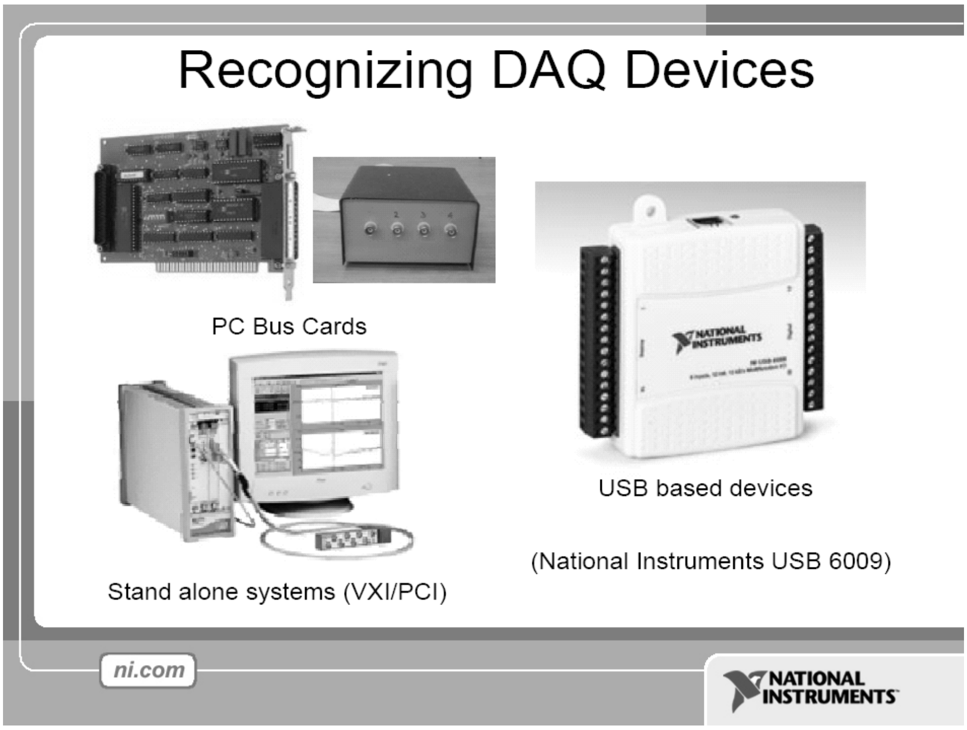
Analogue signal. Physical property being exists as a voltage signal in a wire. (Output of almost all sensor systems.)

A/D converter (measurement)

D/A converter (control)

Digital signal. Physical property is represented as a series of numbers in a computer (Most useful form for signal.)

These two devices are often packaged together and are usually referred to as DAQ cards, or DAQ devices, or Data Acquisition Systems, or Digitizers, or A/D converters



Recognizing DAQ Devices

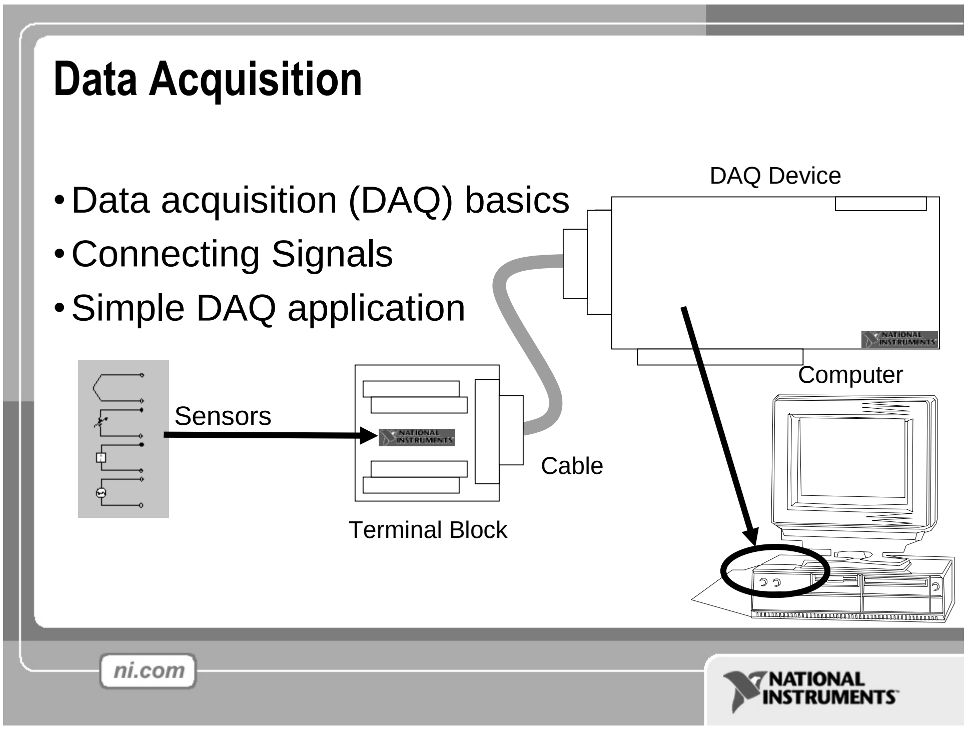
PC Bus Cards

Stand alone systems (VXI/PCI)

USB based devices

(National Instruments USB 6009)

ni.com



Data Acquisition

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

DAQ Device

Computer

Sensors

Cable

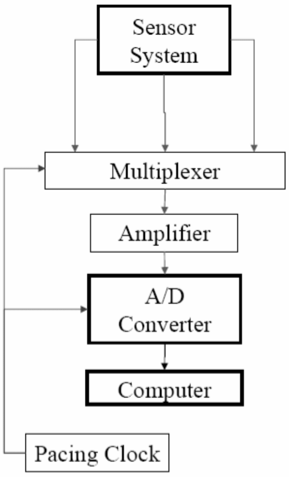
Terminal Block

ni.com

Multiple sensors in practical applications

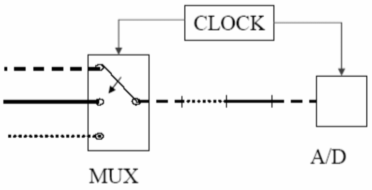
A/D Converter Limitations

□Only sample one channel

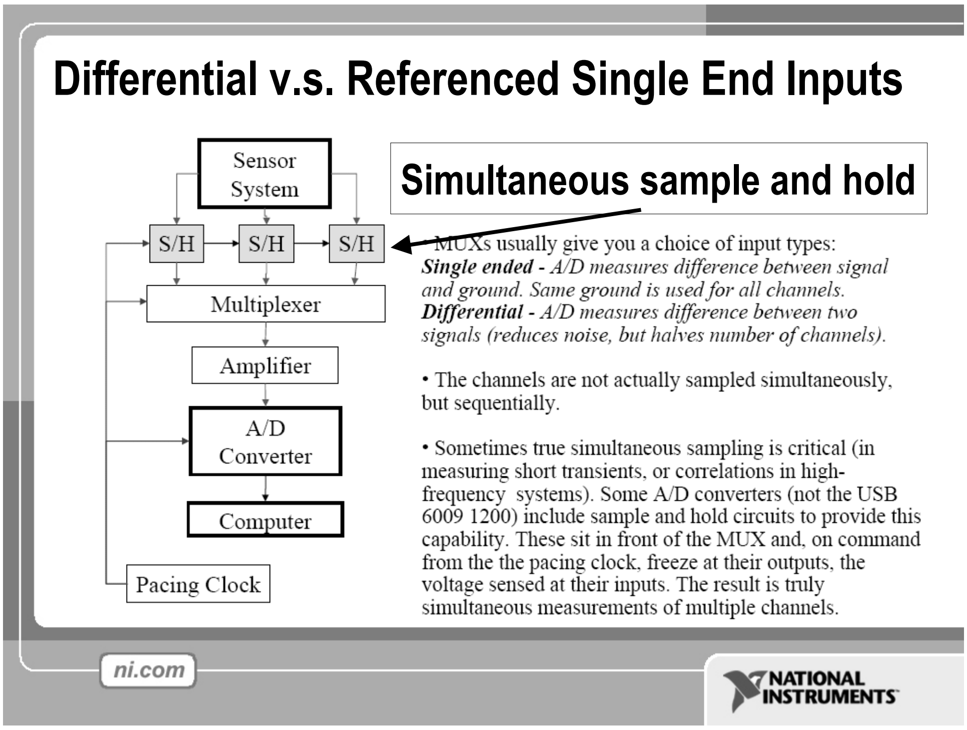


• It is common to want to make measurements of several sensor outputs at the same time.

• To do this, without using multiple A/D converters (an expensive and unnecessary solution), requires a device known as a multiplexer (or MUX). The MUX is essentially an analogue switch that creates a composite signal containing short bursts of each of the signal channels to be measured.



• The MUX is synchronized with the A/D converter using the pacing clock



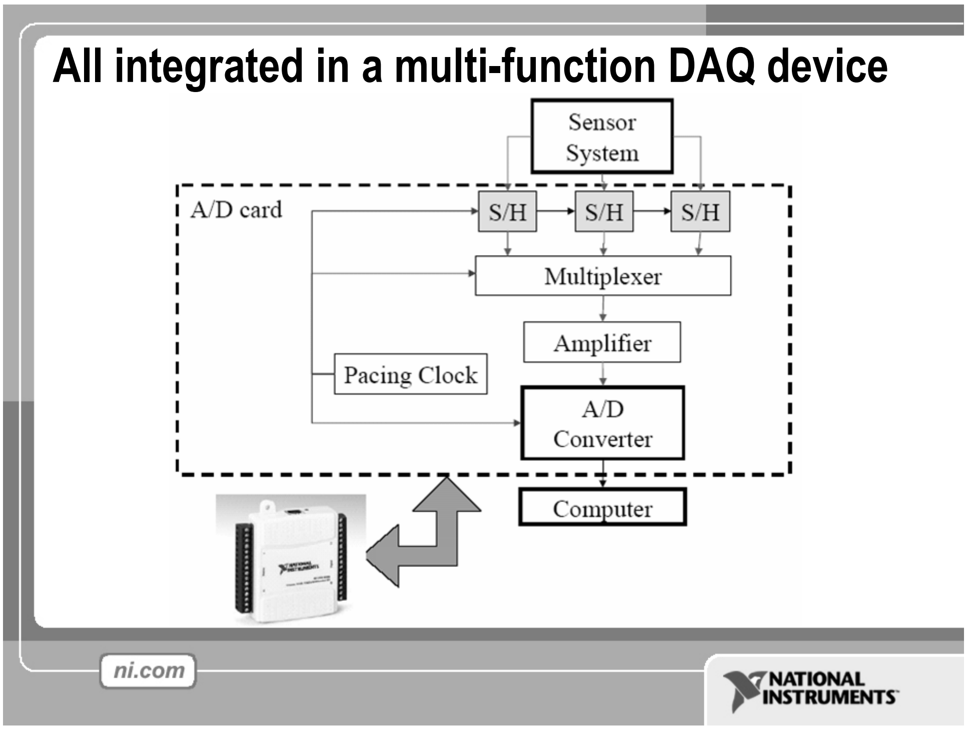
Differential v.s. Referenced Single End Inputs

- MUXs usually give you a choice of input types:

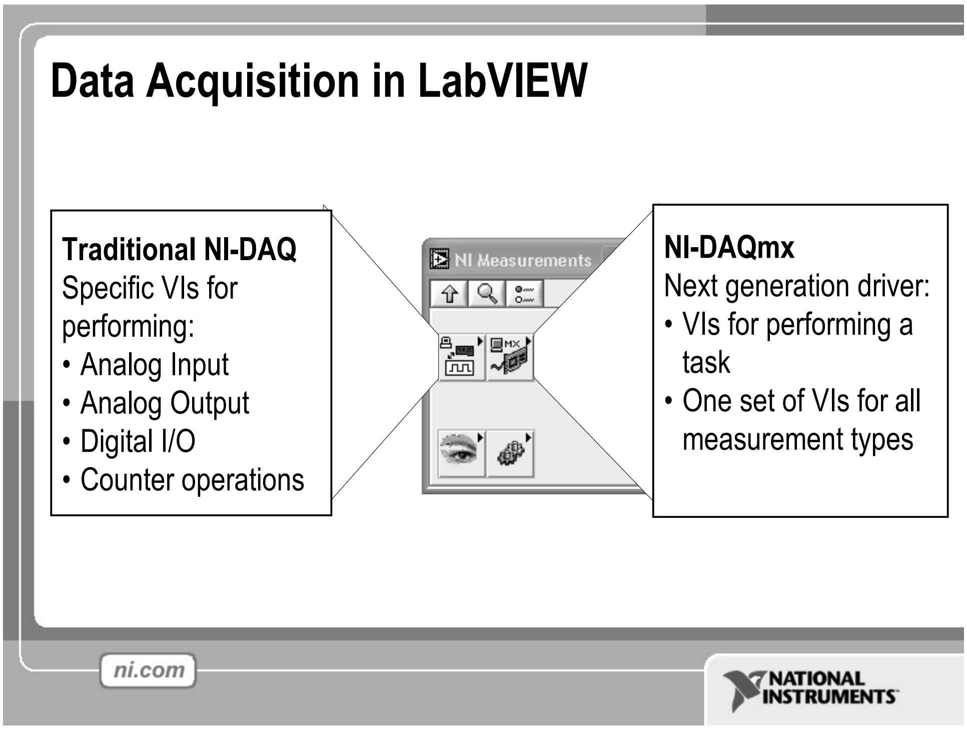
***Single ended** - A/D measures difference between signal and ground. Same ground is used for all channels.*

***Differential** - A/D measures difference between two signals (reduces noise, but halves number of channels).*

- The channels are not actually sampled simultaneously, but sequentially.
- Sometimes true simultaneous sampling is critical (in measuring short transients, or correlations in high-frequency systems). Some A/D converters (not the USB 6009 1200) include sample and hold circuits to provide this capability. These sit in front of the MUX and, on command from the the pacing clock, freeze at their outputs, the voltage sensed at their inputs. The result is truly simultaneous measurements of multiple channels.



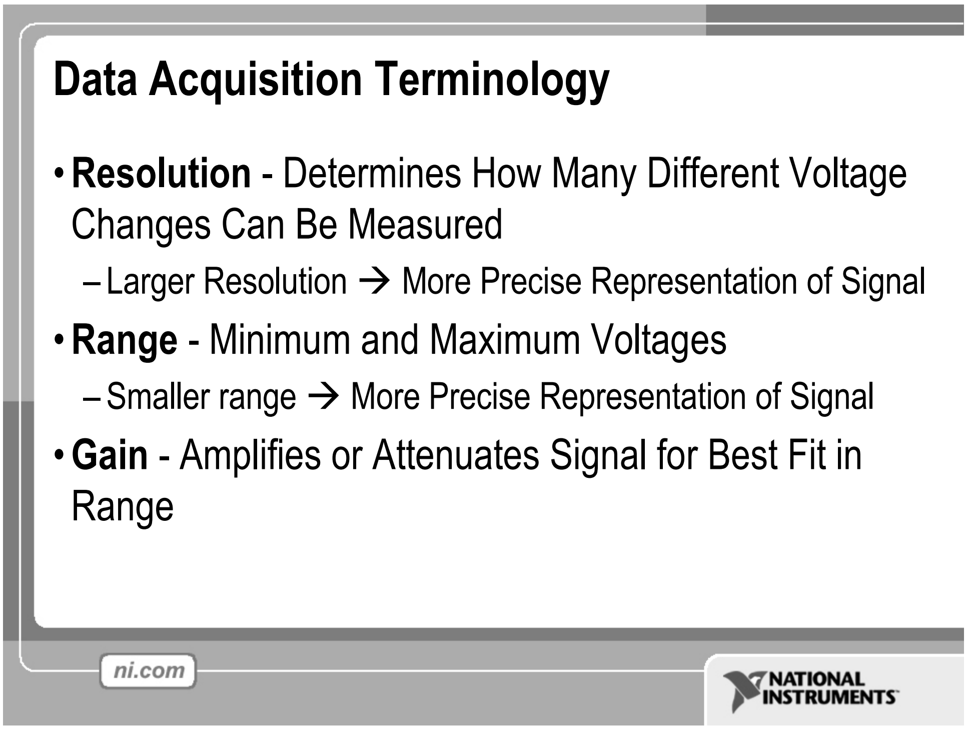
All integrated in a multi-function DAQ device



Data Acquisition in LabVIEW

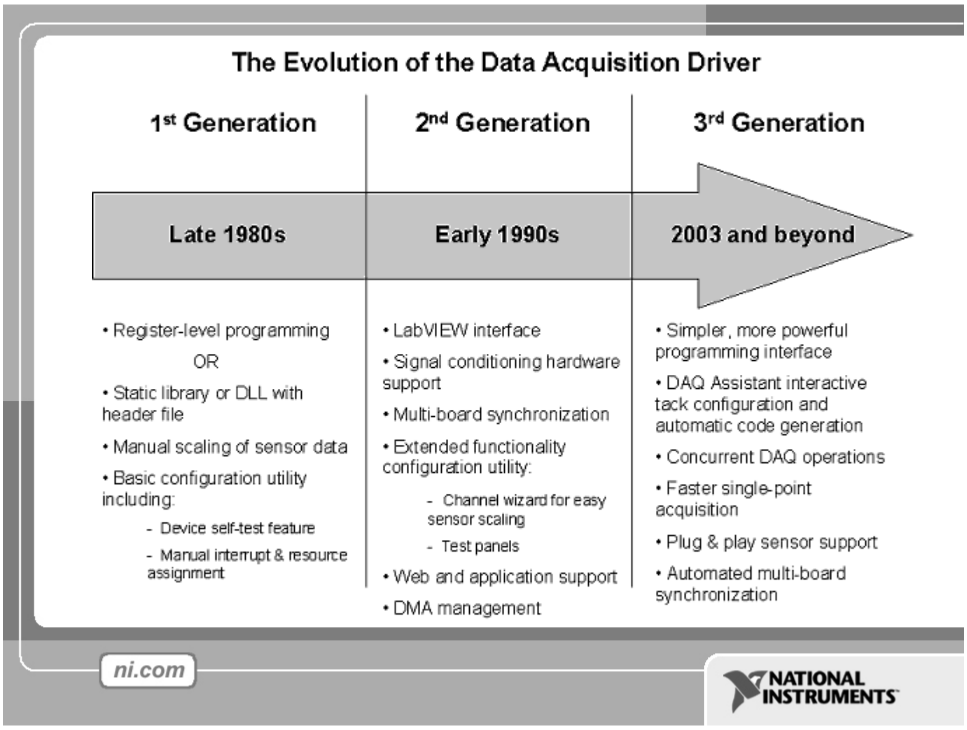
Traditional NI-DAQ
Specific VIs for performing:
- Analog Input
- Analog Output
- Digital I/O
- Counter operations

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



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



The Evolution of the Data Acquisition Driver

1st Generation

Late 1980s

- Register-level programming

  OR
- Static library or DLL with header file
- Manual scaling of sensor data
- Basic configuration utility including:
  - Device self-test feature
  - Manual interrupt & resource assignment

2nd Generation

Early 1990s

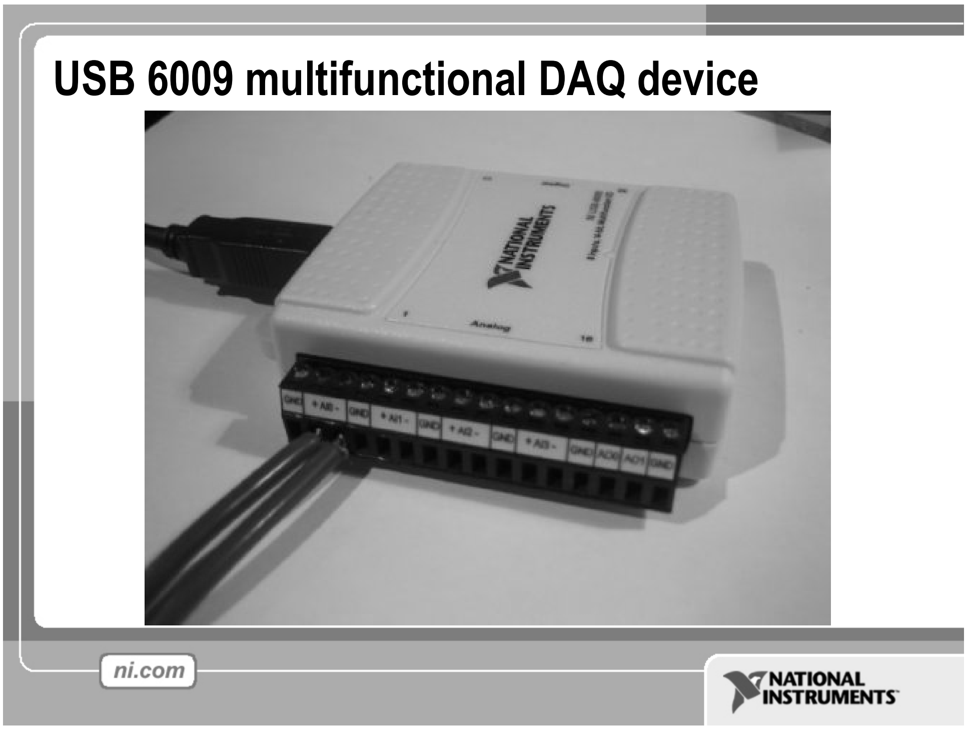
- LabVIEW interface
- Signal conditioning hardware support
- Multi-board synchronization
- Extended functionality configuration utility:
  - Channel wizard for easy sensor scaling
  - Test panels
- Web and application support
- DMA management

3rd Generation

2003 and beyond

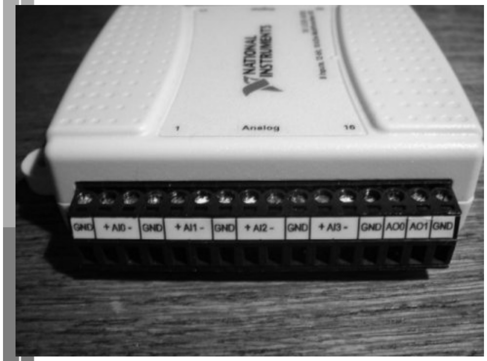
- Simpler, more powerful programming interface
- DAQ Assistant interactive tack configuration and automatic code generation
- Concurrent DAQ operations
- Faster single-point acquisition
- Plug & play sensor support
- Automated multi-board synchronization

ni.com

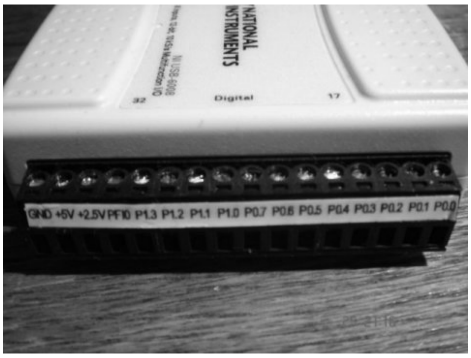


USB 6009 multifunctional DAQ device

ni.com

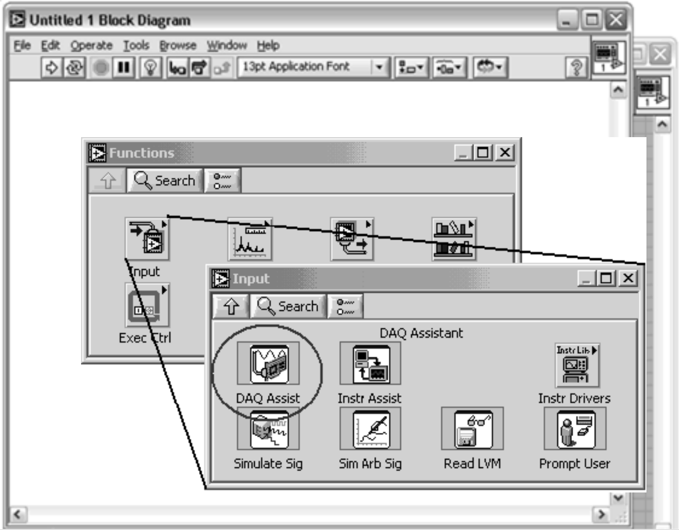


Analog in/out

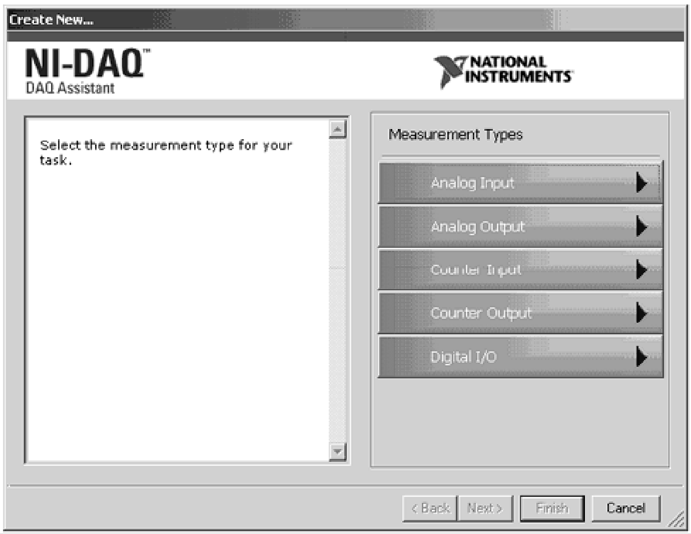


Digital in/out

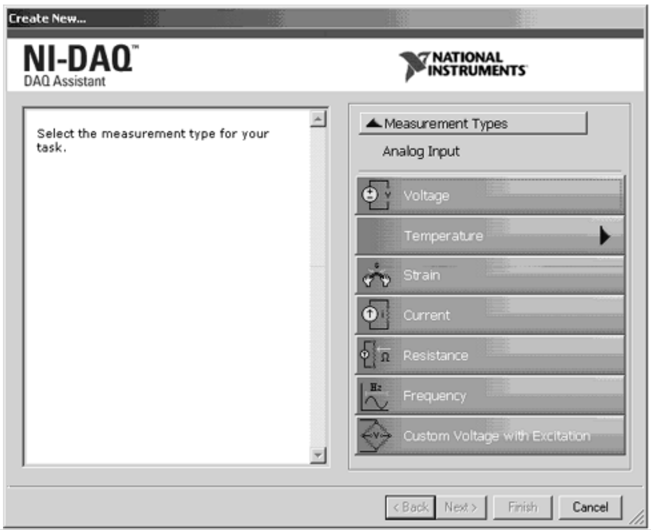
Use of DAQ assistant



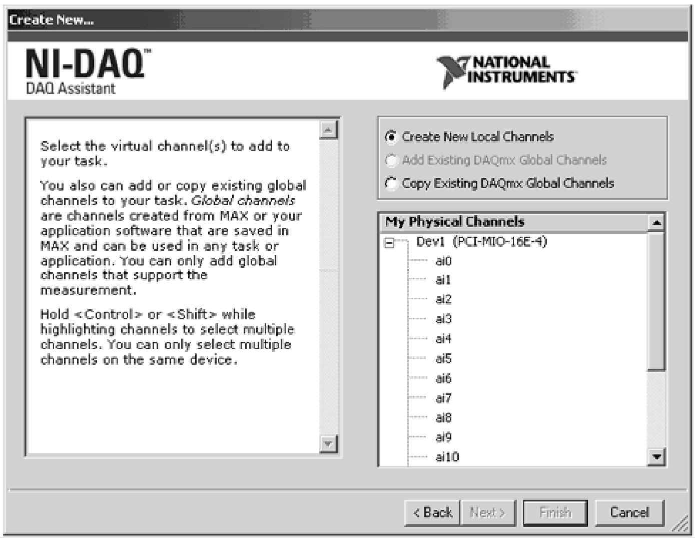
Defining your task


Select the measurement type for your task.

Configuring your task (analog input)


Select the measurement type for your task.

Appointing Channels to acquire from



Create New...

NI-DAQ™ DAQ Assistant

Select the virtual channel(s) to add to your task.

You also can add or copy existing global channels to your task. *Global channels* are channels created from MAX or your application software that are saved in MAX and can be used in any task or application. You can only add global channels that support the measurement.

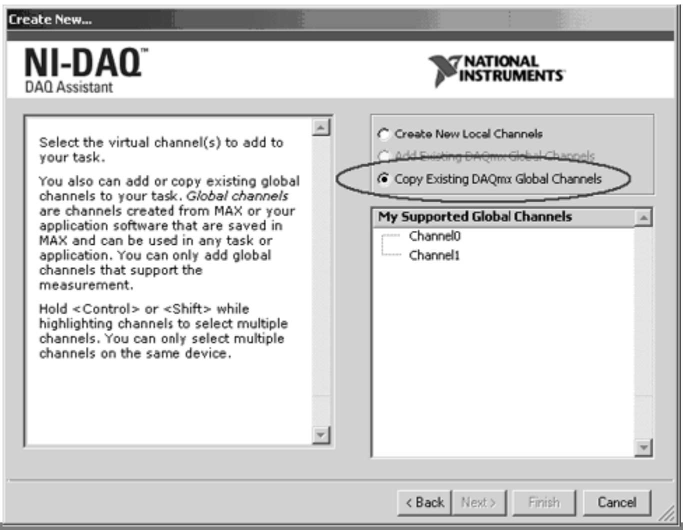
Hold <Control> or <Shift> while highlighting channels to select multiple channels. You can only select multiple channels on the same device.

● Create New Local Channels
○ Add Existing DAQmx Global Channels
○ Copy Existing DAQmx Global Channels

My Physical Channels

- Dev1 (PCI-MIO-16E-4)
  - ai0
  - ai1
  - ai2
  - ai3
  - ai4
  - ai5
  - ai6
  - ai7
  - ai8
  - ai9
  - ai10

Physical channels versus Global channels



Create New...

NI-DAQ™ DAQ Assistant

Select the virtual channel(s) to add to your task.

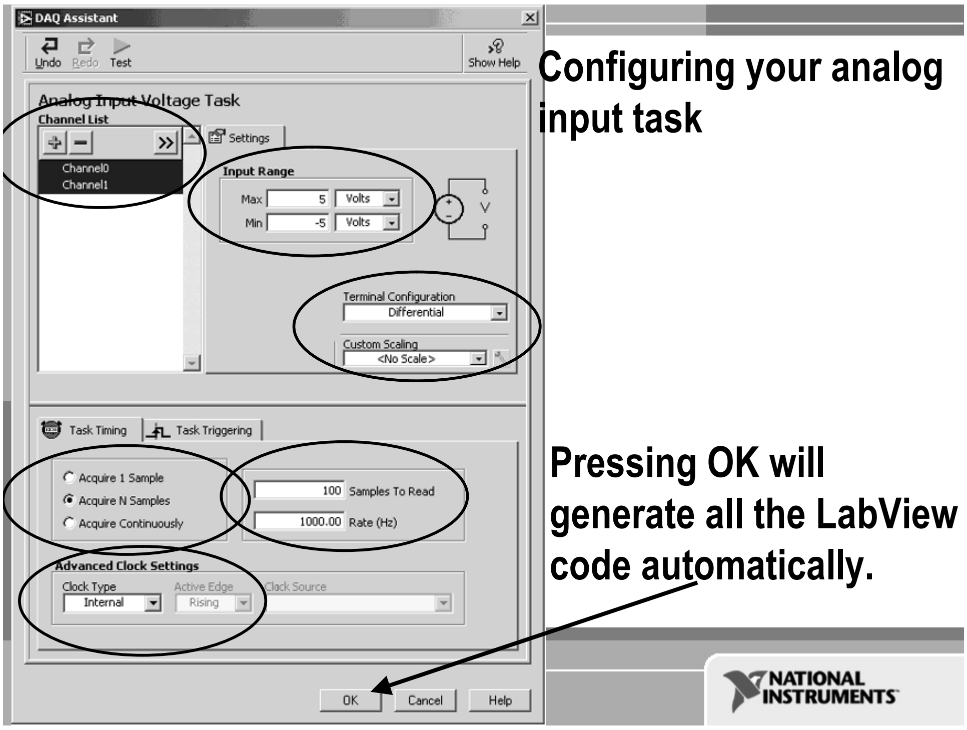
You also can add or copy existing global channels to your task. *Global channels* are channels created from MAX or your application software that are saved in MAX and can be used in any task or application. You can only add global channels that support the measurement.

Hold <Control> or <Shift> while highlighting channels to select multiple channels. You can only select multiple channels on the same device.

○ Create New Local Channels
○ Add Existing DAQmx Global Channels
● Copy Existing DAQmx Global Channels

My Supported Global Channels

- Channel0
- Channel1



Configuring your analog input task

DAQ Assistant

Analog Input Voltage Task

Channel List: Channel0, Channel1

Input Range: Max 5 Volts; Min -5 Volts

Terminal Configuration: Differential

Custom Scaling: <No Scale>

Task Timing

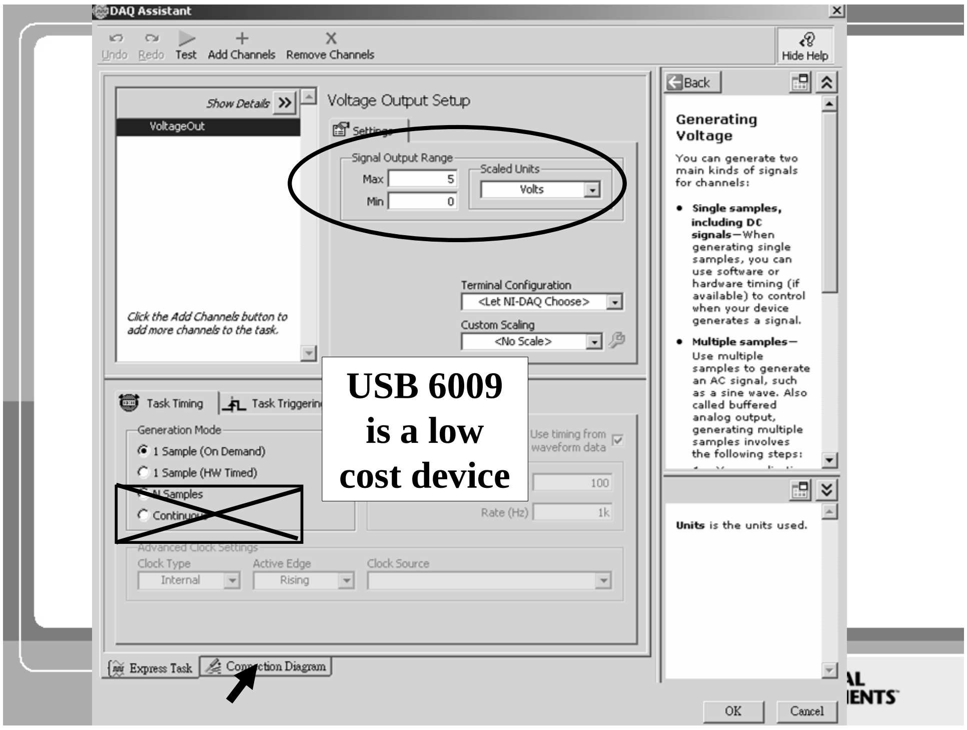
○ Acquire 1 Sample
● Acquire N Samples
○ Acquire Continuously

100 Samples To Read

1000.00 Rate (Hz)

Advanced Clock Settings: Clock Type Internal; Active Edge Rising; Clock Source

Pressing OK will generate all the LabView code automatically.



DAQ Assistant

VoltageOut

Voltage Output Setup

Signal Output Range: Max 5; Min 0

Scaled Units: Volts

Terminal Configuration: <Let NI-DAQ Choose>

Custom Scaling: <No Scale>

Click the Add Channels button to add more channels to the task.

Generation Mode

● 1 Sample (On Demand)
○ 1 Sample (HW Timed)
○ N Samples
○ Continuous

Use timing from waveform data ☑

100

Rate (Hz) 1k

Advanced Clock Settings: Clock Type Internal; Active Edge Rising; Clock Source

Generating Voltage

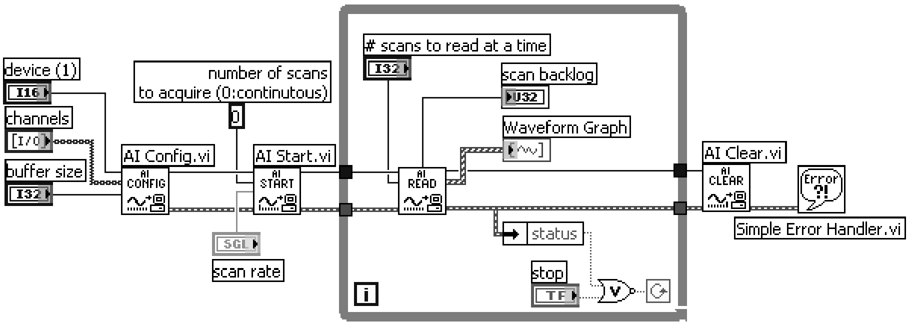
You can generate two main kinds of signals for channels:

- **Single samples, including DC signals**—When generating single samples, you can use software or hardware timing (if available) to control when your device generates a signal.
- **Multiple samples**—Use multiple samples to generate an AC signal, such as a sine wave. Also called buffered analog output, generating multiple samples involves the following steps:

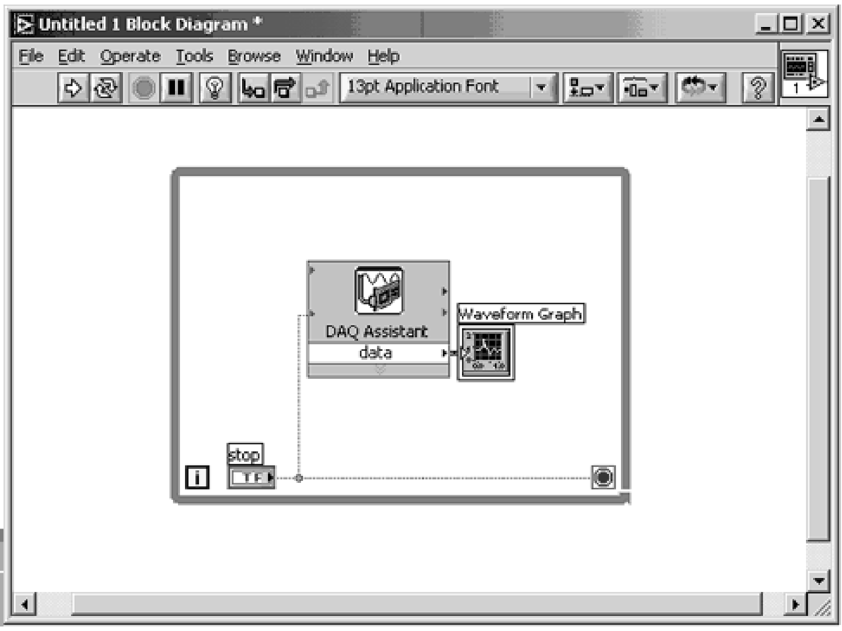
Units is the units used.

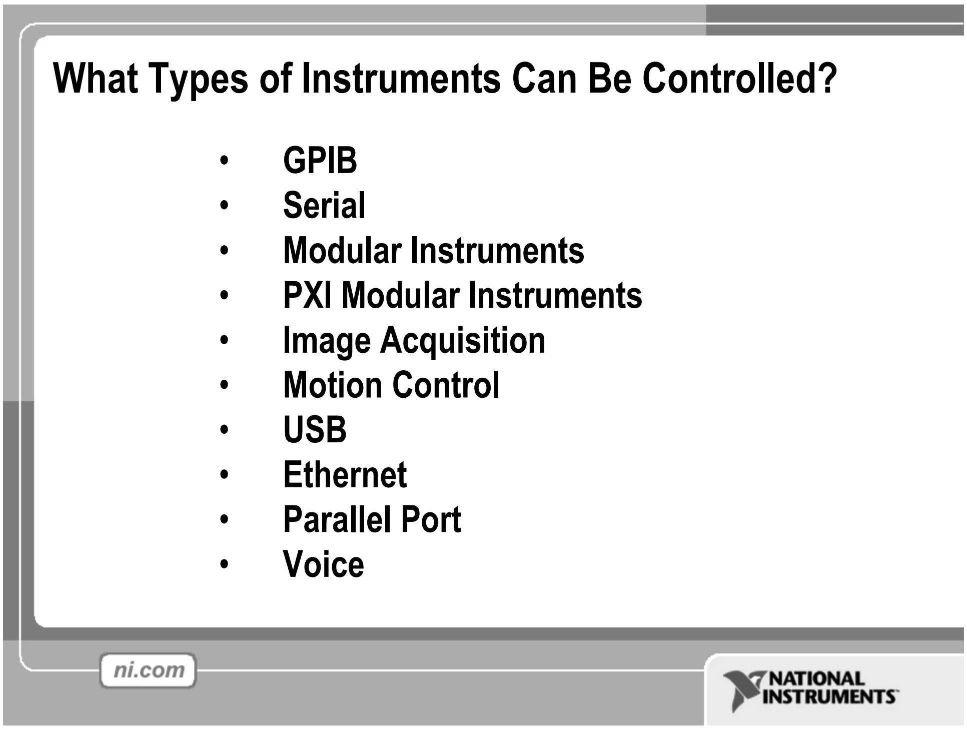
USB 6009 is a low cost device

A typical DAQ VI using traditional programming



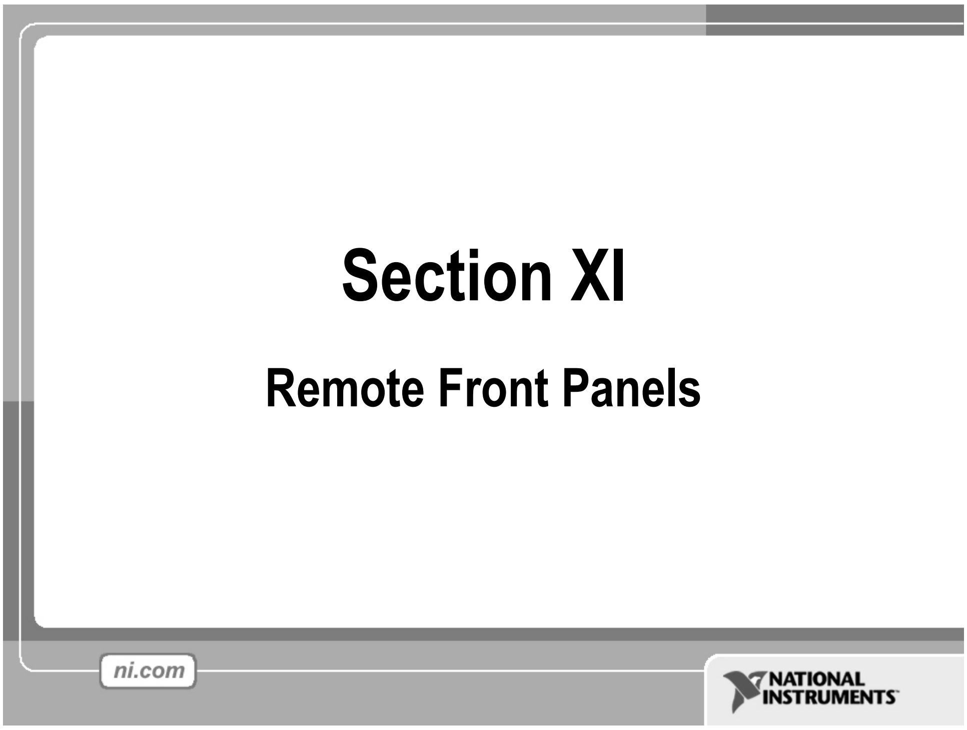
With DAQ assistant, the whole VI can be as simple as:


Untitled 1 Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
DAQ Assistant
data
Waveform Graph
stop
TF



What Types of Instruments Can Be Controlled?

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



Section XI

Remote Front Panels

Section XI – Remote Front Panels

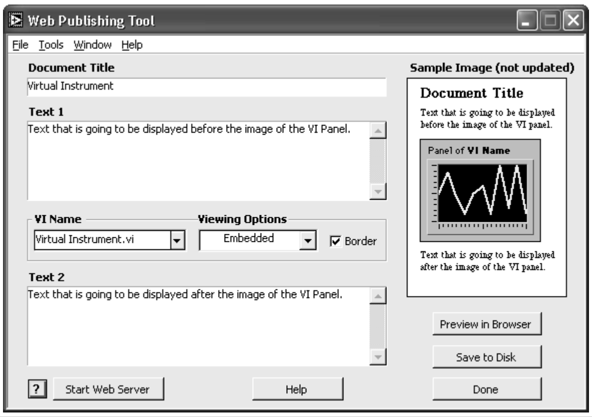
- View & Control LabVIEW Front Panels from a Web Browser
- Requires no programming
- Remote clients see “live” front panel updates
- Multiple clients can view the same panel simultaneously
- Only one client can control the front panel at a time

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



Web Publishing Tool

Document Title: Virtual Instrument

Text 1: Text that is going to be displayed before the image of the VI Panel.

VI Name: Virtual Instrument.vi

Viewing Options: Embedded ☑ Border

Text 2: Text that is going to be displayed after the image of the VI Panel.

Sample Image (not updated)

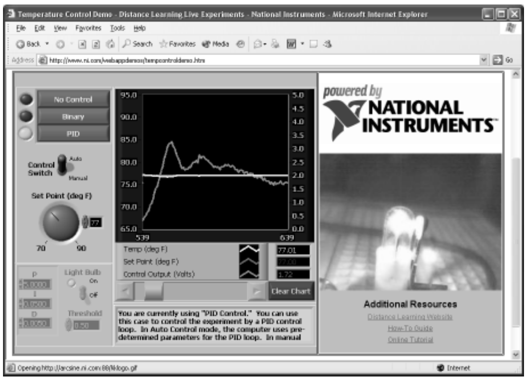
Document Title

Text that is going to be displayed before the image of the VI panel.

Text that is going to be displayed after the image of the VI panel.

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

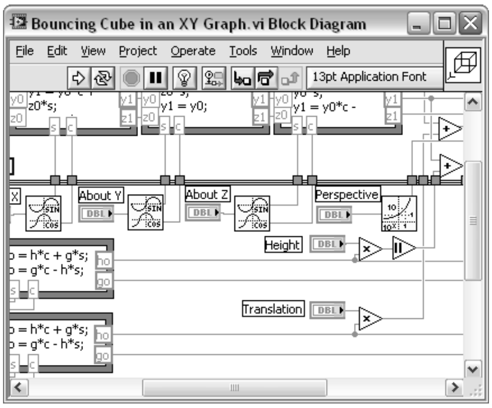

Temperature Control Demo - Distance Learning Live Experiments - National Instruments - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.ni.com/webappdemos/tempcontroldemo.htm
Go
No Control
Binary
PID
Control Switch
Auto
Manual
Set Point (deg F)
77
70
90
95.0
90.0
85.0
80.0
75.0
70.0
65.0
539
639
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Temp (deg F)
Set Point (deg F)
Control Output (Volts)
77.01
1.72
Clear Chart
P
I
D
Light Bulb
On
Off
Threshold
0.50
powered by NATIONAL INSTRUMENTS
Additional Resources
Distance Learning Website
How-To Guide
Online Tutorial
Opening http://arcsine.ni.com:88/lklogo.gif
Internet

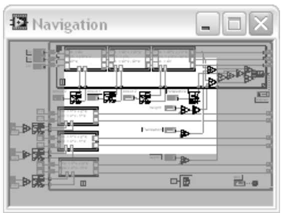
You are currently using "PID Control." You can use this case to control the experiment by a PID control loop. In Auto Control mode, the computer uses pre-determined parameters for the PID loop. In manual

Section XII – Large Program Development and Additional Topics

A. Navigation Window
B. LabVIEW Project
C. Shared Variable
D. Additional topics
   - Property Nodes
   - Local Variables
   - Global Variables
   - DataSocket
   - Binary File I/O

LabVIEW Navigation Window

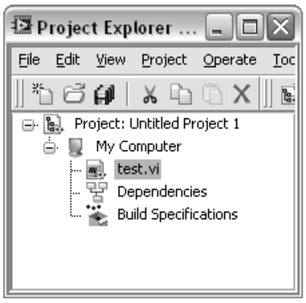




- Shows the current region of view compared to entire Front Panel or Block Diagram
- Great for large programs

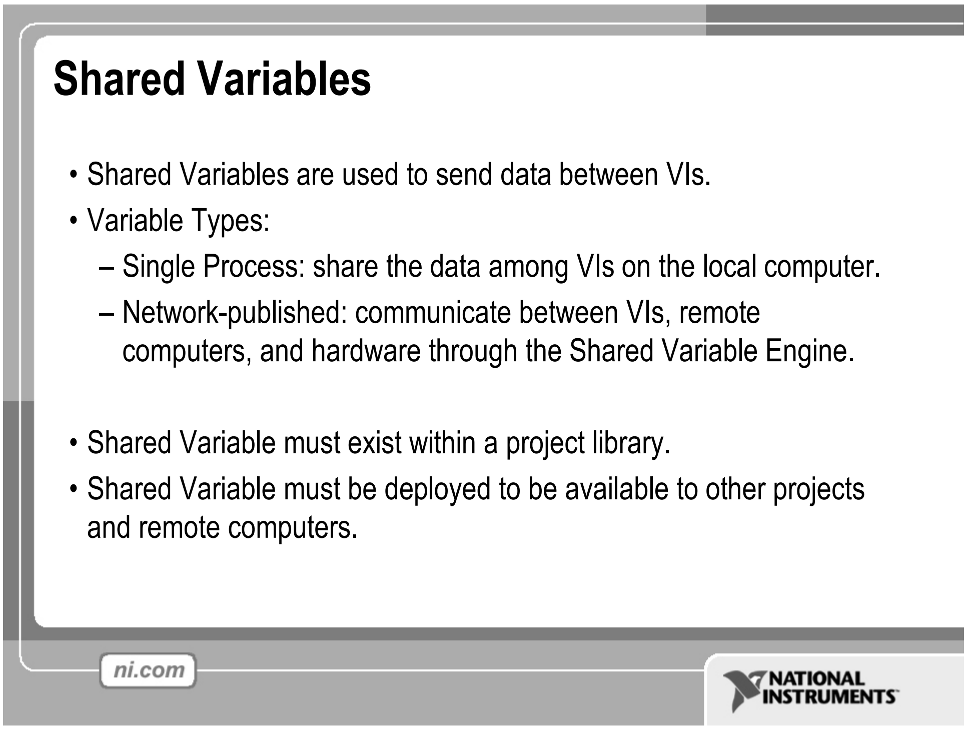
* Organize and reduce program visual size with subVIs

LabVIEW Project



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

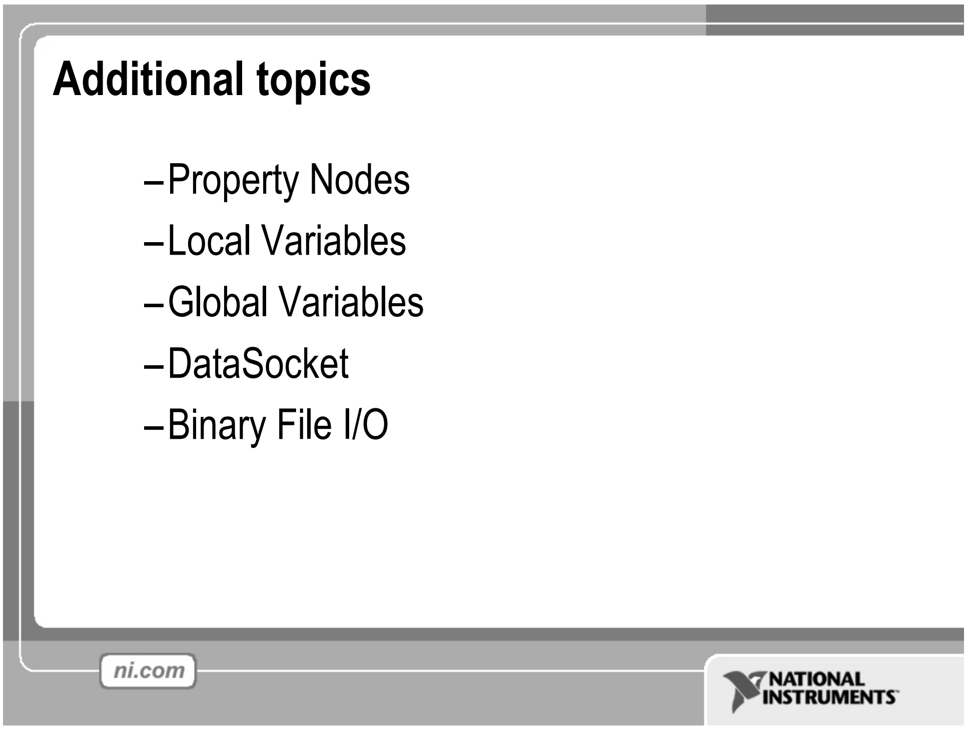
(**LabVIEW»Project»New**)



Shared Variables

- 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 Shared Variable Engine.

- Shared Variable must exist within a project library.
- Shared Variable must be deployed to be available to other projects and remote computers.



Additional topics

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

Additional Resources

- NI Academic Web & Student Corner
  - http://www.ni.com/academic
- Connexions: Full LabVIEW Training Course
  - www.cnx.rice.edu
  - Or search for "LabVIEW basics"

connexions℠ sharing knowledge and building communities

Home

RELATED MATERIAL

Similar content

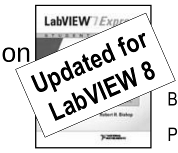
- LabVIEW Graphical Programming Course
- LabVIEW Documentation Resources
- Data Acquisition in LabVIEW

LabVIEW Envir

By: NATIONAL INSTRUMENTS

Summary: In this section, you w

- LabVIEW Certification
  - LabVIEW Fundamentals Exam (free on www.ni.com/academic)
  - Certified LabVIEW Associate Developer Exam (industry recognized certification)
- Get your own copy of LabVIEW Student Edition
  - www.ni.com/academic



Updated for LabVIEW 8

By Robert H Bishop.

Published by Prentice Hall.

The LabVIEW Certification Program

Architect

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

Developer

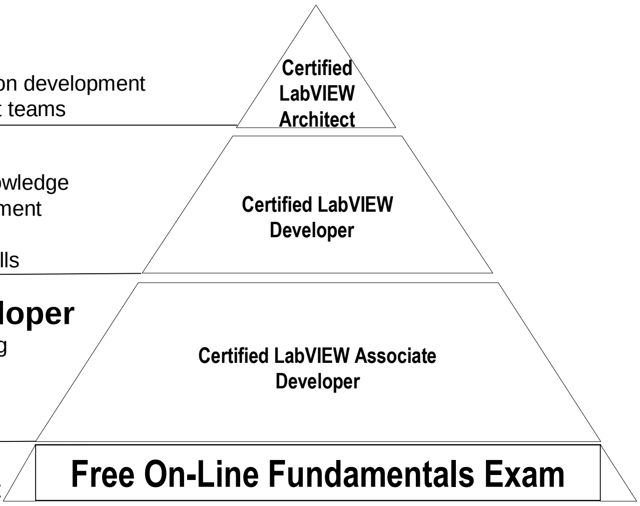
- Advanced LabVIEW knowledge and application development experience
- Project management skills

Associate Developer

- Proficiency in navigating LabVIEW environment
- Some application development experience

Fundamentals Exam

- Pre-Certification Skills Test

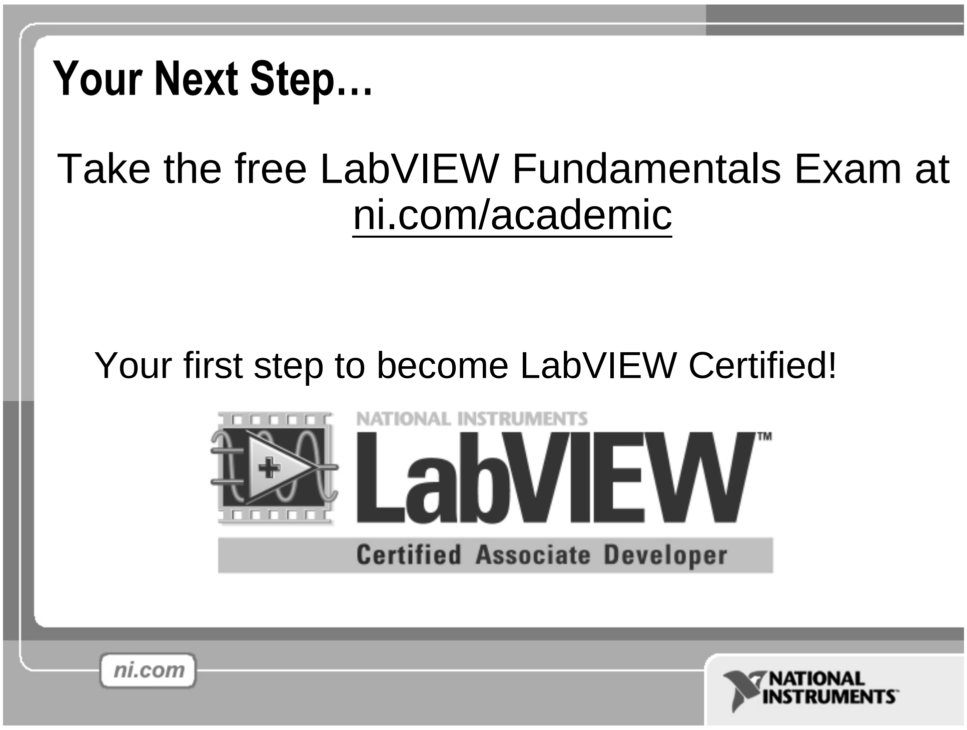


Certified LabVIEW Architect

Certified LabVIEW Developer

Certified LabVIEW Associate Developer

Free On-Line Fundamentals Exam



Your Next Step…

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

Your first step to become LabVIEW Certified!

NATIONAL INSTRUMENTS LabVIEW™

Certified Associate Developer

ni.com

NATIONAL INSTRUMENTS