Introduction to LabVIEW

GRAPHICAL PROGRAMMING
FOR ENGINEERS AND SCIENTISTS

By Matt Downey – National instruments
Course Goals

• Become comfortable with the LabVIEW environment and data flow execution
• Ability to use LabVIEW to solve problems
• LabVIEW Concepts
  – Acquiring, saving and loading data
  – Find and use math and complex analysis functions
  – Work with data types, such as arrays and clusters
  – Displaying and printing results
The Virtual Instrumentation Approach
LabVIEW Graphical Development System

• Graphical Programming Environment
• Compile code for multiple OS and devices
• Useful in a broad range of applications
Virtual Instrumentation Applications

- **Design**
  - Signal and Image Processing
  - Embedded System Programming
    - (PC, DSP, FPGA, Microcontroller)
  - Simulation and Prototyping
  - And more…

- **Control**
  - Automatic Controls and Dynamic Systems
  - Mechatronics and Robotics
  - And more…

- **Measurements**
  - Circuits and Electronics
  - Measurements and Instrumentation
  - And more…

A single graphical development platform

Design — Prototype — Deploy
The NI Approach – Integrated Hardware Platforms

- PXI Modular Instrumentation
- Desktop PC
- Laptop PC
- PDA

- High-Speed Digitizers
- High-Resolution Digitizers and DMMs
- Multifunction Data Acquisition
- Dynamic Signal Acquisition
- Instrument Control
- Digital I/O
- Counter/Timers
- Machine Vision
- Motion Control
- Distributed I/O and Embedded Control

- Signal Conditioning and Switching
- Unit Under Test

Source: ni.com
Section I – LabVIEW Environment

A. Getting Data into your Computer
   • Data Acquisition Devices
     – NI-DAQ
     – Simulated Data Acquisition
     – Sound Card

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

C. Components of a LabVIEW Application
   • Creating a VI
   • Data Flow Execution

D. Additional Help
   • Finding Functions
   • Tips for Working in LabVIEW
A. Setting Up Your Hardware

• Sound Card
  – Built into most computers
Exercise 1 – Setting Up Your Device

• Use Windows to:
  – Verify your Sound Card

Un-Mute Microphone
Open and Run LabVIEW
Start » All Programs » National Instruments LabVIEW 8.0

Startup Screen:

Start from a Blank VI:
New » Blank VI

or

Start from an Example:
Examples » Find Examples...
LabVIEW Programs Are Called Virtual Instruments (VIs)

Each VI has 2 Windows

Front Panel
• User Interface (UI)
  – Controls = Inputs
  – Indicators = Outputs

Block Diagram
• Graphical Code
  – Data travels on wires from controls through functions to indicators
  – Blocks execute by Dataflow
Controls Palette (Controls & Indicators)

Control: Numeric

Indicator: Numeric Slide

Customize Palette View

(Place items on the Front Panel Window)
Functions (and Structures) Palette

(Place items on the Block Diagram Window)

Structure: While Loop
Tools Palette

- Recommended: Automatic Selection Tool
- Tools to operate and modify both front panel and block diagram objects

Automatically chooses among the following tools:

- Operating Tool
- Positioning/Resizing Tool
- Labeling Tool
- Wiring Tool
Status Toolbar

- Run Button
- Continuous Run Button
- Abort Execution

Additional Buttons on the Diagram Toolbar

- Execution Highlighting Button
- Retain Wire Values Button
- Step Function Buttons
Demonstration 1: Creating a VI

Front Panel Window

- Boolean Control
- Graph Indicator

Block Diagram Window

- Input Terminals
- Output Terminal
Dataflow Programming

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

• Finding Errors

  Click on broken Run button. Window showing error appears.

• Execution Highlighting

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

• Probes

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

  You can also select Probe tool from Tools palette and click on wire.
Exercise 2 – Acquiring a Signal with the Sound Card

• Use LabVIEW to:
  – Acquire a signal from your sound card

This exercise should take 15 minutes.
Context Help Window

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

Additional Help

- Right-Click on the VI icon and choose Help, or
- Choose “Detailed Help.” on the context help window
Tips for Working in LabVIEW

• Keystroke Shortcuts
  – <Ctrl+H>  – Activate/Deactivate Context Help Window
  – <Ctrl+B>  – Remove Broken Wires From Block Diagram
  – <Ctrl+E>  – Toggle Between Front Panel and Block Diagram
  – <Ctrl+Z>  – Undo (Also in Edit Menu)

• Tools » Options…  – Set Preferences in LabVIEW

• VI Properties  – Configure VI Appearance, Documentation, etc.
Section II – Elements of Typical Programs

A. Loops
   • While Loop
   • For Loop

B. Functions and SubVIs
   • Types of Functions
   • Creating Custom Functions (SubVI)
   • Functions Palette & Searching

C. Decision Making and File IO
   • Case Structure
   • Select (simple If statement)
   • File I/O
Loops

• While Loops
  – Terminal counts iteration
  – Always runs at least once
  – Runs until stop condition is met

• For Loops
  – Terminal counts iterations
  – Run according to input \( N \) of count terminal \( N \)
Drawing a Loop

1. Select the structure

2. Enclose code to be repeated

3. Drop or drag additional nodes and then wire
3 Types of Functions (from the Functions Palette)

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

Standard VIs: modularized VIs customized by wiring (customizable)

Functions: fundamental operating elements of LabVIEW; no front panel or block diagram (yellow)
What Types of Functions are Available?

• Input and Output
  – Signal and Data Simulation
  – 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
Searching for Controls, VIs, and Functions

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

- Enclose area to be converted into a subVI.
- Select **Edit » Create SubVI** from the Edit Menu.
LabVIEW Functions and SubVIs operate like Functions in other languages

<table>
<thead>
<tr>
<th>Function Pseudo Code</th>
<th>Calling Program Pseudo Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>function average (in1, in2, out) { out = (in1 + in2)/2.0; }</code></td>
<td><code>main { average (in1, in2, pointavg) }</code></td>
</tr>
</tbody>
</table>

**SubVI Block Diagram**

![subvi block diagram]

**Calling VI Block Diagram**

![calling vi block diagram]
Exercise 3.1 – Acquire and Analysis

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

This exercise should take 15 minutes.
How Do I Make Decisions in LabVIEW?

1. Case Structures
   - (a)
   - (b)

2. Select
   - (c)
File I/O

File I/O – passing data to and from files
- Files can be binary, text, or spreadsheet
- Write/Read LabVIEW Measurements file (*.lvm)
File I/O Programming Model – Under the hood

1. Open/Create/Replace File
2. Read and/or Write to File
3. Close File
4. Check for Errors

- Open/Create/Replace File
- Read from Text File
- Close File
- Simple Error Handler.vi
- Write to Text File
Section III – Presenting your Results

A. Displaying Data on the Front Panel
   • Controls and Indicators
   • Graphs and Charts
   • Loop Timing

B. Signal Processing
   • MathScript
   • Arrays
   • Clusters
   • Waveforms
What Types of Controls and Indicators are Available?

- **Numeric Data**
  - Number input and display
  - Analog Sliders, Dials, and Gauges
- **Boolean Data**
  - Buttons and LEDs
- **Array & 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
Charts – Add 1 data point at a time with history

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

- Chart updates with each individual point it receives

Functions » Express » Graph Indicators » Chart
Graphs – Display many data points at once

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

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

**Functions** ➔ **Express** ➔ **Graph Indicators** ➔ **Graph**

![Waveform Graph Diagram]

[Source: ni.com]
Loops can accumulate arrays at their boundaries with auto-indexing.

For Loops auto-index by default.

While Loops output only the final value by default.

Right-click tunnel and enable/disable auto-indexing.

**Auto-Indexing Enabled**
- Wire becomes thicker.
- Only one value (last iteration) is passed out of the loop.

**Auto-Indexing Disabled**
- Wire remains the same size.
- Only one value (last iteration) is passed out of the loop.
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 (i.e. Numeric Control).
How Do I Time a Loop?

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

2. Timed Loops
   - Configure special timed While loop for desired $dt$. 

![Time Delay Diagram](image)
![Timed Loop Diagram](image)
Control & Indicator Properties

- Properties are characteristics or qualities about an object
- Properties can be found by right clicking on a Control or Indicator
  - Properties Include:
    - Size
    - Color
    - Plot Style
    - Plot color
  - Features include:
    - Cursors
    - Scaling
Math with the MathScript Node

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

Prototype your equations in the interactive MathScript Window.
The Interactive MathScript Window

• Rapidly develop and test algorithms
• Share Scripts and Variables with the Node
• View /Modify Variable content in 1D, 2D, and 3D
Review of Data Types Found in LabVIEW

- Boolean
- Double Precision Number
- Integer Number
- Complex Number
- 1D Array of Doubles
- 2D Array of Doubles
- Matrix of Doubles
- Numeric 1
- Numeric 2
- Numeric 3
- Array 1
- Array 2
- Real Matrix
- String
- Waveform Cluster
- Error Cluster
- Simulate Signal
- Dynamic Data
- Sine

ni.com

National Instruments
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”

• LabVIEW Certification
  – LabVIEW Fundamentals Exam (free on www.ni.com/academic)
  – Certified LabVIEW Associate Developer Exam (industry recognized certification)
The LabVIEW Certification Program

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

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

**Associate Developer**
- Proficiency in navigating LabVIEW environment
- Some application development experience

**Fundamentals Exam**
- Pre-Certification Skills Test

**Free On-Line Fundamentals Exam**