Introduction to LabVIEW

What is LabVIEW

- Stands for:
  - Lab Virtual Instrument Engineering Workbench
- Powerful graphical programming environment
  - Measurement
  - Control
  - Acquire, analyze and store data
- Read inputs (acquire data)
- Process and display data
- Control Outputs
- Sound familiar?

LabVIEW Programs

- Called “Virtual Instruments”
  - VIs for short (“Vee Eye”)
- Uses a graphical methodology
  - Easy to tell what the program should do by inspection
  - Easy to learn with little programming background
- Extensive library of common programming functions

Quick Start

- Invoke from icon
- Getting Started Screen
- New → Blank VI

Three Parts of a Program

- Front Panel
  - Various controls and indicators
    - Inputs, outputs
    - Knobs, push buttons, switches, gauges, graphs, LEDs, etc.
- Block Diagram
  - Graphical programming source code
    - Terminals corresponding to front panel controls (lower level VIs), constants, functions
    - Data flow
      - Wires connecting terminals
- Icon
  - For using a VI as a subVI (a subroutine)
  - Shows connections for wiring the VI into other programs
  - Create hierarchy, modularity

Front Panel

- How the user interacts with the program
  - Added by drag-and-drop from controls palette
- Controls
  - Inputs from the user
  - Sources of data
- Indicators
  - Outputs to the user
  - Destinations for data

Block Diagram

- Similar to flow diagrams drawn in PLC labs
- The program source code for VIs
- Terminals
  - Entry or exit points for data (e.g. on functions)
- Nodes
  - Functions (e.g. add, subtract)
- Wires
  - Paths for data to flow

Palettes

- Tools Palette
  - Manipulate objects in a window
  - Run wires between objects
- Functions (or Controls) Palette
  - Place terminals or nodes in a window
  - *Browse through to become familiar*

LabVIEW Projects

- A way to organize files
- Good for large LabVIEW software designs
- Not needed for one or two VIs

VI Data Types

- Three types
  - Number
    - Integer, floating point
    - Various lengths: byte (8 bits), word (16 bits), long (32 bits), double precision (64 bits)
  - Boolean
    - True of false
    - AND, OR, NOT combinations
  - String
    - Alphanumeric characters
- Wires appearance automatically denotes data types being carried

Program Debug

- Execution Highlight
  - Shows motion of all data continuously
  - May slow program execution
- Step Over
  - Observe a single execution step at each node
- Step Into
  - Step into a subVI or loop and observe its execution by single steps
- Step Out
  - Step out of a subVI or loop and end its execution

Program Debug (cont'd)

- Breakpoint
  - Stop execution under a certain condition
- Probe
  - Add a probe that shows the value of a signal during execution

LabVIEW Tutorials

- Go to:
  http://attila.sdsu.edu/me295/modules/labview/intro/introduce.html
  (Link available on WebCT)
- Download exercises from lab computer
  - 8 exercises
  - 2 per day for four days

Student Edition of LabVIEW

- Available through National Instruments
  - www.ni.com
  - $96

Loops

Loop Structures

- While Loop
  - Run a portion of a VI as long as a certain condition is true/false
  - Example: On Switch
- For Loop
  - Run a portion of a VI a specified number of times
  - Example: Turn a stepper motor 200 steps

While Loop

- While loop border encloses a block of VI code
  - Block Diagram > Functions > Structures > While Loop
  - Drag a rectangle around the code of interest
- Includes a conditional terminal
  - Fed by a Boolean input
  - Determines an exit point
- Includes an iteration count
  - Counts number of times a loop has executed
  - First time is “iteration 0”
  - Can be used for calculations in the loop

For Loop

- For Loop border encloses a block of VI code
  - Block Diagram > Functions > Structures > For Loop
  - Drag a rectangle around the code of interest
- Includes a count terminal
  - Holds the pre-determined number of iterations
  - Usually a numeric constant
- Includes an iteration terminal
  - Counts the number of times the loop has been executed
  - First time is interation “0”
  - Can be used for calculations in the loop

Shift Registers

- Used with loops
- Stores data from each iteration of the loop
  - Averaging
  - Storing multiple iteration results

Waveform Charts

- Useful for watching data changes over time

Clusters

- A way to combine different pieces of data in a single connection
- Like individual wires in a cable

Sub VI

- Any VI can be used as a sub VI in another VI
  - An icon is needed to place it
  - Connectors are needed to wire to it