UiO : University of Oslo

FYS3240- 4240
Data acquisition & control

LabVIEW programming I

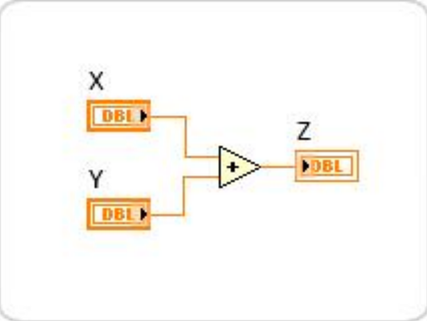
LabVIEW basics

Spring 2019 – Lecture #2

Recommended additional reading: Essick Chapter 1, 2, 3 and 4.

Bekkeng 28.12.2018

What is LabVIEW

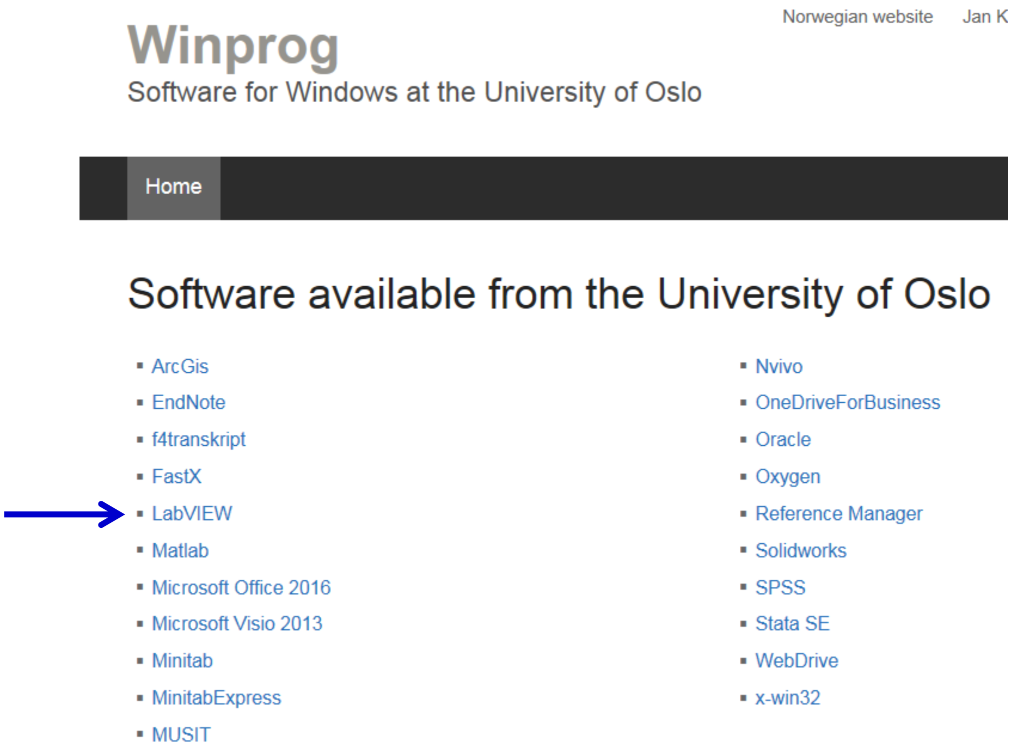


- LabVIEW is a graphical programming environment
- LabVIEW contains the same programming concepts found in most traditional languages
  - data types, loops, event handling, variables, recursion, and object-oriented programming
- LabVIEW is very common for measurement, test, data acquisition and control systems (both in industry and academia)
- Graphical icons and wires resemble a flowchart
- Provides easy integration with thousands of hardware devices
- Contains large built-in libraries
- Available for multiple targets and OS (Windows, Mac, Linux and RTOS)
- VI = Virtual Instrument.

http://www.ni.com/labview/whatis/

Install LabVIEW on own PC

- Go to https://www.winprog.uio.no/
- Select LabVIEW



Norwegian website Jan K

Winprog

Software for Windows at the University of Oslo

Home

Software available from the University of Oslo

- ArcGis
- EndNote
- f4transkript
- FastX
- LabVIEW
- Matlab
- Microsoft Office 2016
- Microsoft Visio 2013
- Minitab
- MinitabExpress
- MUSIT
- Nvivo
- OneDriveForBusiness
- Oracle
- Oxygen
- Reference Manager
- Solidworks
- SPSS
- Stata SE
- WebDrive
- x-win32

LabVIEW Hardware targets

- Standard PC
- Real-time PC
- Embedded controller (in a PXI system)
- FPGA
- 8-, 16-, and 32-bit microprocessors
- ARM Microcontrollers
  - ARM = Advanced RISC Machine

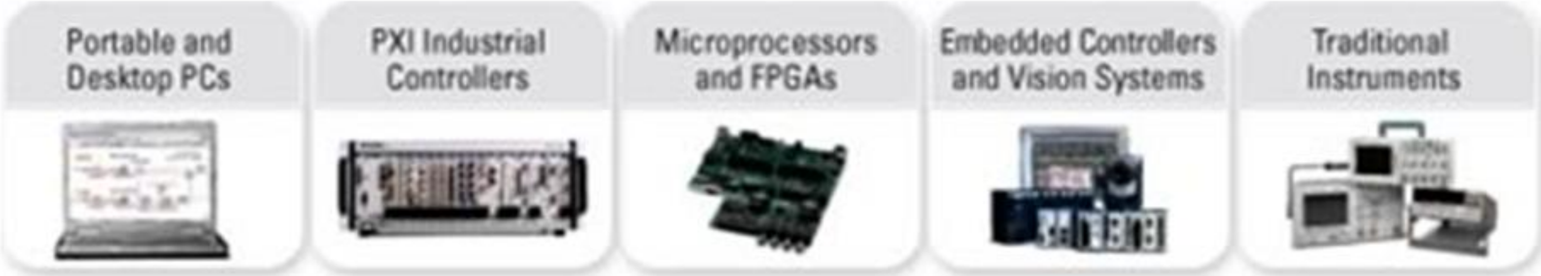
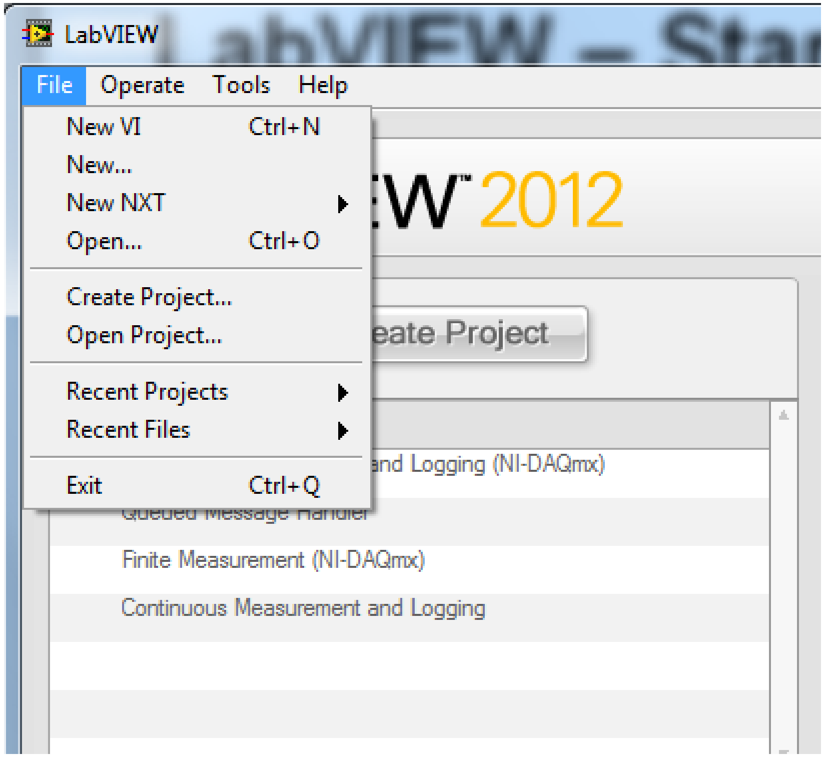

Portable and Desktop PCs
PXI Industrial Controllers
Microprocessors and FPGAs
Embedded Controllers and Vision Systems
Traditional Instruments

Figure 1. Multiple Hardware Targets

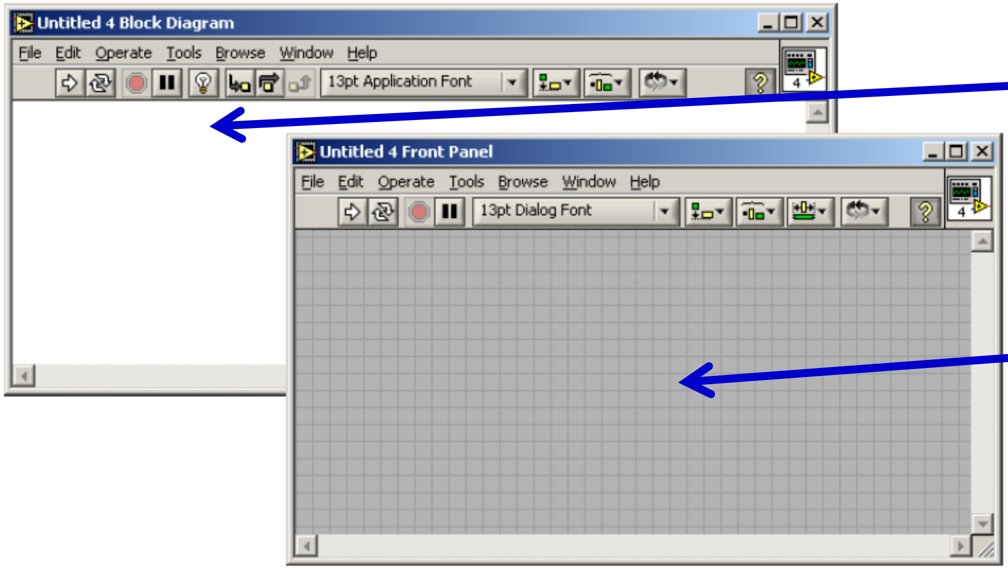
LabVIEW – Start up

- **File – New VI**



LabVIEW – Blank VI (Untitled)

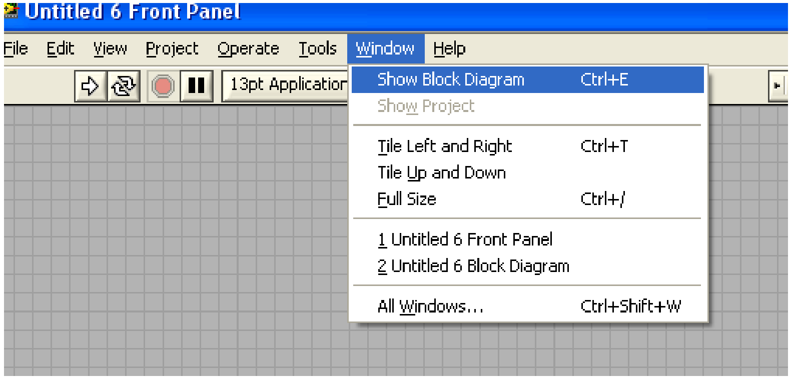
Two windows appear – ‘**Block Diagram**’ & ‘**Front Panel**’


Untitled 4 Block Diagram
File Edit Operate Tools Browse Window Help
13pt Application Font
Untitled 4 Front Panel
File Edit Operate Tools Browse Window Help
13pt Dialog Font

The block diagram contains the graphical source code

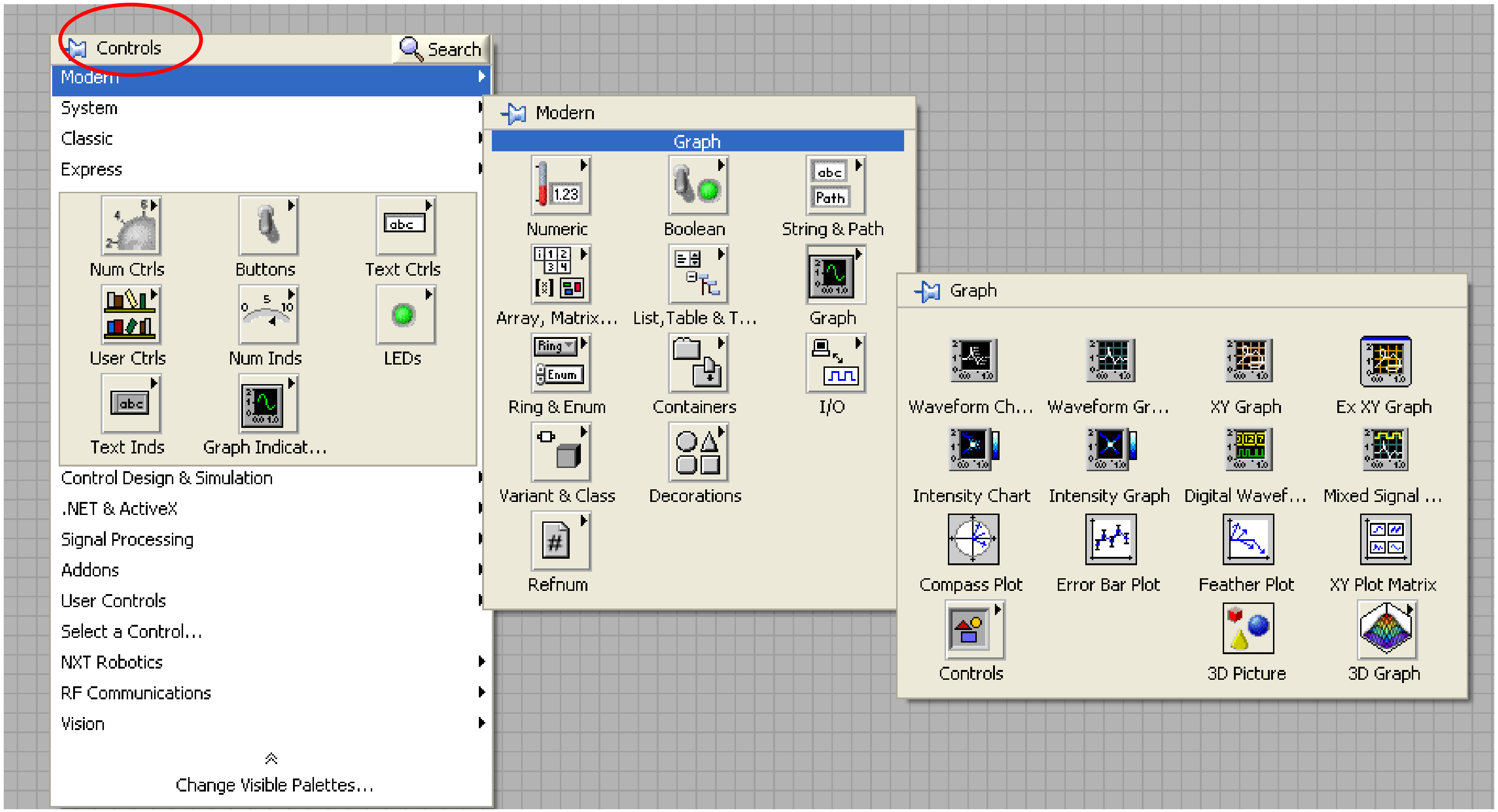
The front panel is the **user interface (GUI)** of a VI. You build the front panel by using controls and indicators

Select window to show:


Untitled 6 Front Panel
File Edit View Project Operate Tools Window Help
13pt Application
Show Block Diagram Ctrl+E
Show Project
Tile Left and Right Ctrl+T
Tile Up and Down
Full Size Ctrl+/
1 Untitled 6 Front Panel
2 Untitled 6 Block Diagram
All Windows... Ctrl+Shift+W

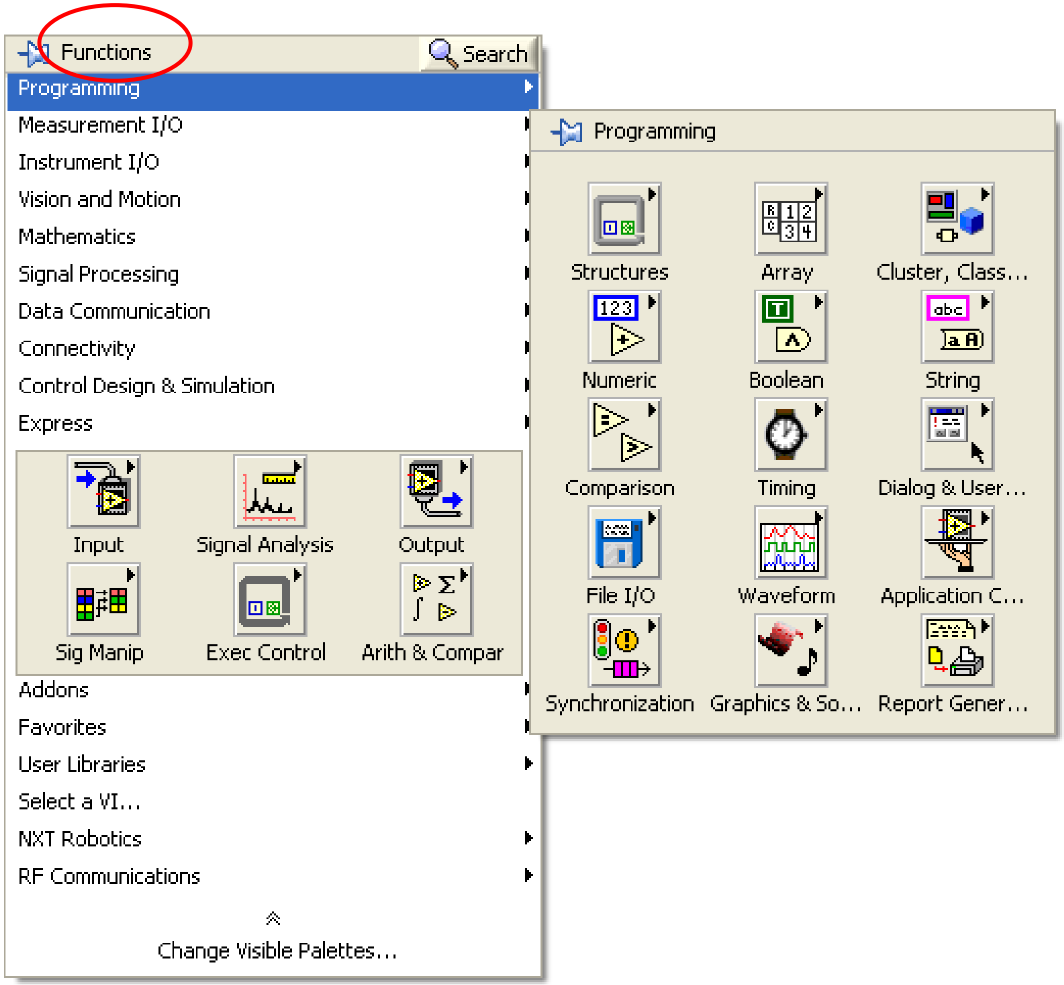
LabVIEW – Front Panel (the GUI)

Right mouse click to open important ‘**Controls**’ palette:



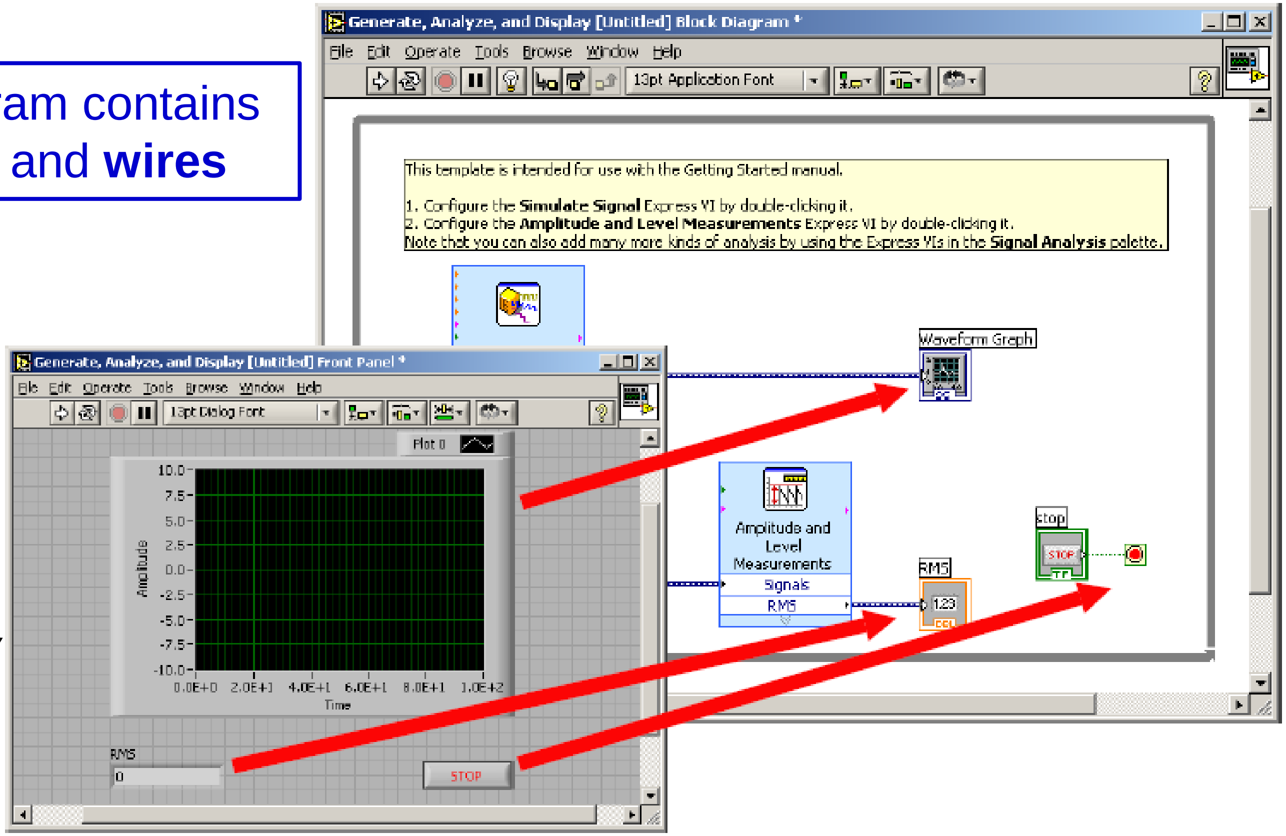
LabVIEW– Block Diagram

Right mouse click to open ‘**Functions**’ palette:



Relationship between Block Diagram and Front Panel

The block diagram contains **Icons** (objects) and **wires**



Generate, Analyze, and Display [Untitled] Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
Waveform Graph
Amplitude and Level Measurements
Signals
RMS
RMS
1.23
stop
STOP
TF
Generate, Analyze, and Display [Untitled] Front Panel *
File Edit Operate Tools Browse Window Help
13pt Dialog Font
Plot 0
10.0
7.5
5.0
2.5
0.0
-2.5
-5.0
-7.5
-10.0
Amplitude
0.0E+0
2.0E+1
4.0E+1
6.0E+1
8.0E+1
1.0E+2
Time
RMS
0
STOP

This template is intended for use with the Getting Started manual.

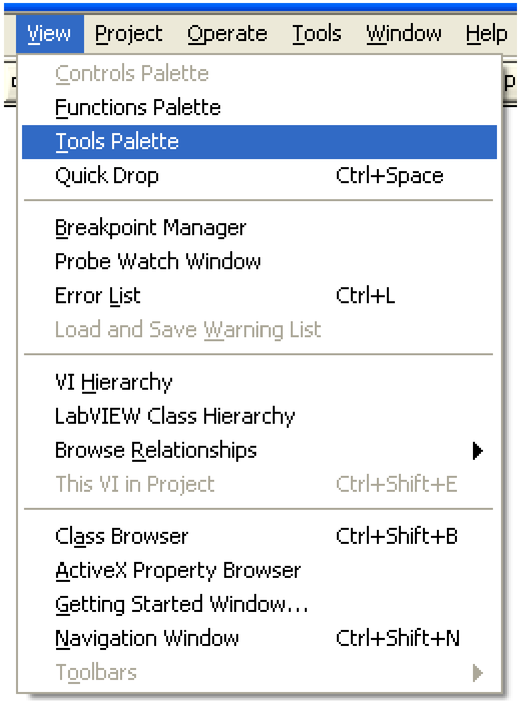
1. Configure the **Simulate Signal** Express VI by double-clicking it.
2. Configure the **Amplitude and Level Measurements** Express VI by double-clicking it.

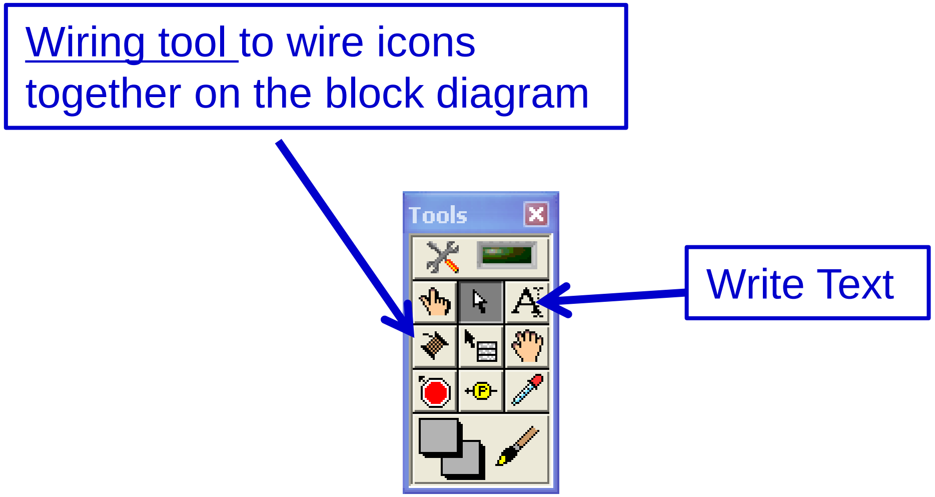
Note that you can also add many more kinds of analysis by using the Express VIs in the **Signal Analysis** palette.

GUI (Front Panel)

Tools palette

- Used on Block Diagram & Front Panel



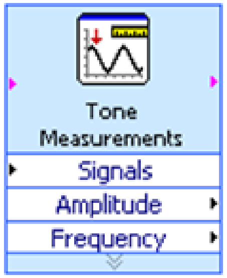


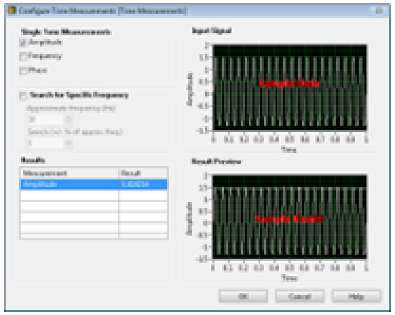
Wiring tool to wire icons together on the block diagram

Write Text

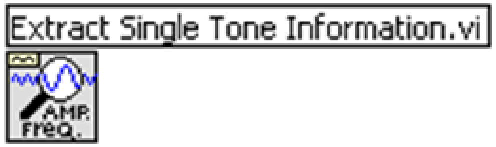
3 Types of Functions

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

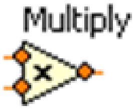




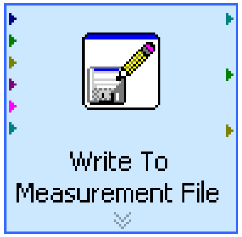
Standard VIs: modularized VIs customized by wiring (customizable)



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



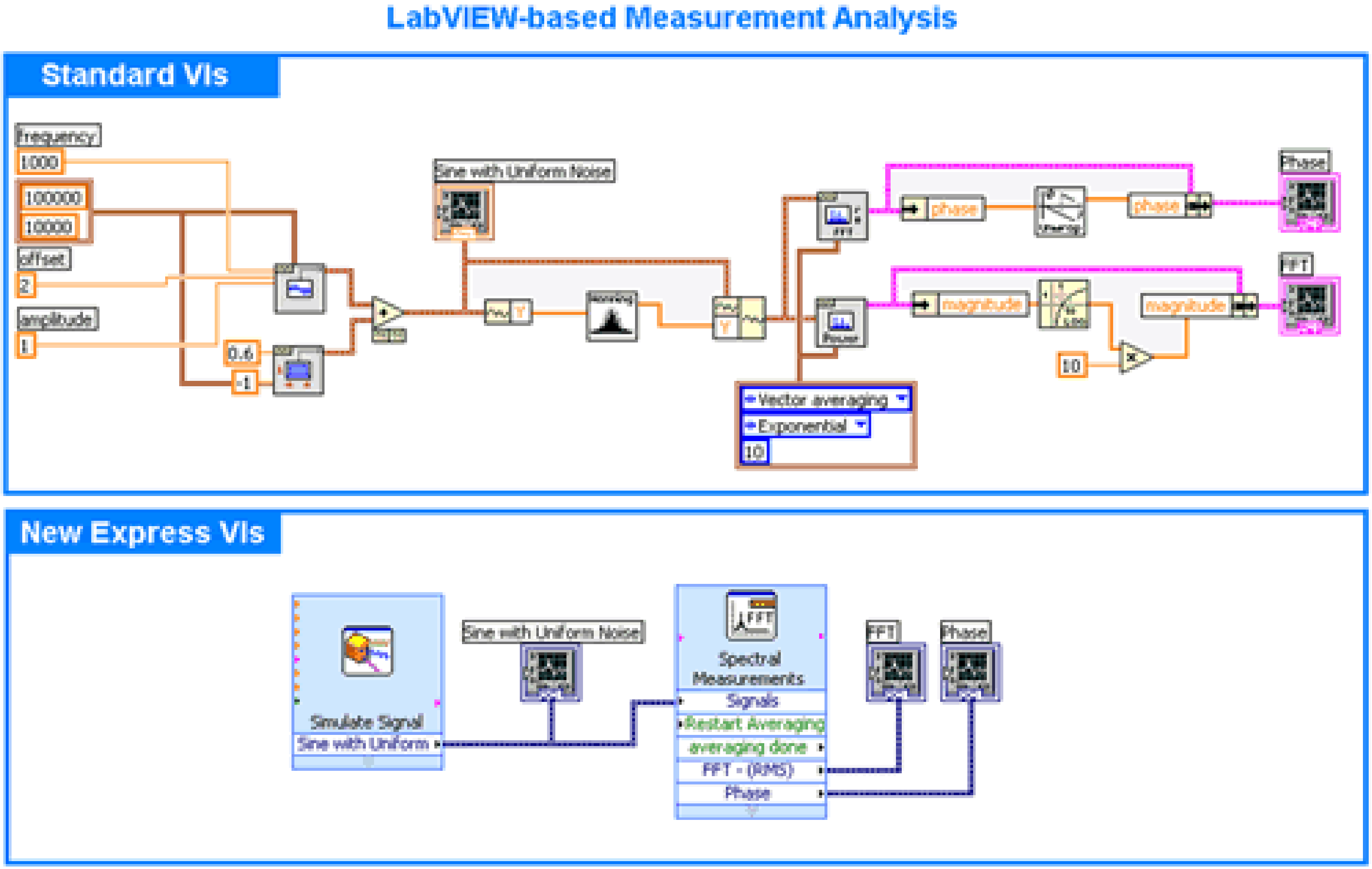
Express VI



LabVIEW Express VIs

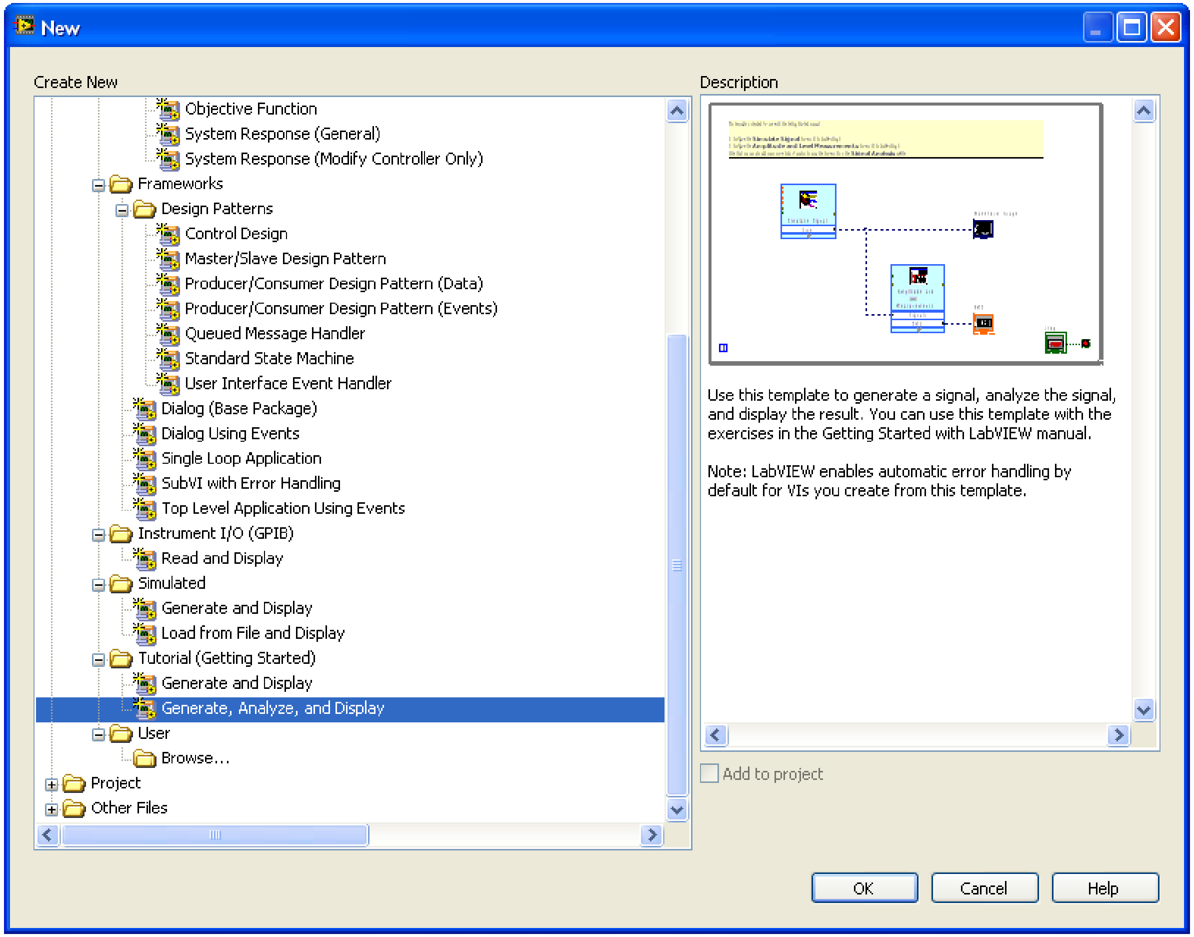
- LabVIEW includes **configuration-based** Express VIs
- With Express VIs for e.g. signal processing you can interactively explore the various analysis algorithms, while immediately seeing the results on the configuration dialog.
- **The complexity associated with adding analysis and signal processing algorithms into your measurement and automation applications is significantly reduced by using Express VIs.**
- You configure them with dialog boxes (instead of programming).
- The Express VIs encompass the most common functions
- **However, there is some overhead involved when choosing to use ExpressVIs instead of using lower level VIs**
  - The Express VIs can degrade performance (speed). Specially the “Write to Measurement File” Express VI should not be used for high speed data streaming in DAQ applications

Standard VIs vs. Express VIs - FFT



Example with Express VIs

Tutorial Name: ‘Generate, Analyze, and Display’
(from **File – New**)

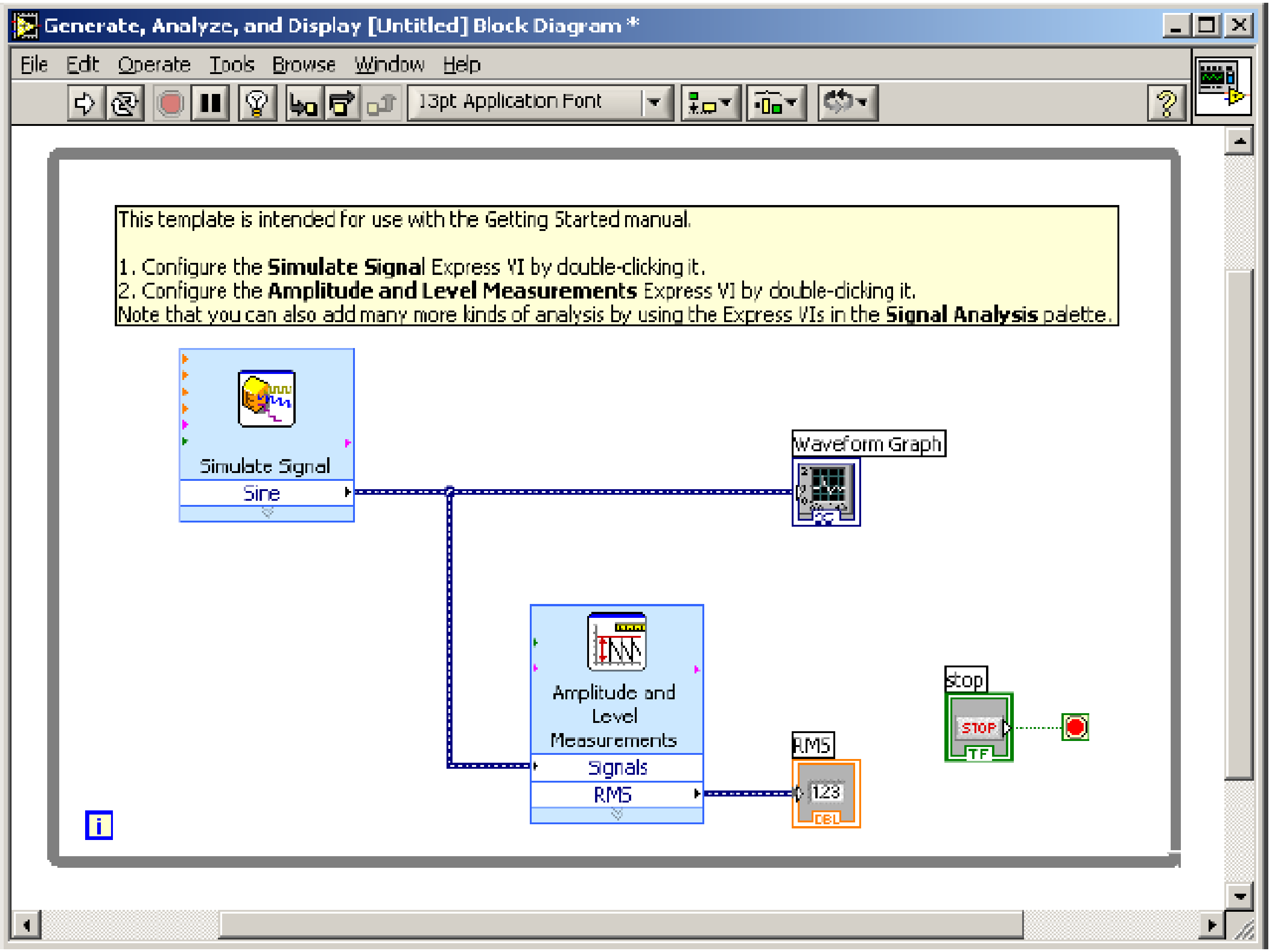

New
Create New
Objective Function
System Response (General)
System Response (Modify Controller Only)
Frameworks
Design Patterns
Control Design
Master/Slave Design Pattern
Producer/Consumer Design Pattern (Data)
Producer/Consumer Design Pattern (Events)
Queued Message Handler
Standard State Machine
User Interface Event Handler
Dialog (Base Package)
Dialog Using Events
Single Loop Application
SubVI with Error Handling
Top Level Application Using Events
Instrument I/O (GPIB)
Read and Display
Simulated
Generate and Display
Load from File and Display
Tutorial (Getting Started)
Generate and Display
Generate, Analyze, and Display
User
Browse...
Project
Other Files
Description
□ Add to project
OK
Cancel
Help

Use this template to generate a signal, analyze the signal, and display the result. You can use this template with the exercises in the Getting Started with LabVIEW manual.

Note: LabVIEW enables automatic error handling by default for VIs you create from this template.

Example with Express VIs II

Block Diagram:



This template is intended for use with the Getting Started manual.

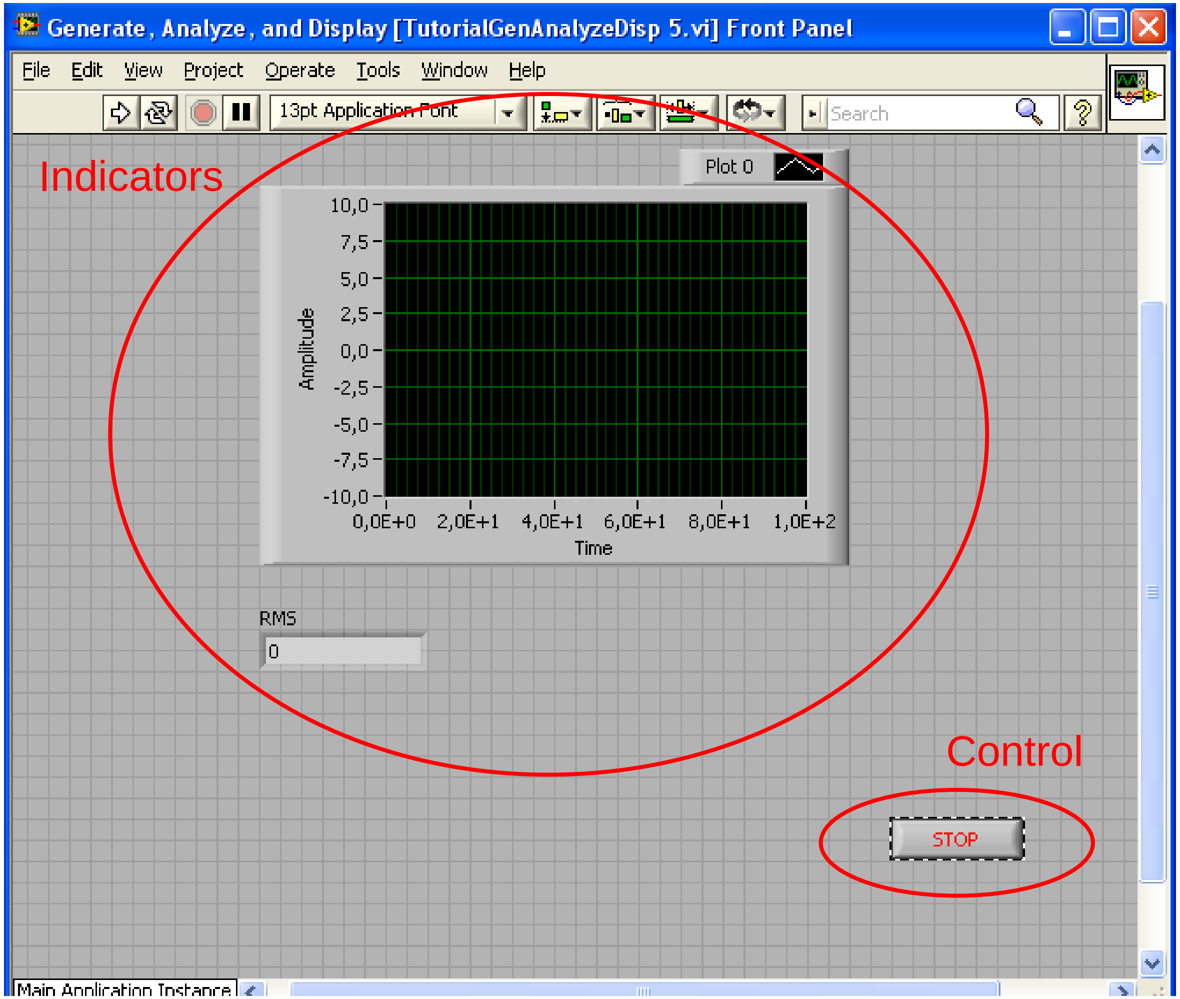
1. Configure the **Simulate Signal** Express VI by double-clicking it.
2. Configure the **Amplitude and Level Measurements** Express VI by double-clicking it.

Note that you can also add many more kinds of analysis by using the Express VIs in the **Signal Analysis** palette.

Generate, Analyze, and Display [Untitled] Block Diagram *
File Edit Operate Tools Browse Window Help
13pt Application Font
Simulate Signal
Sine
Waveform Graph
Amplitude and Level Measurements
Signals
RMS
RMS
123
DBL
stop
STOP
TF
i

Indicators and controls

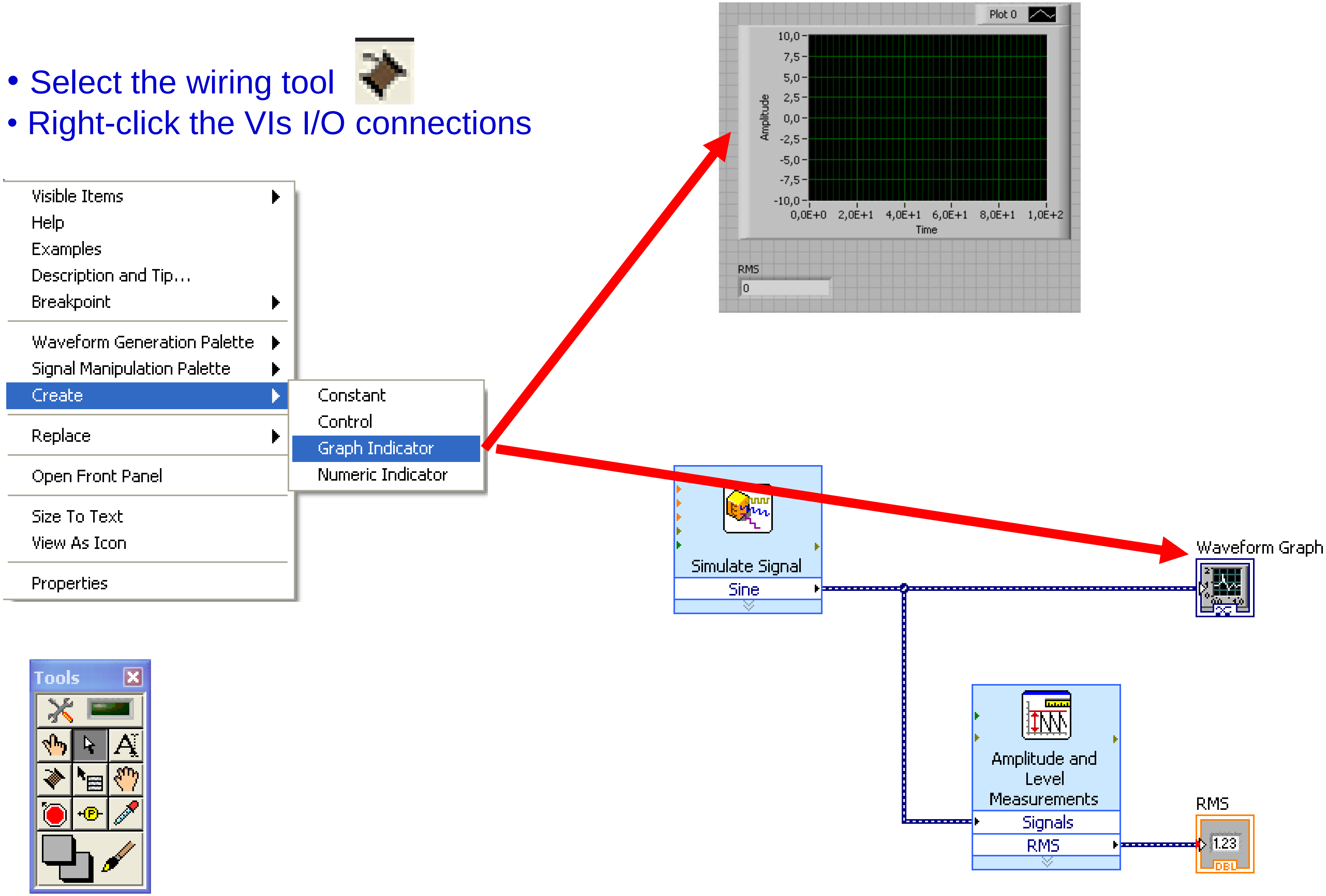
Front Panel:


Generate, Analyze, and Display [TutorialGenAnalyzeDisp 5.vi] Front Panel
File Edit View Project Operate Tools Window Help
13pt Application Font
Search
Indicators
Plot 0
10,0
7,5
5,0
2,5
0,0
-2,5
-5,0
-7,5
-10,0
Amplitude
0,0E+0
2,0E+1
4,0E+1
6,0E+1
8,0E+1
1,0E+2
Time
RMS
0
Control
STOP
Main Application Instance

LabVIEW - Indicators

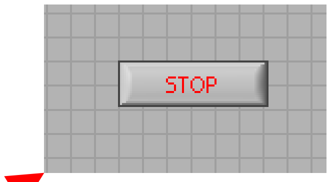
Front Panel

- Select the wiring tool
- Right-click the VIs I/O connections

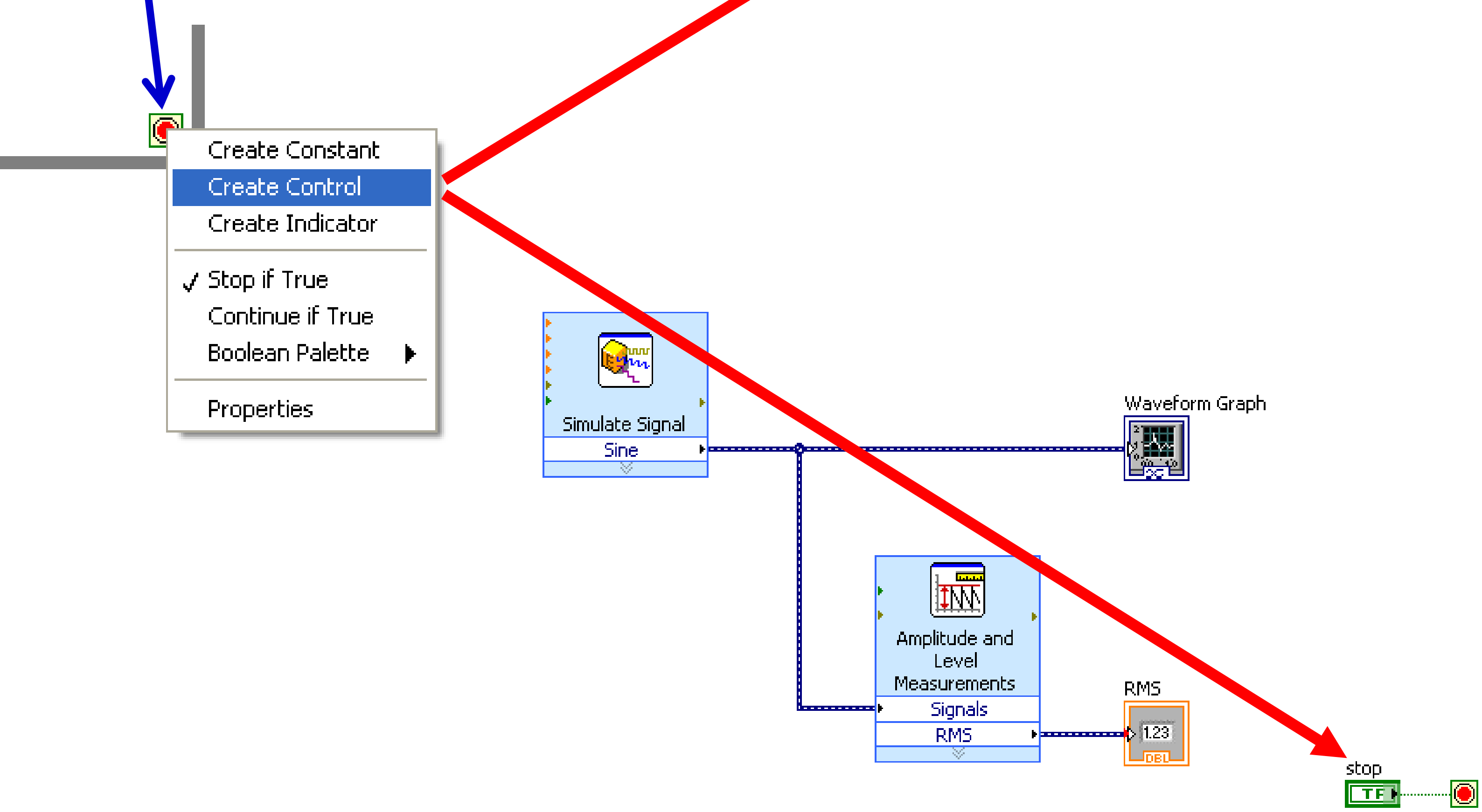


LabVIEW - Controls

Front Panel



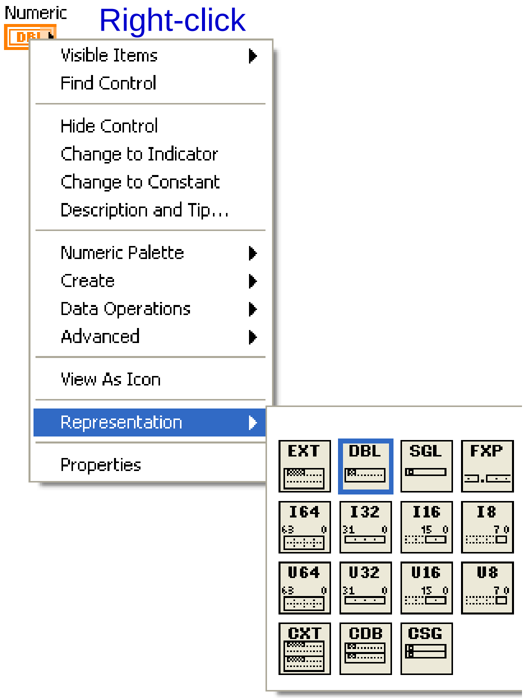
Right-click the While Loop's "stop symbol"



LabVIEW Data Types

- Color coding of terminals and block diagram wires

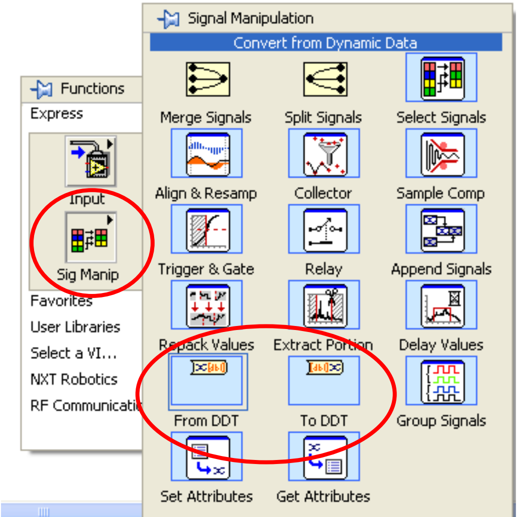
DBL	Floating-point numbers
I8	Integer
abc	String
	Dynamic
TF	Boolean



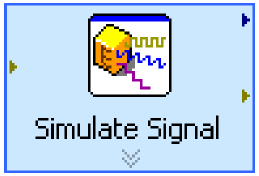
Right-click

The **dynamic data type** is a special type for use with Express VIs. Because **dynamic data undergoes an automatic conversion** to match the indicator to which it is wired, Express VIs can slow down the block diagram execution speed

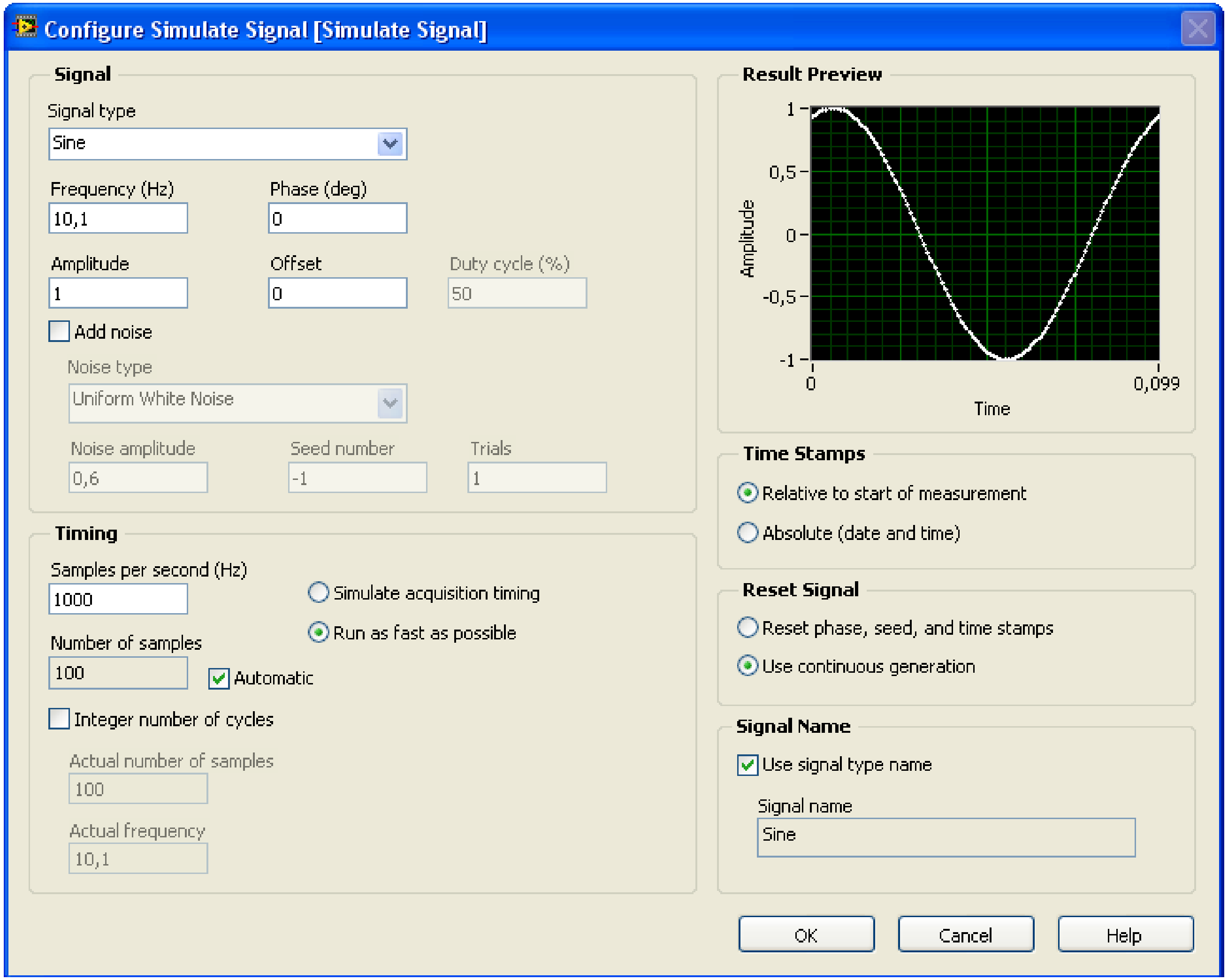
Use the Convert from/to Dynamic Data Express VI to convert dynamic data to/from numeric, Boolean, waveform, and array data for use with other VIs and functions



LabVIEW Express VI – Simulate Signal



Double-click for properties



Configure Simulate Signal [Simulate Signal]

Signal

Signal type: Sine

Frequency (Hz): 10,1

Phase (deg): 0

Amplitude: 1

Offset: 0

Duty cycle (%): 50

□ Add noise

Noise type: Uniform White Noise

Noise amplitude: 0,6

Seed number: -1

Trials: 1

Timing

Samples per second (Hz): 1000

○ Simulate acquisition timing

● Run as fast as possible

Number of samples: 100

☑ Automatic

□ Integer number of cycles

Actual number of samples: 100

Actual frequency: 10,1

Result Preview

Time Stamps

● Relative to start of measurement

○ Absolute (date and time)

Reset Signal

○ Reset phase, seed, and time stamps

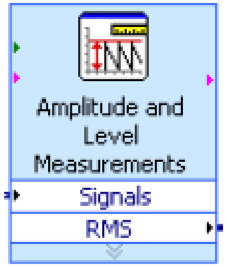
● Use continuous generation

Signal Name

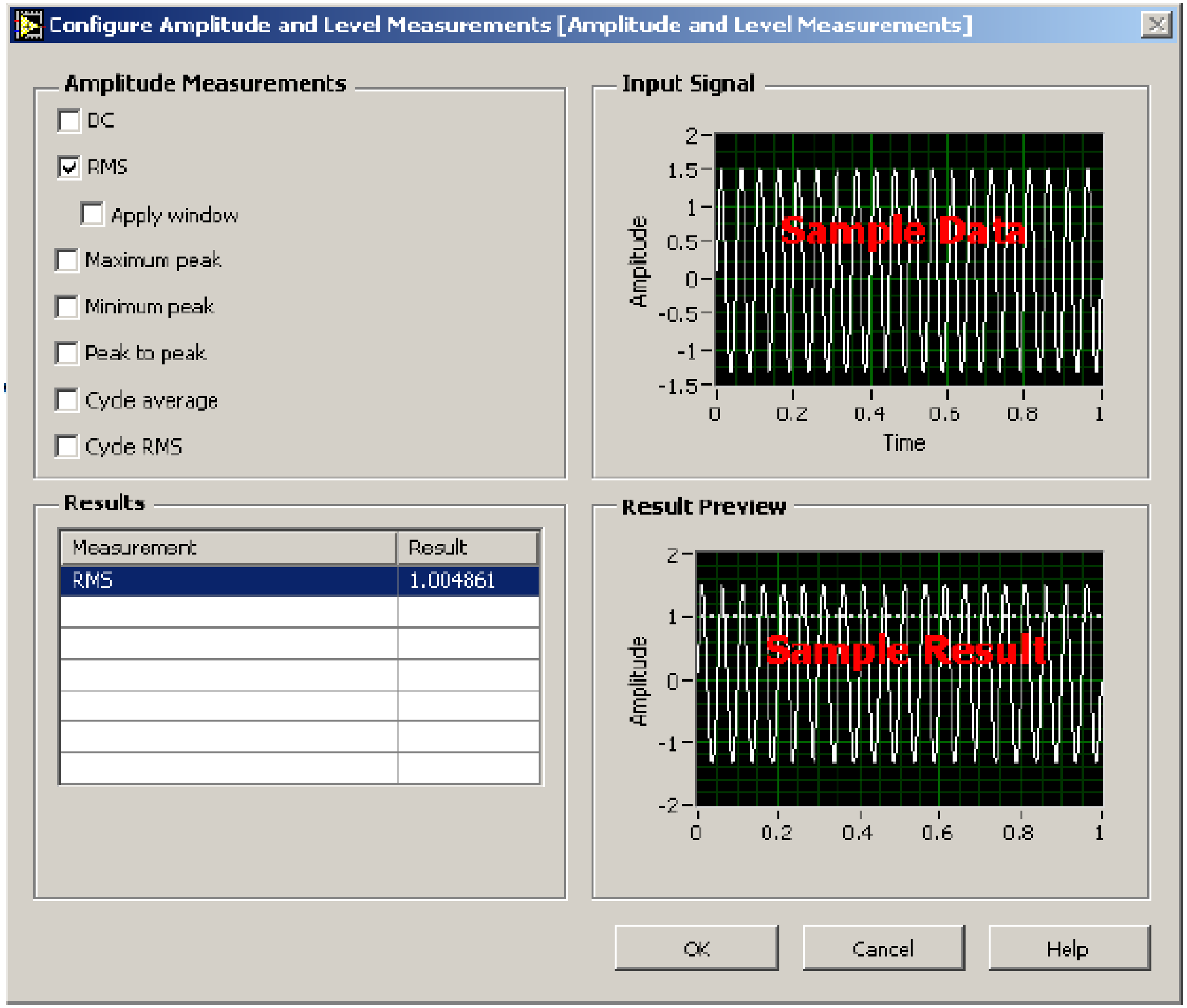
☑ Use signal type name

Signal name: Sine

LabVIEW Express VI – Amplitude and Level Measurements



Double-click for properties



Configure Amplitude and Level Measurements [Amplitude and Level Measurements]

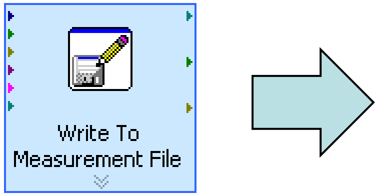
Amplitude Measurements

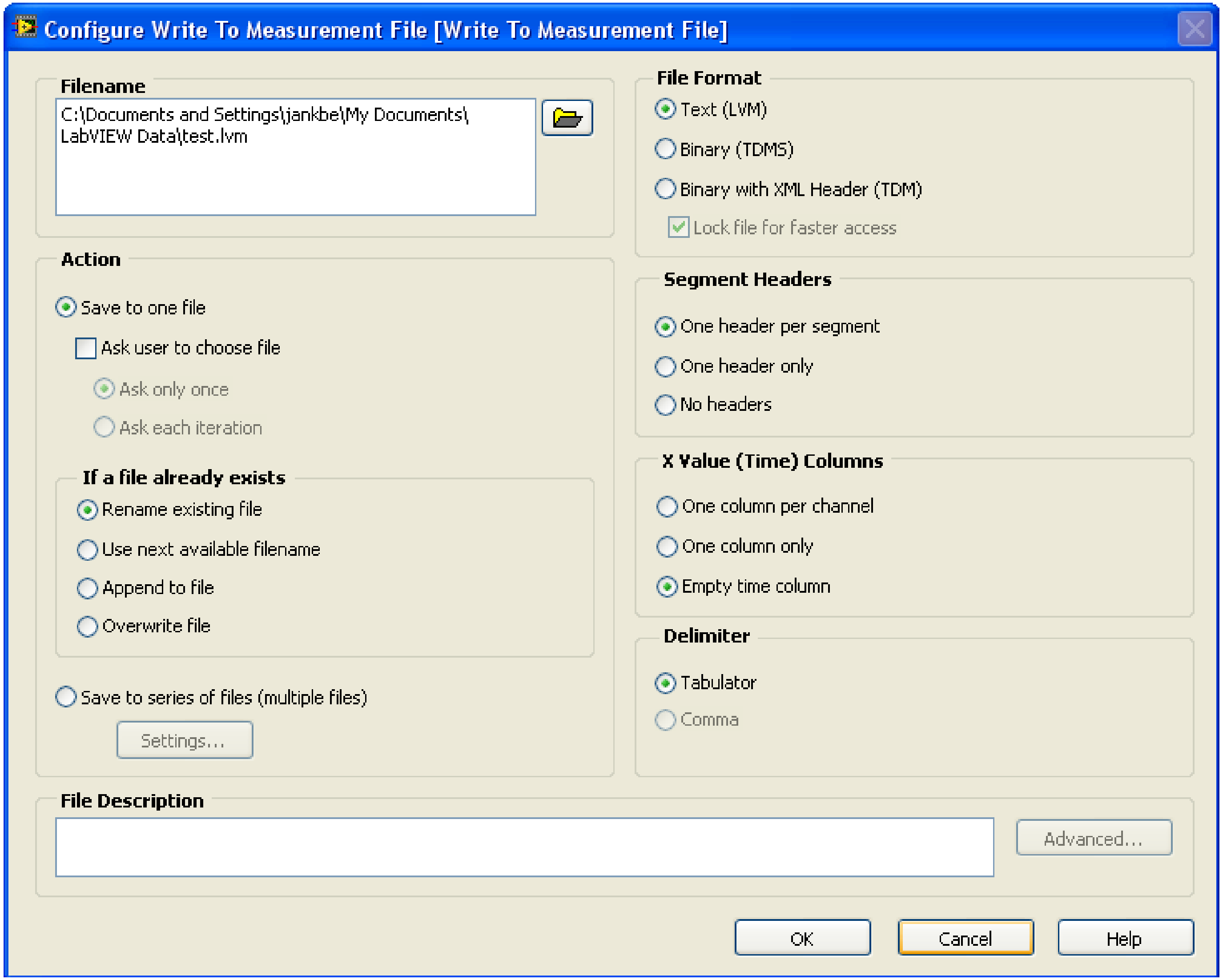
- □ DC
- ☑ RMS
  - □ Apply window
- □ Maximum peak
- □ Minimum peak
- □ Peak to peak
- □ Cycle average
- □ Cycle RMS

Results

Measurement	Result
RMS	1.004861

LabVIEW Express VI – Write to File




Configure Write To Measurement File [Write To Measurement File]

Filename

C:\Documents and Settings\jankbe\My Documents\LabVIEW Data\test.lvm

Action

- ● Save to one file
  - □ Ask user to choose file
    - ● Ask only once
    - ○ Ask each iteration
  - **If a file already exists**
    - ● Rename existing file
    - ○ Use next available filename
    - ○ Append to file
    - ○ Overwrite file
- ○ Save to series of files (multiple files)
  - Settings...

File Format

- ● Text (LVM)
- ○ Binary (TDMS)
- ○ Binary with XML Header (TDM)
- ☑ Lock file for faster access

Segment Headers

- ● One header per segment
- ○ One header only
- ○ No headers

X Value (Time) Columns

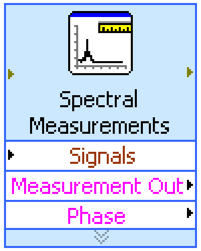
- ○ One column per channel
- ○ One column only
- ● Empty time column

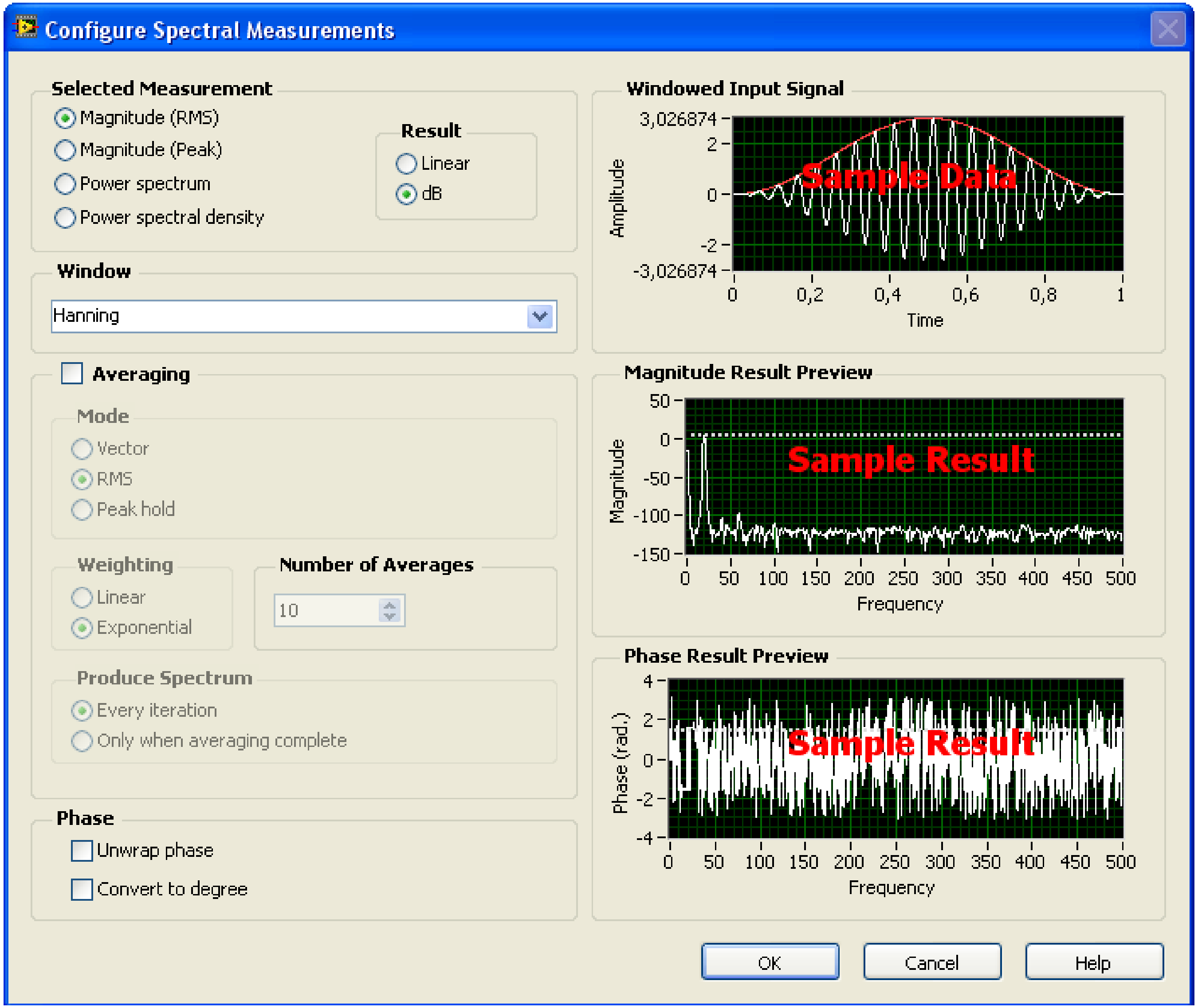
Delimiter

- ● Tabulator
- ○ Comma

File Description

LabVIEW – FFT Express VI





Configure Spectral Measurements

Selected Measurement

- ● Magnitude (RMS)
- ○ Magnitude (Peak)
- ○ Power spectrum
- ○ Power spectral density

Result

- ○ Linear
- ● dB

Window

Hanning

□ **Averaging**

Mode

- ○ Vector
- ● RMS
- ○ Peak hold

Weighting

- ○ Linear
- ● Exponential

Number of Averages

10

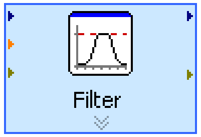
Produce Spectrum

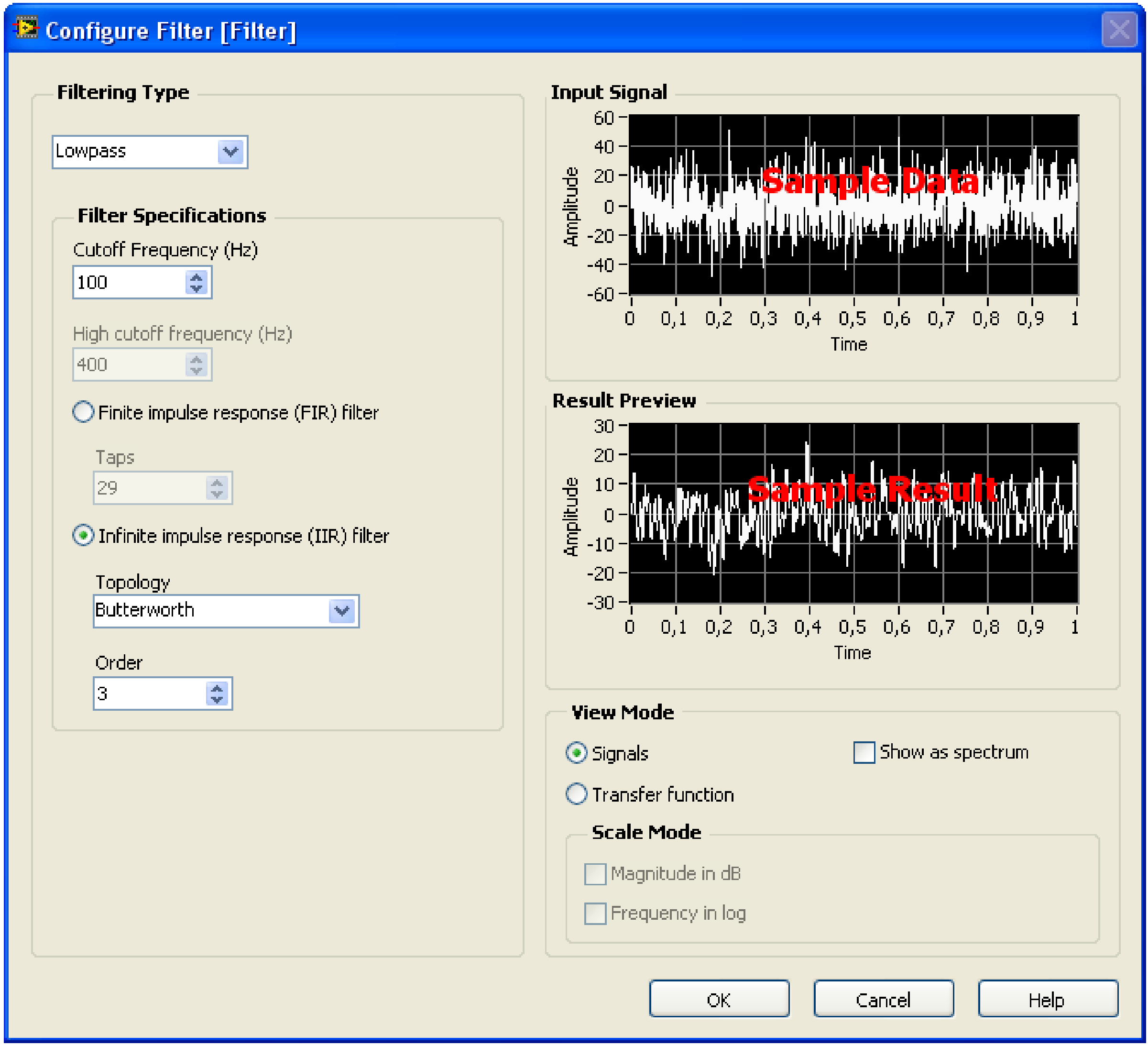
- ● Every iteration
- ○ Only when averaging complete

Phase

- □ Unwrap phase
- □ Convert to degree

LabVIEW – Digital Filter Express VI




Configure Filter [Filter]

Filtering Type

Lowpass

Filter Specifications

Cutoff Frequency (Hz): 100

High cutoff frequency (Hz): 400

○ Finite impulse response (FIR) filter

Taps: 29

● Infinite impulse response (IIR) filter

Topology: Butterworth

Order: 3

Input Signal

Result Preview

View Mode

● Signals

□ Show as spectrum

○ Transfer function

Scale Mode

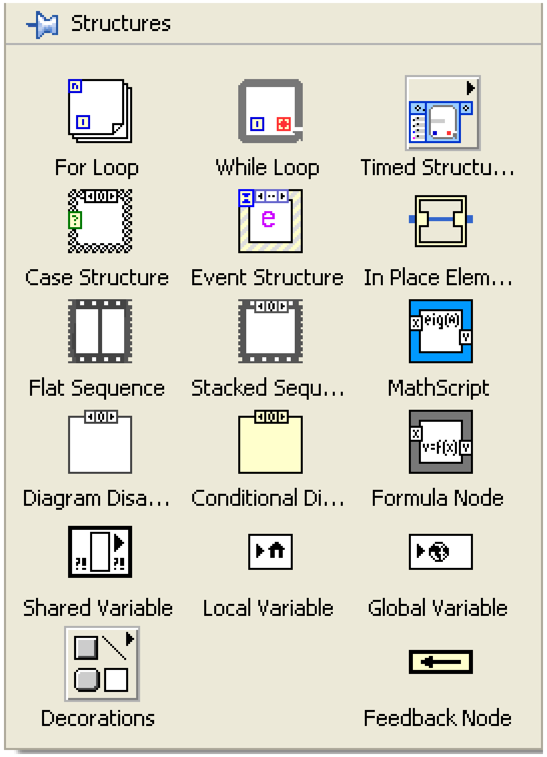
□ Magnitude in dB

□ Frequency in log

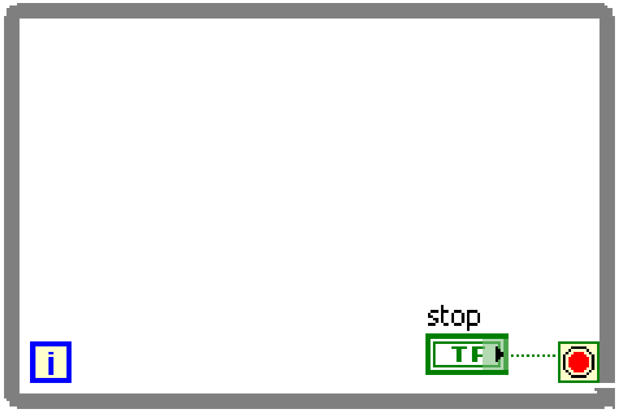
LabVIEW demo
- Signal generation, filtering, FFT, and Write to file

Loops

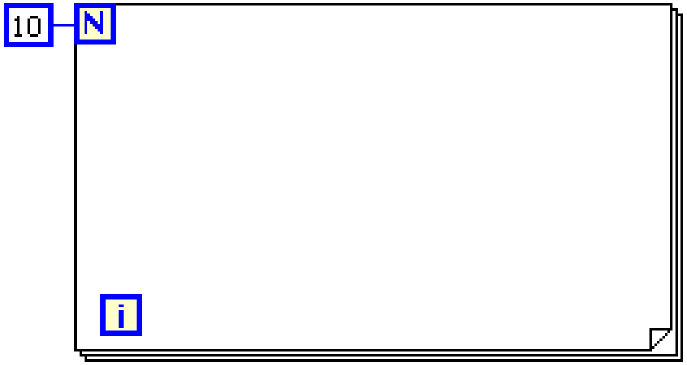
Functions – Programming – Structures:



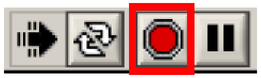
While Loop:



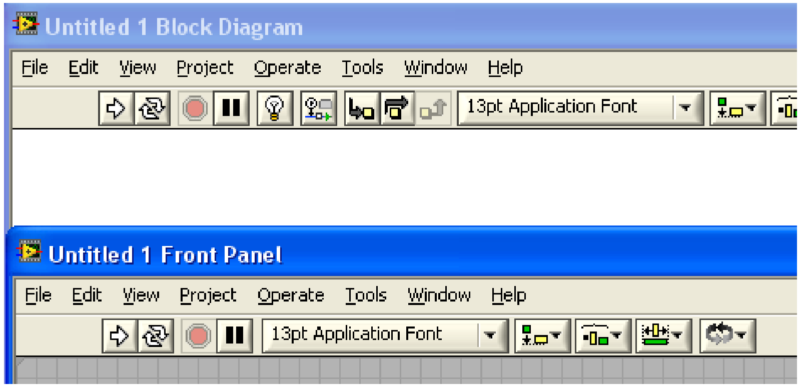
For Loop:



Note: Unless in emergency situations, never use the 'abort button'. Always program in a 'stop button'.



Program Start, Abort execution and Error indication


Untitled 1 Block Diagram
File Edit View Project Operate Tools Window Help
13pt Application Font
Untitled 1 Front Panel
File Edit View Project Operate Tools Window Help
13pt Application Font

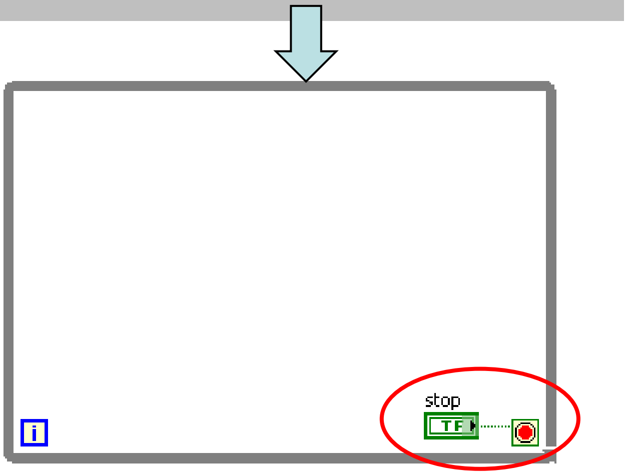
- Aborting a VI that uses external resources, such as external hardware, might leave the resources in an unknown state by not resetting or releasing them properly. Design the VIs you create with a **stop button** and use it to avoid this problem.

Start (RUN) program button

Program Running indicator

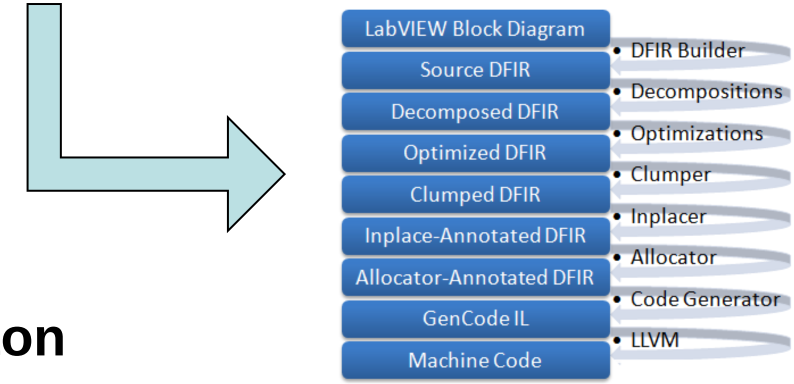
Broken arrow – error in program

Abort Execution button



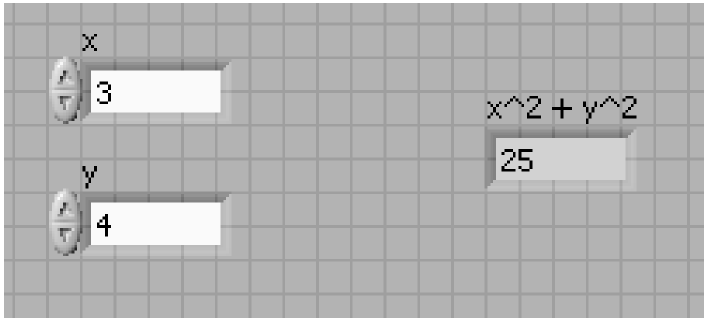
How VIs are compiled

- When you push the Run button, LabVIEW (the G- compiler) translates the block diagram into clumps of **machine code** for your platform
- **LabVIEW will automatically compile your VI during load, run or save (if necessary).** In general, any change that is non cosmetic will set a flag indicating that the VI needs to be recompiled. When this flag is set the VI will automatically compile when you run or save.
- Beginning with LabVIEW 2009 and continuing in LabVIEW 2010 many optimizations were added to the LabVIEW compiler to speed up run-time performance of both VIs and executables



⇨ **Start (RUN) program button**

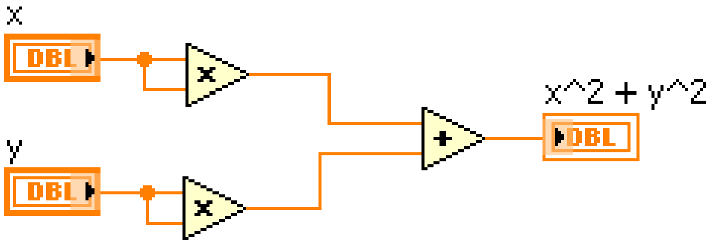
Standard LabVIEW VIs - Example



x 3

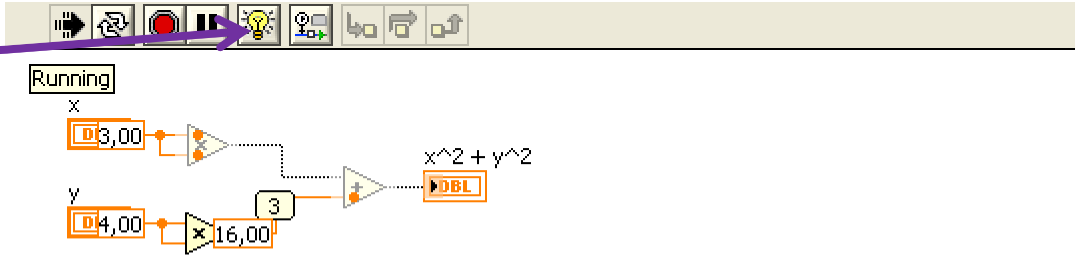
y 4

x^2 + y^2 25



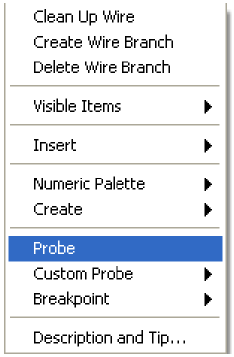
LabVIEW debugging techniques

- Execution highlighting

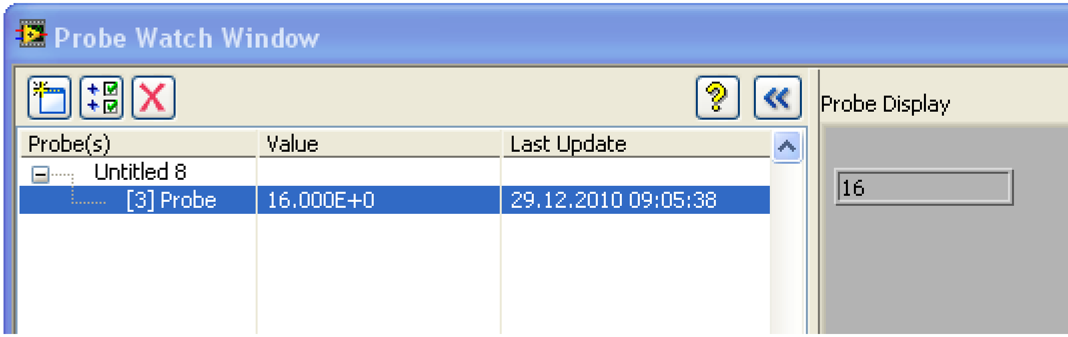


- Single-stepping

- Probe

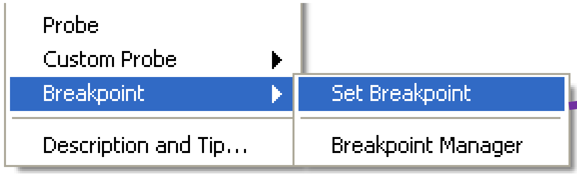


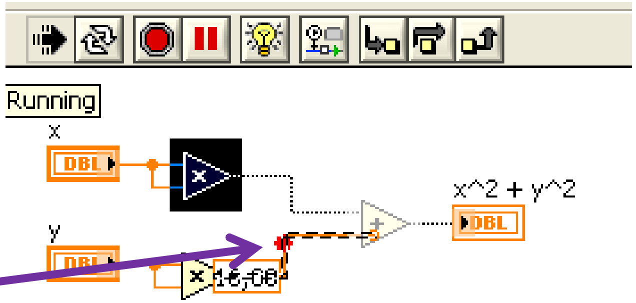
Right-click wire



Probe(s)	Value	Last Update
Untitled 8		
[3] Probe	16.000E+0	29.12.2010 09:05:38

- Breakpoints



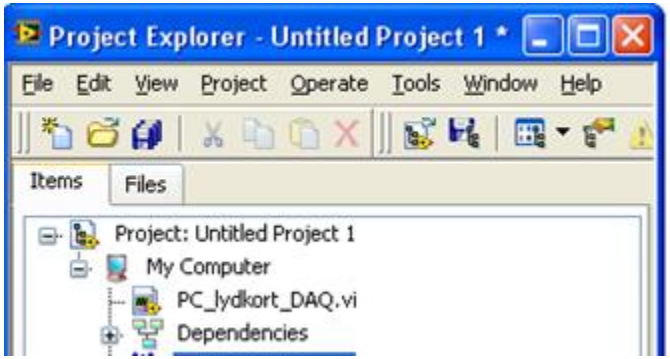


Projects in LabVIEW

- Projects in LabVIEW consist of VIs, files necessary for those VIs to run properly, and supplemental files such as documentation or related links. Use the **Project Explorer** window to manage projects in LabVIEW

- **File – New Project**

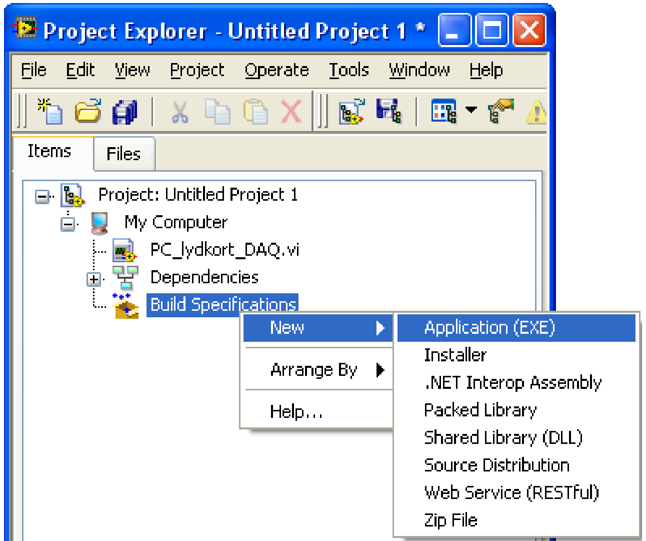
- Adding files to the project:
  - Right-click “My Computer”, and select **Add – File ..**


Project Explorer - Untitled Project 1 *
File Edit View Project Operate Tools Window Help
Items
Files
Project: Untitled Project 1
My Computer
PC_lydkort_DAQ.vi
Dependencies

Extra

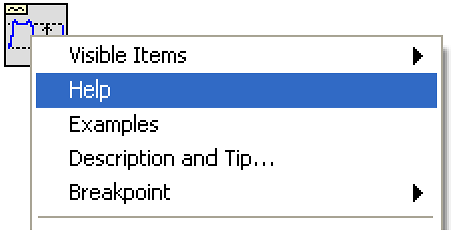
Creating Executables (EXE-file)

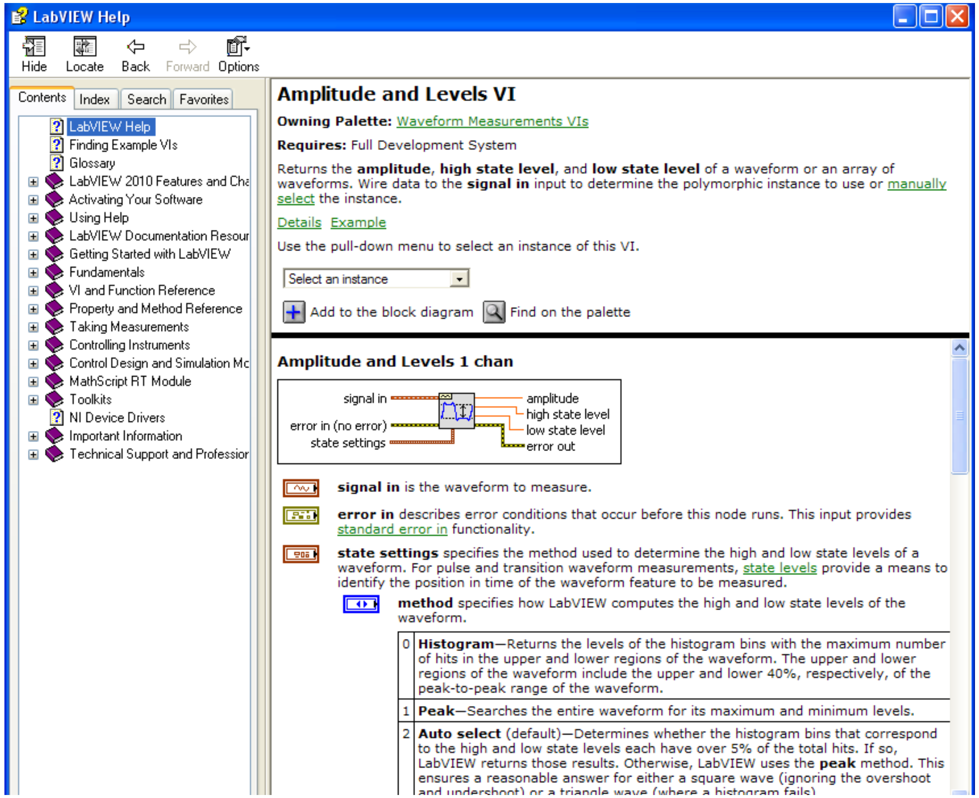
- Create a stand-alone windows application (e.g. a *.exe file)
  - You must have a project open and saved to configure a build specification.
  - The **LabVIEW Run-Time Engine** must be installed on any computer on which users run the application or shared library.
  - You can include the LabVIEW Run-Time Engine in an **installe**r
    - *Create the EXE-file first*
    - *Then, add the EXE file to the installer*



LabVIEW – Help I

- Right-click an icon (VI), and select Help.





Amplitude and Levels VI

Owning Palette: Waveform Measurements VIs

Requires: Full Development System

Returns the **amplitude**, **high state level**, and **low state level** of a waveform or an array of waveforms. Wire data to the **signal in** input to determine the polymorphic instance to use or manually select the instance.

Details Example

Use the pull-down menu to select an instance of this VI.

Select an instance ▾

Add to the block diagram   Find on the palette

Amplitude and Levels 1 chan

signal in is the waveform to measure.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

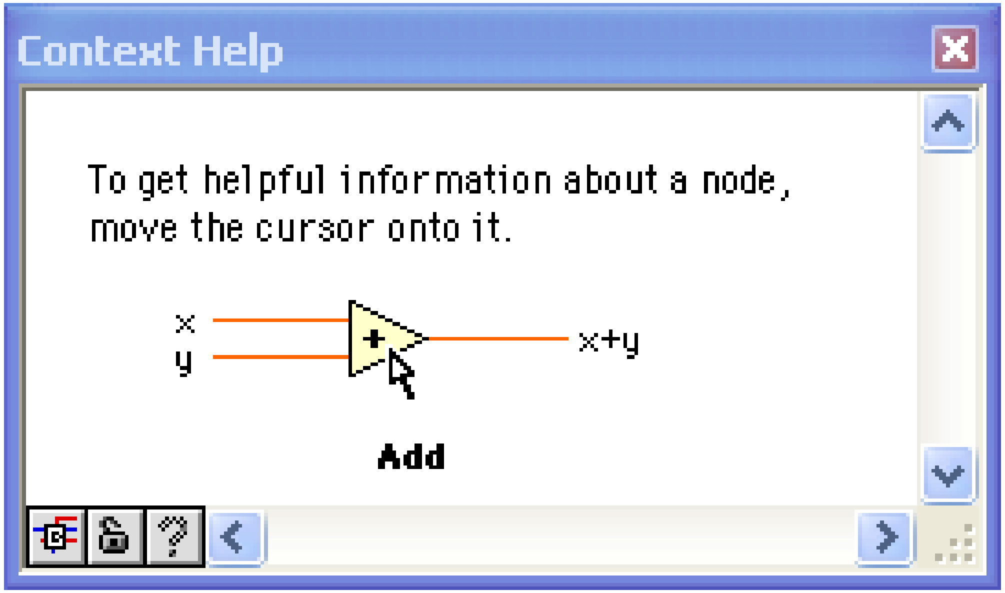
state settings specifies the method used to determine the high and low state levels of a waveform. For pulse and transition waveform measurements, state levels provide a means to identify the position in time of the waveform feature to be measured.

method specifies how LabVIEW computes the high and low state levels of the waveform.

0	**Histogram**—Returns the levels of the histogram bins with the maximum number of hits in the upper and lower regions of the waveform. The upper and lower regions of the waveform include the upper and lower 40%, respectively, of the peak-to-peak range of the waveform.
1	**Peak**—Searches the entire waveform for its maximum and minimum levels.
2	**Auto select** (default)—Determines whether the histogram bins that correspond to the high and low state levels each have over 5% of the total hits. If so, LabVIEW returns those results. Otherwise, LabVIEW uses the **peak** method. This ensures a reasonable answer for either a square wave (ignoring the overshoot and undershoot) or a triangle wave (where a histogram fails).

LabVIEW – Help II

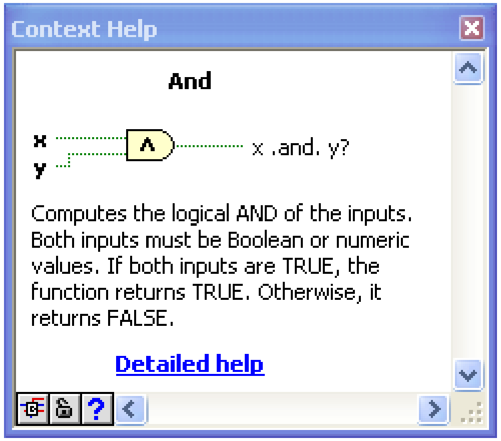
- Select **Help»Show Context Help** from the front panel or the block diagram
- Move the cursor over to the graphical symbol to see the help information
- Very useful when looking at functions in the '**Functions**' palette:



Context Help

To get helpful information about a node, move the cursor onto it.

Add



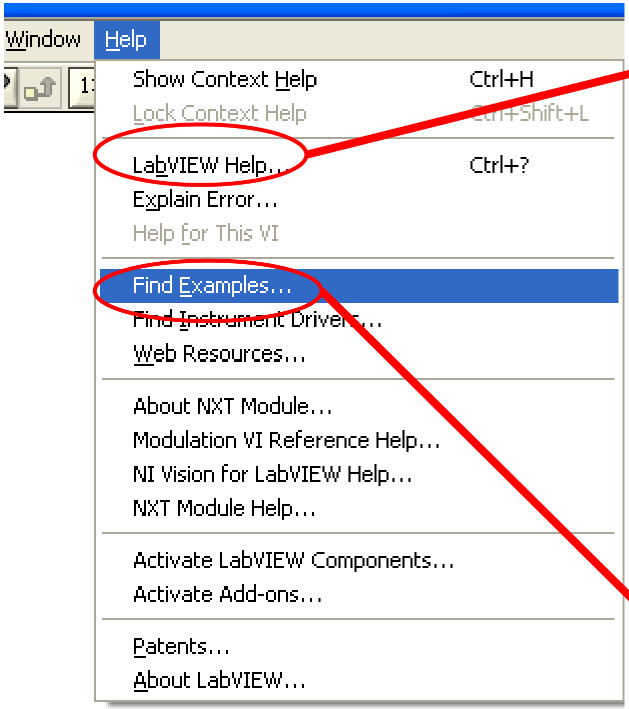
Context Help

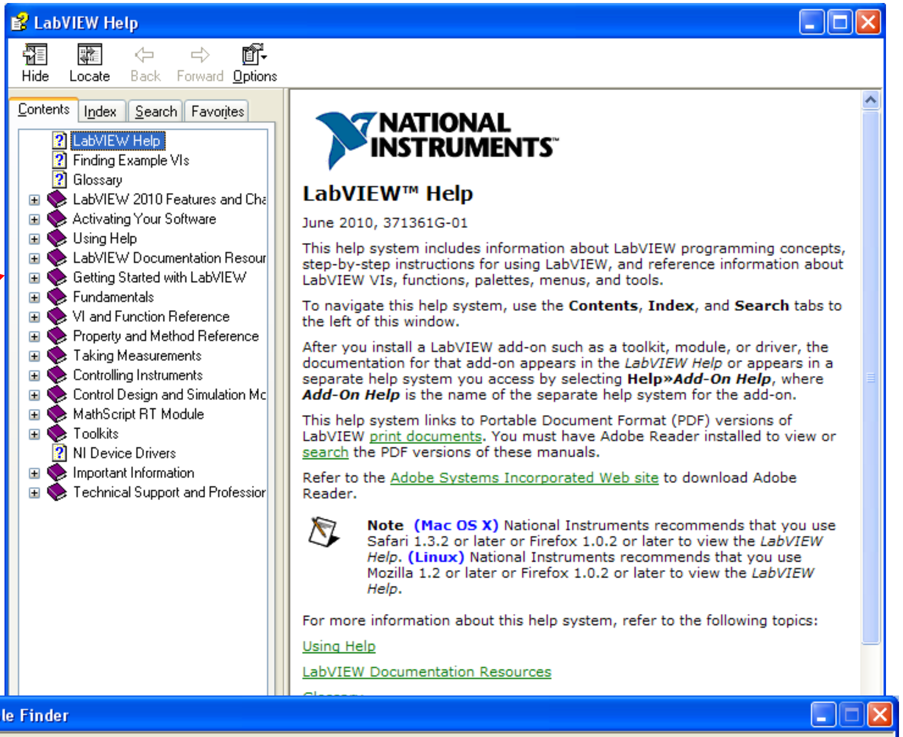
And

Computes the logical AND of the inputs. Both inputs must be Boolean or numeric values. If both inputs are TRUE, the function returns TRUE. Otherwise, it returns FALSE.

Detailed help

LabVIEW Help III





LabVIEW™ Help

June 2010, 371361G-01

This help system includes information about LabVIEW programming concepts, step-by-step instructions for using LabVIEW, and reference information about LabVIEW VIs, functions, palettes, menus, and tools.

To navigate this help system, use the **Contents**, **Index**, and **Search** tabs to the left of this window.

After you install a LabVIEW add-on such as a toolkit, module, or driver, the documentation for that add-on appears in the *LabVIEW Help* or appears in a separate help system you access by selecting **Help»*Add-On Help***, where ***Add-On Help*** is the name of the separate help system for the add-on.

This help system links to Portable Document Format (PDF) versions of LabVIEW print documents. You must have Adobe Reader installed to view or search the PDF versions of these manuals.

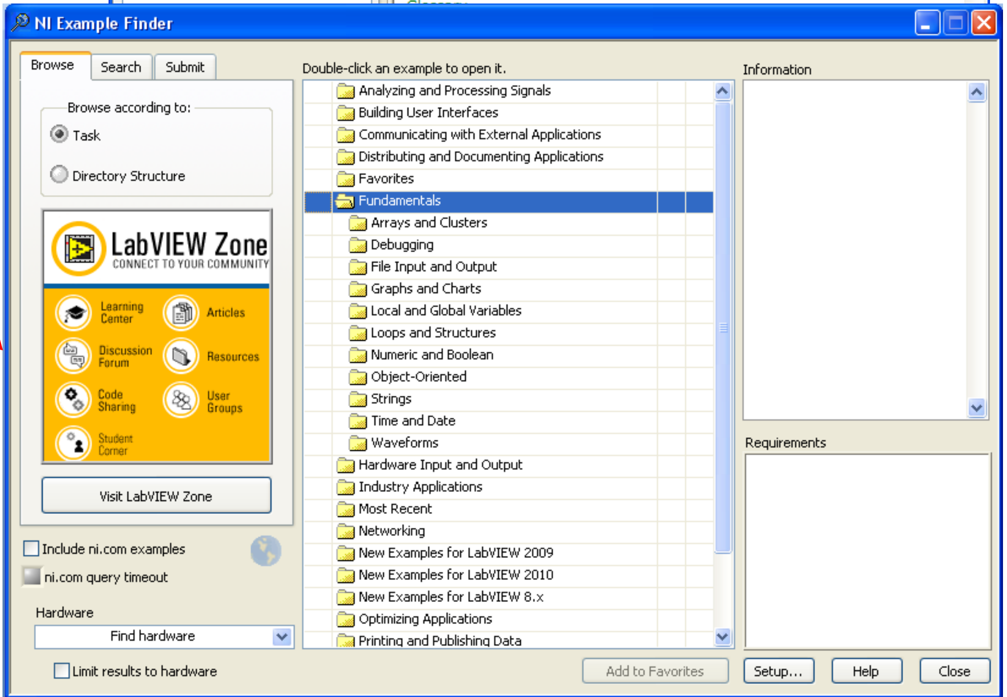
Refer to the Adobe Systems Incorporated Web site to download Adobe Reader.

Note **(Mac OS X)** National Instruments recommends that you use Safari 1.3.2 or later or Firefox 1.0.2 or later to view the *LabVIEW Help*. **(Linux)** National Instruments recommends that you use Mozilla 1.2 or later or Firefox 1.0.2 or later to view the *LabVIEW Help*.

For more information about this help system, refer to the following topics:

Using Help

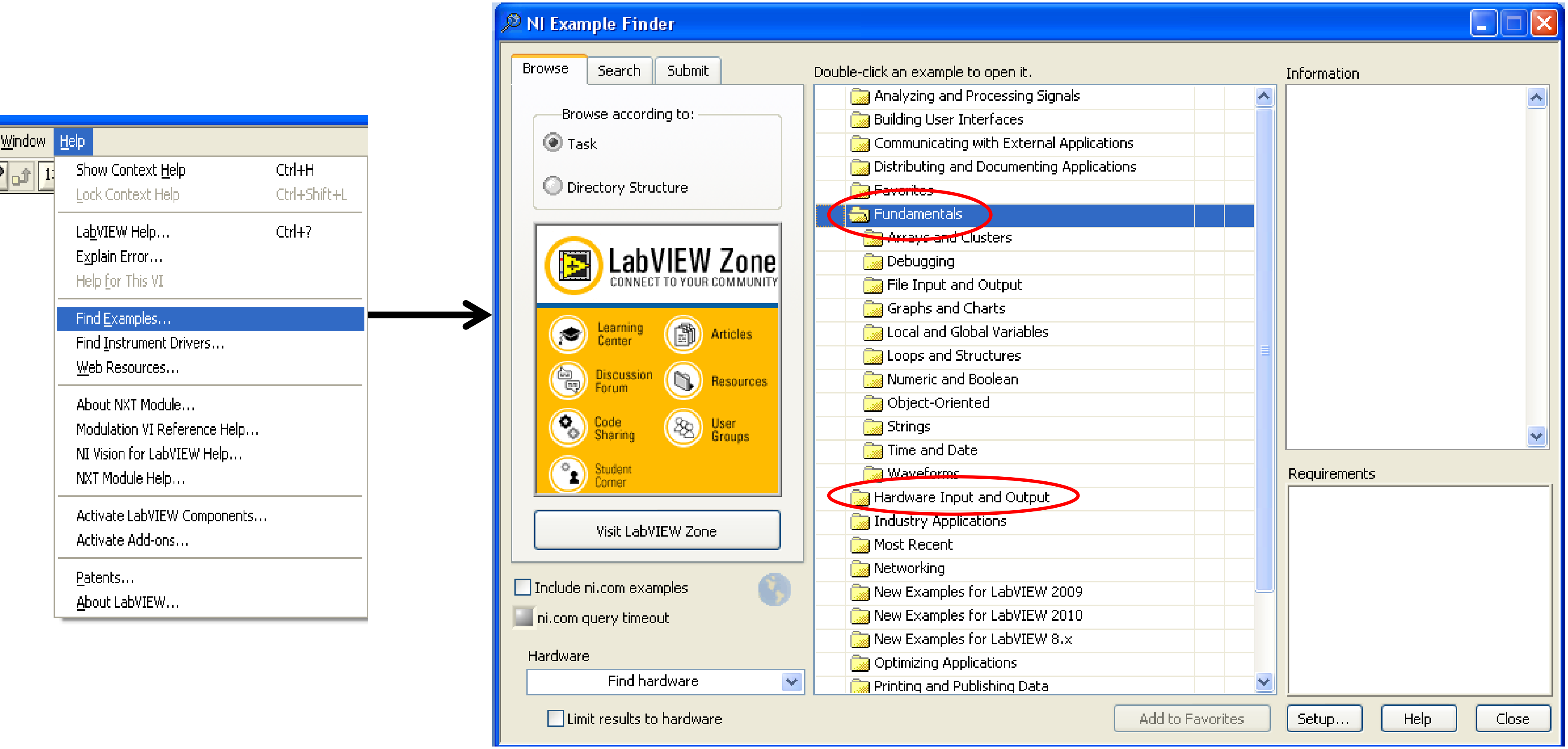
LabVIEW Documentation Resources



Can use existing Examples as a starting point!

NI Example Finder

- Search or browse through example programs
- Modify an existing example program to fit your application
- Copy and paste from an example into your own program



Browse according to:

● Task

○ Directory Structure

□ Include ni.com examples

ni.com query timeout

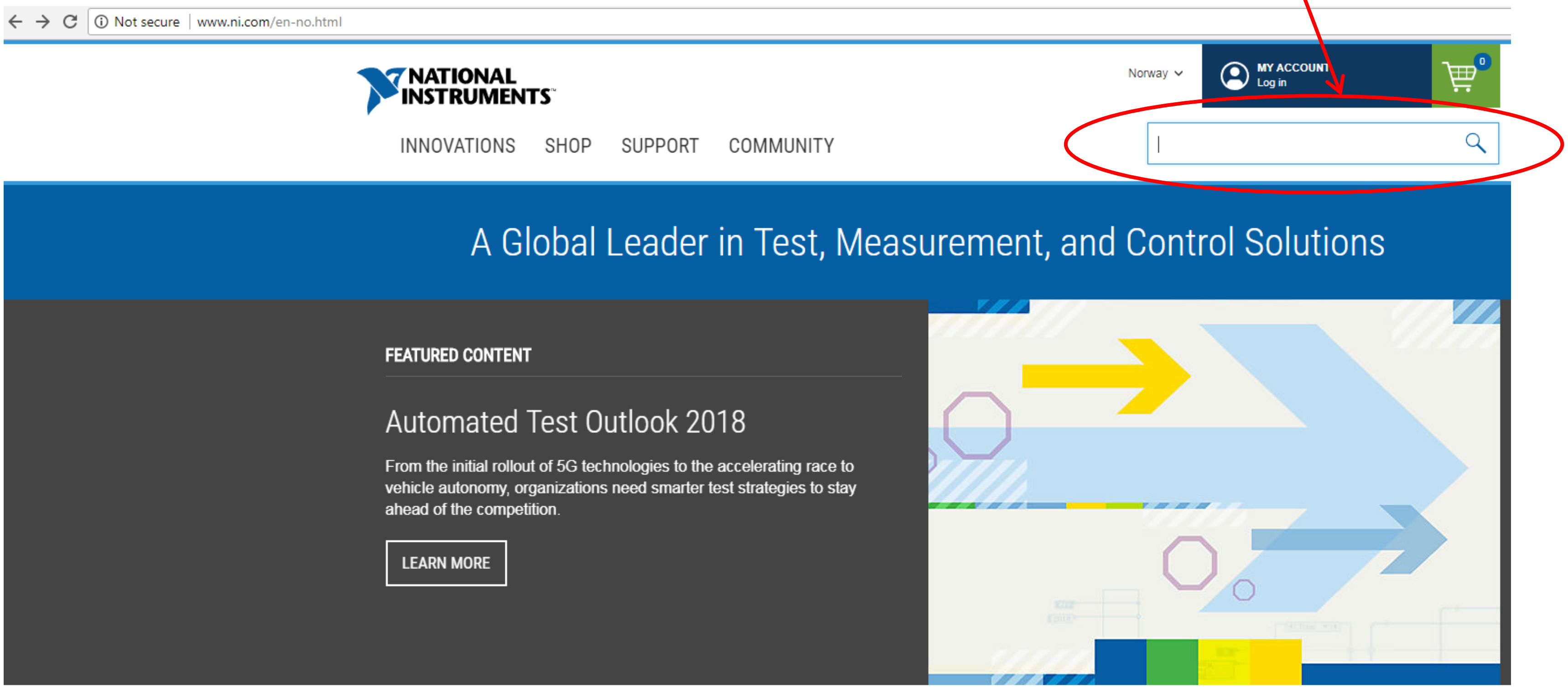
□ Limit results to hardware

Double-click an example to open it.

- Analyzing and Processing Signals
- Building User Interfaces
- Communicating with External Applications
- Distributing and Documenting Applications
- Favorites
- Fundamentals
  - Arrays and Clusters
  - Debugging
  - File Input and Output
  - Graphs and Charts
  - Local and Global Variables
  - Loops and Structures
  - Numeric and Boolean
  - Object-Oriented
  - Strings
  - Time and Date
  - Waveforms
- Hardware Input and Output
- Industry Applications
- Most Recent
- Networking
- New Examples for LabVIEW 2009
- New Examples for LabVIEW 2010
- New Examples for LabVIEW 8.x
- Optimizing Applications
- Printing and Publishing Data

LabVIEW Help III - www.ni.com

- Make a search for solution to problems or to find source code!



A Global Leader in Test, Measurement, and Control Solutions

FEATURED CONTENT

Automated Test Outlook 2018

From the initial rollout of 5G technologies to the accelerating race to vehicle autonomy, organizations need smarter test strategies to stay ahead of the competition.