



UiO : **University of Oslo**

FYS3240

PC-based instrumentation and microcontrollers

# LabVIEW programming I

LabVIEW basics

Spring 2013 – Lecture #2

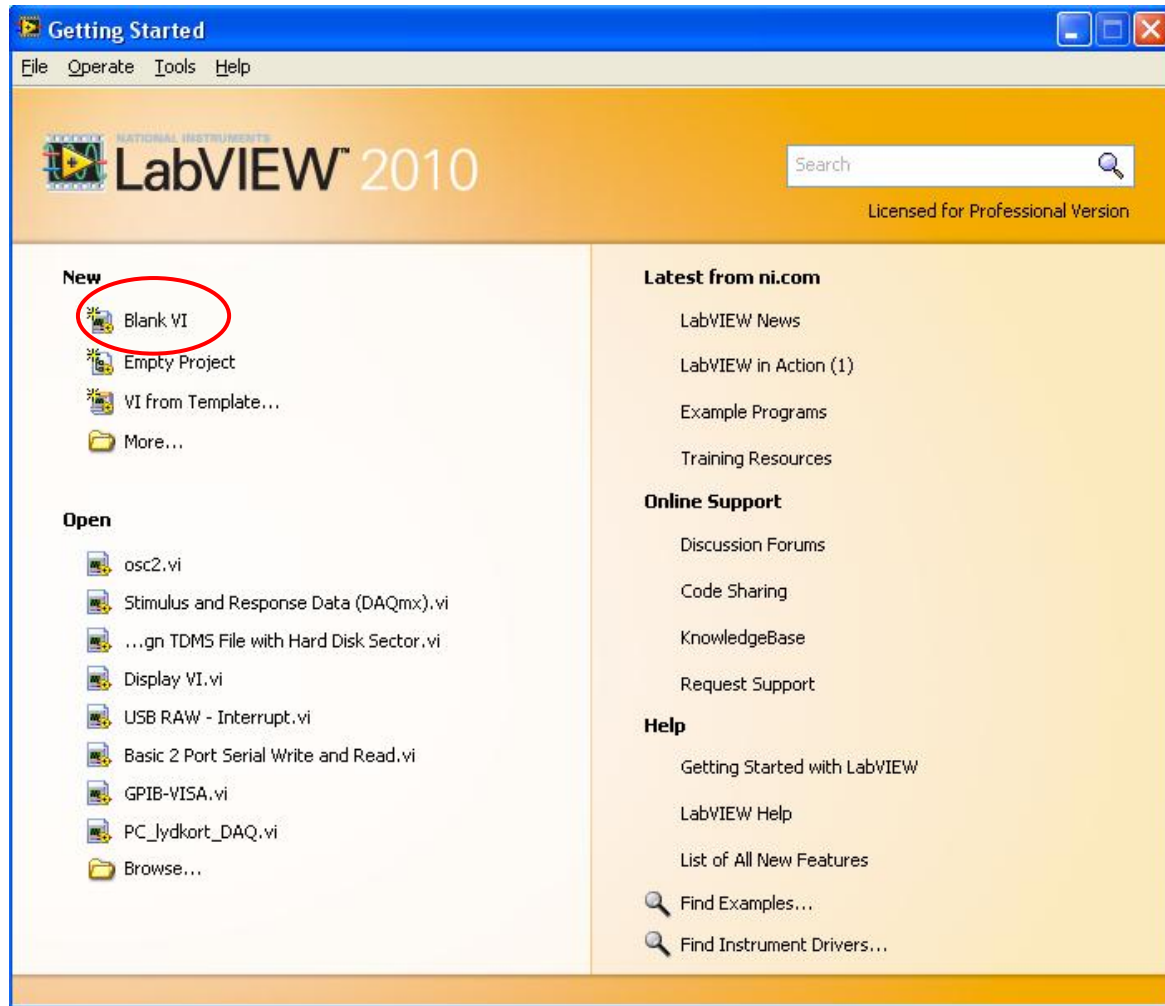
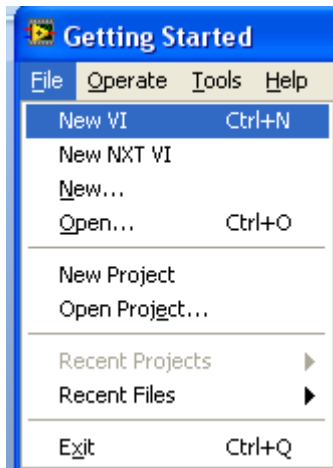


# Virtual Instruments

- LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters.
  - A LabVIEW program has the file ending \*.vi, e.g. **test.vi**
- LabVIEW contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data, as well as tools to help you troubleshoot code you write.
- In LabVIEW a VI is:
  - A LabVIEW program when it is the top-file
  - A SubVI when a VI is used in another VI
    - A SubVI is similar to a function in other programming languages

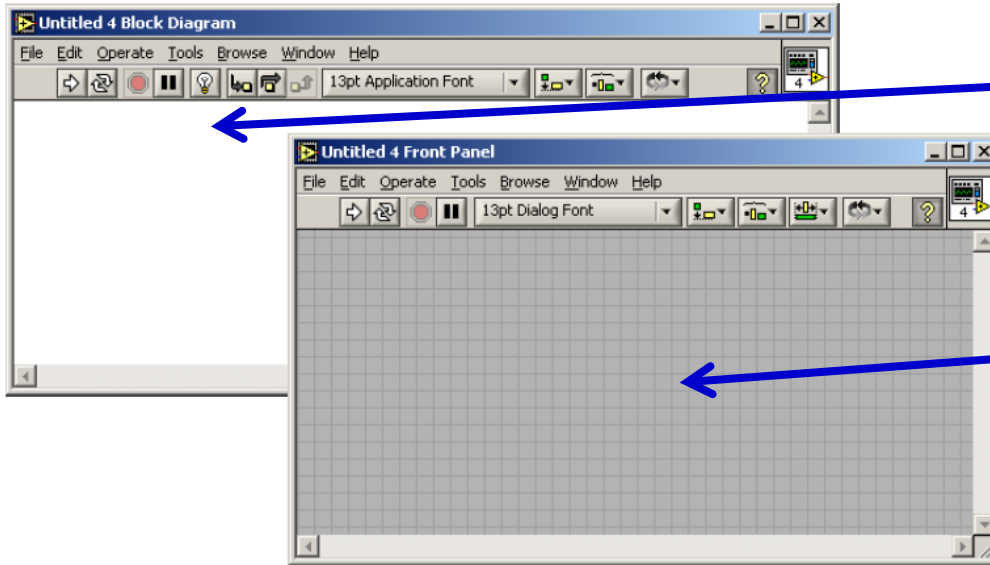
# LabVIEW – Start up

- Click **Blank VI**
- or select **File – New VI**



# LabVIEW – Blank VI (Untitled)

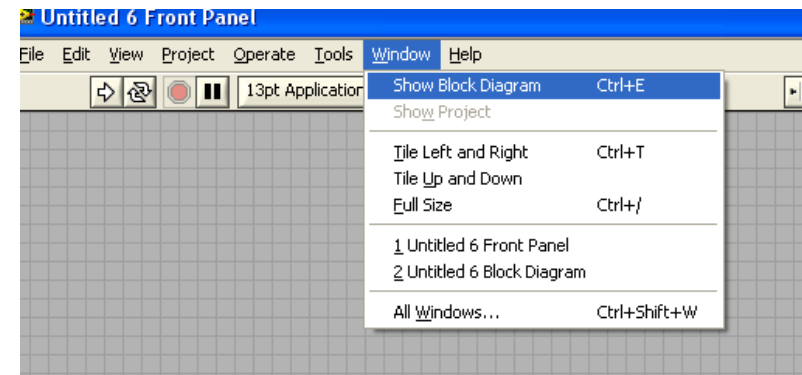
Two windows appear – ‘Block Diagram’ & ‘Front Panel’



The block diagram contains the graphical source code

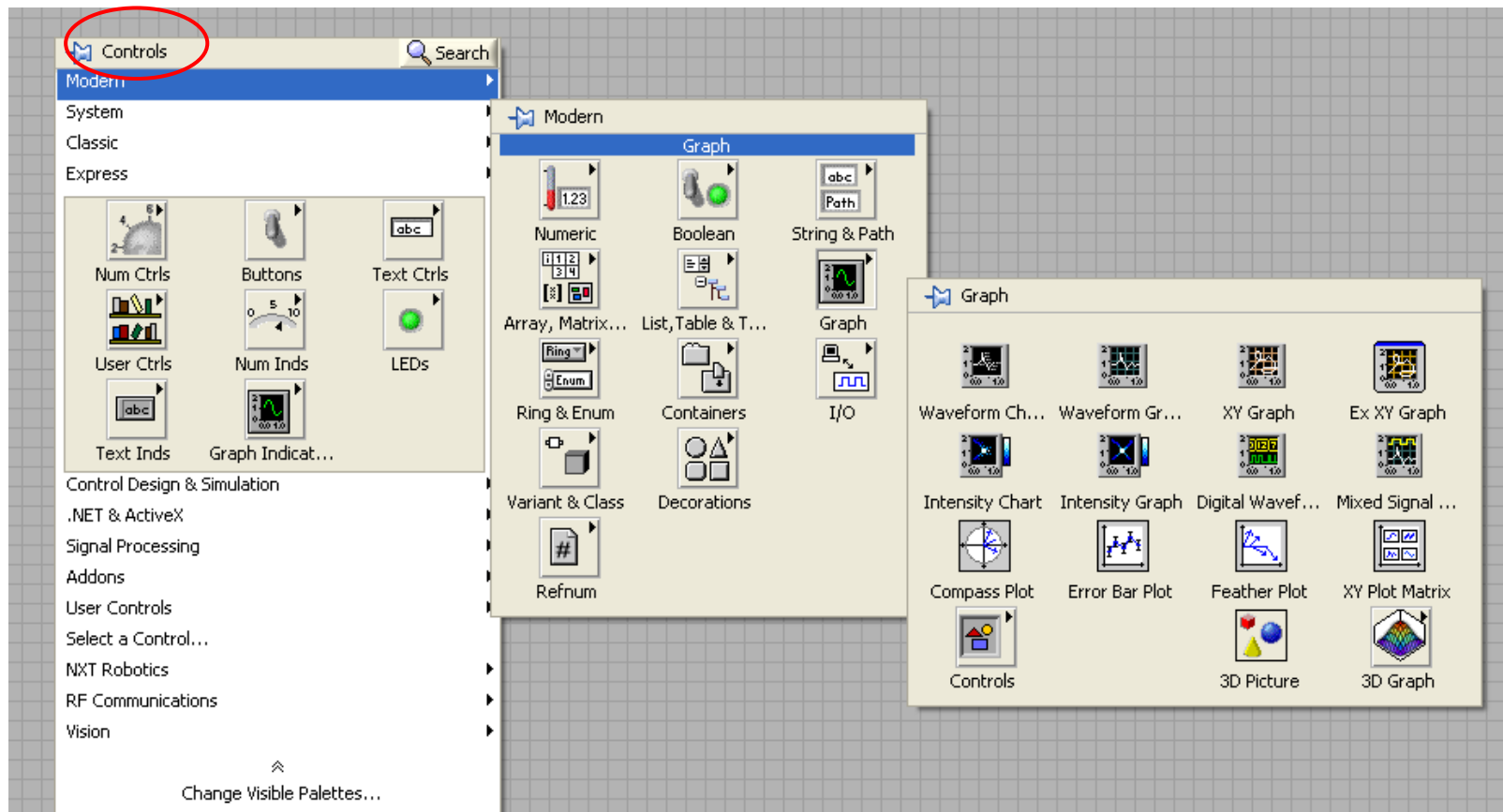
The front panel is the **user interface (GUI)** of a VI. You build the front panel by using controls and indicators

Select window to show:



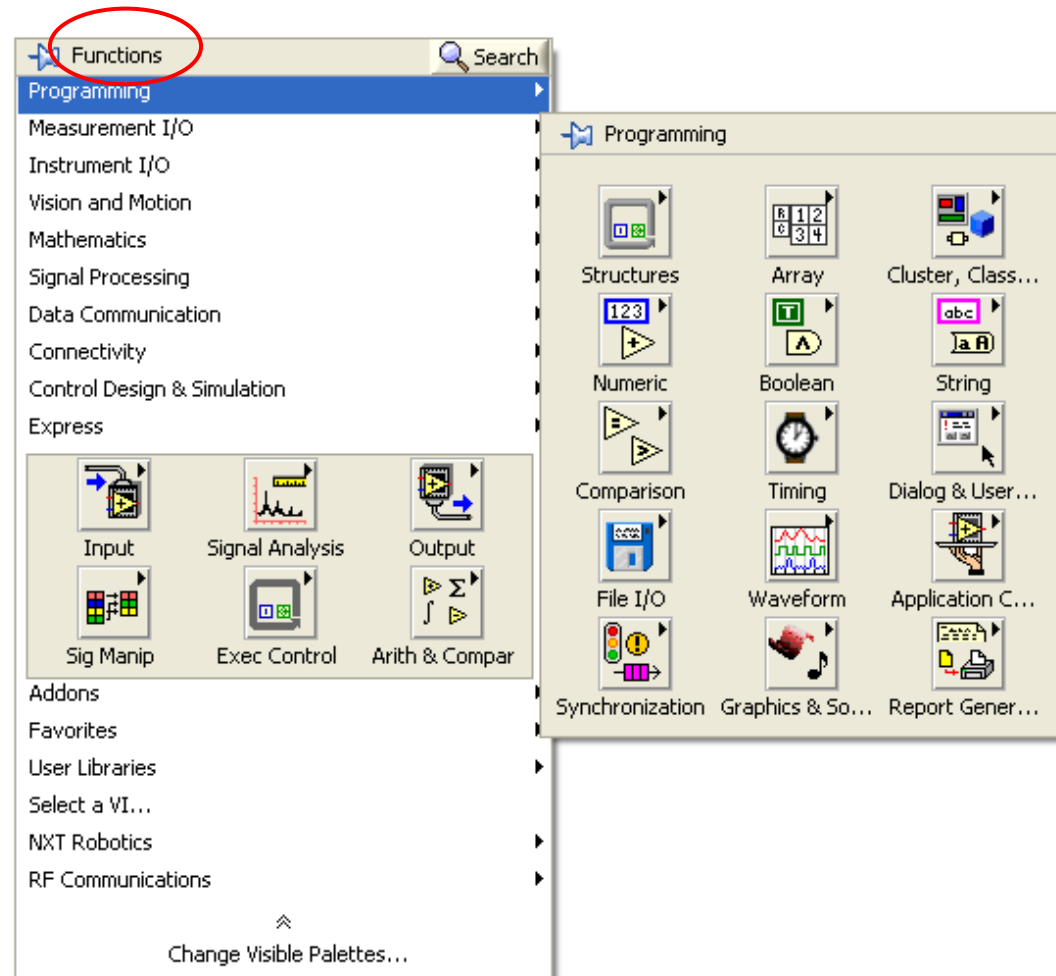
# LabVIEW – Front Panel (the GUI)

Right mouse click to open important '**Controls**' palette:



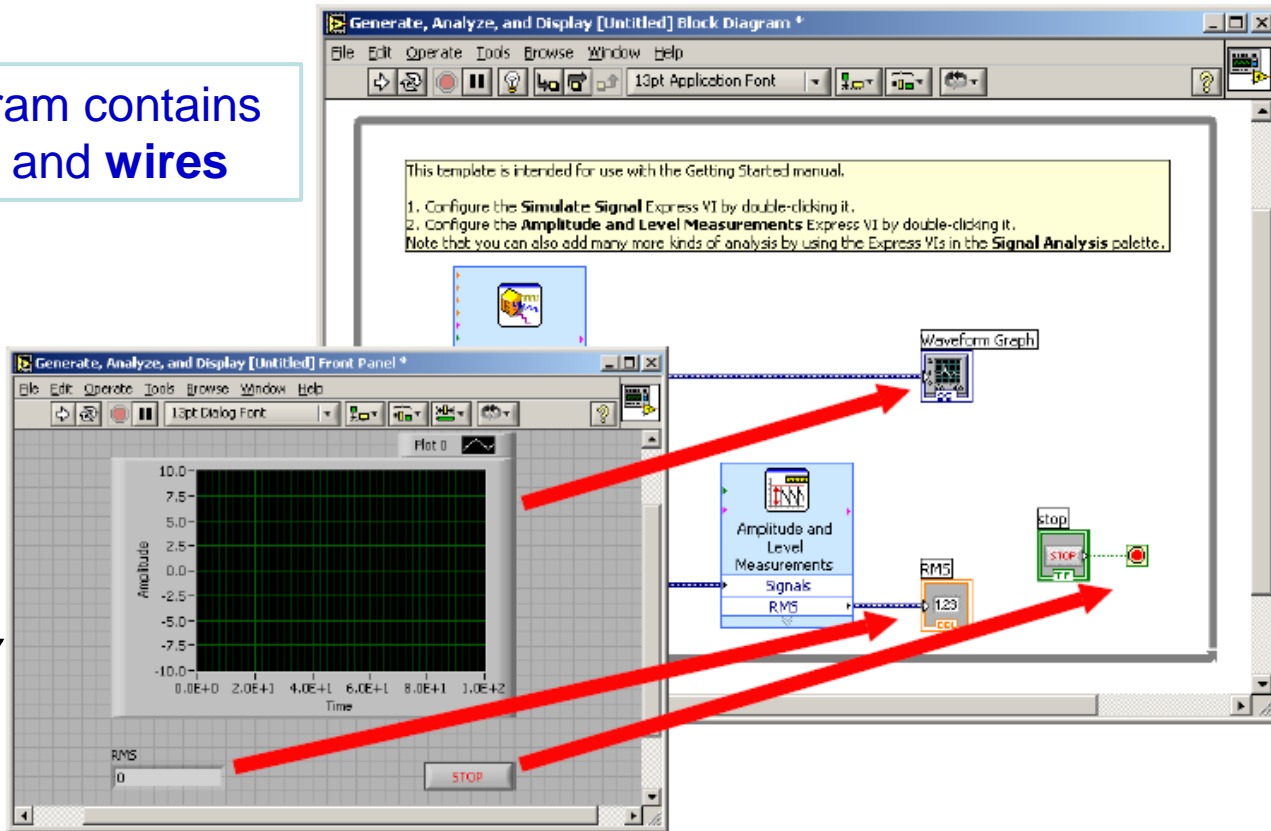
# LabVIEW– Block Diagram

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



# Relationship between Block Diagram and Front Panel

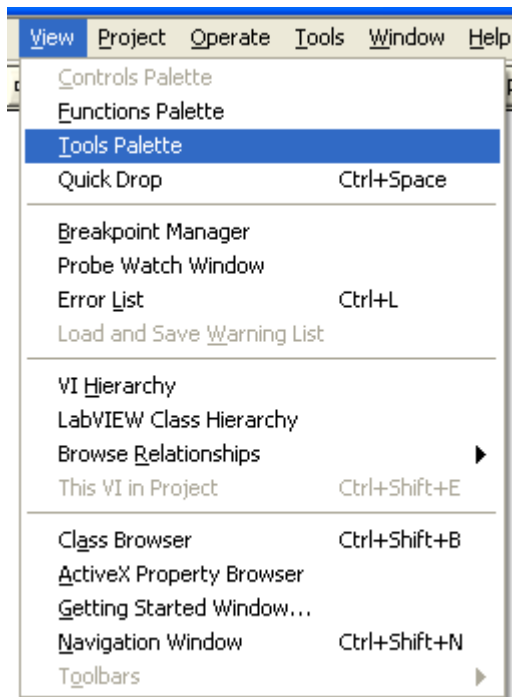
The block diagram contains **Icons** (objects) and **wires**



GUI (Front Panel)

# Tools palette

- Used on Block Diagram & Front Panel



Wiring tool to wire icons together on the block diagram

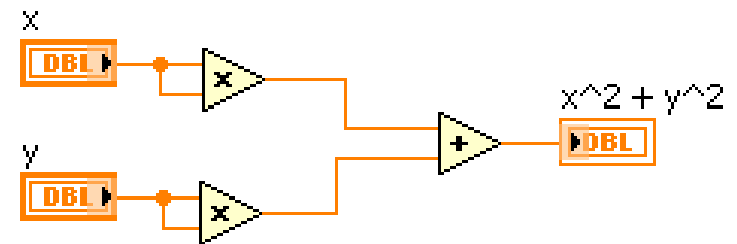
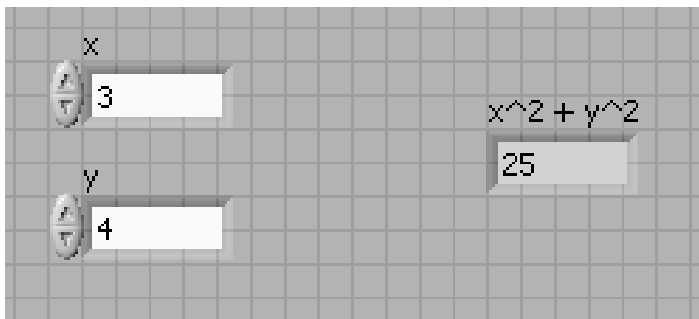


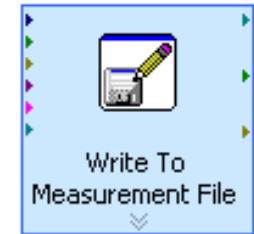
Write Text





# Standard LabVIEW VIs - Example



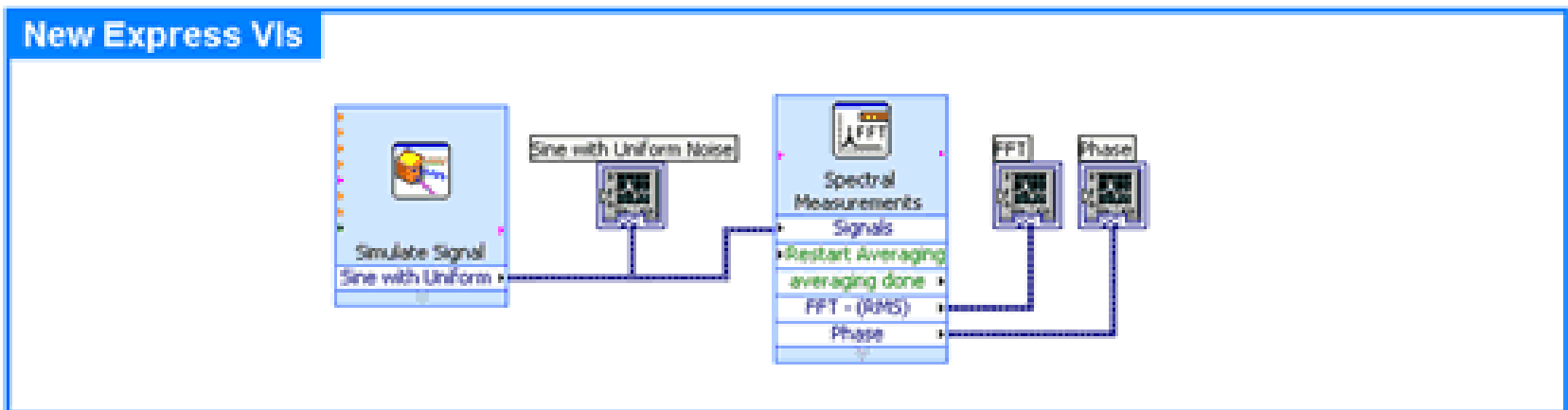
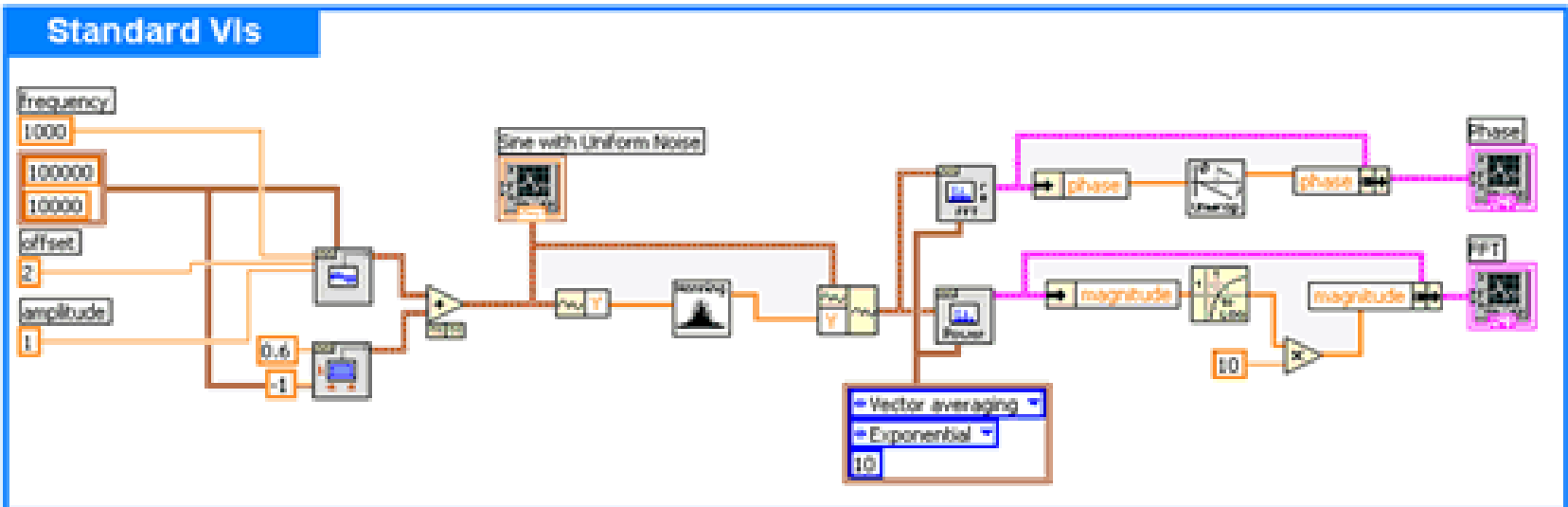


# LabVIEW Express VIs

- 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

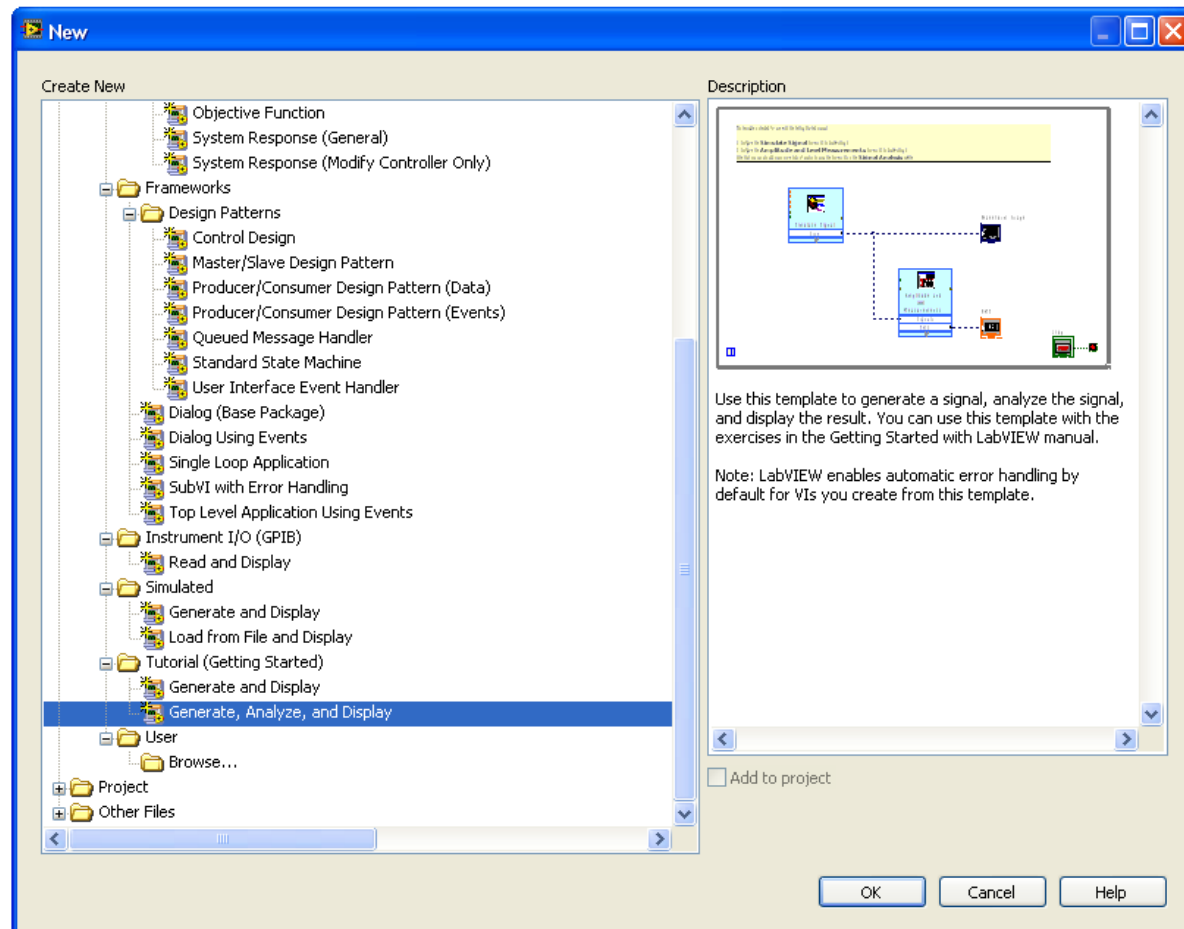
# Standard VIs vs. Express VIs

## LabVIEW-based Measurement Analysis



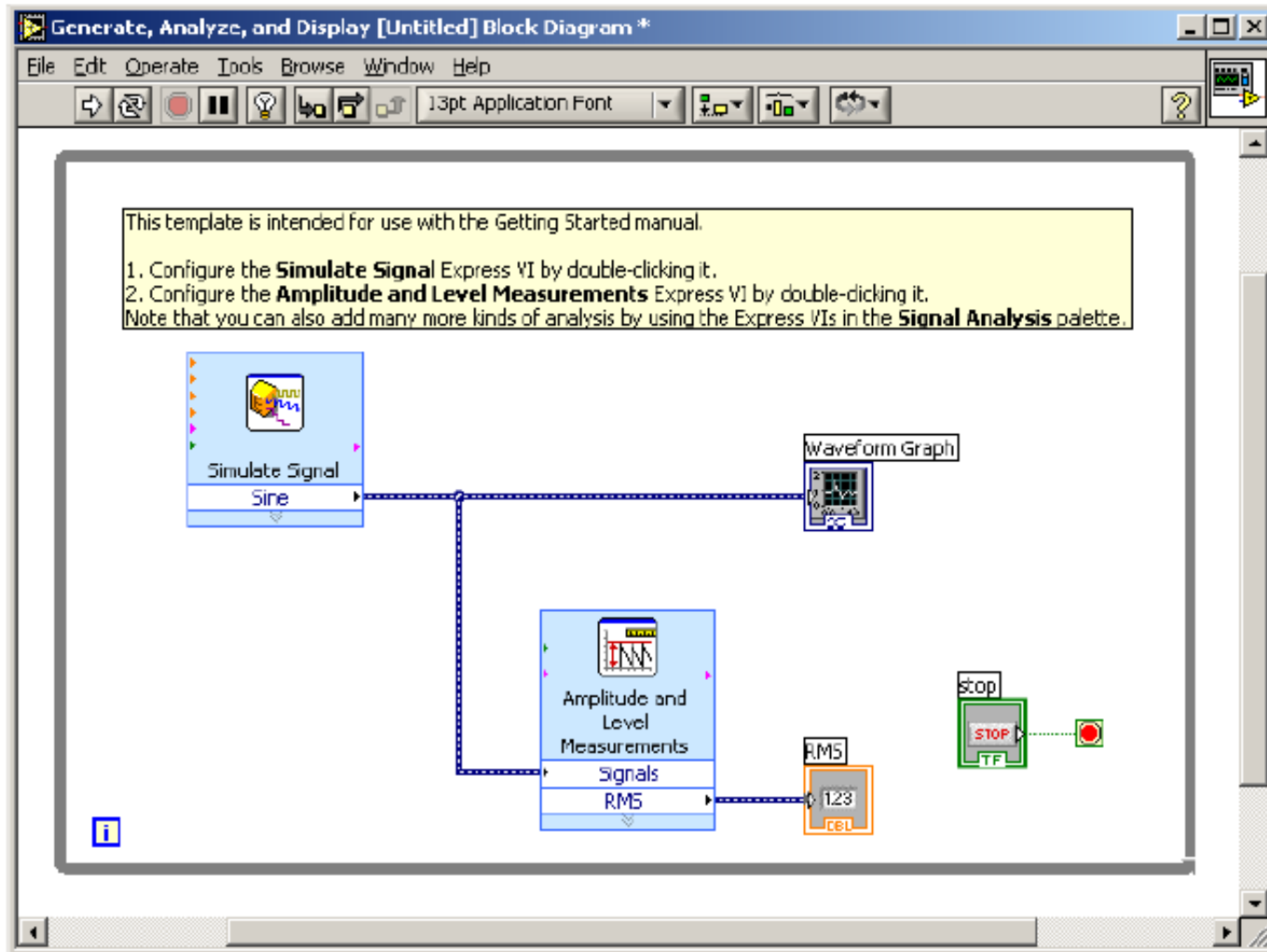
# Example with Express VIs

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



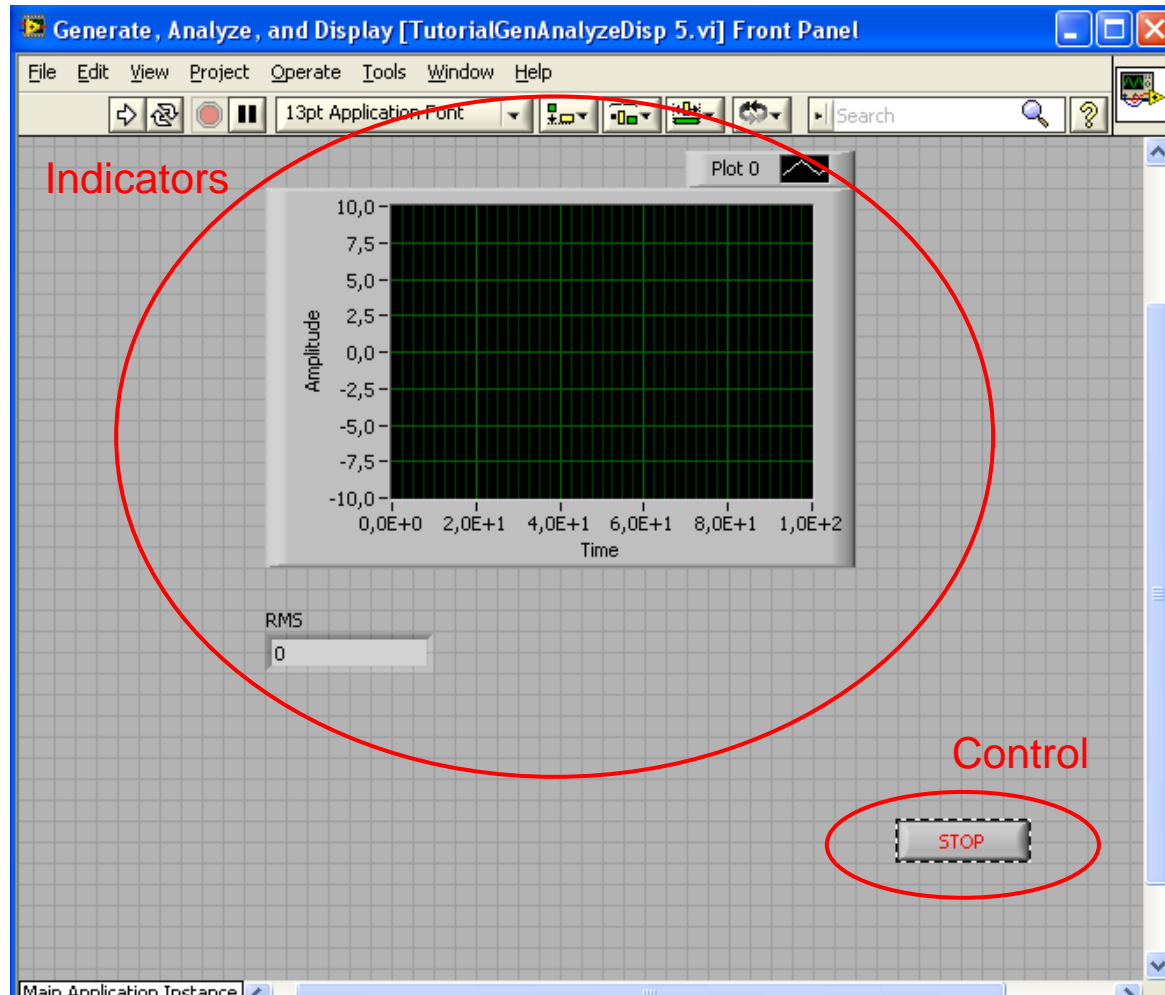
# Example with Express VIs II

## Block Diagram:



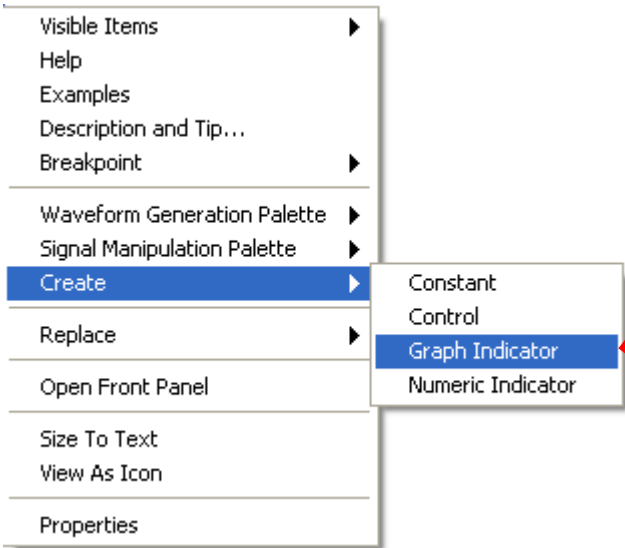
# Example with Express VIs III

## Front Panel:



# LabVIEW - Indicators

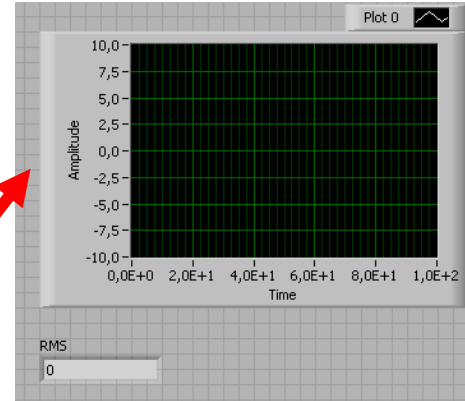
- Select the wiring tool 
- Right-click the VIs I/O connections



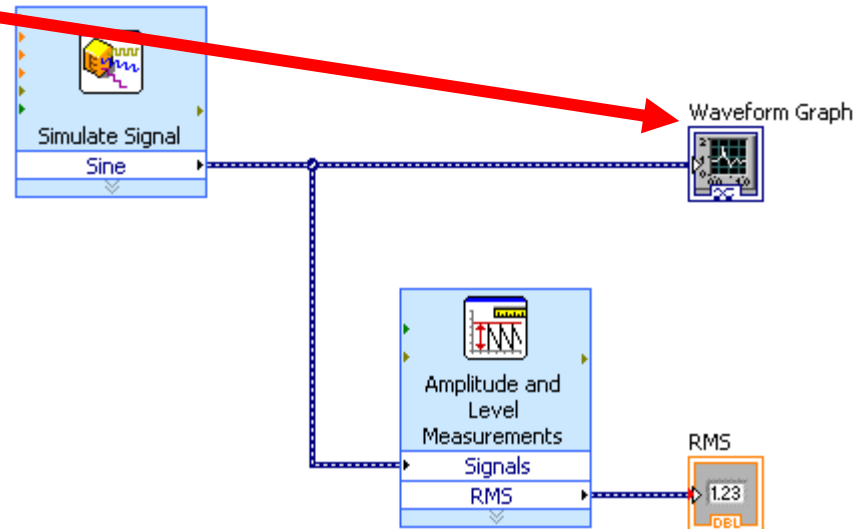
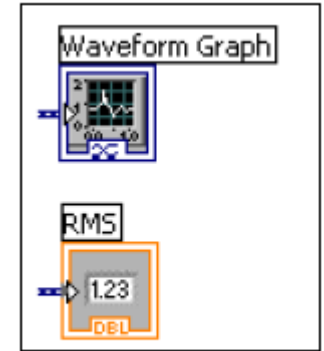
Visible Items ▶  
 Help  
 Examples  
 Description and Tip...  
 Breakpoint ▶  
 Waveform Generation Palette ▶  
 Signal Manipulation Palette ▶  
**Create ▶**  
 Replace ▶  
 Open Front Panel  
 Size To Text  
 View As Icon  
 Properties

Constant  
 Control  
**Graph Indicator**  
 Numeric Indicator

Front Panel

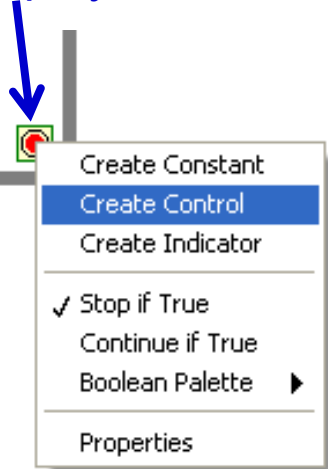


Block Diagram



# LabVIEW - Controls

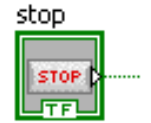
Right-click the While Loop's "stop symbol"



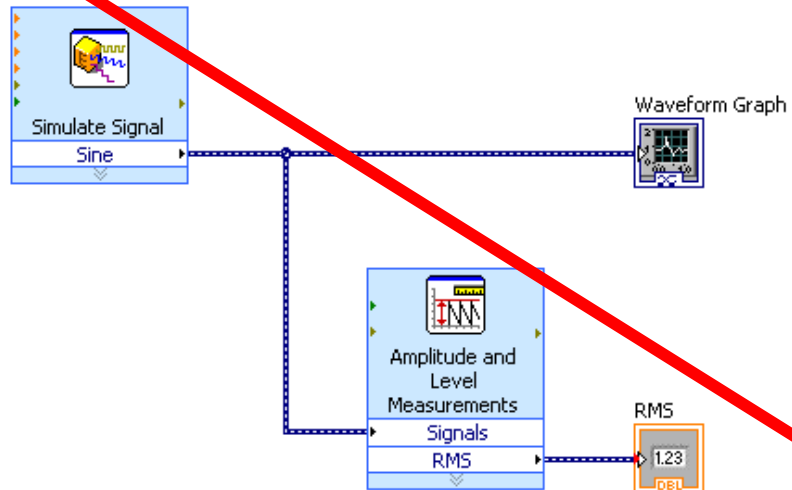
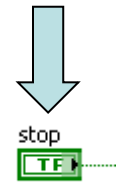
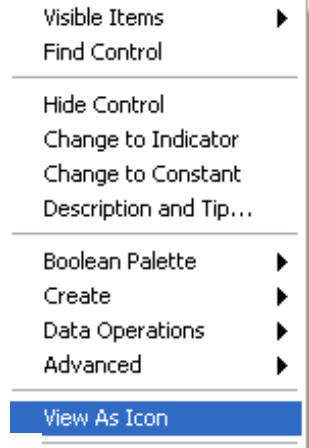
Front Panel



Block Diagram



Right-click






# LabVIEW Data Types

- Color coding of terminals and block diagram wires

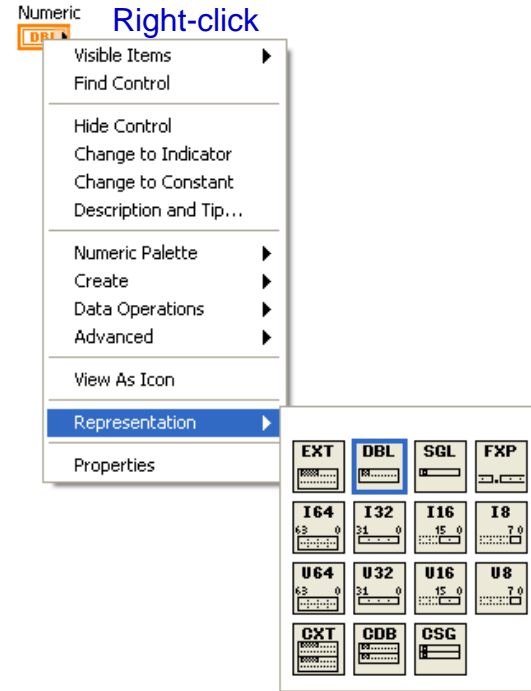
 Floating-point numbers

 Integer

 String

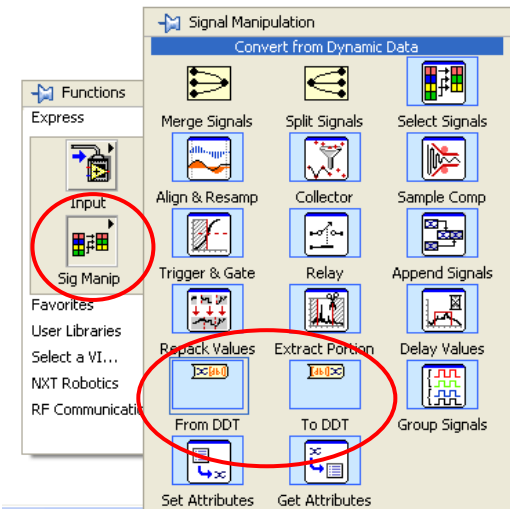
 Dynamic

 Boolean



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

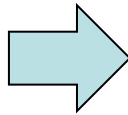
Use the Convert from/to Dynamic Data Express VI to convert dynamic data to/from numeric, Boolean, waveform, and array data for use with other VIs and functions



# 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

Number of samples: 100       Run as fast as possible

Automatic

Integer number of cycles

Actual number of samples: 100

Actual frequency: 10,1

**Result Preview**

Amplitude vs Time graph showing a sine wave oscillating between -1 and 1 over a time range of 0 to 0,099.

**Time Stamps**

Relative to start of measurement

Absolute (date and time)

**Reset Signal**

Reset phase, seed, and time stamps

Use continuous generation

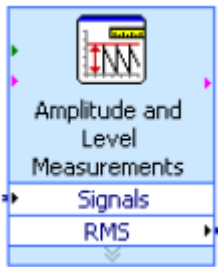
**Signal Name**

Use signal type name

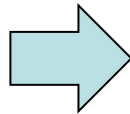
Signal name: Sine

OK      Cancel      Help

# 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

**Input Signal**

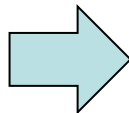
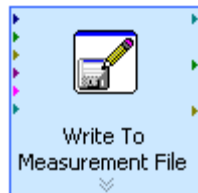
**Results**

Measurement	Result
RMS	1.004861

**Result Preview**

OK Cancel Help

# 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

**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

**Action**

- Save to one file
  - Ask user to choose file
    - Ask only once
    - Ask each iteration
- Save to series of files (multiple files)

**If a file already exists**

- Rename existing file
- Use next available filename
- Append to file
- Overwrite file

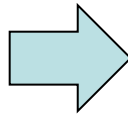
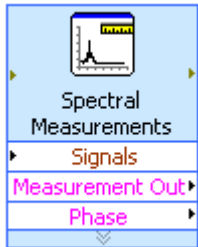
Settings...

**File Description**

Advanced...

OK Cancel Help

# LabVIEW – FFT Express VI



**Configure Spectral Measurements**

**Selected Measurement**

Magnitude (RMS)  
 Magnitude (Peak)  
 Power spectrum  
 Power spectral density

**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

**Result**

Linear  
 dB

**Windowed Input Signal**

Amplitude vs Time

**Magnitude Result Preview**

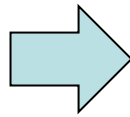
Magnitude vs Frequency

**Phase Result Preview**

Phase (rad.) vs Frequency

OK Cancel Help

# 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**  
Amplitude vs Time (0 to 1). **Sample Data**

**Result Preview**  
Amplitude vs Time (0 to 1). **Sample Result**

**View Mode**  
 Signals  Show as spectrum  
 Transfer function

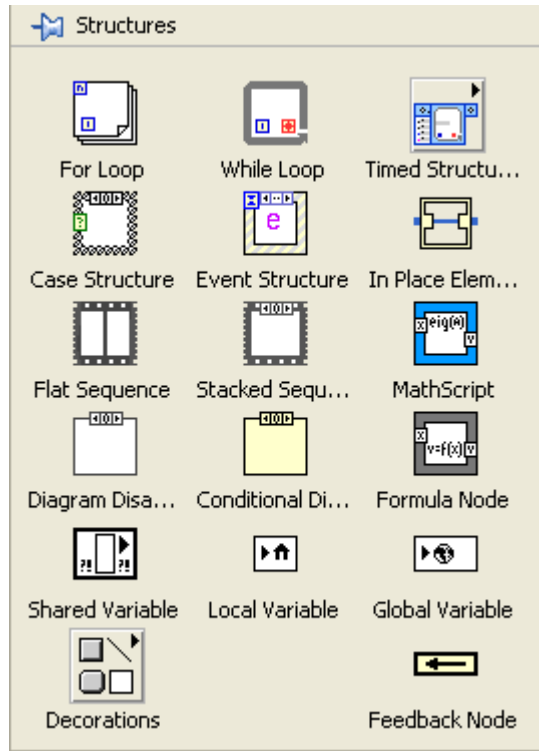
**Scale Mode**  
 Magnitude in dB  
 Frequency in log

OK Cancel Help

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

# Loops

## Functions – Programming – Structures:



## While Loop:



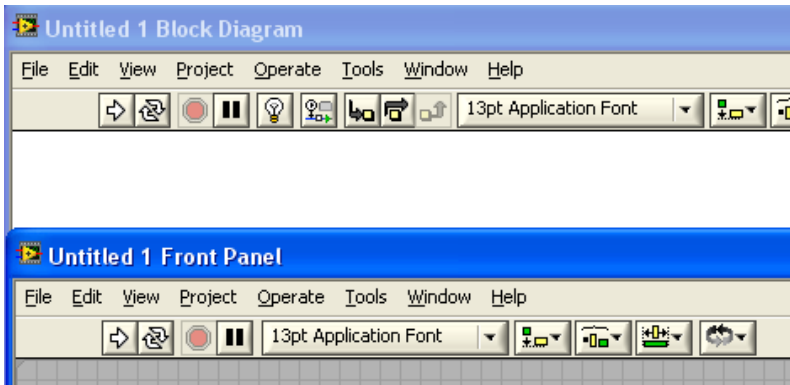
## For Loop:



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



# Program Start, Abort execution and Error indication



**Start (RUN) program button**



**Program Running indicator**



**Broken arrow – error in program**



**Abort Execution button**



- 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.



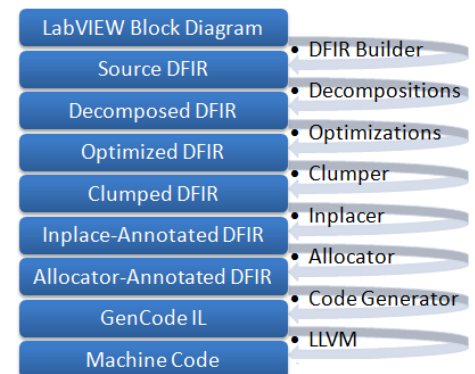
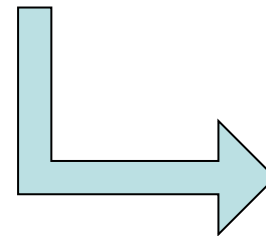


# How VIs are compiled

- 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

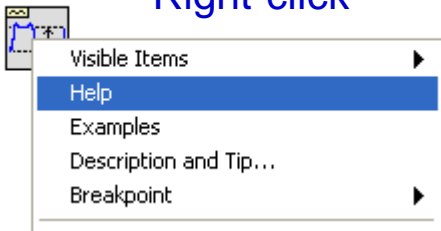


**Start (RUN) program button**



# Standard LabVIEW VI – Help

Right-click



**LabVIEW Help**

Hide Locate Back Forward Options

Contents Index Search Favorites

- LabVIEW Help
- Finding Example VIs
- Glossary
- LabVIEW 2010 Features and Changes
- Activating Your Software
- Using Help
- LabVIEW Documentation Resources
- Getting Started with LabVIEW
- Fundamentals
- VI and Function Reference
- Property and Method Reference
- Taking Measurements
- Controlling Instruments
- Control Design and Simulation Modules
- MathScript RT Module
- Toolkits
- NI Device Drivers
- Important Information
- Technical Support and Professional Services

---

## 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.

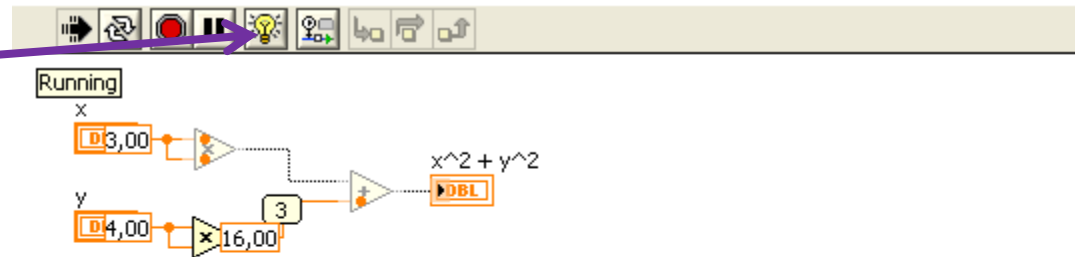
0 **Histogram**—Returns the levels of the histogram bins with the maximum number of hits in the upper and lower regions of the waveform. The upper and lower regions of the waveform include the upper and lower 40%, respectively, of the peak-to-peak range of the waveform.

1 **Peak**—Searches the entire waveform for its maximum and minimum levels.

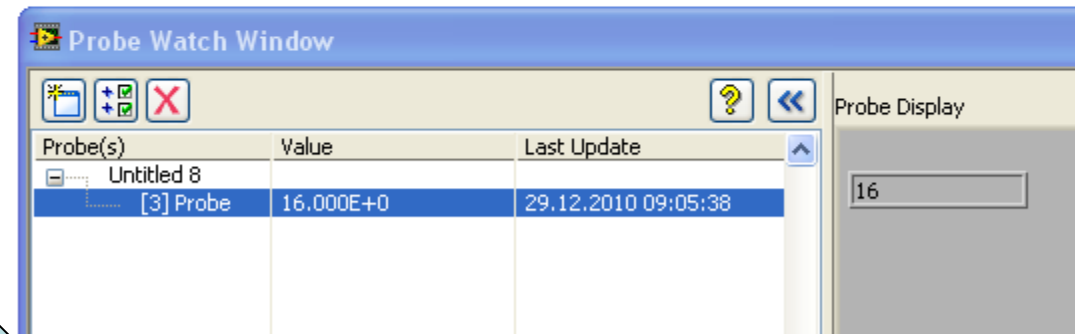
2 **Auto select (default)**—Determines whether the histogram bins that correspond to the high and low state levels each have over 5% of the total hits. If so, LabVIEW returns those results. Otherwise, LabVIEW uses the **peak** method. This ensures a reasonable answer for either a square wave (ignoring the overshoot and undershoot) or a triangle wave (where a histogram fails).

# LabVIEW debugging techniques

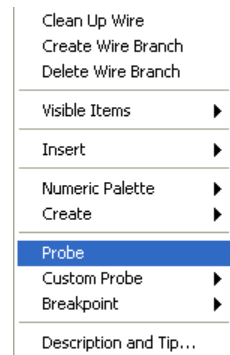
- Execution highlighting



- Single-stepping

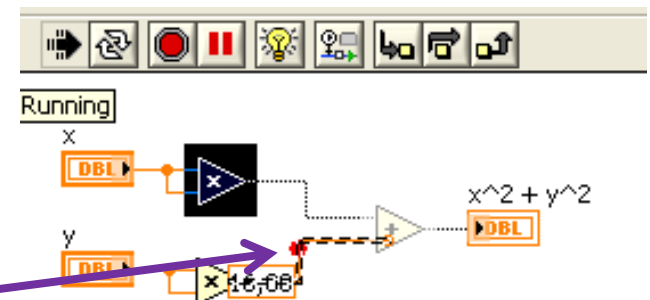


- Probe



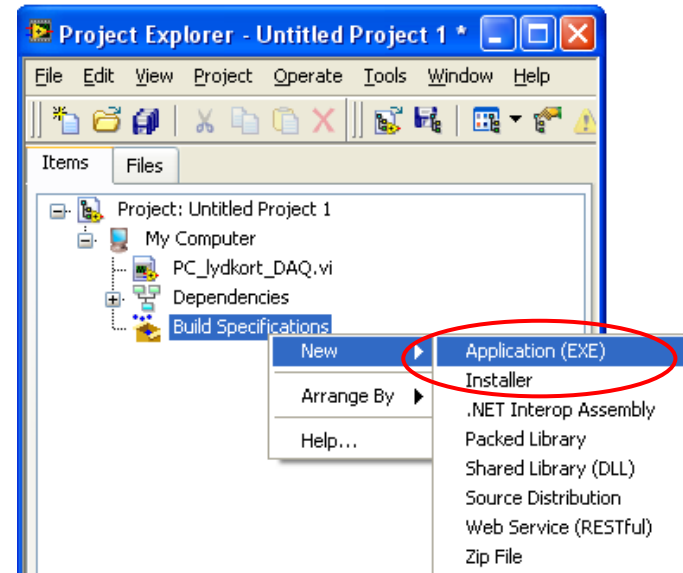
Right-click wire

- Breakpoints



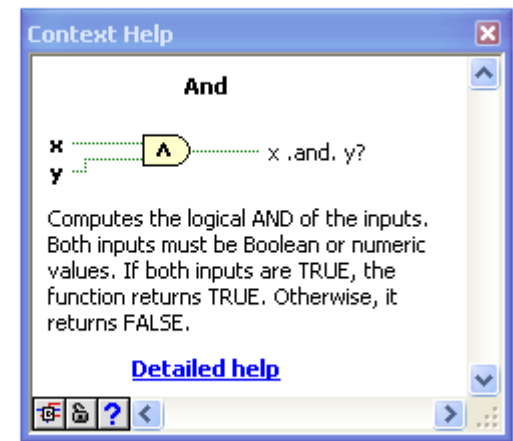
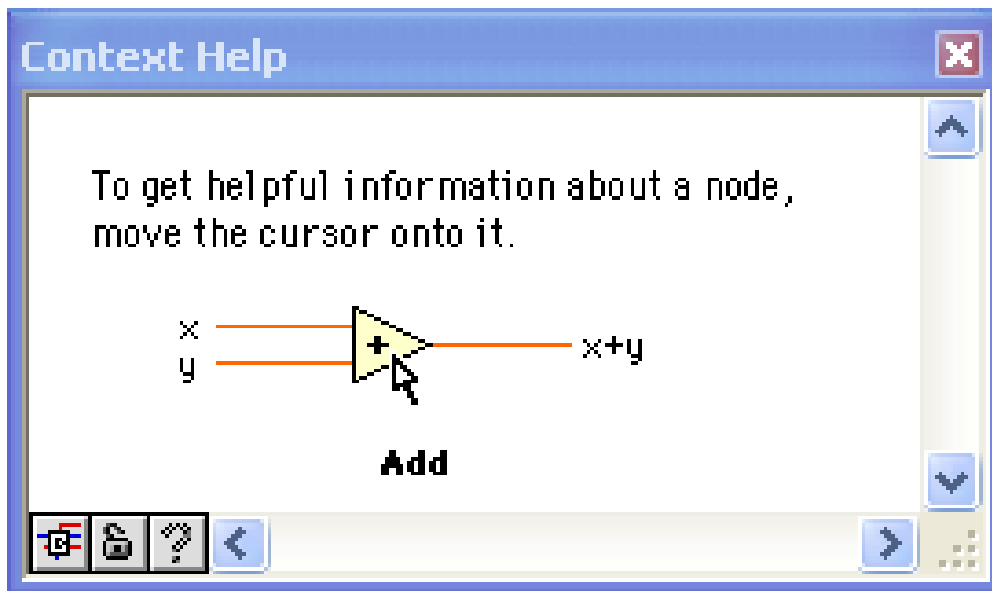
# Projects in LabVIEW & Executables

- 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 ..**
- Create a stand-alone windows application (e.g. \*.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 distribute the LabVIEW Run-Time Engine with the application or shared library. You can also include the LabVIEW Run-Time Engine in an **installer**

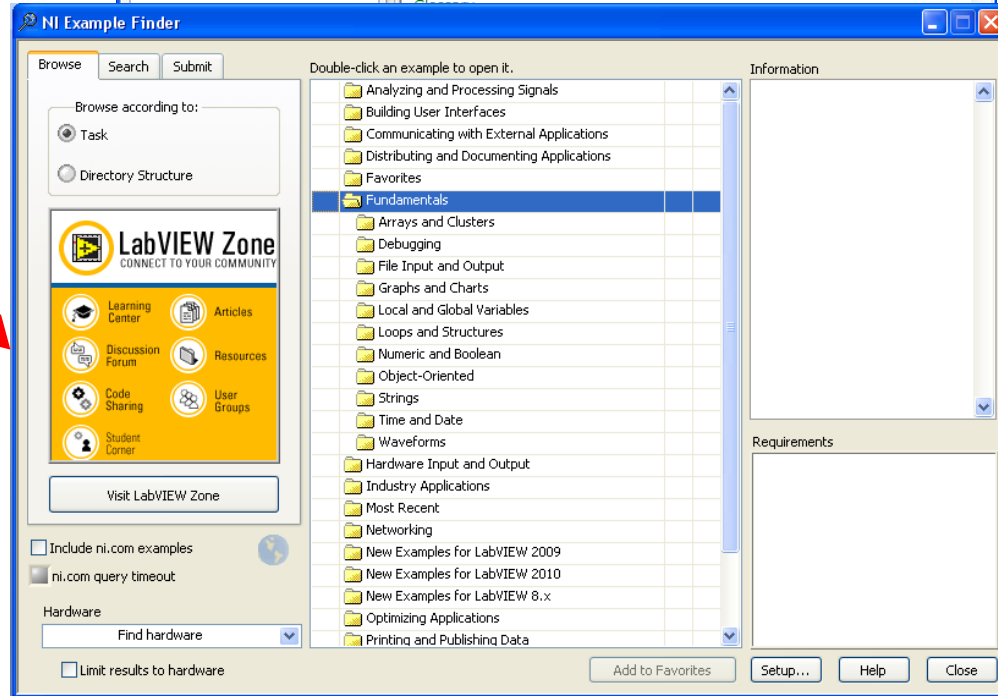
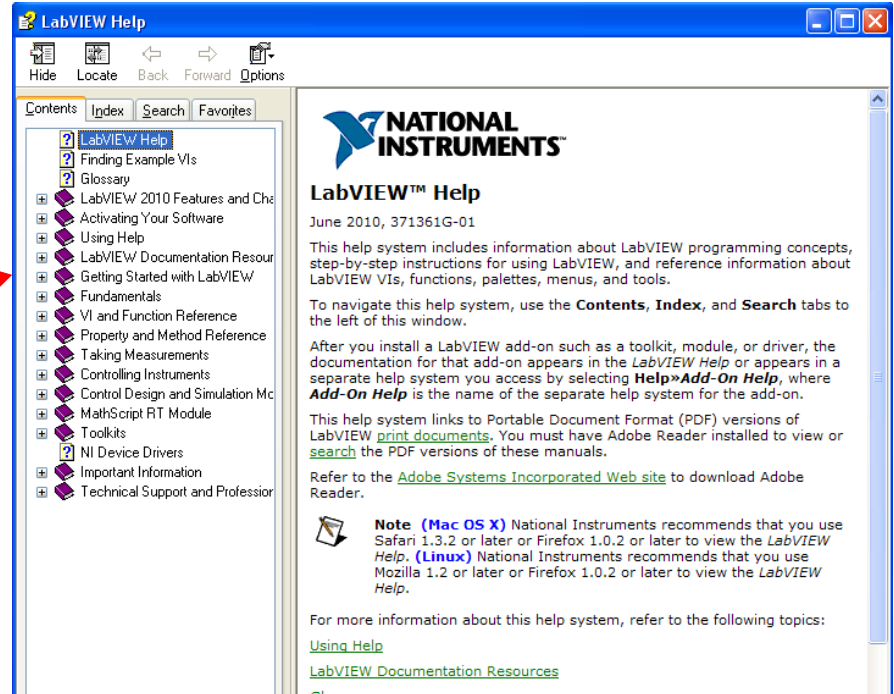
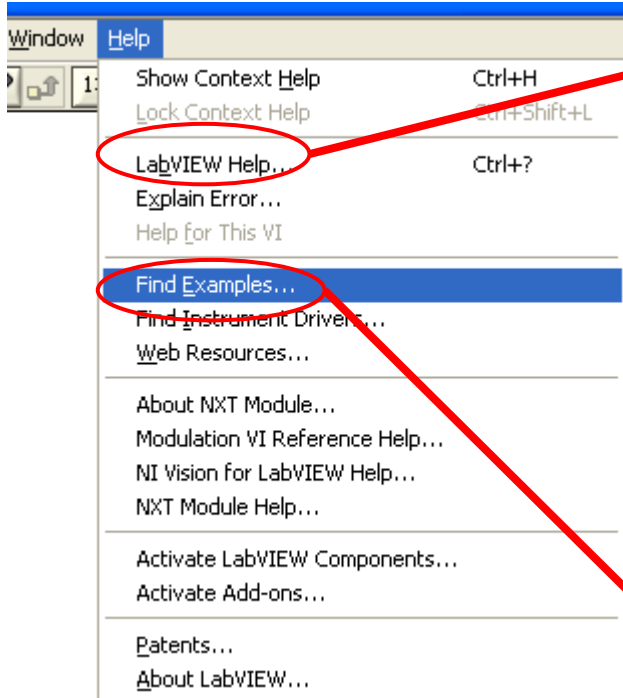


# LabVIEW – Help I

- 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:



# LabVIEW Help II



# LabVIEW Help III - www.ni.com



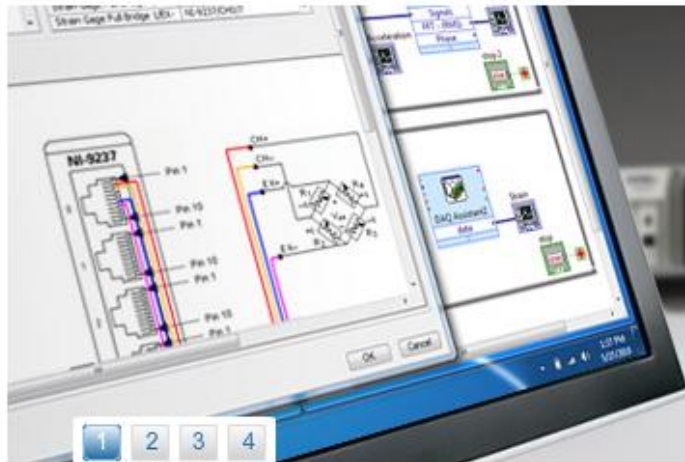
Contact NI | Norway

Products & Services Solutions **Support** **NI Developer Zone** Academic Events Company

Search

My Profile MyNI Notifications Parts List Cart

Improve your ni.com experience. Login or Create a user profile.



## From Setup to First Measurement

See How NI Can Help

[View the getting started guide](#)

### Products and Solutions

### Services and Training

### Support and Community

#### Product Platforms

- LabVIEW
- Data Acquisition
- Modular Instruments
- PXI
- Programmable Automation Controllers (PACs)
- OPIR and Instrument Control

#### Applications

- Automated Test
- Industrial Measurement and Control
- Embedded Design

#### Tutorials, Demos, and Case Studies

#### Featured Video



[Download Software](#)

[Watch Videos](#)

[Read News](#)

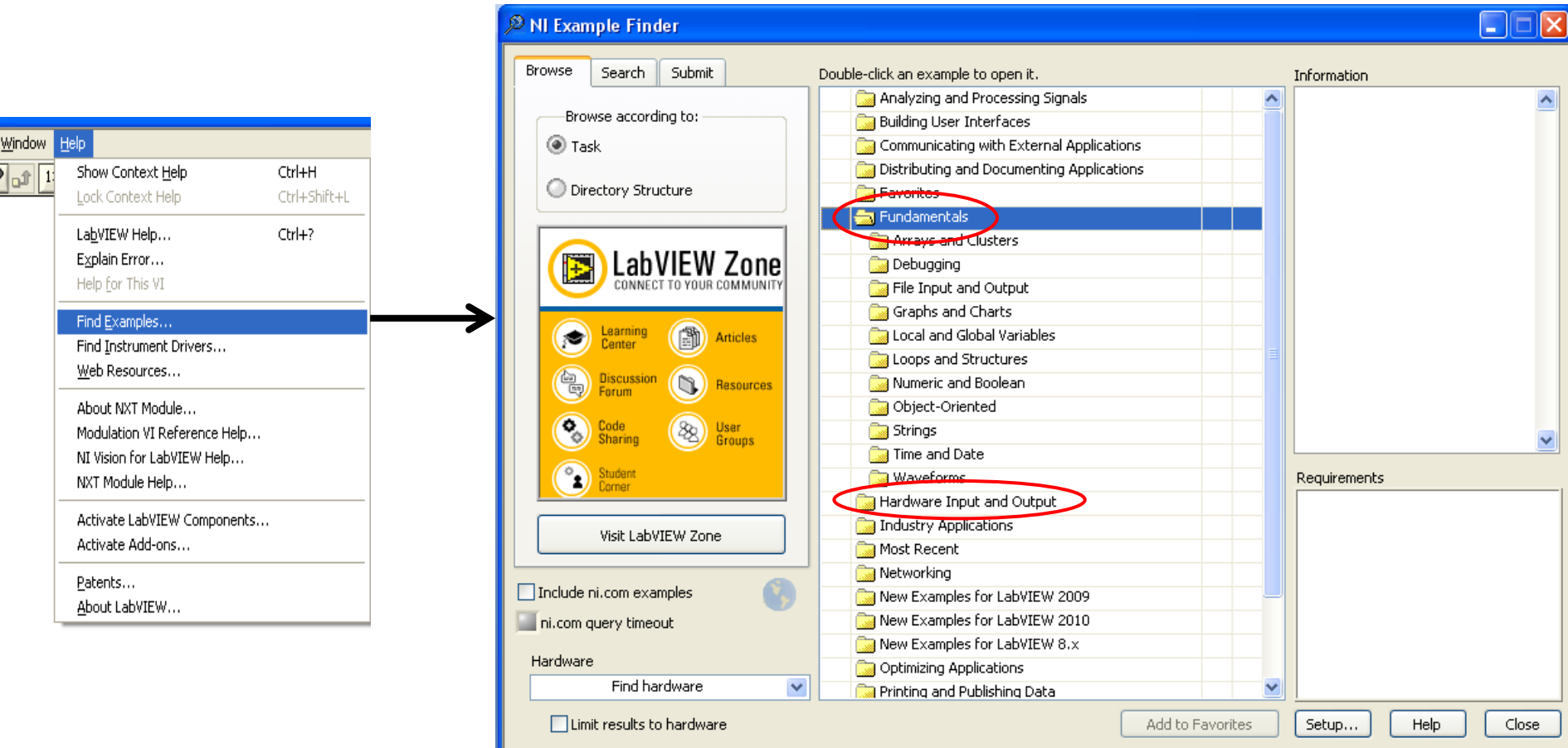
January 4, 2011  
NI Introduces PXI Express Module for High-Channel-Count Dynamic Signal Acquisition

December 13, 2010  
New NI PXI Express Chassis Deliver Added Value, Performance



# 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





# Further introduction to LabVIEW

- **"Getting Started with LabVIEW"** pdf file
- **"LabVIEW Fundamentals"** pdf file