

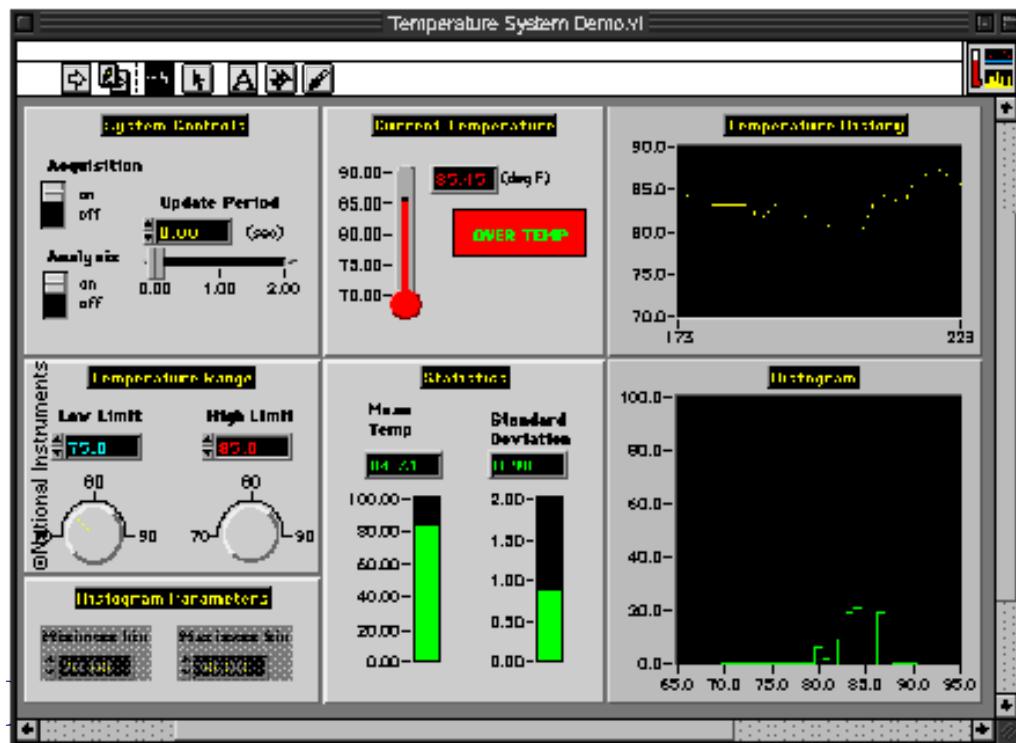
Lab 2 Notes: Introduction To Labview

•What is LabVIEW®?

- LabVIEW® is a software product produced by National Instruments®.
- It incorporates a *graphical user interface* (GUI) programming environment to produce programs that mimic laboratory instruments.
- LabVIEW is programmed with set of icons that represents controls and functions, available in the menu of the software. Such a programming is called *visual programming* and National Instruments calls it *G*.
- These programs are called *Virtual Instruments* (VI) because they imitate real benchtop instruments.
- The user interface which is called a *vi* consists of two parts- a *front panel* and a *diagram* .

•LabVIEW® Front Panel

- The *Front Panel* of the *VI* looks similar to a front panel of a real instrument.
- Front panel is used for input, output controls like a real instrument
- Using the mouse, you will move the cursor around the screen to operate switches, dials, and buttons on the *VI* just as if it were a real instrument.



Can configure a *vi* to include functions and graphs that are fully customizable.

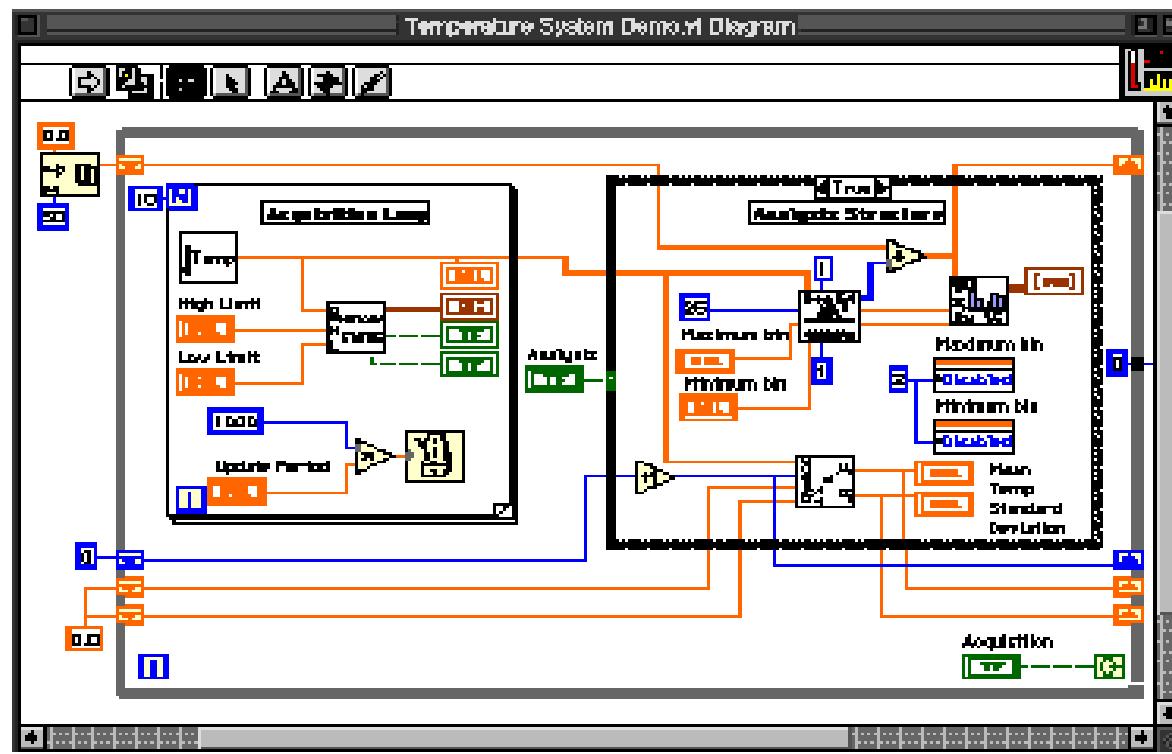
In an example shown, temperature monitoring system consists of data acquisition from a thermometer and plotting on a strip chart recorder.

The *vi* also calculates mean and standard deviation of the data.

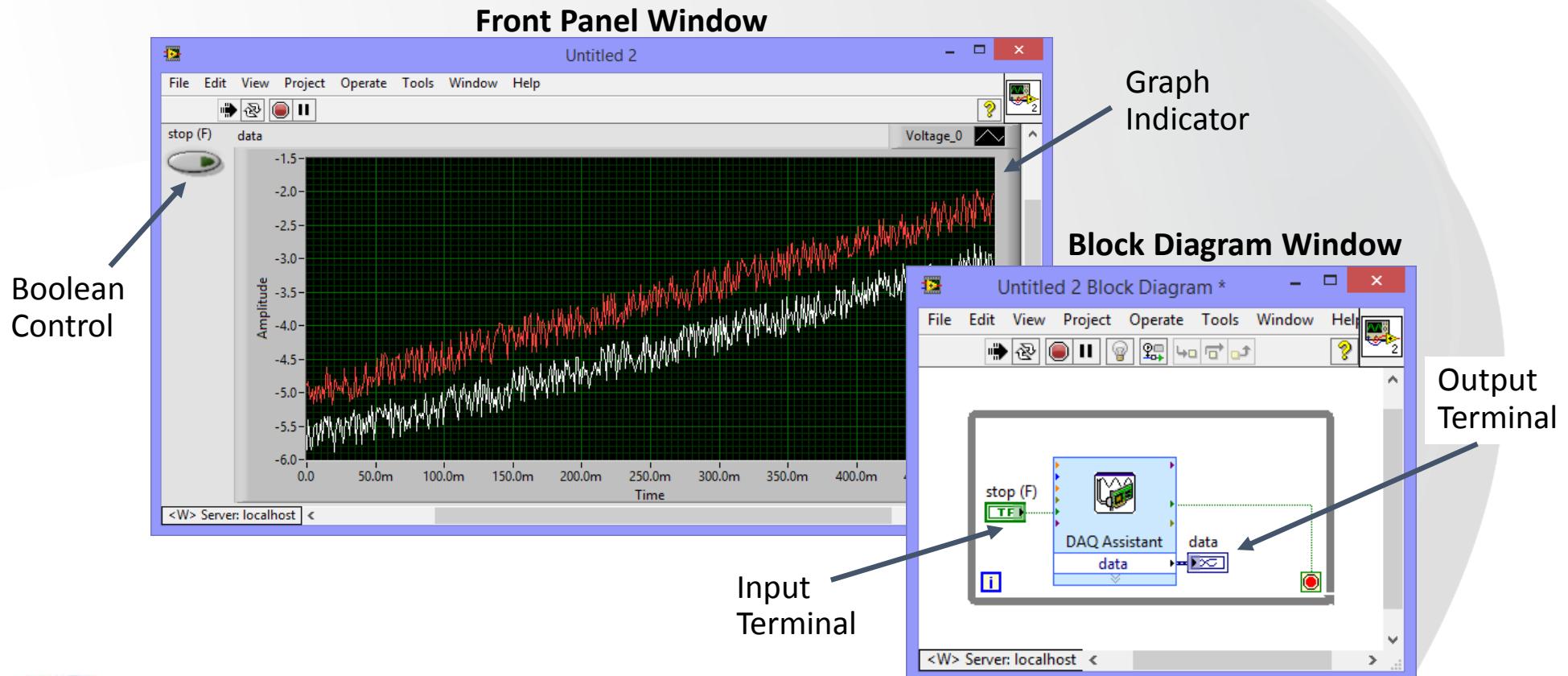
Alarms have been set up so that if the temperature falls below or above a certain set value the alarm goes ON.

•LabVIEW® Block Diagram

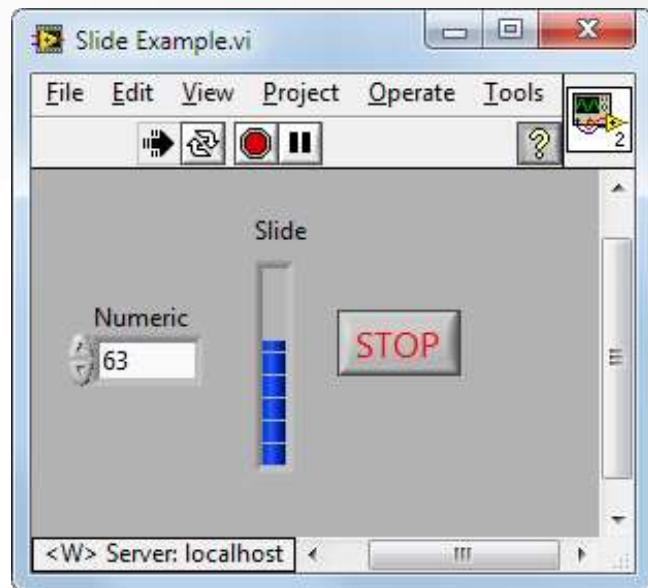
- The *diagram* of the *VI* is analogous to the electronics of an actual instrument, I.e. the circuit board.
- Diagram is where the *VI functionality* is programmed
- The diagram of such a *vi* can look very complex, and the key is using good programming practices to insure that your code is “debuggable”



Demo: Creating our first VI

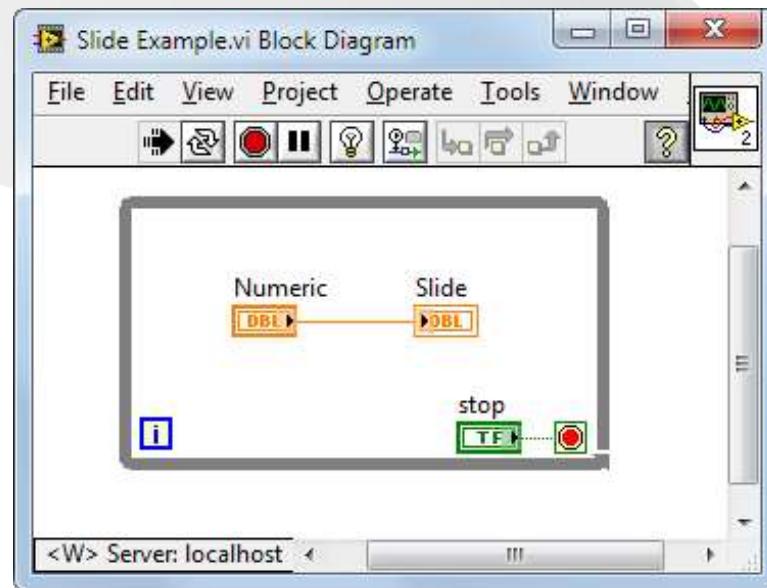


LabVIEW Environment



What is the front panel used for?

What are the inputs and outputs called?



What is the block diagram used for?

How does data travel on the block diagram?

Labview Applications

Often LabVIEW is used to perform system simulations, since it contains many commonly used filter, digital signal processing, and statistical functions. LabVIEW compiles almost as fast as C or Matlab and therefore one can perform complete simulation within a *vi*.

- In addition to data input output, LabVIEW can access digital devices like serial ports, parallel ports and GPIB cards to read data from instruments that have a GPIB interface.
- As you can see, the possibilities of "virtual instrument" are almost limitless and they expand the measurement Capabilities of any system immensely
- *Example applications*

Space shuttle ET hydrogen sensor real time monitor

X-43 Launch Panel Operator Situational Awareness Display Development

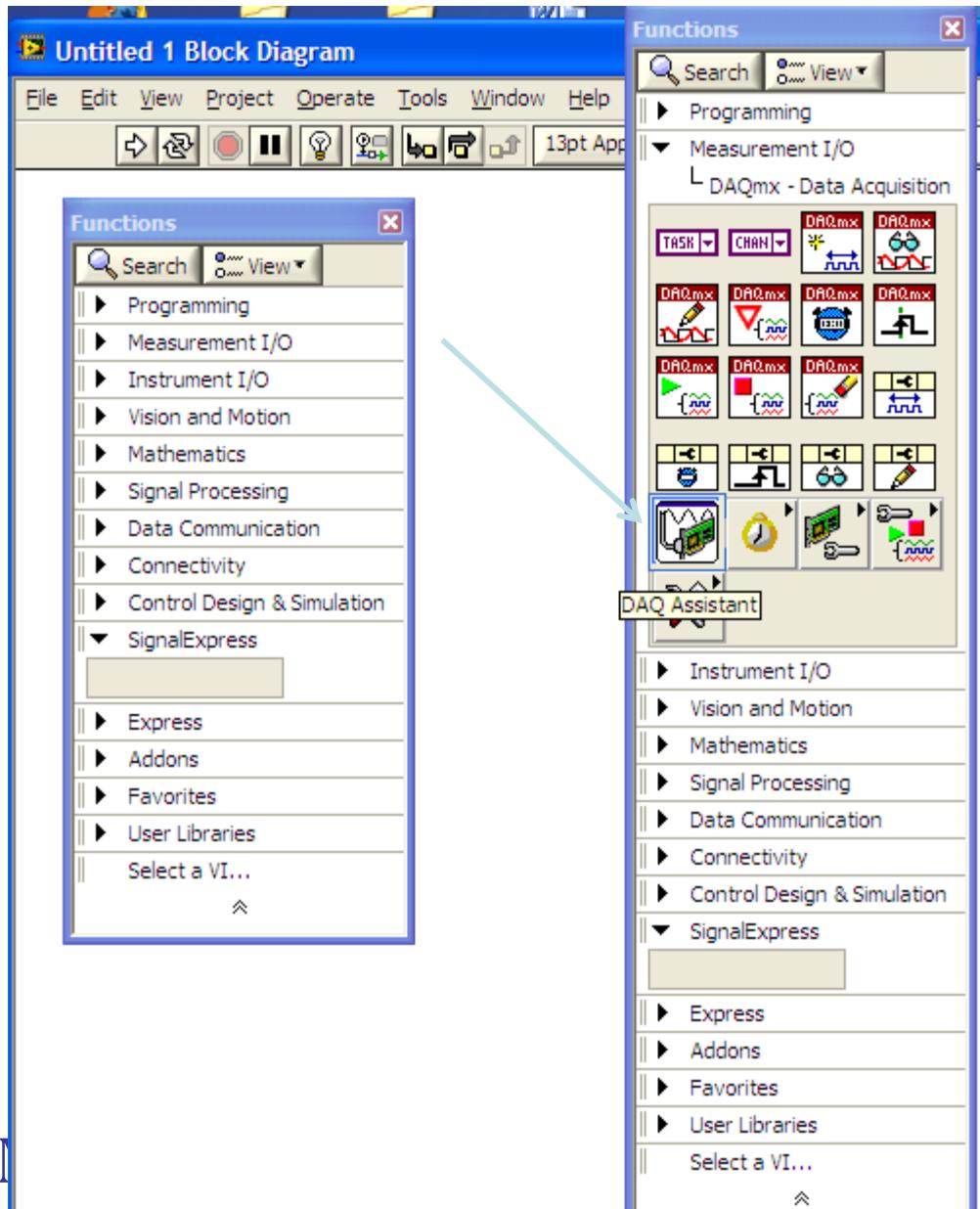
Labview Data Acquisition

“Modern” NI data acquisition is mechanized using **NI-DAQmx** drivers

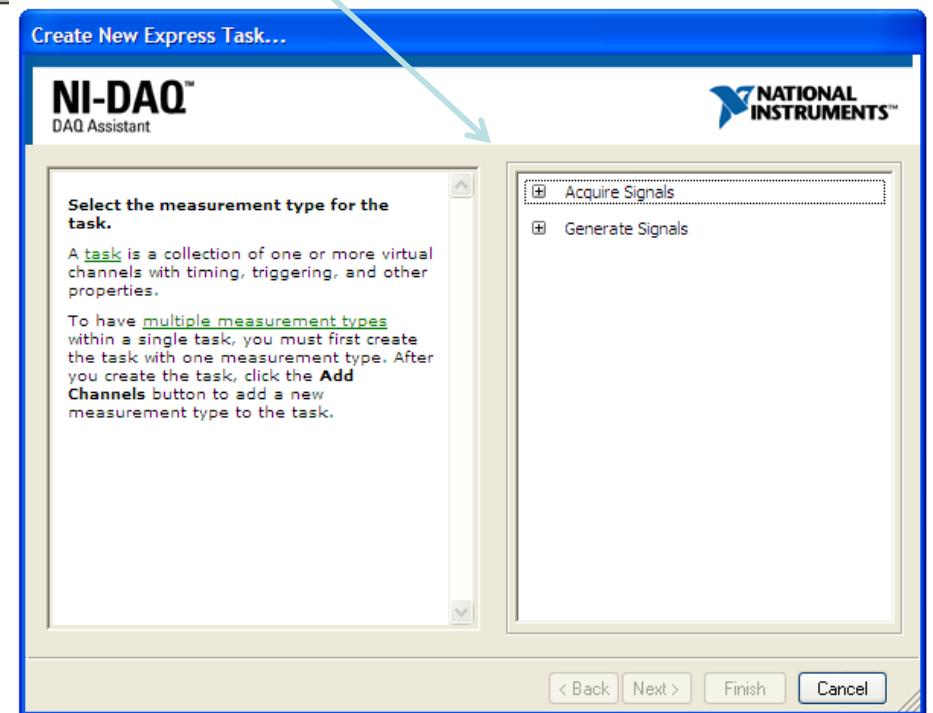
“Best friend” here is the DAQ Assistant .. It helps to set up a particular NI device for data acquisition

- **DAQ Assistant GUI** for interactively creating, editing, and running **NI-DAQmx** virtual channels and tasks.
- **NI-DAQmx virtual channel** consists of a physical channel on a DAQ device and the configuration information for this physical channel, such as input range and custom scaling.
- **NI-DAQmx task** is a collection of virtual channels, timing and triggering information, and other properties regarding the acquisition or generation.

Labview Data Acquisition (2)



Finding DAQ Assistant

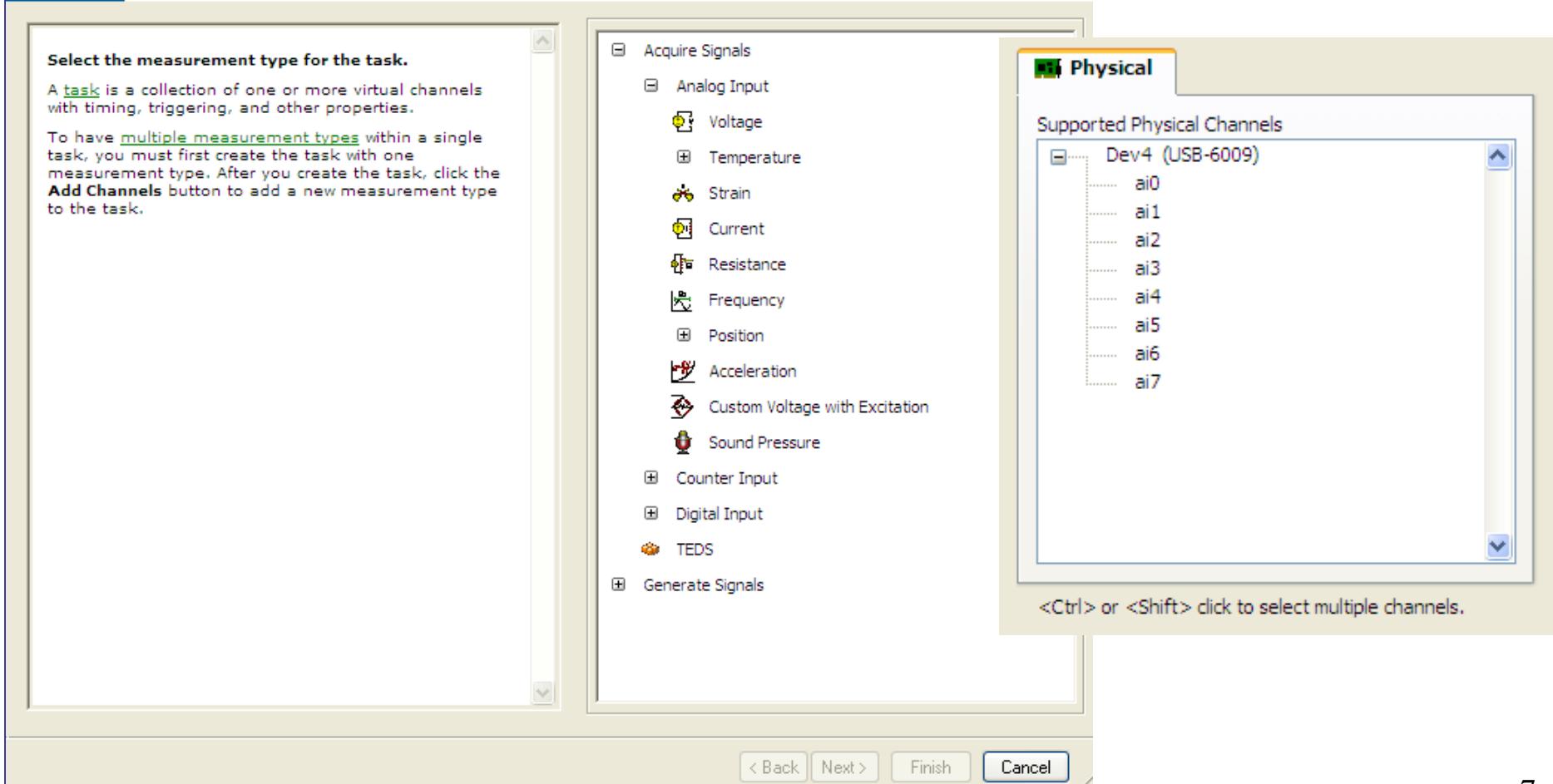


Labview Data Acquisition (3)

Create New Express Task...

NI-DAQ™
DAQ Assistant

NATIONAL
INSTRUMENTS™



Labview Data Acquisition (4)

The screenshot displays two parallel configurations of a LabVIEW DAQ task. Both configurations are titled "Voltage Input Setup".

Task 1 (Left):

- Channel Settings:** A list of channels: "Voltage" and "Voltage_0".
- Voltage Input Setup (Sub-panel):**
 - Signal Input Range:** Max: 5, Min: -5.
 - Scaled Units:** Volts.
- Terminal Configuration:** Differential.
- Custom Scaling:** <No Scale>.
- Rate (Hz):** An empty input field.
- Timing Settings:** Acquisition Mode: N Samples, Samples to Read: 1k, Rate (Hz): 1k.

Task 2 (Right):

- Channel Settings:** A list of channels: "Voltage" and "Voltage_0".
- Voltage Input Setup (Sub-panel):**
 - Signal Input Range:** Max: 50m, Min: -50m.
 - Scaled Units:** Volts.
- Terminal Configuration:** Differential.
- Custom Scaling:** <No Scale>.
- Rate (Hz):** An empty input field.
- Timing Settings:** Acquisition Mode: N Samples, Samples to Read: 1k, Rate (Hz): 1k.

Common Interface Elements:

- Configuration, Triggering, Advanced Timing tabs:** Located at the top of both main panels.
- Channel Settings table:** Shows the order, physical channel, and device type for each channel.
- Timing Settings:** Located at the bottom of both main panels, with options for Acquisition Mode (N Samples), Samples to Read (1k), and Rate (Hz) (1k).

Labview Data Acquisition (6)

The screenshot shows the LabVIEW Data Acquisition configuration interface. A red arrow points from the text "Set DAQ ranges" to the "Signal Input Range" section, which includes fields for "Max" (10) and "Min" (-10) with a "Scaled Units" dropdown set to "Volts". Another red arrow points from the text "We'll set these on front panel" to the "N Samples" dropdown in the "Timing Settings" section.

press Task Connection Diagram

Amplitude

Set DAQ ranges

100 110 120 130 140 150 160 170 180 190 200

Configuration Triggering Advanced Timing

Channel Settings

+ X S Voltage

Details >

Voltage Input Setup

Settings Calibration

Signal Input Range

Max 10

Min -10

Scaled Units Volts

Terminal Configuration Differential

Custom Scaling <No Scale>

Click the Add Channels button (+) to add more channels to the task.

Timing Settings

Acquisition Mode N Samples

Samples to Read 100

Rate (Hz) 1k

Back Measuring Voltage

Most measurement devices are designed for measuring, or reading, voltage. Two common voltage measurements are DC and AC.

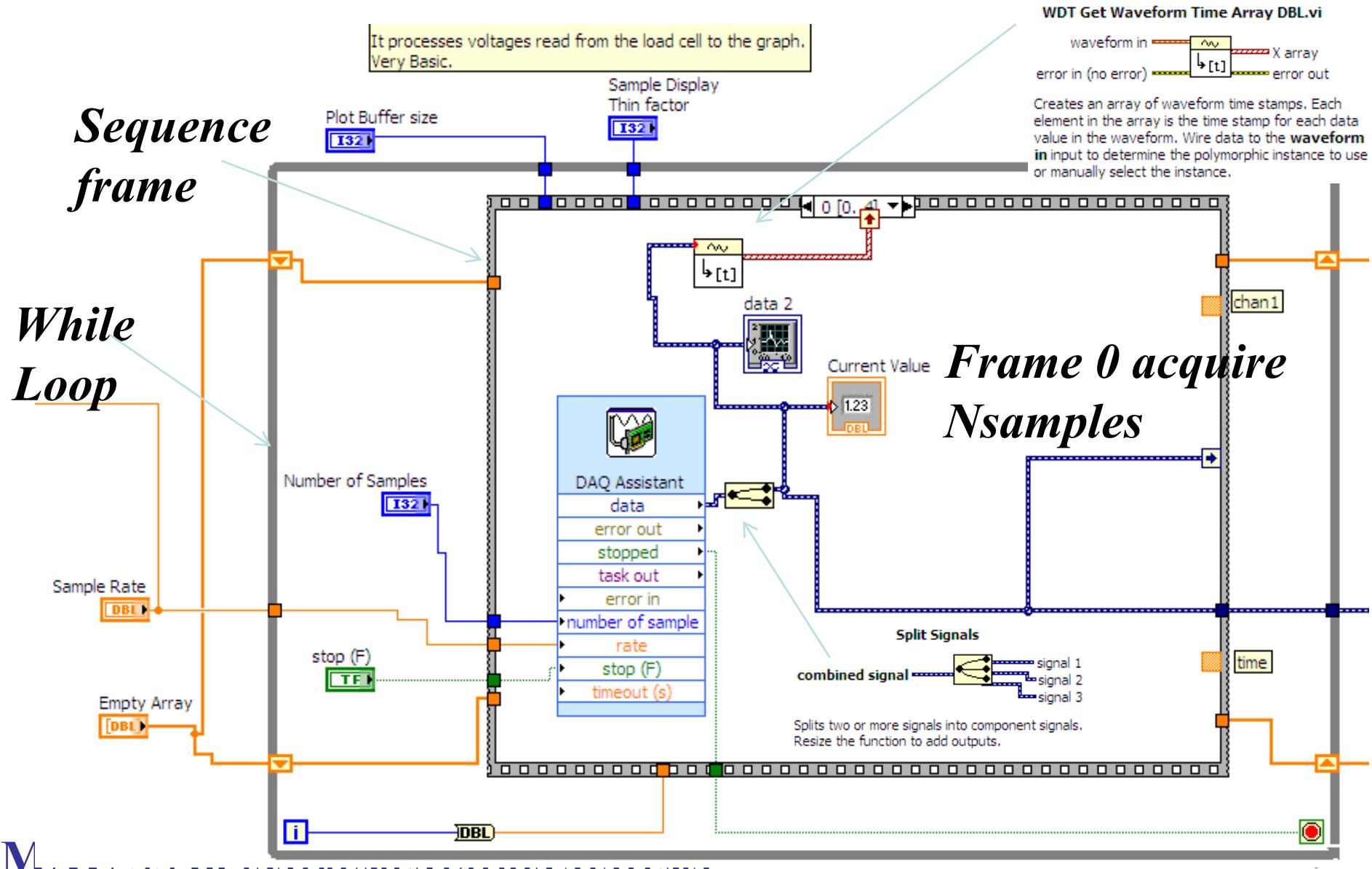
DC voltages are useful for measuring phenomena that change slowly with time, such as temperature, pressure, or strain.

AC voltages, on the other hand, are waveforms that constantly increase, decrease, and reverse polarity. Most powerlines deliver AC voltage.

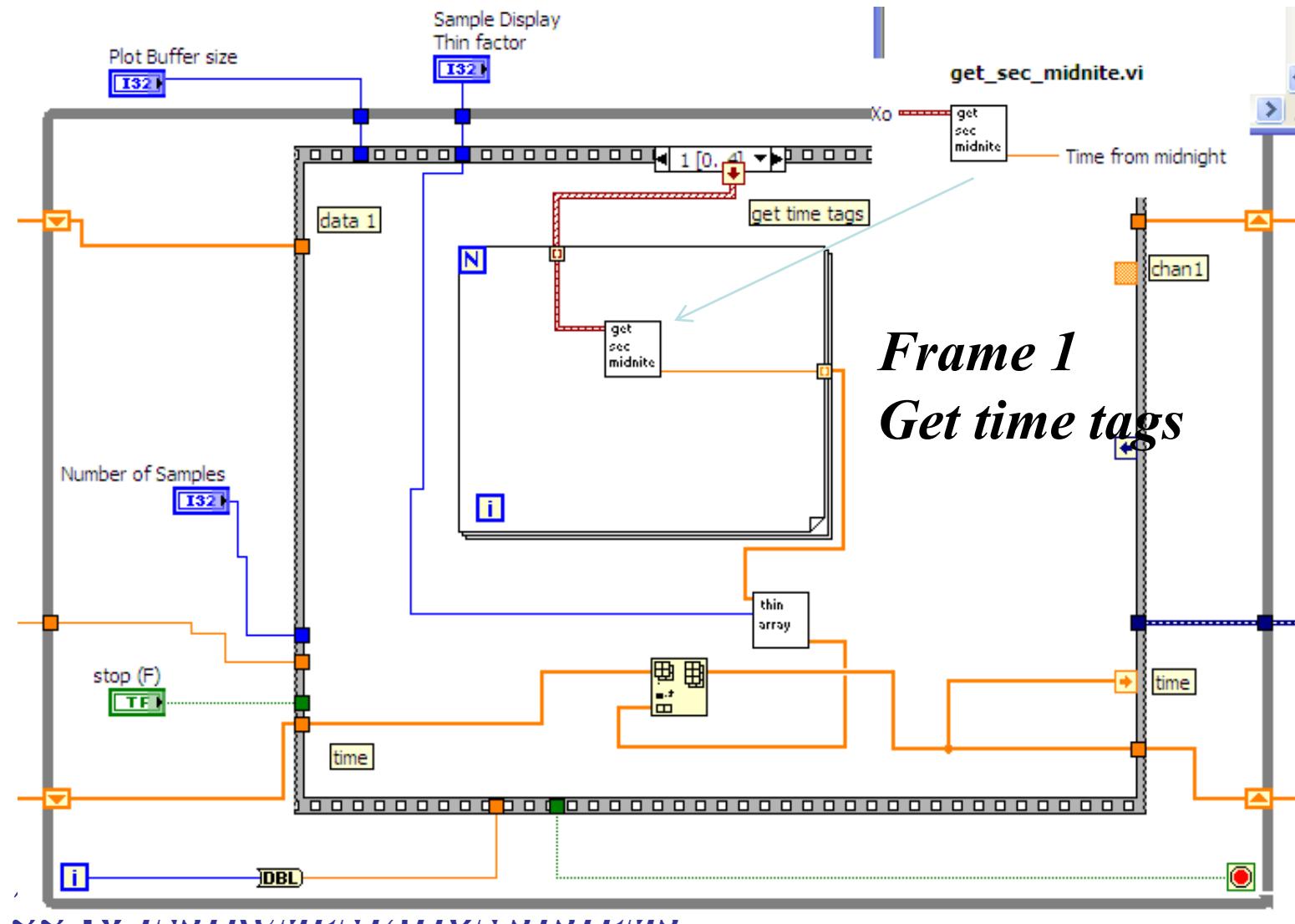
Click to add new virtual channels to the task. Additional virtual channels can be of a different measurement type or of the same measurement type as the original task.

We'll set these on front panel

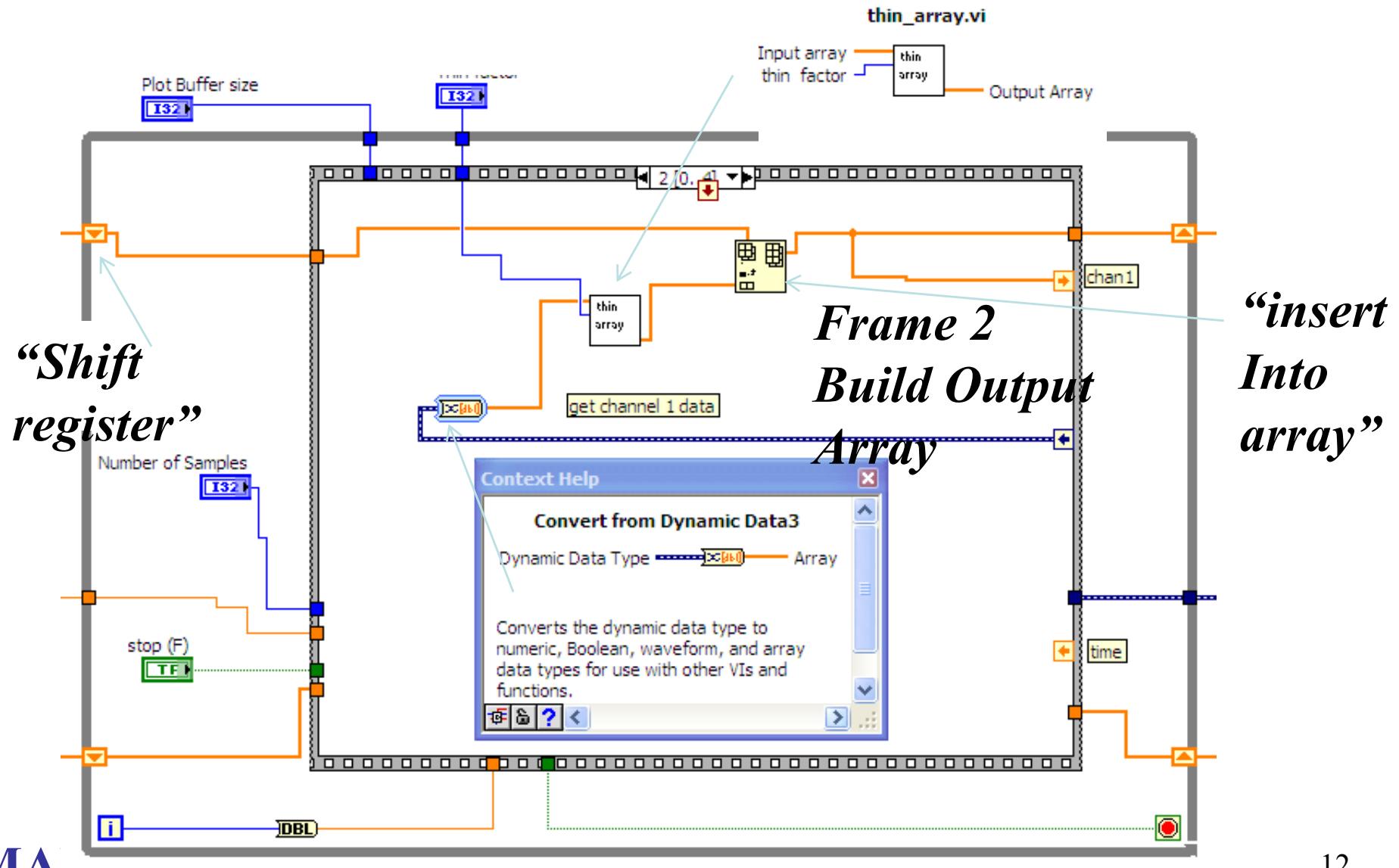
Example VI Read a Load Cells and Plots Data



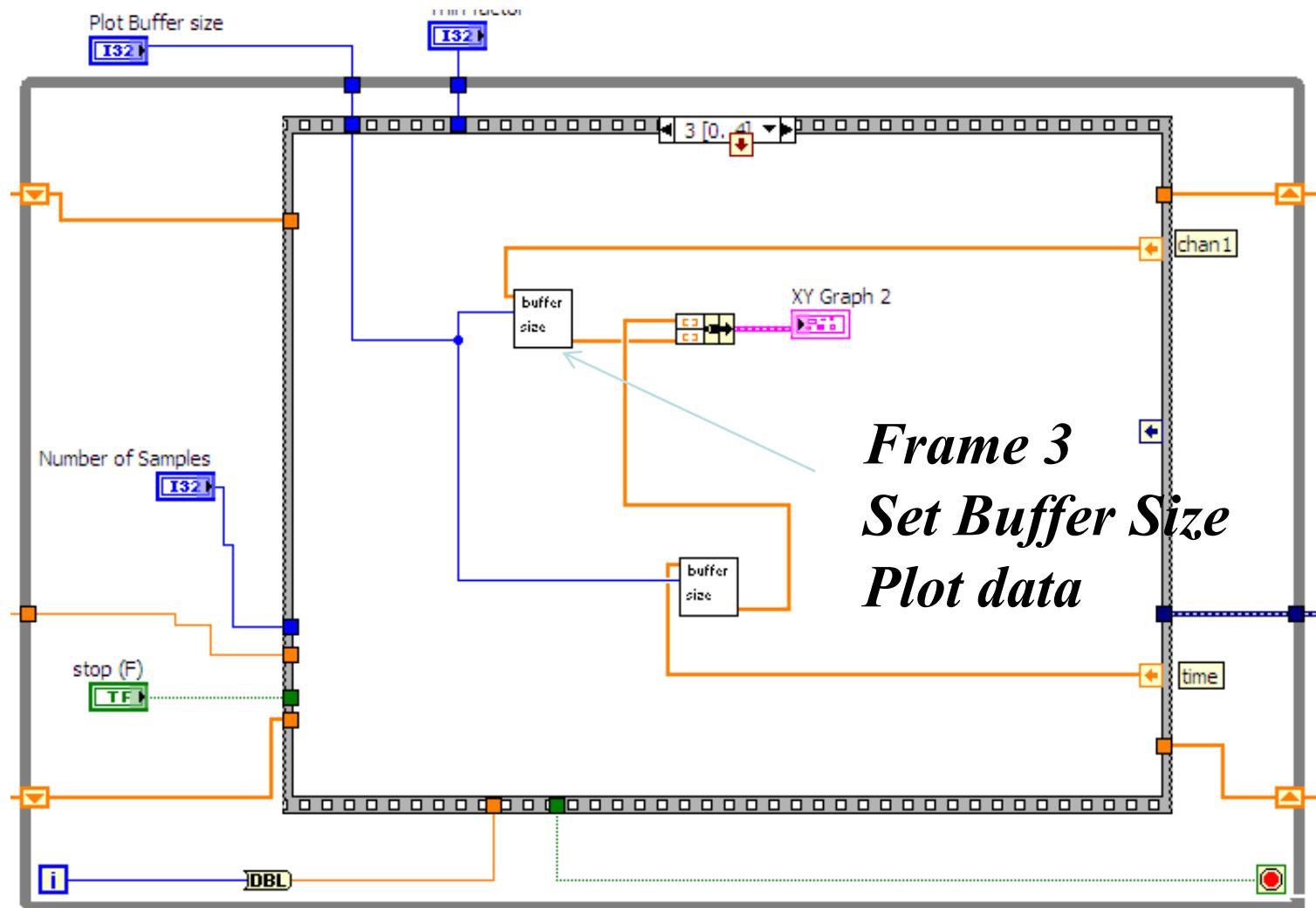
Example VI Read a Load Cells and Plots Data (2)



Example VI Read a Load Cells and Plots Data (3)



Example VI Read a Load Cells and Plots Data (4)



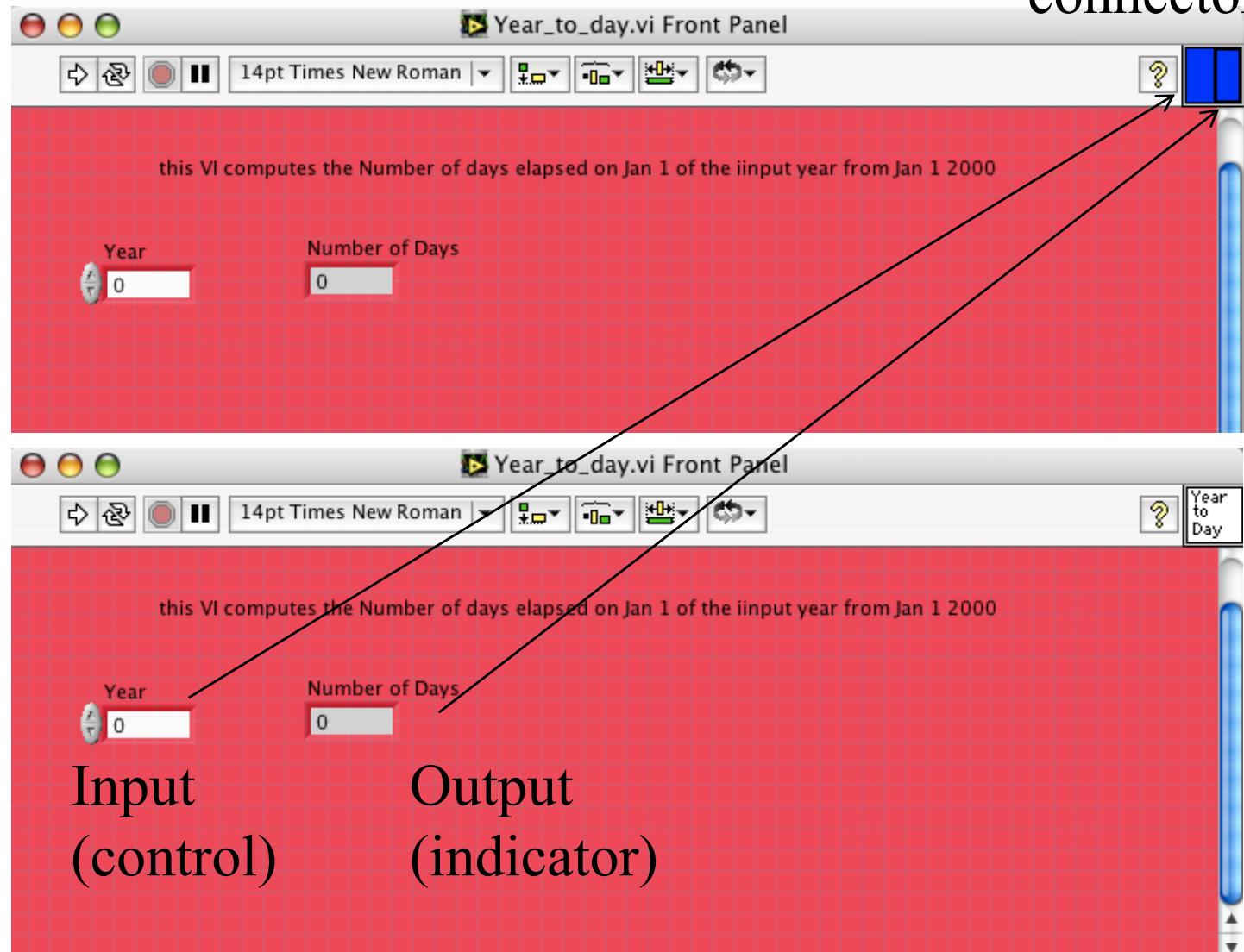
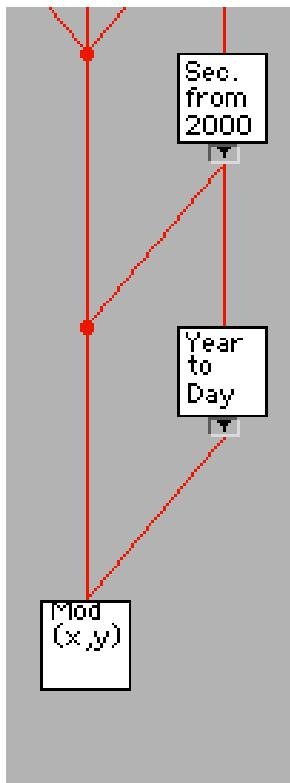
*Frame 3
Set Buffer Size
Plot data*

Labview Programming Tips

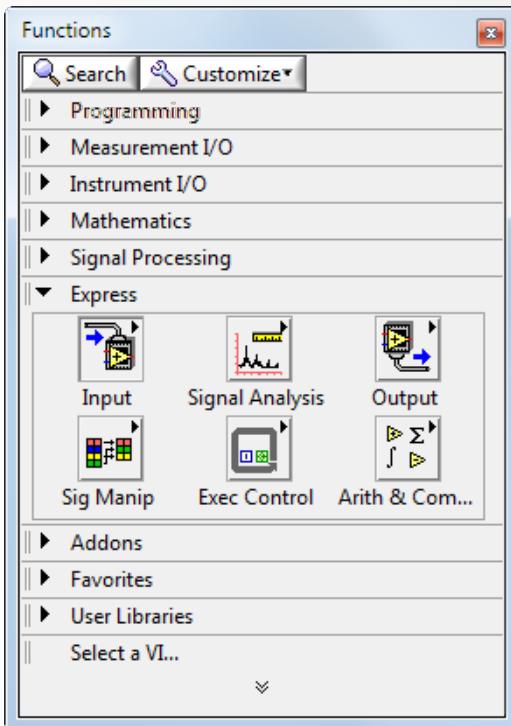
- Labview is a Object Oriented Programming Language .. Computational sequence is Data Driven
- Each Object can work by itself or as a stand alone
- Calculations are passed from one linked object To another to build *truly complex* programs

Passing data from 1 VI to Another

“connector”

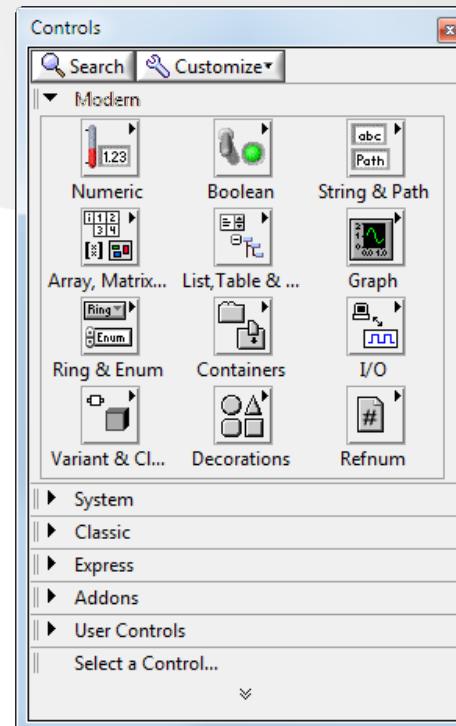


Function and Controls Palettes



Functions Palette:

Found on Block Diagram



Controls Palette:

Found on Front Panel



Tools Palette:

[View >> Tools Palette](#)

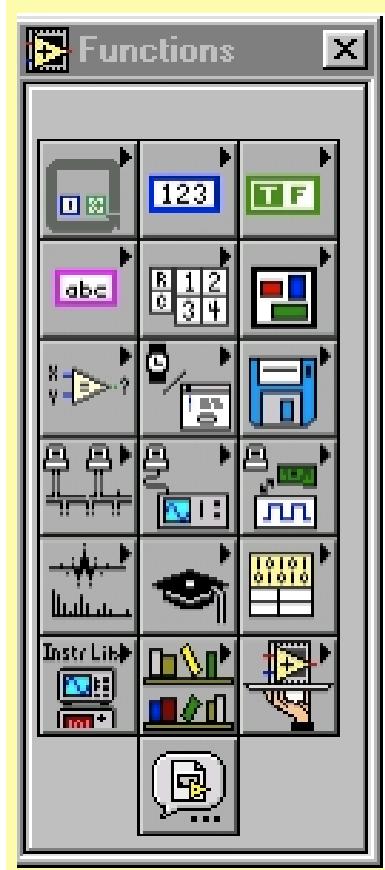


Use the “Search” buttons
At the top of the palettes
to find functions and controls

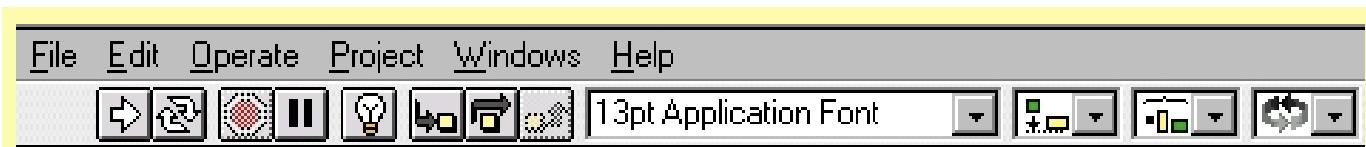
Labview for Dummies(1)

- <http://www.iit.edu/~labview/Dummies.html>

Functions Pallett



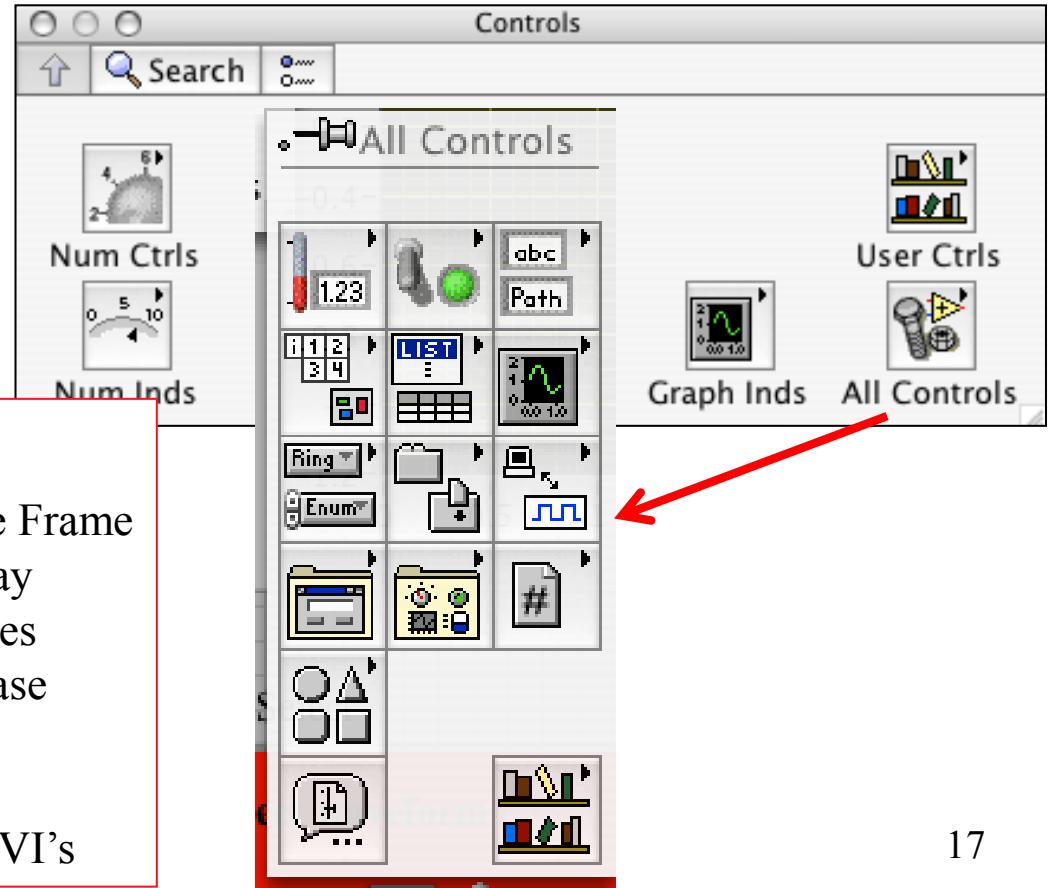
Execution bar



Tools Pallet



Controls Pallet

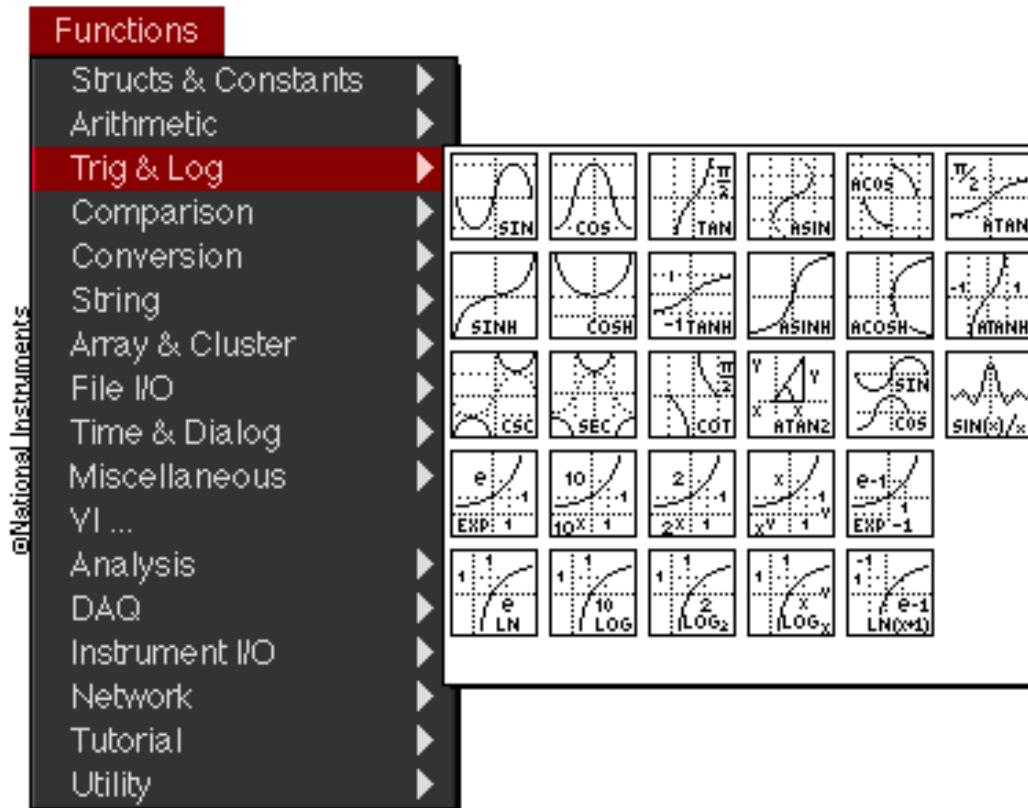


• Key skills:

- Sequence Frame
- Data Array
- Data Types
- Loops/Case
- Clusters
- Graphs
- External VI's

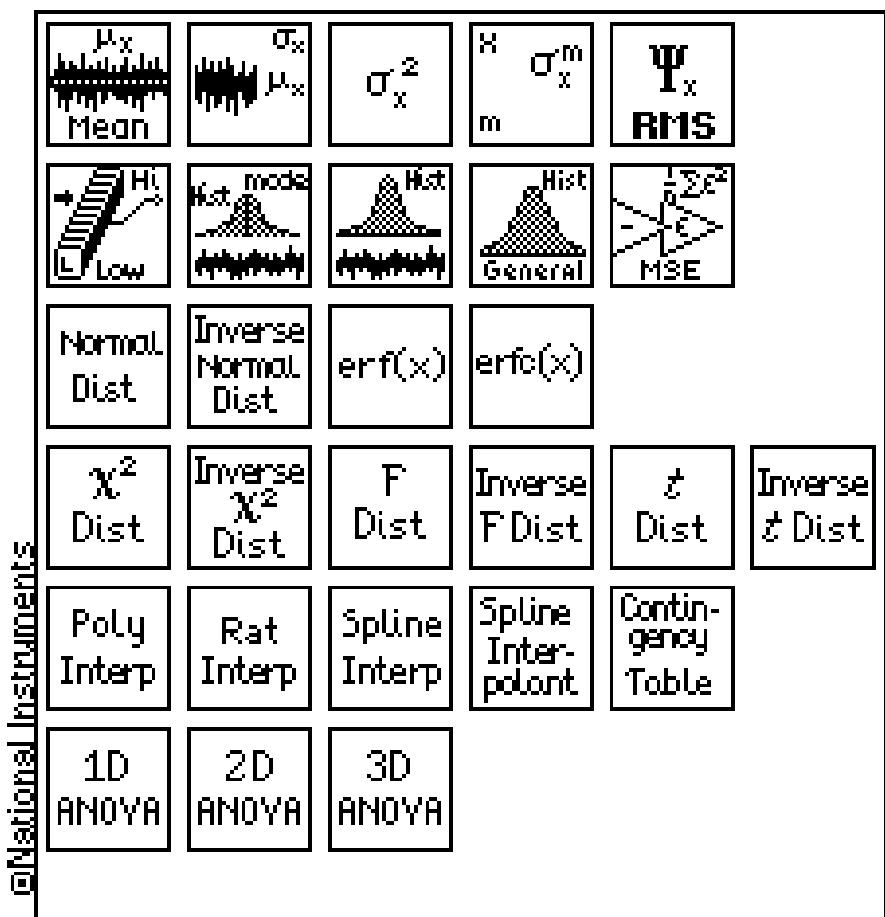
Labview Functions

- Labview contains multiple *intrinsic libraries* that aid with acquisition data display, processing, and analysis



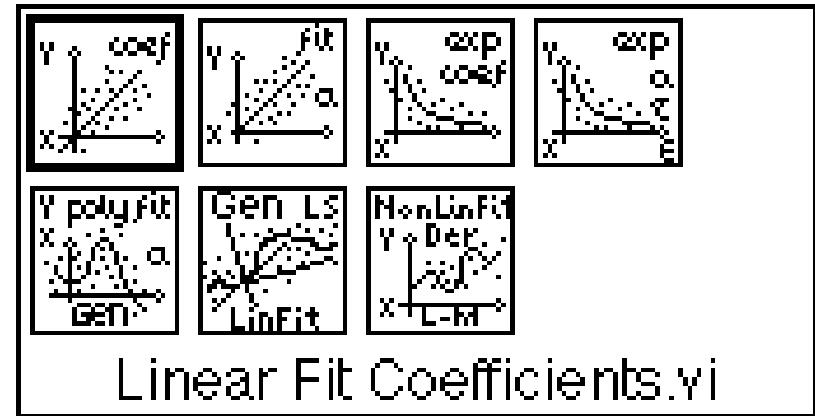
- Example: Trigonometric Functions

Labview Functions (cont'd)



- Example: Statistical Analysis Functions

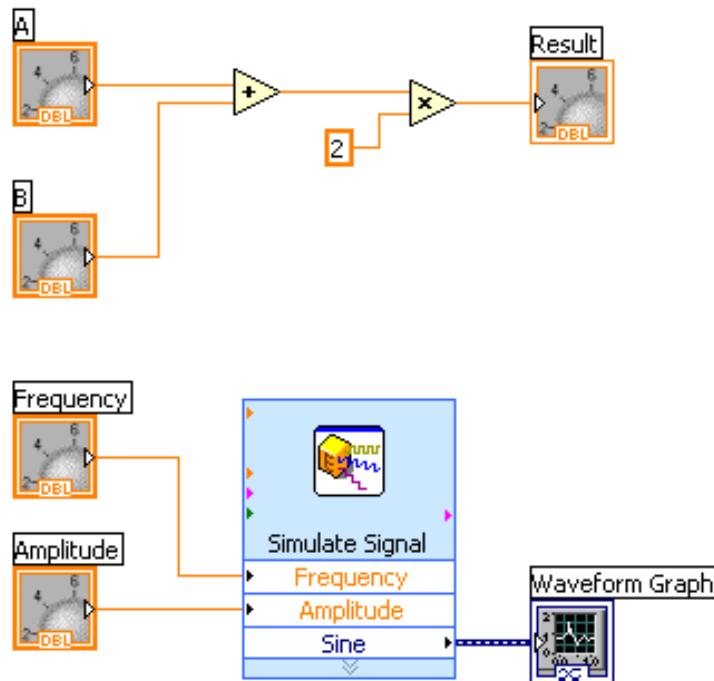
@National Instruments



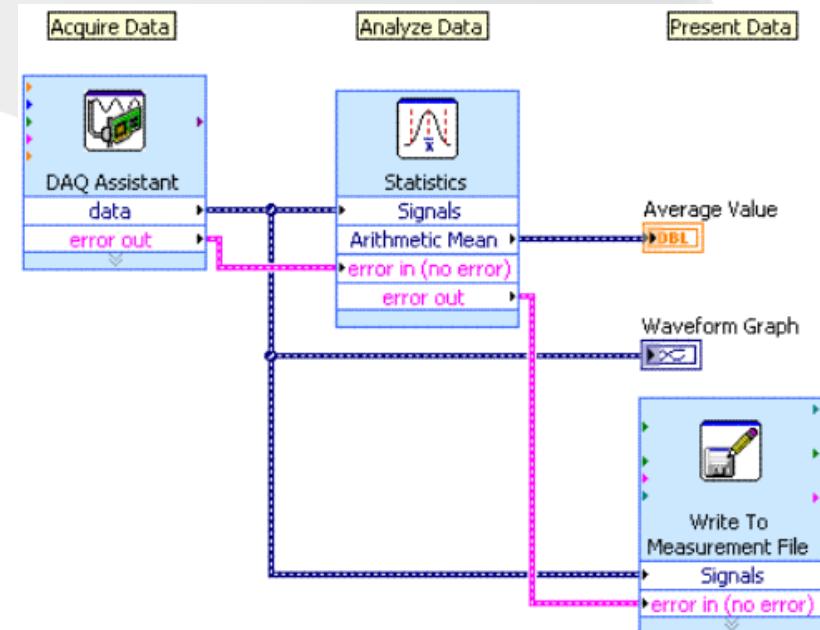
- Example: Curve Fit Functions

Dataflow Programming

Which VI(s) will execute first?

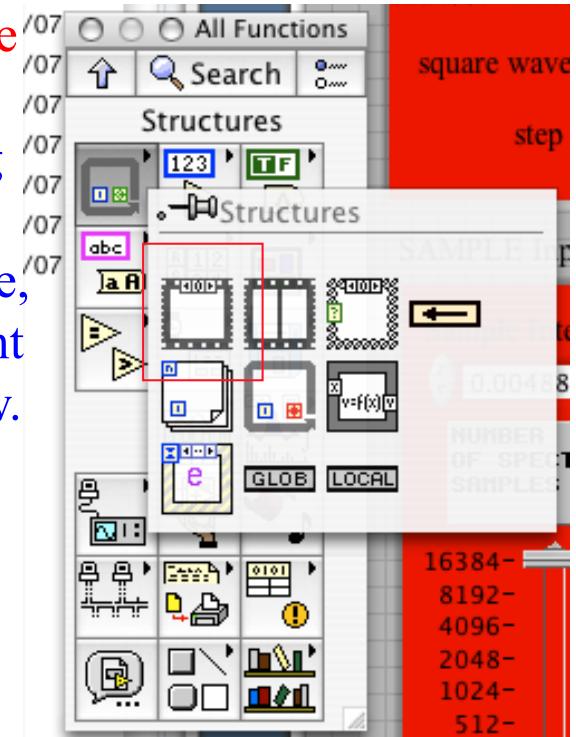


Which VI will execute last?



Sequence Frame

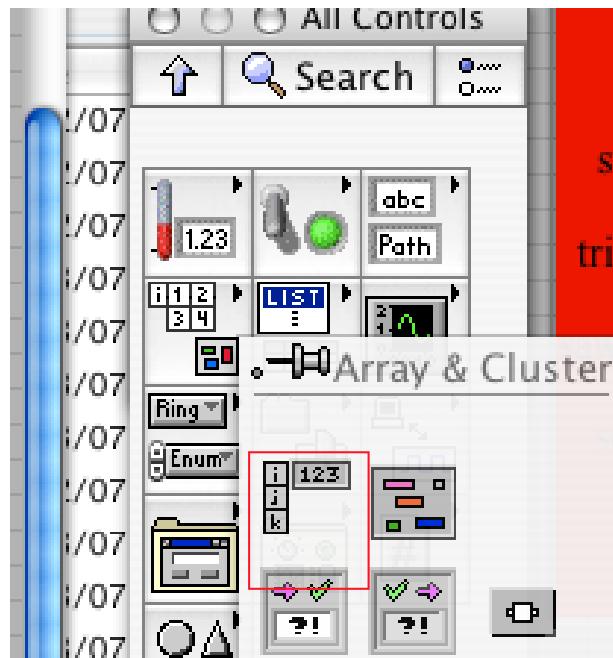
- A *Sequence Frame*, which looks like a piece of film, performs an operation after one has been completed, say after acquiring the data, analyze the data. To add a Sequence frame, right click on the frame and choose Add Frame After or Add Frame Before or Duplicate Frame. They will be used in sequential order according to the numbering scheme on the top center. Many times you will want the data from a previous frame to be used in a later sequence, Add Sequence Local, where you can attach any wire to be brought out in a later frame. An arrow will show the direction of data flow. If you move a sequence local, it will move the same way in the Other frames.
- Use of these structures is essential to keep code readable Every VI shpuld have a main sequence in which all other operations or nodes are embedded



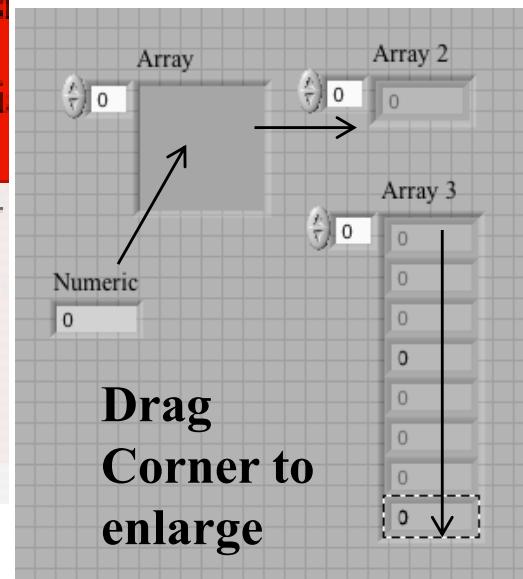
Data Array

- Arrays are invaluable. Nearly every application will need them. You can define vectors (a 1-D array) or matrices (a 2-D array) or a general N dimensional array, like tensors.

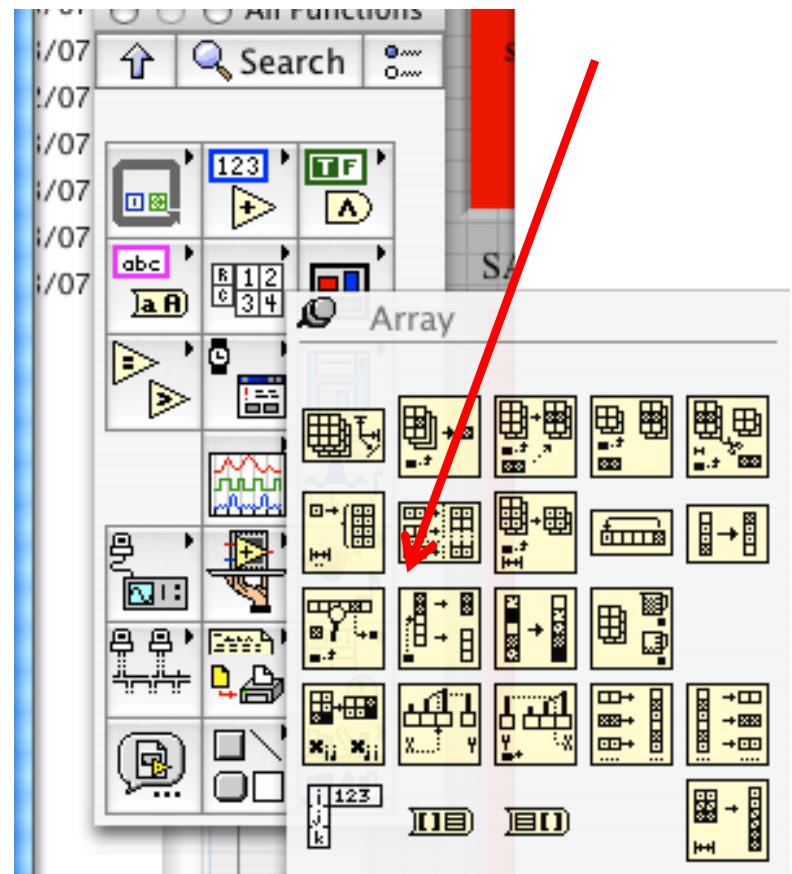
- *Build Array Control on Front Panel*



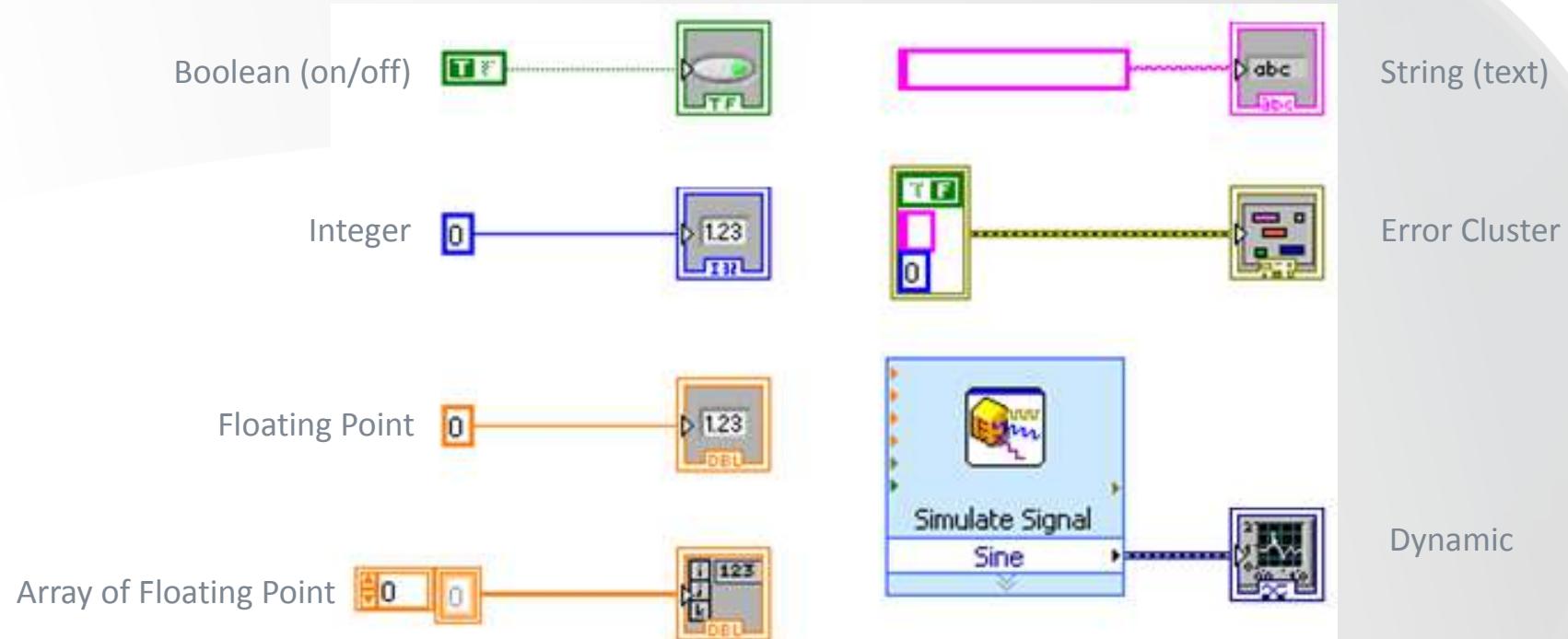
• *Drag Control / indicator to array*



- *Build Array in Block Diagram*



Data Types

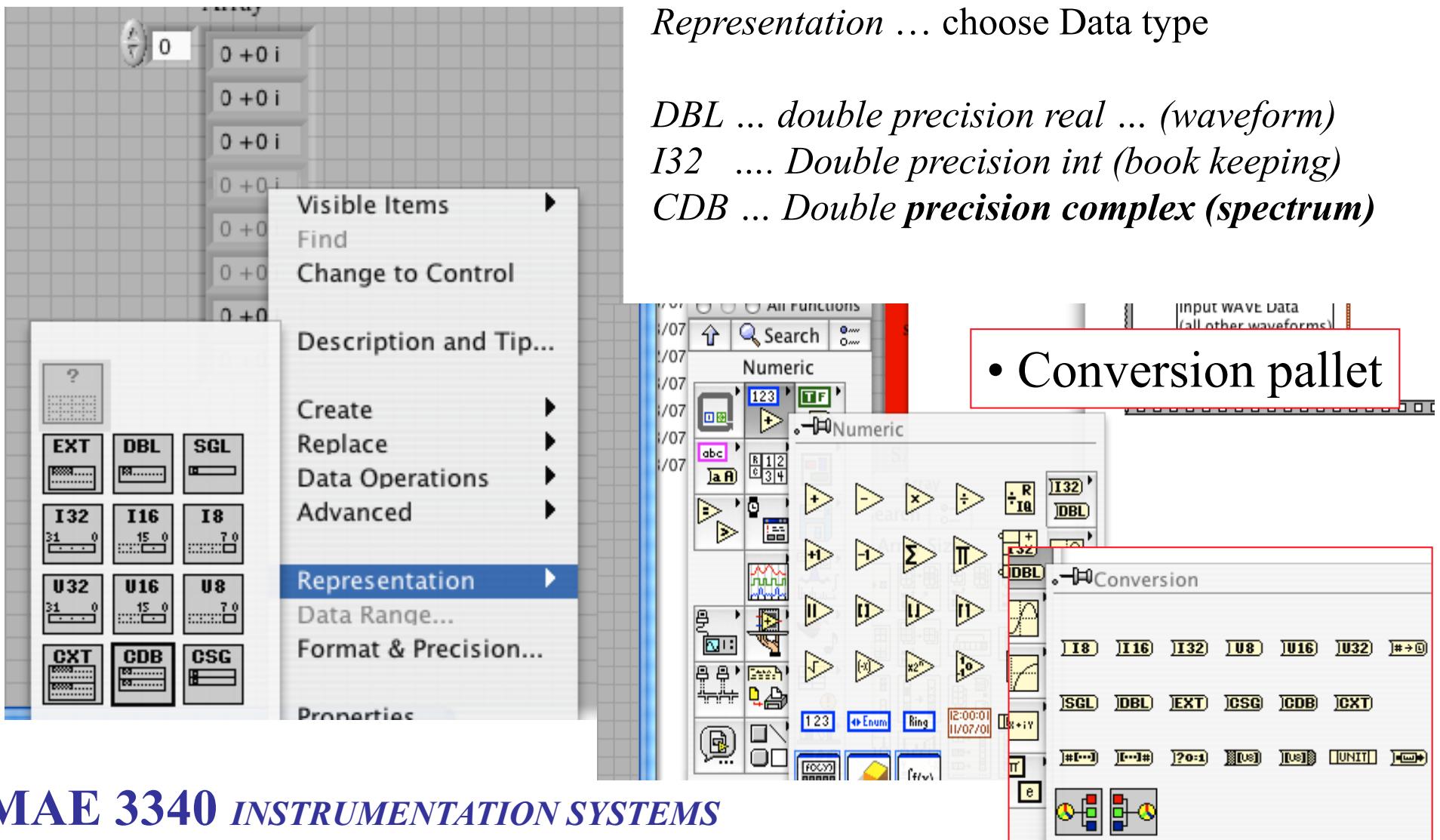


Data Types

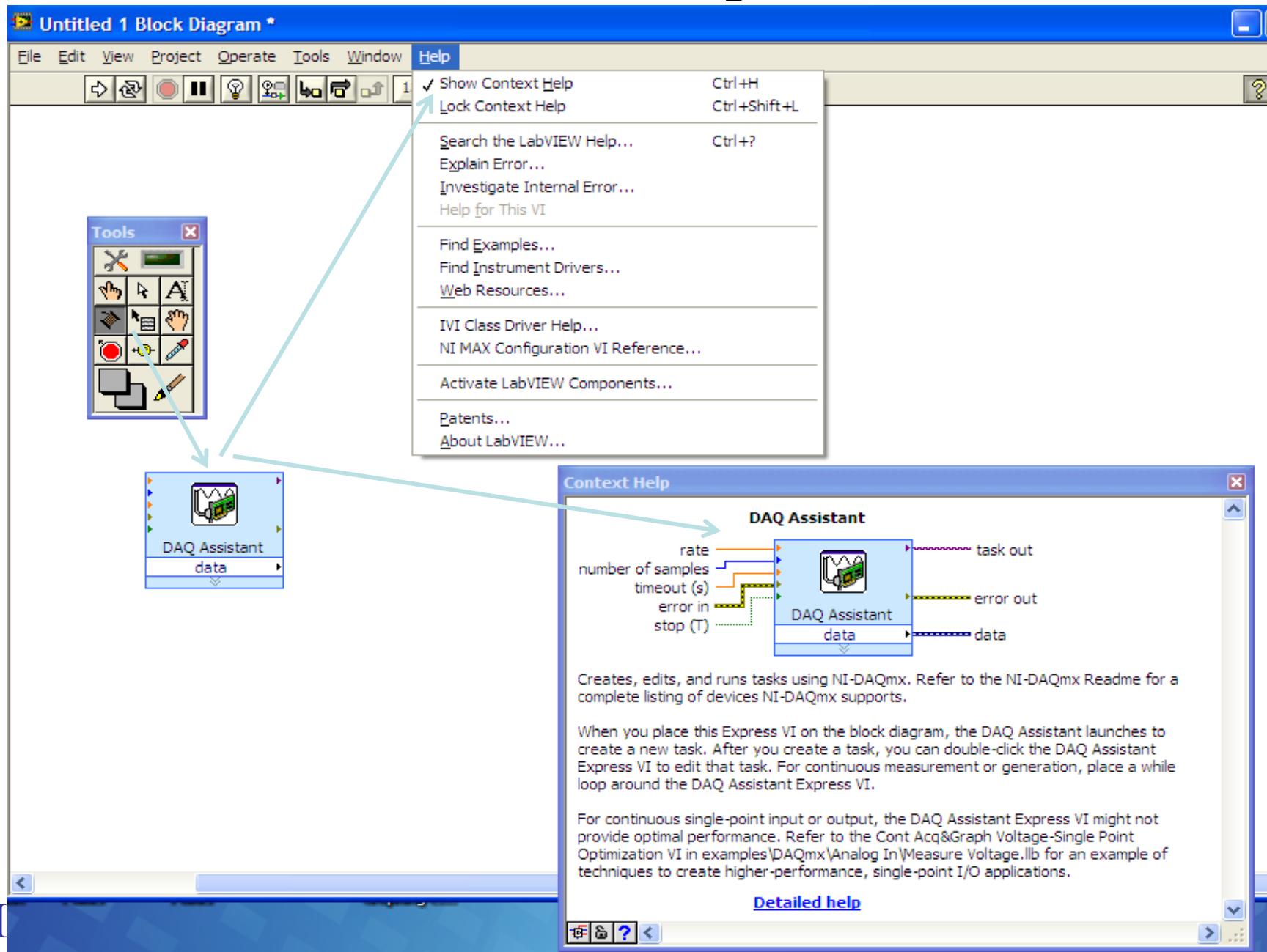
- Data Types represent how data are stored internally in the computer

- Right click on control Or indicator
Representation ... choose Data type

*DBL ... double precision real ... (waveform)
I32 ... Double precision int (book keeping)
CDB ... Double precision complex (spectrum)*



Context Help Guide



Help

Context Help Window: Ctrl-H

Extract Single Tone Information.vi

time signal in
export signals
error in (no error)
advanced search

detected frequency
detected amplitude
detected phase (deg)
error out
measurement info

Takes a signal in, finds the single tone with the highest amplitude or searches a specified frequency range, and returns the single tone frequency, amplitude, and phase. The input signal can be real or complex and single-channel or multichannel. Wire data to the **time signal in** input to determine the polymorphic instance to use or manually select the instance.

[Detailed help](#)

LabVIEW Help: Help >> LabVIEW Help

LabVIEW Help

Array Functions

Owning Palette: Programming VIs and Functions

Requires: Base Development System. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

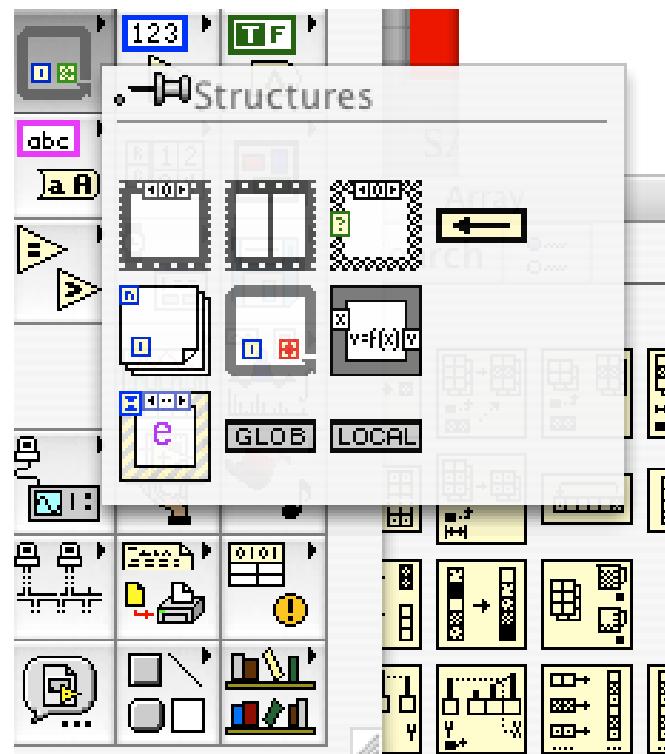
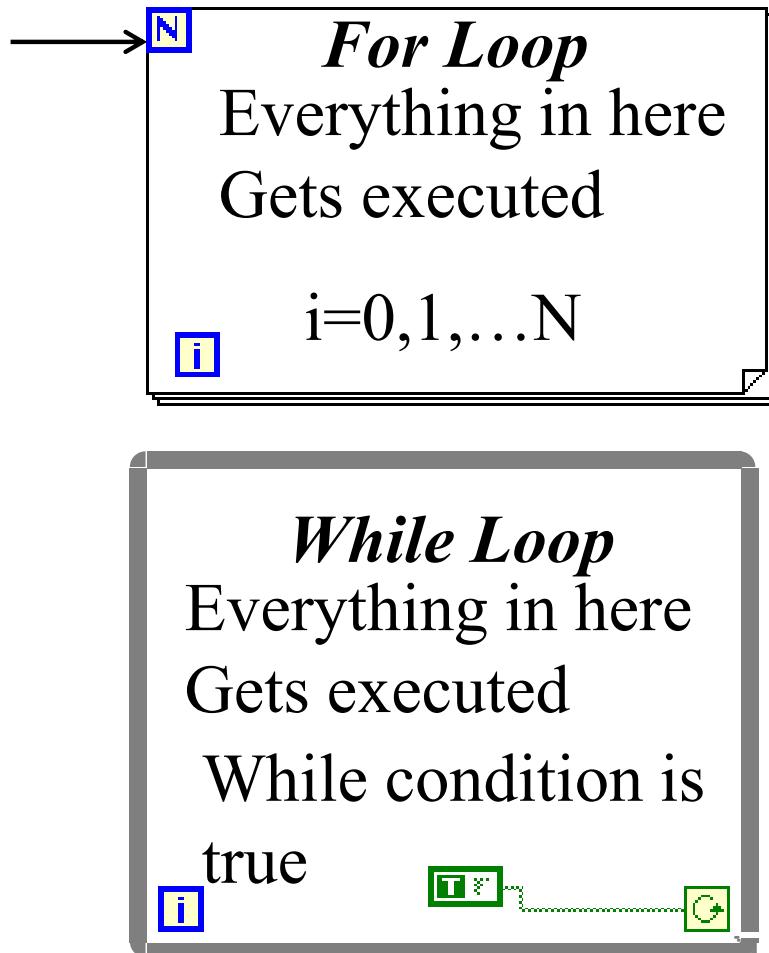
Use the **Array** functions to create and manipulate arrays.

Example

Palette Object	Description
Array Constant	Use this constant to supply a constant array value to the block diagram.
Array Max & Min	Returns the maximum and minimum values found in array , along with the indexes for each value.
Array Size	Returns the number of elements in each dimension of array .
Array Subset	Returns a portion of array starting at index and containing length elements.
Array To Cluster	Converts a 1D array to a cluster of elements of the same type as the array elements. Right-click the function and select Cluster Size from the shortcut menu to set the number of elements in the cluster.
Array To Matrix	Converts an array to a matrix of elements of the same type as the array elements. Wire data to the Real 2D Array input to determine the polymorphic instance to use or manually select the instance.
Build Array	Concatenates multiple arrays or appends elements to an n-dimensional array .
Cluster To Array	Converts a cluster of elements of the same data type to a 1D array of elements of the same data type.
Decimate	Divides the elements of array into the output arrays .

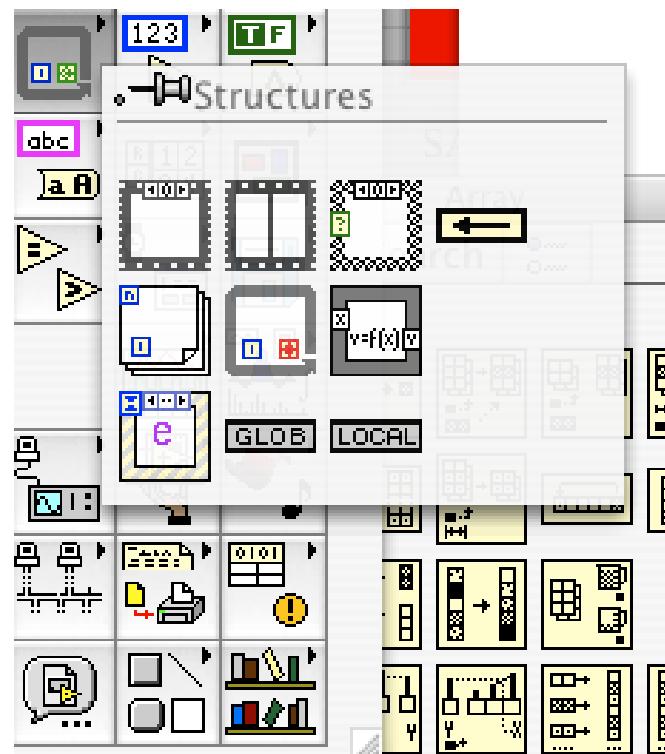
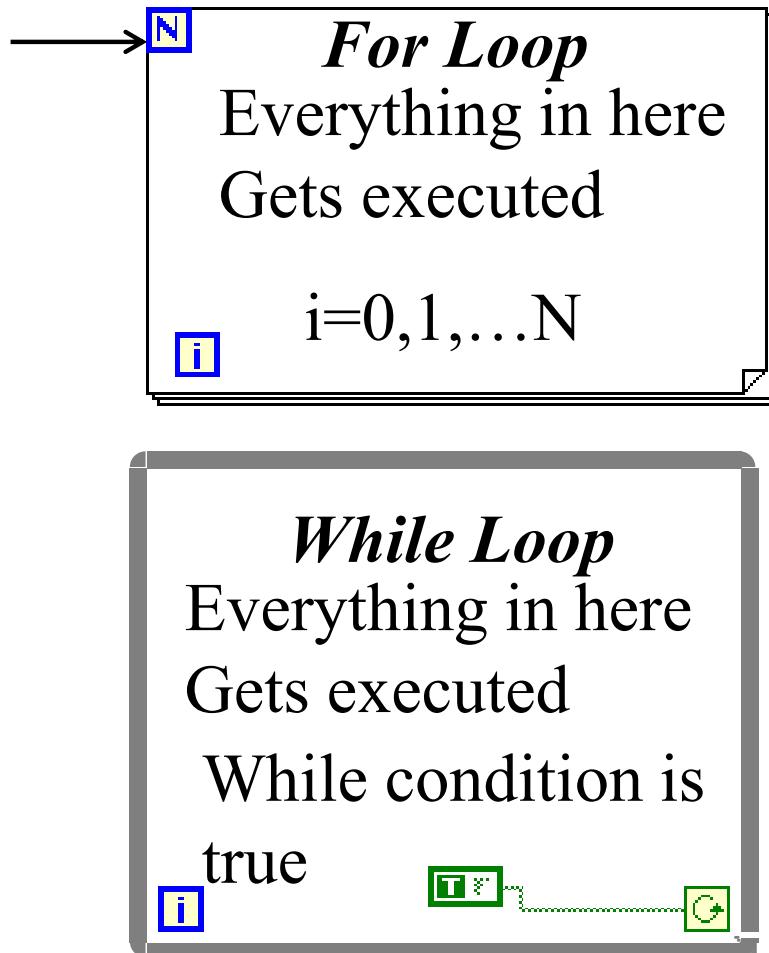
Loops/Cases (1)

- Loops *for, while ...* similar to loops in FORTRAN and C++



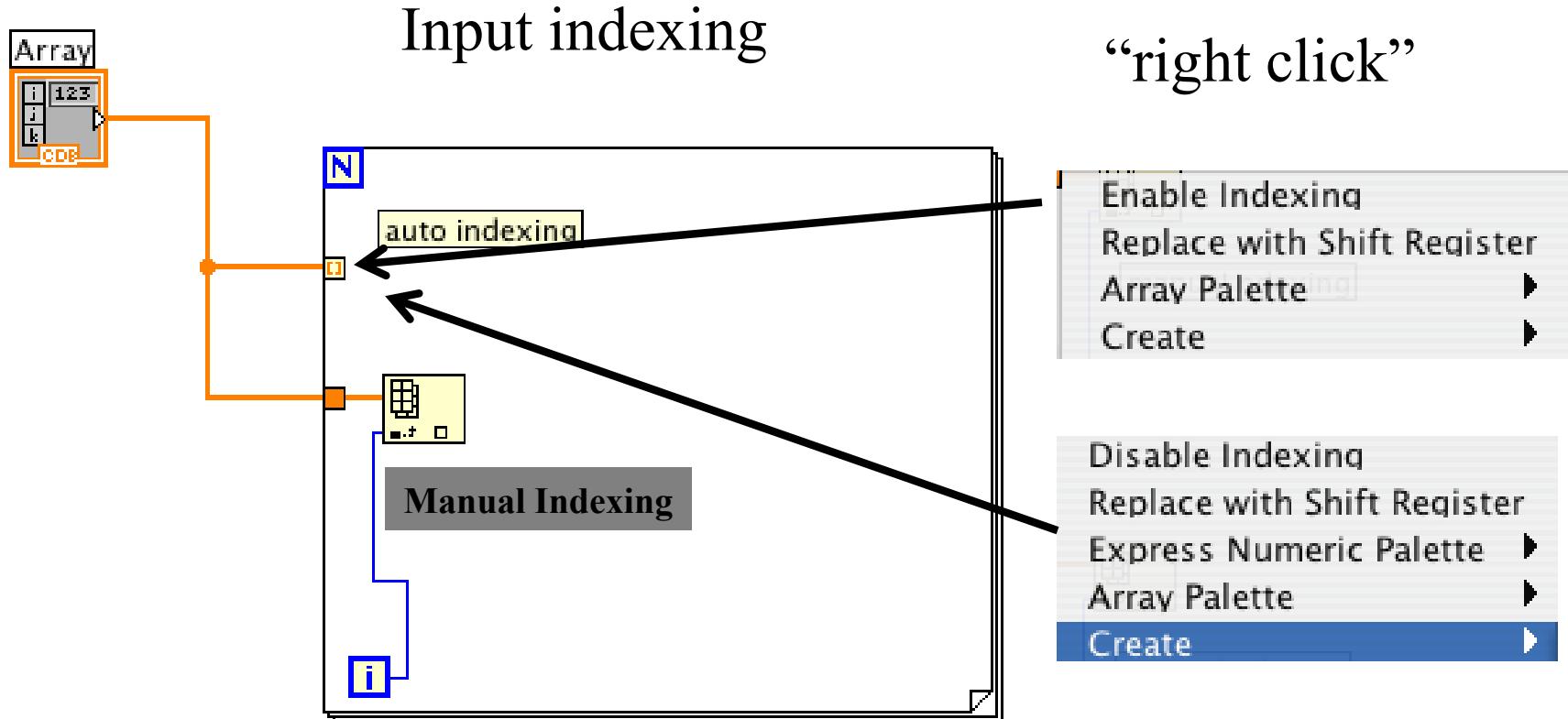
Loops/Cases (1)

- Loops *for, while ...* similar to loops in FORTRAN and C++



Loops/Cases (Input Indexing)

- Auto/manual indexing ... within *for/while* loop

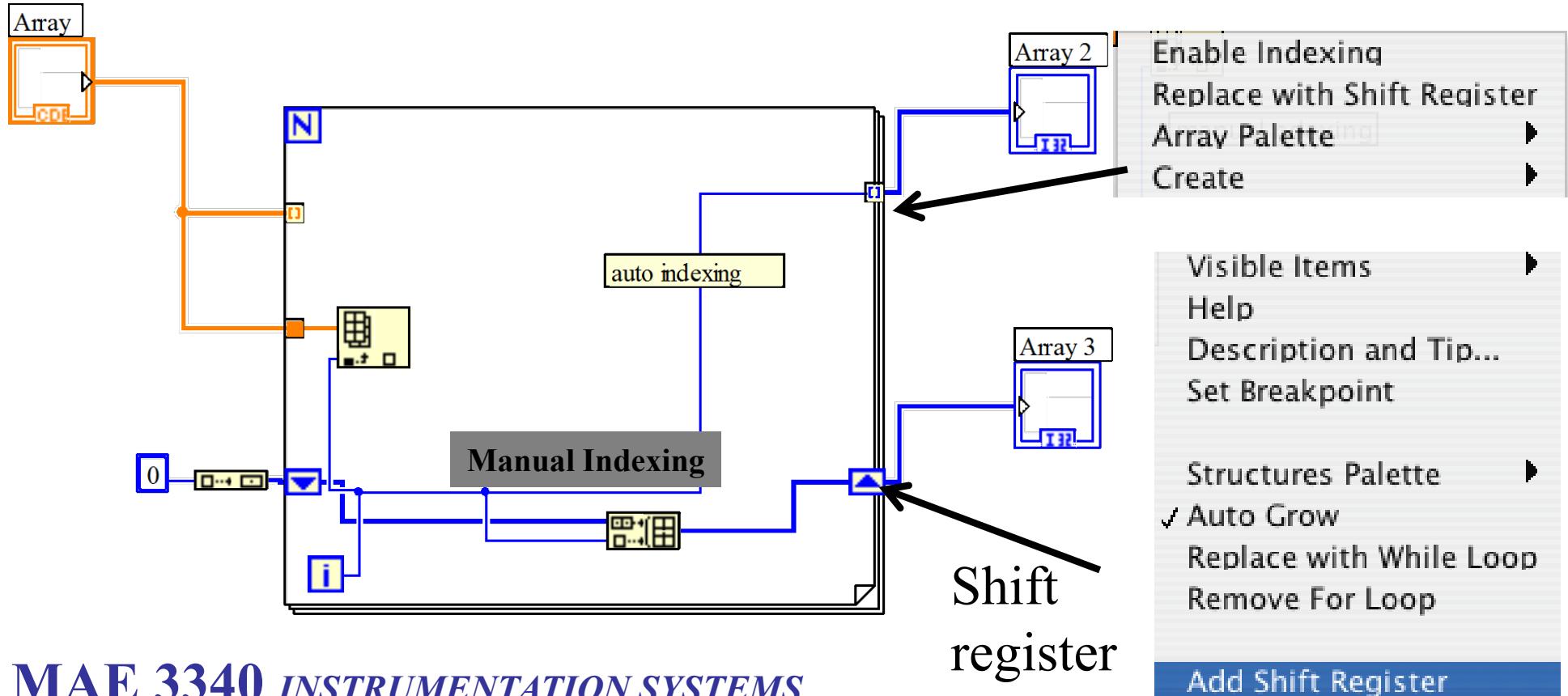


Loops/Cases (Output Indexing)

- Auto/manual indexing ... within *for/while* loop

Output indexing

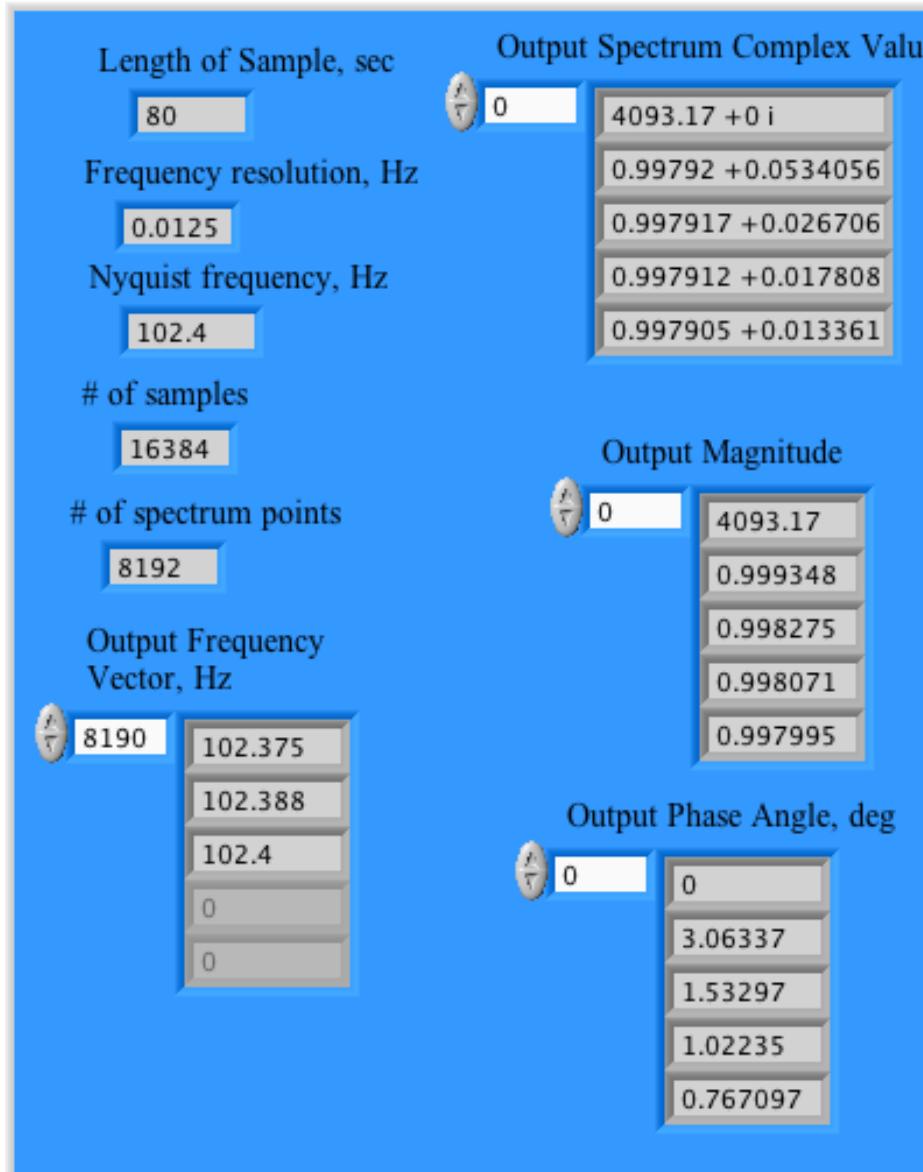
“right click”



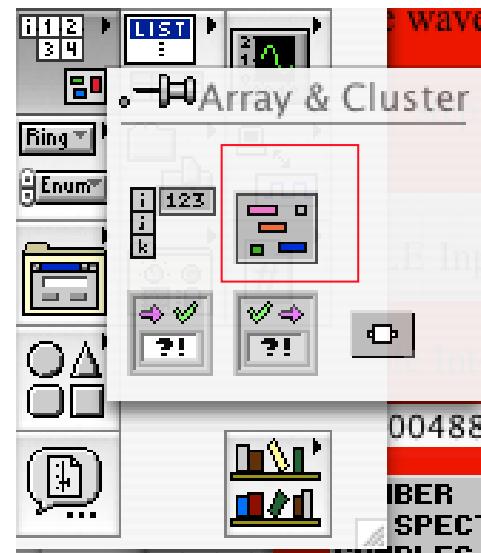
Clusters (1)

- Allow different data types to be grouped together

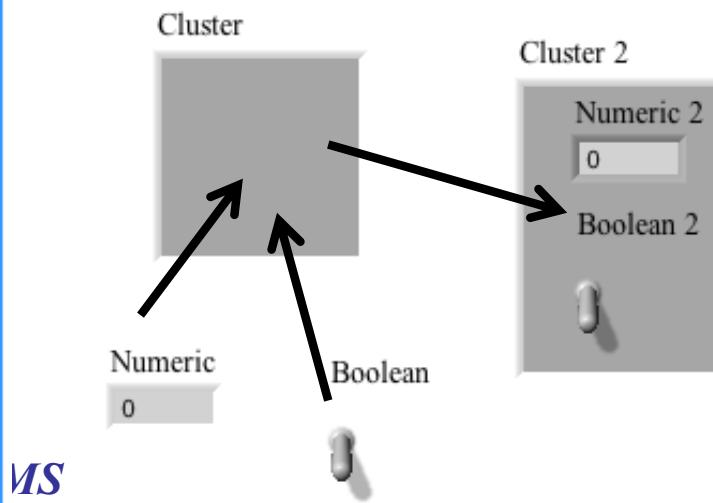
Wave Spectrum data



Controls pallet (control panel)



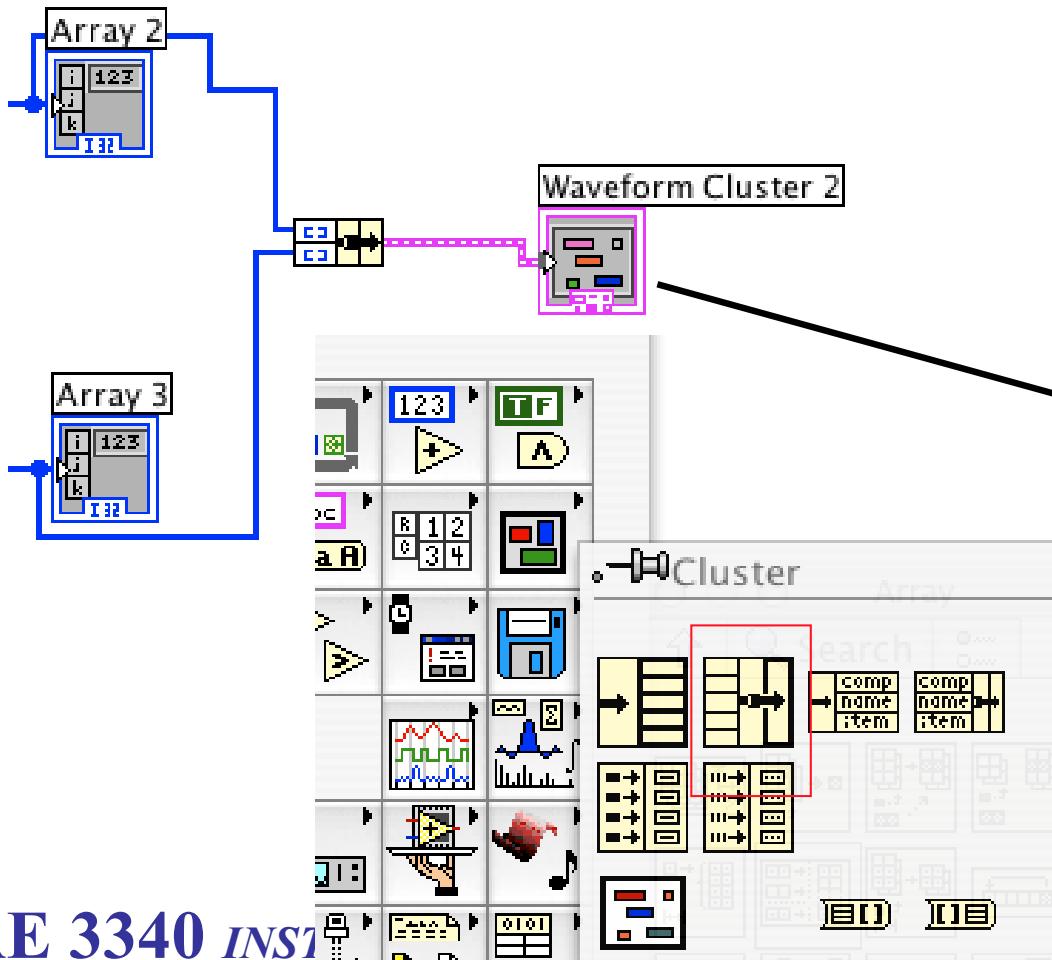
- Drag build
Similar to
Array control



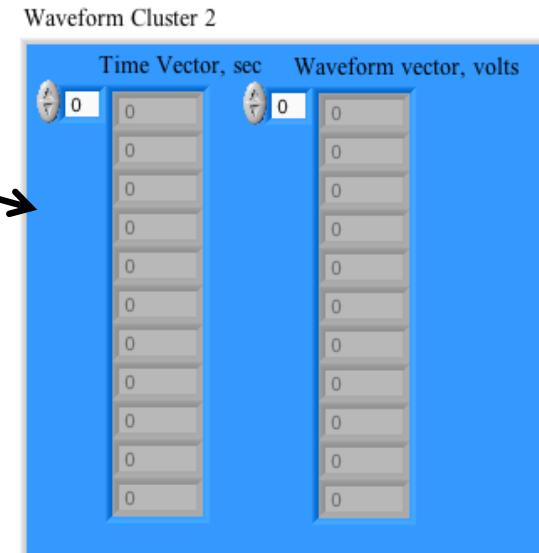
Clusters (2)

- Inserting Elements in a Cluster

Functions Pallet (block diagram)

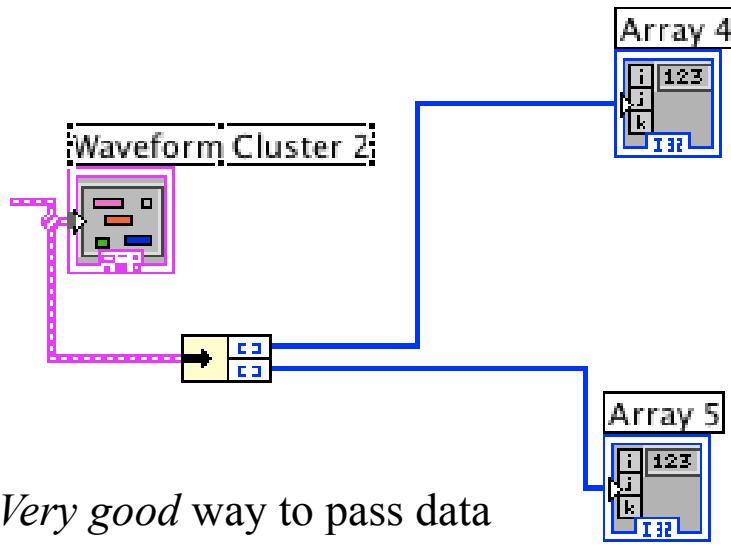


On Control panel

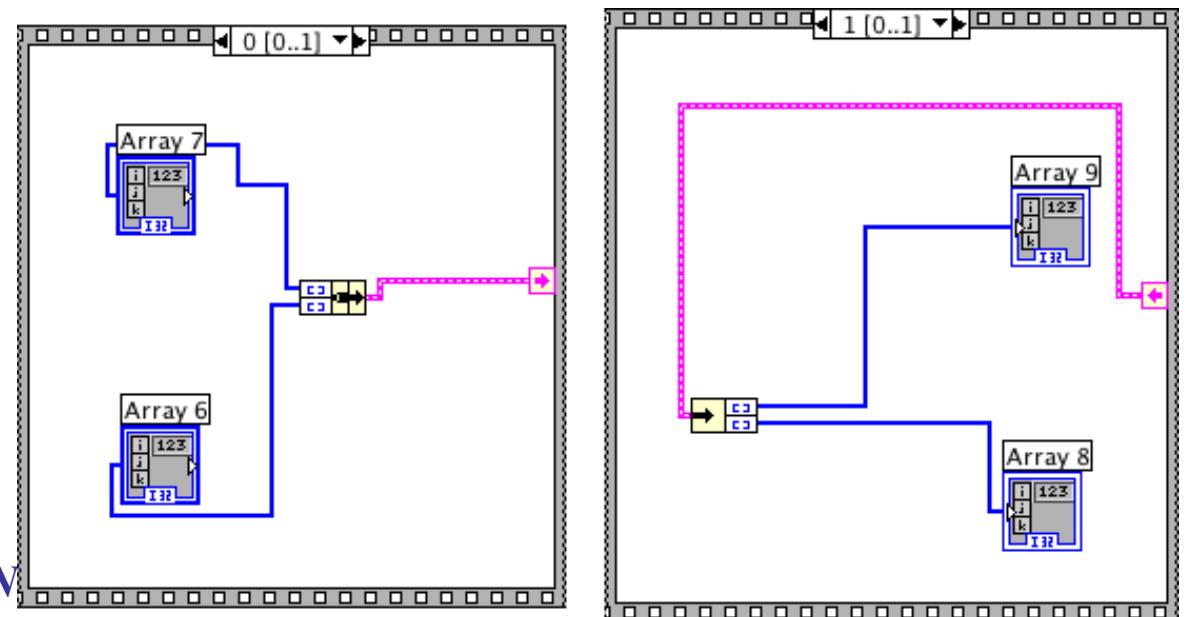
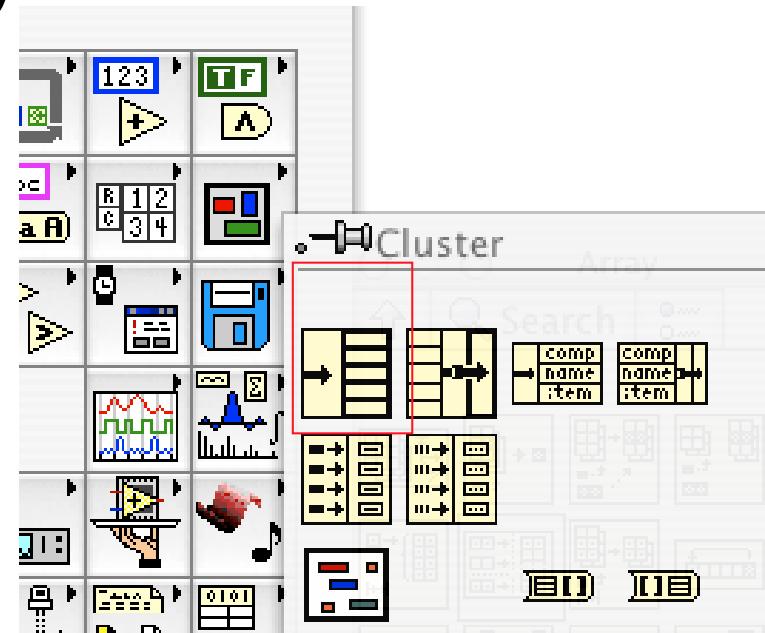


Clusters (3)

- Extracting Elements from a Cluster
(diagram)



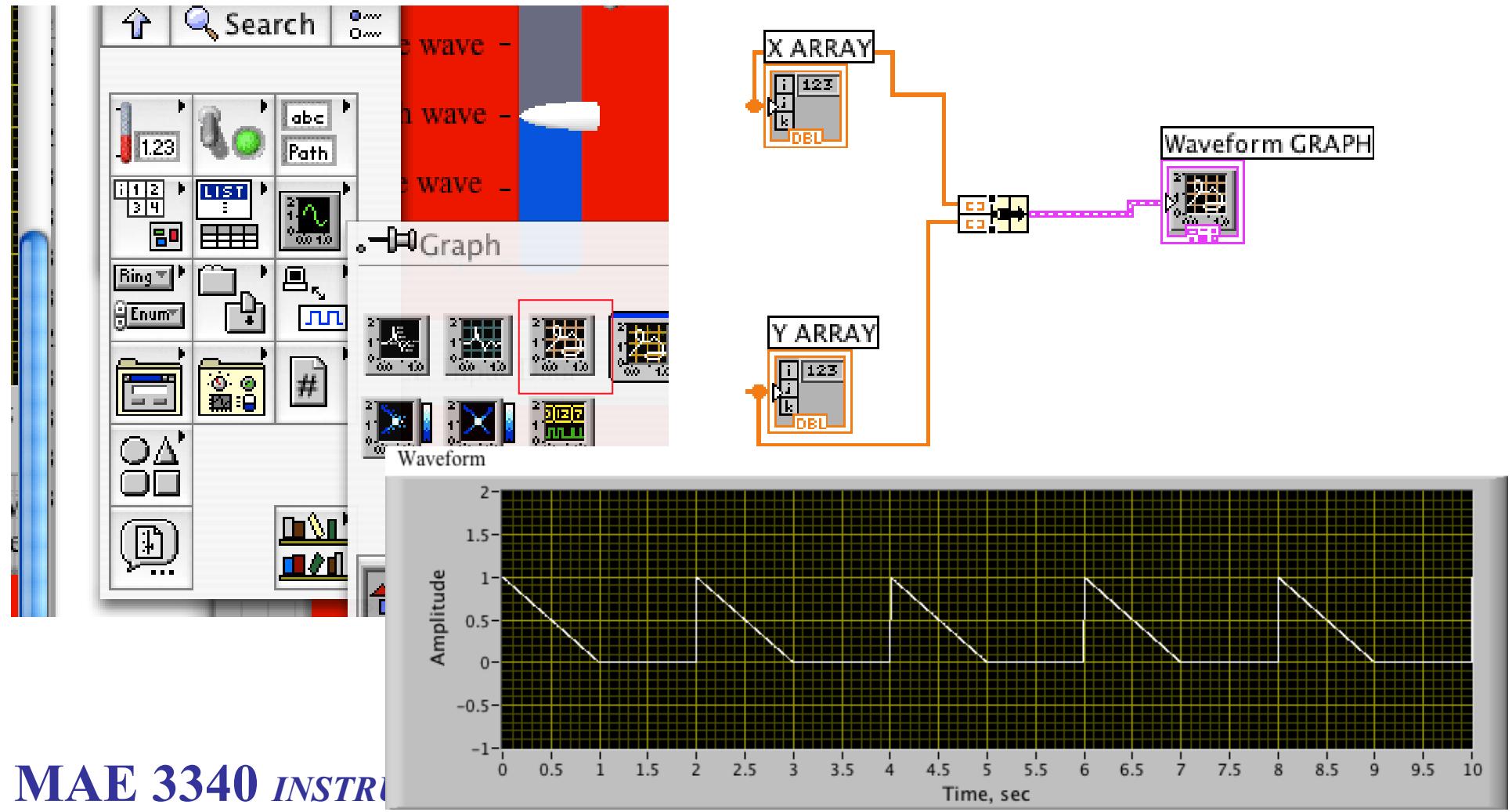
*Very good way to pass data
from one frame to another
... makes for more readable code*



Graphs (1)

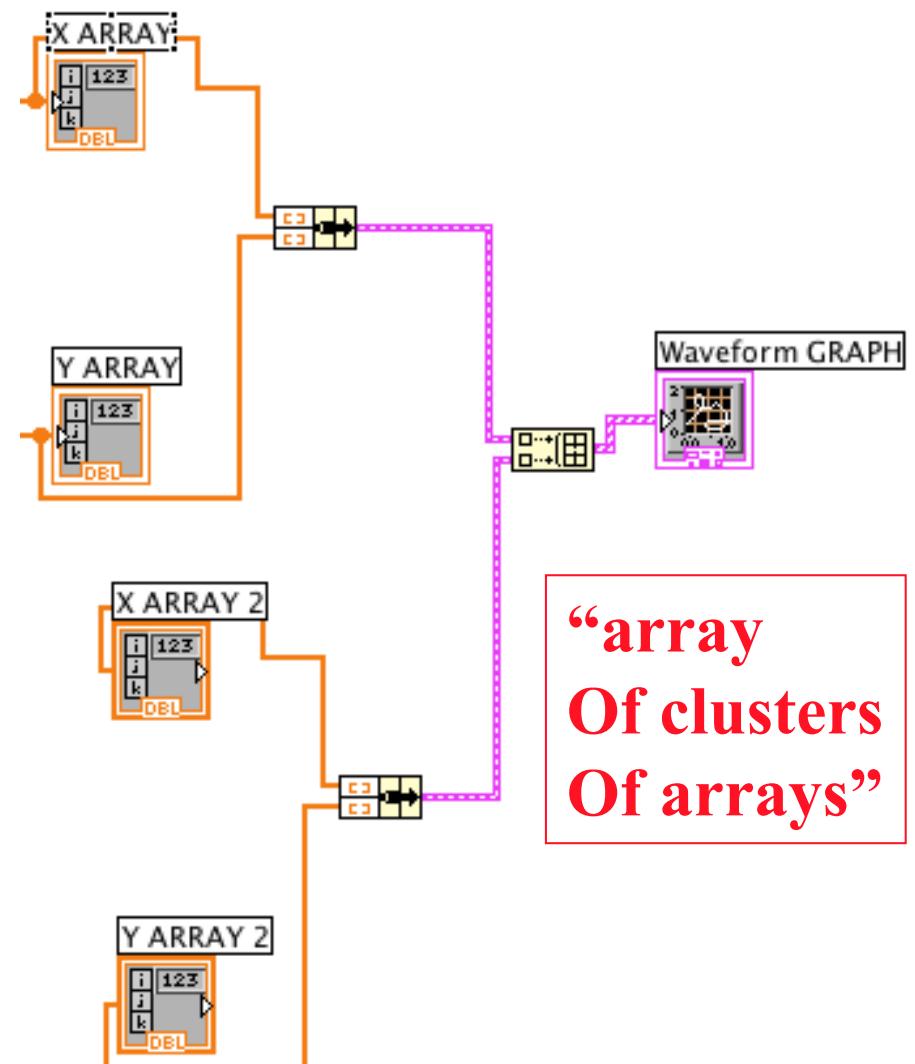
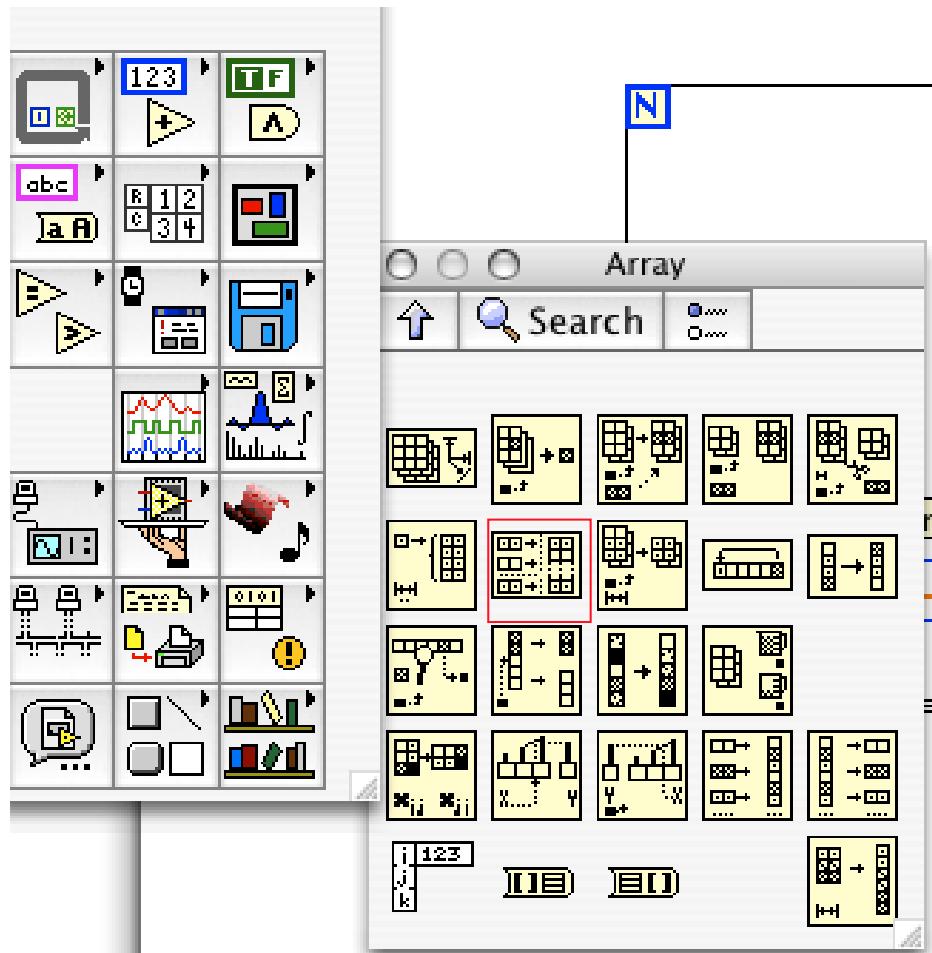
- Most Valuable is X/Y Plot

Plots X and Y vectors
against each other



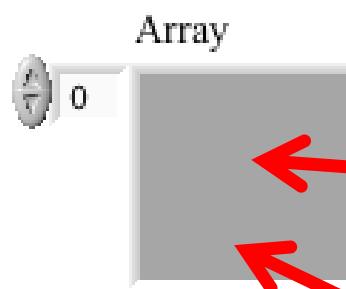
Graphs (2)

- Multiple Plots on One graph

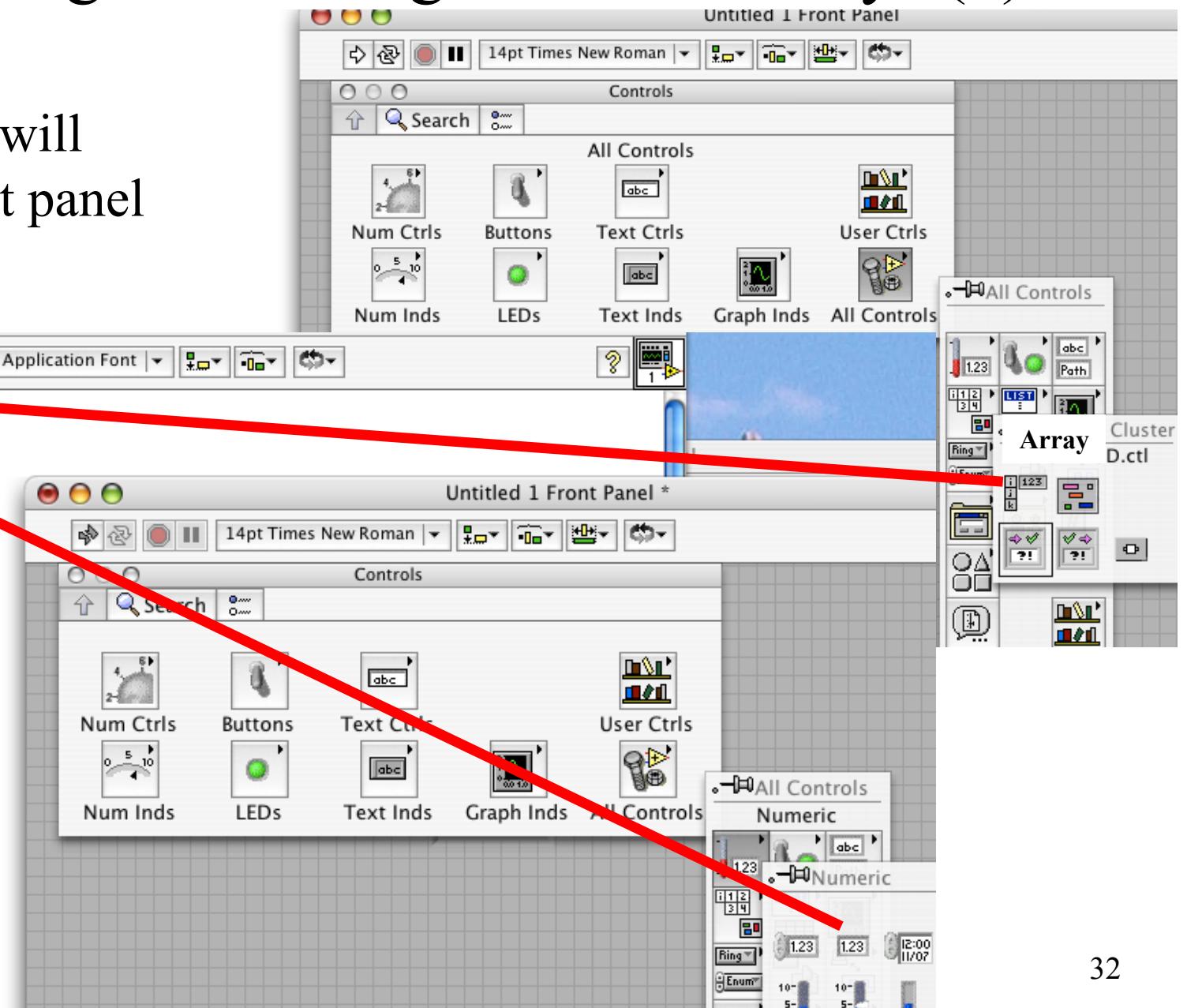


Plotting Data Using Labview Arrays (1)

- Empty array will appear on front panel



- Drag real indicator into array

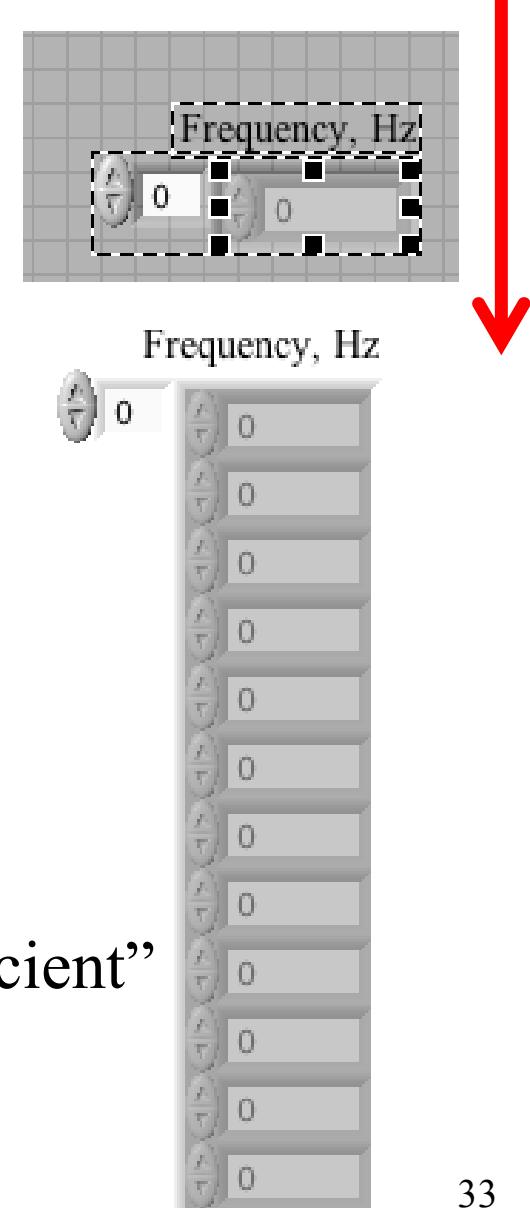


Plotting Data Using Labview Arrays (2)

- Rename Array as “Frequency, Hz”



- Drag down on array
To show multiple
elements

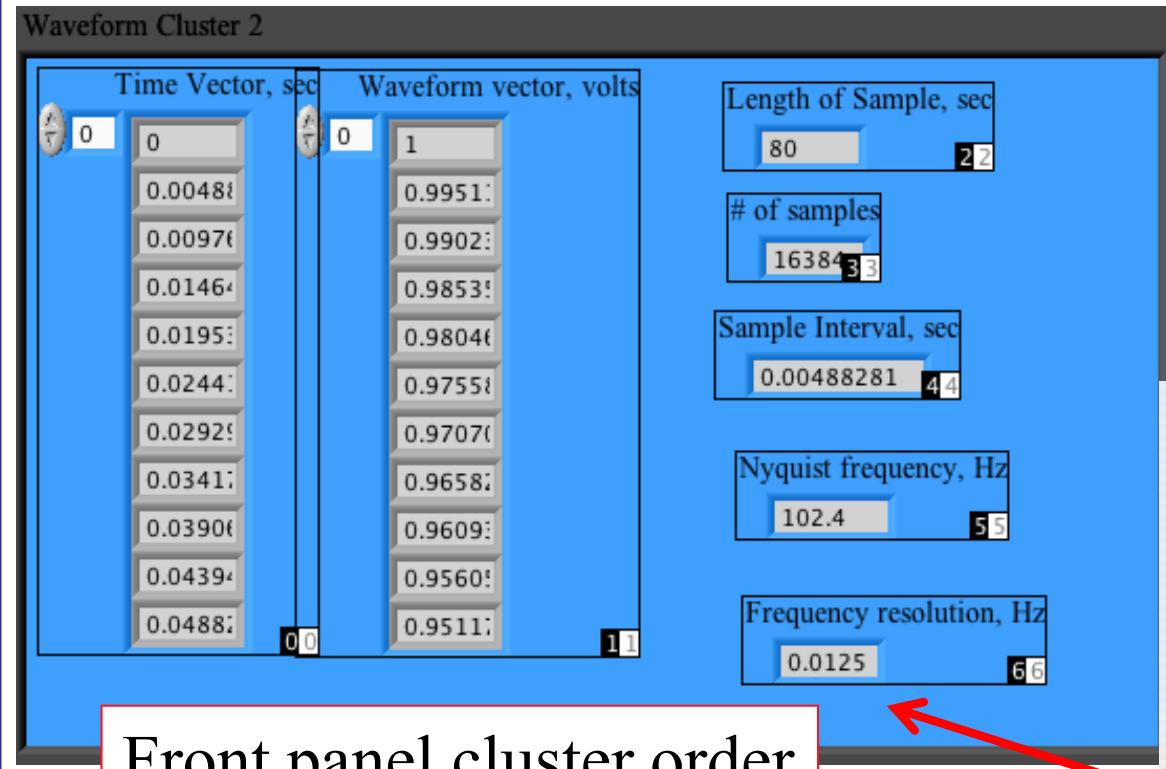


- Repeat process for arrays called
“Fourier A Coefficient”, “Fourier B coefficient”
“magnitude”, and “Phase”

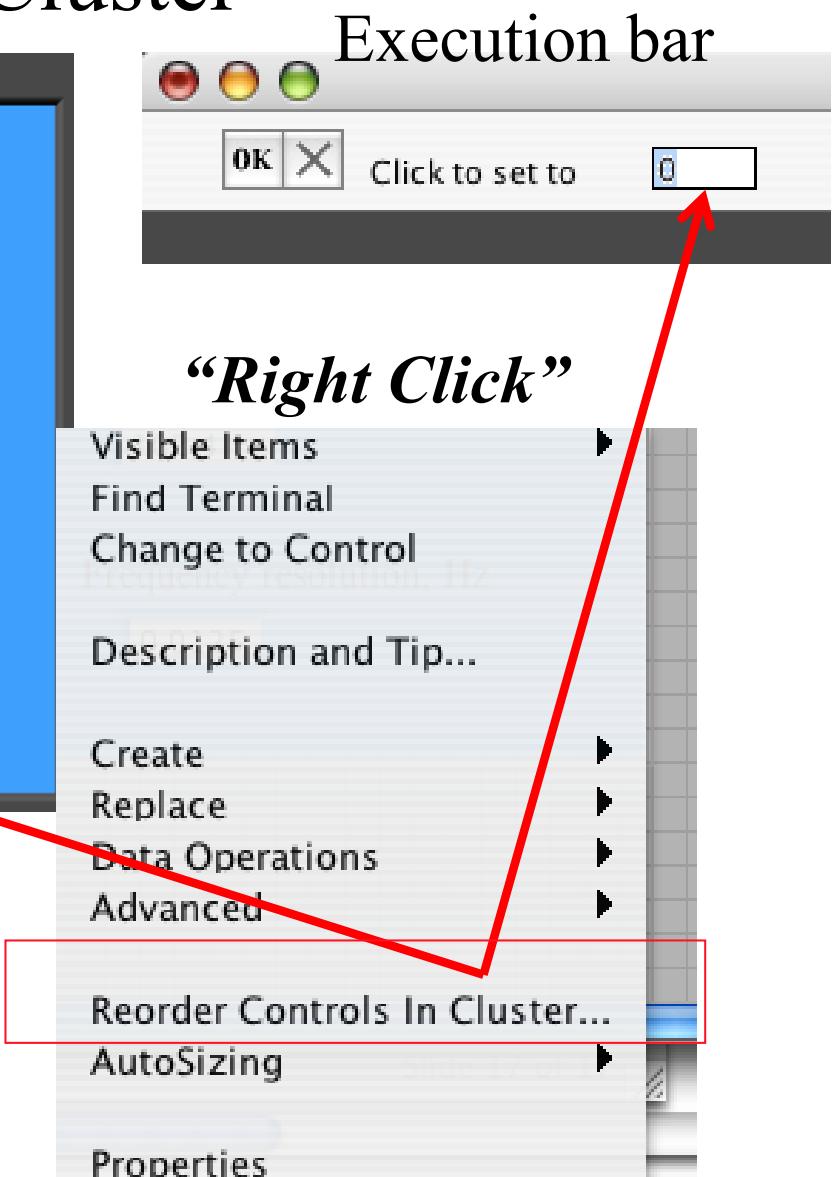
Arrays as they appear on front panel

Output Frequency Vector, Hz	A(ω_{nt})	B(ω_{nt})
39	0.48753	-1.73435E-7
	0.500031	0.318323
	0.512532	-1.52712E-7
	0.525032	-1.52193E-7
	0.537533	-1.4973E-7
	0.550034	-1.46905E-7
	0.562535	-1.44028E-7
	0.575035	-1.41196E-7
Magnitude	Phase Angle, deg.	
39	0.000117816	0.000117816
	0.202276	0.202276
	0.000126117	0.000126117
	0.000124043	0.000124043
	0.000123351	0.000123351
	0.000123006	0.000123006
	0.000122799	0.000122799
	0.000122661	0.000122661

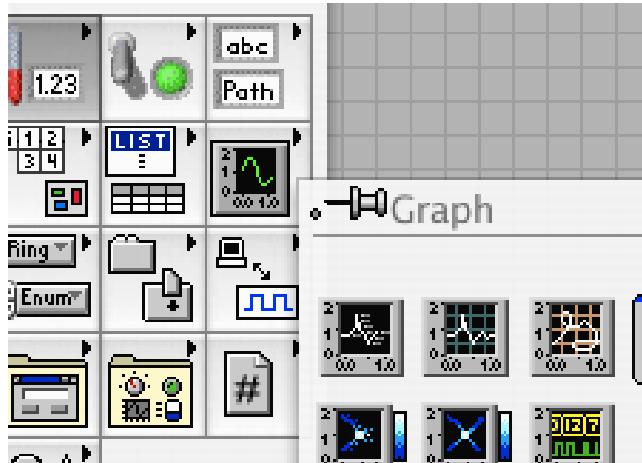
Front Panel Cluster



Front panel cluster order

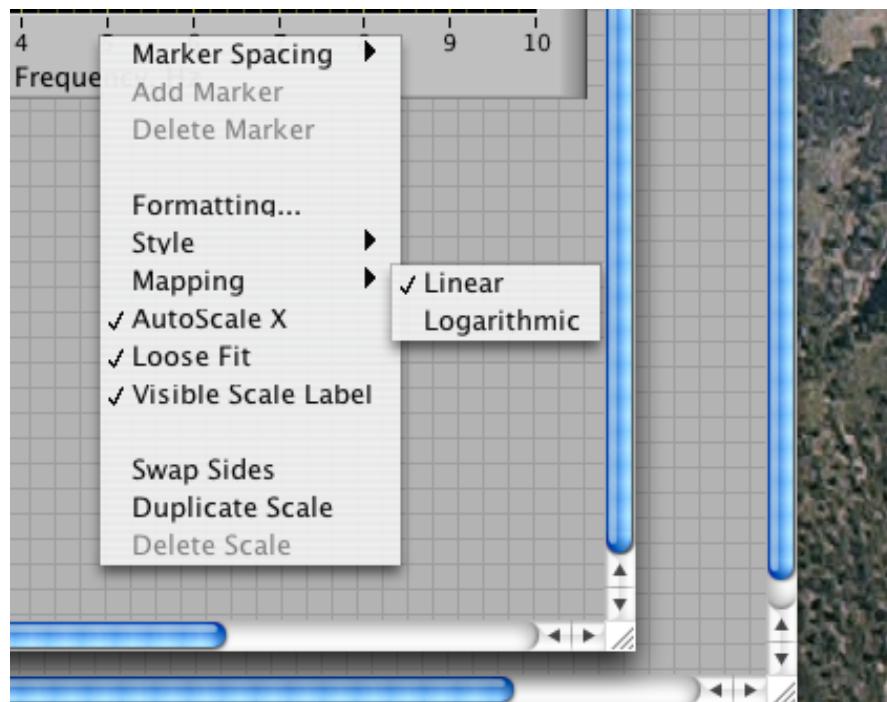


Plotting Data Using Labview Arrays (3)



Drag an X/Y
graph to your font panel

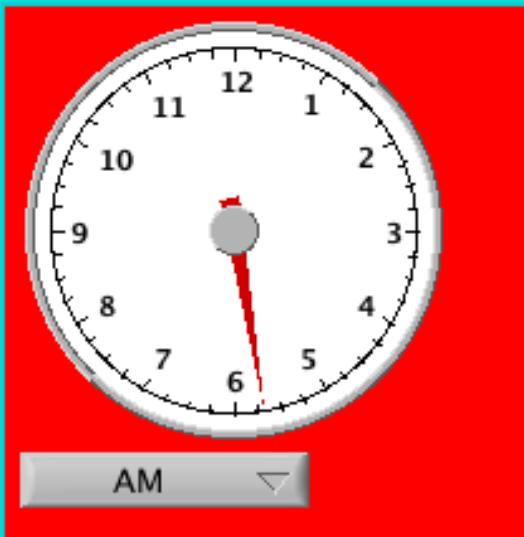
Label it:
Magnitude response
Fourier Coefficients,
X Axis:frequency, Hz
Y axis: A, B Coefficient



- Right click on X-axis .. Select Mapping-> logarithmic
- Same with Y- axis

Example: My own alarm clock (1)

Set Alarm Time



Set Alarm Date

Month
November ▾ 10

Day
20

Year
2006

Alarm Time

5:41 AM

Date

11/20/06

Current Time

6:56 PM

Current Date

1/3/07

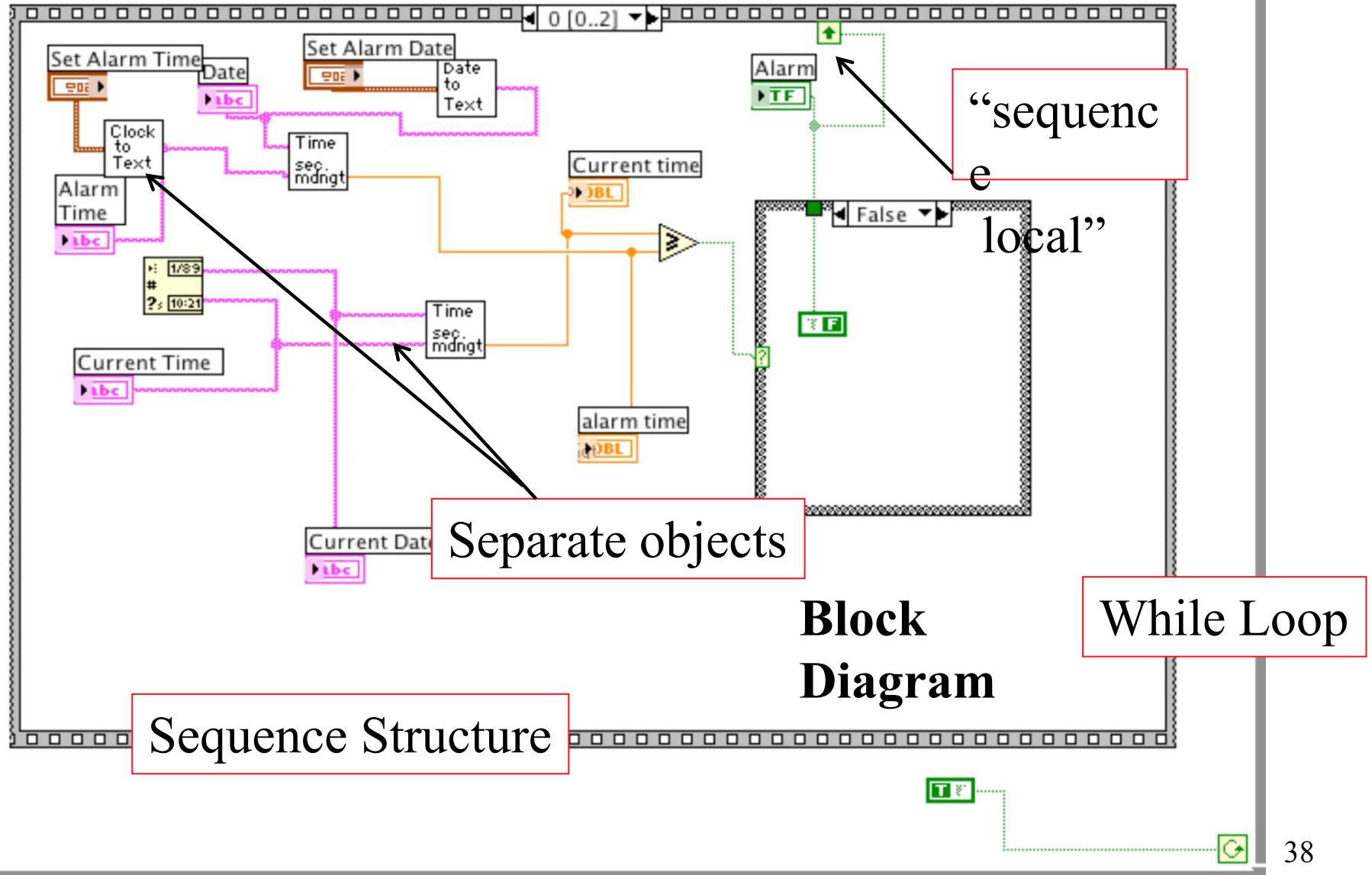


Alarm

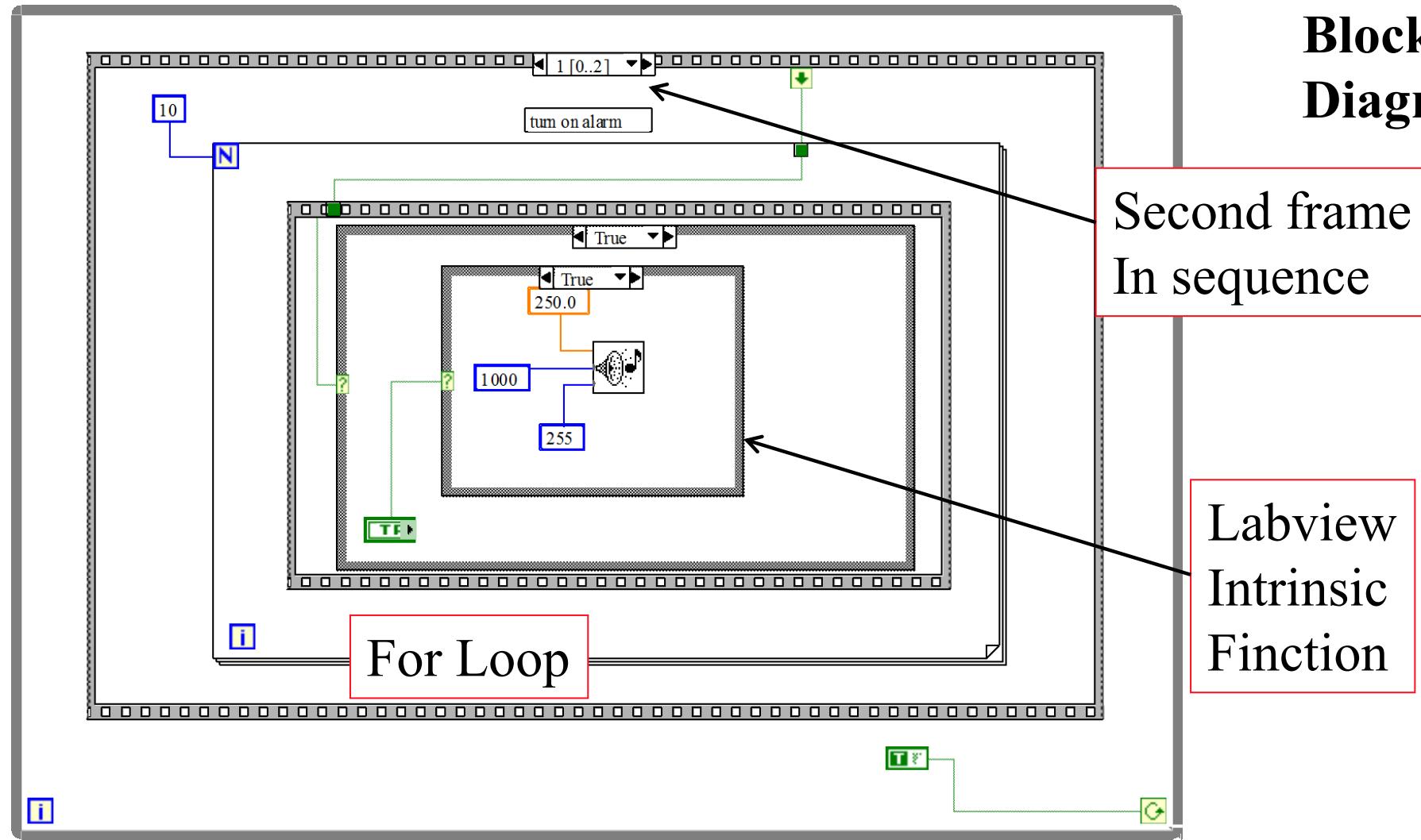
Alarm on!

Front Panel

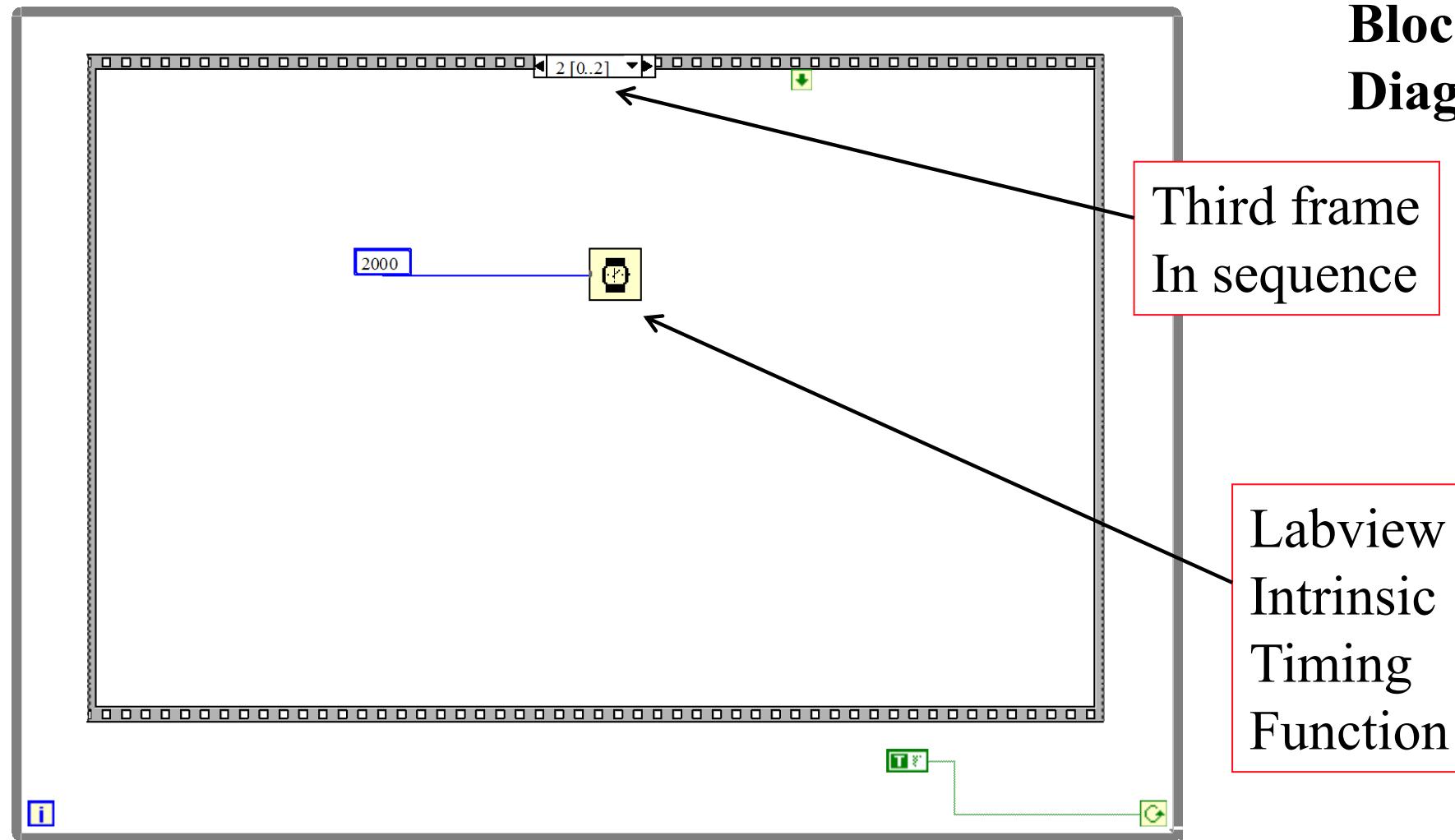
Example: My own alarm clock (2)



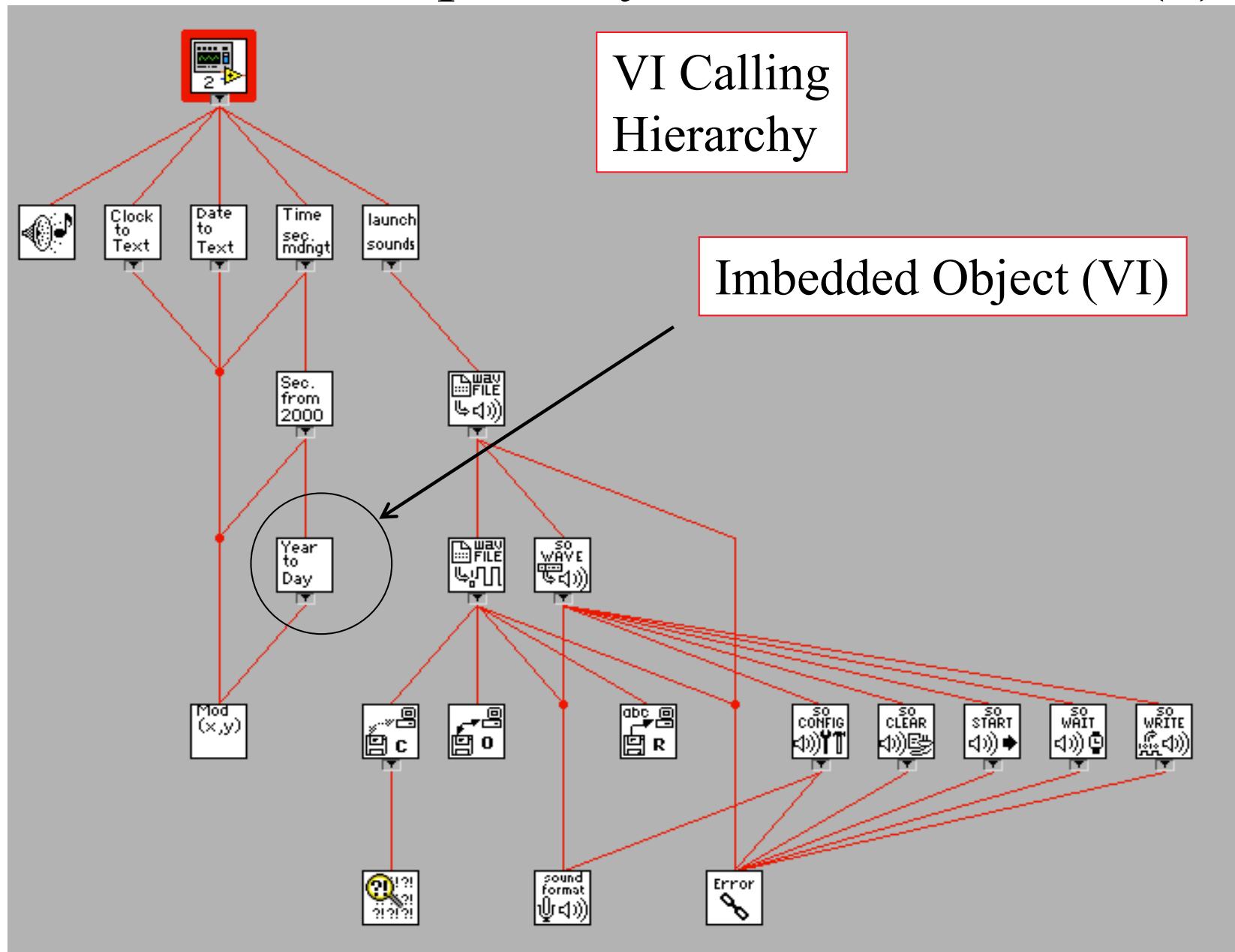
Example: My own alarm clock (3)



Example: My own alarm clock (4)



Example: My own alarm clock (5)

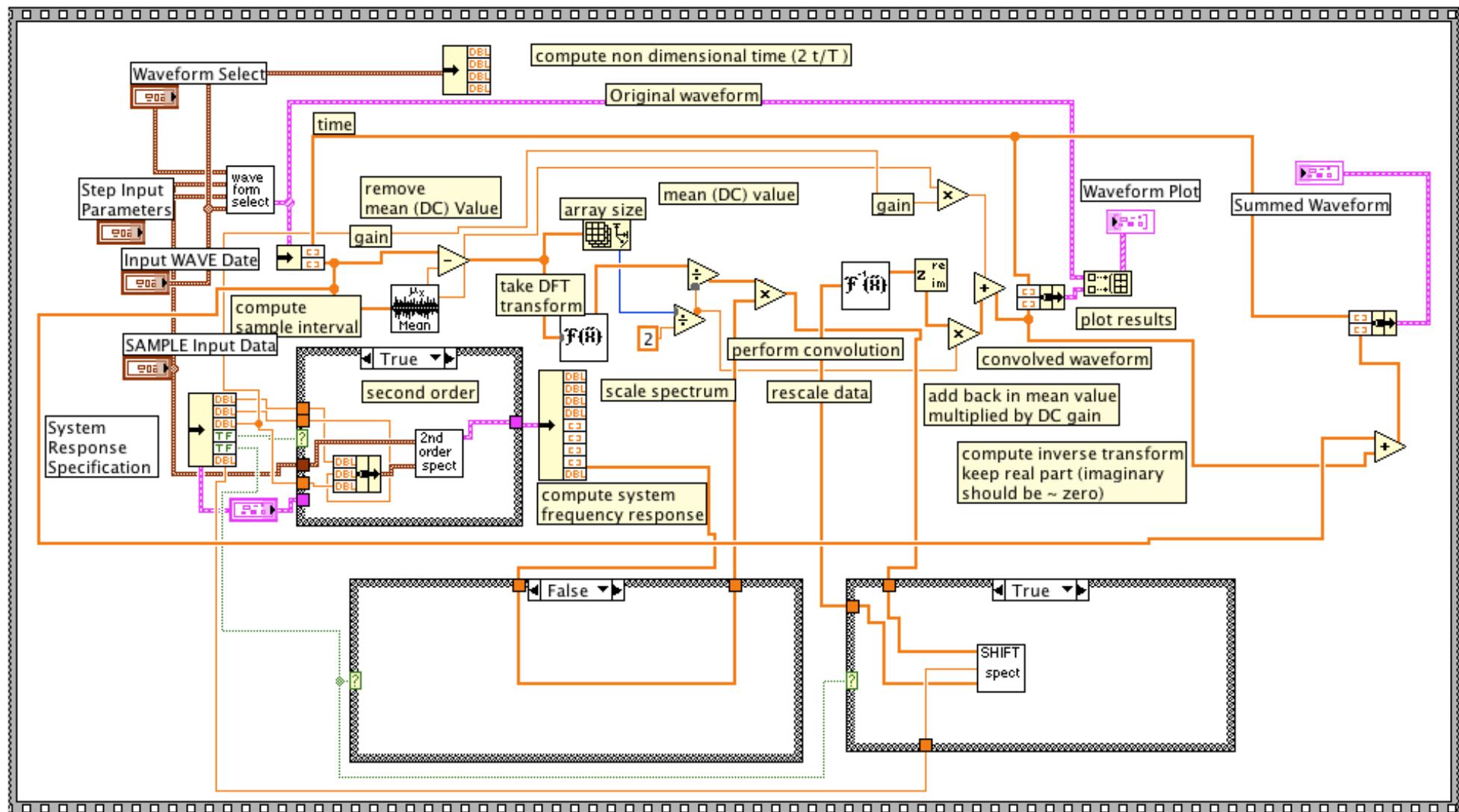


DFT Applications ... Noise Canceling headsets



- Essentially, this involves using a microphone, placed near the ear, and electronic circuitry which generates an "antinoise" sound wave with the opposite polarity of the sound wave arriving at the microphone. This results in destructive interference, which cancels out the noise within the enclosed volume of the headphone

“Simple” Noise Canceling Logic Written in Labview



Labview® Web Links

-- **National Instruments Main Page**

<http://www.ni.com/>

-- **Labview Overview**

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

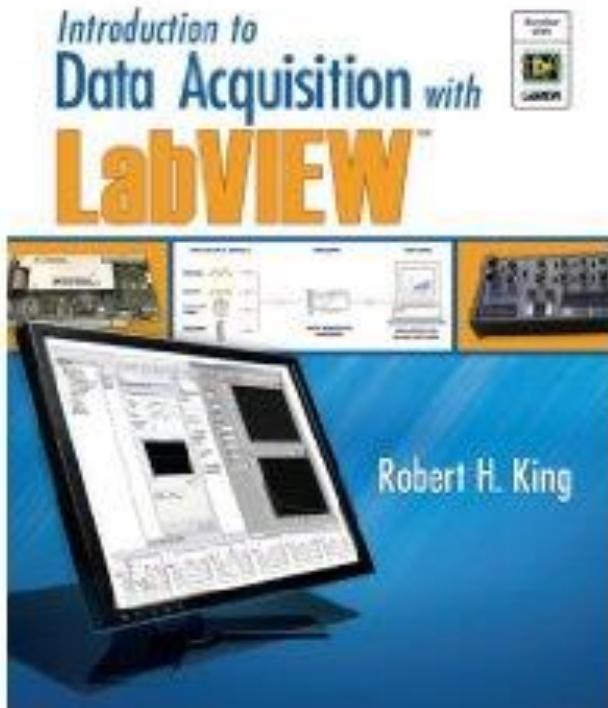
-- **Classroom Resources and Courseware
for Measurements and Instrumentation**

http://www.ni.com/academic/measurements_curriculum.htm

-- **Labview Tutorial page**

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

Recommended Reference Books



A copy of the Student Edition of LabVIEW is included with each textbook.

The text prepares students to pass the NI international student certification examination for LabVIEW.

The text contains numerous worked out examples that demonstrate common uses of LabVIEW with guidance for students.

Authors and Organization:
Robert H. King, Colorado School of Mines

Publisher: [McGraw-Hill](#)

Copyright Year: 2009

Edition: 1

ISBN-10: 0077299612

ISBN-13: 9780077299613

Questions?

