

# AOE 3054

Data Acquisition with LabView

# Connect NI USB-6211

- The USB DAQ connects like any other USB device.
- You can connect the white cable to the extension cable provided on the workbench or connect directly to the computer.

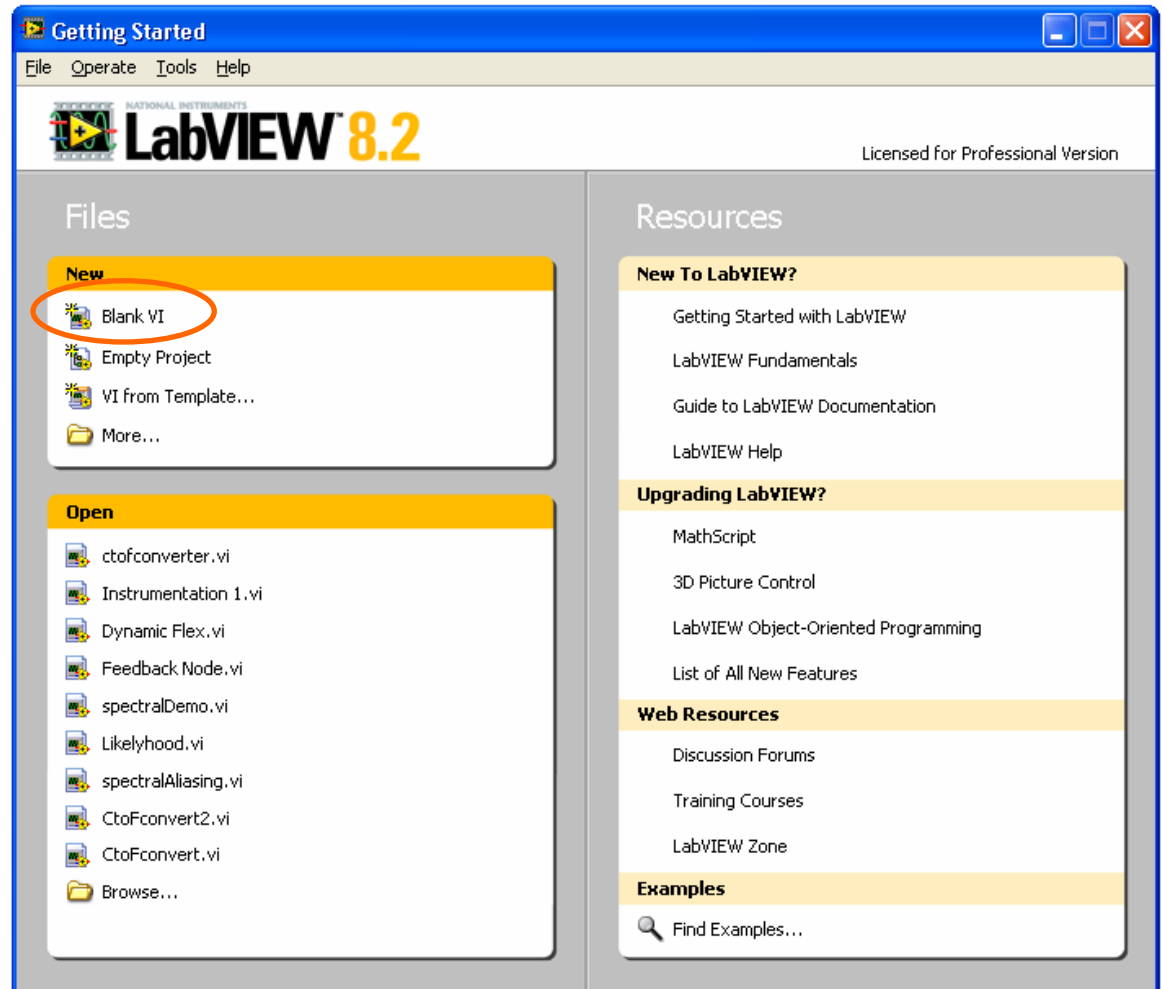
If this is the first time the DAQ has been plugged into the computer, you will need to follow the driver install procedure which is similar to any other USB device.

Next, Open a Blank VI.



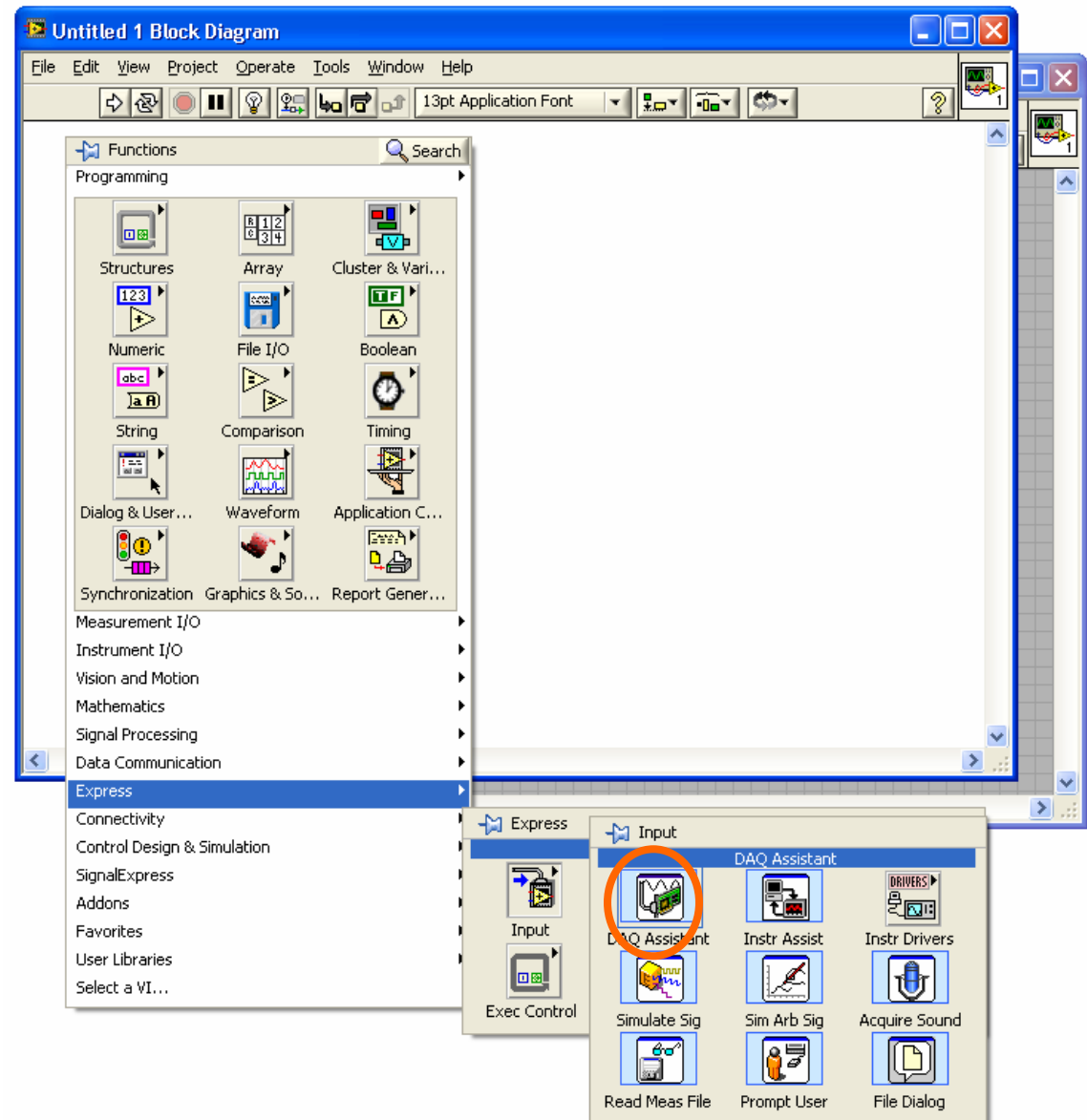
# Open a New Blank VI

- To begin a new program (or VI) click **Blank VI**.

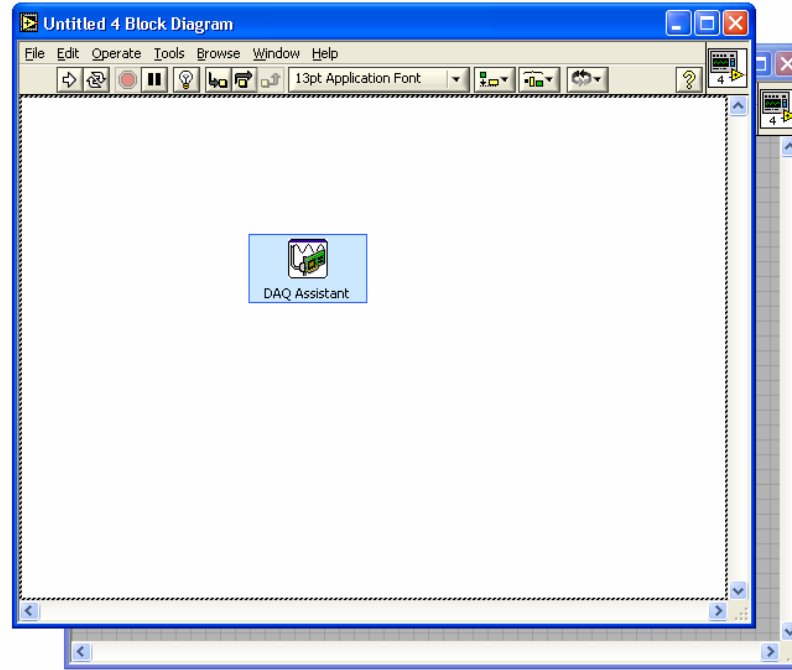


# Select DAQ Assistant

From the Block Diagram,  
Right click and select Input,  
DAQ Assistant.



# Place VI on the Diagram

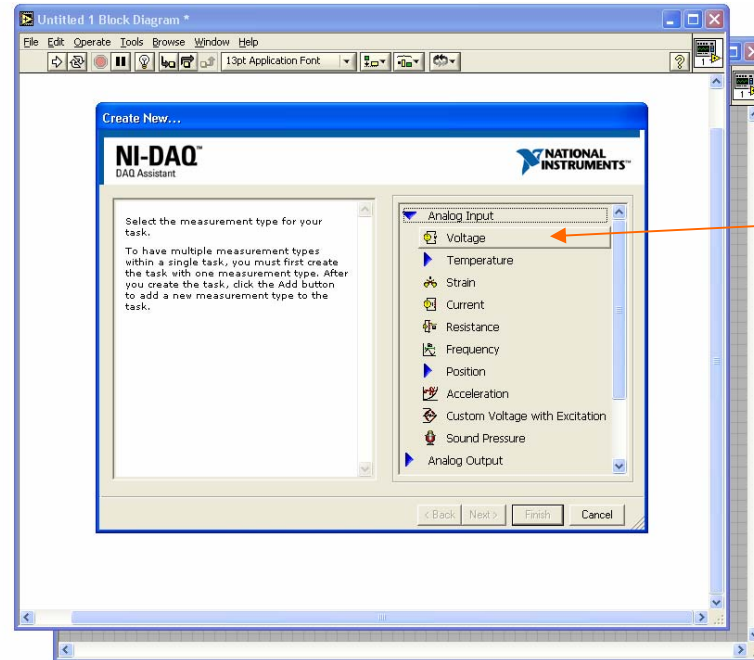


# Configure the DAQ Assistant...



DAQ Assistant is a special type of sub VI which will automatically open a configuration window. The sub VI allows several operations to be configured. To read data into the computer, select Analog Input.

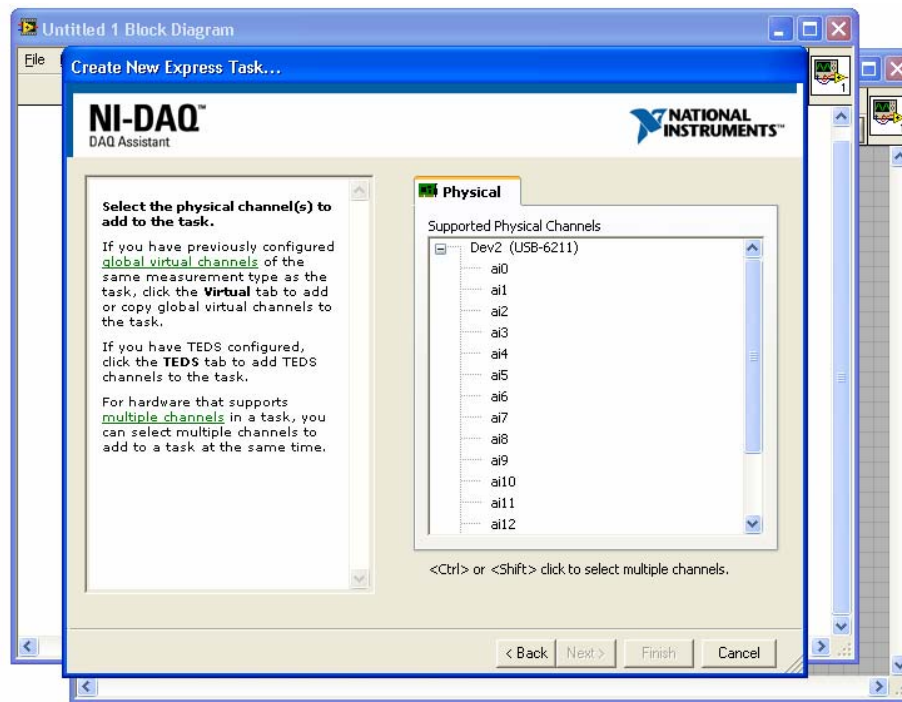
# Configure the DAQ Assistant



For now, select *Voltage* under *Analog Input*.

Analog input channel configuration allows conversion of inputs to some standard values. Use these when possible.

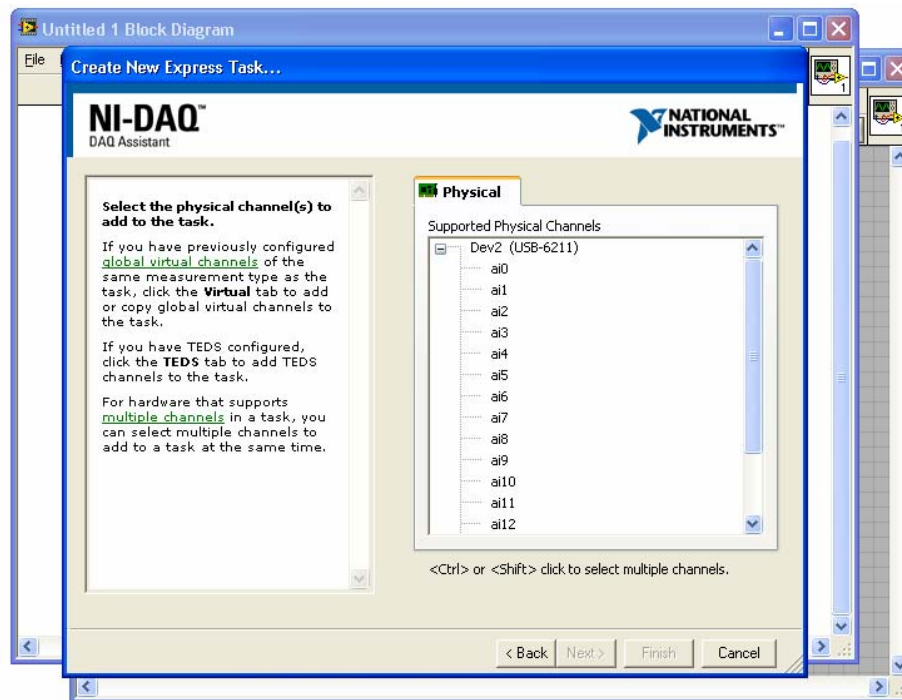
# Physical Channels



You should then see this screen with the device and channels listed of the DAQ you are using. If you get “No supported devices found”, then check the installation of the DAQ device in NI MAX (Measurement and Automation Explorer).



# Physical Channels

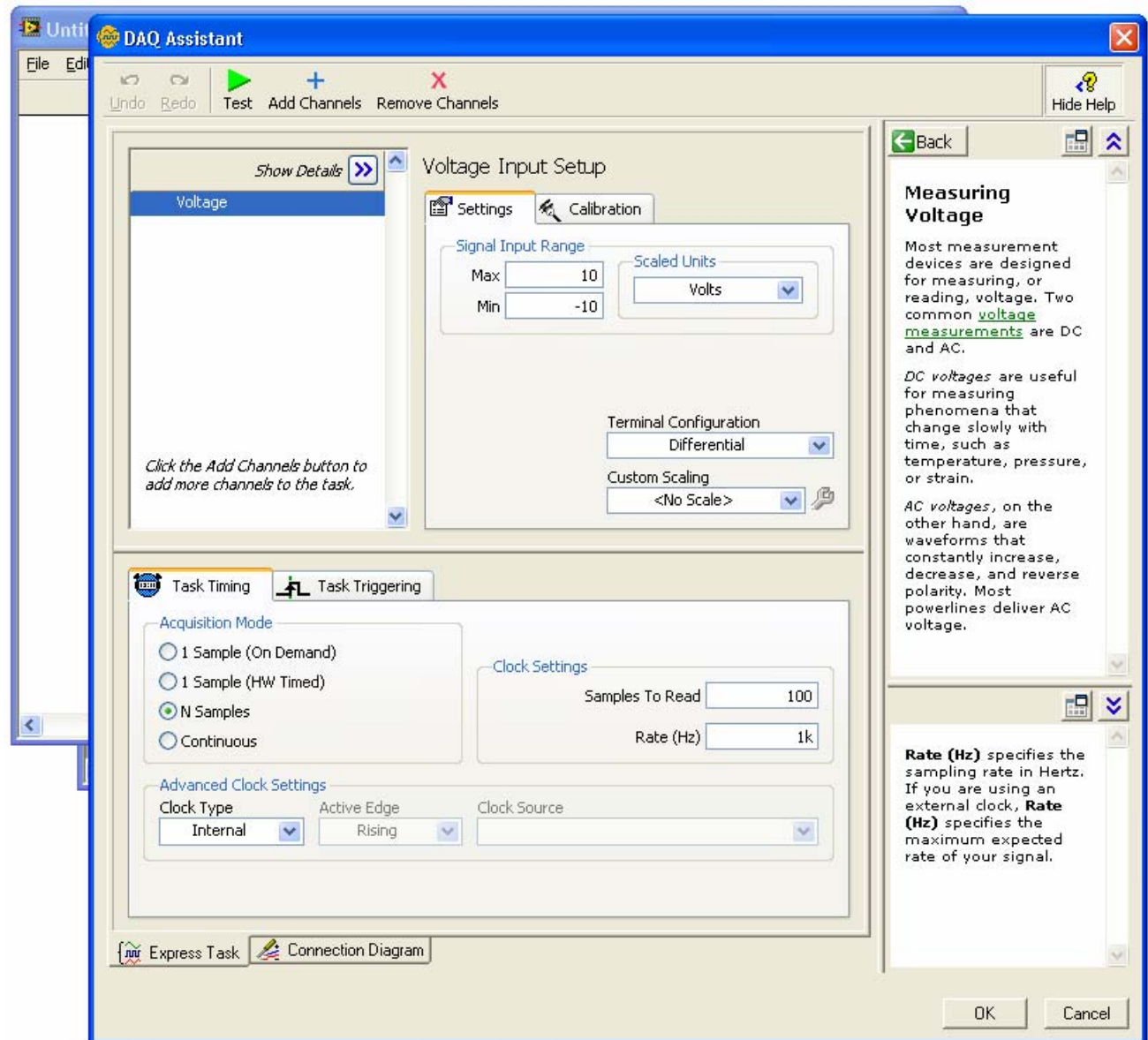


Select ai0 and click **Finish.**

You can select **one** channel or **multiple** channels here. **This device allows for up to 16 single ended channels of input.**

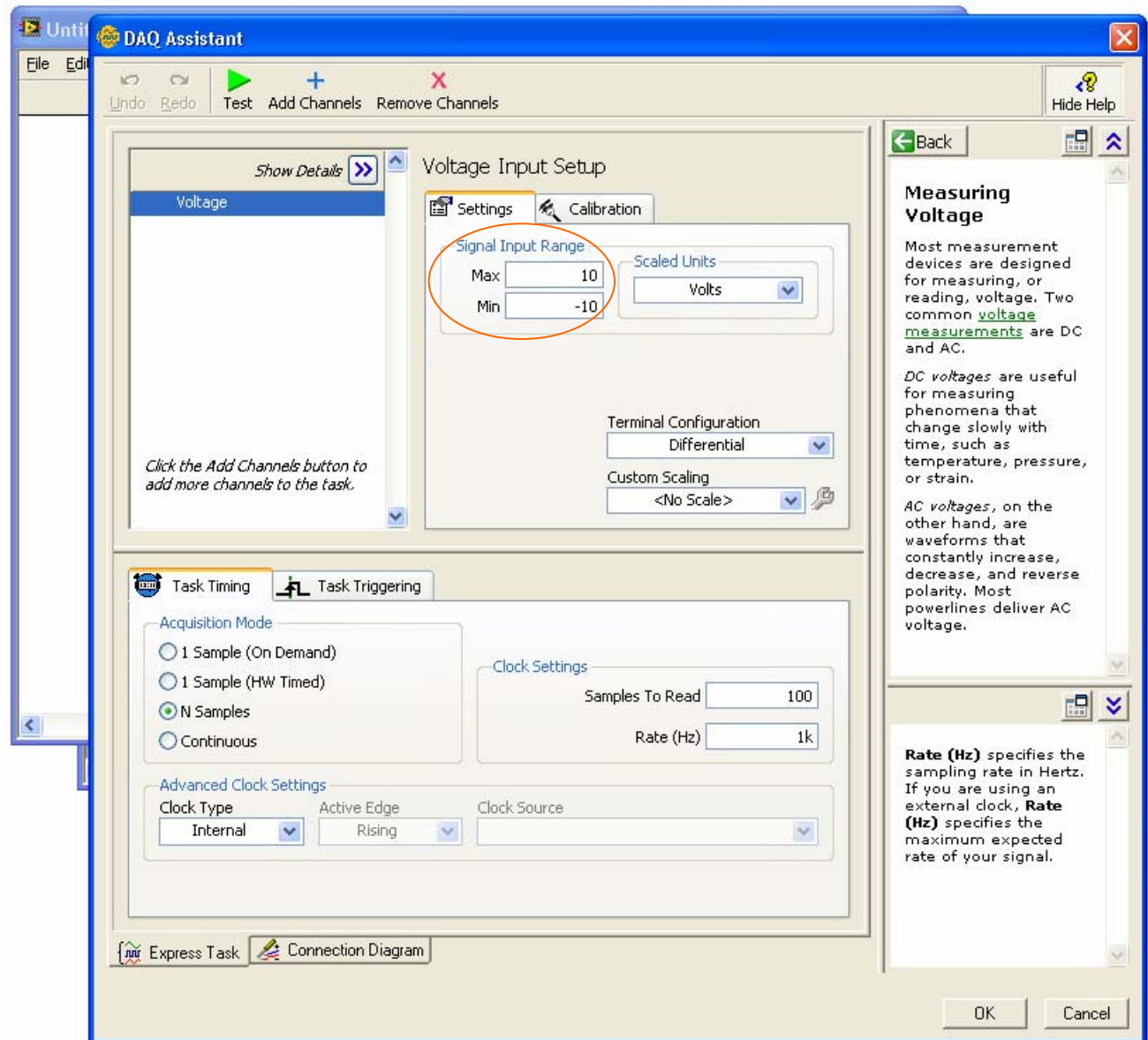
# Voltage Input Setup

Here we setup the amplifier of the DAQ. Voltages enter through the wire connected to the DAQ, that will be connect later, and are translated into computer numbers that we can manipulate and record.



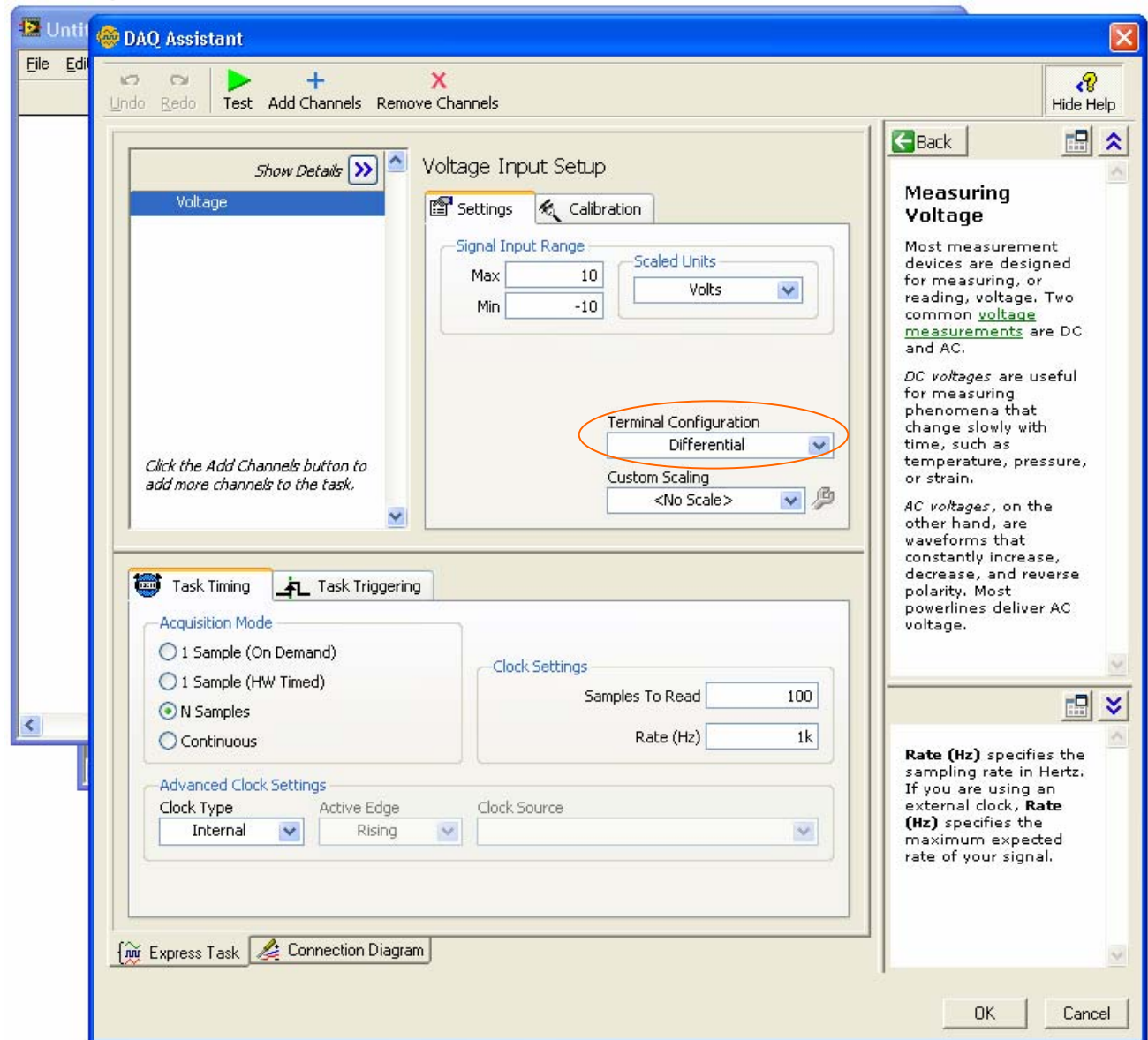
# Voltage Input Setup

*Input Range* defines the span of voltage the DAQ will measure. Narrowing the range will concentrate the Analog to Digital Converter allowing better resolution of measurements. This translates to amplifier gain of the input of the DAQ.



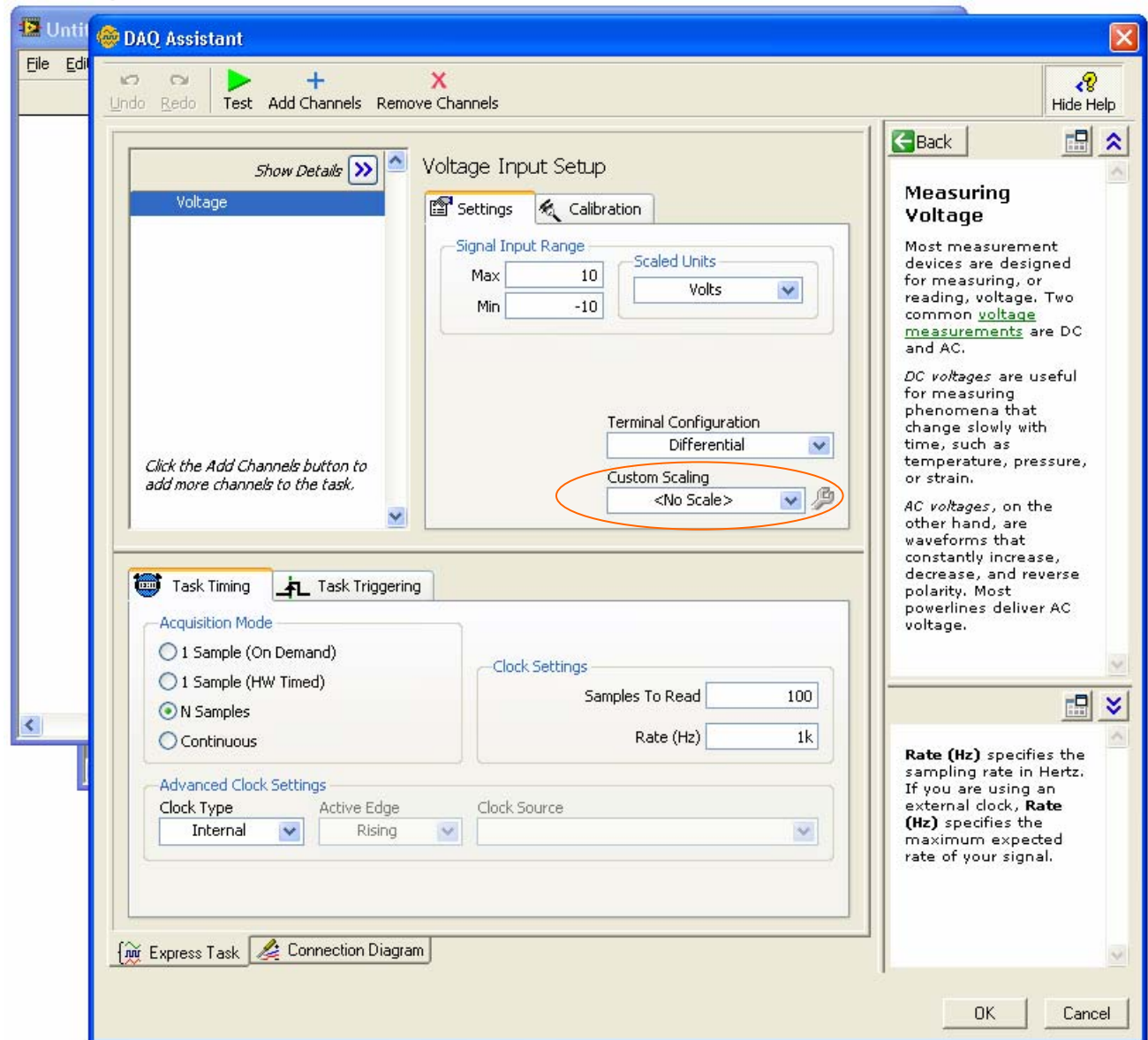
# Voltage Input Setup

Terminal connections can be made using **Differential** or **Single Ended**. Differential gives better common mode rejection and cleaner signals if the connections are properly wired.



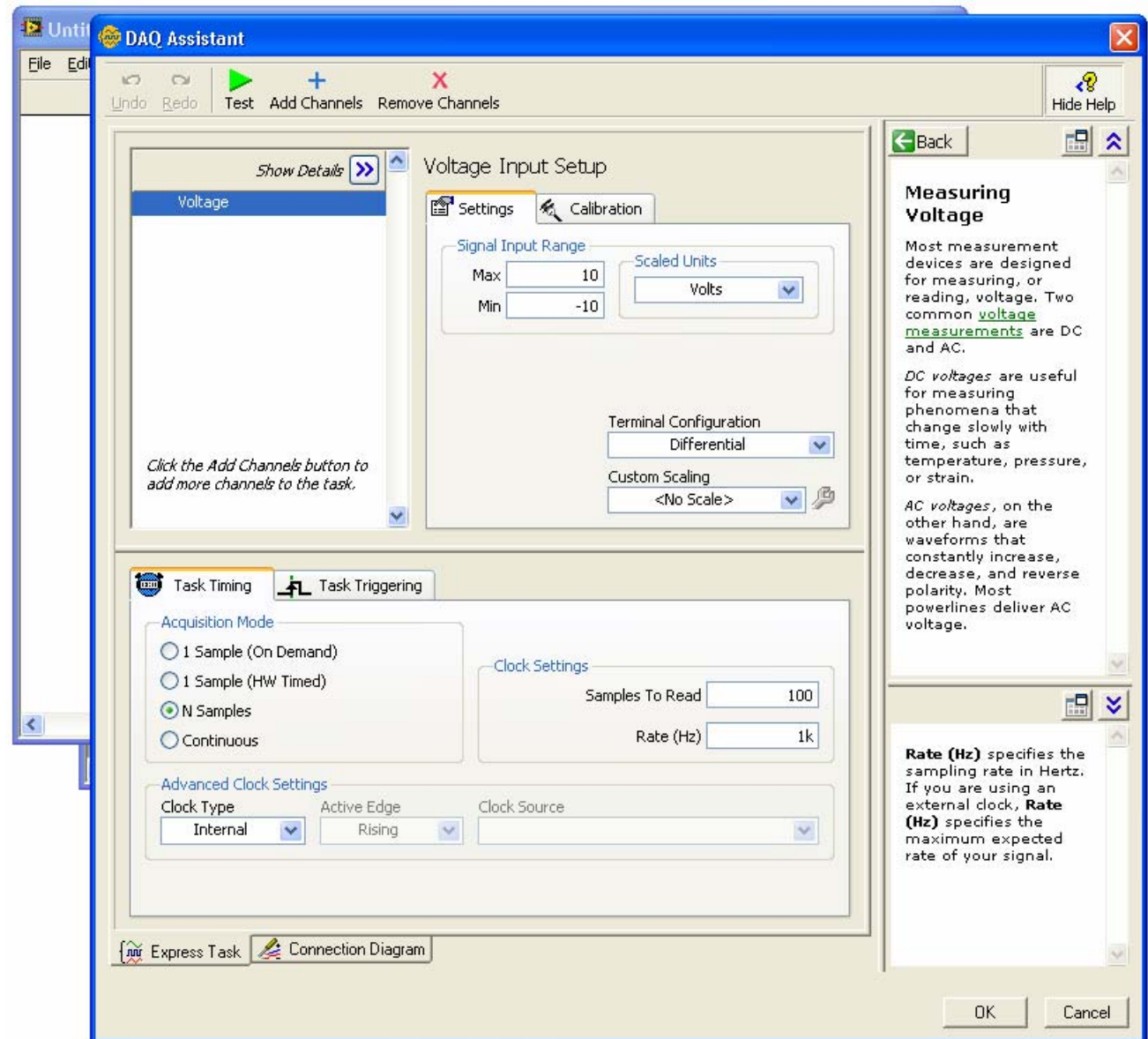
# Voltage Input Setup

Scaling converts a somewhat arbitrary voltage into a meaningful unit of measure. For instance, 1 Volt equals 2 PSI. There are different ways to achieve scaling

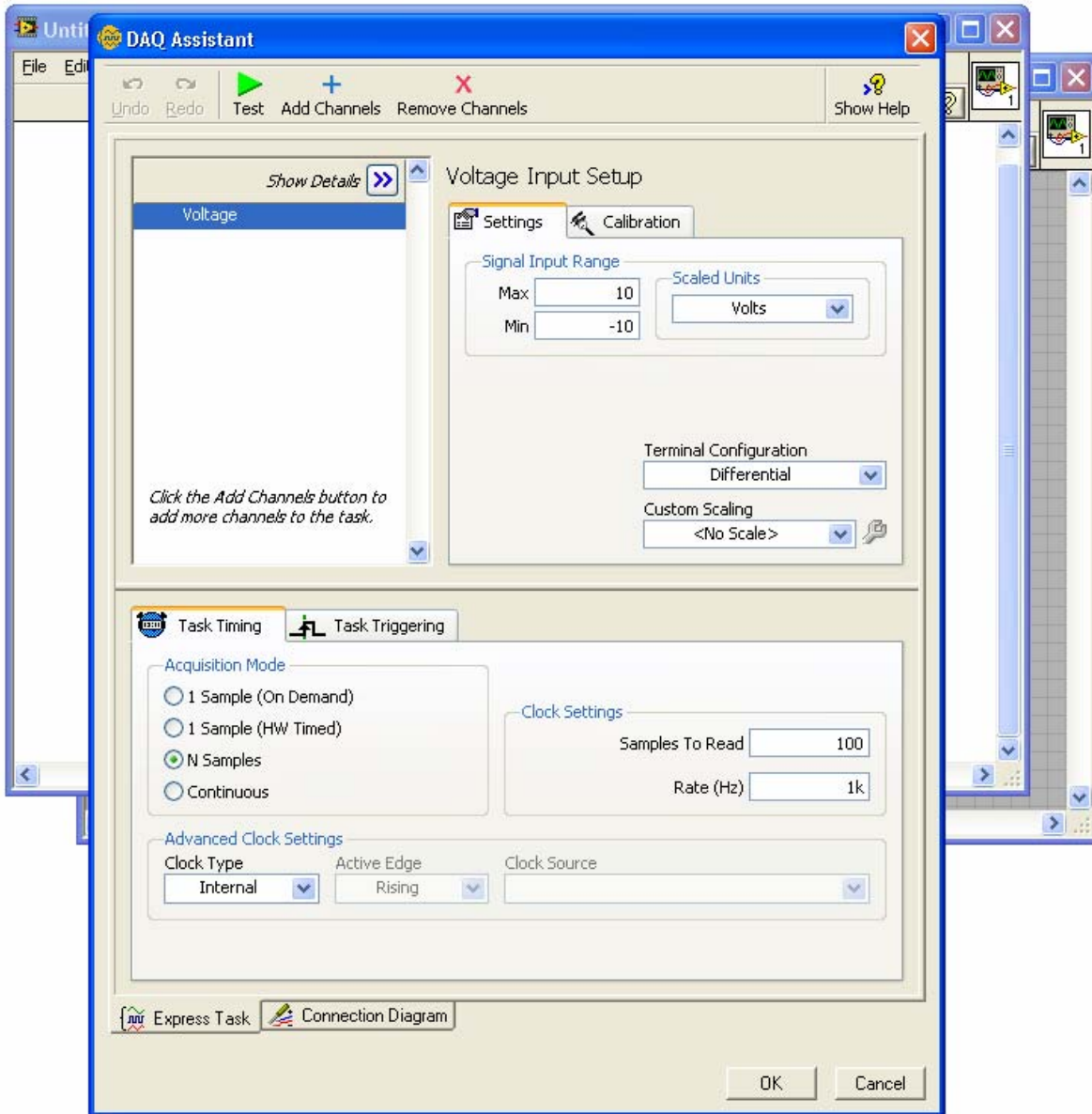


# Voltage Input Setup

For our first measurement, we can use the default settings.

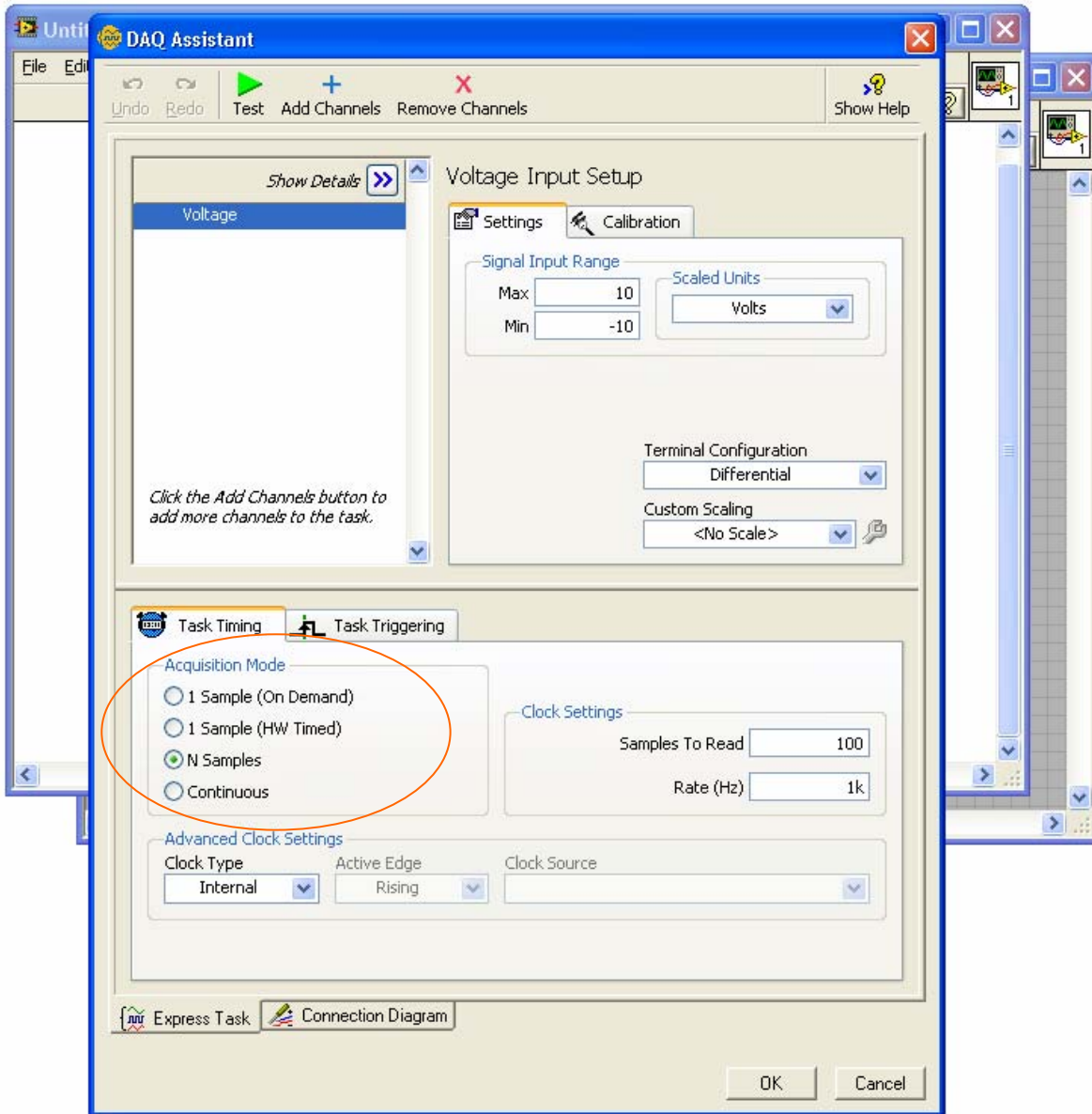


# Task Timing



We can configure the operation of data acquisition timing for the program. Multiple point data acquisition can be thought of as gathering several individual data points and putting them together like a moving picture show.

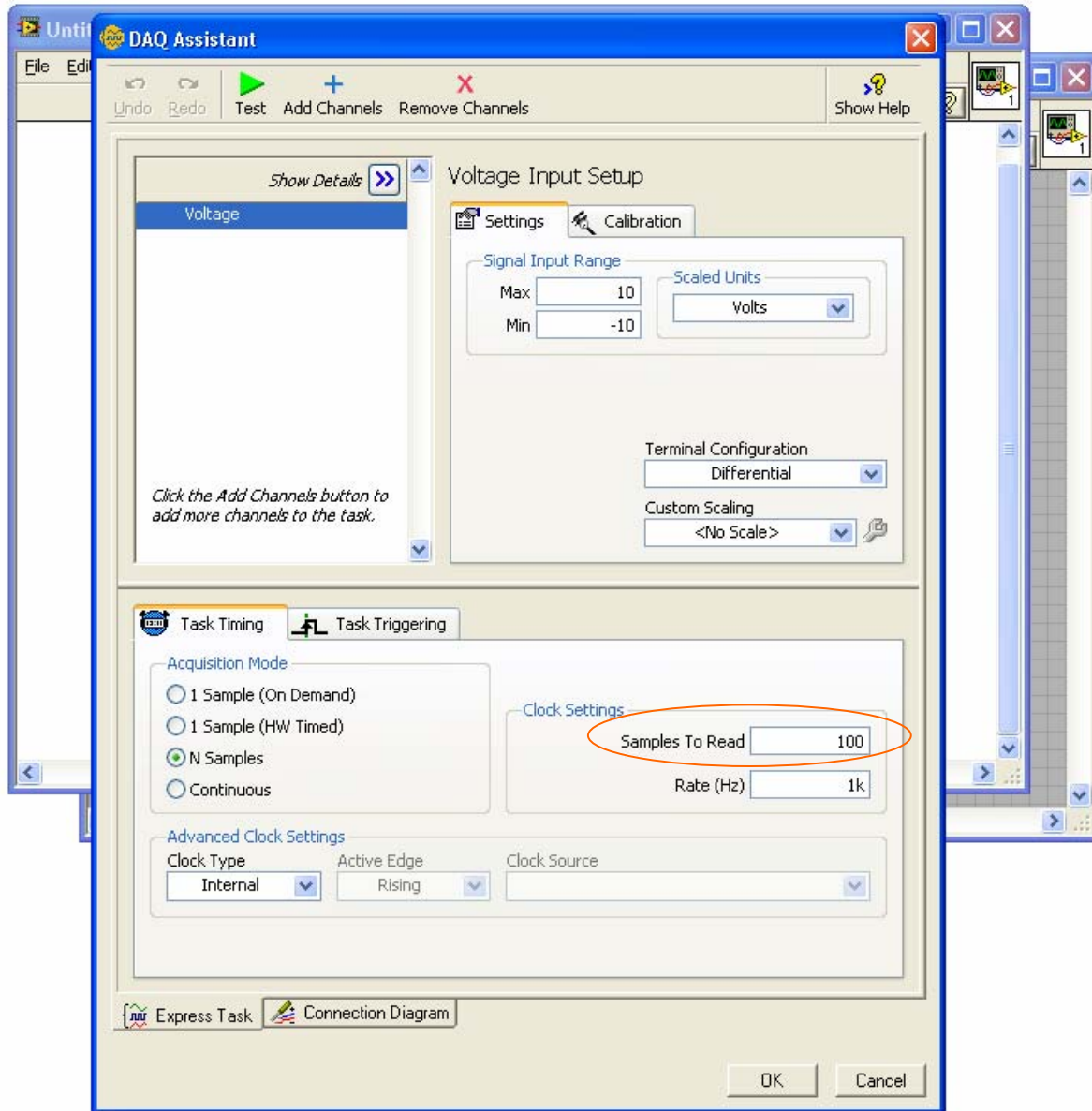
# Task Timing



Several options exist for acquisition mode from *single point* to *continuous*. We will look at the default selection, *N Samples*.

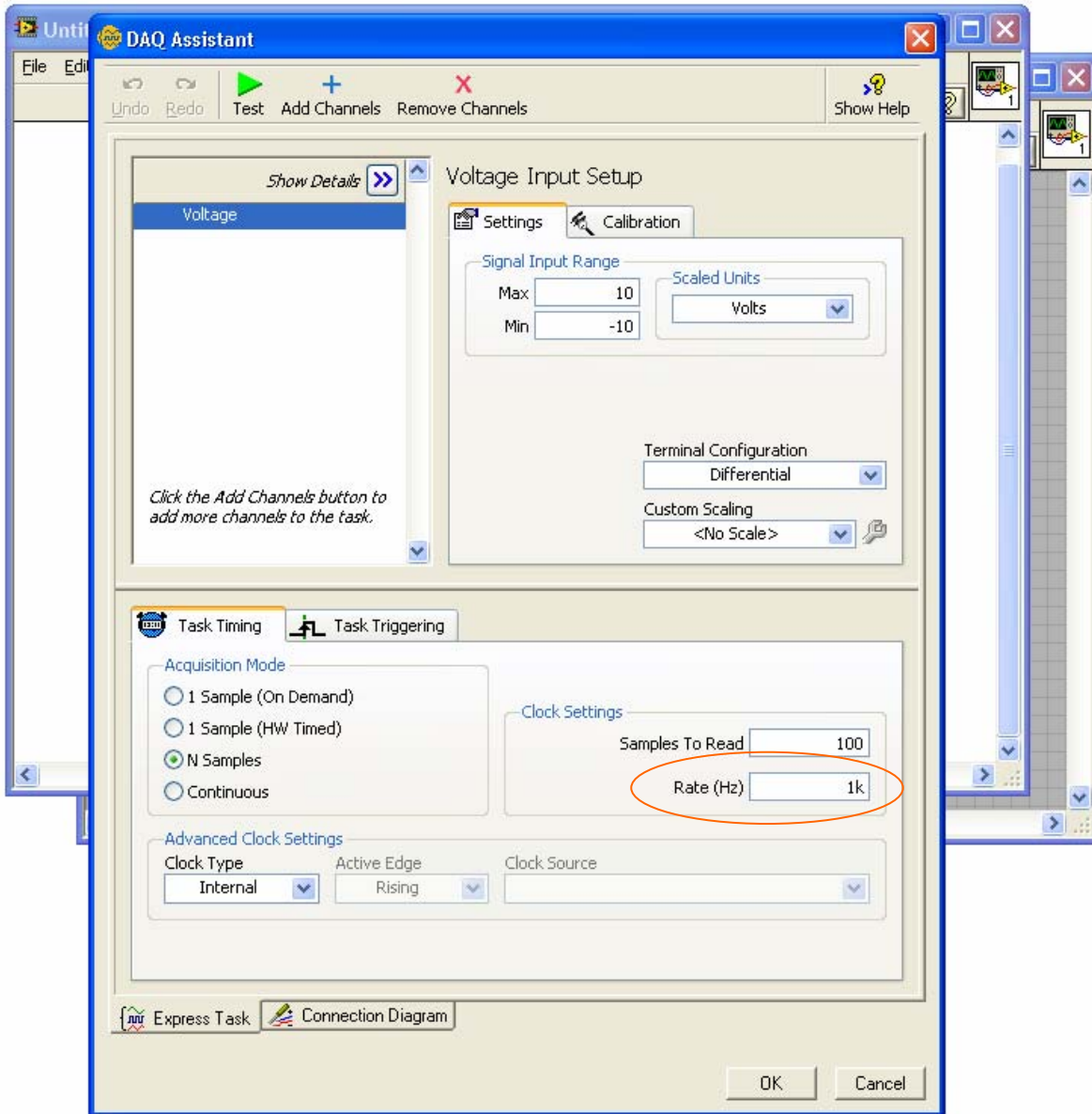


# Task Timing



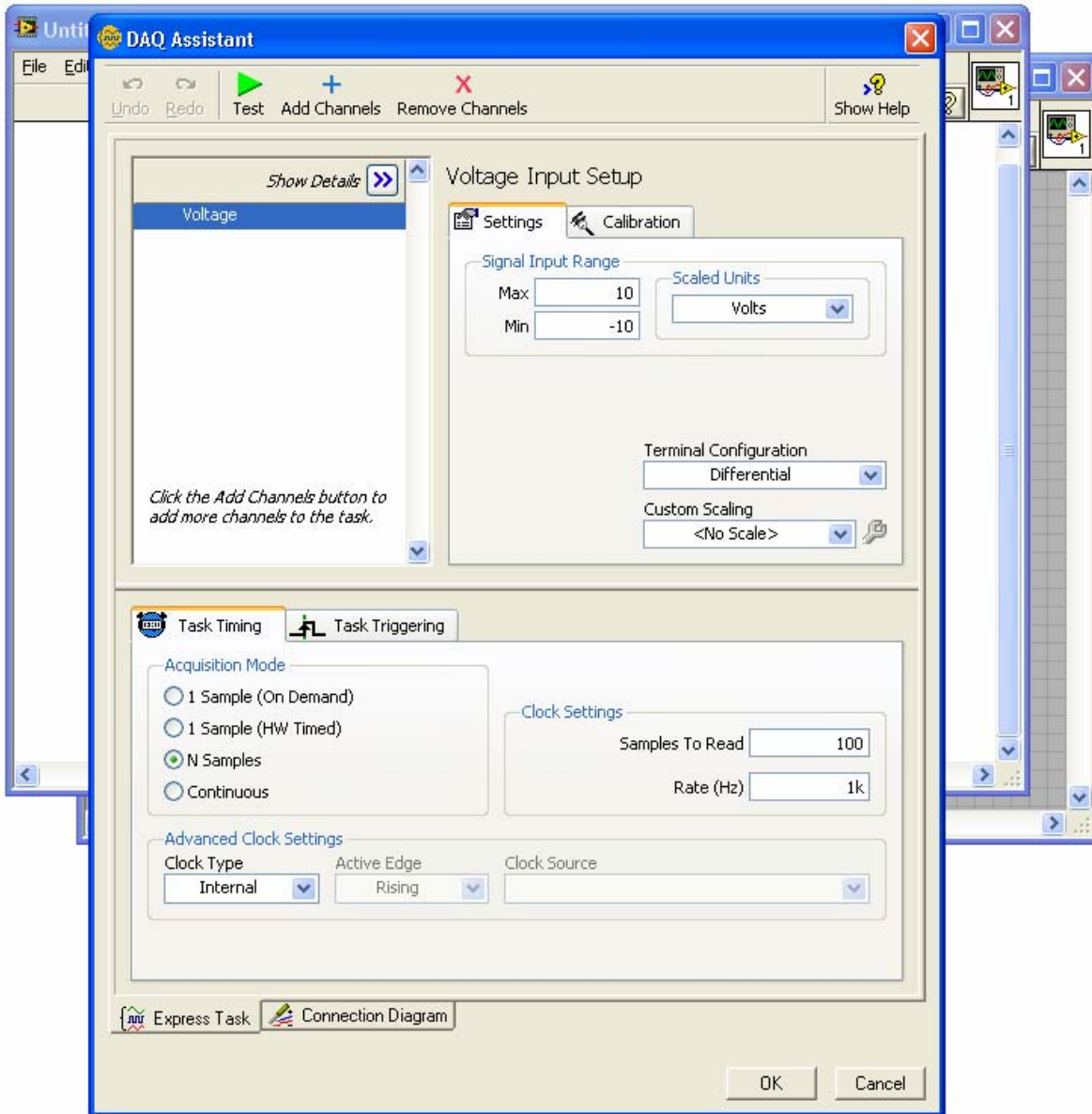
*Samples to read* defines how many data points the DAQ will return when this VI is called.

# Task Timing



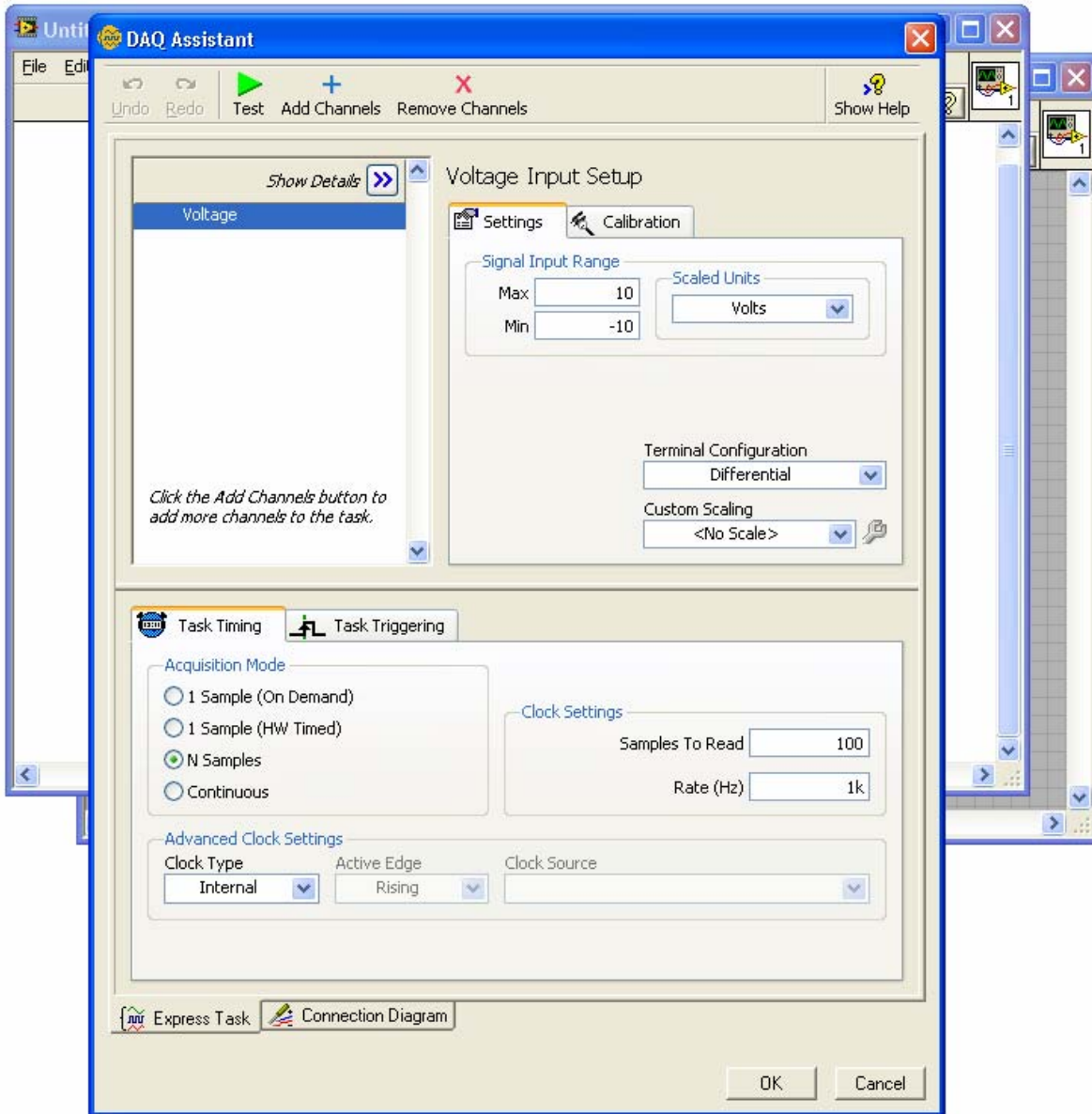
*Rate* tells the DAQ how fast to acquire the data.

# Task Timing



Divide the sample rate by the number of samples to determine the sample time. If we use 1000 samples at 1000Hz, that will be 1 second of data.

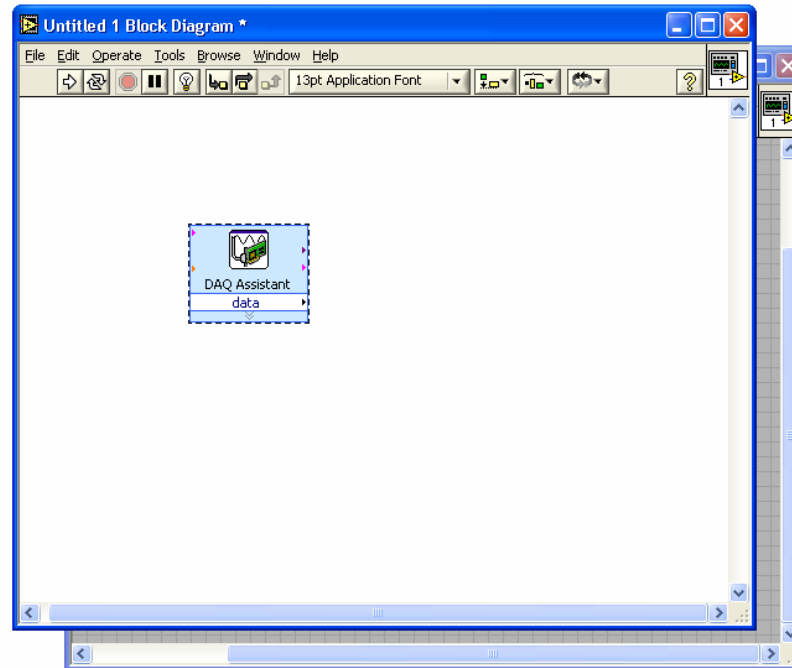
# Task Timing



Click OK and the DAQ Assistant will build the code for the operations you selected. These can be changed later by double clicking the DAQ Assistant Icon.

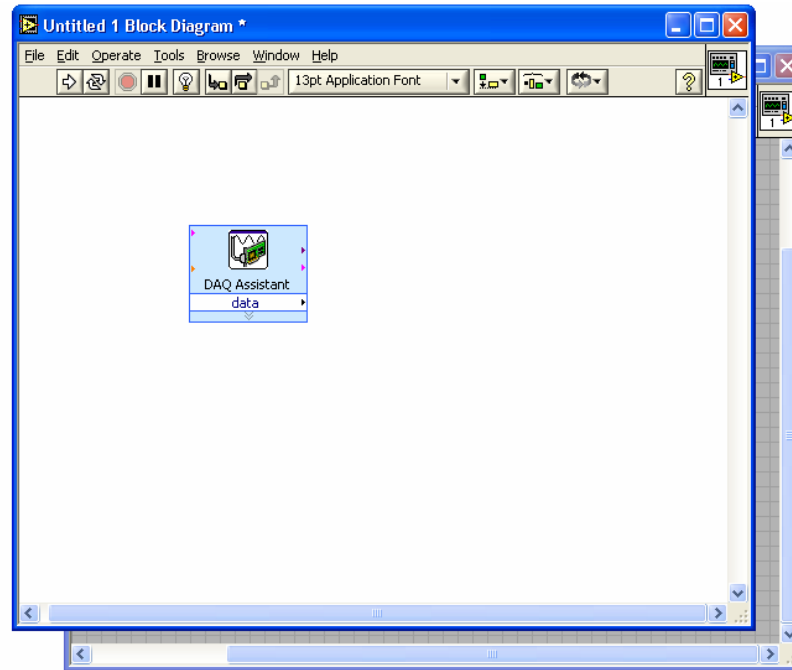
# Getting Data out of the VI

Click in the  
White space...

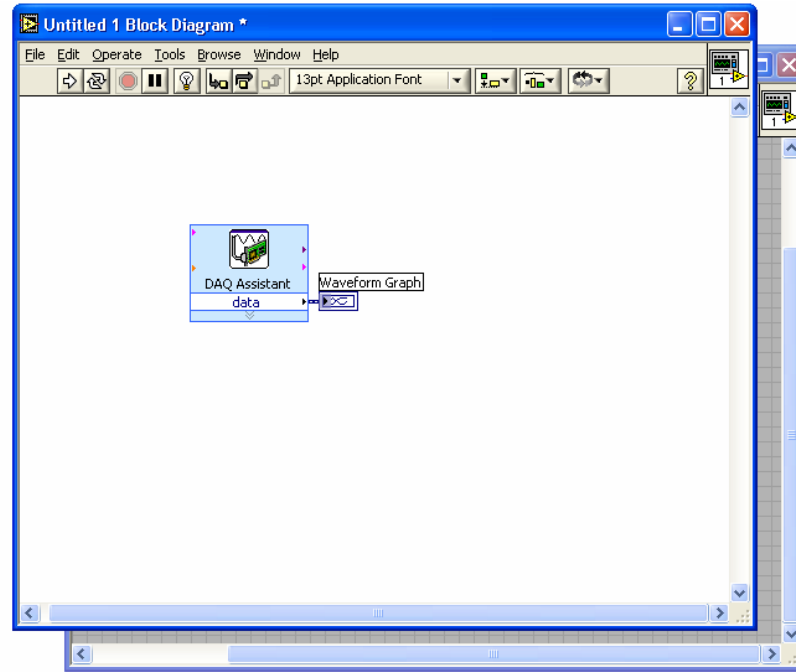


# Create Graph Indicator

Hover over the right arrow of the terminal *Data...* until the wiring tool appears and then right click and Select Create >, Graph Indicator.

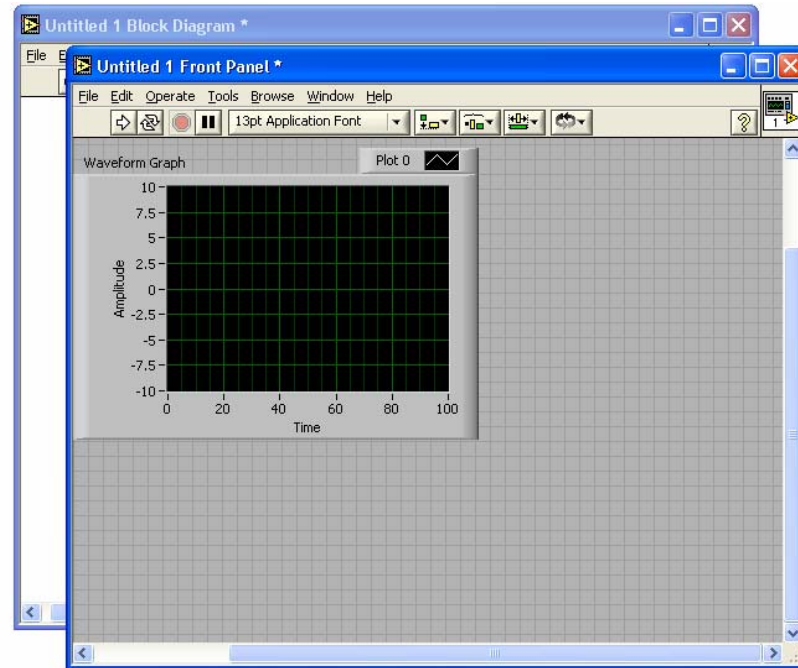


# Create Graph Indicator



Now take a look at the Front Panel...

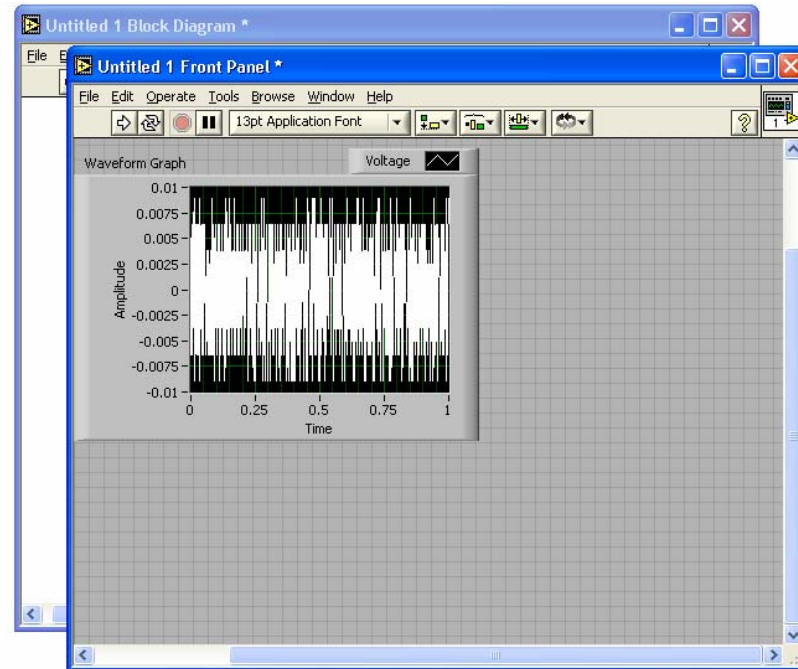
# Ready to Test!



Click Run, wait for the second of data to be collected and...

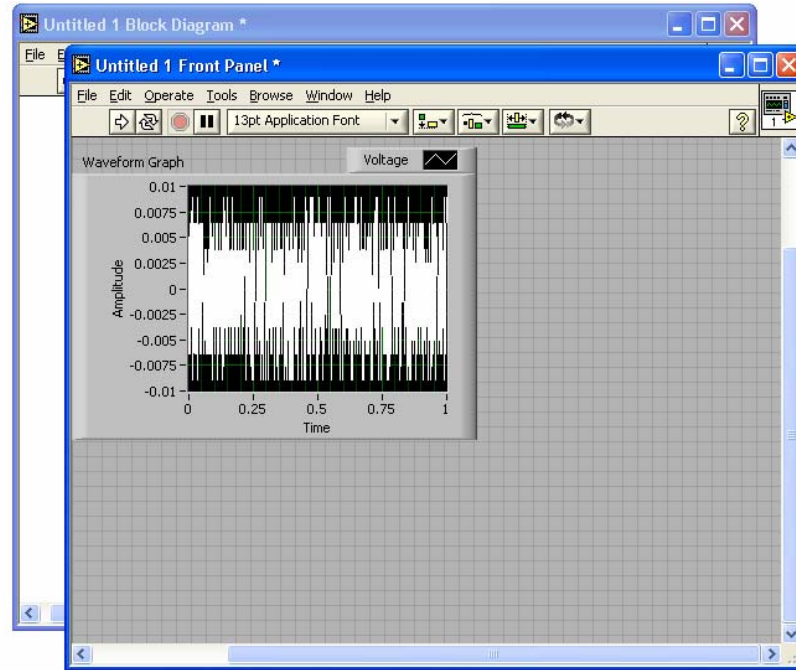


# Data!



But What is it? Shouldn't it be zero?

# Data!



Since the DAQ is not connected to anything, we don't know what will result from the measurement. But we can see it is picking up some noise. If it were a flat line or just one point, that would indicate some other problem.

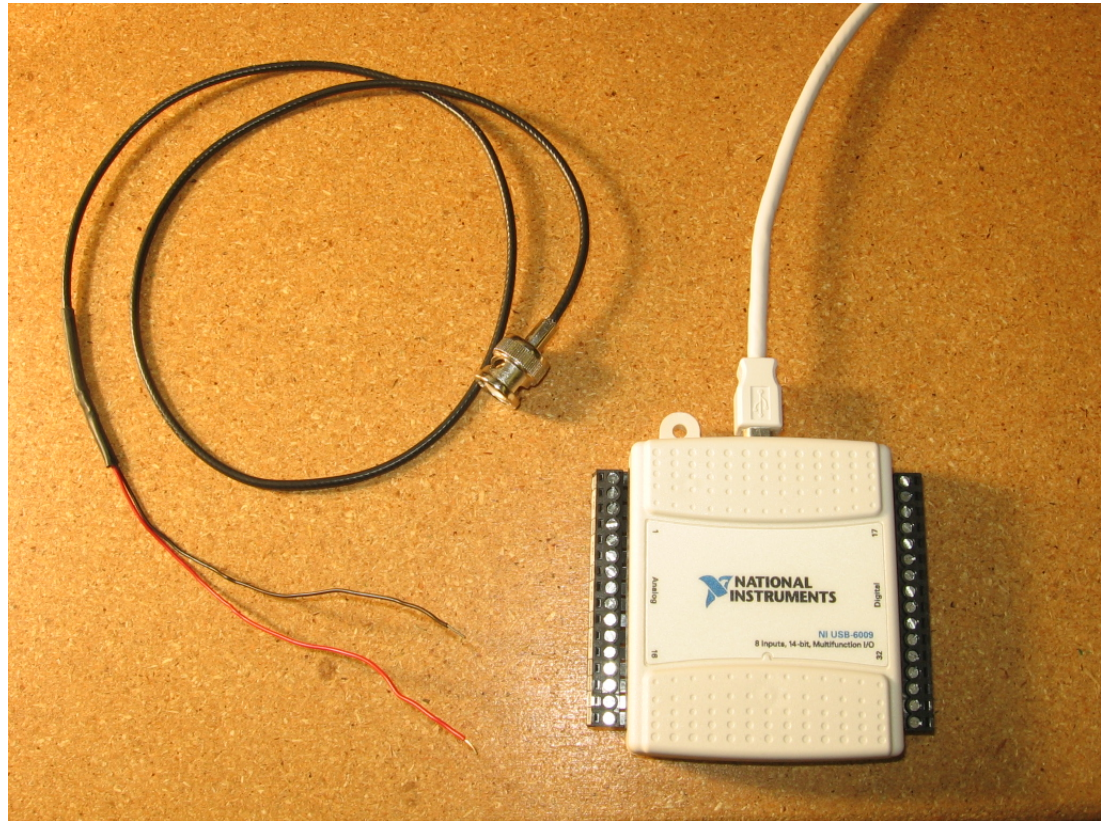
# Saving the vi

- File, Save to save the vi.
- Make current values default to save front panel settings.

# Wiring

- Making the device useful.

Use a cable with a BNC connector on one end and bare wire on the other end to connect the DAQ to the function Generator.

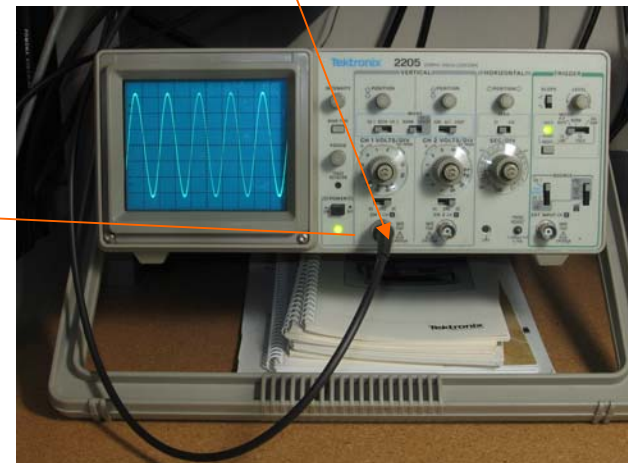
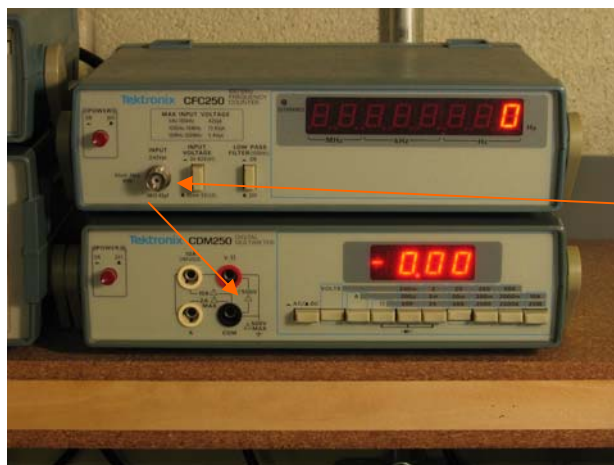
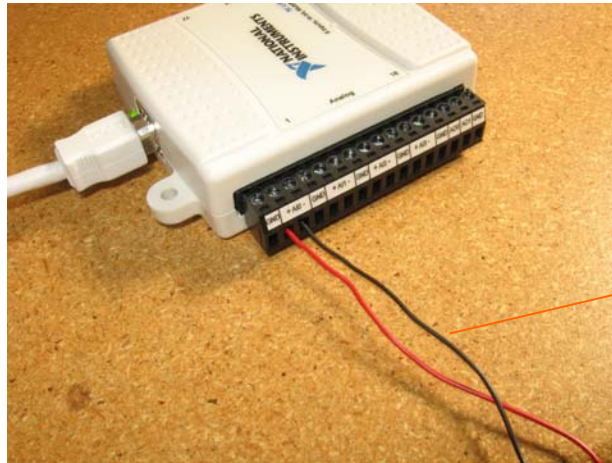


# Analog In Terminals



Connect the bare wires to the AI0 + and – terminals on the DAQ

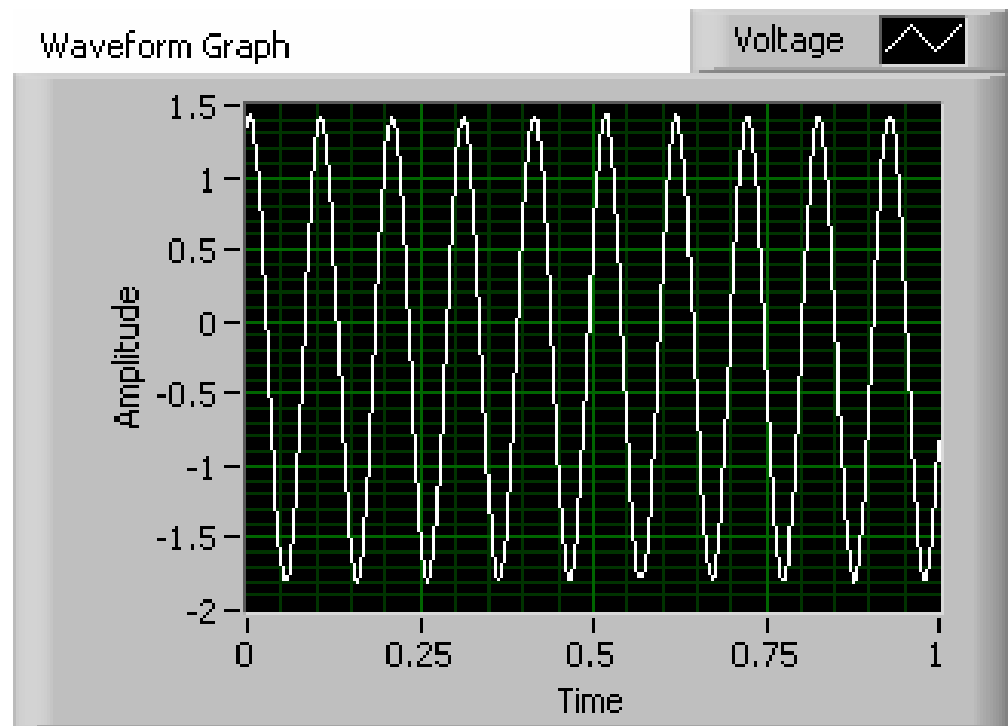
# Complete Circuit



# Run the program

- Now set the function generator on 10Hz
- Run the VI

Depending on the settings of the function generator, the amplitude scale could be different from the picture.



# Task Timing Settings

- To explore the operation of the equipment, A few changes need to be made to the settings.
  - DAQ Assistant
    - Under *Clock Settings*, Change **Sample Rate** to **10000Hz**.
    - Change **Samples** to **500**.
  - Function Generator
    - Set Frequency to **100Hz** Sine Wave.

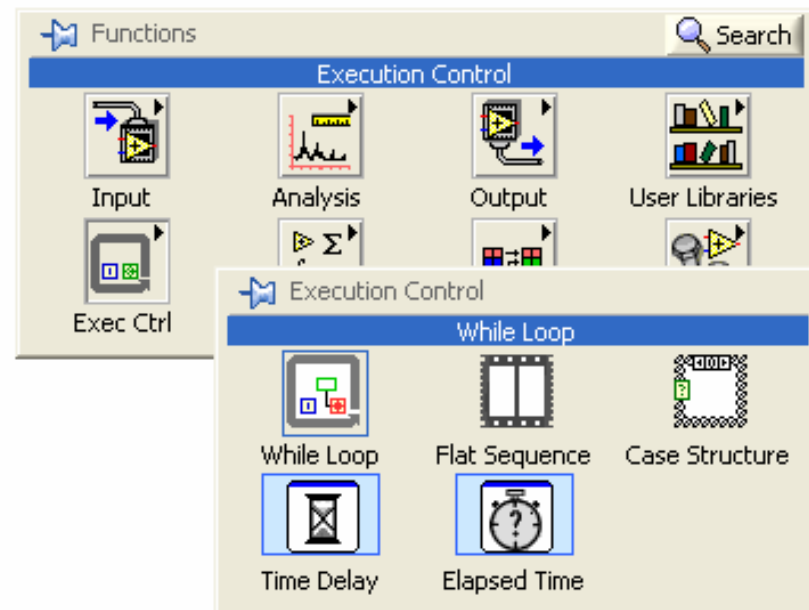


# Task Timing Settings

- Run the VI again and the output should be very similar to the previous test with the exception of the time scale which should be 0 to 0.05 seconds.
- Now modify the vi to have a **while loop** around the section of code on the block diagram.

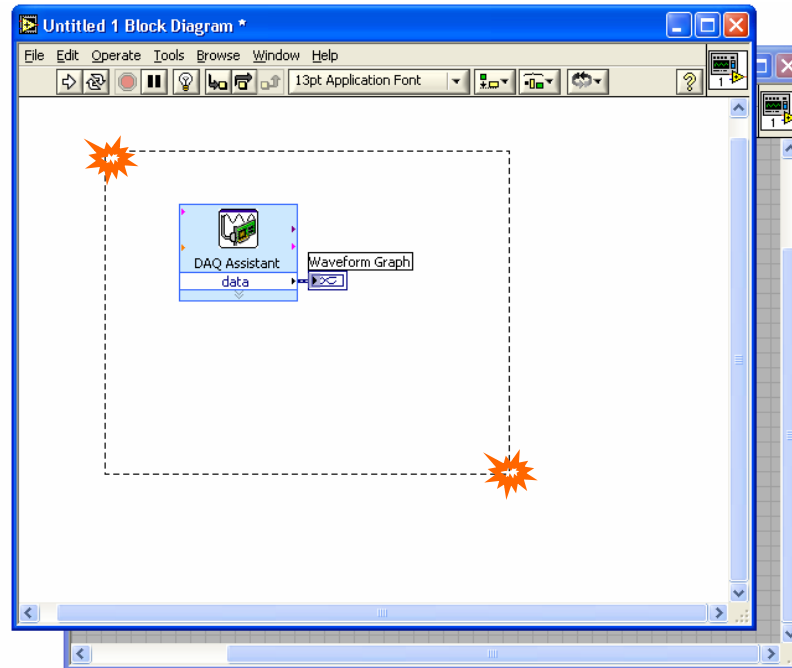
# Add While Loop

- On the Block diagram, right click and select while loop from the Execution Control Pallet.



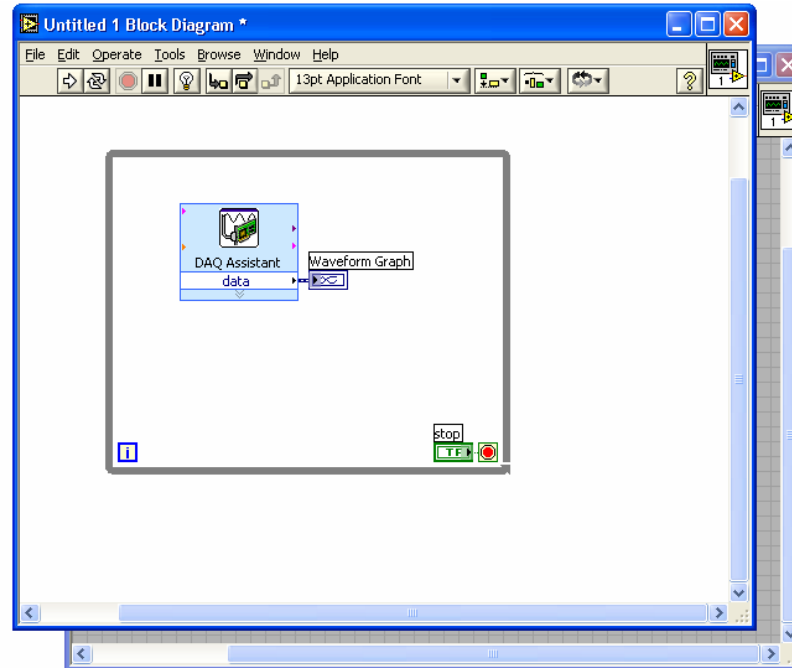
# Add While Loop

Click in the  
upper Left  
corner...



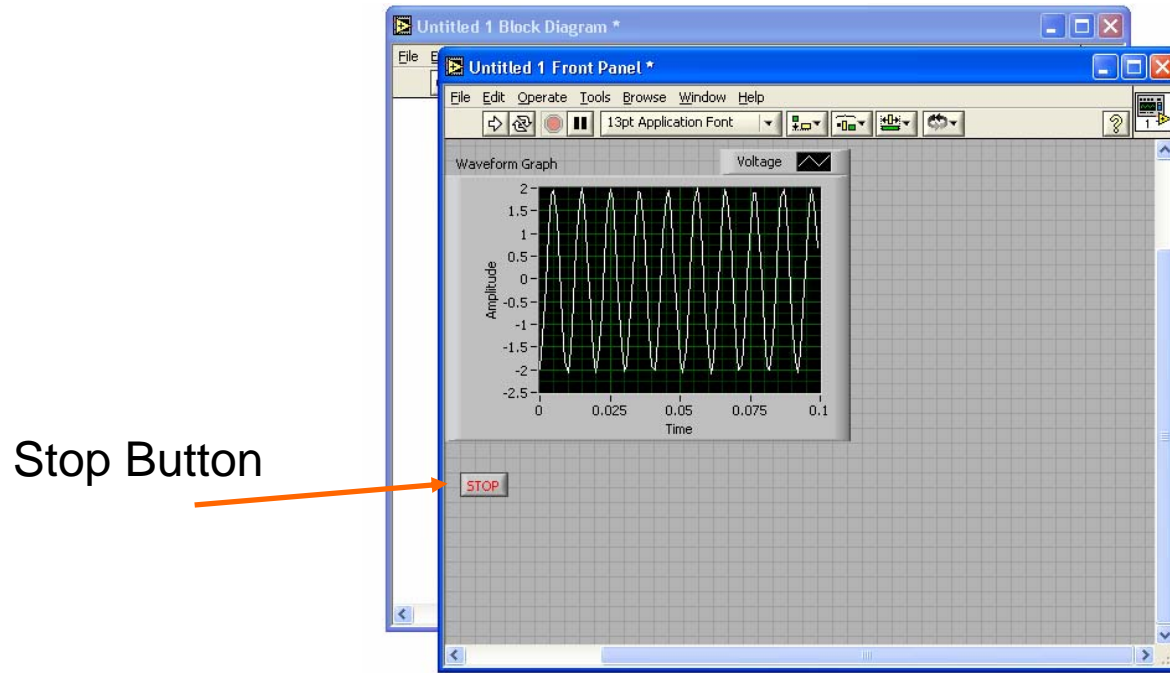
And enclose the code  
with the selection box  
and click in the lower  
right corner.

# Add While Loop



The While loop causes the code inside the loop to be executed until the stop condition is met. In the default situation, a stop button is connected to the while loop that will halt execution of the vi.

# Add While Loop



- If a stop button has been programmed in the vi, it should always be used instead of the Abort Button on the toolbar.
- Run the program and observe the output.

# Looped Execution

- Adjust the Amplitude on the function generator to produce a 4 V peak-peak signal.
- Notice:
  - The voltage scale will auto adjust to the signal being measured.
  - The scale can be changed with settings on the graph. Just right click the graph in specific places and explore.

# Range

- Compare this setting using the Oscilloscope.
- What is the RMS Reading on the Volt Meter? 1.41 VAC

# Continuous Acquisition

- The configuration we have appears to be taking data continuously, however if the data is plotted, a small discontinuity will be noticed between each 500 samples (or “samples to read” setting).
- True continuous acquisition can be performed by setting the *Task Timing* to continuous.
- When using continuous timing, connect the stop button to the DAQ Assistant so it will know when to halt taking data. This is an example of when it is not appropriate to stop the VI using the abort button. The stop button will cleanly shutdown all the components in the program.

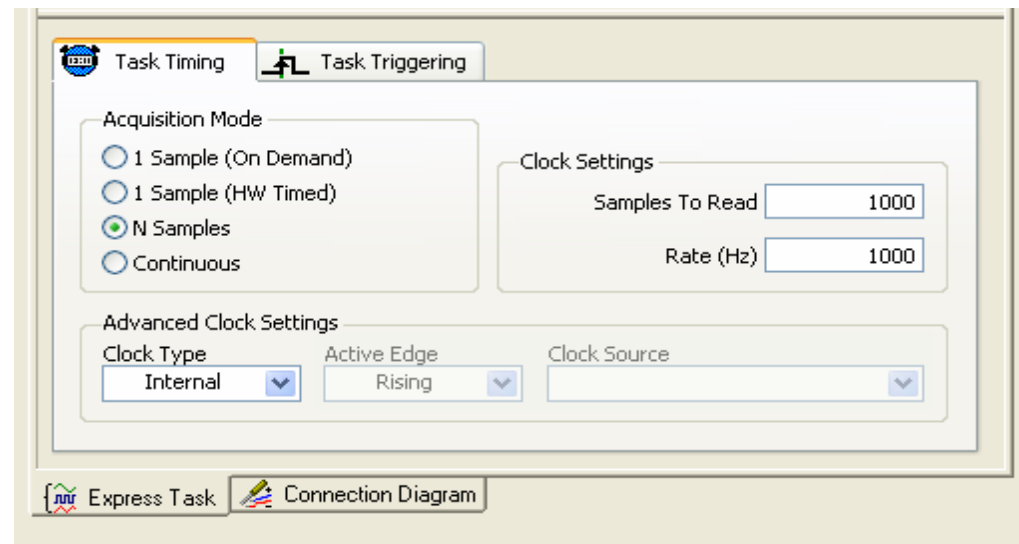


# NI USB 6211 Specs

- Max sample rate 250kS/s.
- Max voltage +/- 10 VDC.
- Resolution 16-Bits.
- Analog Input: 8 Diff, 16 Single Ended.
- 2 12-Bit Analog Outputs (0-5 Volts, 150 Updates/sec max).
- 12 Digital I/O lines.

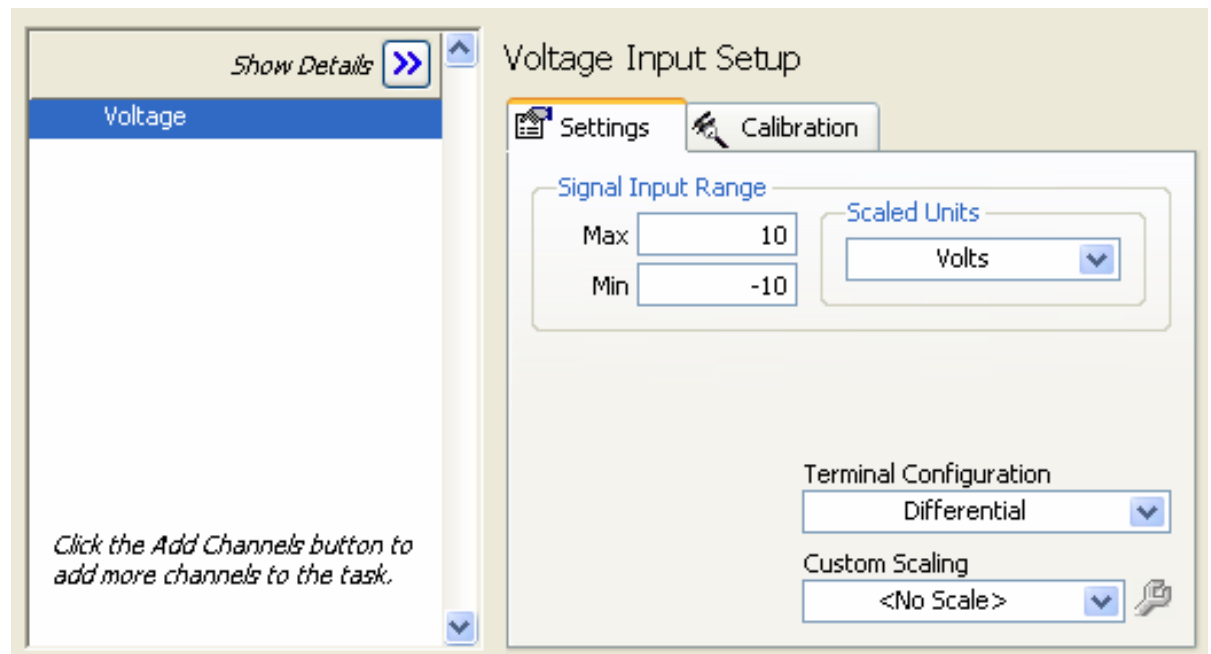
# Rate

- Max 250kS/s (250,000 Samples/sec) for NI USB 6211.
- The maximum rate the DAQ can function.
- Divided among channels (Multiplexed).
- Some slight Inter-channel delay.



# Range

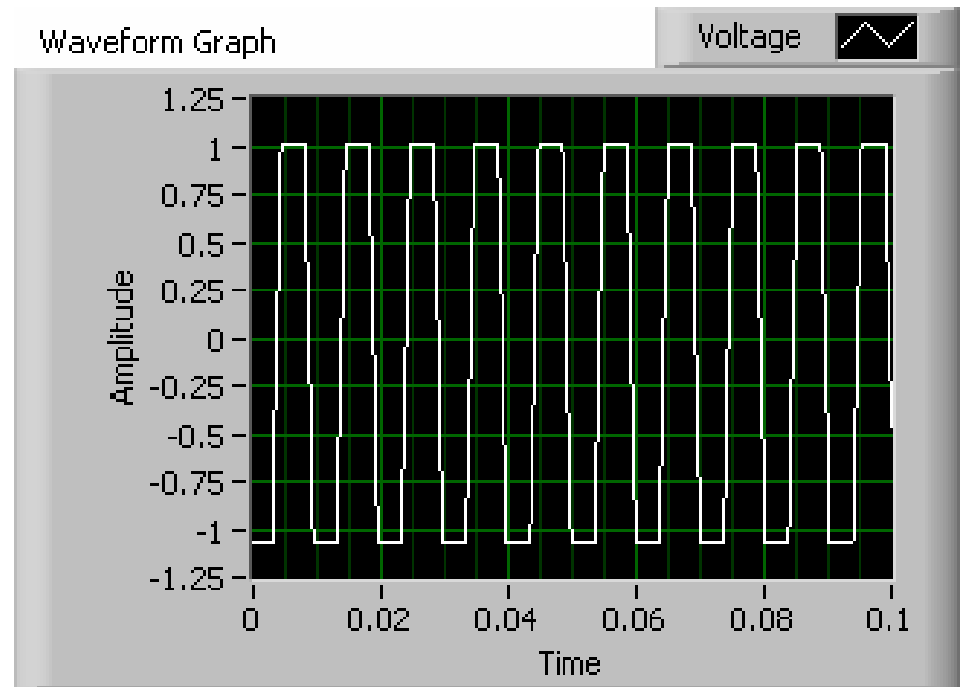
- The NI USB 6009 DAQ can be set to  $\pm 20$ ,  $\pm 10$ ,  $\pm 5$ ,  $\pm 4$ ,  $\pm 2.5$ ,  $\pm 2$ ,  $\pm 1.25$ ,  $\pm 1$  V input range.
- The Digital to Analog Converter changes the *analog voltage* into a *number* the computer can work with.
- 16 bits allow  $2^{16}$  or 65,336 **integer** numbers to be used.
- These numbers are spread over the **range** of the input voltage.



# Range Example

- Now change the voltage range to -1, +1 in DAQ Assistant Input Range.
- Observe the difference in the output.

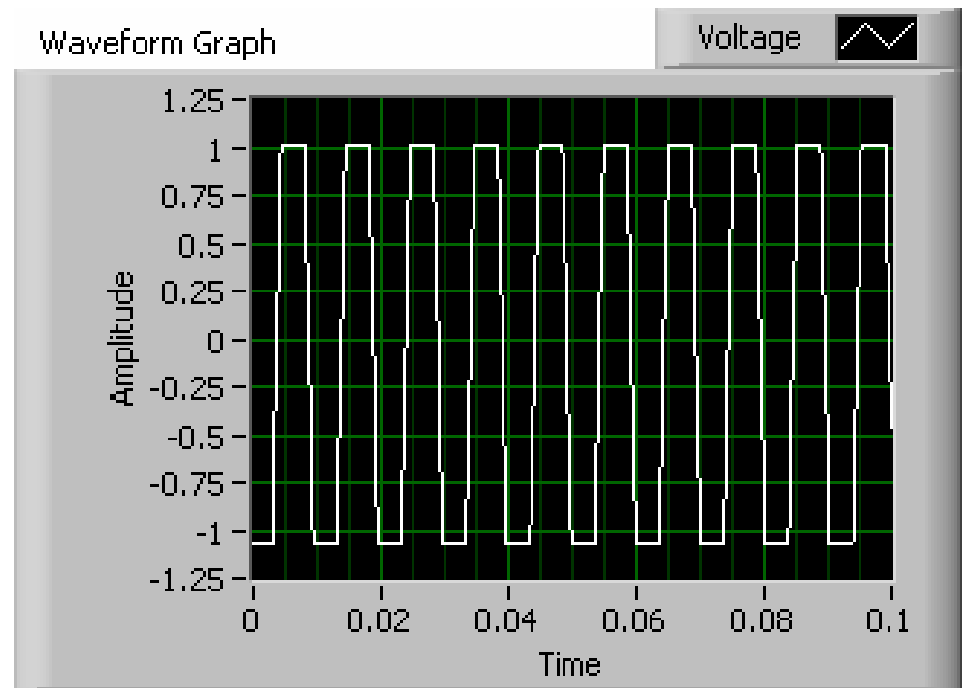
Notice the flat top and bottom of the waveform at 1 and -1 volt. No matter how high the input voltage, the result will not exceed the set range.



# Range Example

- Now change the voltage range to -1, +1 in DAQ Assistant Input Range.
- Observe the difference in the output.

Absolutely do not exceed the maximum input rating of the board!!!  
(NI USB 6211 Overvoltage protection: +/-10V)

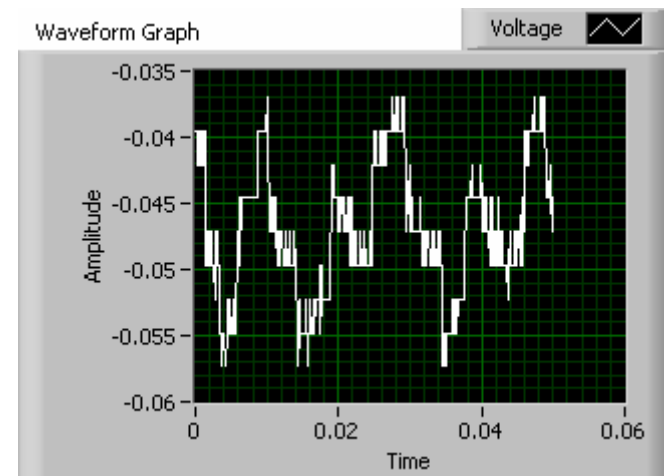


# Resolution

- **Resolution** is the voltage range one number can represent.
- Example:  $+20 - (-20) = 40$ ,  
 $40/16383 = .00244V$  or  $2.44mV$  resolution.
- Set the Range back to  $\pm 20V$
- Adjust the amplitude of the function generator to the minimum value. (Use the 0-2 Vp-p Range button)

# Resolution Example

- This is typical of a DAQ input that is not using very much of it's range.
- You can see the step like pattern that should be an indication of this.



# Aliasing-Varying Sample Rate

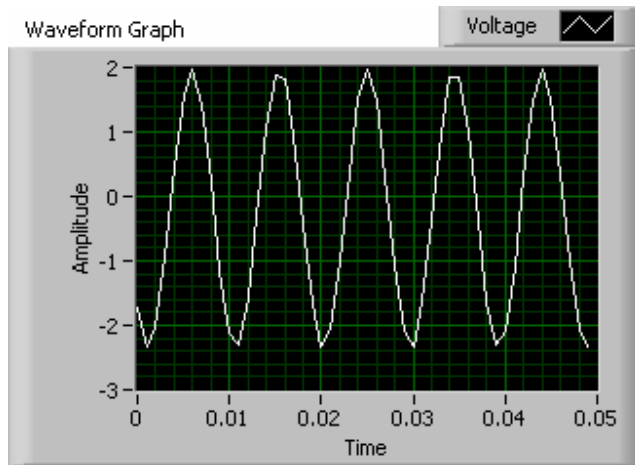
- Aliasing is when the DAQ is giving a result that is not what is really happening.
- Filters at the Nyquist frequency are the only way to prevent aliasing.
- Now set the function generator to
  - 2 Vp-p
  - 100 Hz
  - Sine Wave
- Set the DAQ to
  - 1000Hz sample rate
  - 50 Samples
- What is the sample time? 50mSec. Samples/rate
- How many wave cycles should we see? 5. Sample time/ (1/rate)



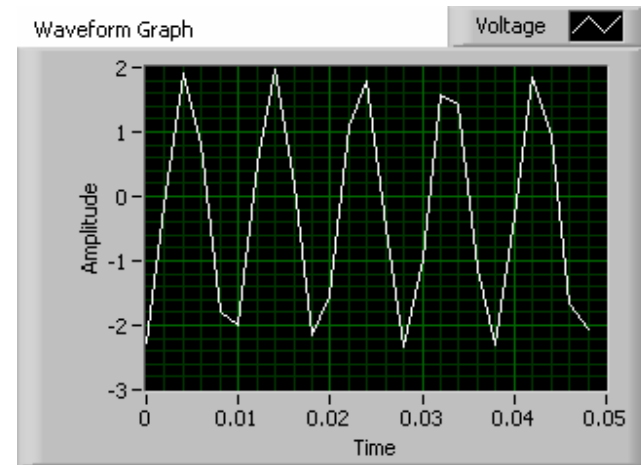
# Aliasing-Varying Sample Rate

- Now try
  - 1000Hz and 50 samples
  - 500Hz and 25 samples
  - 300Hz and 15 samples
  - 210Hz and 10 samples
  - Need Frequencies below Nyquist here:
    - 
    - 
    -

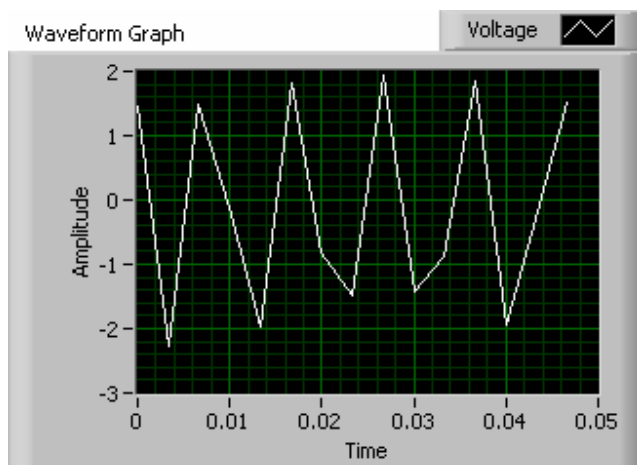
# 105 Hz signal at different sample rates above Nyquist Frequency



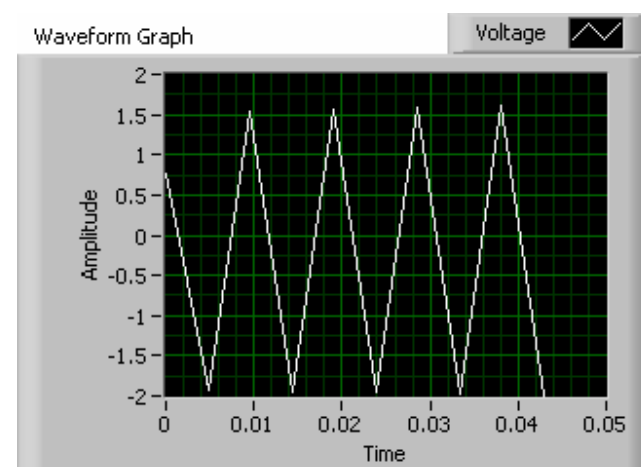
1000Hz sample rate, 50 samples



500Hz sample rate, 25 samples

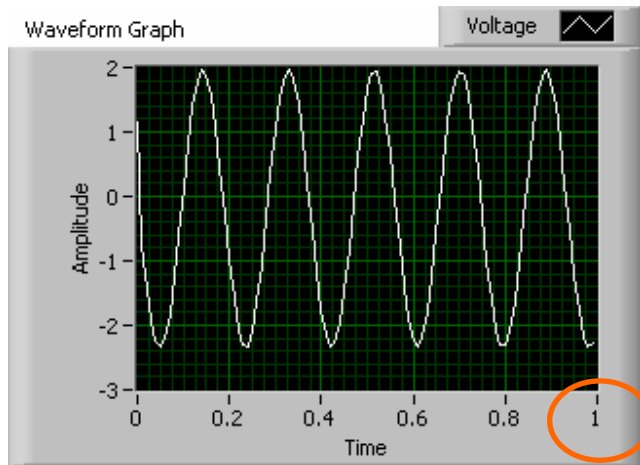


300Hz sample rate, 15 samples

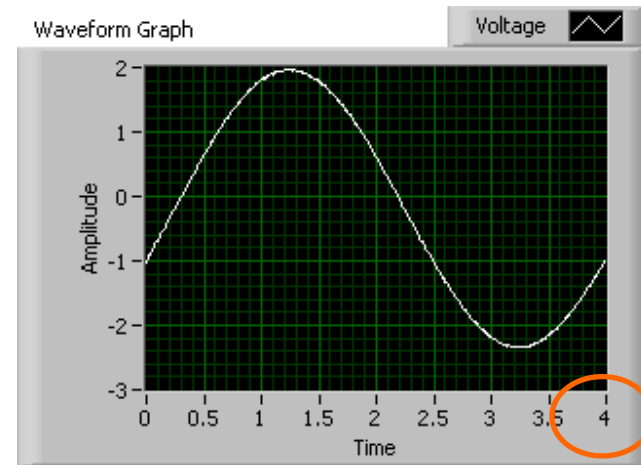


210Hz sample rate, 10 samples

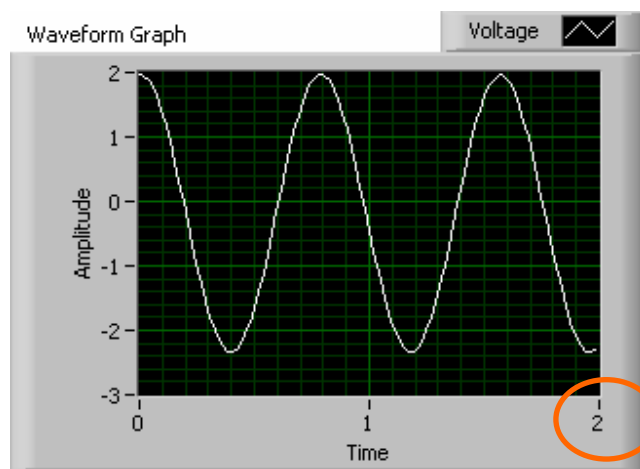
# 105 Hz signal at different sample rates below Nyquist Frequency



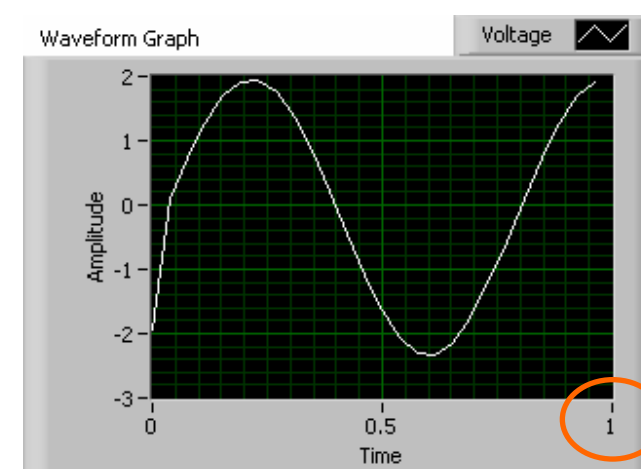
100Hz sample rate, 100 samples



105 Hz sample rate, 420 samples



52Hz sample rate, 100 samples



26Hz sample rate, 26 samples

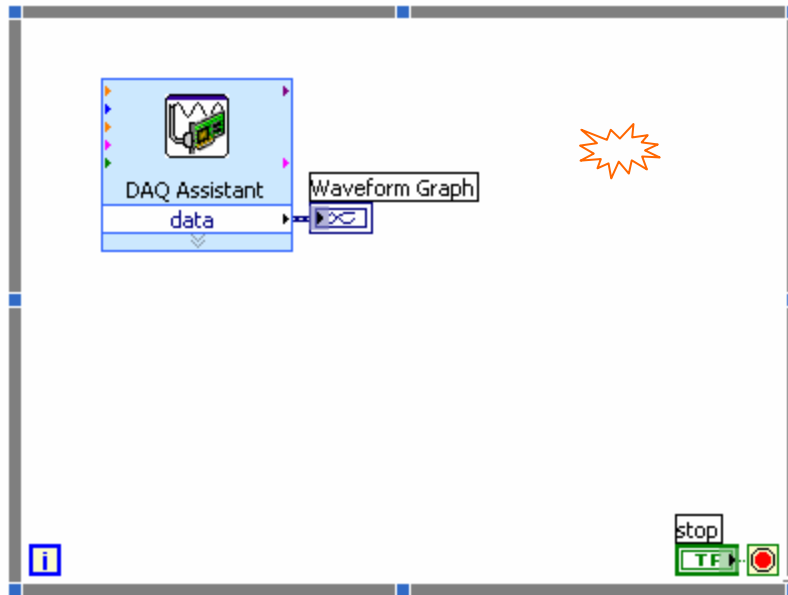
# Sampling Below Nyquist

- When the sample rate is less than half the frequency under measurement, aliasing or misinformation will result.
- The apparent frequency will not be the actual frequency.
- A perfect waveform can be observed that is likely some multiple of the actual frequency.
- Hardware filtering is the only way to avoid aliasing when the sample rate cannot be set high enough to include all frequency content being measured.

# Saving Data

- Data can be saved with the Write LabView Measurement File express VI.
- Add the **Write Meas File** Express VI inside the while loop.
- The Write Meas File VI is located in the *Output Express Pallet*.
- Observe all the options in the express dialog.
- The default options will get the data saved.
- Other settings can optimize the program, for instance it may be better to not have the save VI overwrite data that has already been taken!

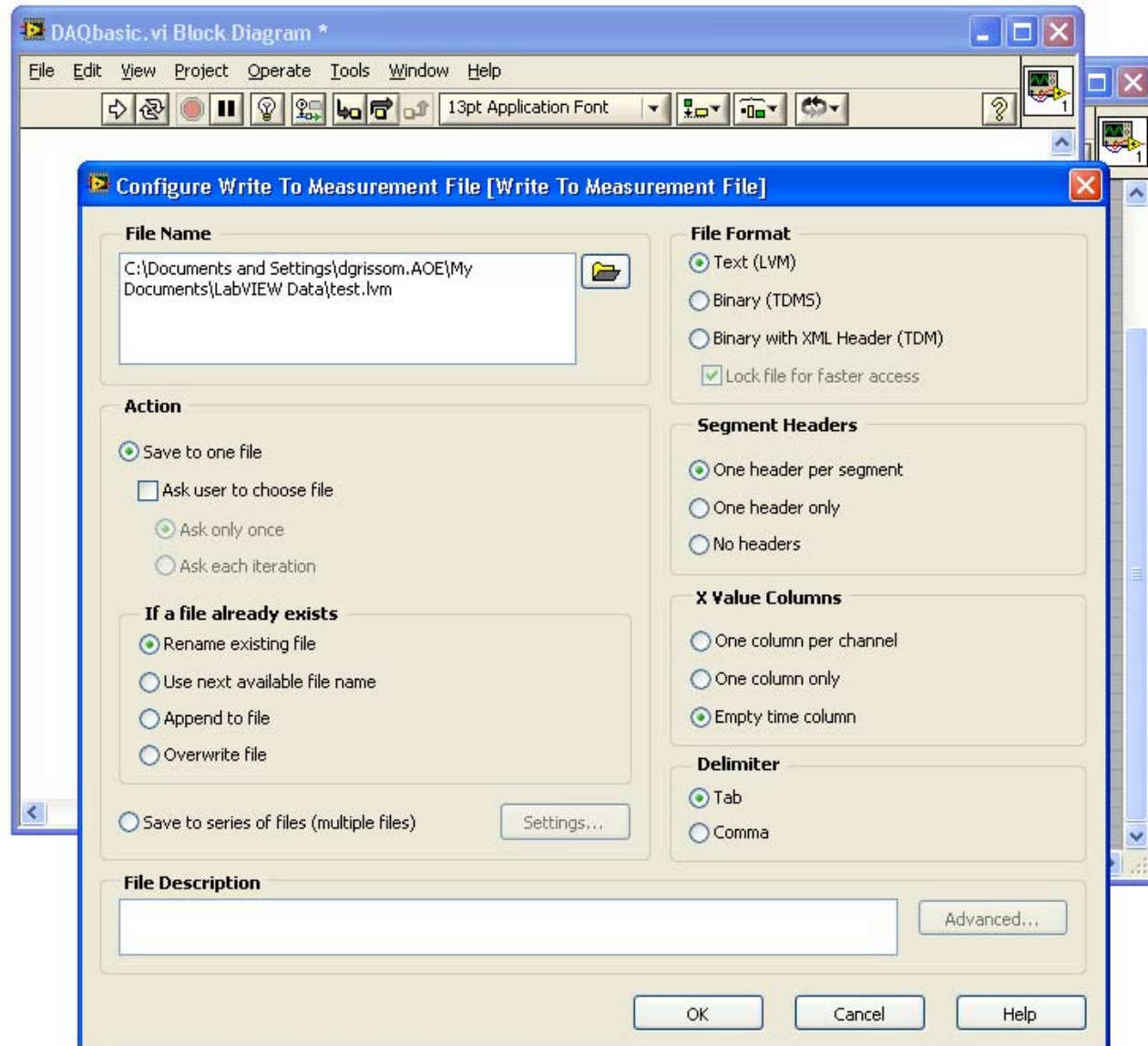
# Saving Data



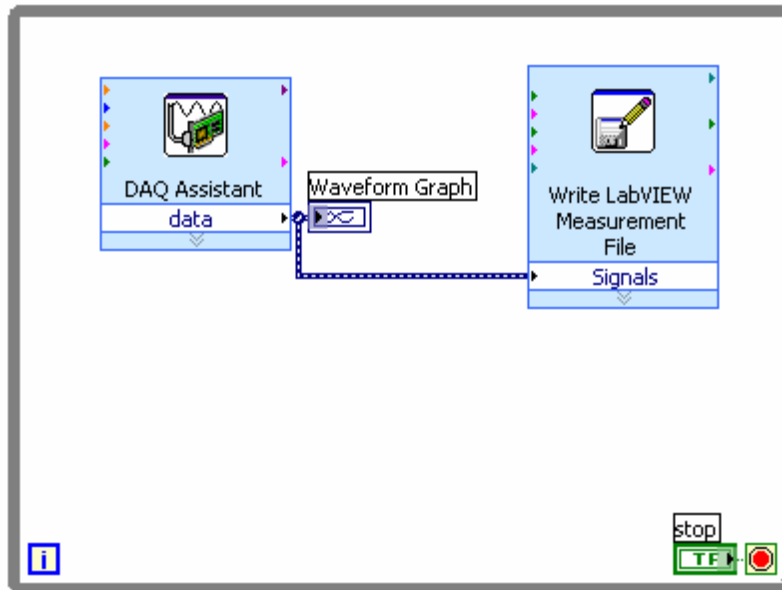
Drop the “Write to Measurement File.vi” to the diagram to the right of the DAQ assistant.

The Write to Measurement File.vi can be found under Output on the Express vi pallet.

# Saving Data



# Saving Data



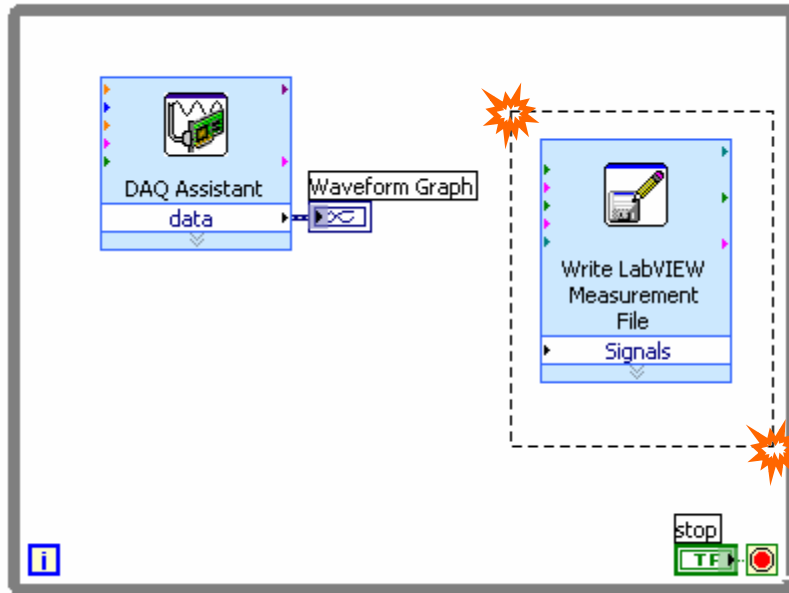
Connect the Signals Terminal on the Write to Measurement File to the wire that comes out of the DAQ Assistant.



# Save Data Button

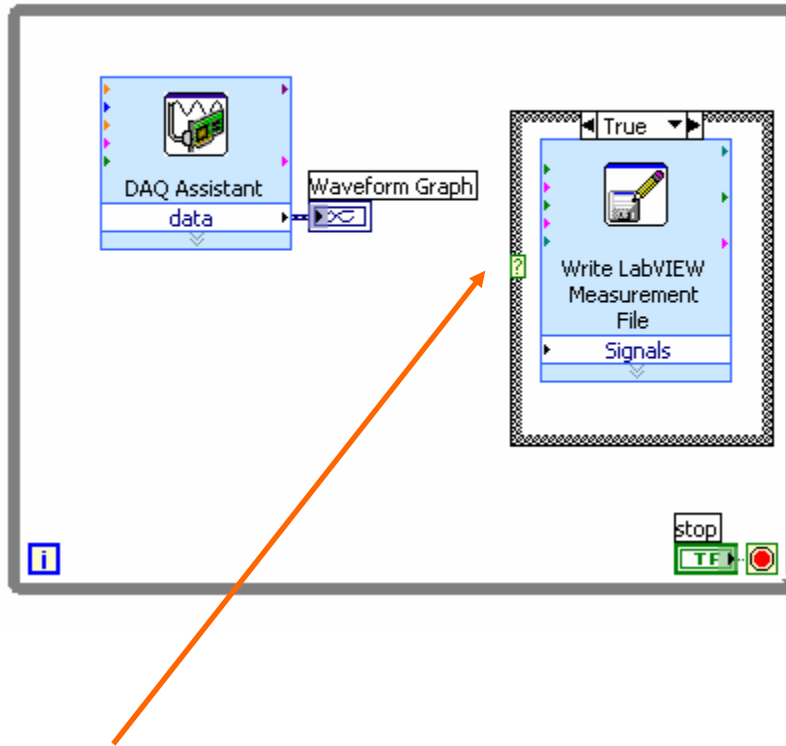
- In order to keep the program from saving data every cycle of the while loop, we should add a button to the front panel.
- The button will control a case structure that will decide if the save routine should run.
- Located on the *Exec Ctrl* express pallet, select the **Case Structure**.

# Add Case Structure



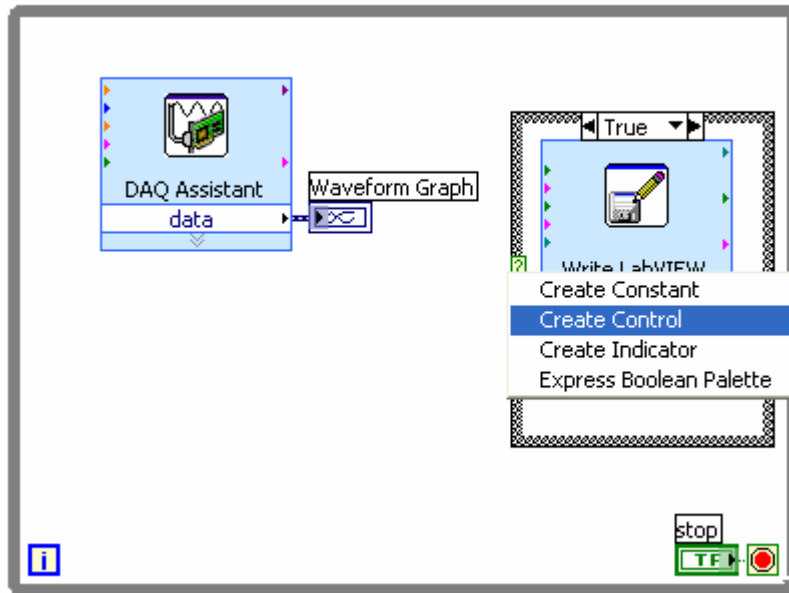
Click to surround the functions that should be included in the case structure. Include just the Write to Measurement file VI.

# Add Case Structure



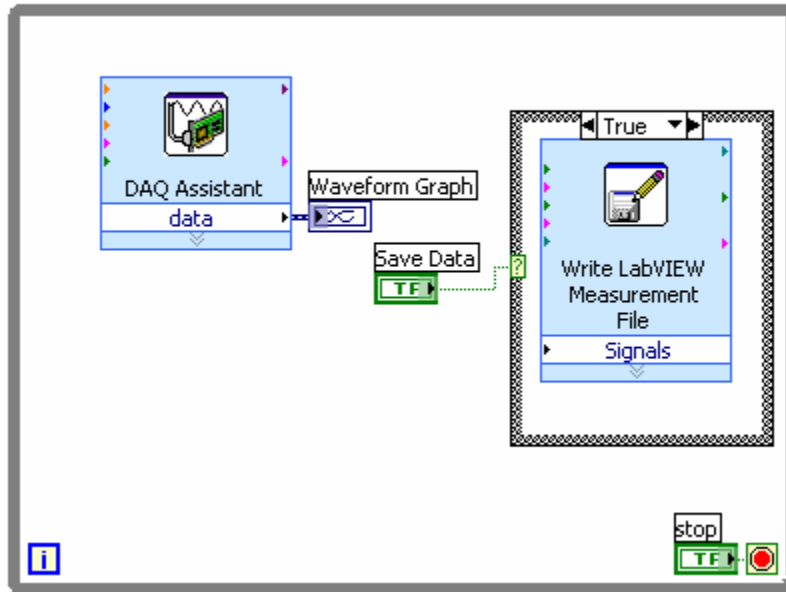
Now Right Click the left side of the decision terminal and select create control.

# Add Case Structure



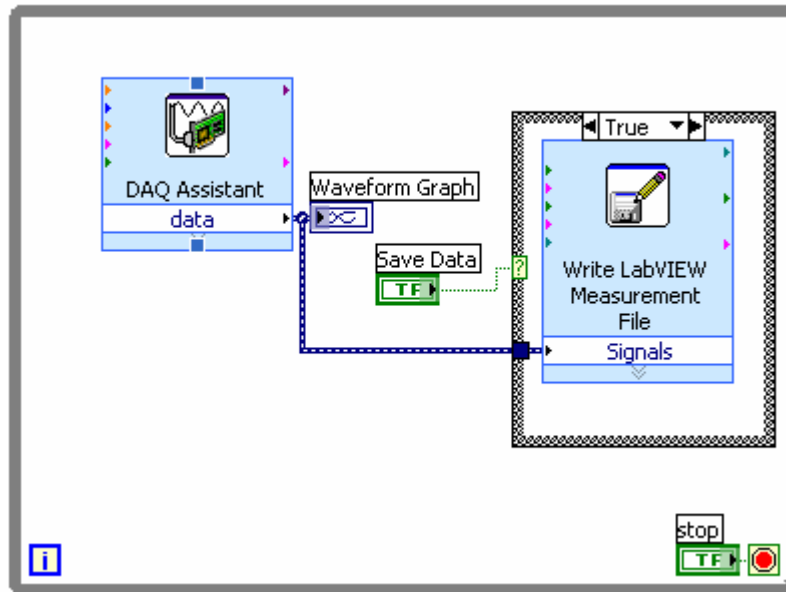
Create control will make a button to save data.

# Add Case Structure



And rename it to “Save Data”. Just start typing after selecting Create Control since the label is highlighted once it is created.

# Add Case Structure



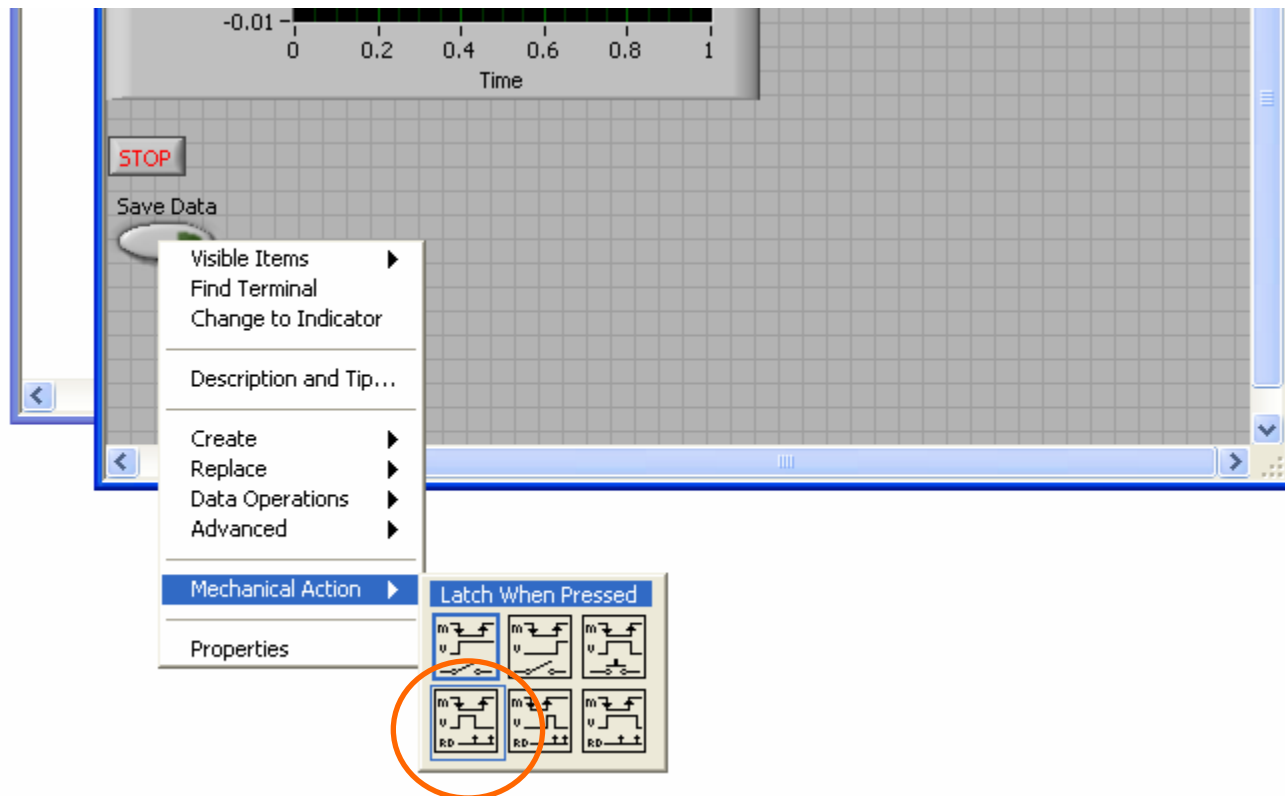
Connect the signals terminal to the DAQ output line.

# Boolean Button

- A button is normally a click on, and click off type.
- For this program, it is best if the button is turned off after it has been selected for an operation.
- From the front panel, right click the newly created button and select **Mechanical Action**

# Setting the action of a Button

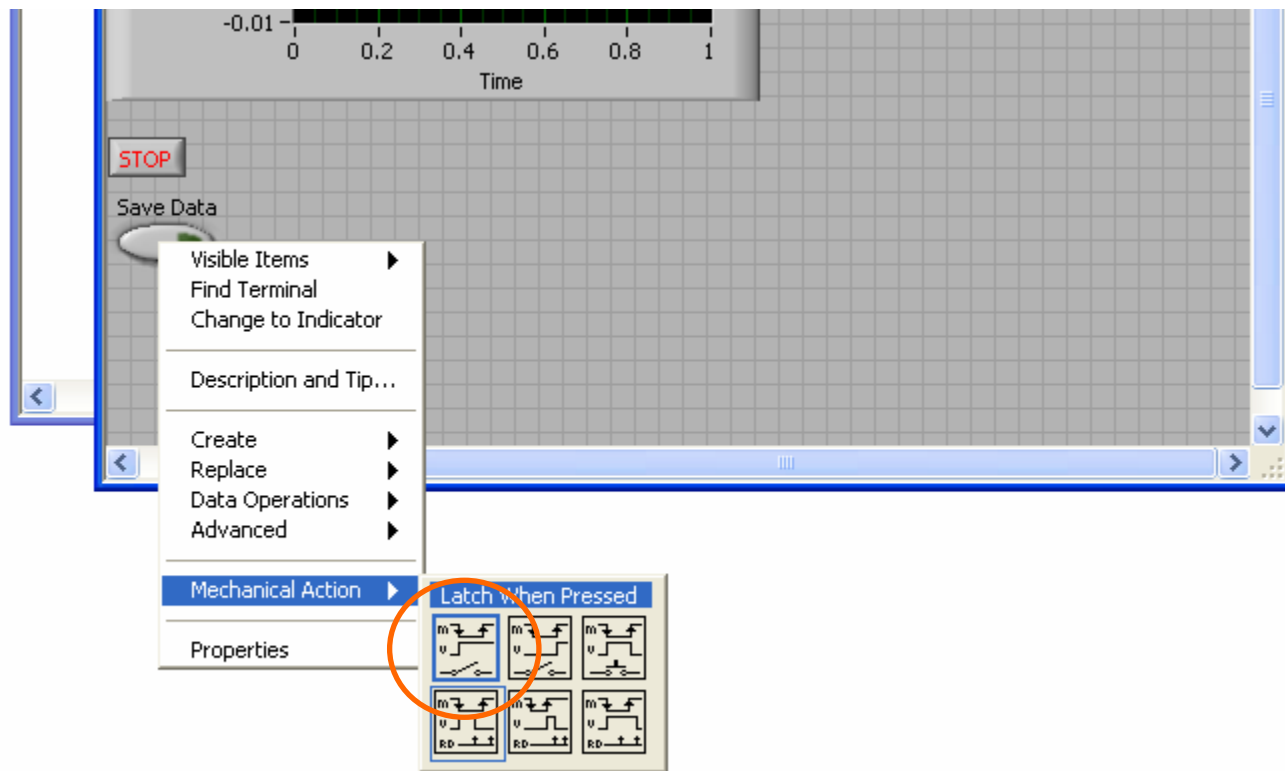
Set the action of the button to “Latch When Pressed”. This will allow the program to turn off the button once the desired action has been performed once. This is a kind of one-shot.





# Setting the action of a Button

If you want the program to save data as long as the button is pressed, leave it set to “switch when pressed”. If this is done, remember to turn it off or an extremely large file could result. The file save settings will need to be adjusted for desired results.



# Examining Data File

- Run the program and save some data.
- Open the file with Excel and look at the data it recorded.

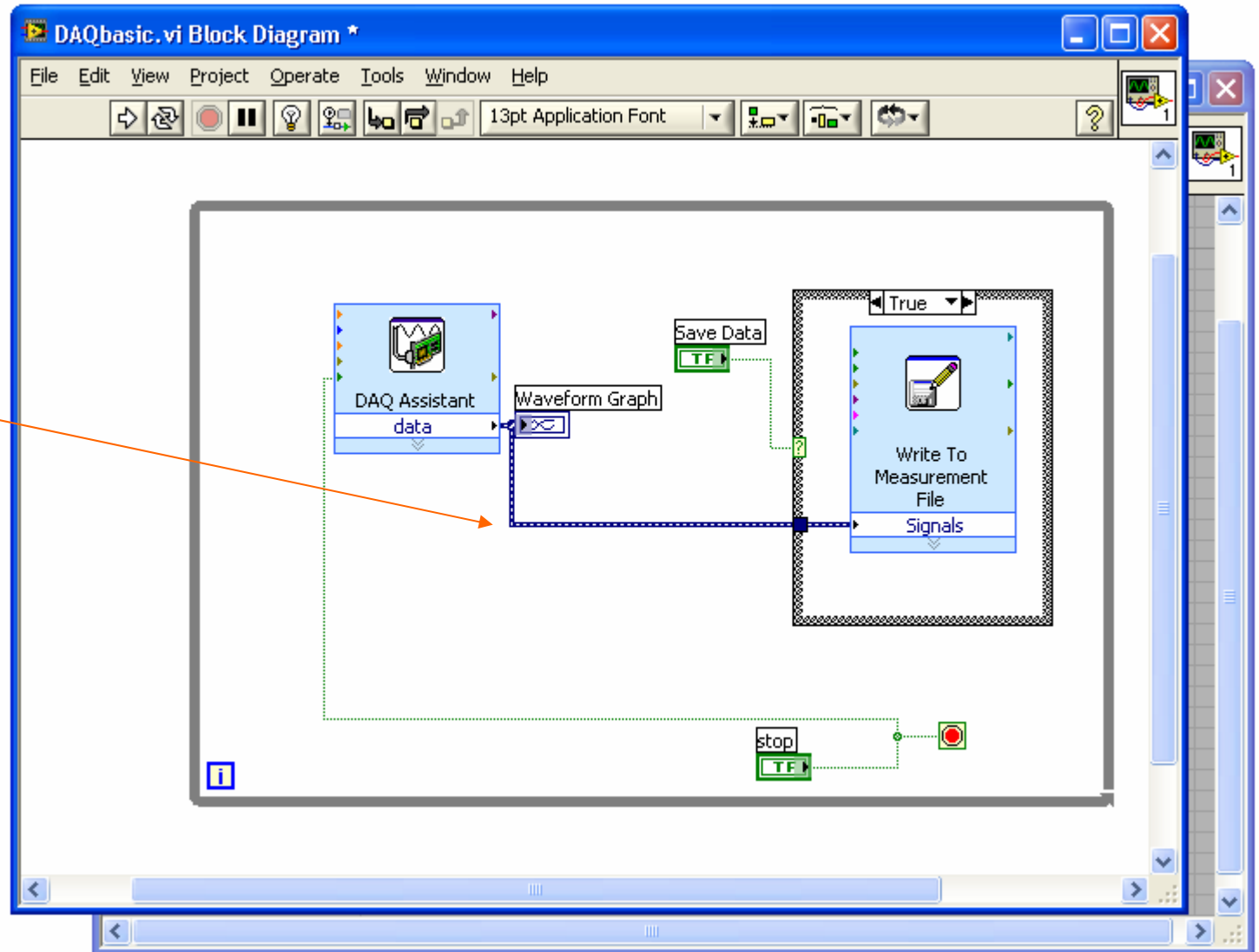
	A1	LabVIEW Measurement		
	A	B	C	
1	LabVIEW Measurement			
2	Writer_Version	0.92		
3	Reader_Version	1		
4	Separator	Tab		
5	Multi_Headings	Yes		
6	X_Columns	Multi		
7	Time_Pref	Absolute		
8	Operator	LabUser		
9	Date	3/6/2006		
10	Time	27:55.1		
11	***End_of_Header***			
12				
13	Channels	1		
14	Samples	1000		
15	Date	3/6/2006		
16	Time	27:56.2		
17	Y_Unit_Label	Volts		
18	X_Dimension	Time		
19	X0	0.00E+00		
20	Delta_X	0.001		
21	***End_of_Header***			
22	X_Value	Voltage	Comment	
23	0	-0.002691		
24	0.001	0.007495		
25	0.002	0.002402		
26	0.003	-0.002691		
27	0.004	0.007495		
28	0.005	0.002402		
29	0.006	-0.000144		
30	0.007	0.004948		

# Multiple Analog Input Signals

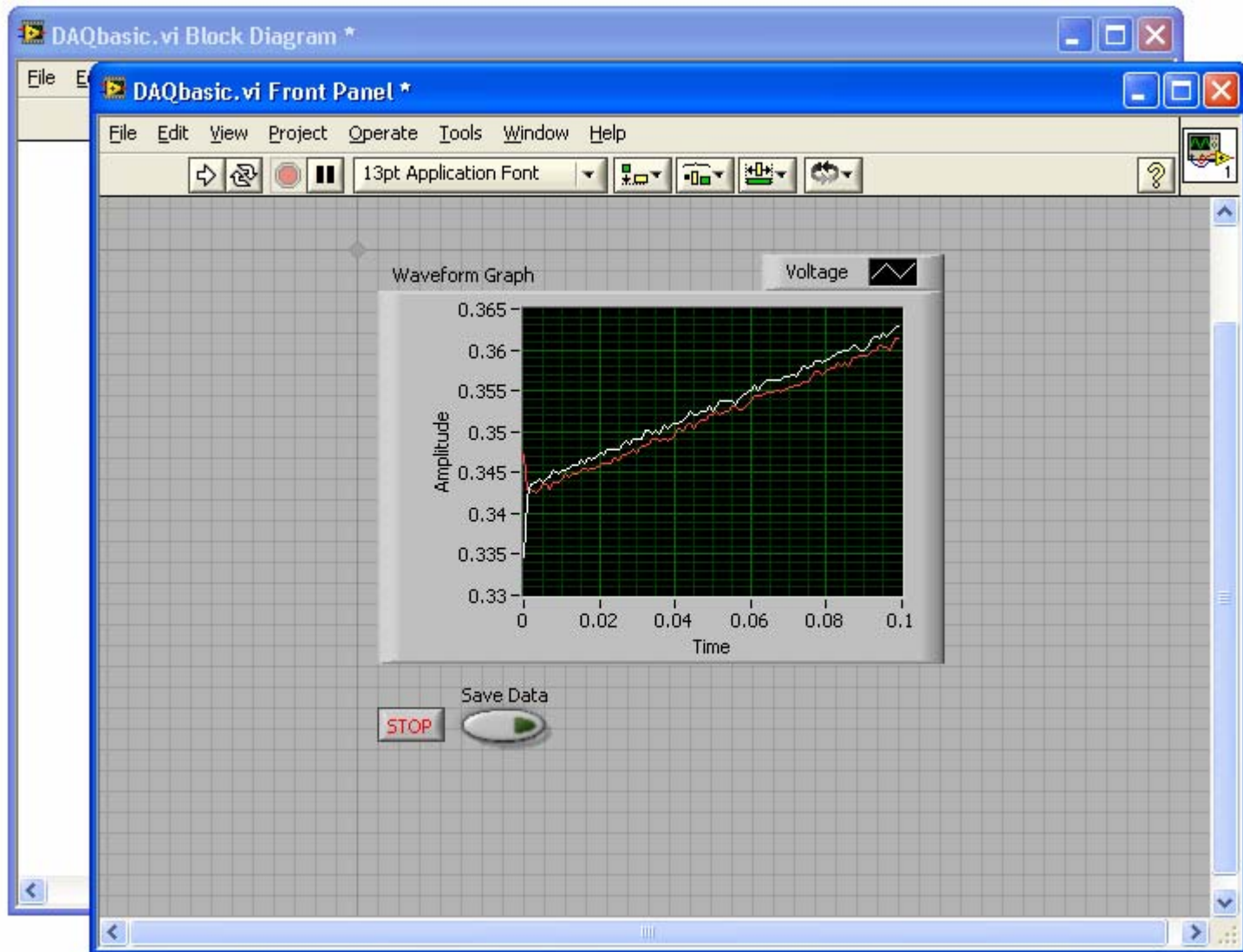
- Add a second channel to the DAQ express vi.
  - Double Click the vi
  - Push Add Channels
  - Select Voltage
  - Select the channel(s) to be added.
- The channels can be renamed in the express vi dialog. (right click on them.)

# Where is our second channel

The blue wire is for a dynamic data type. The second signal is simply bundled in with the first.



# Check the signal from both channels



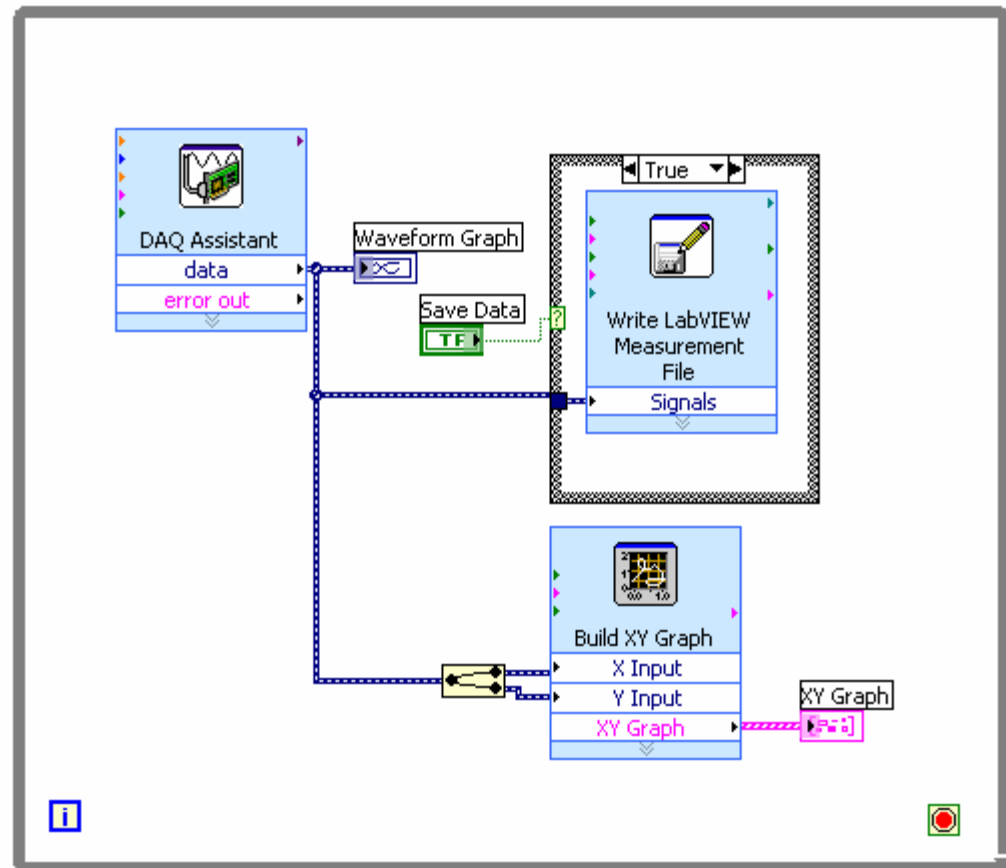
# X-Y Graphs

- Add the X-Y Graph to the display with CH0 on one axis and CH1 on the other to the front panel.

The image shows a LabVIEW front panel with a 'Waveform Graph' and a 'Controls' palette. The Waveform Graph has two y-axes: 'Excitation V' on the left (ranging from 0.28 to 0.31) and 'Response V' on the right (ranging from -6.5 to -6.16). The x-axis is 'Time' (ranging from 1 to 1.25). The graph displays two waveforms: a green 'Excitation' waveform and a red 'Response' waveform. The Controls palette is open, showing a search bar and several categories: 'Num Ctrls', 'Buttons', 'Text Ctrls', 'User Ctrls', 'Graph Indicators', and 'All Controls'. The 'Express XY Graph' sub-palette is also open, showing three options: 'Chart', 'Graph', and 'XY Graph'.

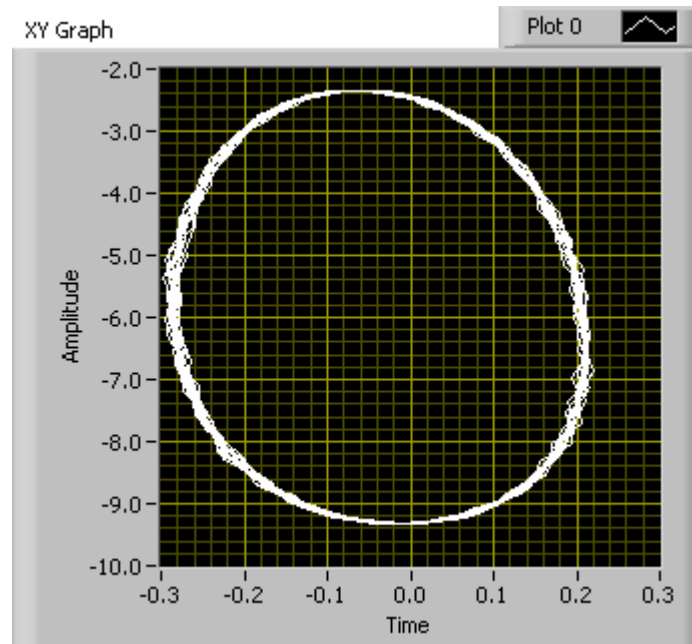
# X-Y Graphs

- LabView adds the express vi to create the X and Y values and it adds the graph indicator.
- Under signal manipulation, add a “Split Signals” and connect the output from the DAQ Assistant vi.



# X-Y Graphs

- Using this with the dynamic Beam can give the representation like the scope X-Y Mode.





# Applications to Dynamic Beam

- Two inputs allow force and displacement voltages to be measured.
- Voltages converted to force and distance.
- Dynamic flexibility and spring constants measured at low frequencies and the result displayed.
- Data file saved and analyzed.