LabVIEW HW #6 (a longer assignment!)

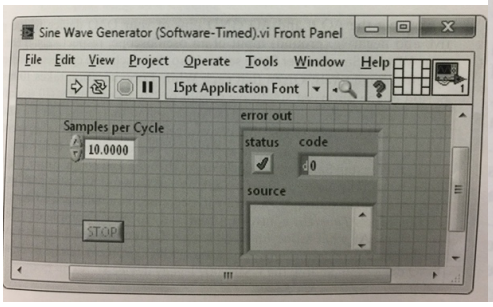
Each problem begins with a suggested descriptive name (including the *.vi* extension) for the solution VI that you will write. Suggested icons for use in the VI are given at the end of some problem statements. The palette locations of the cited icons are not give explicitly: these icons can be found with the aid of **Quick Drop**.

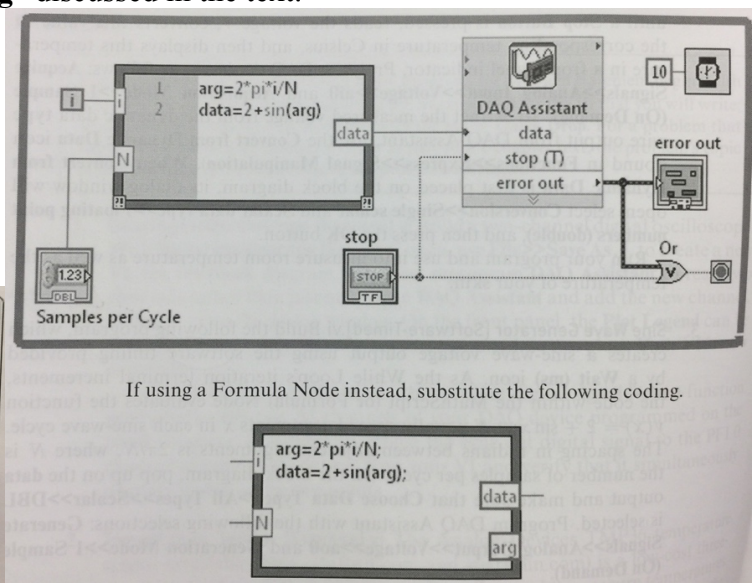
1. Complete the Do-It-Yourself/"USE IT!" section ending the chapter on Data Acquisition Using DAQ Assistant.

2. **Dual Digital Oscilloscope (Express).vi** Build a two-channel digital oscilloscope. Open up the **Digital Oscilloscope (Express)** VI that you constructed while working through this chapter. Use **File>>Save As…** to create a new VI. On the block diagram, reprogram **DAQ Assistant** to read *two* channels rather than just one: open **DAQ Assistant** and add the new channel, and then close the dialog window. (That was easy!) On the front panel, the **Plot Legend** can be expanded using a Resizing Cursor to accommodate two plots. Your VI is now complete. [Remember this method, as it is quite common to want to make measurements from more than one analog input, and this is far better than trying to run multiple copies of DAQ Assistant.]

   Attach, to the two AI channels of your (real) DAQ device that you have programmed into the block diagram, a voltage waveform and the sync out digital signal, from a (real) function generator in the lab. Additionally, for triggering, attach the sync out digital signal to the PFI 0 channel of your DAQ device. Run your VI and verify that it simultaneously displays traces of the two inputs.

3. **Sine Wave Generator (Software-Timed).vi** (This problem can be done with *any* DAQ device, including those incapable of "hardware timing.") Build the following program to create a sine-wave voltage output using a **Wait (ms)** icon, which provides the (***less*** precise/accurate) "**software timing**" discussed in the text:





Whenever the While Loop's iteration terminal increments, the code within the MathScript Node (or Formula Node) will evaluate the function $y(x) = 2 + \sin(x)$, at N equally spaced values of the argument, x, in each sine-wave cycle. That is, the spacing, in radians, between adjacent values of the argument, x, is $2\pi/N$, where N is the number of samples per cycle. – To ensure double precision, on your block diagram, pop up on the **data** output and make sure that **Choose Data Type>>All Types>>Scalar>>DBL** is selected. Set up DAQ Assistant with the following selections: **Generate Signals>>Analog Output>>Voltage>>ao0**, and **Generation Mode>>1 Sample (On Demand)**.

Next, connect the AO 0 and AO GND pins of your DAQ device to the central pin and outer shell (ground) inputs, respectively, of a real oscilloscope in the lab. Then run your VI with **Samples per Cycle** set to 10 (meaning that each sine cycle will contain ten points). Given that each While Loop iteration takes 10 ms, the period of the observed sine wave should be:

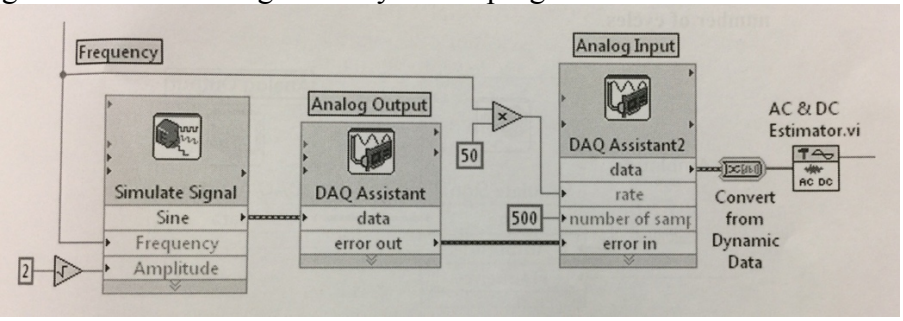
(10 ms/update) × (10 updates) = 100 ms, making the frequency 10 Hz.

On the oscilloscope, you should find that, although the underlying sine pattern is apparent, the ten voltage levels used to form this waveform are too well resolved to give the *appearance* of a continuous function. You can do a bit better by increasing **Samples per Cycle** to 20, then 30, and so on. Try it. – Since the While Loop iteration time remains constant, as you increase **Samples per Cycle**, the frequency of the output waveform will *decrease*. You will need to adjust your oscilloscope's *Time*/*Div* control accordingly to view a complete cycle of the waveform. Question: at what value of **Samples per Cycle** does the output waveform start to take on a continuous appearance?

4. **Bode Magnitude Plot.vi** (This problem requires a DAQ device capable of more accurate / precise "**hardware timing**," such as the USB-6211, the ELVIS II, or *any* of the PCI devices, such as the PCI-6251 and the ELVIS I).

This VI is designed to do the following: Apply an AC voltage with frequency f and 1 V rms amplitude to the input of any electrical circuit whose output, as a function of frequency, is to be mapped out. (Below, we'll start with a basic RC circuit, but this is a useful program that you will use over and over again, in the lab, for many different circuits that you build.) At each frequency, f, your DAQ should measure the resulting rms voltage amplitude V_{out} at the output of the circuit being tested, and then your program should calculate the Gain (in decibels): $G = 20 \log(V_{out}/V_{in})$, where, in this case, $V_{in} = 1\ V_{rms}$. This process is repeated for frequencies in the range $f = 10$ Hz to $f = 2010$ Hz, in increments of 25 Hz, and the resulting readings are displayed on a "Bode plot," which is a log-log plot displaying G *vs.* $\log(f)$, used to delineate the "useful operating range" ("bandwidth") of any electrical circuit.

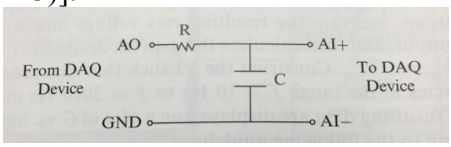
Here is some guidance for creating this very useful program:



a. **Block Diagram**: The workhorse of this VI is formed by configuring three Express Vis as shown above. Here, **Simulate Signal** is used to create a sine wave of frequency f and 1 V rms amplitude, which is passed to a **DAQ Assistant** configured to make this into a real signal by applying this waveform to an analog output channel of a real DAQ device. Assuming that you connect this signal to the input of an electrical circuit that you wish to test, the second **DAQ Assistant** can sample the circuit's output, and should be configured to read 500 samples at a sampling frequency of $50 \times f$ (*i.e.*, 10 cycles of the output will be read and made available at the data terminal). Finally, these 10 cycles are supplied to an icon called **AC & DC Estimator.vi**. Note: the icon called **Convert from Dynamic Data** will be automatically generated when you wire the second DAQ Assistant to the AC & DC Estimator.vi. A dialog window for Simulate Signal will allow you to set the **Samples per second (Hz)>> 100000** or, if you are using a slower DAQ device, whatever the maximum sampling rate is (*e.g.*, **10000**, for a USB-6008). For **Number of samples**, select both **Automatic** and **Integer number of cycles**. For the analog output DAQ Assistant, select **Generation Mode>>Continuous** and check **Use Waveform Timing**. For the analog input DAQ Assistant, select **Acquisition Mode>>N Samples**.

b. **Front Panel**: Place an **XY Graph** on the front panel, and supply it with bundled arrays of f and G from the block diagram. To obtain a log scale on the x-axis, pop up on the XY Graph and select **X Scale>>Mapping>>Logarithmic**. (Of course, by using decibels in your formula for the Gain, the y-axis is already logarithmic.)

c. When completed, let the first electrical circuit you test be the simple RC circuit shown below, where $R = 4.7$ kΩ (or so), and $C = 0.1$ μF (or so). [Record the nominal values of the components that you actually use, and then compare [the frequency where your data shows that the circuit's Gain has fallen by 3 dB] to [the (ballpark) characteristic frequency you would calculate from $1/(2\pi RC)$].



Such as circuit is called a "low-pass filter." By passing an analog signal though a low-pass filter prior to inputting it to a digitizer, the aliasing effect can be suppressed. [Often, though, one would use a slightly fancier low-pass filter, with a steeper roll-off than the simple ("first-order") circuit used here.]