Multi-Channel Data Acquisition System.

In this lab you will use Graphical Object Oriented programming techniques and Queued Menu Handlers with Menu Bar interface.

1) Open up MAX and make sixteen virtual input channels (labeled channel 0 to Channel 15). The channels should be in volts, single ended, non-referenced. You can do this quickly by first defining channel 0, then using ‘duplicate’ (right hand click on the channel 0 virtual channel) and select ‘increment channel numbers’ and make 15 copies. Close MAX once this is done.

2) Use the VISTA tools to generate a new Class called DAQClass. The Objects of the Class should be a Boolean control called Running?, a Waveform output, a channel name [string], a scan rate control [double], a channel control [DAQ virtual I/O channel], a device number control [U16], and a number of samples [u32], High Limit[double], LowLimit[double]. Close the class.

3) Now create the following methods from the ReadWrite Class Template:
   - DAQClassInitDevice (sets the device number)
   - DAQClassInitName (sets the channel name)
   - DAQClassInitScanRate (sets the scan rate)
   - DAQClassInitChannel (sets the DAQ virtual I/O channel)
   - DAQClassInitSamples (sets the number of samples)
   - DAQClassInitLimits (sets the high and low limits)
   - DAQClassGetWaveform (gets the waveform for the specified virtual channel and saves it to the Class Object Waveform; waveform does not need to be connected to a vi connector panel connector)

4) Close all objects, and create a new application from the Queued message Handler with Menu Bar interface template. Put controls on the front panel for each of the above initialization numbers (defined in 3). Also add a waveform display to display the output. Add some menu items for adding a new channel, deleting a channel. Add front panel buttons for run and stop, as appropriate.

5) The object of this system is to read analog data from the specified channel and display it to the screen. You should be able to add more channels (up to 15) or delete them, and select which channel you want to view by selecting the appropriate ‘combo box’ selection. This works the same way that the ’Bouncy ball example’ did in class, except that instead of creating/destroying individual bouncy ball objects you are creating/destroying DAQ channels.

6) Use the Queued menu interface, in a manner similar to class BouncyBall example, to create new channels, assign them unique identifiers, and record data from them (if they are running). Use the ‘Analog Input Multi-point’ DAQ function to record the data for each selected channel, passing along the high/low limits, number of samples, scan rate, etc for each time it is accessed. By analogy, there should be a function like BouncyBallMove which gets called if the event loop goes into a timeout, this is the above function DAQClassGet Waveform. So you need to make a Message called ‘getData’ which loops through each of the defined channels, records the data from the channel if the channel is running (otherwise keep the old data), and then send an Update Display Message to the Queued message Handler.

7) Your final code should be able to create and destroy DAQ objects (through the menu), and select the current one to view on the front panel graph by clicking on the various channels in the Combo box. If you select a different Channel in the combo box be sure to update all the controls on the front panel with their proper values as well as the waveform chart. You should be able to sort the waveforms in the combo box up and down, create new. Destroy old, as we did in class with Bouncy Ball, using the run-time menu. Be careful about the cases where only one channel has been defined, and where you have defined all 15 channels!